servicenow

ServiceNow

ServiceNow Fundamentals

participant guide

ServiceNow provides this document and the information therein “as is” and ServiceNow assumes no responsibility for any inaccuracies. ServiceNow hereby disclaims all warranties, whether written or oral, express or implied by law or otherwise, including without limitation, any warranties of merchantability, accuracy, title, non-infringement or fitness for any particular purpose.

In no event will ServiceNow be liable for lost profits (whether direct or indirect), for incidental, consequential, punitive, special or exemplary damages (including damage to business, reputation or goodwill), or indirect damages of any type however caused even if ServiceNow has been advised of such damages in advance or if such damages were foreseeable.

TRADEMARKS

ServiceNow and the ServiceNow logo are registered trademarks of ServiceNow, Inc. in the United States and certain other jurisdictions. ServiceNow also uses numerous other trademarks to identify its goods and services worldwide. All other marks used herein are the trademarks of their respective owners and no ownership in such marks is claimed by ServiceNow.

ServiceNow Fundamentals

Table of Contents

Module 1: User Interface & Navigation 7
- Lab 1.1: ServiceNow Overview.....21
- Lab 1.2: Lists and Filters 43
- Lab 1.3: Forms.....62
- Lab 1.4: Branding 76

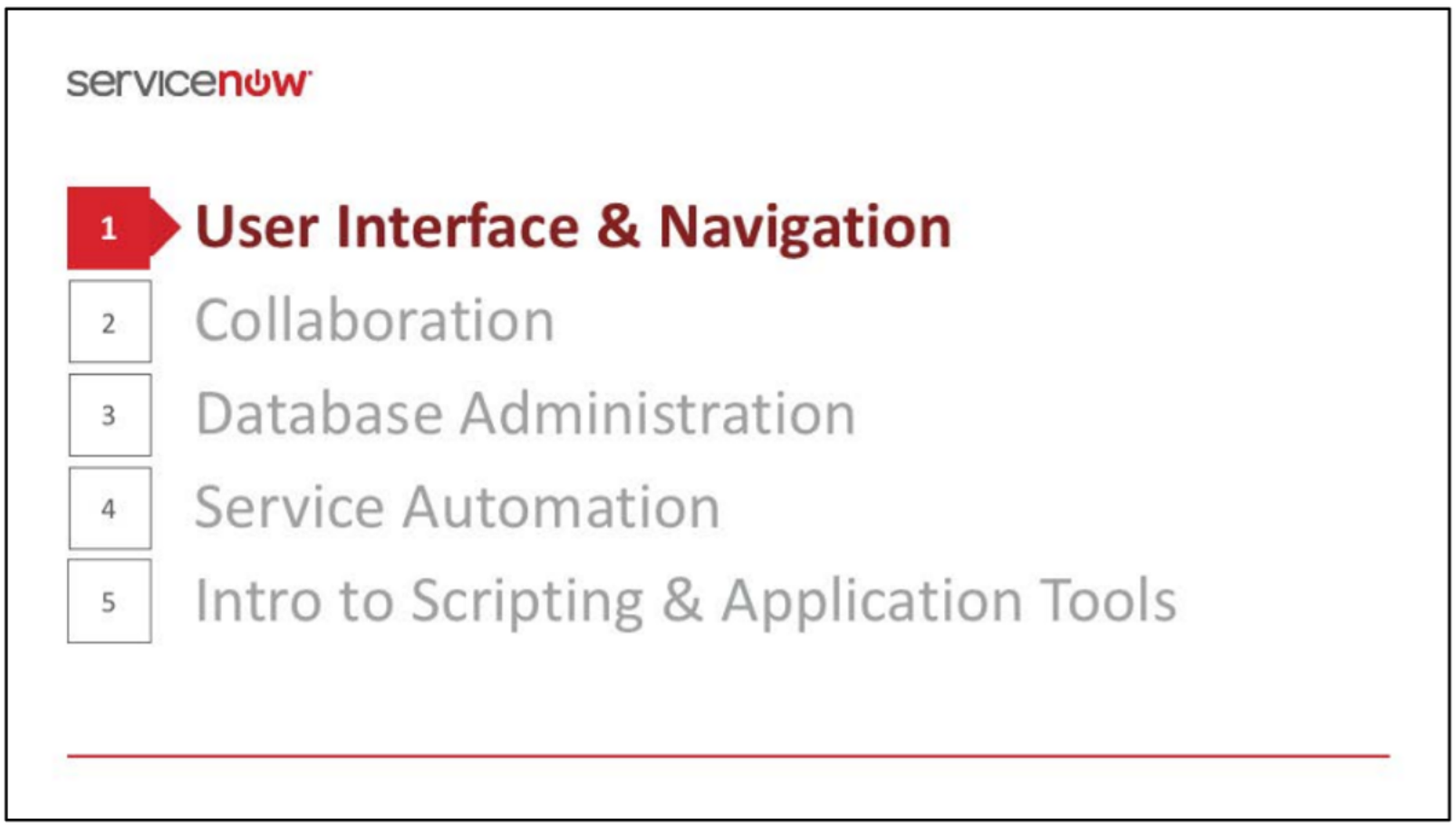
Module 2: Collaboration 81
- Lab 2.1: Task Management 96
- Lab 2.2: Notification.....111

Module 3: Database Administration 119
- Lab 3.1: Data Schema.....132
- Lab 3.2: Data Security 150
- Lab 3.3: Import Sets 163
- Lab 3.4: CMDB.....181

Module 4: Service Automation 189
- Lab 4.1: Service Catalog 200
- Lab 4.2: Flow Designer & Workflow.....216
- Lab 4.3: Service Level Agreements.....232
- Lab 4.4: Knowledge Management 242
- Lab 4.5: Reporting.....260

Module 5: Intro to Scripting & Application Tools.....267
- Lab 5.1: Scripting.....277
- Lab 5.2: System Update Sets.....288
- Lab 5.3: Development.....299

Module 6: Capstone Project.....307
- Lab 6.1: Capstone Project Challenge Format 311



ServiceNow Fundamentals Course Module Agenda

Module 1 – 1.1 ServiceNow Overview

servicenow

Objectives

- What is ServiceNow?
- What is the ServiceNow Instance?
- Users and Groups
- Key Platform UI Components
  - Application Navigator
  - Content Frame
  - Banner Frame
- Mobile Access
- Product Documentation

What is ServiceNow?

servicenow

ServiceNow is a cloud-based platform deployed in a browser that contains applications and data that can vary by instance and user, automating common business processes

A leader in Enterprise Service Management (ESM), the ServiceNow Service Automation Platform provides a modern, easy-to-use, service management solution in the cloud allowing your organization to automate manual repetitive setup tasks, manage your core IT processes, standardize service delivery, and focus on your core business, not just ITSM infrastructure.

ServiceNow provides all of this to users from a configurable web-based user interface, built on top of a flexible table schema.

The ServiceNow platform and the applications that run on it use a single system of record and a common data model to consolidate your organization's business processes.

Another advantage to this single system is that it can be leveraged to build custom applications.

The ServiceNow platform provides a Platform as a Service (PaaS), a cloud-based computing model that provides the infrastructure needed to develop, run, and manage applications.

It is not limited to a specific department or function but encompasses the entire enterprise.

What is the ServiceNow Instance?

servicenow

An **instance** is a single implementation of the ServiceNow platform

- Independent, changeable, and highly configurable
- Not shared with other ServiceNow customers (single-tenant)
- Each instance has applications
- Each instance has customer data that can be exchanged between instances
- Upgrades are made on individual instances

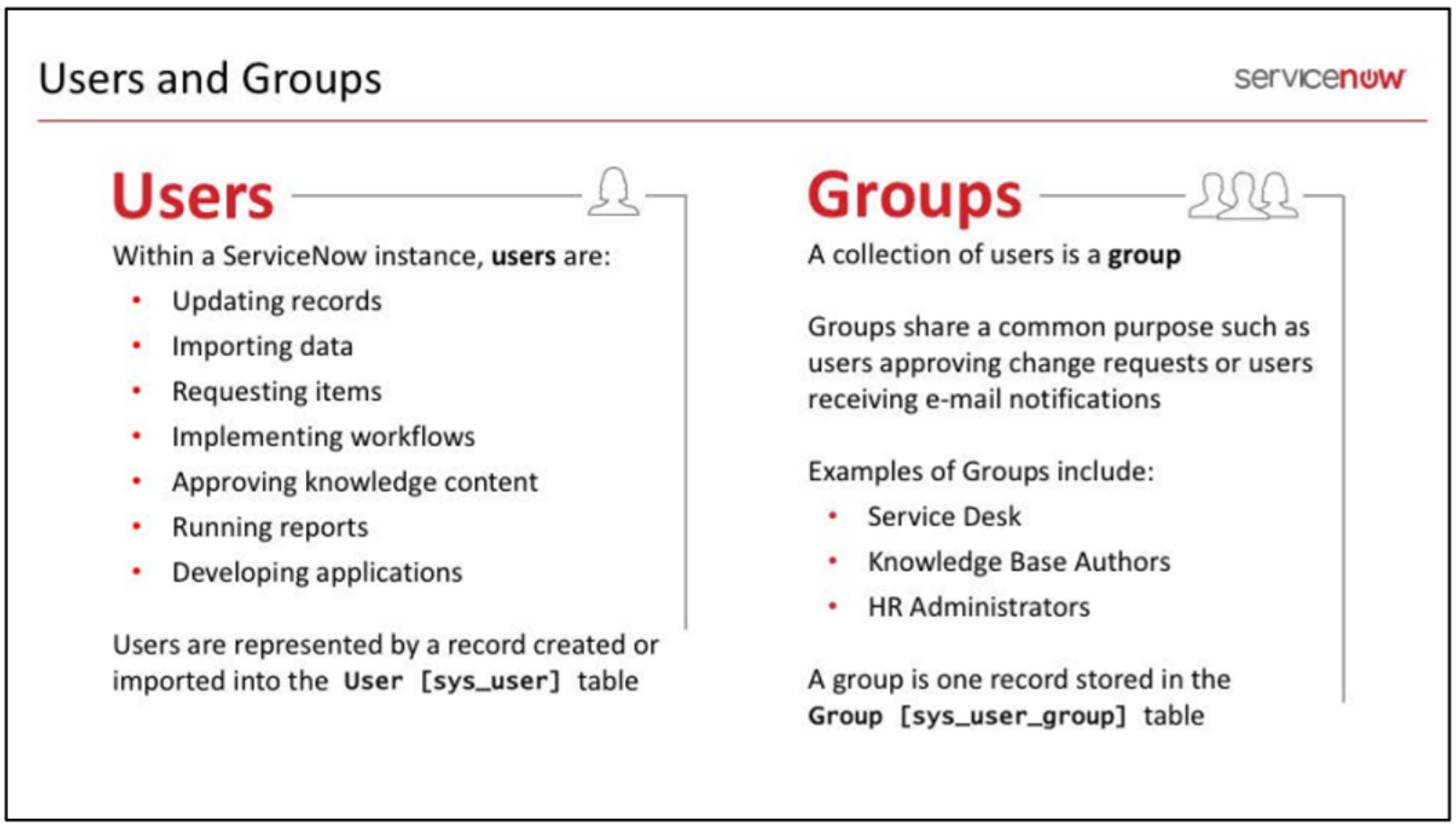
An instance is located (hosted) in one of the ServiceNow Data Centers around the world, or for a very, very small percentage of our customers, an instance can be implemented onsite at the customer's location. Each ServiceNow instance has a unique URL that uses a format similar to https://<instance name>.service-now.com/.

ServiceNow utilizes an advanced, multi-instance, single-tenant architecture as the default offering for customers, meaning an instance features an individually isolated database containing data, applications, and customizations.

The ServiceNow multi-instance architecture, organized in an instance stack, provides these distinct advantages:

- The multi-instance architecture allows ServiceNow to perform actions on individual customer instances such as performing an upgrade, on a schedule that fits the compliance requirements and needs of your enterprise.
- Data is truly isolated in their own databases, making hardware and software maintenance on these unique customer instances far easier to perform and issues can be resolved on a customer-by-customer basis.

Each customer organization receives two instances of ServiceNow: production and sub-production. They have the ability to obtain additional sub-production instances to be used for User Acceptance Testing (UAT), Review, Development, or Quality Assurance (QA).



Manage the individuals who can access ServiceNow by defining them as users in the platform.
NOTE: User names (represented by user IDs) are unique in ServiceNow.

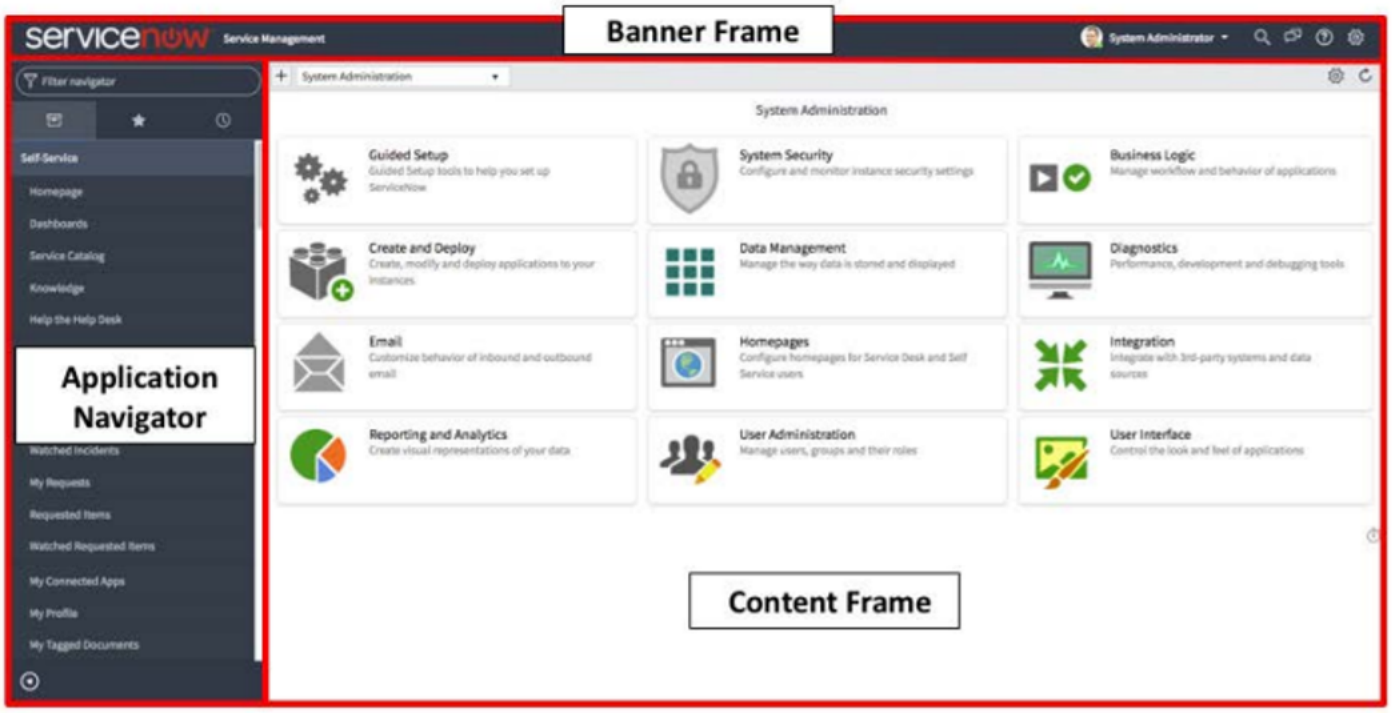
Users are authenticated by various methods, including:

- **Local database**: The user name and password in their user record in the instance database
- **Multifactor**: The user name and password in the database and a passcode sent to the user's mobile device that has Google Authenticator installed
- **LDAP**: The user name and password are accessed via LDAP in the corporate directory, which has a matching user account in the ServiceNow database
- **SAML 2.0**: The user name and password configured in a SAML identity provider account, which has a matching user account in the database
- **OAuth 2.0**: The user name and password of OAuth identity provider, which has a matching user account in the database
- **Digest Token**: An encrypted digest of the user name and password in the user record

User credentials are matched to different saved credentials for each method. Multiple Provider SSO allows the selection/use of several identity providers (IdPs) to manage authentication as well as retain local database authentication.

A group is part of the user hierarchy, and a user is part of a group. Groups may be imported from a corporate directory (LDAP) or created manually in ServiceNow.

Key Platform UI Components

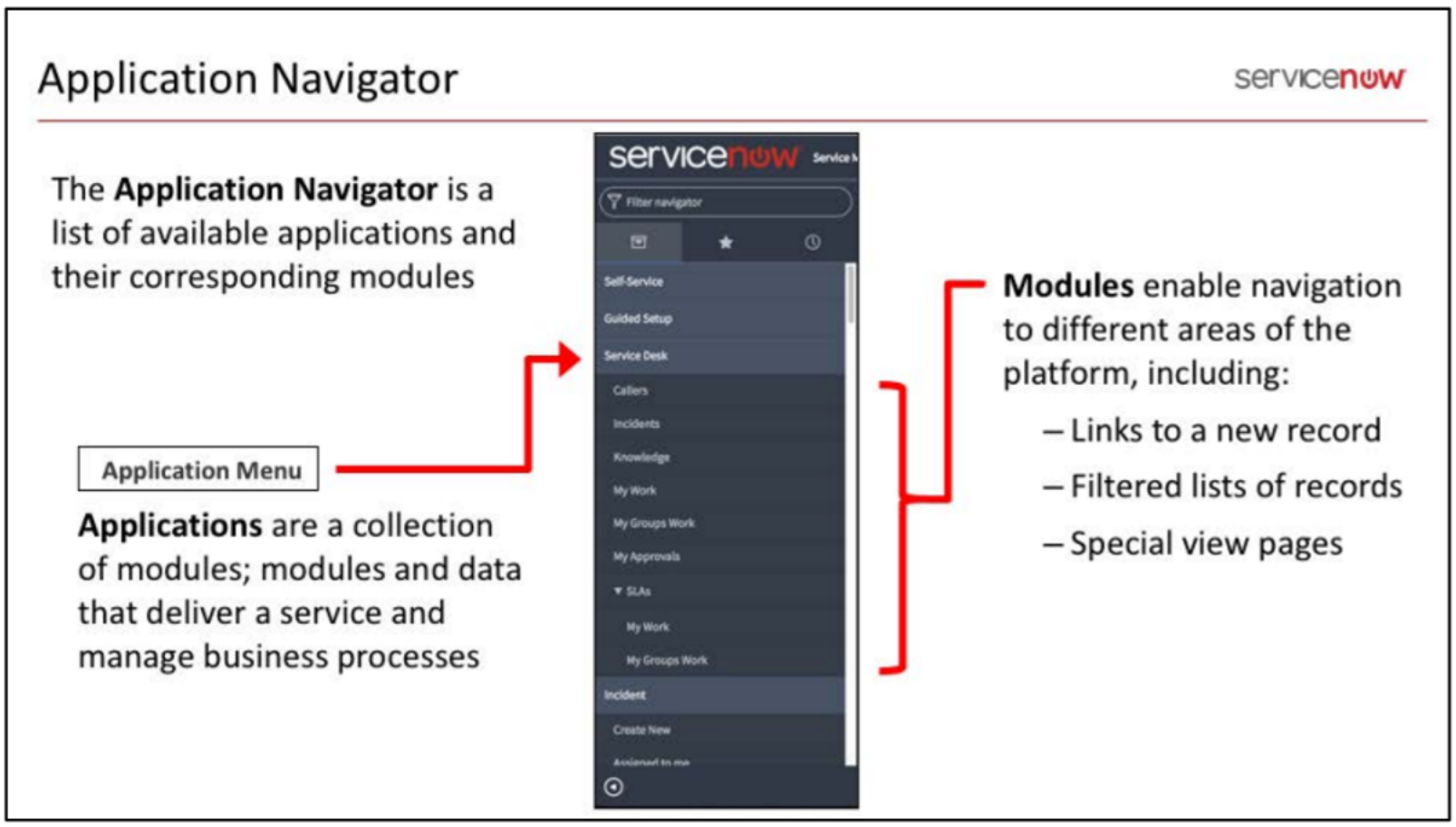


The User Interface (UI) is the main way for users to interact with the applications and information in a ServiceNow instance. Notable ServiceNow features include real-time form updates, user presence, an application navigator designed with tabs for favorites and history, and enhanced activity streams all of which you will explore in this training. This is an example of the System Administration homepage.

The ServiceNow user interface is divided into three areas:

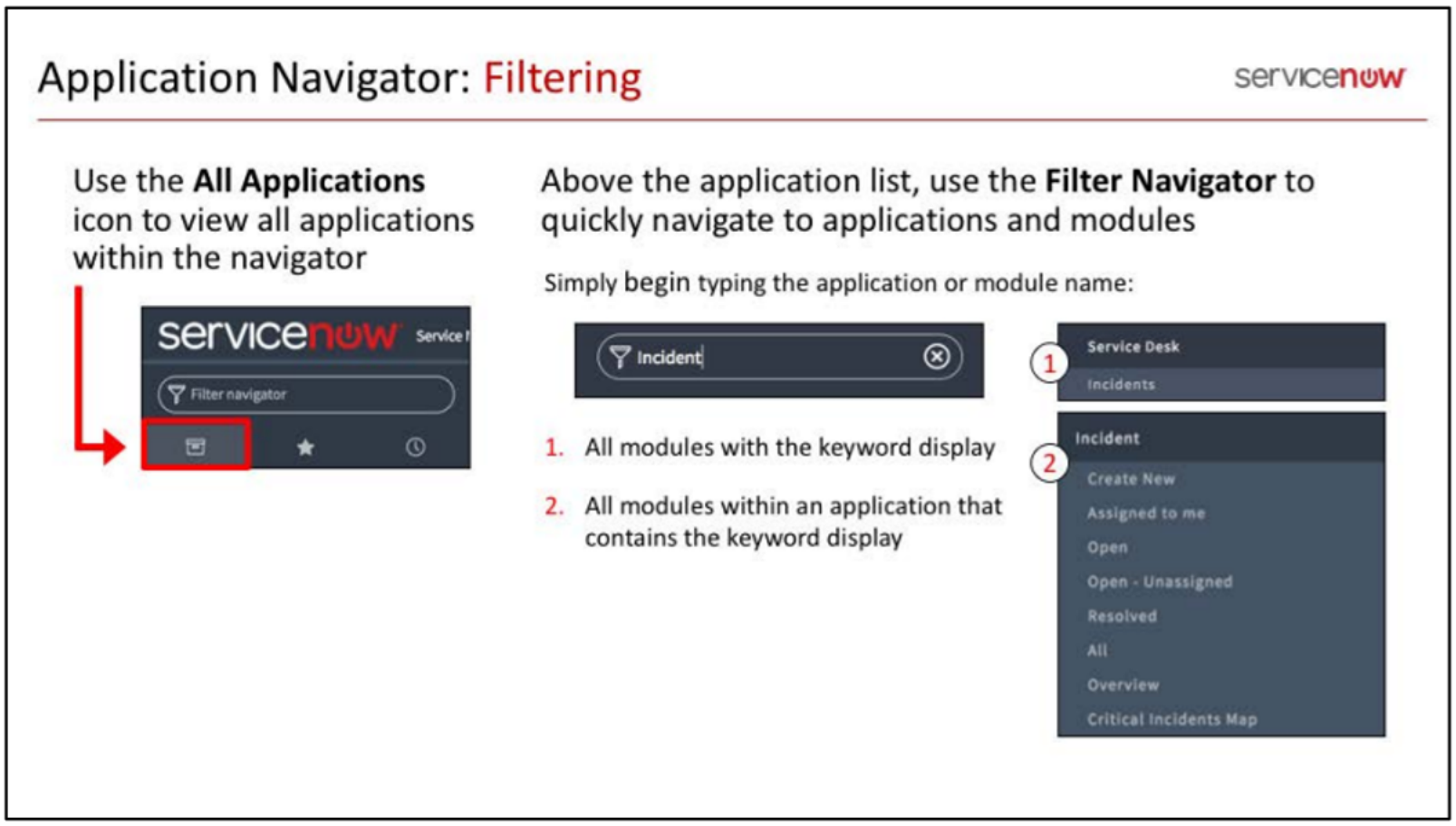
1. **Banner Frame**: The Banner Frame highlights important tools and settings that apply to your instance.
2. **Application Navigator**: The components of the Application Navigator, the panel on the left side, are based upon your assigned role(s). The navigator may be expanded (as shown above) or collapsed. The navigator provides links to all application menus and modules, based on your permissions.
3. **Content Frame**: The Content Frame displays information, such as lists, forms, dashboards, knowledge bases, and service catalogs depending on where you navigate within the platform. This also impacts how the information is visually represented.

NOTE: The position of these components on your screen may vary depending on your region.



Applications are a group of modules, or pages, that provide related information and functionality in an instance. Modules can contain links to a new record, lists of records with varying filters applied, and special visual tools.

For example, the Incident application contains modules for creating and viewing incidents. The Configuration application contains modules for changing and accessing servers, databases, and networks.



The Application Navigator provides access to all applications and the modules they contain, enabling users to quickly find information and services.

To view all applications within the navigator, ensure that the **All Applications** icon is selected at the top left of the navigator.

TIP: Double-click the **All Applications** icon to expand and/or collapse all applications. Click any application to expand or collapse all of its modules.

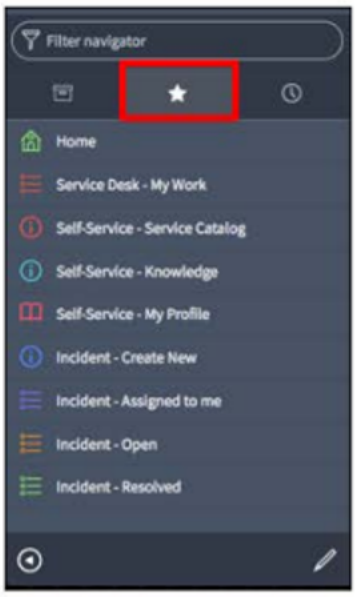
To quickly search throughout the application navigator to view a particular application or module, use the **Filter Navigator.** The Filter Navigator is located at the top of the Application Navigator.

As soon as a user begins typing, the Application Navigator displays only applications and/or modules matching the keyword. For example, if the keyword “Incident” is typed into the Filter Navigator, the Incident application and a list of *all* its modules will display, as well as any modules containing the word “Incident” within other applications, such as **Service Desk > Incidents.**

Application Navigator: Favorites and Your History

servicenow

Favorites



Items added as favorites appear in the **Favorites** tab of the Application Navigator, which is represented by a star icon

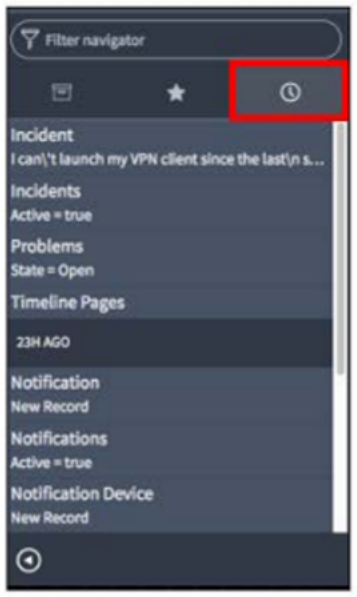
Favorites display as icons in a collapsed view

Your History

Your History provides a scrolling view of recent activities including:

- Forms and lists you accessed
- Homepages you visited

Simply click on any recent activity to open the item in the Content Frame

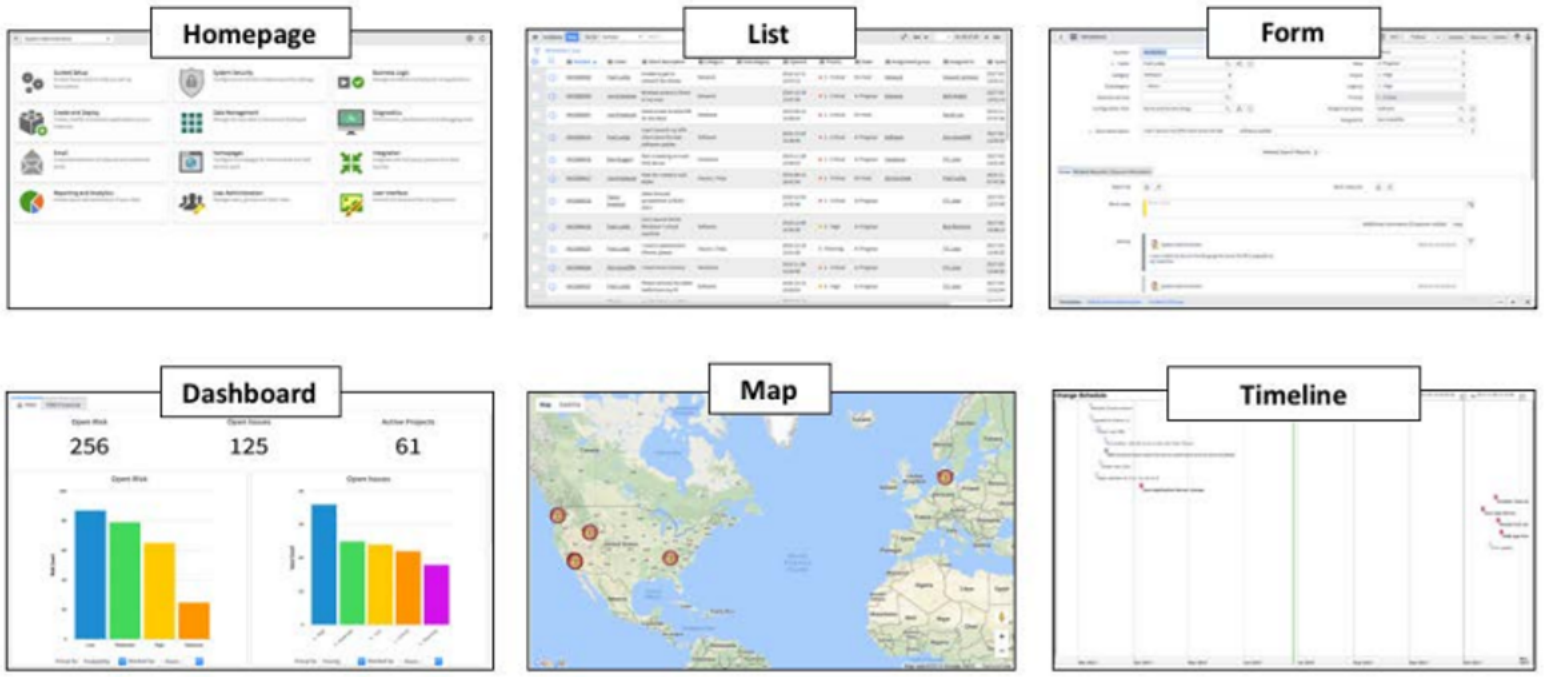


Favorites: Access the favorites menu to see all your favorites in one place. Favorites include application menus and modules which you may wish to access quickly and often. Favorites will also display in the Application Navigator even when a filter is applied, so long as the Favorite matches the search term.

Your History: The Application Navigator contains a scrolling list of your recent history within ServiceNow. For example, **Your History** will display forms you were filling out or lists you were searching on. Simply click on an item to open any recent activity in your content frame. Some content types are not tracked, including UI pages and other non-standard interfaces.

Content Frame: Common Types of Interfaces

servicenow



Homepage: A homepage consists of navigational elements, functional controls, and platform information. When a user logs in to an instance, the default homepage defined for their role appears unless the user switched to another homepage or has set a dashboard to appear.

All users with a role can use the **Add content** link on the homepage to customize the homepage and display important changes and emergency information to other users.

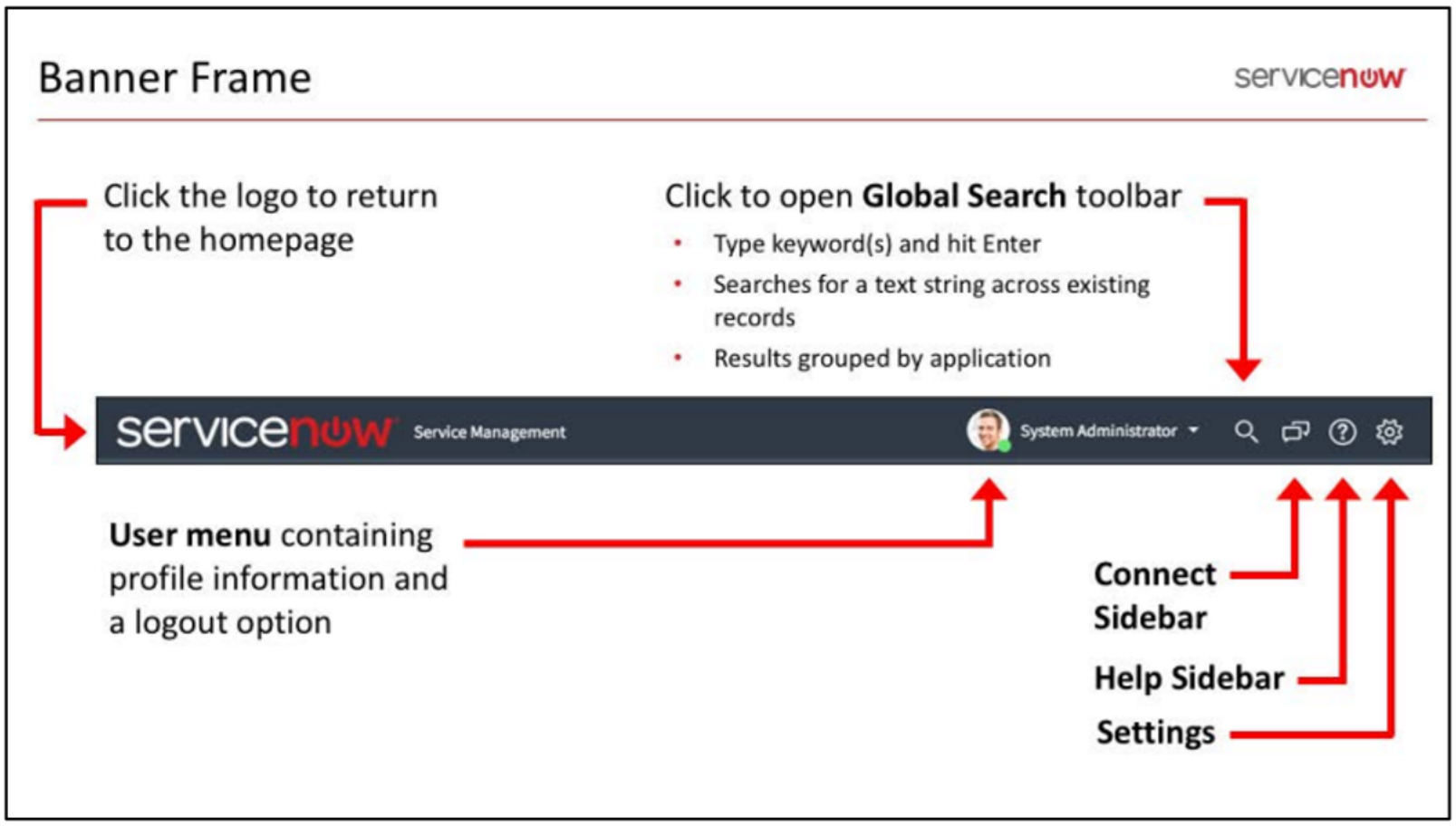
List: View data records as a list. Lists display records from a data table, as well as allow users to edit the record information using the List Editor functionality.

Form: View individual data records as a form. Data is typically entered into ServiceNow through forms.

Dashboard: Dashboards enable the display of multiple performance analytics, reporting, and other widgets on a single screen.

Map: Display ServiceNow data graphically on a Google map. Drill-down into a map to view specific data points.

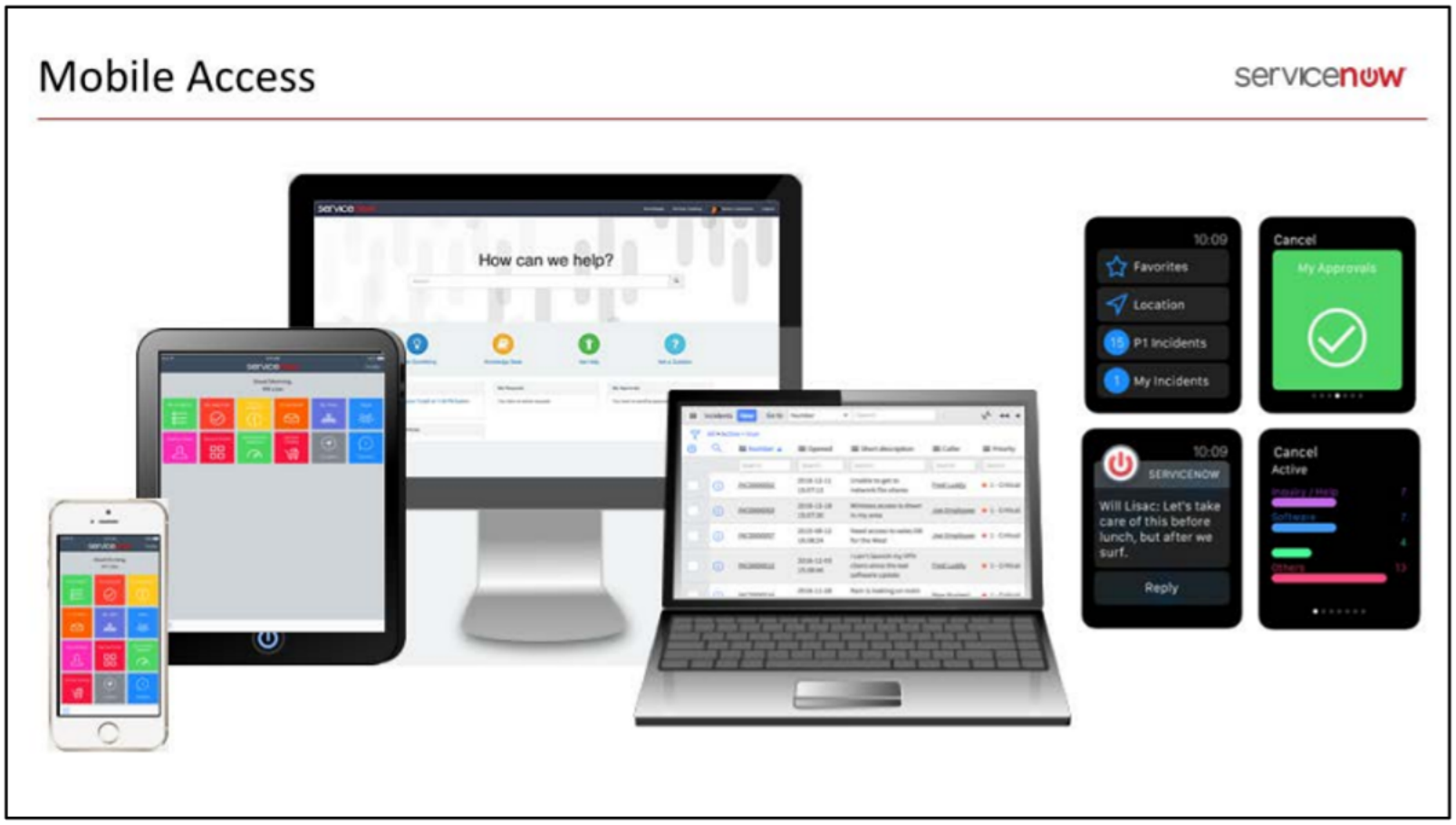
Timeline: Used to track tasks or projects.



The Banner Frame runs across the top of every page and contains global navigation controls and several key functionalities and features:

- Your logo in the top-left hand corner, which also navigates you back to your homepage when you click on it
- Information about the logged-in user: click the down arrow to the right of the user name to view the user profile or log out
- Click the magnifying glass to expand the Global Search toolbar and use this to search across all data in ServiceNow, such as a keyword, record number, and more
- Toggle on and off the Connect Sidebar, which is used to communicate with other users in real-time
- Get help, including Product Documentation and new features
- Personalize your settings

NOTE: With additional rights, a user may see Impersonate User and Elevate Roles as additional options from the user menu. These are features useful for testing and visibility.



In addition to accessing a ServiceNow instance from a laptop or desktop computer, ServiceNow supports the following technologies:

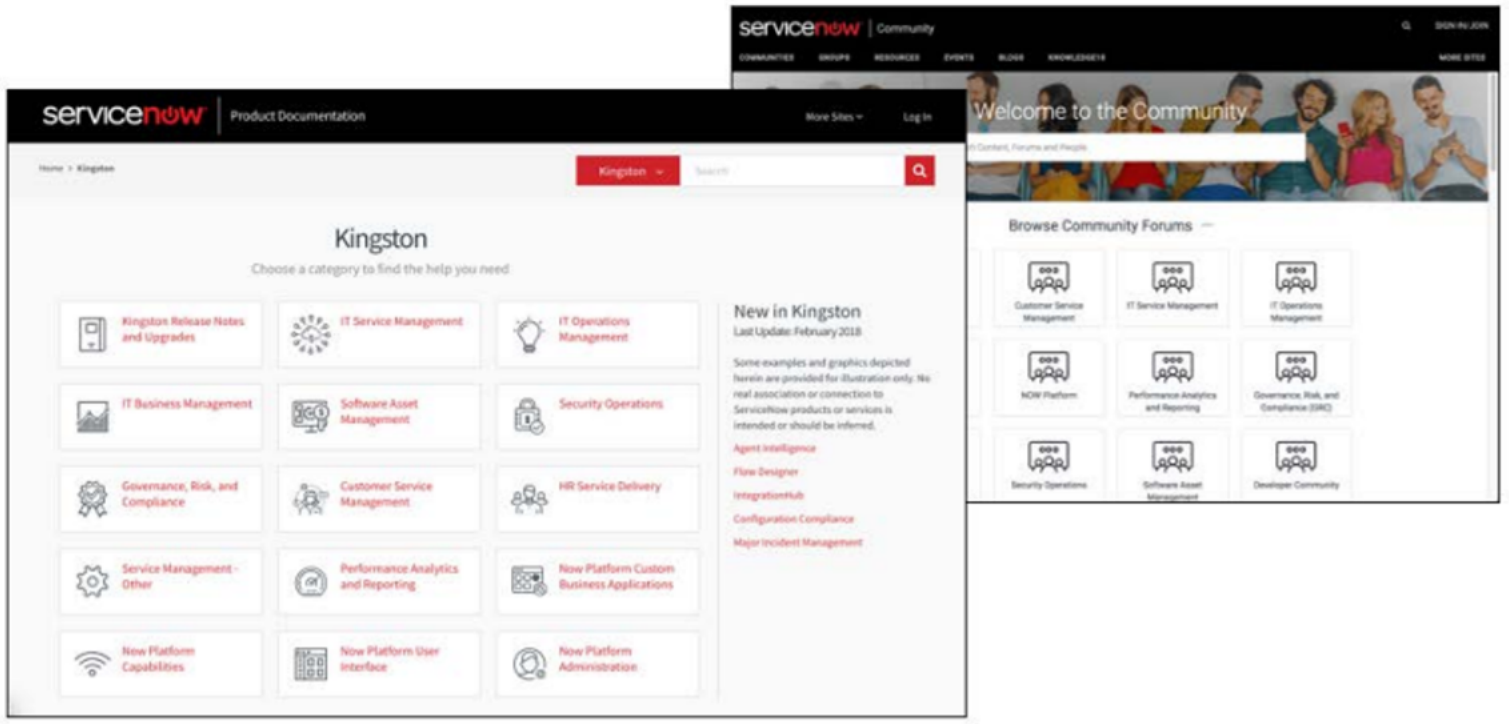
Smartphone: The smartphone interface supports many of the features found in the standard desktop/laptop browser interface, including lists, forms, favorite/shortcut management, and filtering. There are no special configurations needed for the iPhone or Android phones; the smartphone interface uses familiar, industry-standard techniques for performing most actions.

Tablet: The ServiceNow instance automatically detects the tablet and redirects to the desktop interface.

Apple Watch: Features include: notifications, favorites, record monitoring, chat messaging, dashboard charts, and record interaction via canned responses and voice to text (Siri).

Depending on how ServiceNow is accessed, the user interface and features may vary.

Product Documentation: Docs and Community

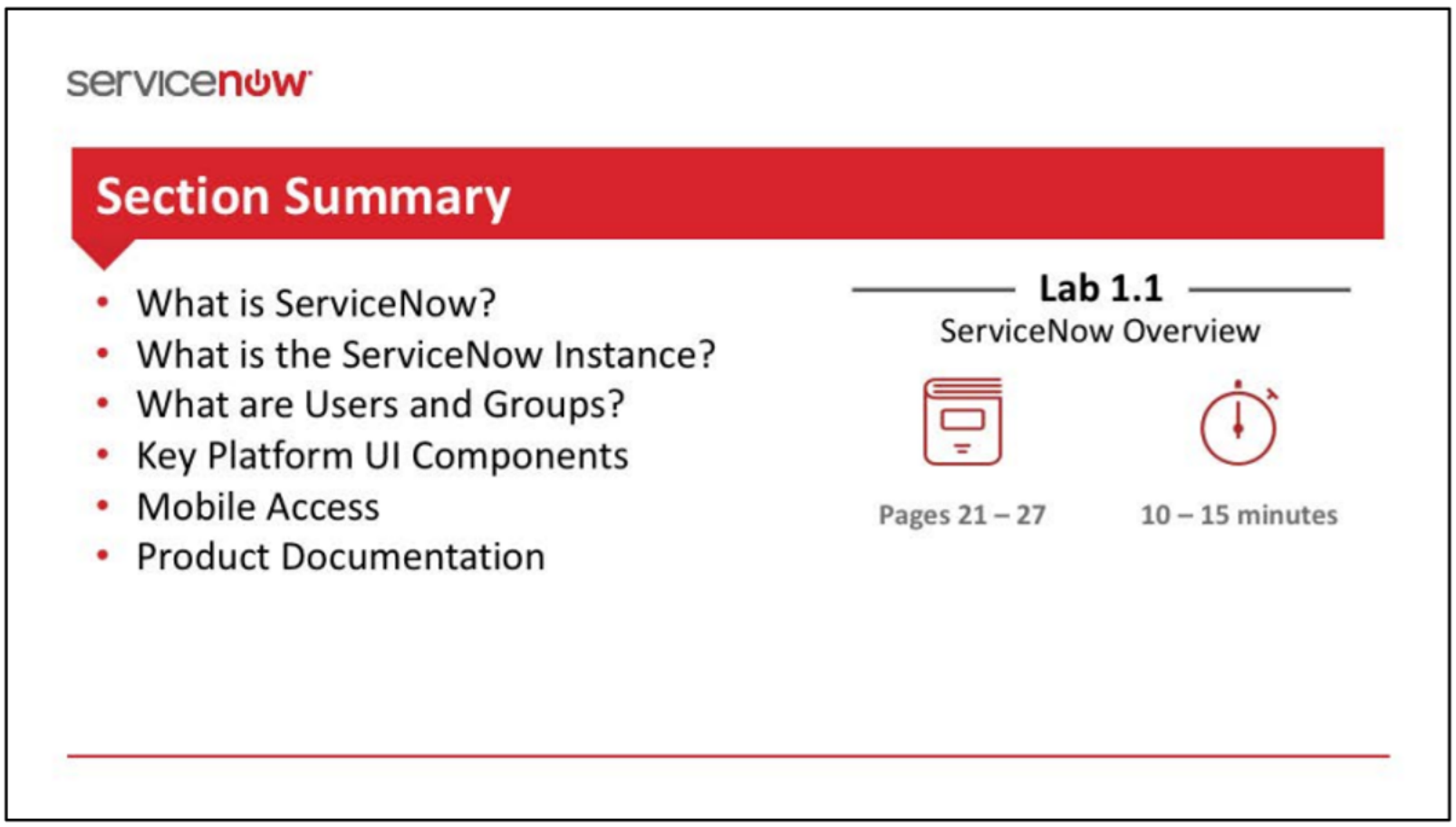


If anything in this class seems interesting, we highly encourage you to explore the topic in more detail through either of the following websites:

docs.servicenow.com is the official documentation resource for ServiceNow, with content produced by ServiceNow. From features to functionality, and even release notes, this resource should have all of the information needed to get the most out of the platform.

community.servicenow.com is similar to the Docs website, in that it provides useful information about the ServiceNow platform. However, where Community really excels is by bringing together actual ServiceNow users to collaborate, share, and produce ideas, content, and even answers to questions you may have!

This is a great resource to learn from users with real-life experience on the platform!



Lab 1.1 – ServiceNow Overview:

- Log on to your training instance
- Use the Application Navigator and its filter to access different areas of ServiceNow
- Add Knowledge and Service Catalog modules to Favorites

LAB

ServiceNow Overview

1.1

10 – 15 minutes

Lab Goal

This lab will show you how to do the following:

- Log on to your training instance
- Use the Application Navigator and its filter to access different areas of ServiceNow
- Add Knowledge and Service Catalog modules to Favorites

This course builds on a scenario where you work for a division of a fictitious electronics company called Cloud Dimensions.

Upon the reveal of their Infinity product; a portable holographic projector, you support a team of department Subject Matter Experts (SMEs) with the implementation of ServiceNow.

ServiceNow will initially be used by Cloud Dimensions for tracking Infinity inventory, order fulfillment, and customer support.

You will be required to impersonate various user personas – representing Cloud Dimension employees – throughout this course's labs.

To start, you will assume the system administrator identity to accomplish a series of tasks.

NOTE: *Screen shots are often cropped so what you see in the participant guide may not match exactly what you see in your instance.*

A. Log on to Your Training Instance

1. Navigate to **your assigned ServiceNow Lab Instance** in the web browser of your choice.

NOTE: *Your instructor provides you with your own instance URL.*

---

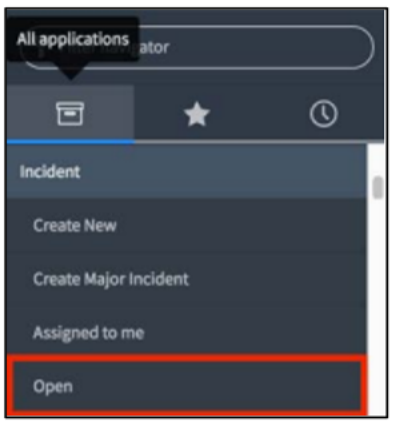
2. Log on using the **System Administrator (admin) credentials** provided by your instructor.

B. Use the Filter Navigator

1. Set the Application Navigator view to display all applications in an expanded view (double-click the **All applications** navigator icon to expand/collapse all), then locate the **Incident** application to view the 9 incident modules.

2. **Incident > Open**.

NOTE: *The **Application Menu > Module Name** formatting indicates the navigation path to use in the expanded Application Navigator. This shorthand will be used in the lab instructions going forward. For this step, select the Incident **Open** module:*

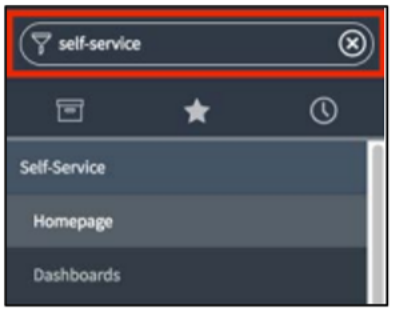


Notice how the user interface changed in the Content Frame from the System Administrator homepage to a list of open incident records.

3. **Incident > Create New**.

   Notice how the user interface has changed in the Content Frame from a list of incident records to an individual incident record/form.

4. From the Application Navigator, use the **Filter navigator** to filter the list of application menus and modules by typing **self-service** into the Filter navigator:



NOTE: *A single application menu, **Self-Service**, appears with many modules. Scroll down to see all of the modules under the Self-Service application.*

---

5. **Self-Service > Service Catalog**.

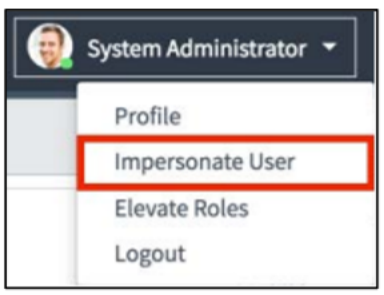
   Notice yet another user interface type displayed in the Content Frame.

6. From the Application Navigator, type the keyword **service** into the **Filter navigator**.

NOTE: *Scroll to see all of the applications and modules that contain the text "service" display.*

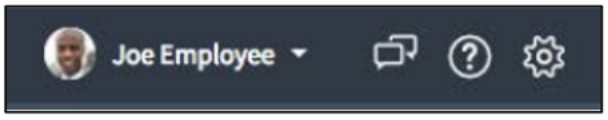
C. Set Module Favorites

1. Open the User menu on the Banner Frame, then select **Impersonate User**:



2. Impersonate the Cloud Dimensions employee **Joe Employee** by typing their name into the **Search for user** field.

NOTE: *After selecting their name from the drop-down list, ServiceNow should reload and you are now impersonating Joe Employee:*



---

3. Filter the Application Navigator using the keyword **self**.

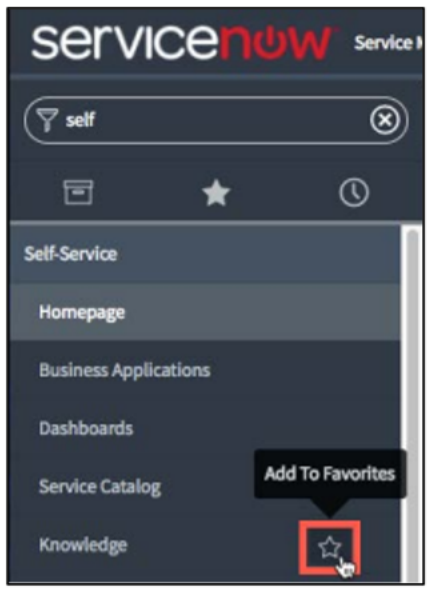
4. **Self-Service > Knowledge**.

5. Open the **ServiceNow Fundamentals Class** Knowledge Base**:**

6. Download the necessary lab files for class by selecting the **Class Lab Files** article.

NOTE: *Selecting the Class Lab Files article will download a zip file to your local machine titled* ***ServiceNow Fundamentals Class Files****. At your convenience, upzip the file.*

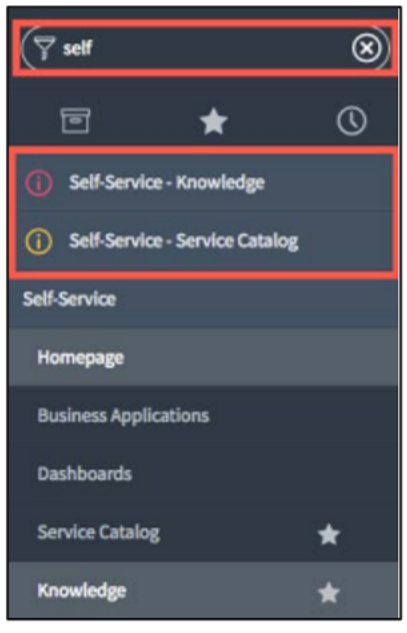
7. Next, hover over the **Knowledge** module, then add the Knowledge module as a favorite by selecting the **Add to Favorites** icon (star) to the right of Knowledge:



NOTE: *Upon selection, the module star will appear filled in.*

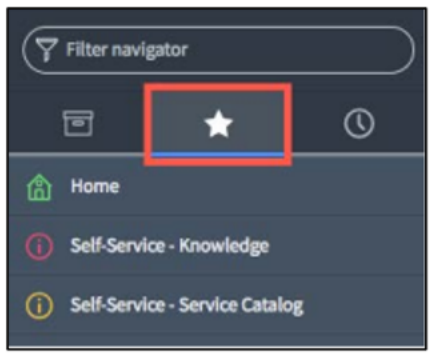
8. Repeat this step for the **Service Catalog** module.

NOTE: *In addition to application menus and modules, the* ***Filter navigator*** *will also display favorites based on keywords:*

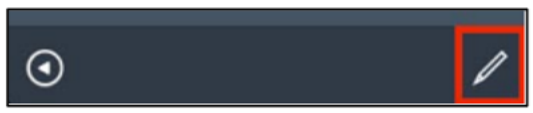


9. Clear the Filter navigator keyword by selecting the **X** to the right of the Filter navigator.

10. Next, navigate to the **Favorites** tab of the Application Navigator to see the module favorites you have created:



11. On the bottom-right of the Application Navigator, select the **Edit Favorites** icon (pencil):



NOTE: *The* ***Set up your favorites*** *screen displays in the Content Frame. A favorite can be customized to have any name, color, and icon.*

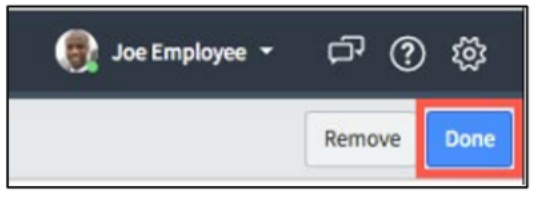
---

12. Select the **Self-Service - Knowledge** favorite in the Application Navigator.

13. Select any color and icon for the **Self-Service - Knowledge** favorite.

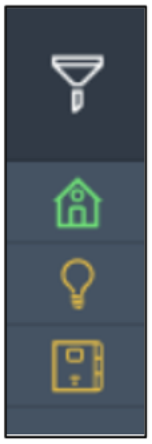
14. Repeat steps 12 and 13 for the **Self-Service > Service Catalog** favorite.

15. Click the **Done** button:



16. Minimize (collapse) the Application Navigator by selecting the **Minimize Navigator** icon (circled arrow) at the bottom of the Navigator:

17. From the minimized Application Navigator, notice that the two favorites appear in the color and icon you have selected:



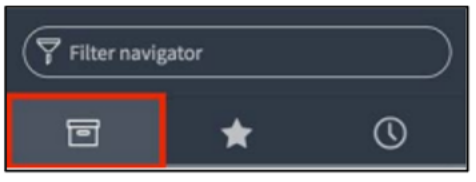
NOTE: *Your color and icon choices may vary from what is shown here for demonstration purposes.*

---

18. Navigate to the homepage by selecting the **Home** favorite displayed on the minimized Navigator:

19. Select **Maximize Navigator** at the bottom of the minimized Navigator:

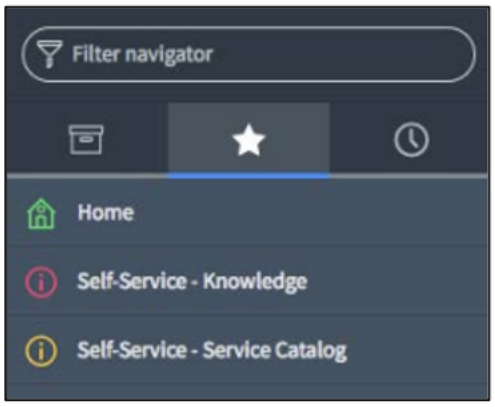
20. Next, select **All applications**:



LAB VERIFICATION

The Lab Verification displays screen shots of what you should have created during this lab. Sometimes the Lab Verifications have already been shown in earlier steps – as is the case below.

Self-Service Favorites



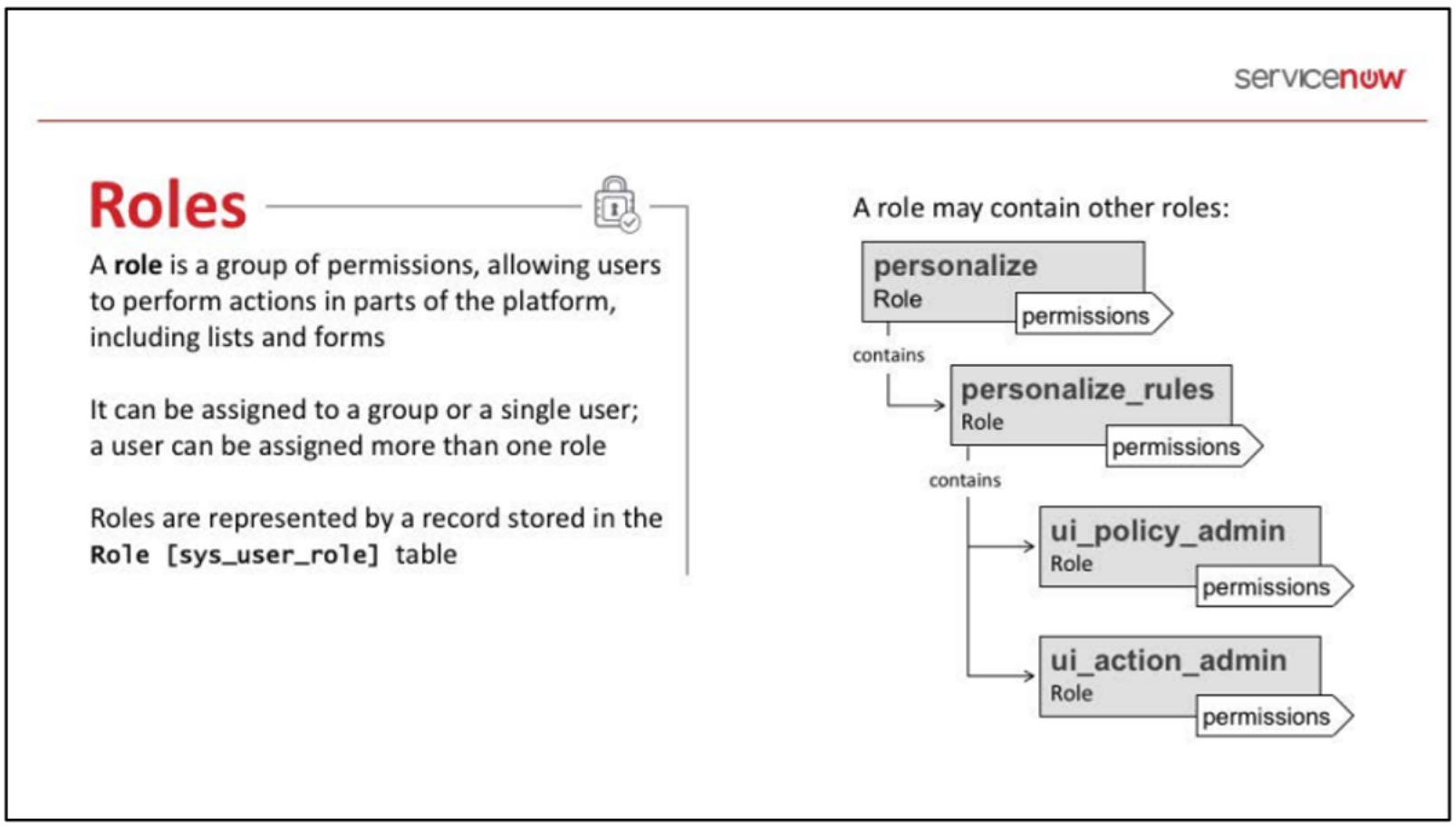
Congratulations on completing our first lab, ServiceNow Overview!

Module 1 – 1.2 Lists

servicenow

Objectives

- What is a Role?
- What is a List?
  - Anatomy
  - Views
  - Context Menus
- Layout Configuration
- List Personalization
- List Editing
- Tags
- Filters
- Finding Information: ServiceNow Search



A role is a collection of permissions used to:

- **Grant access** to applications and other parts of the platform
- **Assign security** rights

Once access has been granted to a role, all of the groups or users assigned to that role are granted the same access.

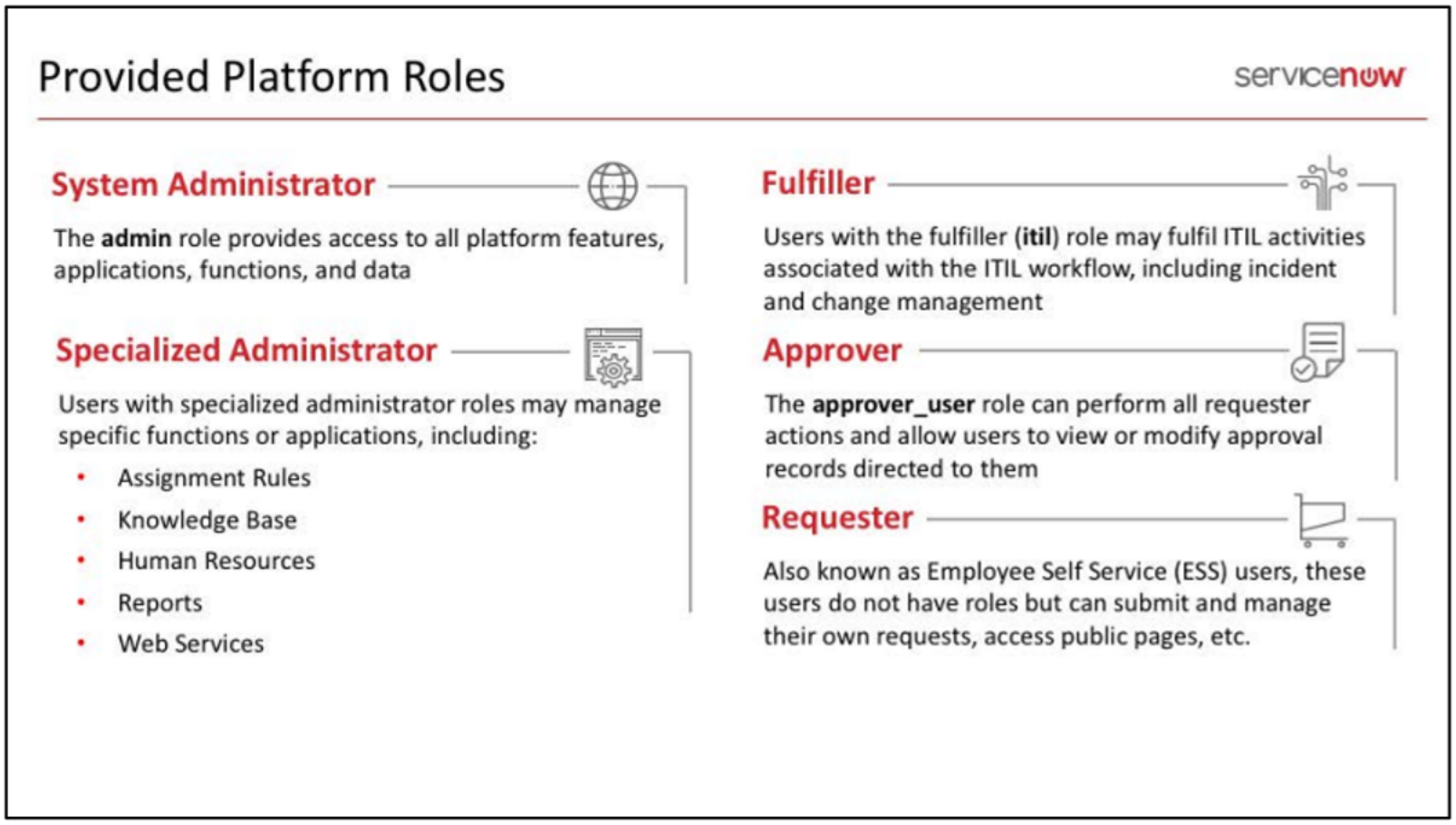
Additionally, a role may contain other roles and any access that is granted to one role is automatically granted to any role that contains it.

In this example, the **personalize** role is able to personalize forms, lists, rules, controls, and scripts. It has its own permissions and also contains the **personalize_rules** role.

The **personalize_rules** role has its permissions and contains both the **ui_policy_admin** and **ui_action_admin** roles.

The **ui_policy_admin** role can manage UI Policies. The **ui_action_admin** role can manage UI Actions.

Taken with all of these relationships, the **personalize** role contains all of the roles below it in the hierarchy. However the **ui_policy_admin** and **ui_action_admin** roles do not contain the permissions of the roles above them in the illustration.



The System Administrator (**admin**) role has almost all roles and access to all platform features, functions, and data, with some exceptions such as HR and Security Operations constraints. **Grant this privilege carefully**.

Users holding the **admin** role can create and modify user roles, as well as impersonate other users. However, not even users with the admin role can impersonate a **security_admin** role user and elevate privileges while impersonating to access higher security functionality.

The **impersonator** role can be assigned to a user to allow impersonation of other users, excluding admins, for testing and visibility purposes.

Specialized Administrator roles have broad access but generally manage specific functions or applications.

Fulfiller/Process users have clearly defined paths and workflows in the platform and have one or more roles, including the **itil** and **approver_user** roles. They can access all functionality based on assigned roles.

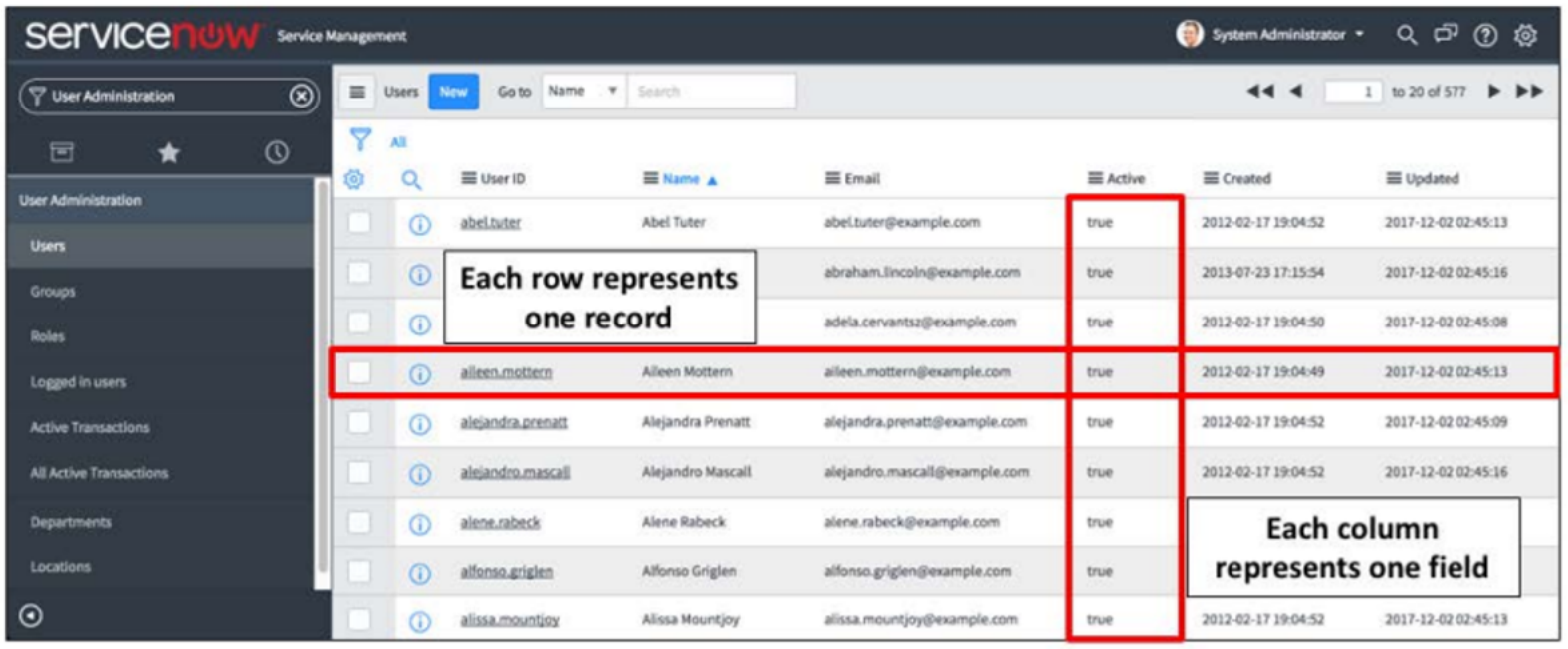
Approvers have the **approver_user** role, but no other roles.

Requesters use the Service Catalog and Self Service applications. They can make requests only on their own behalf, and are not assigned roles.

What is a List?

servicenow

A **list** displays a set of records from a table within the content frame of ServiceNow



Lists and forms are the most common ways to interact with data. A list displays a set of records from a table. Lists can be filtered and customized to display the information you need.

NOTE: Two different versions of list functionality may be encountered; referred to as List v2 and List v3. List v3 is enabled by a ServiceNow plugin and offers additional functionality such as displaying information in a split format.

In this example, the system administrator is accessing the **User Administration** application and a list of users through the **Users** module. Other roles, such as **user_admin**, grant users the permissions to manage users, groups, and roles.

List: Anatomy

servicenow

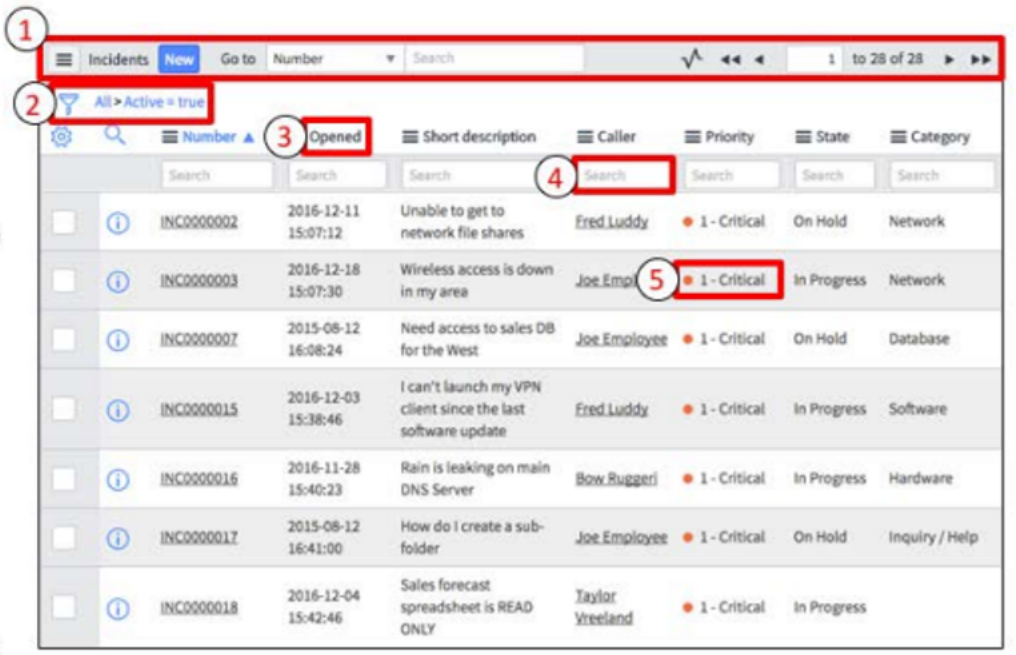
1. **Title Bar**
   Displays the list title and, in some cases, the view name, as well as search list values, and a record count

2. **List Filters/Breadcrumbs**
   Offers a quick form of filter navigation

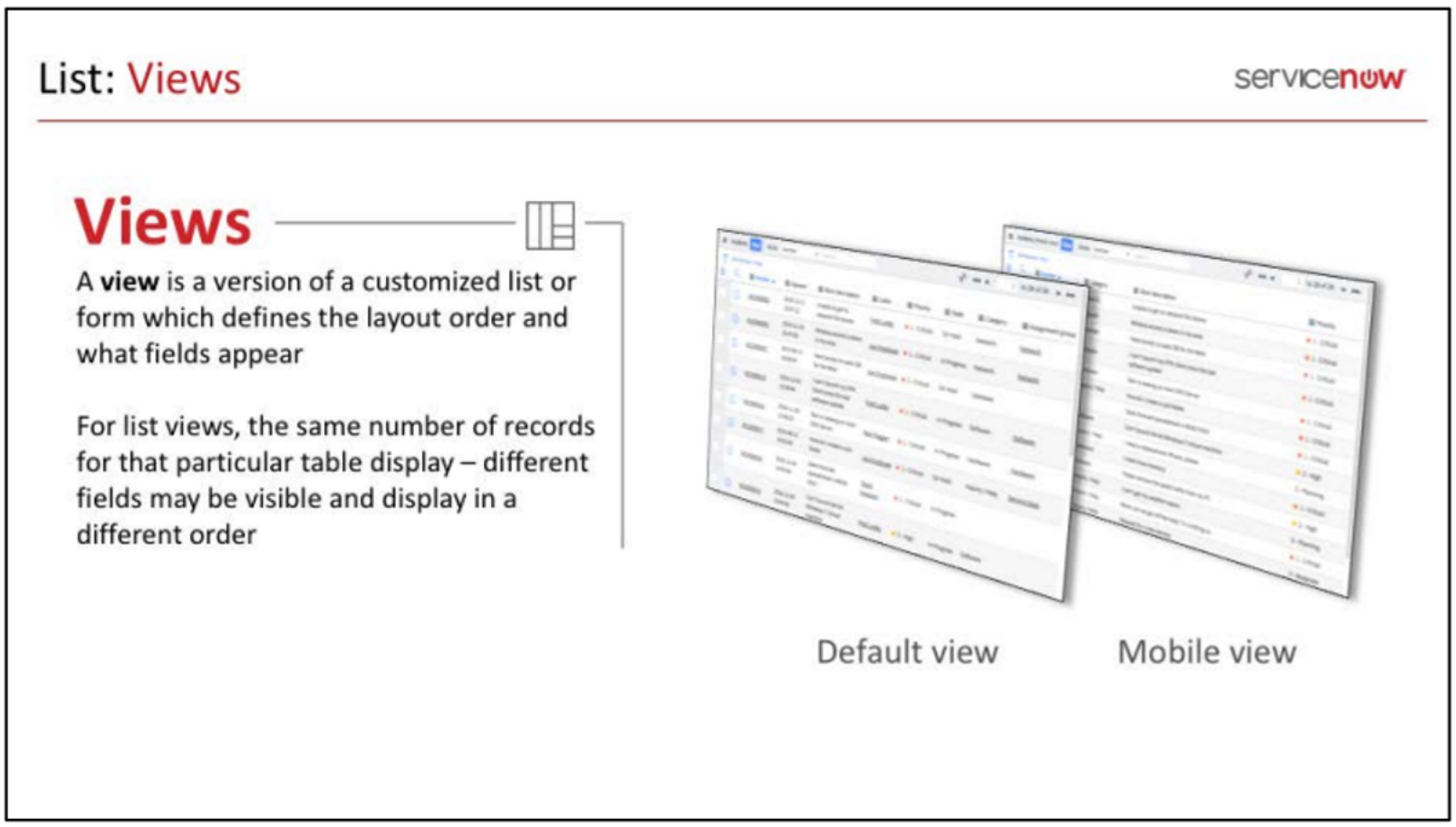
3. **Column Headings**
   Displays column (table field) names and provides some list controls

4. **Column Header Search**
   Provides a search within a specific column

5. **Field Values**
   Data; right-click a field value to access additional actions



Although lists display data captured in different tables, their interface remains consistent with common features.



Views: Views enable users to quickly display the same list or form in multiple ways. System administrators can create views for lists or forms. For example, different views can be created and used on Incident for an ESS user, an ITIL user, and a mobile user.

To switch between the different views of columns on a list (as shown here), open the List Context Menu then select **View**. Then, select the name of the desired view.

The view name appears in brackets beside the table list title and form record type when a view other than the Default view is selected.

NOTE: Switching views on a form will attempt to save all changes made to the record. A message displays asking to save or discard all changes made to the record, before the form reloads and displays the selected view.

Sort Controls: A list that is displayed to a user for the first time will be sorted by one of the following:

- The **order** field, if one is present in the table
- The **number** field, if one is present in the table
- The **name** field, if one is present in the table
- The field specified as the display field for the table

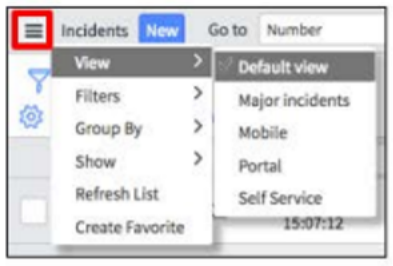
List: Context Menus

servicenow

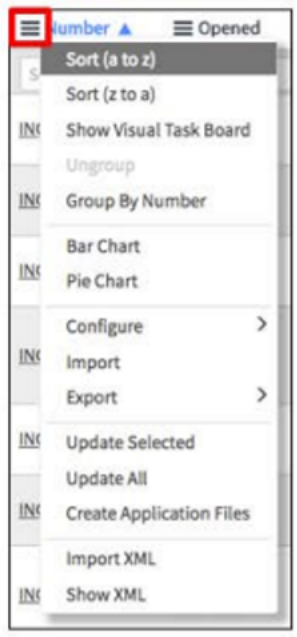
Context menus provide different levels of controls for a given list view

Context menus can be accessed by clicking the list menu icon (☰) or by right-clicking the list header and column headers respectively

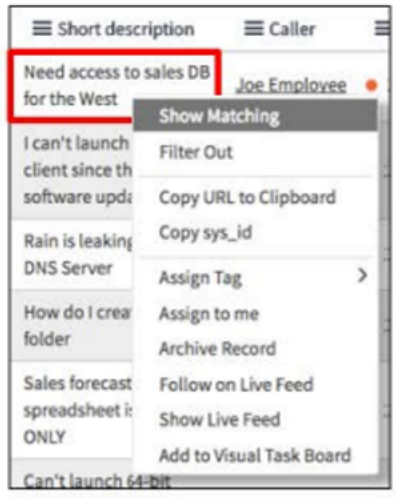
List Context Menu



Column Context Menu



Record Context Menu



Access the Record Context Menu by right-clicking in a row's cell

List Context (or control) menus, also sometimes called Additional Actions, can be accessed from lists, columns, or on records by using right-click menus which provide different levels of controls:

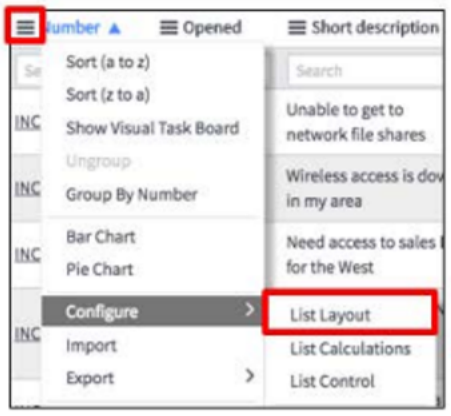
- **List Context Menu**: Click the list context menu icon next to the title of the list (Incidents in this example) to access options related to viewing and filtering the entire incidents list.
- **Column Context Menu**: Click the column context menu icon in the desired column header to display actions related to that column, such as creating quick reports, configuring the list, and exporting data.
- **Record Context Menu**: Right-click in a row's cell to see a menu with actions related to the values in that cell, such as filtering options, assigning tags, and more.

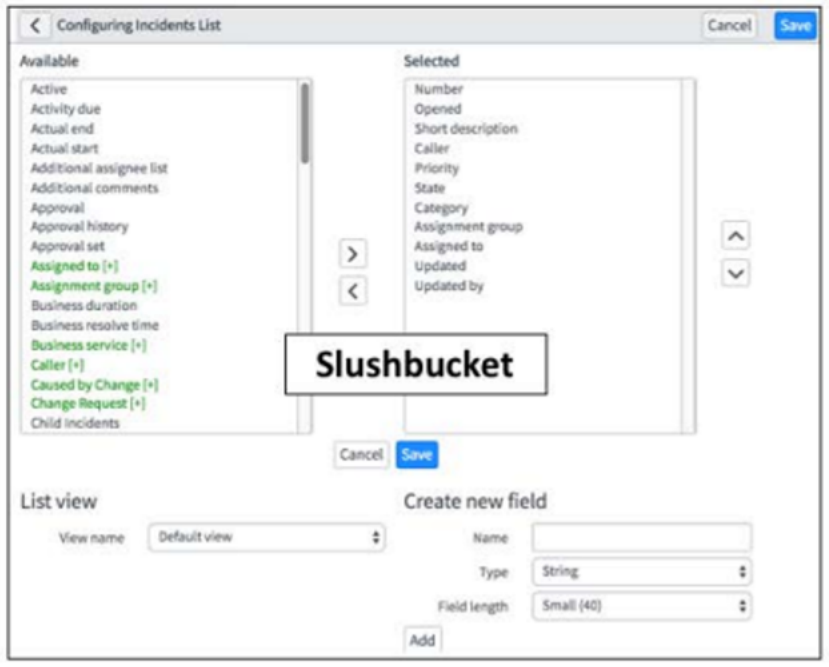
List: Layout Configuration

servicenow

Configure a list to show or hide fields from a view, as well as change the list column order

Click the column context menu icon and select **Configure > List Layout**





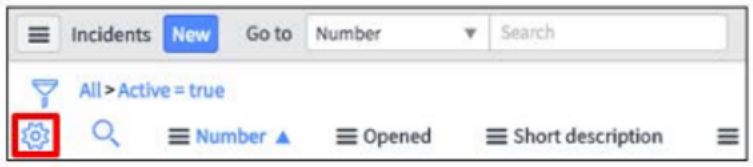
Users with the **admin** or **personalize_list** role can add or remove columns (fields) from a list or change the order in which the columns appear in the list, for all users.

To do so, navigate to the list, then open the **column context menu**, then select **Configure,** finally, select **List Layout**.

The slushbucket opens and has two sections: the available items on the left, and the selected items on the right. Items from the available section can be added to the list and items from the selected section can be removed from the list.

Once items are in the selected section, there are controls to adjust their order (up or down) on the list.

List: Personalization

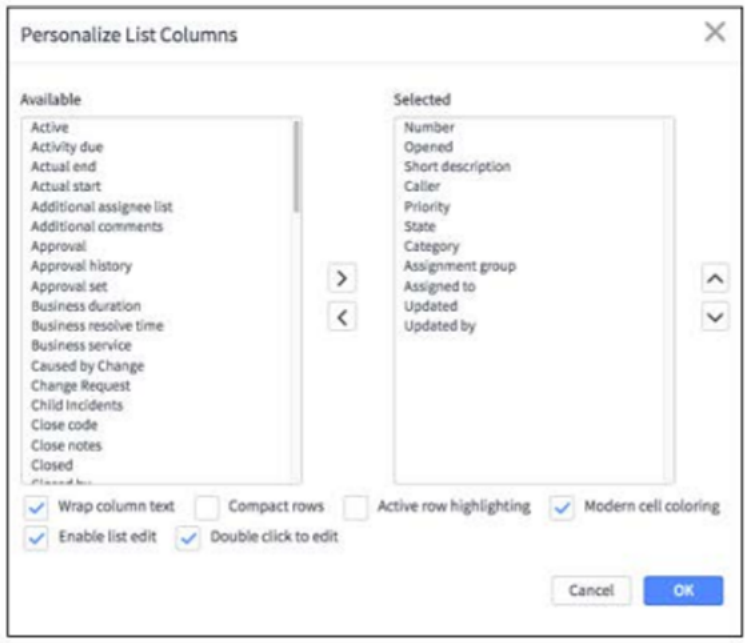


Personalize a list to show or hide fields on a view, as well as change the list column order

Unlike configuring a list, personalizing will not affect what other users see while viewing the list

To personalize a list view, click the **Personalize List** icon or gear ( ) from the list column header

After personalizing a list, this icon will change to look like this ( )



Personalize List Columns modifies a list for an individual user; it does **not** affect the platform default. List layout changes made using **List Layout** will affect everyone, across the the entire instance, except for individuals using personalized layouts set via Personalize List.

The following can be done through Personalize List Columns:

- **Add Columns:** In the available section, select each column you want to add and press the add icon
- **Remove Columns**: In the selected section, select each column you want to remove and press the remove icon
- **Rearrange Columns**: In the selected section, select the column(s) you want to reorder and use the up or down icons to place the columns in the desired order
- **Reset Column Defaults**: Return the list's columns to the default list's view definition

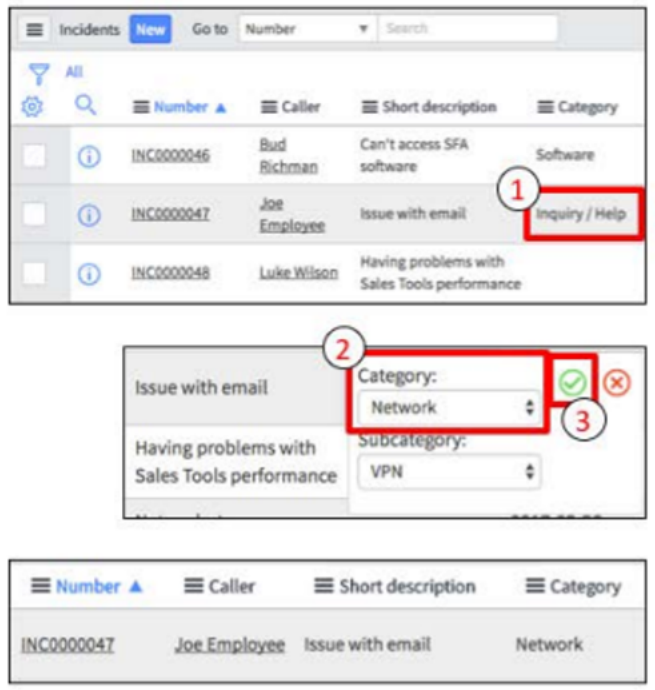
List: List Editing

servicenow

The **List Editor** allows a field value to be edited in a list without opening the form

Locate a record with the field value to change:

1. Double-click in an empty area of the field
2. Enter the appropriate value(s)
3. Save the record by clicking the save (✓) icon
   - Clicking the cancel (✗) icon or pressing the Escape key retains the original value



Users can edit data in lists using various methods but certain field types cannot be edited. Additionally, list editing is disabled for some tables.

The list editor is the quickest method to update a field on multiple records.

Procedure

1. Select the records to be edited.
2. Open the list editor by double-clicking (or clicking, depending on setup) in an empty area of the field. The number of selected rows that will be edited is indicated. If any rows cannot be edited due to security constraints, that is indicated. Administrators can configure the list editor and by default, list editing is disabled for some tables.
3. Enter the appropriate values and click the save icon.

Quick edit functions may also be used to edit records. Right-click a field and select the appropriate function:

- **Assign to me:** For records that use assignments, places the logged-in user's name into the **Assigned to** field
- **Approve:** For records that use approvals, changes the approval state of the record to **Approved**
- **Reject:** For records that use approvals, changes the approval state of the record to **Rejected**
- **Assign tag:** For records that are to be tracked based on a user-defined label

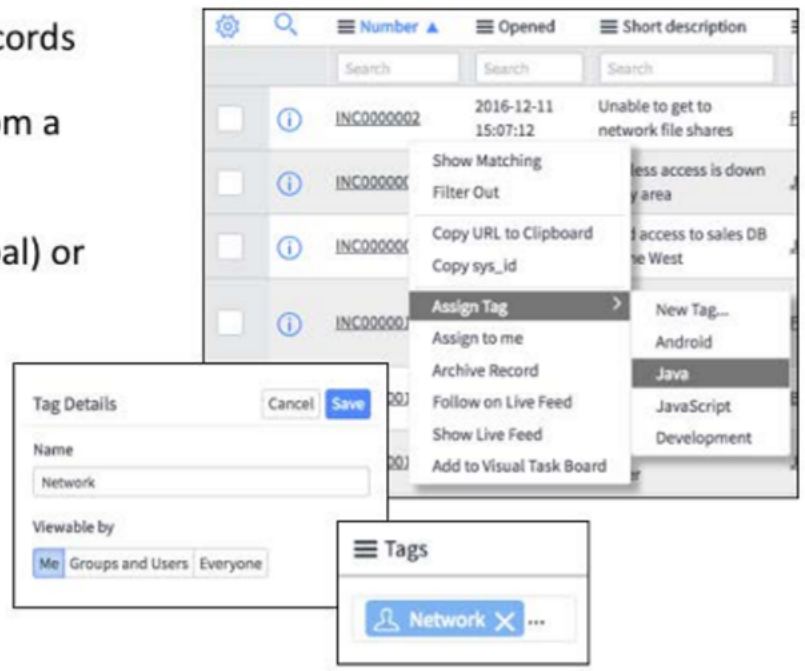
Tags

servicenow

Use **tags** to categorize, flag, and locate records

Tags can be created against any record from a list or form view

Tags can be made visible to any user (global) or visible only to specific users



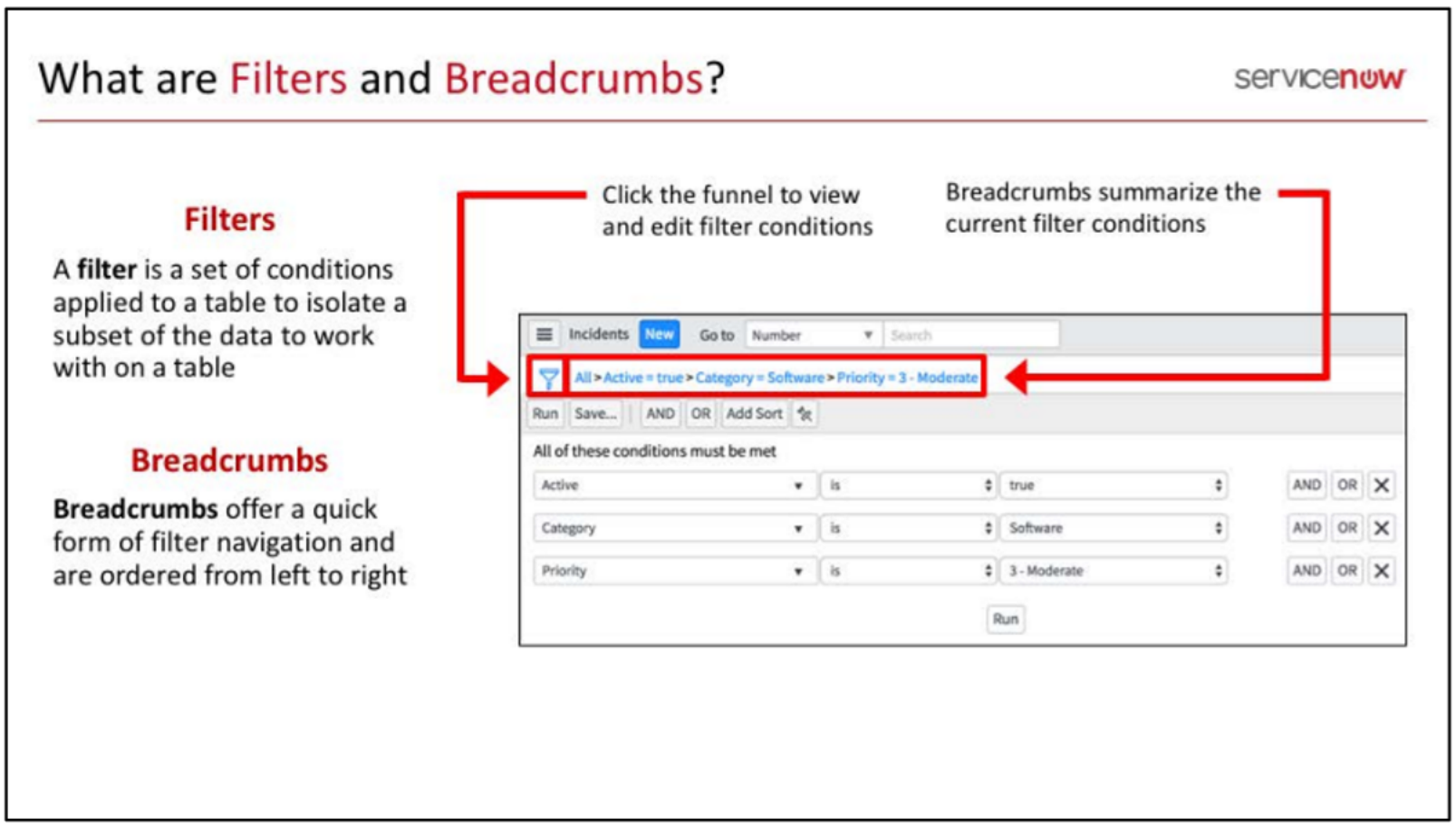
There are a few ways to assign tags to records:

- From the list view using inline field editing
- From a list using the record context menu
- Configuring tags to assign automatically

Use the **Viewable by** field when editing a tag to control how it is shared: visible only to the owner (Me), visible to the owner and specific groups or users (Groups and Users), or visible to everyone (Everyone).

To use the Everyone option under Viewable by, a user must have the **admin** or **tags_admin** role.

NOTE: Editing personal tags can be accomplished using the **My Tags** or **My Tagged Documents** modules.



Filters are used to specify exactly which records to displaywithin a selected list. For example, you may start with a list of all incidents but filter those records to view only active incidents assigned to you. Users can apply, create, modify, and save filters.

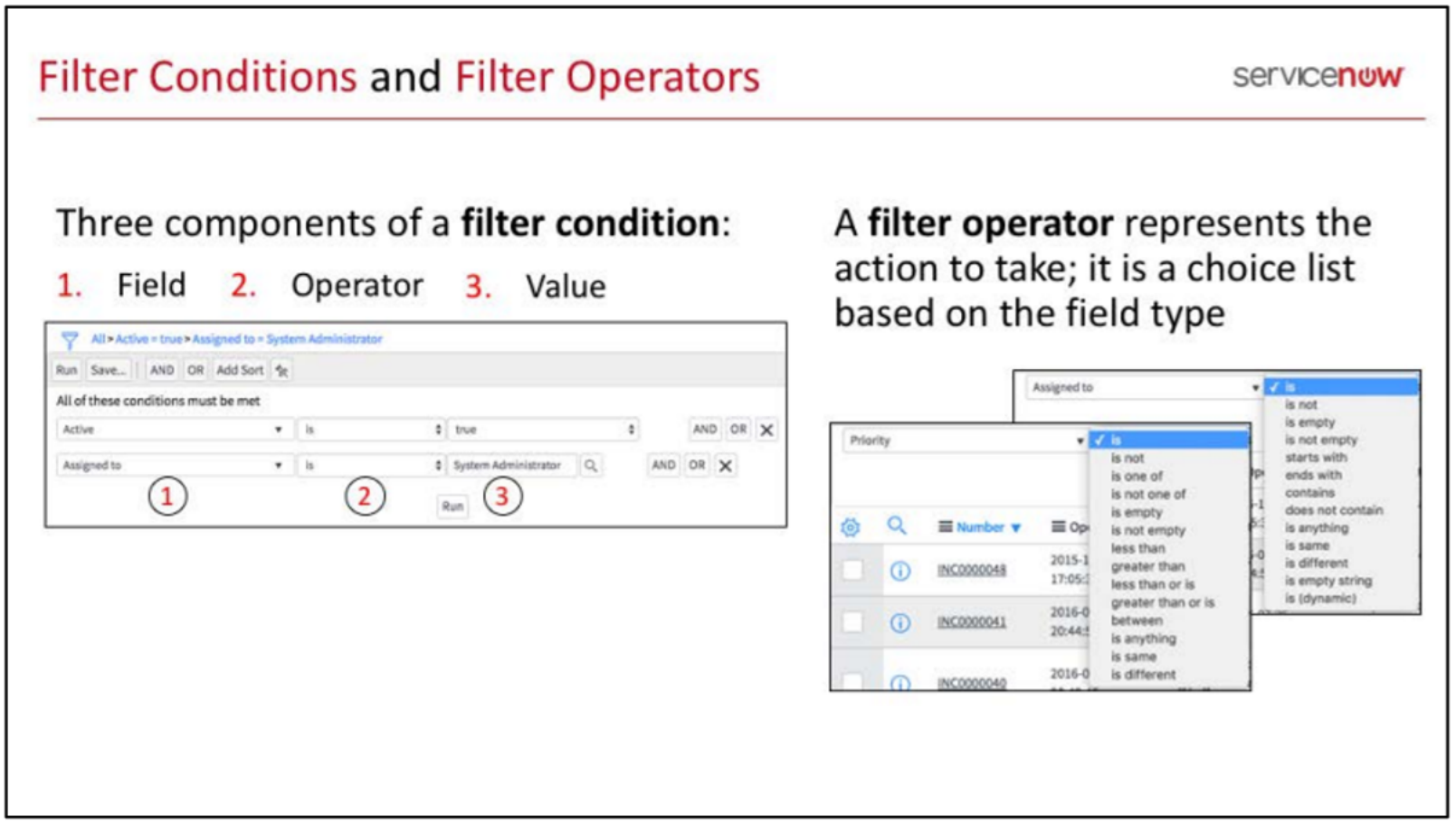
To view the filter applied to a list, click the funnel icon on the top left of the list. Here, you can add, remove, or edit **filter conditions** and rerun or save your filter.

Click **Run** to see the results of your filter, displayed in the list. To save a filter, click **Save**. A new field will appear where you can name your filter. After naming the filter, click the **Save** button to the right of the name field. The new filter will be available by selecting **Filters** from the list context menu.

The filter conditions applied to the list are summarized in the **breadcrumbs,** shown in blue letters across the top of the list. Not only do the breadcrumbs provide an “at-a-glance” view of the filter’s conditions, but you can also modify conditions and add to favorites your filter directly from the breadcrumbs.

Click a breadcrumb to remove all conditions to its right. Clicking the condition separator (>) before a condition to remove only that condition.

Add filter conditions to your favorites menu simply by dragging and dropping the breadcrumbs onto the navigator



The three parts of a filter condition are:

1. **Field**: A choice list based on the table and user access rights. The choice list includes fields on related tables by dot-walking.
2. **Operator**: A choice list based on the field type. For example, in the incident table, the greater than operator does not apply to the Active field but it does apply to the Priority field.
3. **Value**: A text entry field or a choice list, depending on the field type. For example, in the incident table, the Active field offers a choice list with the values true, false, and empty, while the Short description field offers a text entry field.

A filter operator can specify conditions including: it **is** this, it **is not** this, it is **same as,** it **is different** from, etc.

Filter operators will change depending on field data type, for example:

- **Text value**: is, is not, contains, is one of, starts with, ends with
- **Numeric**: is, is not, greater than, less than, greater than or is, less than or is
- **Date**: on, before, after, between, is more than, is less than

Finding Information: ServiceNow Search

servicenow

Use any of the following search methods to find information in ServiceNow:

- **Lists**: Find records in a list; search in a specific field (Go to), all fields (Search), or in a specific column
- **Global text search**: Find records in multiple tables from a single search field
- **Knowledge Base**: Find knowledge articles
- **Service Catalog**: Find catalog items
- **Filter Navigator**: Filter the items in the application navigator

List and Global Wildcard Searches

Wildcard Search Syntax	Does a Search
**mySearchTerm*	Contains
!**mySearchTerm*	Does not contain
mySearchTerm%	Starts with
%mySearchTerm	Ends with
=mySearchTerm	Equals
!=mySearchTerm	Does not equal
mySearchTerm No leading or trailing Wildcard(s)	Greater than or equal to *mySearchTerm* when using the list "Go to" search **only**

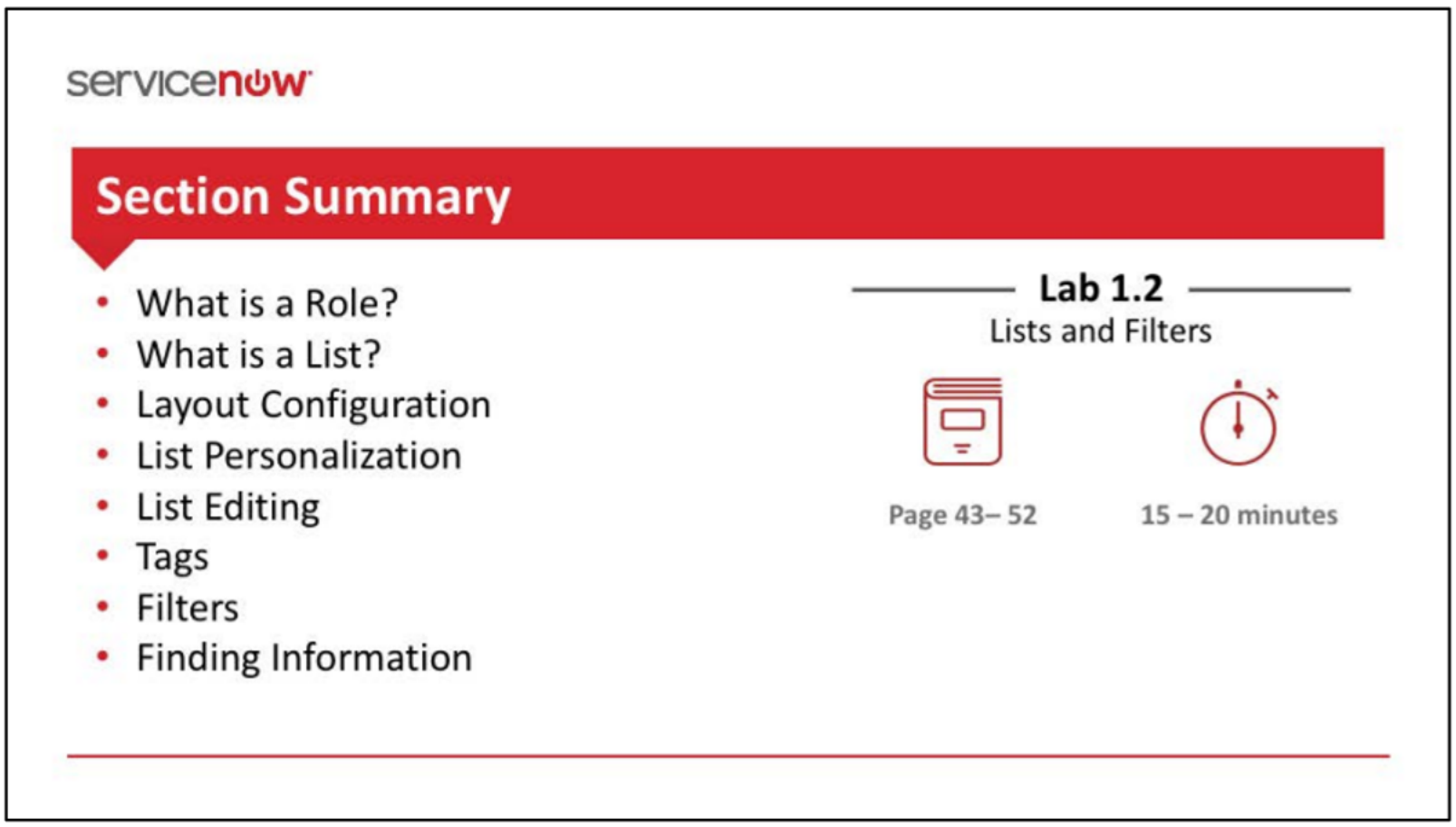
Find information quickly in ServiceNow by using any of the available searches:

- **Wildcards**: Use a symbol to represent zero or more characters
- **Phrase Searches**: Find a phrase with multiple terms
- **Searching Lists**: Control the query for list searches of a specific field
- **Boolean Operators**: Refine searches with operators such as AND and OR
- **Attachment Searches**: Search in files that are attached to Knowledge Article records
- **International Character Sets**: Perform searches with any Unicode characters
- **Punctuation**: Perform searches that contain punctuation

Wildcards use a symbol to represent zero or more characters and are available for searches. Various wildcards can be used to refine the search in lists (text searches of all fields), the global text search, and the Knowledge Base. Results with using wildcards may vary depending on the search method used.

Searches are not case sensitive. Use advanced options for more specific queries.

NOTE: Zing is the text indexing and search engine that performs all text searches in ServiceNow.



Lab 1.2 – Lists and Filters:

- Open an Update Set
- Create a new Infinity list view on the Incident table
- Practice filtering data on an incident list and saving a new filter
- Locate and update incident records using inline editing

LAB

Lists and Filters

1.2

15 – 20 minutes

Lab Goal

This lab will show you how to do the following:

- Open an Update Set
- Create a new Infinity list view on the Incident table
- Practice filtering data on an incident list and saving a new filter
- Locate and update incident records using inline editing

One goal of Cloud Dimensions with using ServiceNow is handling Infinity support.

Before the product is launched, however, Cloud Dimensions employees are actively testing Infinity devices.

Winnie Reich – manager of the Service Desk – has requested help from the Cloud Dimensions system administrator in creating a new Infinity view on the incident table.

This view will be configured to include the necessary fields for supporting Infinity, for both internal and external users alike.

Winnie has also asked her direct report, Kevin Edd, to create and share a list filter that will filter active incidents and display only those submitted by Infinity employee testers.

A. Open an Update Set

Before starting the lab, create an Update Set. Update Sets are used to capture configuration changes made to the platform and are explored later in class. Create an Update Set now and follow up with it at a relevant time.

1. Impersonate the **System Administrator** using the user menu.

   NOTE: *If you have logged out of your instance, use the* ***System Administrator*** *(admin)* ***credentials*** *provided by your instructor*

2. **System Update Sets > Local Update Sets**.

3. From the list of local Update Sets, click the **New** button.

4. Fill out the form as follows:

   Name: **ServiceNow Fundamentals**
   State: **In progress** (auto-fills)
   Description: **Includes work completed in the ServiceNow Fundamentals course.**

5. Click **Submit and Make Current**.

NOTE: *A message appears at the top of your instance indicating the Update Set has been set to the current Update Set. This means changes made to the platform will be captured in this Update Set moving forward.*

B. Create the Infinity List View

The system administrator user has the appropriate permissions for creating a new list view on incident – we will assume they have already received the requirements from Winnie Reich.

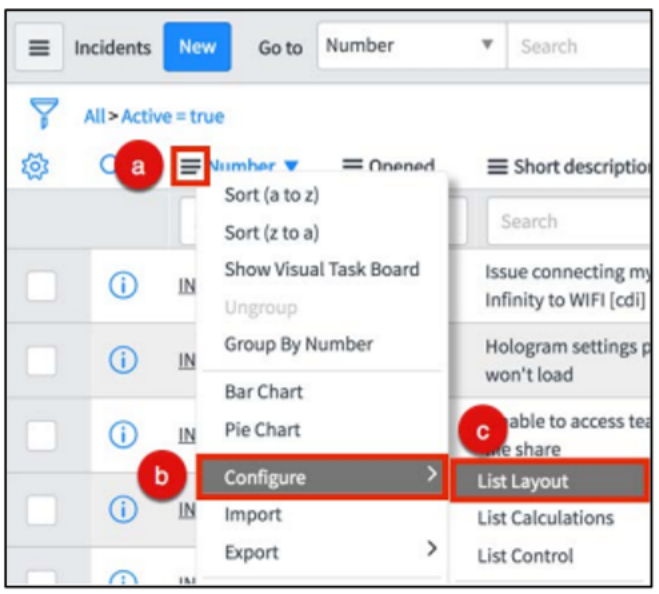
1. **Incident > Open**.

2. From the list column header, open the "**slushbucket**" to create a new list view:

   a) Open the **Column Context Menu**

   b) Select **Configure**

   c) Select **List Layout**

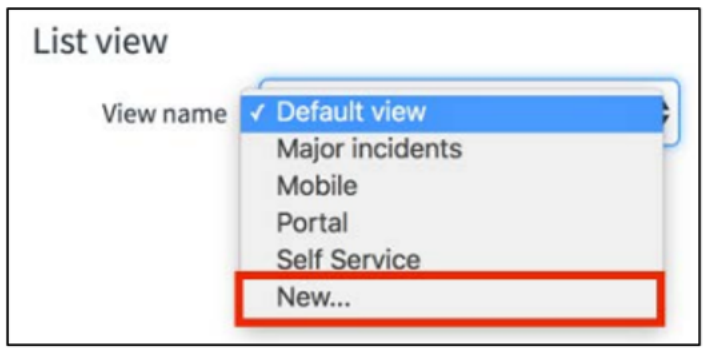


NOTE*: Selecting any field will work, but* ***Number*** *was used in this example.*

---

3. Beneath the Available and Selected buckets, open the **View name** drop-down menu from the List view section.

4. Select **New…** at the bottom of the list:



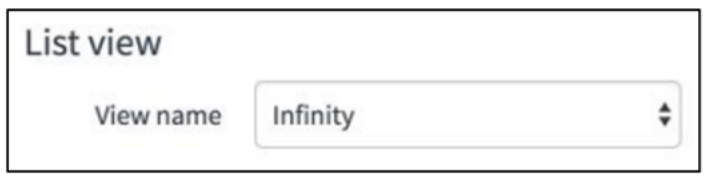
***NOTE**: Choosing an existing view from this list will allow you to modify it.*

---

5. Enter the View name: **Infinity**.

6. Click **OK**.

   Nothing appears to have happened to the page but you should now notice **Infinity** as the selected List view:



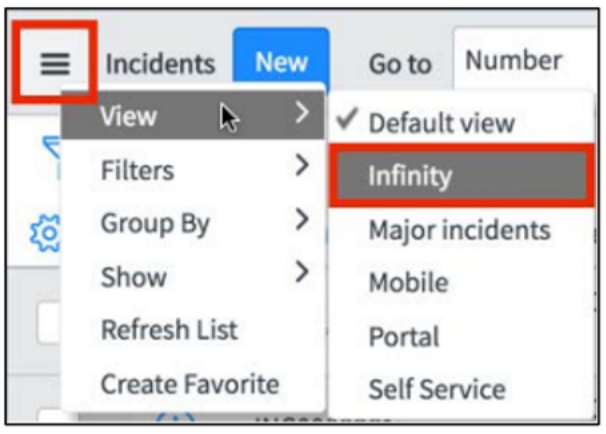
7. Working with the **Available** and **Selected** buckets, use the **Add** and **Remove** buttons (“>” and “<” icons, respectfully) to create the Infinity list view with the following fields:

   Number
   Priority
   State
   Caller
   Category
   Subcategory
   Short description
   Assignment group
   Assigned to
   Tags
   Updated

***NOTE**: Ensure the fields are listed in this same order under **Selected**. Use the **Move up** and **Move down** arrows, on the right of Selected, to set the correct order.*

***TIP**: You may select multiple fields under Available or Selected, then add or remove them with one click.*

8. Once the fields have been added to the Selected bucket, click **Save**.
9. Impersonate **Kevin Edd** to confirm the view is available for the Service Desk group.
10. As Kevin Edd, navigate to **Incident > Open**.
11. Open the **List Context Menu** and select **View**, then finally select **Infinity**:



***NOTE**: The list view name now appears at the top of the list in square brackets:*

12. Confirm the fields appear in order, from left to right, as listed in step 7 above.

C. Apply and Save a Filter

Filters allow users to locate specific data quickly, and filters are also reusable. Kevin Edd will apply a filter that displays Cloud Dimensions Infinity incidents related to employee testing, then save the filter to share with his team for future use.

1. Open the filter condition builder by selecting the **Show / hide filter** icon (funnel):

2. Add the following **AND** condition:

   Tags | EIT

   NOTE: *This will search for all active incident records with* ***EIT*** *as one of its tags. The EIT tag is something Cloud Dimensions employees have created to help distinguish internal testing incidents from customer incidents – it stands for* ***Employee Infinity Testing****.*

   The filter should look like this:

3. Click **Run** to apply the filter.

   There should be two incident records returned.

4. Open the filter condition builder again to save the filter for later use.

5. Click **Save…**

6. Enter **Infinity Testing** into the **Save as** field.

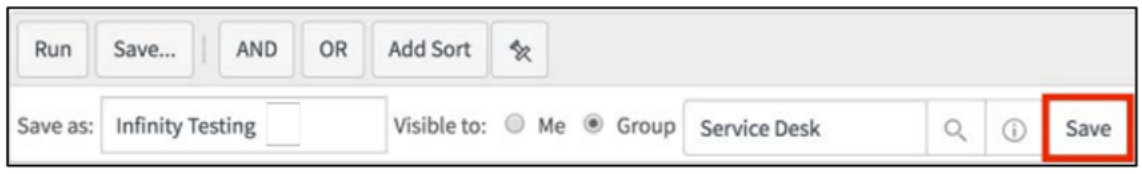
7. Next, select **Group** for **Visible to**.

   NOTE: *The ability to select a group to share wi dth others is provided by additional user permissions. For this exercise, the Service Desk group was granted the* ***filter_group*** *role.*

   ---

8. Input **Service Desk** into the group reference field to share this filter with its members.

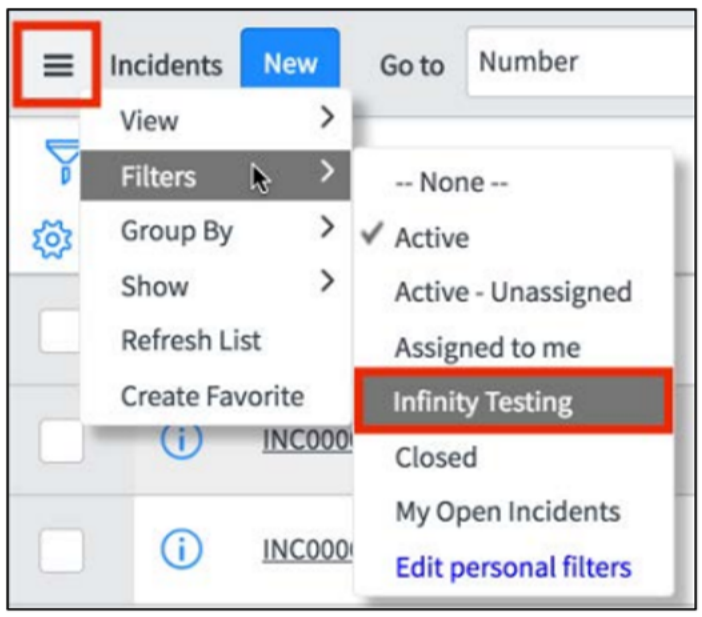
9. Click **Save**:



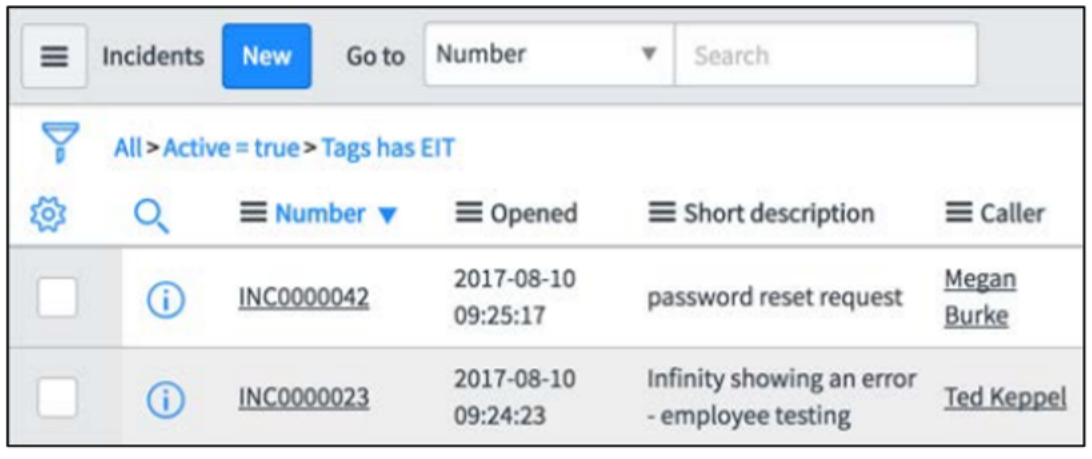
10. Impersonate **Megan Burke**, another member of the Service Desk group, to confirm the filter is now available for the Service Desk group.

11. **Incident > Open**.

12. Open the **List Context Menu** and select **Filters**, then finally select **Infinity Testing**:



13. The filtered incident list, containing two records, appears:



D. Locate a Missing Incident

Winnie Reich has emailed Kevin Edd to report an incident submitted by another employee that did not follow the current **EIT** tagging convention.

INC0000061 was submitted by Alissa Mountjoy and will need to be updated to include the **EIT** tag as the incident reports an error found with the Infinity holographic settings page.

1. Impersonate **Kevin Edd**.
2. Using methods of your choice, locate and open Alissa's incident record; **INC0000061**.

   HINT: Access the active record by navigating to an appropriate module and using list column header searches, filter conditions, or a global search for **INC0000061**.
3. With the record form displayed, open **More options** from the form header:

4. Click **Add Tag**.
5. Type **EIT** into the **Add tag…** field.
6. Press **Enter** on your keyboard to add the tag to the incident record:

7. **Update**.
8. Apply the **Infinity Testing** filter to the incident list to confirm all three Infinity testing records display:

Number ▼	Priority	State	Caller	Category	Subcategory	Short description
INC0000061	5 - Planning	New	Alissa Mountjoy	Network		Infinity holographic settings page will not display
INC0000042	5 - Planning	New	Megan Burke	Database		password reset request
INC0000023	5 - Planning	New	Ted Keppel	Inquiry / Help		Infinity showing an error - employee testing

E. Update Infinity Incident Records

Now that all Infinity incident records are accounted for, Winnie Reich has asked Kevin Edd to ensure all records' categories are accurate based on the issue reported and described.

Use the inline editor to update a record's category value right from the list.

1. Find **INC0000042** and double-click on the category, **Database**:

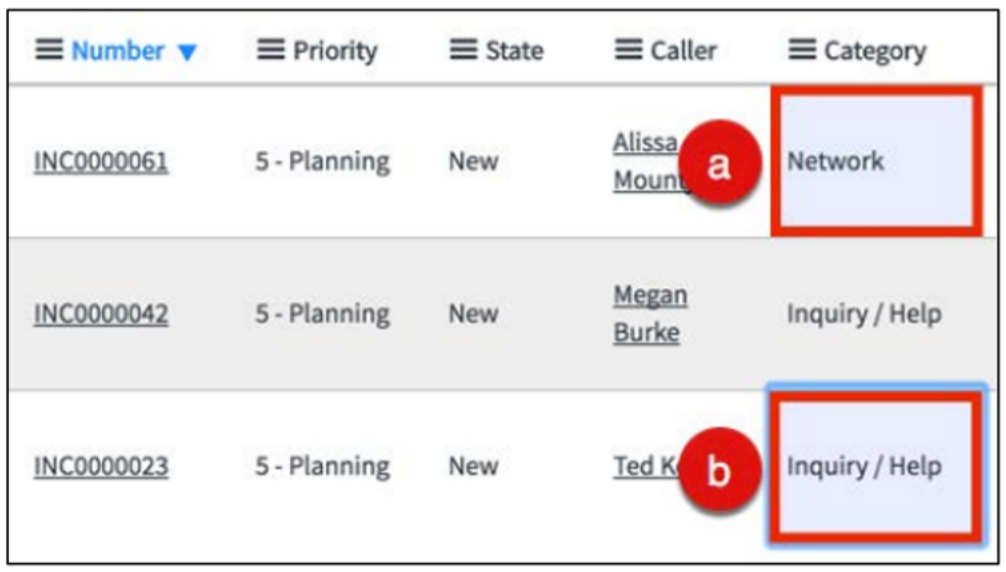
2. Use the Category drop-down to select **Inquiry / Help.**

3. Click the **Save** icon (green checkmark) to update the record:

4. Select multiple records to update field values with one set of steps:

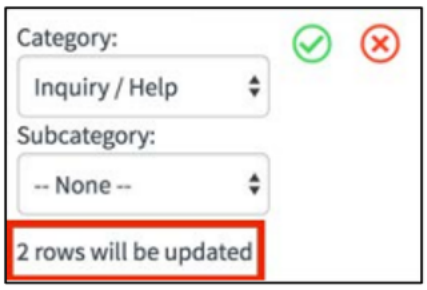
   a) Press **Shift** and click the **Network** category for INC0000061

   b) Hold **Shift** + **Ctrl** (**Shift** + **Command** on Mac) and click the **Inquiry / Help** category for INC0000023



5. Double-click on the **Inquiry / Help** category value for INC0000023 to open the **Category** drop-down.

6. Notice it indicates two records will be updated:



7. Use the **Category** drop-down to select **Software**.

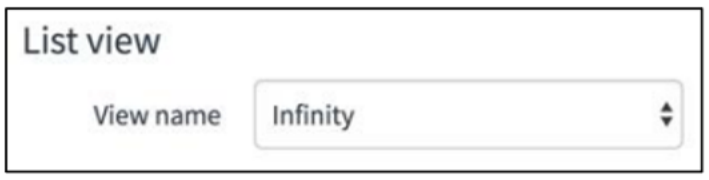
8. **Save** to update both records.

9. Your Infinity incident list should look like the following:

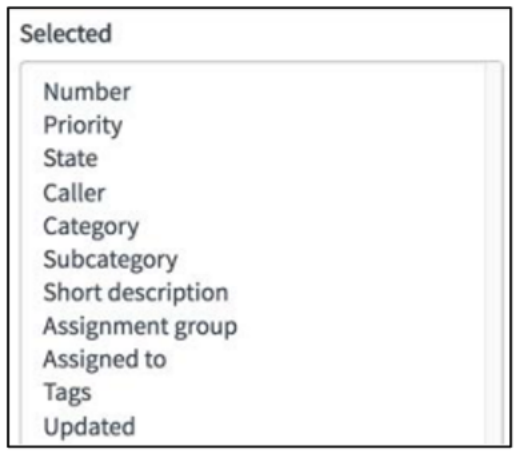
Number ▼	Priority	State	Caller	Category
INC0000061	5 - Planning	New	Alissa Mountjoy	Software
INC0000042	5 - Planning	New	Megan Burke	Inquiry / Help
INC0000023	5 - Planning	New	Ted Keppel	Software

LAB VERIFICATION

Infinity Incident List View



Infinity Incident List Fields



Infinity Testing List Filter

Updated Infinity Incident Records

Number ▼	Priority	State	Caller	Category
INC0000061	5 - Planning	New	Alissa Mountjoy	Software
INC0000042	5 - Planning	New	Megan Burke	Inquiry / Help
INC0000023	5 - Planning	New	Ted Keppel	Software

Excellent! By completing this lab successfully, you have set things up perfectly for the next.

Module 1 – 1.3 Forms

servicenow

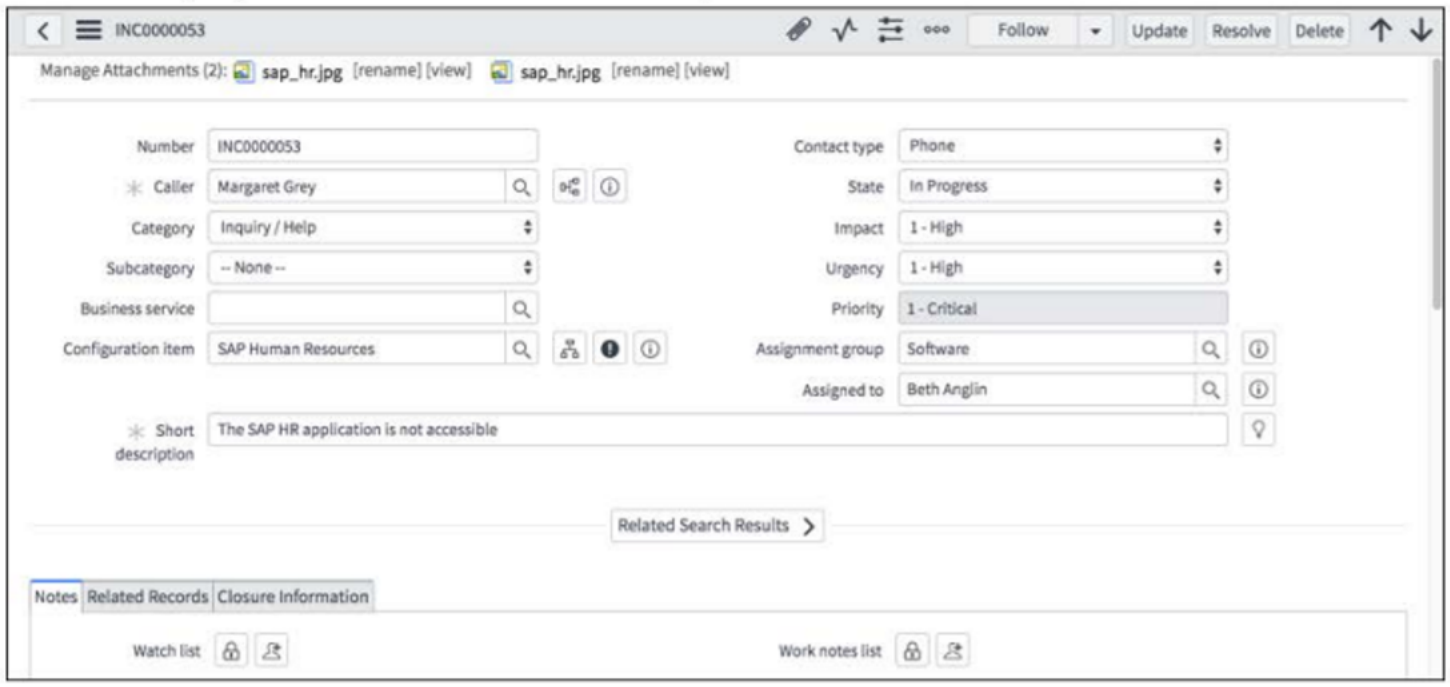
Objectives

- What is a Form?
  - Header Icons
  - Field Types
- Formatters and Related Lists
- Configuration
- Personalization
- Templates
- Saving Forms

What is a Form?

servicenow

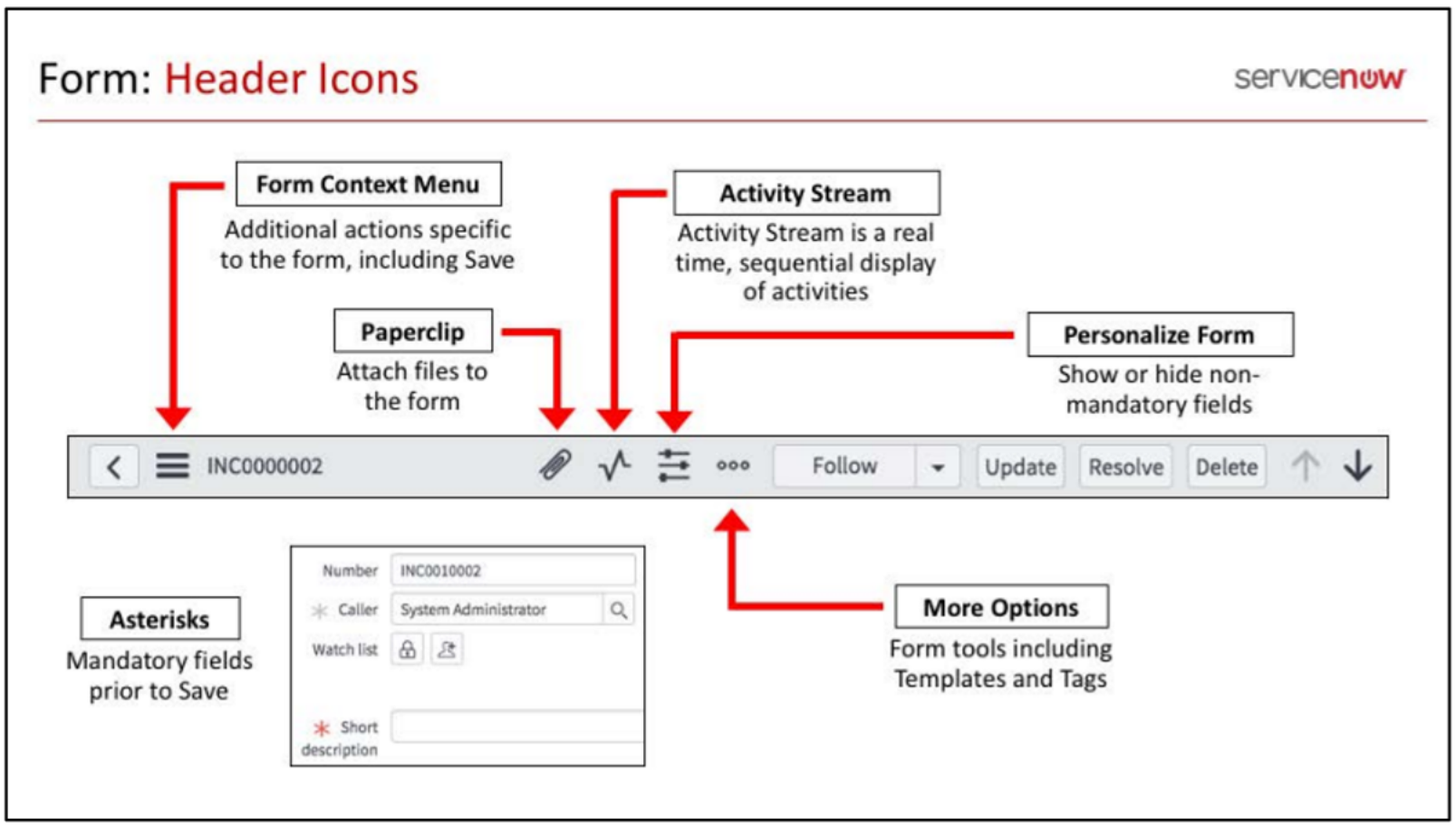
A **form** displays fields from one record; users can view and edit the record data



A form displays information from one record in a table. The specific information depends on the type of record displayed. Users can view and edit records in forms. Administrators can configure what appears on forms.

In addition to fields, the form can also contain sections and Related Lists. Related Lists show records in tables that have a relationship to the current record. For example, the User form features Roles and Groups Related Lists. Related Lists do not appear on a form until a record has been saved to the database.

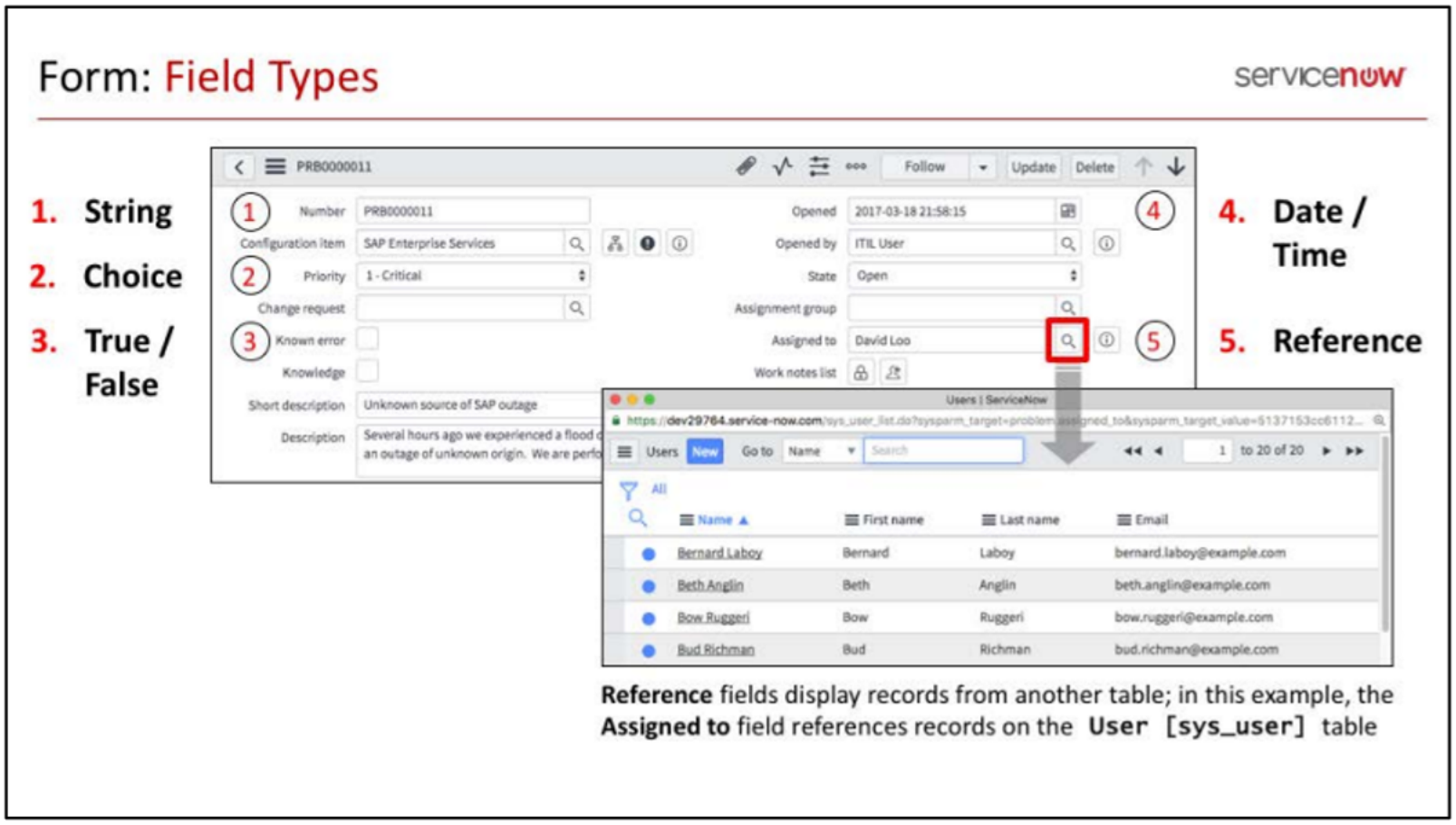
A form can load directly by searching on a record number in the **Global Text Search** or by clicking a record in a list.



Each form has different fields, UI actions, and options specific to the application under which it was created.

However, all forms have certain icons and functionalities in common:

- The **Form Context Menu** provides additional options specific to the form. **Save** can be found in the Form Context Menu and be used to save a form while remaining on the page.
- Use the paperclip icon to attach, remove, or rename files on a form.
- **Show Activity Stream** will display a time stamped history of all actions taken within a record.
- **Personalize** a form to show or hide important fields. **NOTE:** Mandatory fields can not be hidden.
- All fields marked with an asterisk are mandatory and must be filled out prior to saving the form. **NOTE:** The asterisk is red prior to filling out the field and grey once information has been entered
- Click **More options** to tag a form, use templates, send an email, and more.



Forms include various field types, each with unique attributes.

Some common field types include:

1. **String**: Freely populated using letters, numbers, and special characters. For 254 characters or less, the string field will be a single-line text field. Anything 255 characters or over will appear as a multi-line text box.
2. **Choice**: Drop down list of choices that can be configured.
3. **True/False**: Boolean field that appears as a check box.
4. **Date/Time**: Day and time of day, which can be selected with a calendar widget.
5. **Reference**: Query that displays records from another table.

Form: Formatters and Related Lists

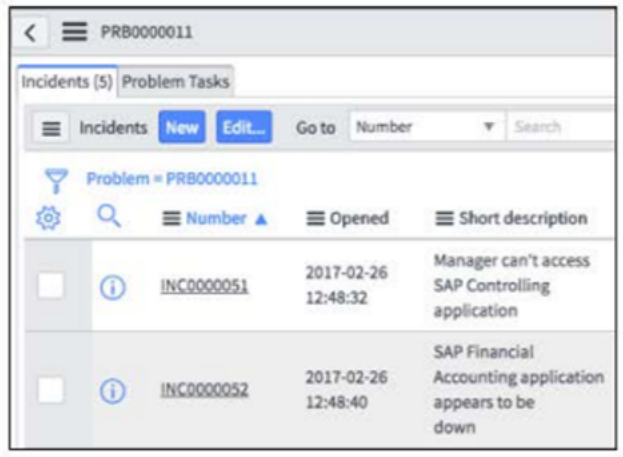
servicenow

Formatter

A **formatter** is a form element used to display information that is not a field in the record

The activity formatter provides an easy way to track items not saved with a field in the record, for example, journal fields like comments and work notes

Related List



Related lists appear on forms and show records in tables that have relationships to the current record

Examples of **formatters** in the base platform include:

- **Activity formatter:** Displays the list of activities, or history, on a task form
- **Process flow formatter:** Displays the different stages in a linear process flow across the top of a record
- **Parent breadcrumbs formatter:** Provides breadcrumbs to show the parent or parents of the current task
- **Approval summarizer formatter:** Displays dynamic summary information about the request being approved
- **CI relations formatter:** Displays on the CI form a toolbar for viewing the relationships between the current CI and related CIs

Like any other list, users can view and modify information in **related lists,** as well as add a new record to the database. A default filter that is applied to a related list when a form loads can be created.

Administrators can configure related lists to appear on forms and in hierarchical lists. Related lists do not have a size limit.

If there are many related lists on a form or many records in the related lists, the form may load slowly.

Form: Configuration and Personalization

servicenow

Configuration

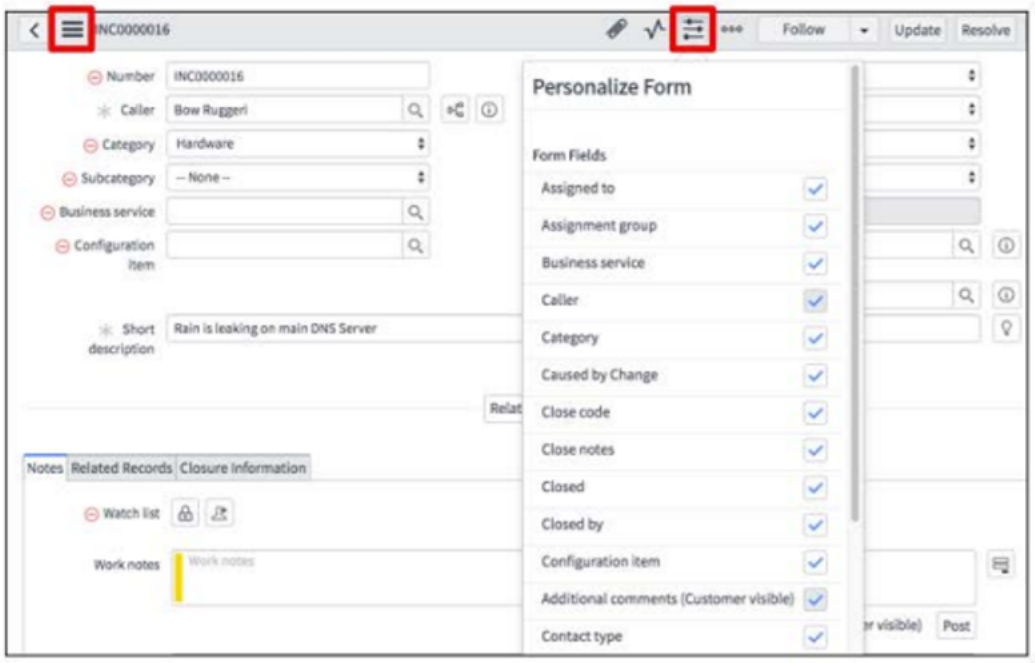
Configure a form to show or hide fields from a view, as well as change their position on screen

Click the form context menu icon and select **Configure > Form Layout**

Personalization

Click the **Personalize Form** icon (⚙) to personalize which fields display

Personal form customizations will only affect what you see on a form – these changes do not affect others



With the personalize_form role, users can configure a form to show or hide fields from a view. Nfields can even be created on the table that is associated with the form, although this is not best practice.

Using the slushbucket, select the fields and the order in which you want them to appear. Available items that appear in green followed by a plus (+) sign represent reference fields. Accessing these fields on related tables is referred to as dot-walking.

Warning: It is not recommended to add the same field to more than one section of a form unless the field displays read-only data. Having two or more instances of an editable field can cause data loss and prevent the proper functioning of UI and data policies.

When the form personalization feature is activated, users can personalize fields to appear on a specific form view according to individual preferences.

In contrast to configuring a form, personalizing a form does not enable users to perform the following actions:

- Change the order of fields on the form
- Add fields that are not configured to appear on the form
- Hide mandatory fields

Form: Templates

servicenow

Templates allow fields to be populated automatically, simplifying the process of submitting new records

Click the **More options** icon (•••) from the form header, then click **Toggle Template Bar** to work with templates

Use the template bar at the bottom of the form to manually apply, create, or edit templates

To use a template, populate the most-used fields for a specific table, save it as a template, and then make the template accessible to users. Users can manually apply a template when creating records, or an administrator can define scripts to apply templates automatically. Fields updated by the application of a template will have a checkmark icon next to the field label.

Create templates for the forms that are used frequently, such as incident, problem, and change. There is no limit to the number of templates that a user can create or access, but having many templates for each form makes the templates more complex to manage.

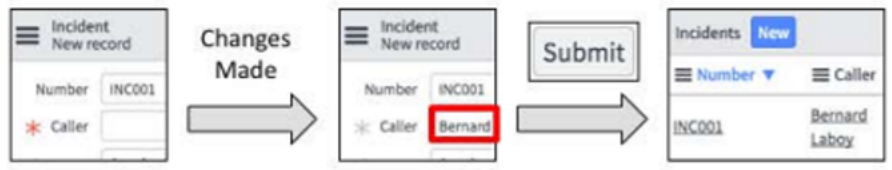
NOTE: Template creation should be restricted to select groups as it can be used to by-pass process, like mandatory fields, UI policies, etc. This is especially important for any record using condition based workflows.

Form: Saving Forms

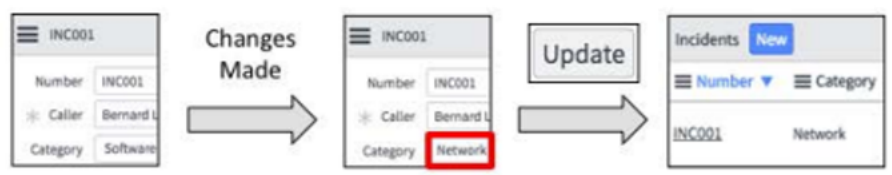
servicenow

Save records by using one of the following methods:

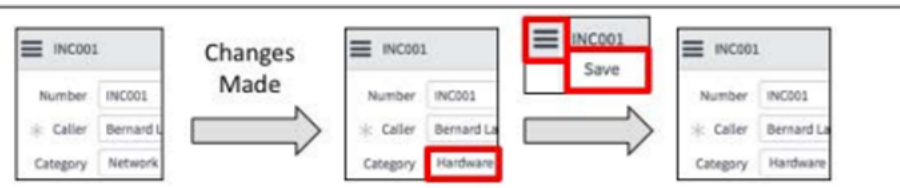
Click **Submit** to save changes on a new form and return to the previously viewed page



Click **Update** to save changes on an existing record and return to the previously viewed page



Select the Form Context menu icon in the header bar, then select **Save** to save changes without leaving the form view



The Insert, and Insert and Stay options are disabled by default for task records such as incidents and change requests but these options can be configured for task records.

NOTE: These options are enabled by default on User and CMDB records because they enable bulk entry of similar items.

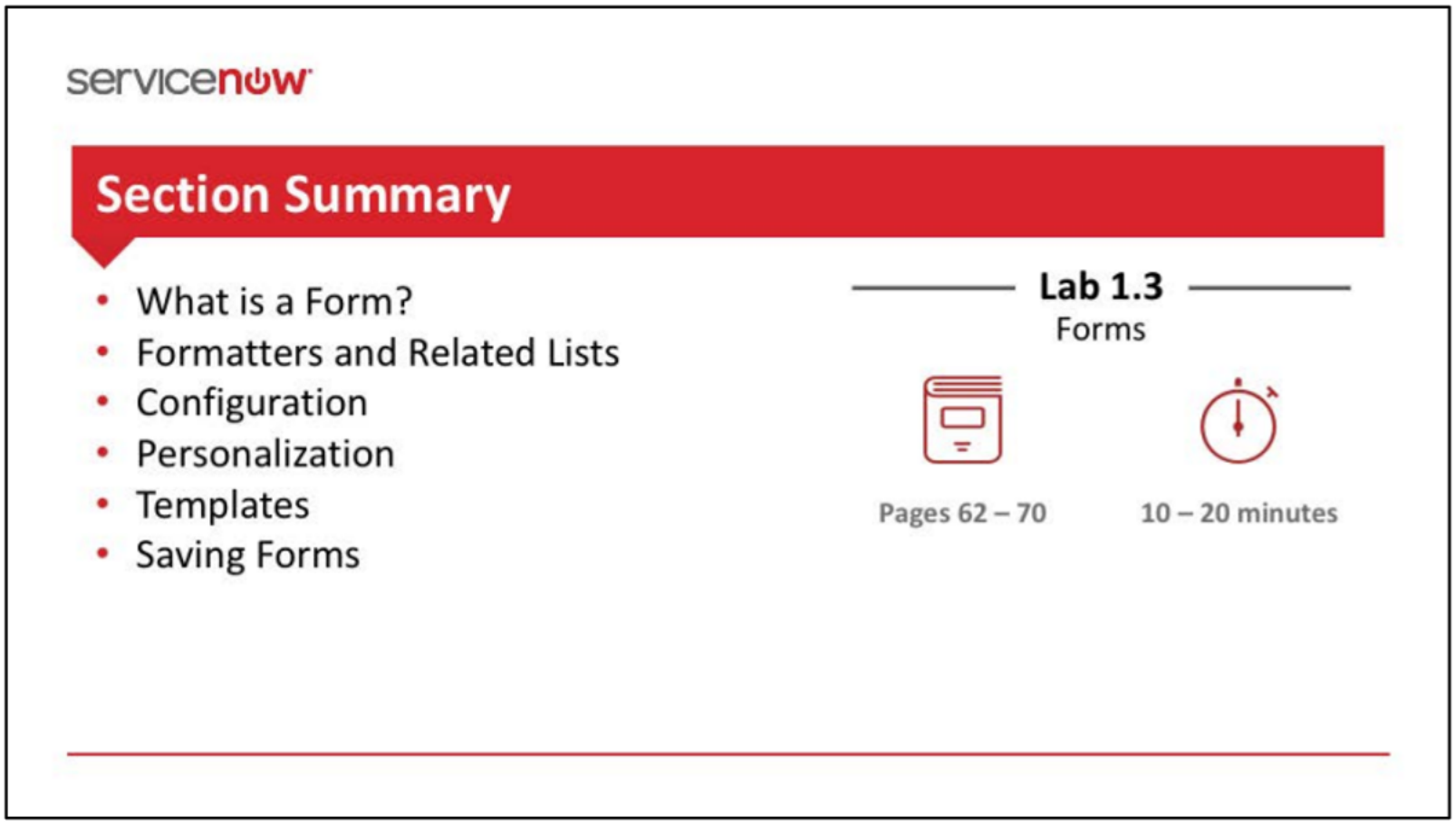
Once enabled, select the Form Context menu icon in the header bar then select **Insert** or **Insert and Stay** to save a new record to the database instead of updating the current item.

There is no "Copy" but **Insert** emulates copy functionality and leaves the form.

Insert and Stay does the same but stays on the form.

When a form is saved, all the text in the Work Notes field is recorded to the Activity Log field. Work Notes and Additional Comments are fields that share information with various users associated to certain record types like incident or problem. Additional Comments are visible to all users accessing the record, whereas Work Notes are visible to only users with the **itil** role. The content in Additional Comments is emailed to the Watch List and Caller, and the content in Work Notes is emailed to the ITIL Watch List and Assigned to user when the form is saved.

NOTE: If you make changes to an existing record and then attempt to leave the form (whether using web browser controls such as the 'Back' button, or through the ServiceNow user interface), you will be prompted with a message asking if you are sure you want to leave the record without saving.



Lab 1.3 – Forms:

- Create and configure a new form view using the Form Designer
- Create and update Infinity incident records

LAB

Forms

1.3

10 – 20 minutes

Lab Goal

Lab Dependency: Requires the completion of Lab 1.2.

This lab will show you how to do the following:

- Create and configure a new form view using the Form Designer
- Create and update Infinity incident records

Internal employee testing of Infinity has proven worthwhile for a number of reasons.

Winnie Reich will lead an initiative to further improve and organize Infinity testing support by creating a form view on the incident table containing the appropriate fields and being capable of capturing field values that accurately identify reported issues.

A. Create the Infinity Form View

1. Impersonate **Winnie Reich**.

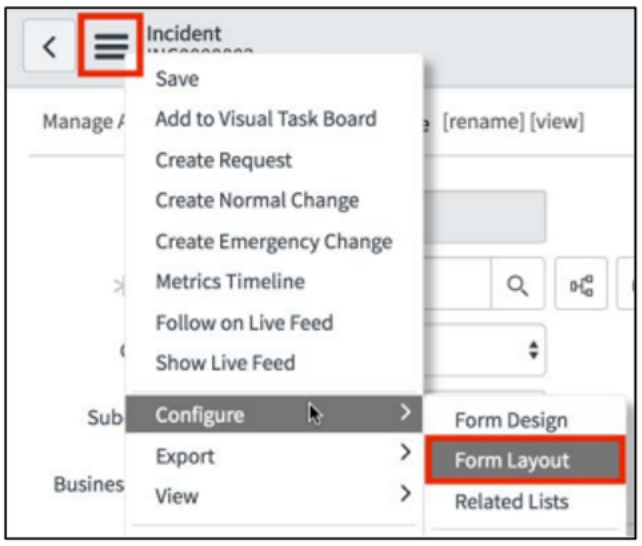
NOTE: *Winnie is inheriting the* ***personalize*** *role from a group that she belongs to, which allows her to configure forms and create new views.*

---

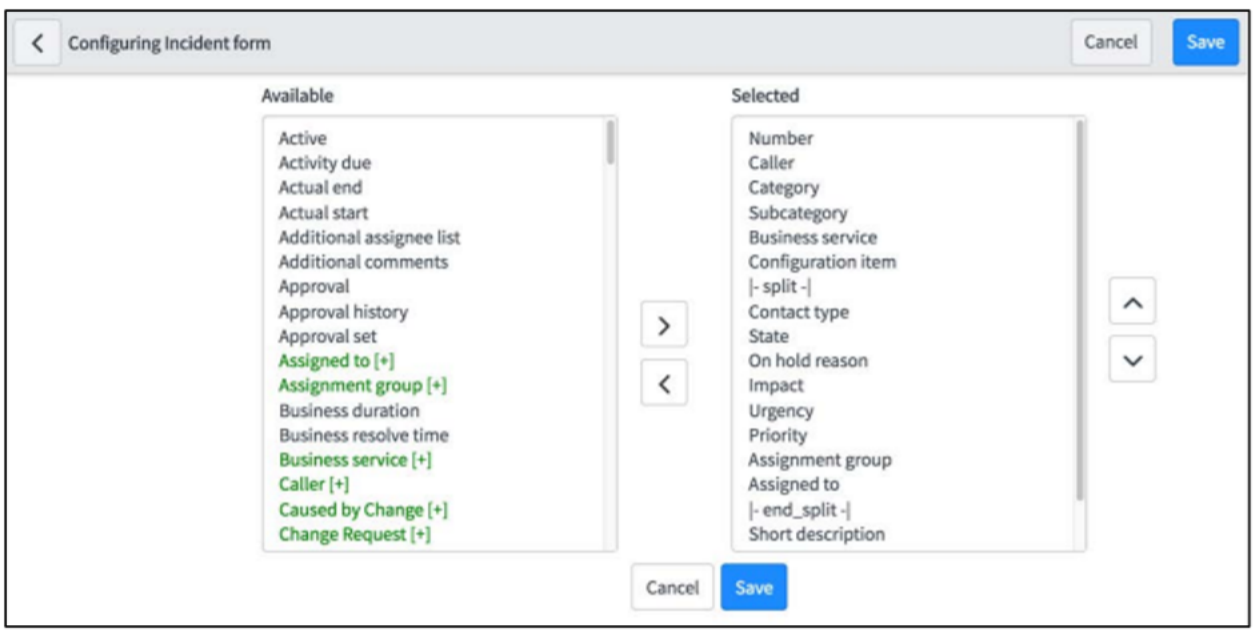
2. **Incident > Open**.

3. Open the record for **INC0000061**.

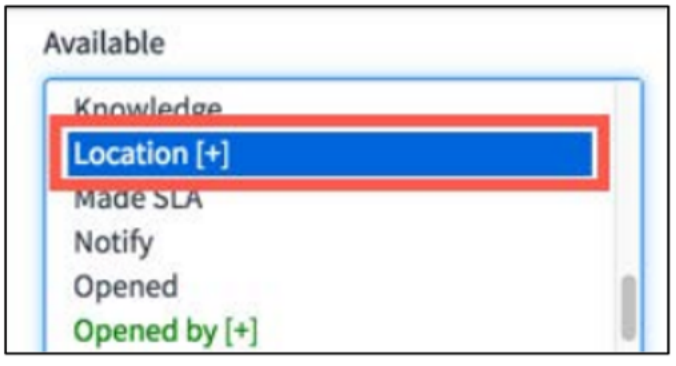
4. Open the **Form Context Menu** and select **Configure**, then finally select **Form Layout**:



The **Configuring Incident form** page displays:



5. Find and highlight the **Location** field under the Available list:



6. Click the **Add** button (>) between the Available and Selected list:

7. Doing this adds the Location field to the bottom of the **Selected** list.

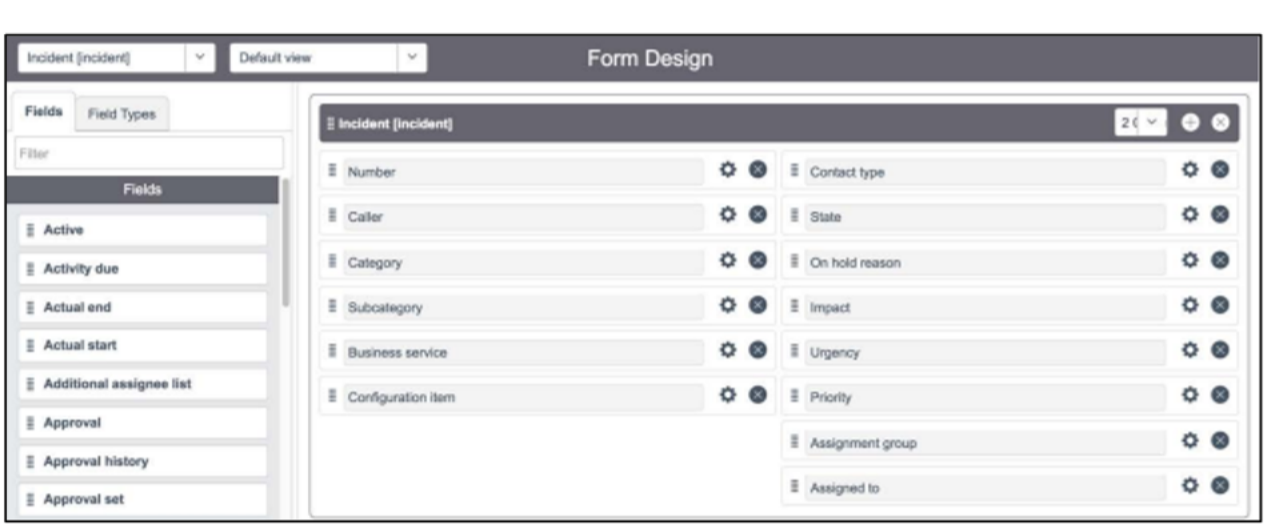
8. Use the **Move up** button (^) to move the **Location** field under **Configuration item**.

9. From the **Configuring incident form** page, click **Save** to return to the incident form.

Modify Form View with the Form Designer

1. Open the **Form Context Menu** and select **Configure**, but then select **Form Design**.

   The **Form Designer** will open in a new tab or window. Go to the page that looks like this:



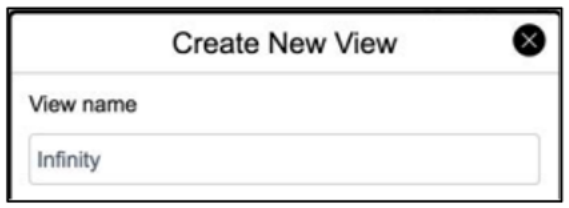
The **Form Designer** offers an improved experience because of its graphical user interface, making it easier to visualize the form view's end result.

Additionally, there are several configuration options available in this single interface.

At the top left of the page are two drop-down menus in the header; the menu on the left indicates the table the form view is associated with, and the menu on the right includes the various views defined for the selected table.

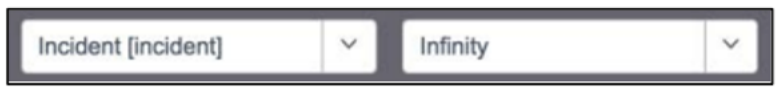
2. Open the view (right) menu and select **New…** at the bottom of the list**.**

3. Enter the View name: **Infinity**:



4. Click **OK**.

   Notice the new view is automatically selected in the view menu on the page's header:



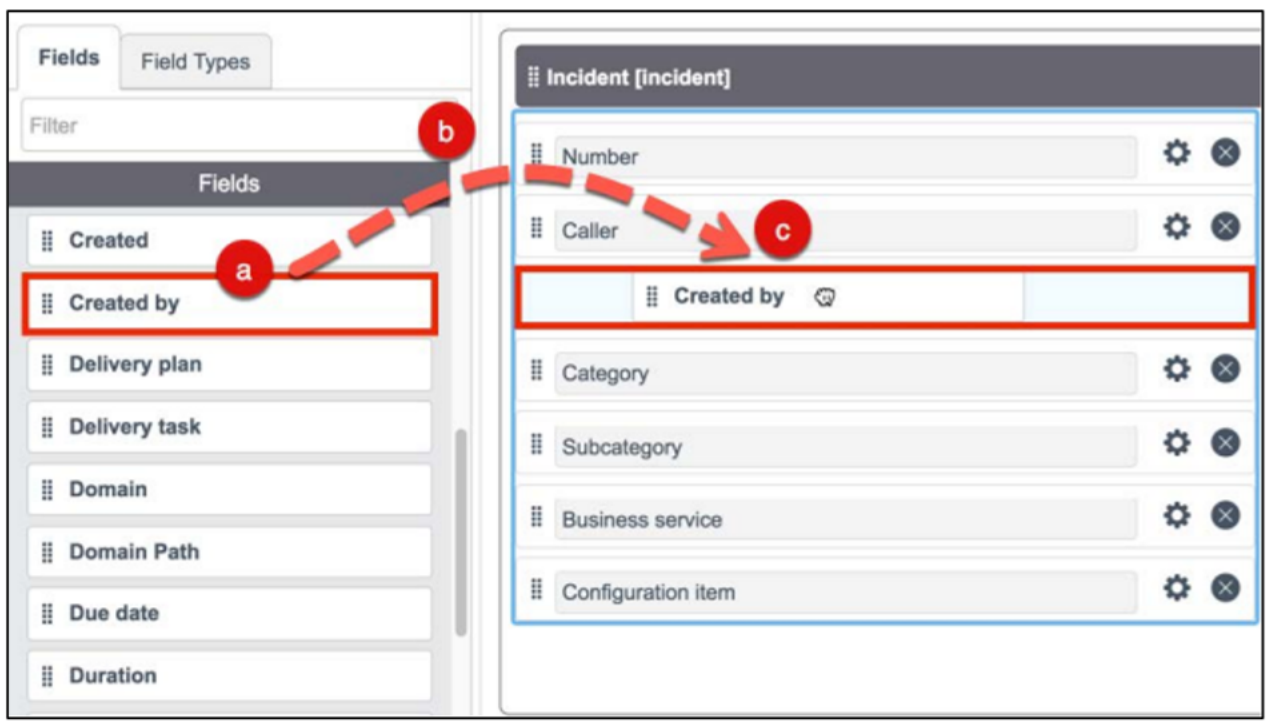
5. On the **Fields** tab of the Field Navigator on the left, scroll down to locate the **Created by** field.

6. Add the **Created by** field to the form view:

   a) Click and hold on the **Created by** field

   b) Drag **Created by** to the form layout, between **Caller** and **Category**

   c) Release your click to add the field



7. Repeat these steps to add the **Updated** and **Updated by** fields to the form layout, within the Incident section.

NOTE: *We will be reordering fields later.*

---

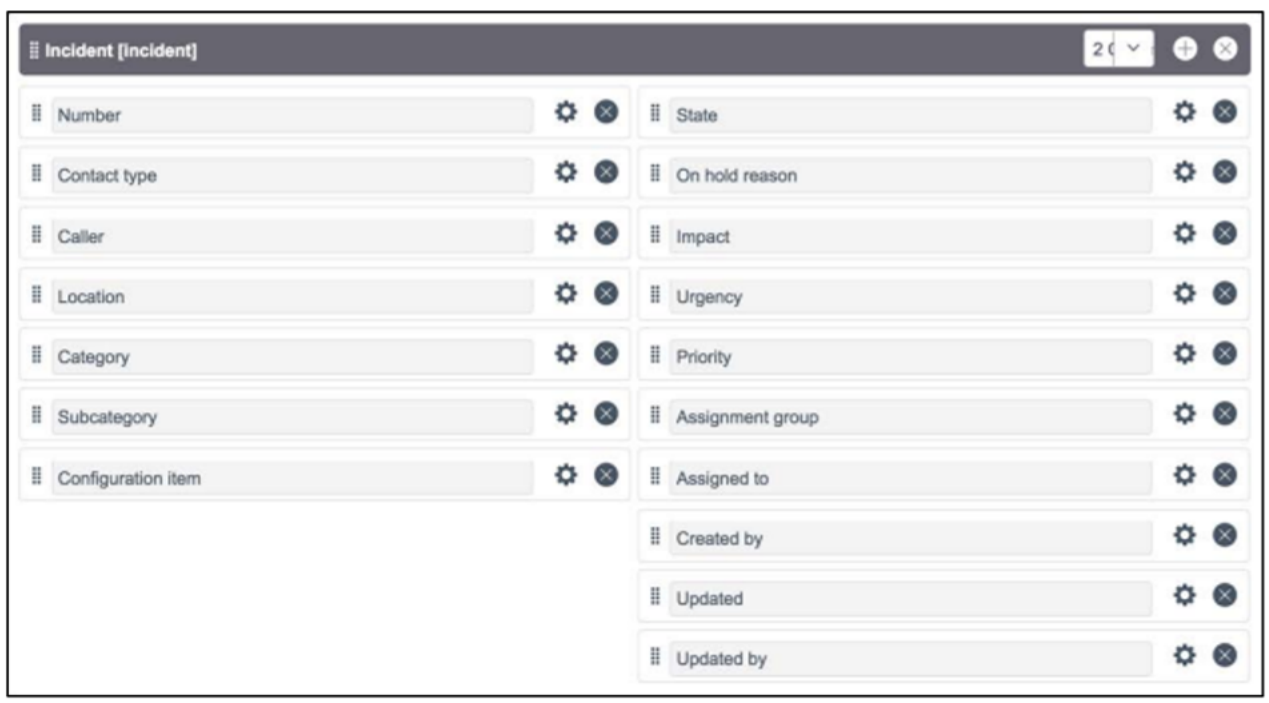
8. Find the **Business service** field on the form layout, then click the **Remove this field** icon (circled x) to remove it from the view:

NOTE: *Removing a field returns it to the* ***Fields*** *tab of the Field Navigator, so it may be re-added if desired.*

---

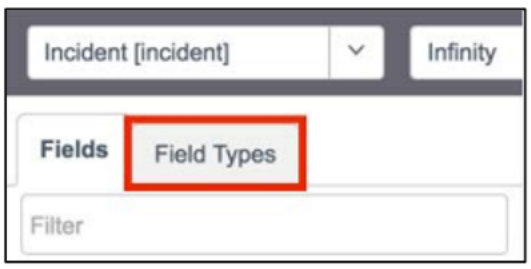
9. Click and drag the **Contact type** field to be listed beneath the **Number** field:

10. Repeat this step to reorganize the fields in the Incident section to match this layout:



Define a New Field

1. From the Field Navigator, click the **Field Types** tab to add a new field to the form layout:



2. Scroll down to find the **True/False** field type, then add the field to the form layout under the **Caller** field:

3. Click the **Edit this field** icon (gear) to configure the field's properties:

4. Input the following values:

   Label: **Employee**
   Name: **u_employee**

   NOTE: *The name features the prefix* ***u_*** *to indicate it is a user-created item. This is a common naming convention used throughout ServiceNow.*

   ---

5. Close the **Properties** window by clicking the close icon (circled x).

6. Click the **Save** action from the page header to save the form view:

7. Close the **Form Designer** tab/window, and return to the ServiceNow instance.

8. Use the **Form Context Menu** to reload the form (**Reload form**).

9. Now open the **Form Context Menu** and select **View**, then finally select **Infinity**.

   The new **Infinity** form view should load and display as designed!

B. Create and Update Incident Records

With the new form view defined, the next step is to create and update Infinity testing records by inputting the information into the correct fields.

Steps for creating a new single incident with the State “In Progress” and EIT tag, and “employee” checkbox selected. Make sure Assignment group is set as well.

1. **Incident > Create New**.

2. Fill in the rest of the fields as follows:

   Contact type: **Walk-in**
   Caller: **Buster Wubbel**
   Employee: **[checked]**
   Category: **Software**
   State: **In Progress**
   Assignment group: **Service Desk**
   Short description: **Issue discovered with two step authentication**
   Description: **Authentication requirements for logging into Infinity are not working as expected.**

   ***NOTE**: The **Location** field may have populated automatically because of the Caller value inputted.*

   ---

3. **Save**.

4. Open **More options** from the form header.

5. Click **Add Tag**.

6. Search for and select the **EIT** tag.

7. **Update**.

8. If the active incidents list does not display, navigate to **Incident > Open**.

9. Apply the **Infinity Testing** filter.

   There are now four total open Infinity employee incident records:

All > Active = true > Tags has EIT

Number	Priority	State	Caller	Category	Subcategory	Short description	Assignment group	Assigned to	Tags
INC0010001	5 - Planning	In Progress	Buster Wubbel	Software		Issue discovered with two step authentication	Service Desk		EIT
INC0000061	5 - Planning	New	Alissa Mountjoy	Software		Infinity holographic settings page will not display			EIT
INC0000042	5 - Planning	New	Megan Burke	Inquiry / Help		password reset request			EIT
INC0000023	5 - Planning	New	Ted Keppel	Software		Infinity showing an error - employee testing			EIT

NOTE: *The new incident number may be different in your instance.*

Challenge

Update records to change the state, assignment group and check "Employee" checkbox.

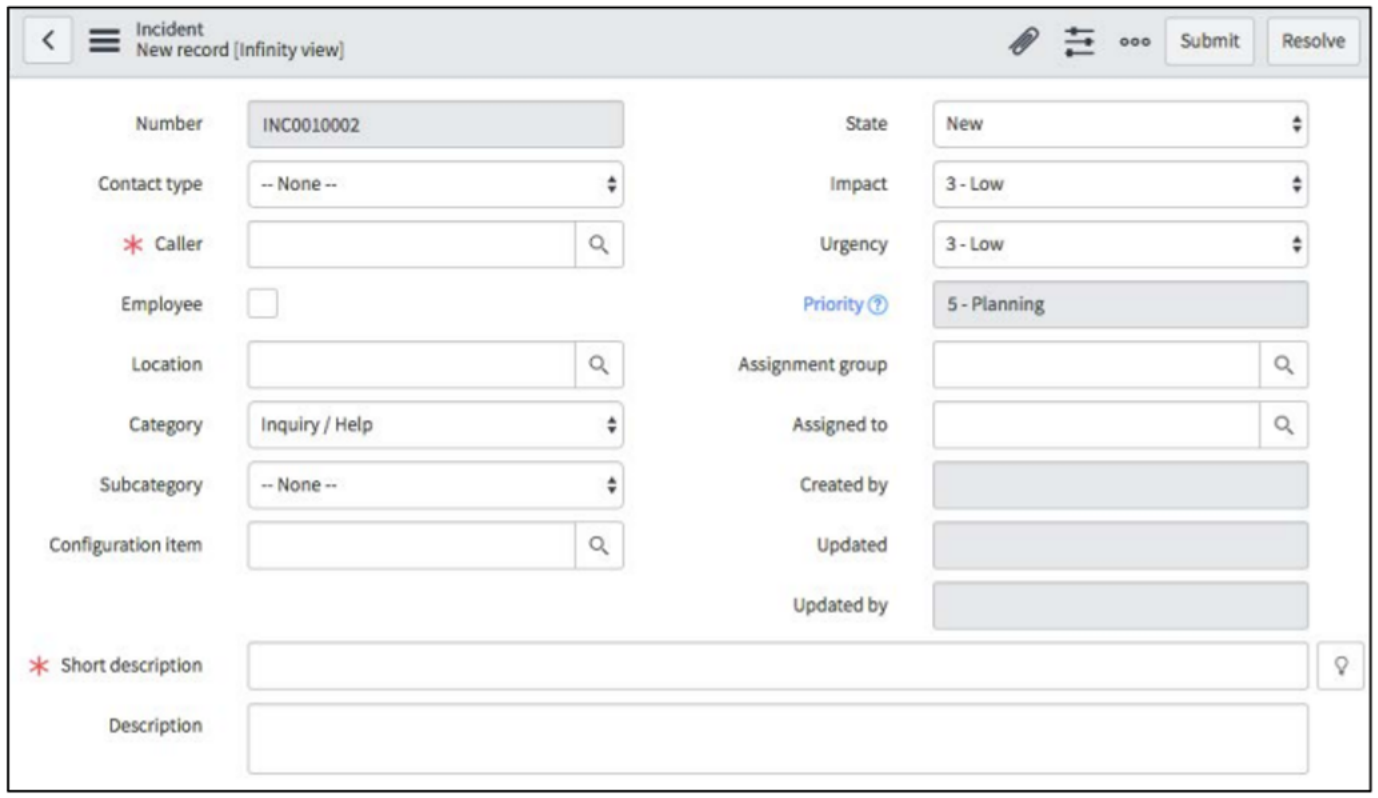
As **Kevin Edd**, use the strategies of your choice to update the following records:

Number	Employee	State	Assignment group
INC0000023	True	In Progress	Service Desk
INC0000042	True	In progress	Service Desk
INC0000061	True	On Hold	Service Desk

NOTE: *If required, set the value for the* ***On hold reason*** *field to* ***Awaiting Caller****.*

LAB VERIFICATION

Infinity Incident Form View



Updated Infinity Incident Records

Number	Priority	State	Caller	Category	Subcategory	Short description	Assignment group	Assigned to	Tags	Updated
INC0010001	5 - Planning	In Progress	Buster Wubbel	Software		Issue discovered with two step authentication	Service Desk		EIT	2018-02-26 09:09:43
INC0000061	5 - Planning	On Hold	Alissa Mountjoy	Software		Infinity holographic settings page will not display	Service Desk		EIT	2018-02-26 09:15:49
INC0000042	5 - Planning	In Progress	Megan Burke	Inquiry / Help		password reset request	Service Desk		EIT	2018-02-26 09:16:14
INC0000023	5 - Planning	In Progress	Ted Keppel	Software		Infinity showing an error - employee testing	Service Desk		EIT	2018-02-26 09:13:53

Well done – you have created a new form view!

Module 1 – 1.4 Branding

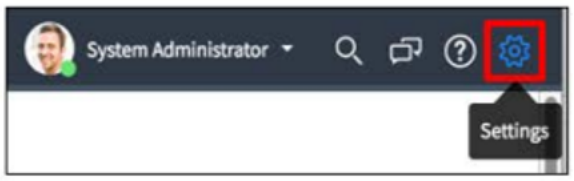
servicenow

Objectives

- Instance Settings
- Application Configuration: Guided Setup
- Basic Configuration: System Branding

Banner Frame: Instance Settings

servicenow



Click the **Settings** icon from the banner frame to customize your instance

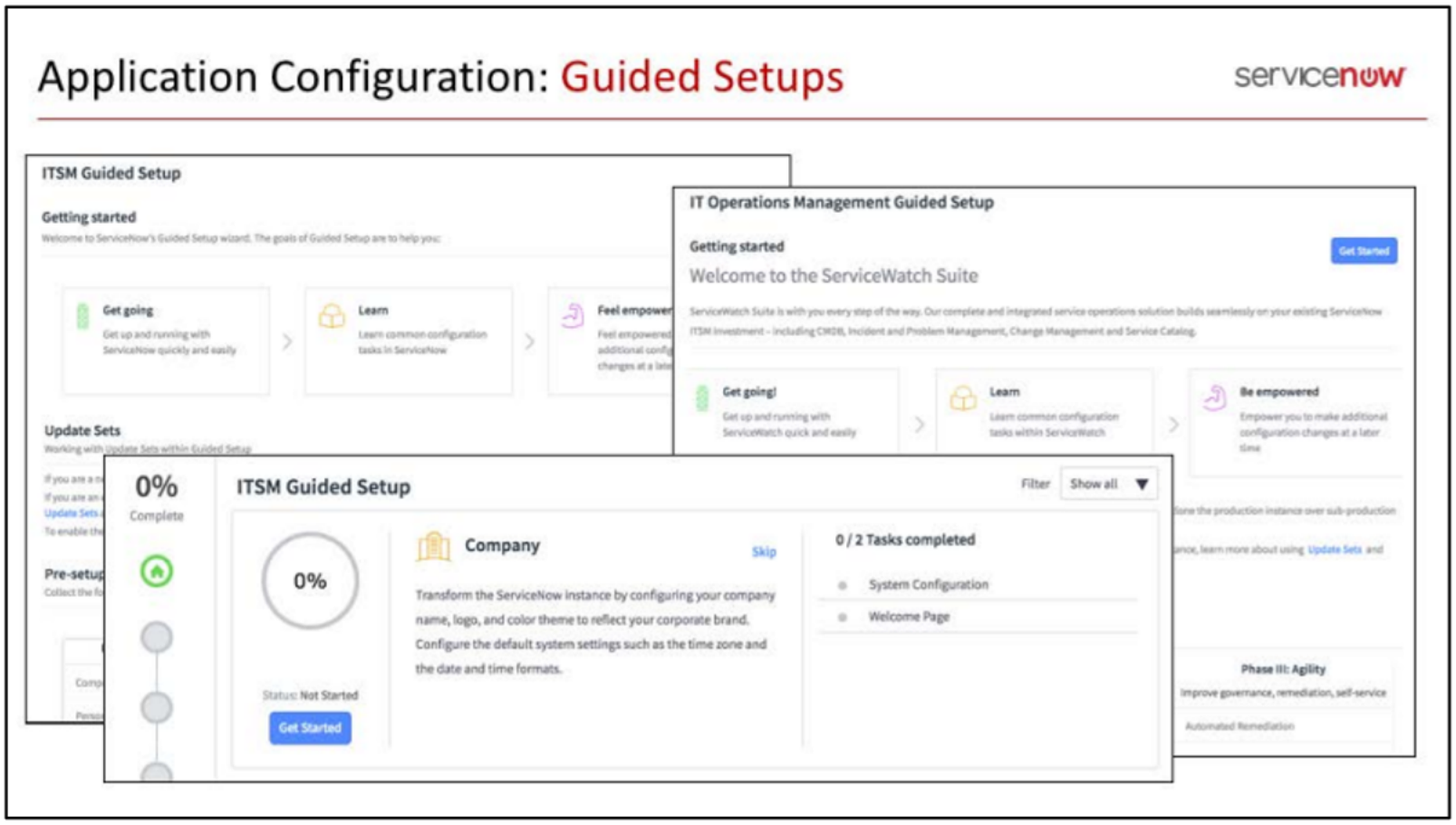
These settings affect only your user account and are retained each time you log in

The Settings icon (gear) in the upper-righthand side of the Banner Frame contains additional settings and options for personalizing your view of the platform.

NOTE: Users may be limited to what settings they have access to based on their role.

After selecting the Settings icon, the categories on the left (General, Theme, Lists, Forms, Notifications, and Developer) provide different settings, including:

- **General Tab: Compact the user interface** optimizes the UI to display more information in the browser window when this setting is enabled
- **Theme Tab**: Select a theme for the user interface. Select the **System** theme to return to the default theme
- **Lists Tab: Wrap longer text in list columns** allows for long strings to wrap in list columns instead of appearing as one long line
- **Forms Tab:** Form sections and related lists appear in tabs when the **Tabbed forms** setting is enabled. Also **Related list loading** is used to determine when Related Lists load on forms.
- **Notifications Tab:** Allows you to enable various notification channels, as well as manage your notification subscriptions
- **Developer Tab**: Settings for ServiceNow Application developers



Users with an administrator role may use a **Guided Setup** to go through suggested steps to configure applications on the ServiceNow instance.

Using the **ITSM Guided Setup**, users can perform structured configuration activities that help configure ITSM applications and can monitor the progress. Each configuration activity in ITSM Guided Setup is designed to simplify the configurations by providing access to contextual embedded help, contextual documentation on the ServiceNow product documentation site, and guided tours (if available for an activity).

The ITSM Guided Setup configures the common platform settings through the following categories:

- **Company**: Activities under this category help you to configure company name, logo, and color theme to reflect your corporate brand and to configure the default system settings such as the time zone and the date and time formats
- **Connectivity**: Activities under this category help you to configure your ServiceNow instance to support inbound and outbound email notifications and to integrate it with your existing LDAP and Single Sign-On (SSO) solutions
- **People**: In case you do not use LDAP to import data into your ServiceNow instance, activities under this category can help you to import Users, Groups, Group Members, Companies, Departments, and Locations and to assign roles to groups

In addition to the ITSM Guided Setup, there are guided setups for ITOM and HR as well.

Basic Configuration: System Branding

servicenow

Browser Tab

Cloud Dimensions ServiceNow

Banner Image

servicenow

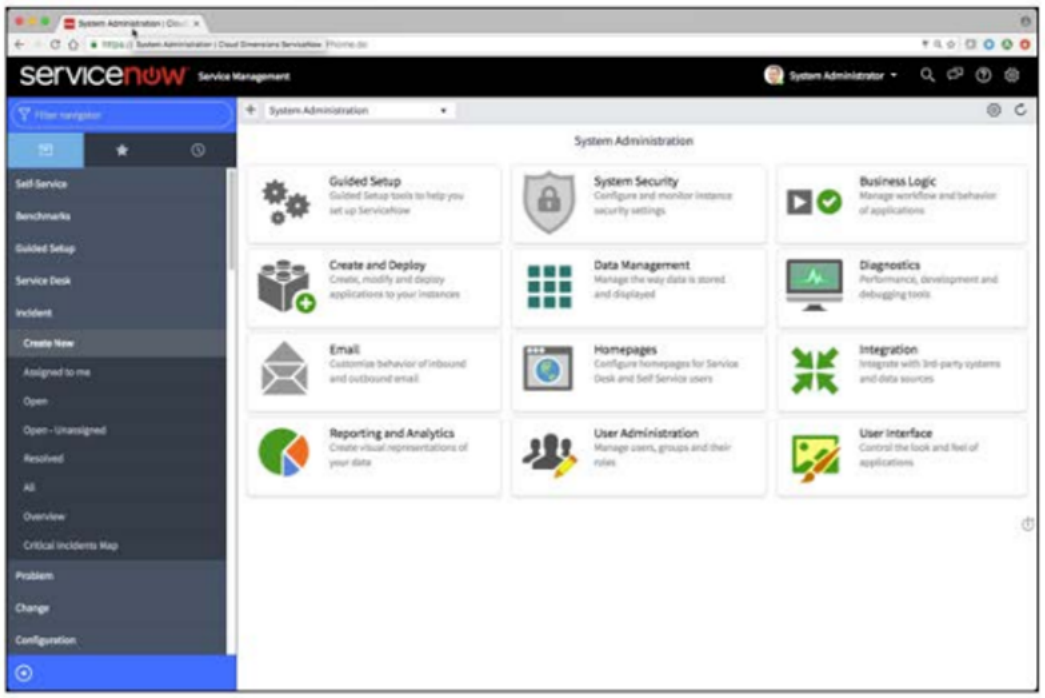
Navigation Colors:

System Properties

System Properties

Basic Configuration

Basic Configuration UI16



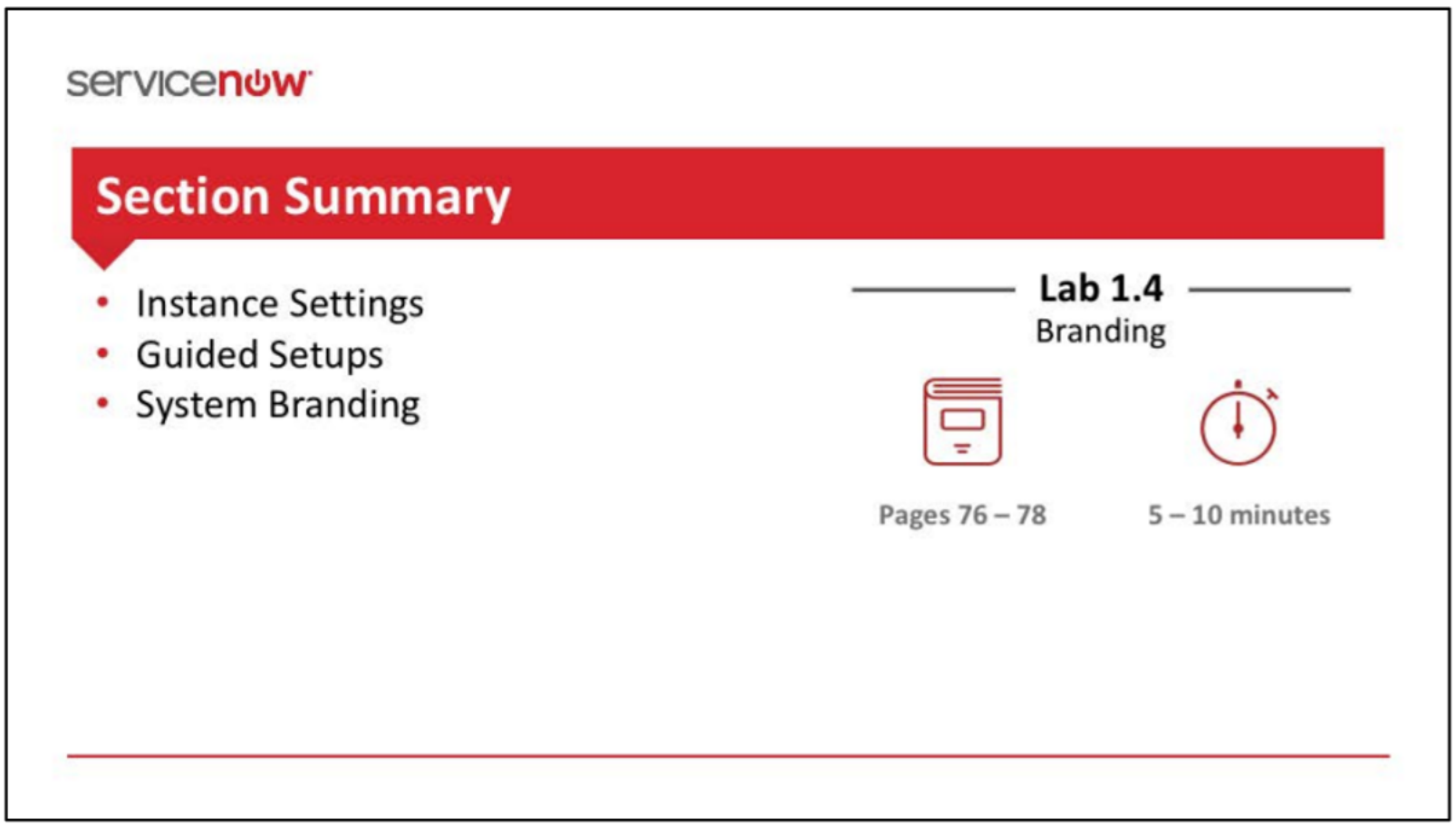
Many branding activities are grouped together in **System Properties > Basic Configuration UI16.**

Customization and branding options include:

- Banner image, text and colors
- Navigator background and text colors

Features which can also be customized:

- Browser tab title
- Color: Use the built-in color pickers to dynamically pick and preview branding options
- System date/time formatting



Lab 1.4 – Branding:

- Use the ITSM Guided Setup to apply branding to the instance

LAB

Branding

1.4

5 – 10 minutes

Lab Goal

This lab will show you how to do the following:

- Use the ITSM Guided Setup to apply branding to the instance

Cloud Dimensions recognizes the importance of aligning ServiceNow's branding with the rest of the organization so that users automatically feel familiar, seeing it as a trusted platform. Therefore the ServiceNow instance will be branded to achieve this.

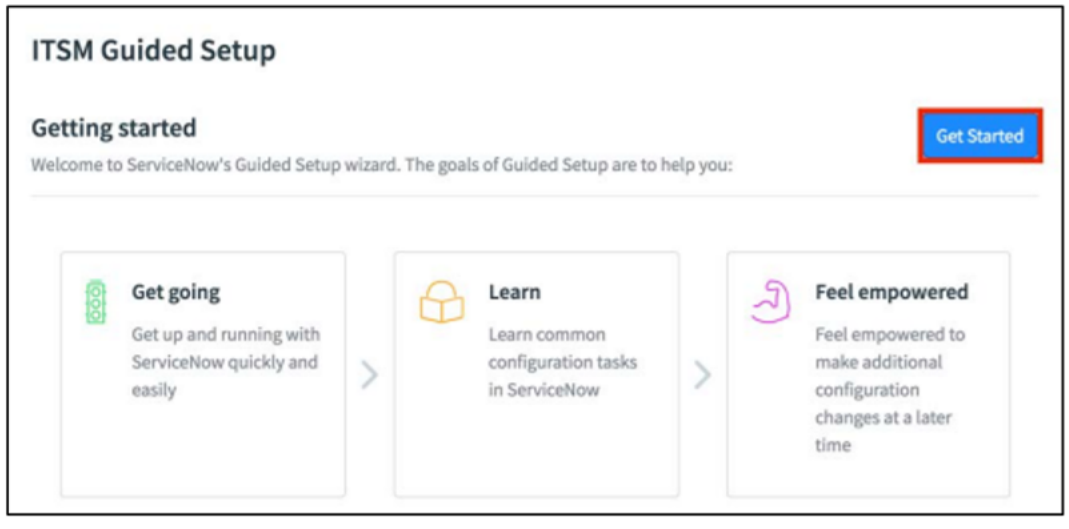
Required Resource: CloudDimensions-CD-Logo.png

A. Apply Branding with the ITSM Guided Setup

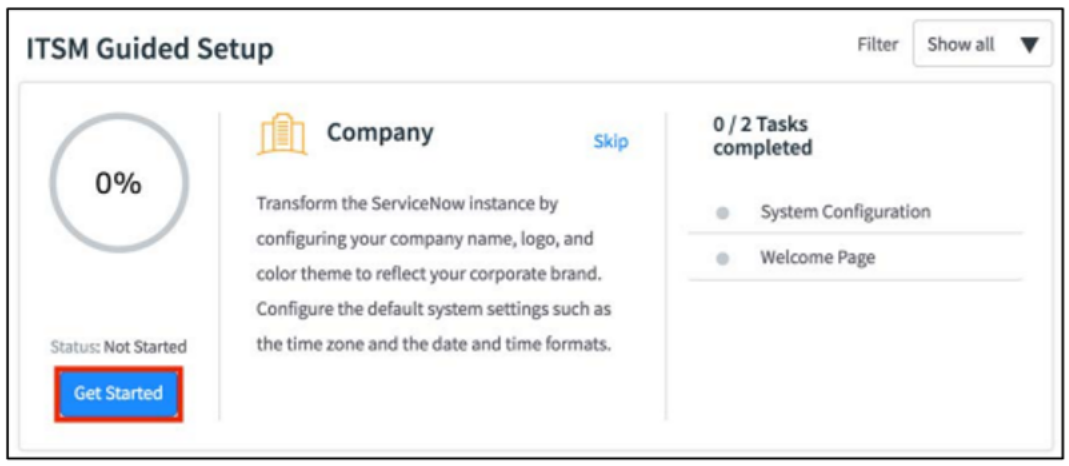
Begin the lab by impersonating the system administrator. This user has the appropriate permissions for defining platform-wide properties. We will work with the ITSM Guided Setup to get started on company branding.

1. Impersonate the **System Administrator**.
2. **Guided Setup > ITSM Guided Setup**.

   After the Guided Setup page loads, click the **Get Started** button:

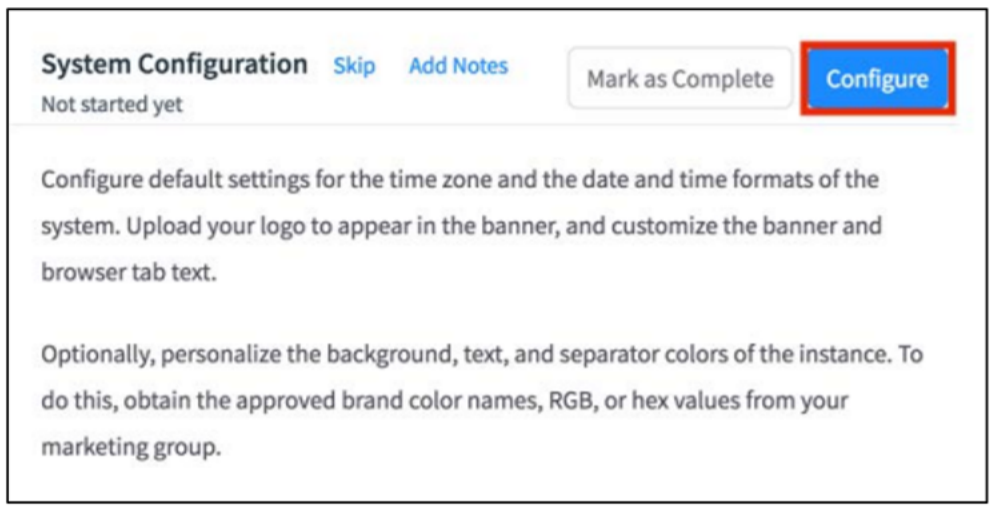


Next, click **Get Started** from the **Company** section:



The first task to complete is **System Configuration**.

3. Read the summary text, for what to expect, then click the **Configure** button:



4. Fill out the fields as follows:

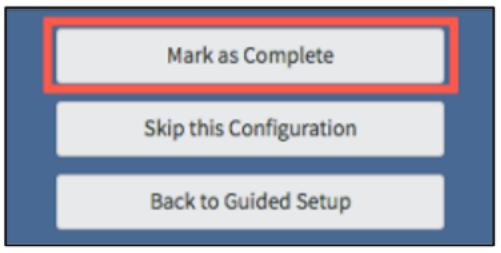
   Page header caption: **Cloud Dimensions <YourFirstName YourLastName>**

   NOTE: *Replace* ***<YourFirstName YourLastName>*** *with your own name (for example, Cloud Dimensions* ***Joe Employee****).*

   Browser tab title: **Cloud Dimensions ServiceNow**
   Banner image for UI16: **[CloudDimensions-CD-Logo.png]**
   Header background color: **#2b3a5a**
   Header divider stripe color: **#d84833**
   Navigation header/footer and navigation background expanded items: **#486a93**
   Navigation selected tab background color: **#ffffff**
   Background for navigator and sidebars: **#536171**
   Currently selected Navigation tab icon color for UI16: **#000000**

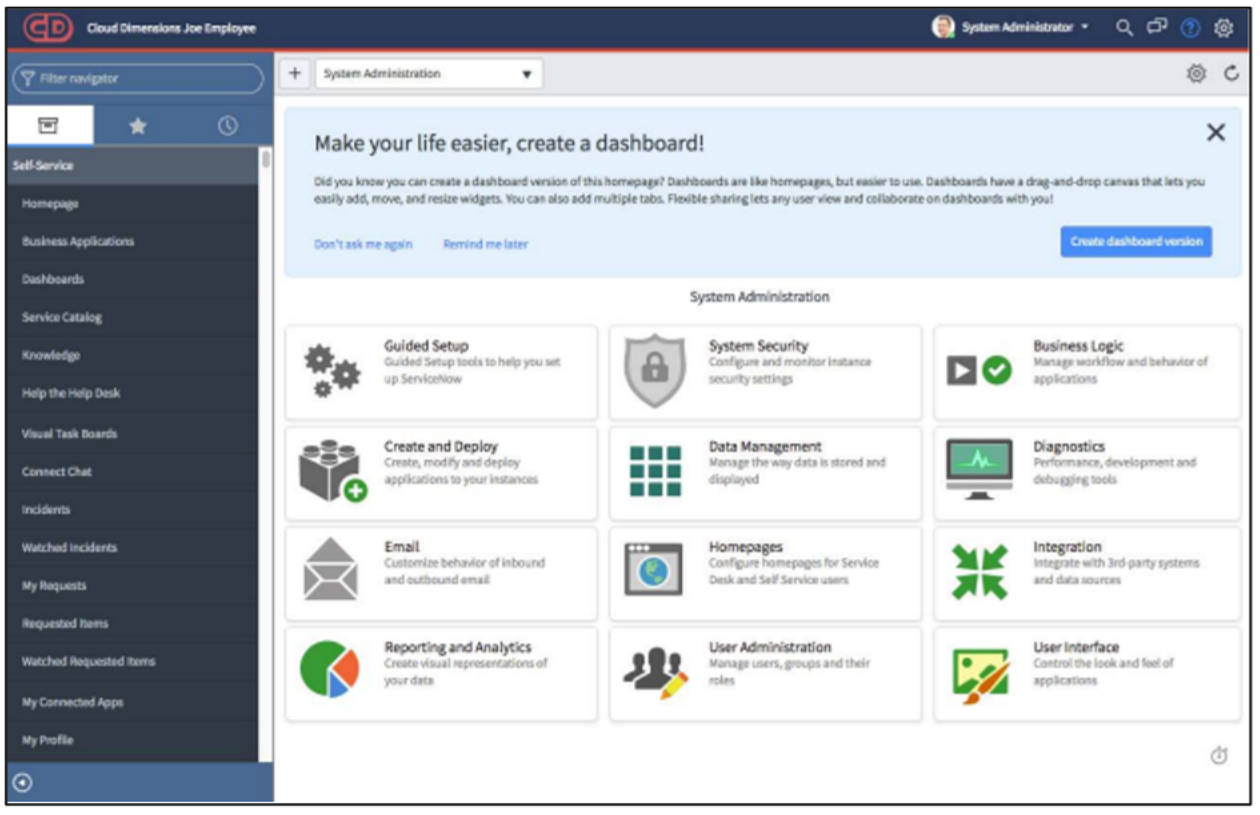
***NOTE**: As you enter values, especially for color fields, changes may display in real time.*

5. When finished, click the **Save** button.
6. Refresh your browser to ensure the changes take full effect.
7. From the **Help** sidebar, click **Mark as Complete** for this Guided Setup task:



LAB VERIFICATION

Cloud Dimensions System Theme

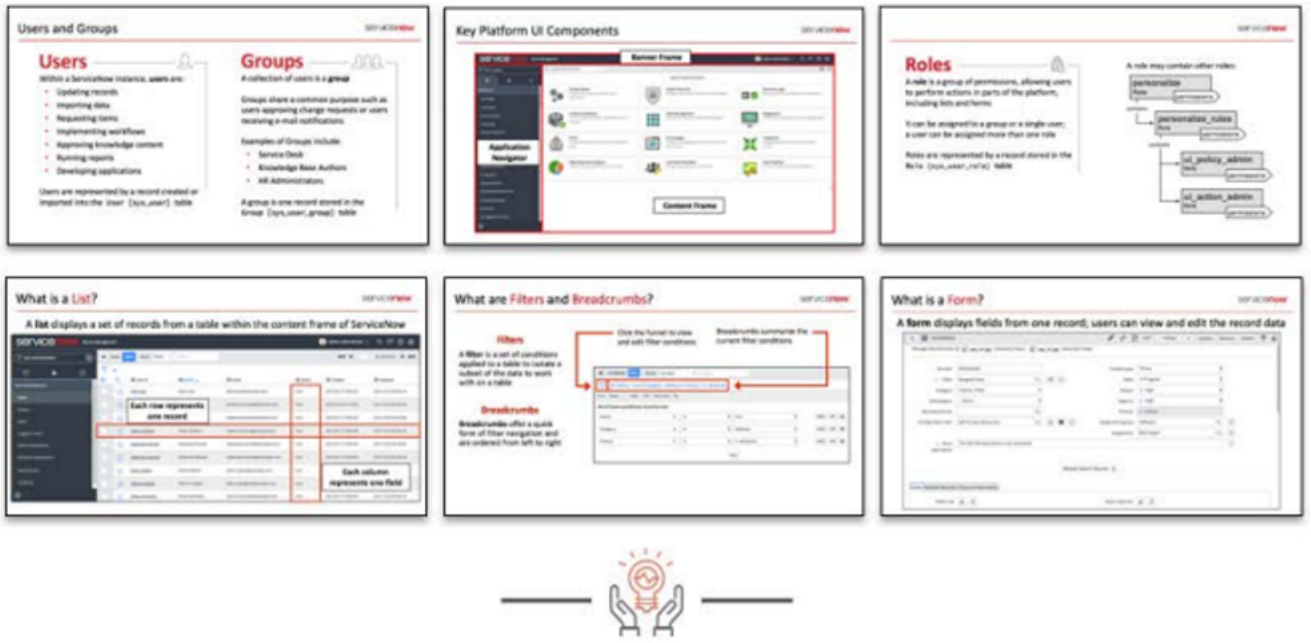


***NOTE**: For demonstration purposes, as well as clarity of these training materials, screen shots in future labs will use the "Cloud Dimensions" system theme as defined in this lab.*

Congratulations, you have completed the Branding lab!

Module 1 Recap

User Interface & Navigation



For these selected topics, discuss:

Why would you use these capabilities?

When would you use these capabilities?

How often would you use these capabilities?

servicenow

1 User Interface & Navigation

2 **Collaboration**

3 Database Administration

4 Service Automation

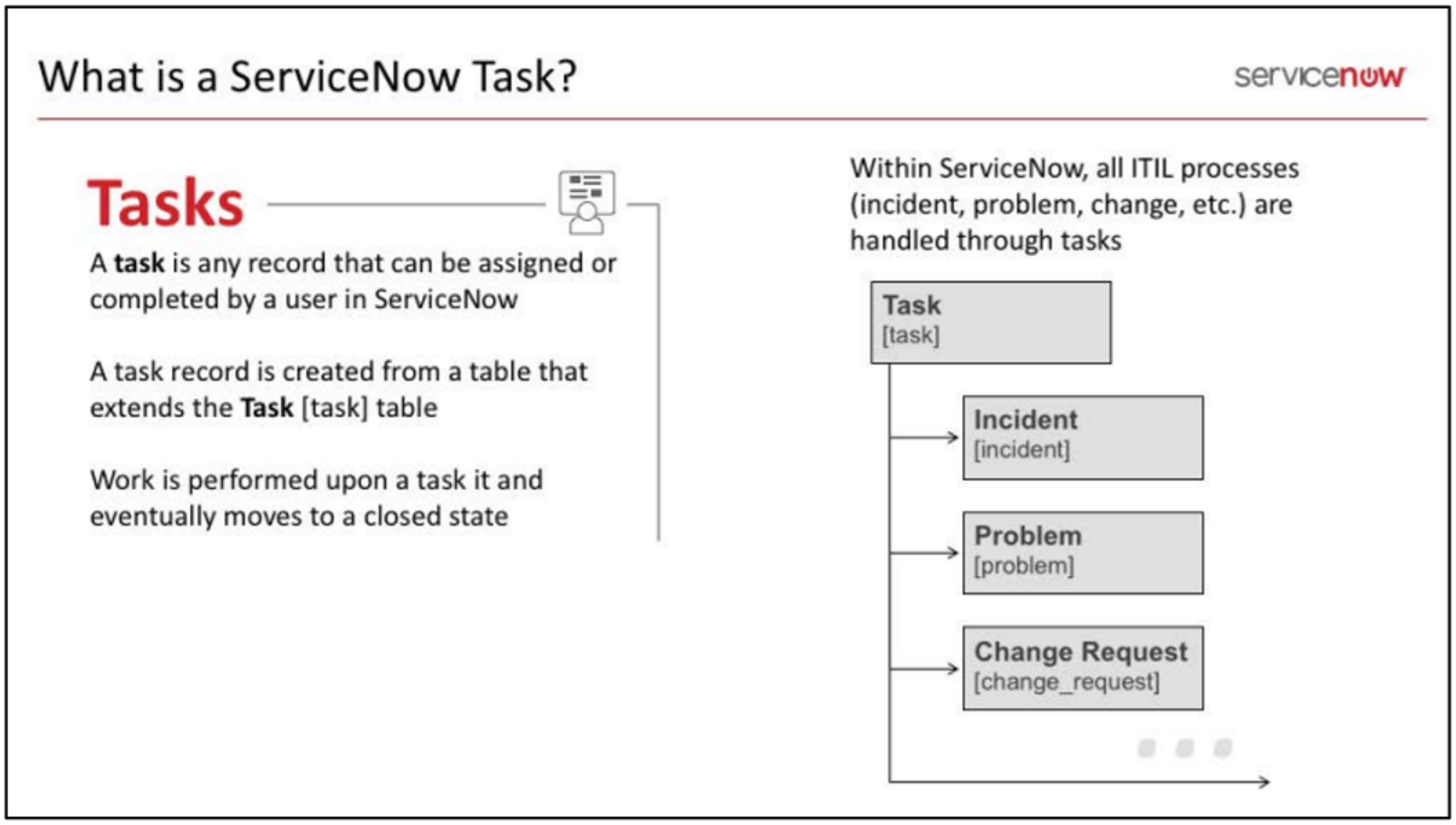
5 Intro to Scripting & Application Tools

Module 2 – 2.1 Task Management

servicenow

Objectives

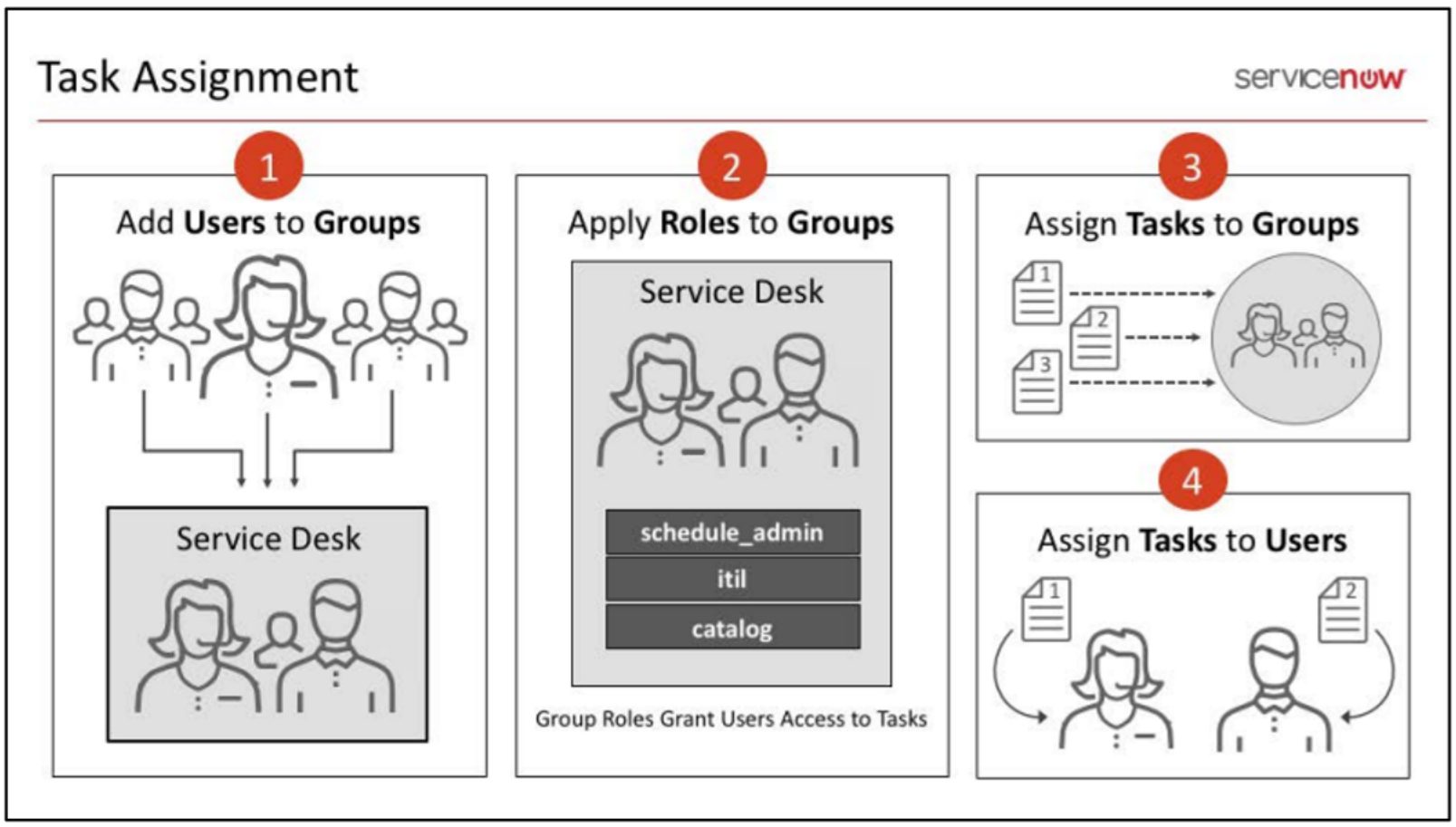
- What is a ServiceNow Task?
- Task Assignment
  - Assignment Rules
  - Agent Intelligence
- Accessing Tasks: My Work / My Groups Work
- Presence
- Notes Tab
- Activity Stream Inline Editing
- Visual Task Boards



Tasks are created by users who are requesting the task to be performed, and are then updated as the task moves along a workflow. Tasks can be assigned to specific users or user groups.

The **Task** [task] table is one of ServiceNow's core tables and provides a series of standard fields used on each of the tables that extend it, such as the Incident [incident] and Problem [problem] tables. In addition, any table which extends Task can take advantage of task-specific functionality for driving tasks, including:

- **Approvals:** Approvals can be generated to a list of Approvers, either manually or automatically, according to Approval Rules. Approvals can be incorporated into workflows or can stand alone.
- **Assignments:** Assignment rules can automatically assign tasks to users or groups, ensuring that tasks are handled by the most appropriate team members.
- **Service Levels:** Service level agreements can track the amount of time that a task has been open, to ensure that tasks are completed within an allotted time.
- **Inactivity Monitors:** Inactivity monitors ensure that tasks do not fall by the wayside by notifying users when tasks have been untouched for a predefined period of time.
- **Workflow:** An administrator can specify a specific workflow process to apply to tasks that meet certain conditions. After a task is created that meets the conditions, the workflow applies an automated process to the task. The process is defined in the graphical workflow editor.



Typically, user administration helps to facilitate task assignment in ServiceNow.

Users can belong to more than one group and groups identify a subset of users based on roles. Users in groups can be assigned permissions to approve, change, or resolve incidents and requests, provide a reference for alerts and notifications, and receive email notifications.

Every user belonging to a group inherits that group's roles, so the preferred method of role management is:

- add users to group
- apply roles onto groups

When removing a user from a group, roles inherited by that group are revoked for that user.

Similarly, a group may contain other groups, where a child group inherits all roles owned by its parent. Users added to child groups gain roles of that child group plus any parent groups.

NOTE: Group names are unique in ServiceNow.

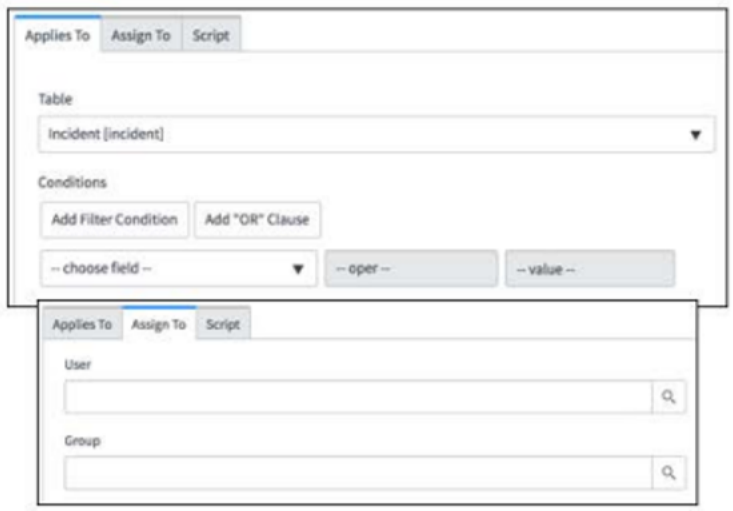
With groups defined, tasks can be assigned to them and then to single users belonging to that group. In other words, a task record can be assigned to an assignment group and an assigned user.

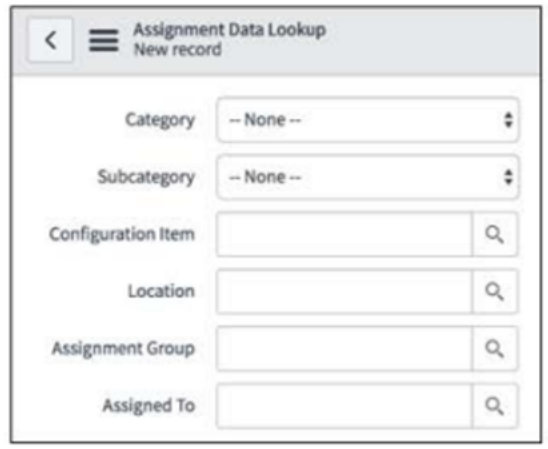
Task Assignment: Assignment Rules

servicenow

Assignment rules can automatically set a value in the **assigned_to** and **assignment_group** fields of a task record

Conditions defined in the assignment rule determine when the rule will trigger and what values it will set





Create an assignment rule by navigating to the **System Policy > Rules > Assignment** module.

An assignment rule must also meet these additional criteria to run:

- The task record has been created or updated. Assignment rules do not apply to unsaved changes on a form.
- The task record must be unassigned. The record cannot have an existing value for either the assigned_to or assignment_group fields. Assignment rules cannot overwrite existing assignments (including assignments set by a default value or a previously run assignment rule).
- The assignment rule is the first rule that matches the table and conditions. If more than one assignment rule matches the conditions, only the rule with the lowest order value runs

Additionally, assignment rules can be scripted, giving even more flexibility on the trigger and outcome.

Assignment data lookup rules can be created by navigating to the **System Policy > Rules > Assignment Lookup Rules** module. These rules only apply to incident records and have less options, compared to the other assignment rules.

Task Assignment: Agent Intelligence

servicenow

Agent Intelligence uses machine-learning algorithms to set field values during record creation

Improve efficiency and quality, and reduce cost by:

- Reducing task resolution times
- Reducing the number of interactions required to resolve tasks
- Reduce the error rates of categorizing and assigning work

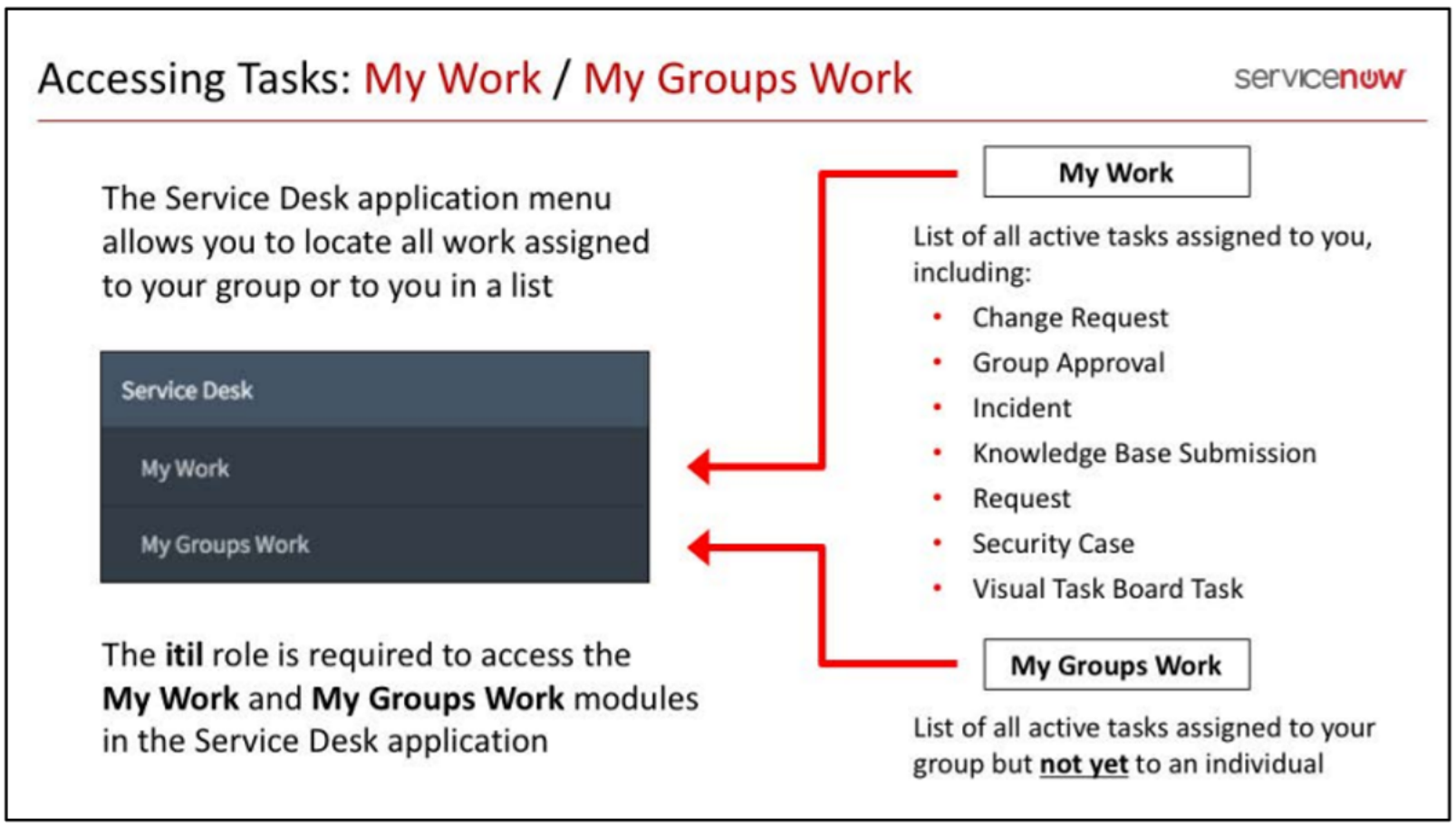
Users with the **admin** or **ml_admin** role will be able to initiate the setup of Agent Intelligence.

This functionality does require multiple plugins – more information about setup and initial configuration can be found here:

https://docs.servicenow.com/bundle/kingston-servicenow-platform/page/administer/agent-intelligence/task/get-started-agent-intelligence.html

Once activated and defined, Agent Intelligence is truly personalized machine learning tailored to your data.

For example, when an incident is created, ServiceNow will automatically assign the right category, priority and assignment group based on the record's short description. A requester no longer has to scroll down multiple lists to choose the most appropriate category, and the incident created gets the right attention and SLA it deserves; thus increasing the overall service level satisfaction



There are various features for managing and closing task records.

You can quickly locate all work assigned to your group or to you specifically in ServiceNow, using the Service Desk application.

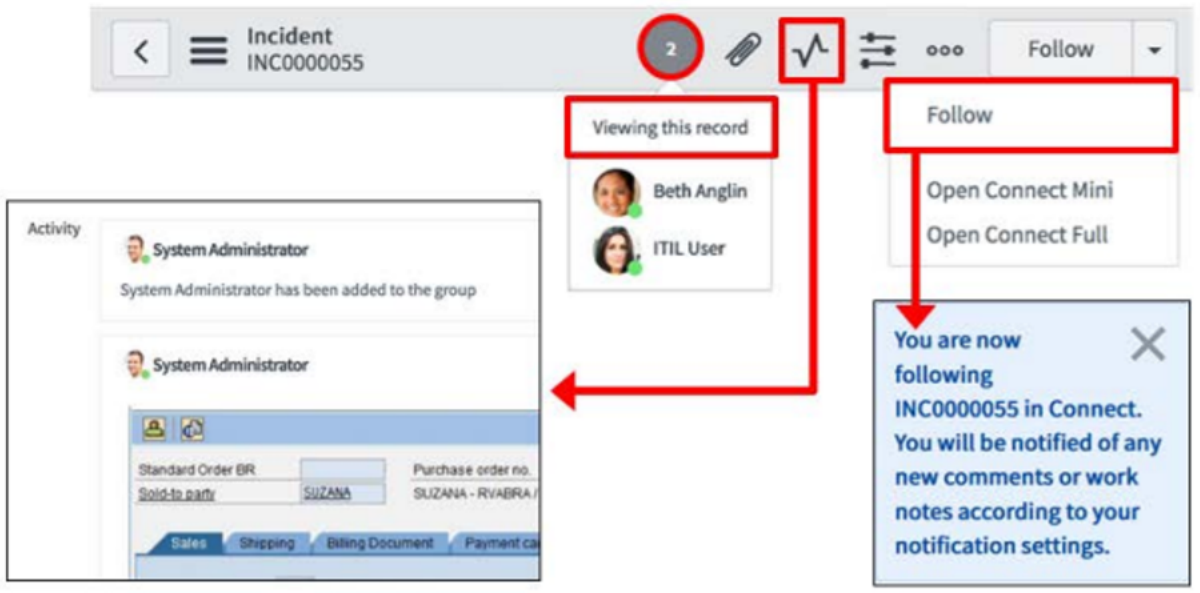
When an active task (which might include work such as incidents, problems, changes, and more) is routed to your group, it can be located under the **Service Desk > My Groups Work** module. From there, a group member or manager may assign the task to an individual within the group.

At that point, locate any active task specifically assigned to you under the **Service Desk > My Work** module.

Presence

servicenow

The **Presence** feature facilitates synchronous collaboration within one record. You can see who is online, view their current status, and what they are viewing or editing, all in real-time



Imagine a scenario in which you have a critical issue, documented in a Priority 1 record. Multiple stakeholders may need to view and update the record simultaneously. The **Presence** feature facilities that collaboration, showing you who is viewing the record, displaying the record activity stream, and even allowing you to customize notifications alerting you to record updates.

The number of active viewers is listed in the form title bar. Click for a list of viewers.

NOTE: If you do not see this icon, you are the only viewer on this record.

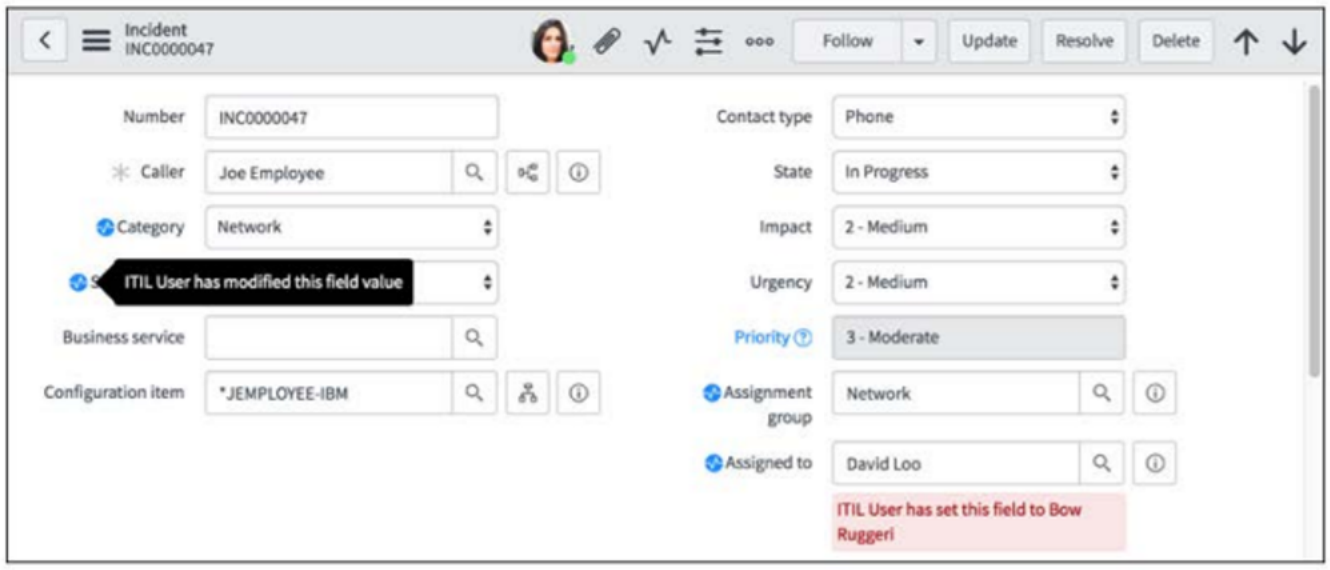
Click the **Show Activity Stream** icon to jump to the record Activity section, which includes the record history and updates by you and other viewers.

Follow the record to receive notifications when the record is updated. Open **Connect** to customize these notification or start a real-time conversation with record viewers and other stakeholders.

Presence: Real-Time Editing

servicenow

Edit records in real-time, and see edits saved by other users (just like Google Docs) which improves collaborative efforts



Real-time editing is an extension of Presence. It allows you to work with others on the same record, indicating their state (editing or viewing) as well as what their edits are (shown through the blue "pulse" icons in this screen capture).

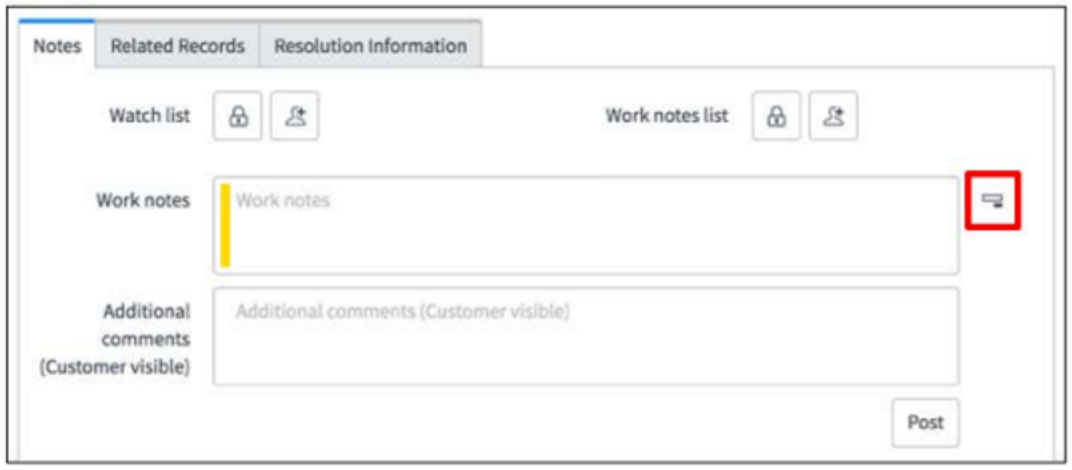
Connect enables you to work seamlessly with others using entirely different interfaces or devices to perform tasks. You can use Connect on Visual Task Boards (VTBs), Connect Chat, Apple Watch, and more.

Connect is about working with others in real-time; reducing record resolution from days to mere minutes.

Notes Tab: Work Notes and Additional Comments

servicenow

Use the **Notes** tab in a record to communicate to stakeholders and document task activities throughout the lifecycle



The **Show all journal fields** icon ( ) allows you to display multiple fields under the Notes tab, including **Work notes** and **Additional comments**

Once enabled, you can then click the **Show one journal field** icon ( ) to only display the **Work notes** field

The **Notes** tab allows you to document task activities throughout its lifecycle for both an external and internal audience. Depending on the task record type, additionally fields may be available to accomplish similar outcomes including, but not limited to, the **Additional comments** field. In the example seen here, an incident record's **Notes** tab is displayed.

1. The **Work notes** field provides a log to document all the technical and behind-the-scenes work on a task. Upon saving, Work notes are stored in the record Activity section, where they can be viewed and added upon by users with permissions to view the record. Fully documenting Work notes is beneficial for Knowledge Management and critical for continuity in the task management process. Work notes are only visible to Fulfillers and are not available to external users or customers.

2. Use the **Additional comments (Customer visible)** field to communicate back and forth with the requester and other stakeholders directly in ServiceNow. For example, you may want to keep the customer apprised of progress on their record or request additional information. Upon saving, the Additional comments (including the updated information and comments history) are emailed directly to the requester. When the requester receives an email notification containing Additional comments, they can respond directly to the email and their feedback will be documented in the **Activity** log of the record, along with your Additional comments.

   NOTE: When responding to an email from ServiceNow, do not change the Subject as it may not be saved to the correct record.

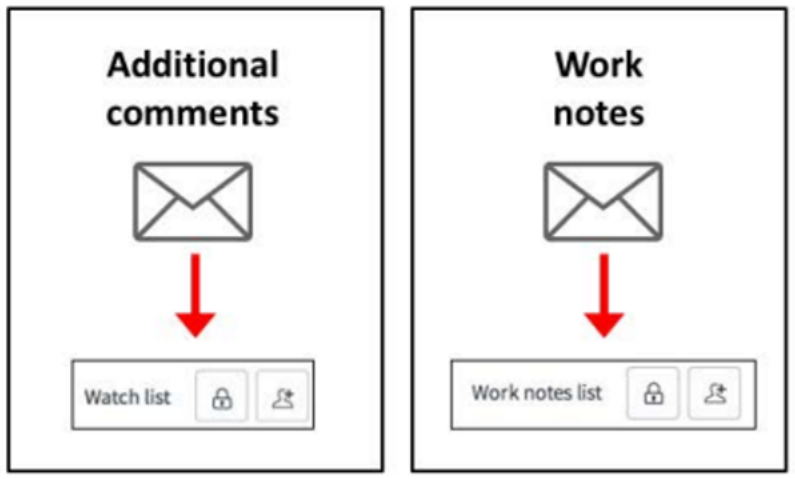
Notes Tab: Add Stakeholders

servicenow

Send users additional notifications with the **Watch list** and **Work notes list** fields

Users added to the **Watch list** will receive anything entered as **Additional comments (Customer visible)**

Users added to the **Work notes list** will receive anything entered as **Work notes**



NOTE: One or both of these fields will appear on the record, under the Notes tab, depending on the record type. Incident records contain both the **Watch list** and **Work notes list** for use.

Notes Tab: Activity

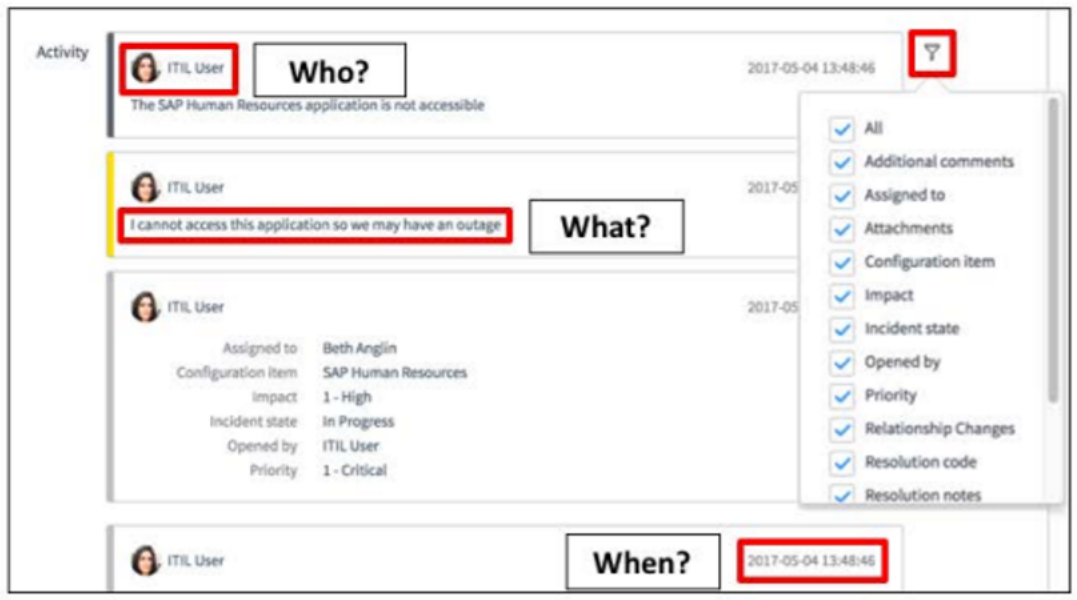
servicenow

The **Activity** section located under the Notes tab provides a complete history of a record

It details:

- Who made an update?
- What was the update?
- When was the update made?

Clicking the filter icon (▽) allows activity information to be filtered



From creation through to closure, the entire history of the Incident record is automatically tracked and recorded in the Incident **Activity** section, located in the **Notes** tab.

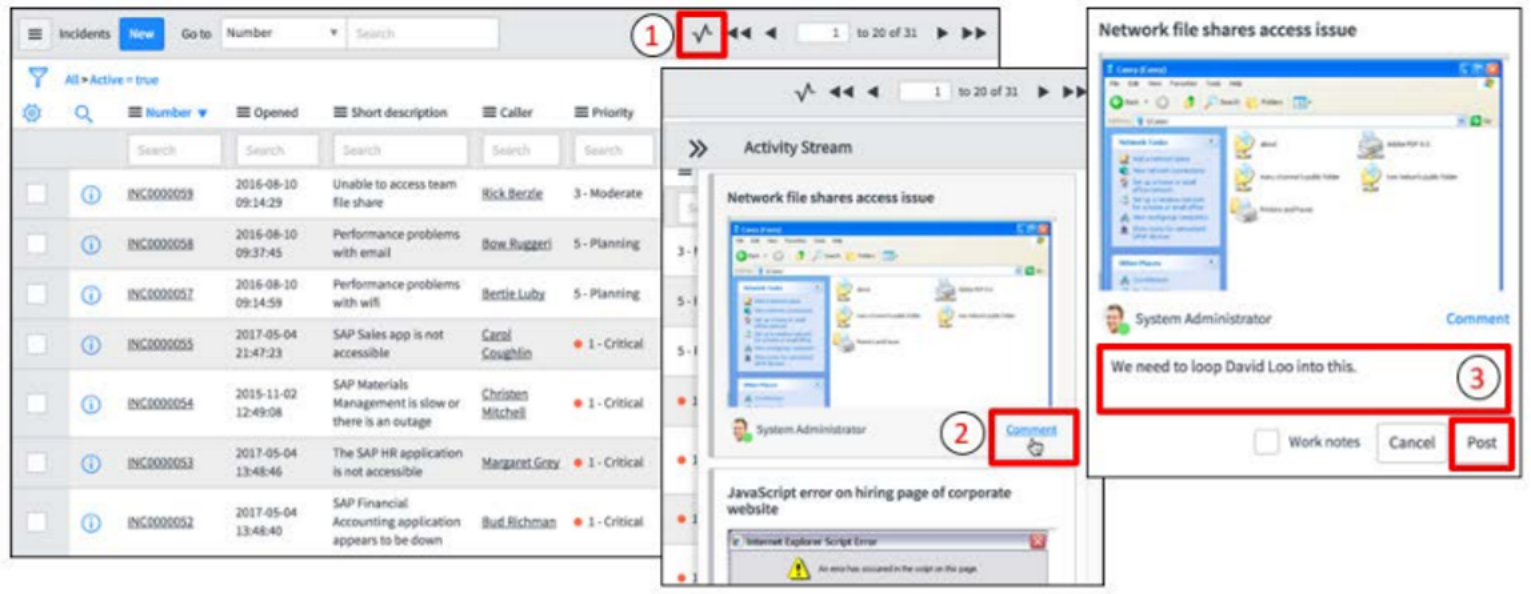
The Activity section, which is read-only, documents when a change was made and by whom. These changes include assignment and reassignment, Additional comments and Work notes, updated field values, State changes, and more.

The funnel icon in the top-right of the Activity section allows you to filter your view to see only your desired categories of information.

Presence: Activity Stream Inline Editing

servicenow

The **activity stream inline editor** enables users to contribution to actual work within a record without opening a form interface



Just like real-time editing on a form, inline commenting on the activity stream means you can annotate active records as updates are made, allowing multiplied efforts across several pieces of work simultaneously.

To do so, navigate to a list of active task records, then:

1. Click **Show activity stream in a flyout window** from the list header
2. With the window open, scroll down to browse the records recently updated and hover over an update you wish to comment, then click **Comment**
3. Enter your comment into the text field, then click the **Post** button

A benefit of activity stream inline editing is that you are able to update multiple active records without having to open a single record.

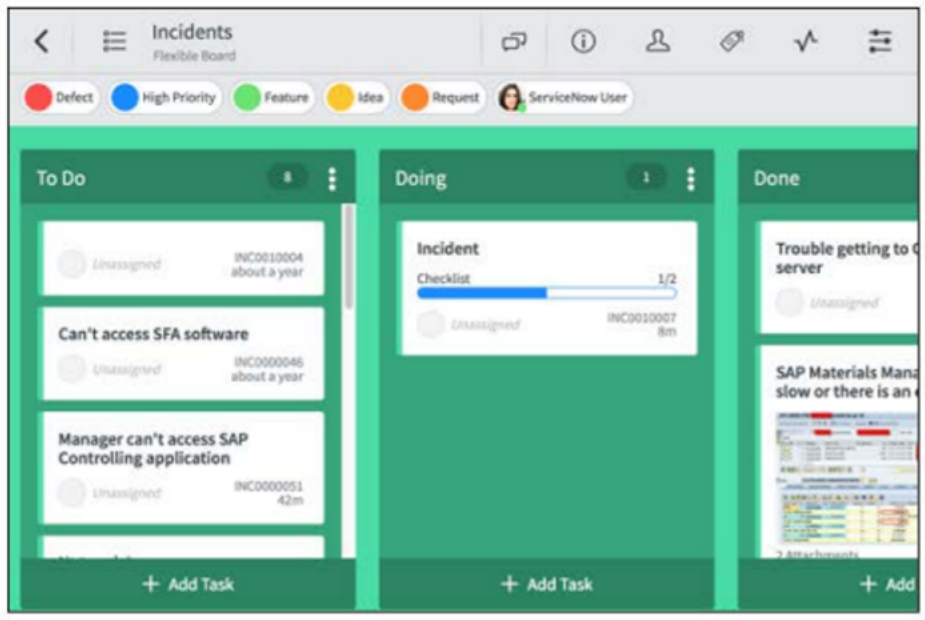
Visual Task Boards

servicenow

Transform your lists and forms into an interactive graphical experience using **Visual Task Boards** (VTB)

Visual Task Boards allow you to:

- Manage your tasks through a visual, drag-and-drop interface
- Identify process bottlenecks at a glance, in real-time
- Track embedded activity screens to view updates all in one place

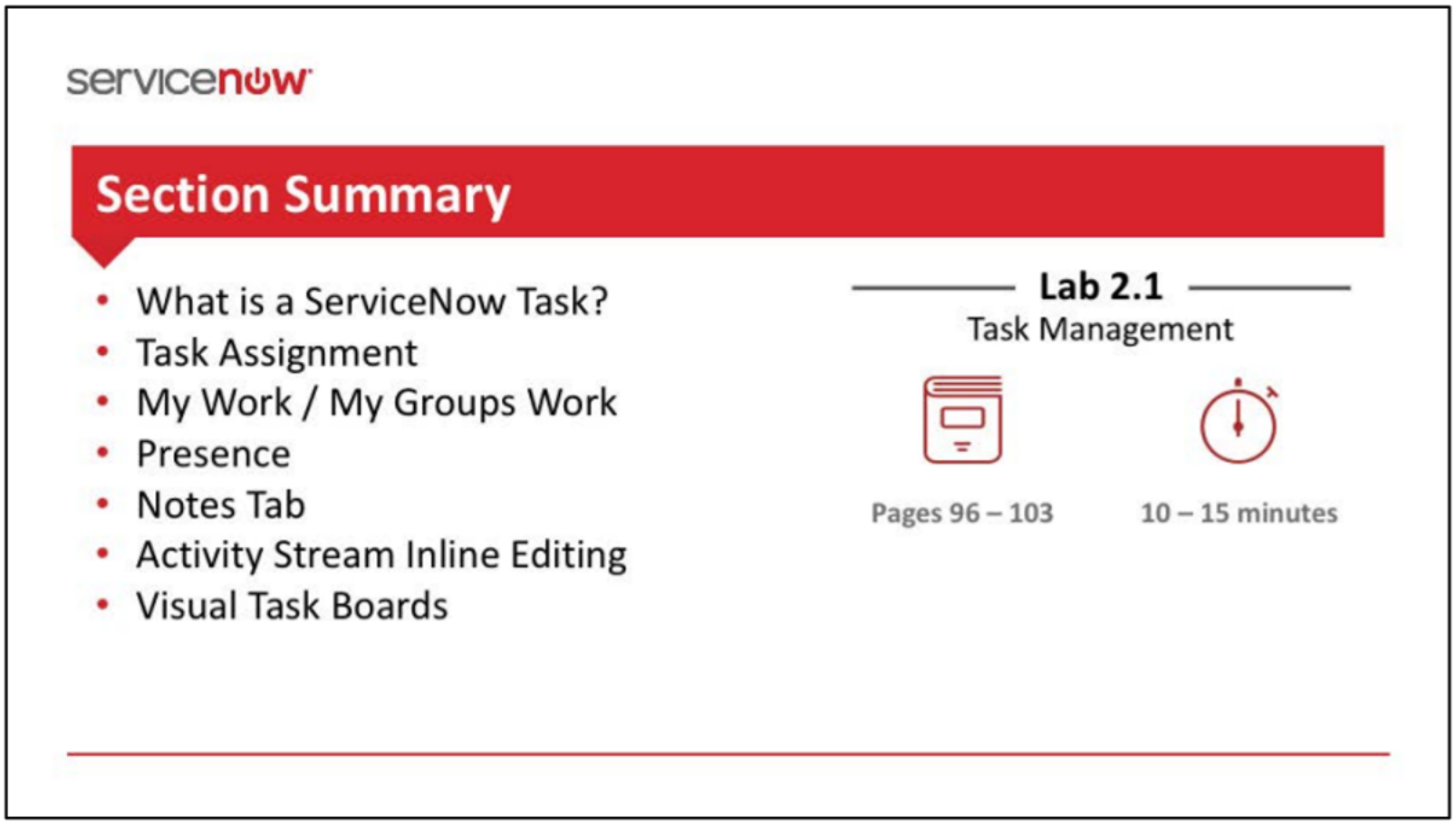


Use **Visual Task Boards (VTB)** to create your personal to-do list, collaborate in real-time with group members on assignments, and more. Displayed graphically as lanes and cards, VTBs provide a landing page to view and organize all your work in ServiceNow.

There are three types of VTBs:

- **Freeform**: Use Freeform boards as your personal organizer, creating individual tasks of any kind and freely adding, removing, and modifying cards and lanes
- **Guided boards**: Alternatively, use a Guided board, which is created from a list and uses a field value (e.g. Incident States) as lanes. Records in that list, which appear as cards, are actually modified when you edit cards or change lanes in a Guided board.
- **Flexible**: Flexible boards are also created from a list but lane changes do not update underlying task data

To get started with VTB, navigate to **Self-Service > Visual Task Boards** and follow the instructions for creating a board.



Lab 2.1 – Task Management:

- Browse the Service Portal
- Submit an incident from the Service Portal
- Use work notes, comments, and chat to cooperatively manage an incident

LAB

Task Management

2.1

10 – 15 minutes

Lab Goal

This lab will show you how to do the following:

- Browse the Service Portal
- Submit an incident from the Service Portal
- Use work notes, comments, and chat to cooperatively manage an incident

In addition to internal testing, Cloud Dimensions has begun allowing major partners to test Infinity devices.

Before submitting an incident to report a problem, these partner users have been instructed to first browse the Knowledge Base in Cloud Dimensions Service Portal.

If the user is unable to find a solution to their problem, submitting an incident will receive attention by a support agent in the Infinity Customer Support group.

The Infinity Customer Support group has defined processes for incident assignment and resolution, which uses assignment rules and connect chat to ensure the best help is offered to an end-user.

A. Browse the Service Portal

Start this lab by logging into the Cloud Dimensions Service Portal as a partner user experiencing problems during Infinity testing.

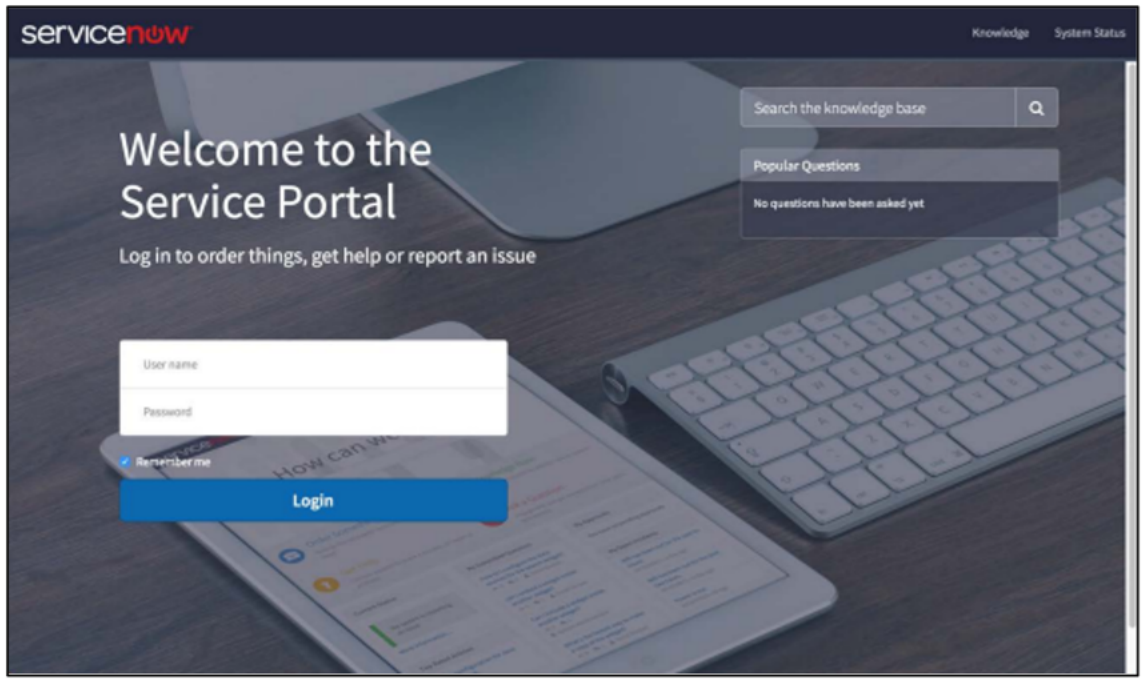
TIP: Open up another browser to complete these steps.

1. If you are not using a different browser, log out of your instance.
2. Add **/sp** to the end of the URL:

   https://instance-###.lab.service-now.com**/sp**

3. Press **Enter** on your keyboard.

You should be brought to the Service Portal:

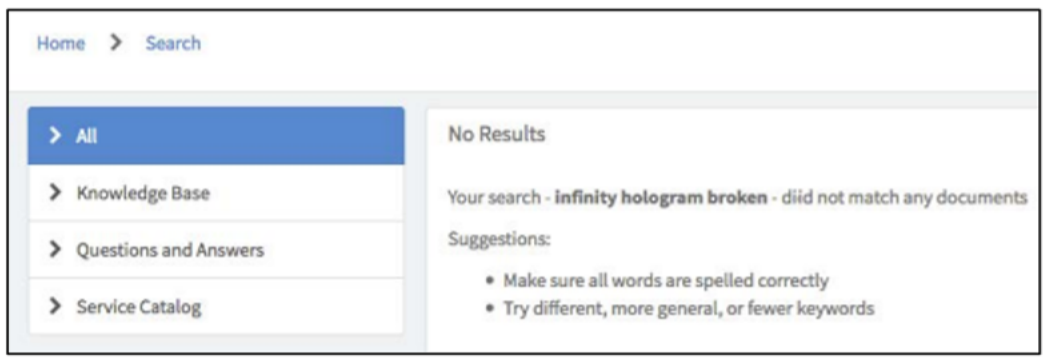


4. Log in using the following user credentials:

   User name: **jon.floyd**
   Password: **floydpass**

5. Type **infinity holograph broken** into the **How can we help?** search bar on the home page; then press enter on your keyboard.

   No Results will display:



***NOTE**: Typing either **infinity** or **Infinity** will result in **No Results**.*

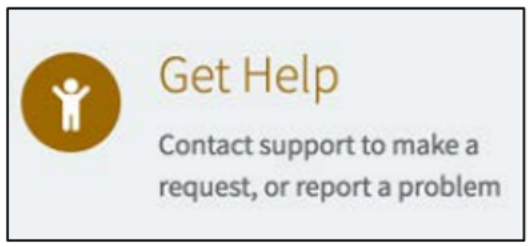
At this time, we will assume Jon Floyd continued browsing the Knowledge Base, but was unable to discover a solution for the issue he is experiencing.

As established by partner procedure, he will now submit an incident to receive help from Cloud Dimensions.

6. Click the **ServiceNow logo** in the top-left corner to return to the homepage.

Submit an Incident

1. Select **Get Help**:



2. Scroll down and select **Create Incident**.

3. Fill out the form as follows:

   Urgency: **3 – Low**
   Please describe your issue below: **The Infinity is having trouble displaying clearly.**

4. Click **Submit**.

   You may briefly see a message on the top of the screen indicating your incident was created and your profile badge should show one Request.

5. Log out of the instance as **Jon Floyd.**

***NOTE**: If you opted to use another browser for these steps, return to the original session and move to step 1 of the **Update the Incident** section below. Otherwise, continue on to the next step.*

---

6. From your instance URL, remove the **/sp** suffix (including everything that follows).

7. Press **Enter** on your keyboard to return to the normal login screen.

B. Update the Incident

1. Log into the instance as the **System Administrator**.

***NOTE:** If you experience issues with the interface caching, refresh the browser and click on the ServiceNow logo.*

---

2. Impersonate **Rita Center**.

NOTE: *Rita is a Customer Support Agent on the Infinity Customer Support group specializing in software-related issues.*

3. **Service Desk > My Groups Work**.

NOTE*: The incident was automatically assigned to Rita's group (**Infinity Customer Support**) by an assignment rule created for the purpose of this exercise.*

4. Open the incident record and update it as follows:

   State: **In Progress**
   Assigned to: **Rita Center**

5. **Save** the record, staying on the form.

6. Assume Rita has remotely accessed Jon's device and found no apparent issues.

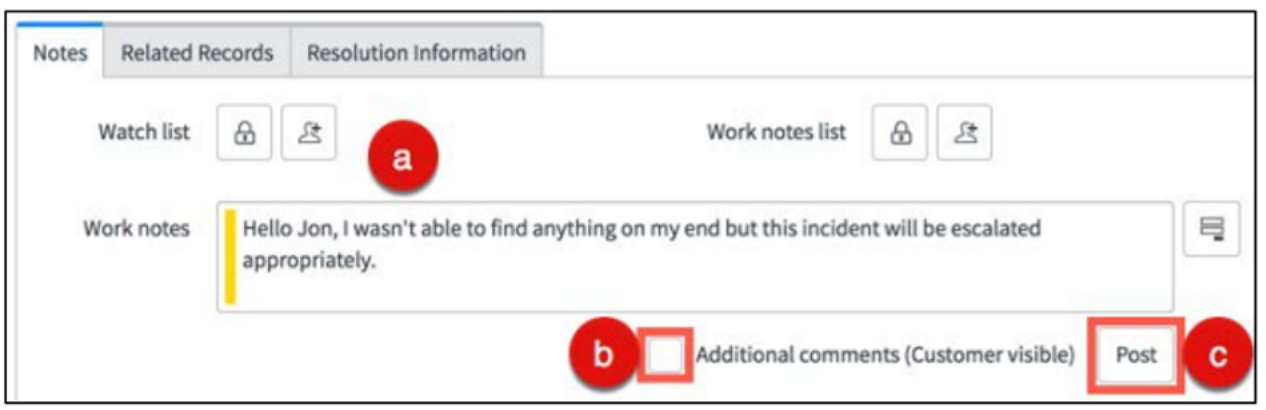
7. Update the incident record to inform Jon that it must be escalated to another team:

   a) From the Notes tab, type into the **Work notes** field: **Hello Jon, I wasn't able to find anything on my end but this incident will be escalated appropriately**.

   b) Check the **Additional comments (Customer visible)** checkbox

NOTE*: Confirm the **Work notes** field label changed to **Additional comments (Customer visible** before posting.*

   c) Click the **Post** button



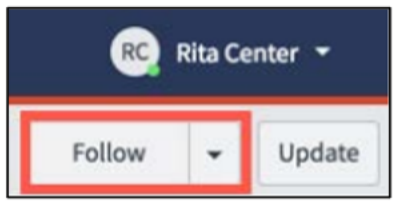
8. The comment now appears under the **Activity** section:

9. Update the incident as follows:

   Category: **Hardware**
   Assigned to: **Trey Tout**

10. **Save**.

11. From the form header, click the **Follow** button:



    Rita will now receive notices when any new comments or work notes are added. Additionally, Rita can now use chat to facilitate the resolution of this incident with peers.

12. Open the **Connect Sidebar** to locate the conversation around the incident:

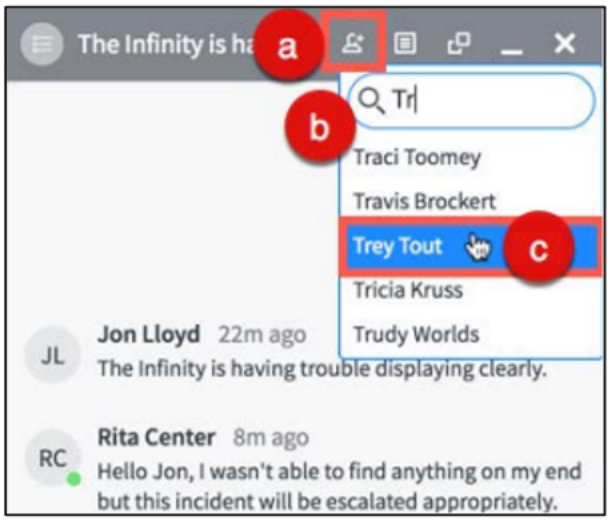
13. Select the “Infinity is having trouble” conversation from the Connect Sidebar to open a chat window.

14. Add **Trey Tout** to the conversation:

    a) Click the **Add User** icon

    b) Search for **Trey Tout**

    c) **Select his name**



15. With Trey added to the conversation, type the following message into the **Worknote** text field at the bottom of the chat window:

    Hello Trey, I thought you could help with this as there are no software issues detected. Thanks!

16. Press **Enter** on your keyboard to send the message.

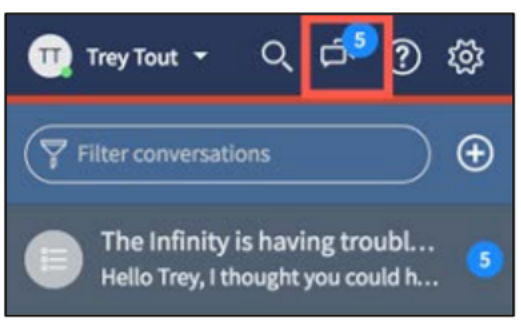
C. Update the Incident

1. Impersonate **Trey Tout**.

NOTE: *Trey is an Engineer on the Infinity Customer Support group. If you are unable to find his name while impersonating Rita, switch back to System Administrator before impersonating again.*

---

2. Open the **Connect Sidebar** to view the conversation from Rita:



3. **Service Desk > My Work**.

4. Open the incident.

NOTE: *You could also access the incident directly from the chat window by clicking on the* ***View Document*** *icon on the chat window header:*

---

5. If open, collapse the chat window and close the Connect Sidebar.

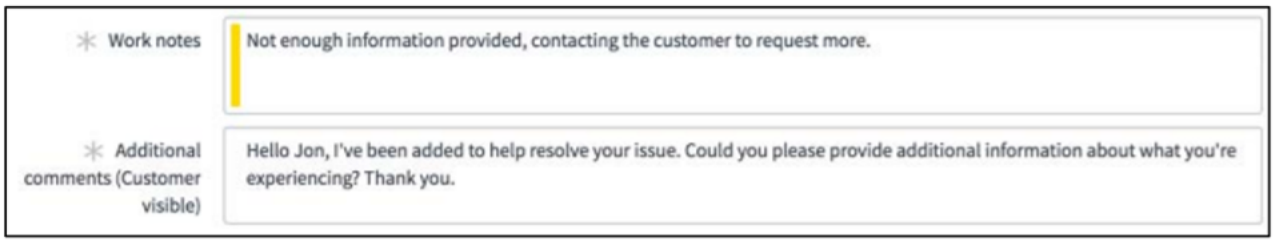
6. With the record open, under the **Notes** tab, click the **Show all journal fields** icon to the right of **Work notes**:

7. Update the record as follows:

   Work notes: **Not enough information provided, contacting the customer to request more.**

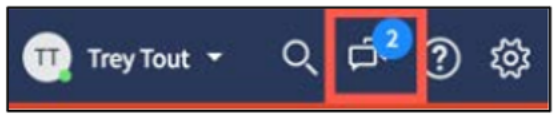
   Additional comments (Customer visible): **Hello Jon, I've been added to help resolve your issue. Could you please provide additional information about what you're experiencing? Thank you.**

   Your fields should look like this:



8. Click the **Post** button to add these comments to the incident.

9. Notice the Connect Sidebar icon has changed, indicating two new messages have been received:



10. Set the record's **State** to **On Hold**.

11. Select **Awaiting Caller** as the **On hold reason**.

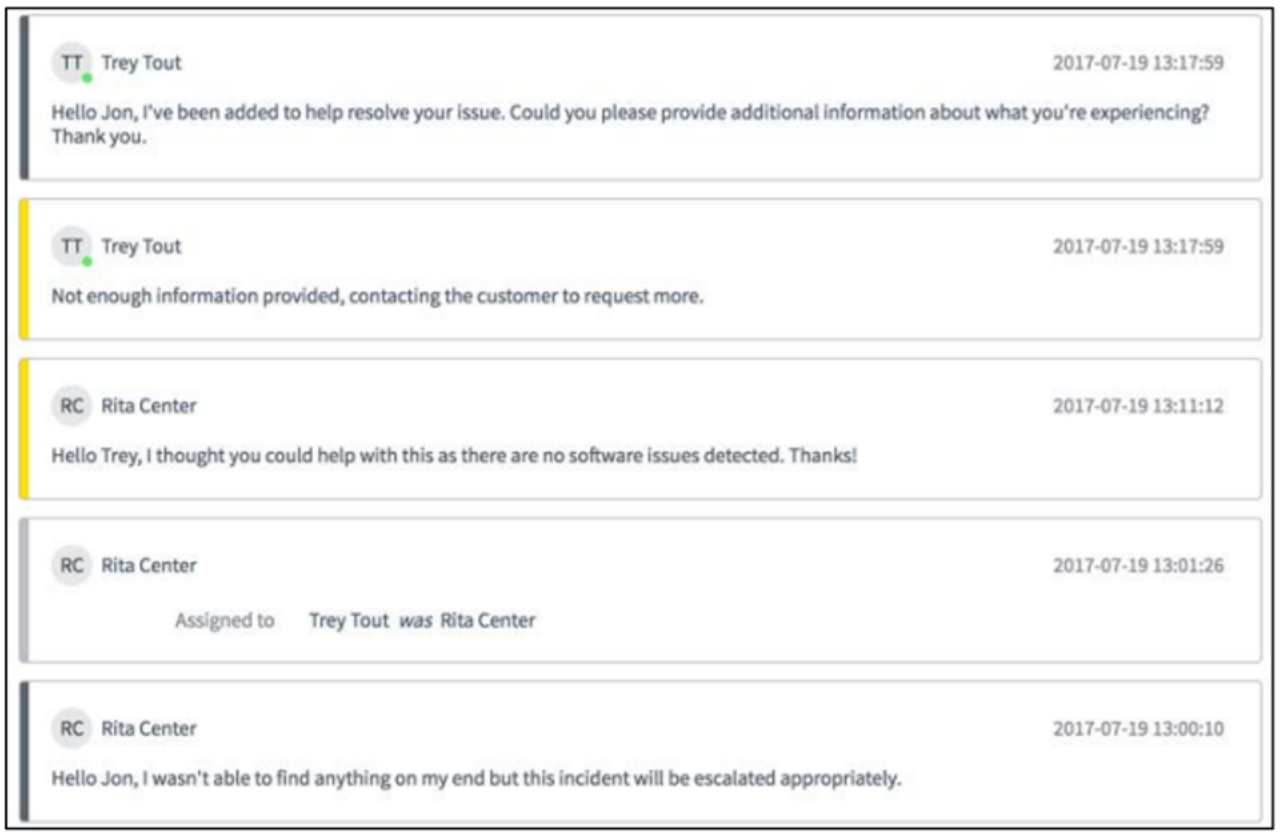
12. **Save** the record.

NOTE: *Although the incident was not resolved at this time, Jon will be able to respond to Trey's inquiry and provide more information to identify a solution.*

LAB VERIFICATION

Updates to an Incident

Review the updates made to the Activity section, recognizing internal (work notes) and external (additional comments) communication:



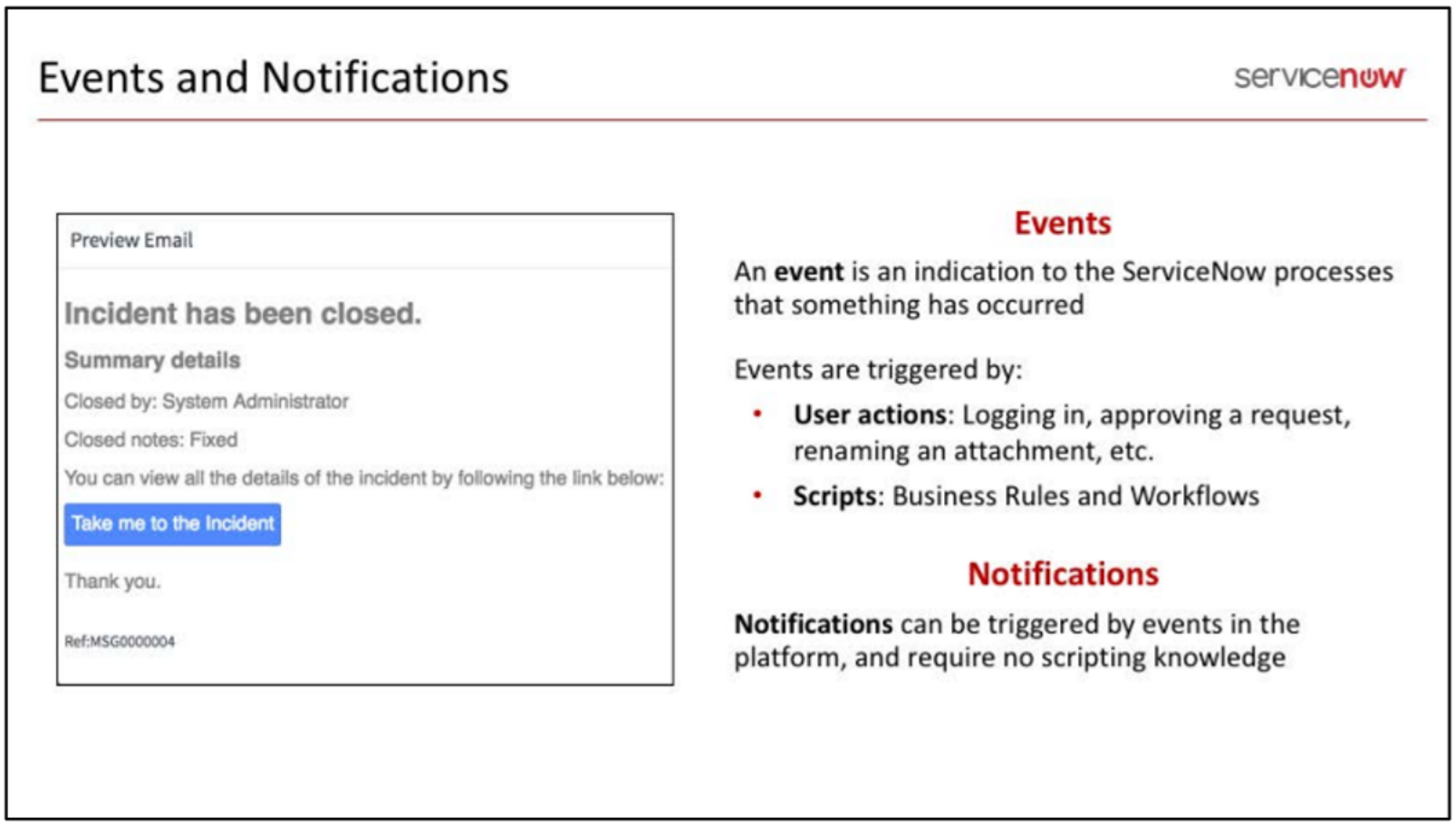
Congratulations, you have completed the Task Management lab!

Module 2 – 2.2 Communication

servicenow

Objectives

- What is an Event?
- What is a Notification?
- Creating Notifications
- Dot-Walking
- Subscriptions
- Connect Chat and Workspace



All baseline events have built in logic to respond when an event occurs. Possible responses include making a change to a record in the database, creating a new record, sending a notification, or logging a message. The event definitions are in the Event Registry [**sysevent_register**] table. The Event Log displays records from the Event [**sysevent**] table. To see a log of every generated event navigate to **System Policy > Events > Event Log**. By convention, events are named using the syntax <table name>.<unique event name>. For example, **incident.updated**, or **problem.closed**.

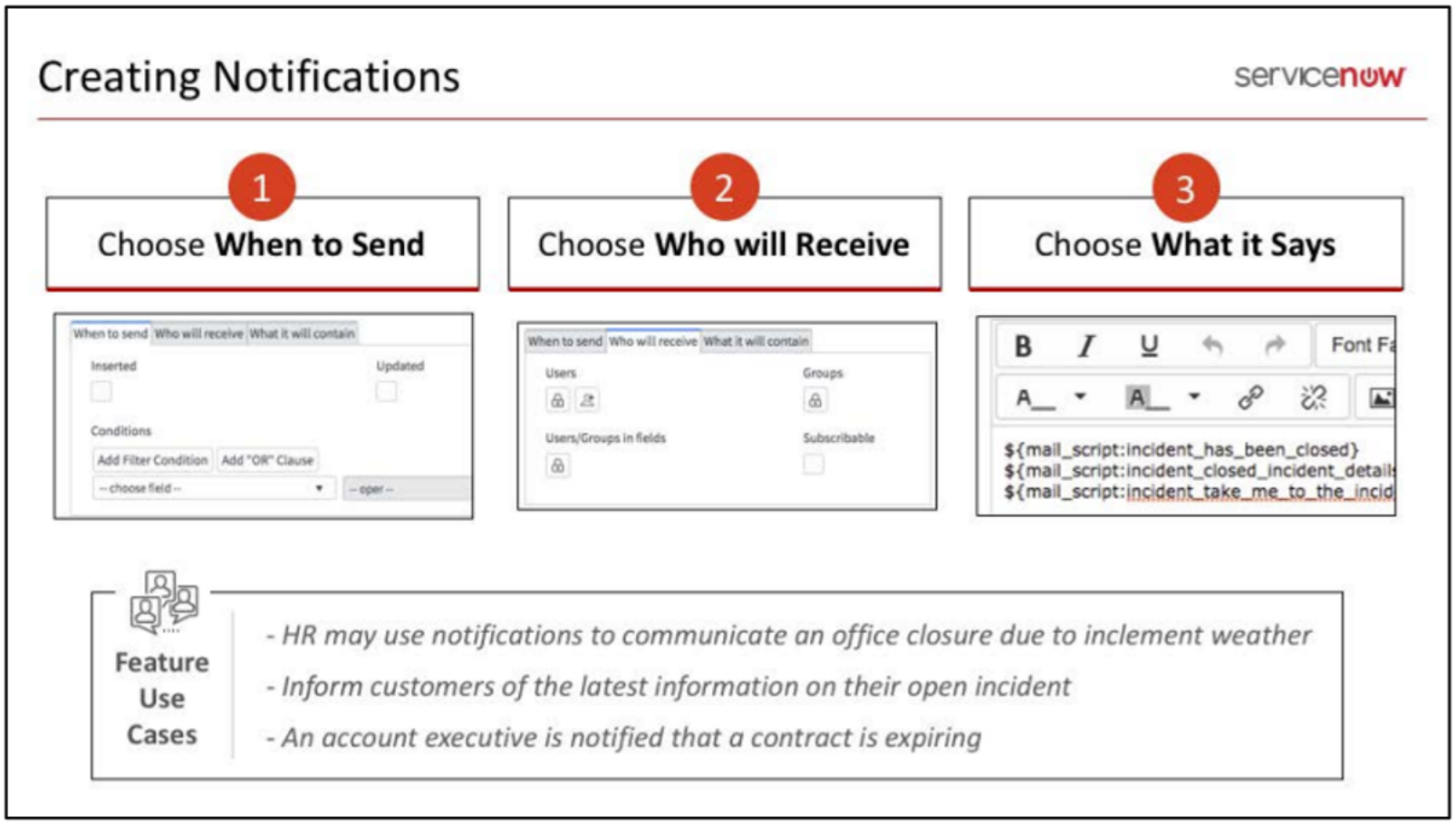
NOTE: Use email notifications to send selected users email about specific activities in the platform, such as updates to incidents or change requests.

A Notification is a tool for alerting users that events that concern them have occurred, including the following methods:

- Email
- SMS
- Meeting Invitation

Notifications are received by configured users and voluntary recipients.

The platform enables you to send Notifications to email addresses of your choice and to any SMS capable device (such as a mobile phone).



Use Notifications to notify users about specific activities in ServiceNow, such as updates to incidents or change requests. Notifications allow administrators to specify:

- When to send the Notification
- Who receives the Notification
- What content is in the Notification

Notifications can be sent when a record is **Inserted** or **Updated** (or both) into the **Table** specified above, only if the specified **Conditions** are met.

Notifications can be sent to specific **Users** and **Groups**. If you address the Notification to a user with an inactive record in the User [**sys_user**] table, the system does not send the notification to that user.

TIP: Consider limiting the recipient list of any notification to 1000 users. By default, if a Notification has more than 100 intended recipients, the system creates multiple notification messages with up to 100 recipients each. If you want to change the recipient limit, set the system property glide.email.smtp.max_recipients.

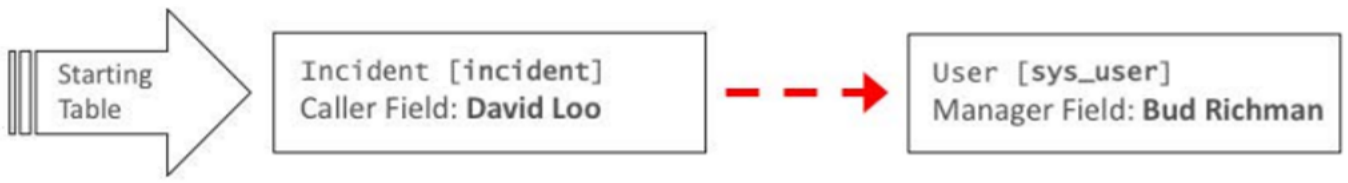
If using an **Email Template** then **Subject** and **Message** will be used from the template unless overridden with a **Subject** and **Message** on this form.

Notifications: Dot-Walking

servicenow

Dot-Walking can get information from a series of referenced fields from different tables, including **Users** and **Groups**

1. From **Users/Groups in fields**, select the reference field you will dot-walk through
2. Click the **Expand Item** icon (+)
3. Select the field which holds the value to be referenced
4. Click the **Add Item** icon (>)



Who Will Receive Notifications?

Users and **Groups** - You might send Notifications of an incident update to the person who opened the incident and the person to whom the incident was assigned by including the **Opened by** and **Assigned to** fields. You may also want to send Notifications to the group assigned to the incident by including the **Assignment group** field.

If the user or group field you want to reference is not on the current table (e.g., **Manager**), add the field by dot-walking to the table it is found on.

To achieve this, from the Notification **Who will receive** tab, click the **Unlock** icon for **Users/Groups in fields** and find the reference field under the Available column. Then click the **Expand Item** (+) icon to dot-walk through this field onto the table it is referencing. In doing so, all fields on this referenced table will be accessible to the table it is found on via a reference field.

In the example illustrated above, the **Caller** field on an incident is **David Loo**. By using dot-walking, this Notification can be sent to the caller (David Loo) of the incident, as well as the caller's manager (Bud Richman). These values are automatically populated because of fields selected – not hardcoded individuals names. In other words, this Notification will automatically identify the caller and the caller's manager (if applicable) based on the information provided in the incident.

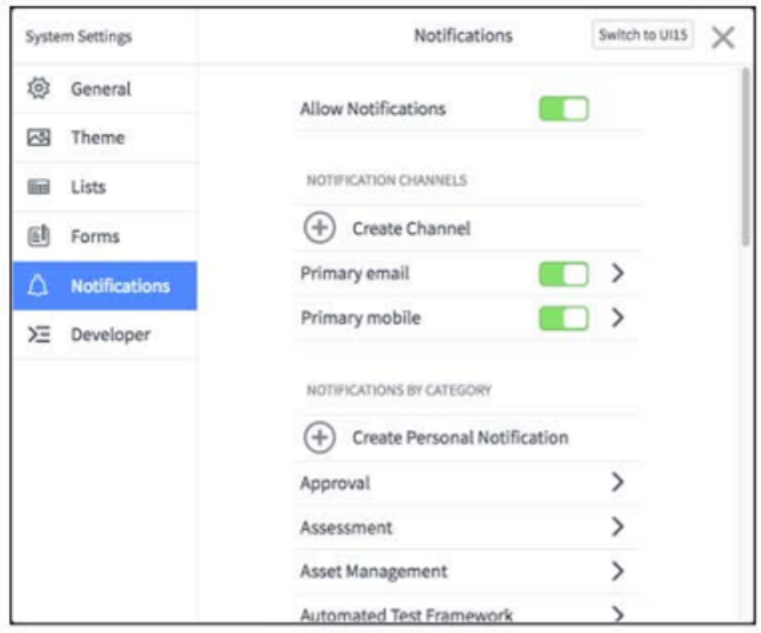
Notifications: Subscriptions

servicenow

Subscriptions

Subscriptions allow users to be informed of various activity occurring in the platform, whether it directly relates to them or not

In addition to email notifications, Short Message Service (SMS) is supported, also known as text messages on telephones



The Notifications page of the Settings is where users can define Notification channels (methods of receiving Notifications), as well as manage their subscriptions to system Notifications.

SMS (Short Messaging Service) is the standard protocol used to deliver short text messages to mobile phones.

Most mobile phones support SMS, even if they do not support more sophisticated messaging, like email.

Alerts and Notifications can be sent to SMS devices as well as standard email notifications.

Notifications to SMS devices are particularly useful when critical events require immediate attention, and waiting for an email notification to be accessed and viewed is too slow.

Connect Chat and Workspace

servicenow

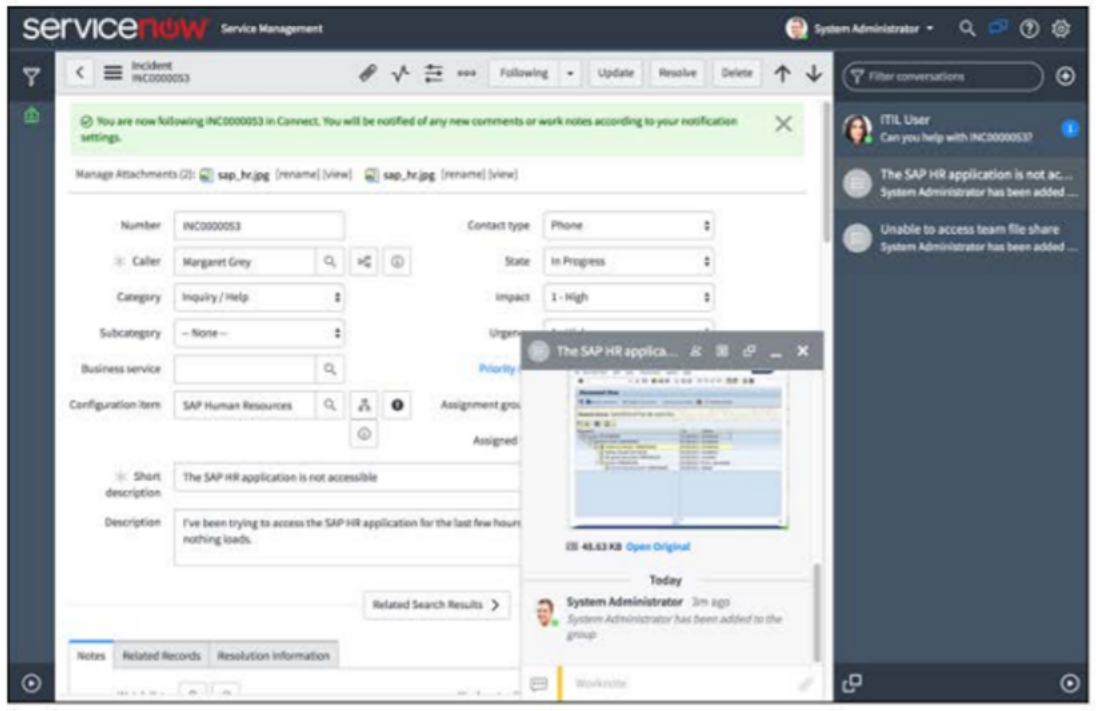
Chat

Connect Chat is a messaging tool that lets you work with others in real-time

Workspace

The **Connect Workspace** provides a full-screen view of all your Connect Chat conversations in one place

Click the **Open Connect standard interface** icon ( ) to open the Workspace

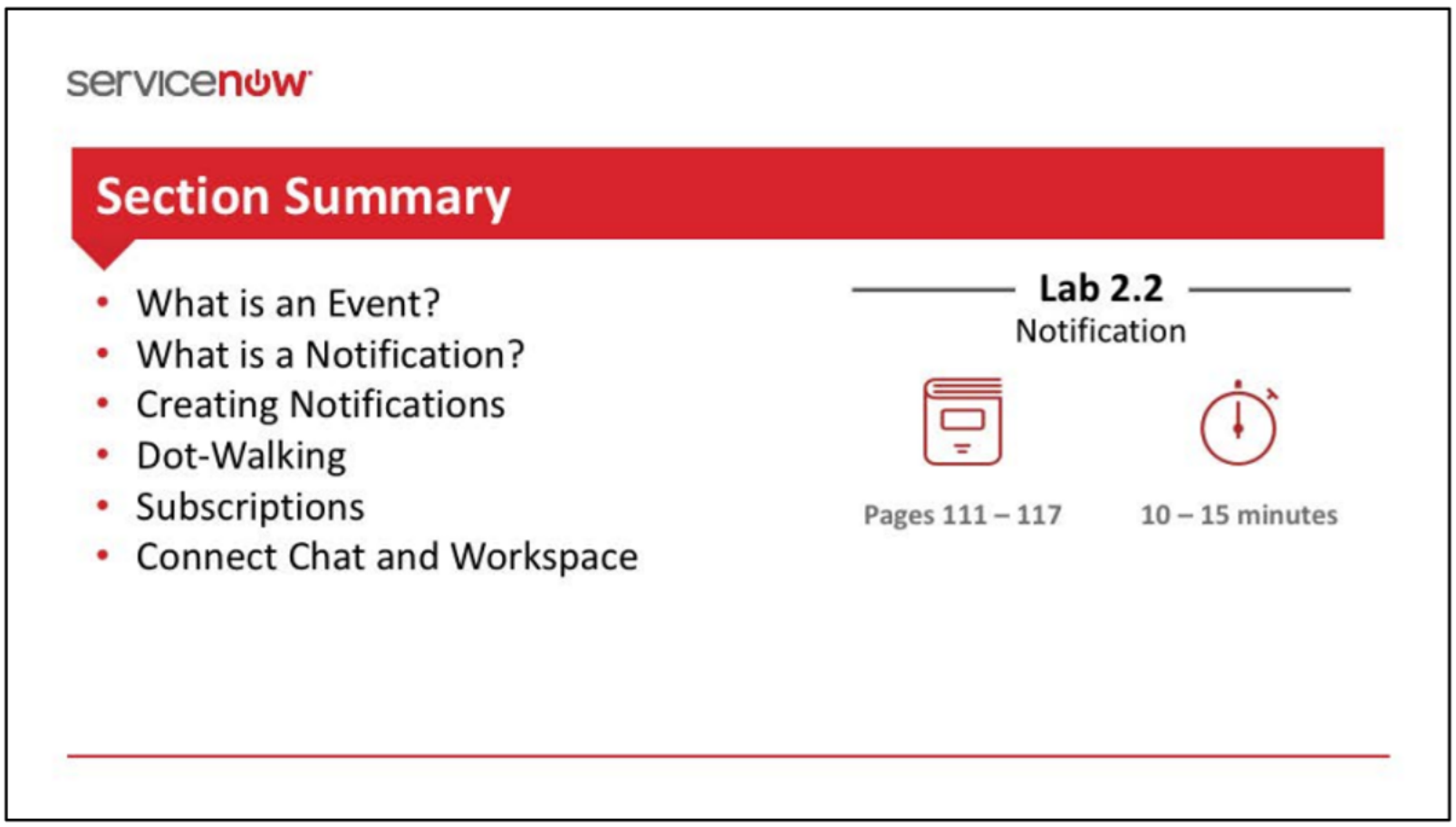


ServiceNow provides several tools to help you communicate with your team in real time, based on information within the platform. **Connect Chat** is accessible from the **Collaborate > Connect Chat** Module (opens full view in new tab) or from the Connect icon in the Banner (opens a sidebar within Content Frame). You can create new conversations with individual ServiceNow users or create custom chat groups. A green dot indicates participants who are currently online. Additional options allow you to add attachments to the chat, customize your notifications to stay in the loop on the conversation, and easily view and update related records.

The Connect Workspace provides a full-screen view of all your Connect Chat and Connect Support conversations in one place, plus additional tools to help keep track of important information in conversations. To open the Connect Workspace, navigate to **Collaborate > Connect Chat** or click the **Open Connect standalone interface** icon from the Connect Sidebar. If you do not have any recent conversations, a screen appears with helpful information about Connect.

The Connect Workspace interface is made up of three major components:

- **Connect Sidebar**: Provides access to conversations. The Connect Sidebar behaves the same way in the Workspace as it does in the Connect overlay. The only difference is that the sidebar appears on the left edge of the Connect Workspace.
- **Conversation Pane**: Displays the currently selected conversation
- **Conversation Tools**: Provide quick access to key information, conversation members, attachments, and notification preferences for the currently selected conversation. Some of the conversation tools vary depending on the type of conversation.



Lab 2.2 – Notification:

- Develop a new email notification
- Test the notification

LAB

Notification

2.2

10 – 15 minutes

Lab Goal

This lab will show you how to do the following:

- Develop a new email notification
- Test the notification

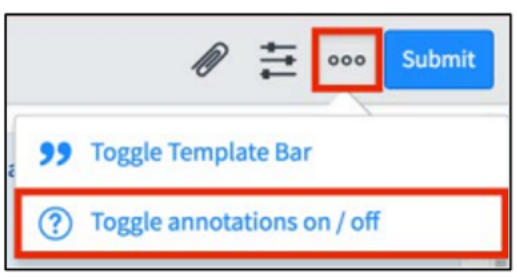
Buster Wubbel – manager of Infinity Security – has requested that a notification be created to alert him and his team whenever a critical employee Infinity incident is active, and categorized with Security.

Additionally, the notification should alert any manager of the assigned group, seeing as the incident as urgent.

A. Develop a Notification

We will assume Buster Wubbel has shared the notification requirements with the system administrator – whom which you will impersonate to create the new notification.

1. Impersonate **System Administrator**.
2. **System Notification > Email > Notifications**.
3. **New**.
4. After the new notification record loads, open the **More options** menu from the form header, then select **Toggle annotations on / off**:



***NOTE**: **Annotations** appear on various forms and contain useful information. It is always recommended to read them before toggling them off.*

5. Set the **Name** to **P1 Infinity Incident**:

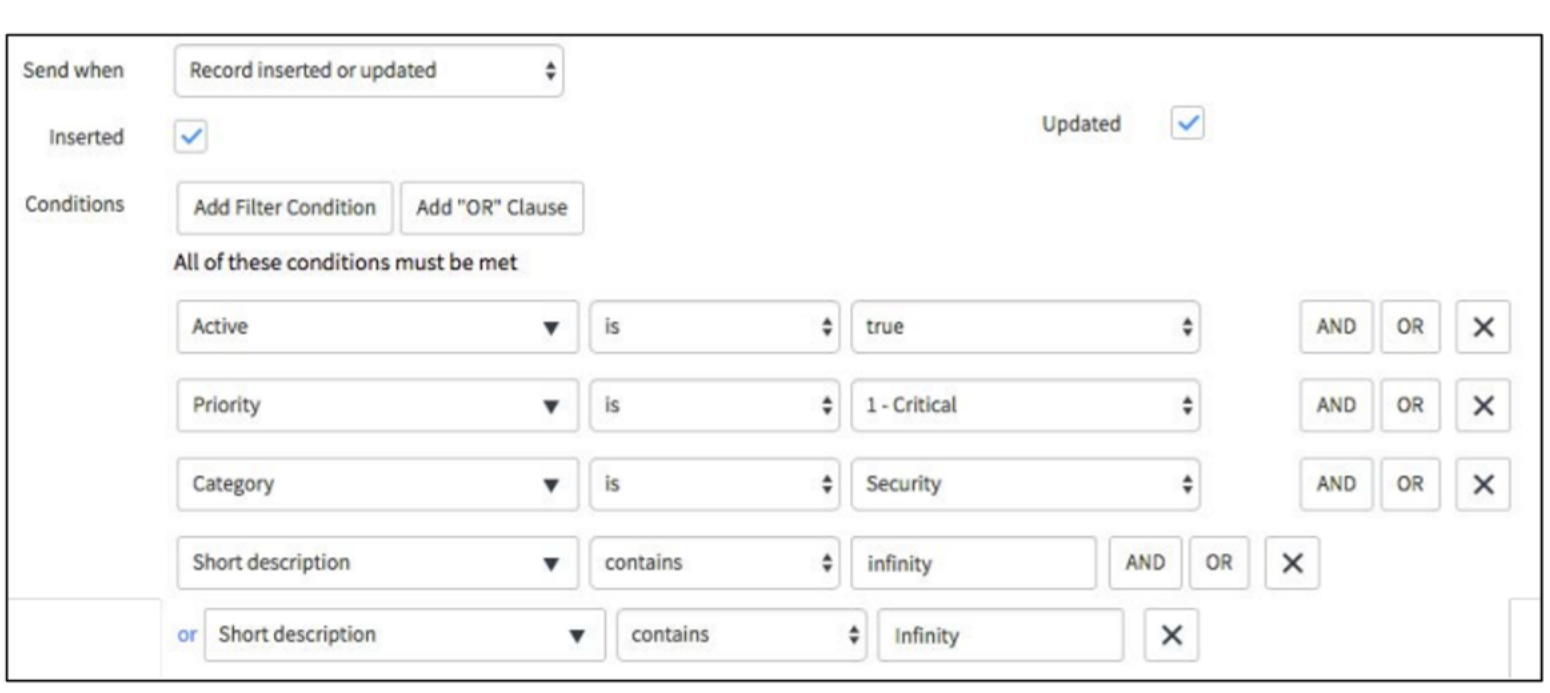
This notification will be defined to inform the Infinity Security group and the assignment group's manager whenever a critical (Priority 1) employee Infinity security incident is active as a result of being created or updated.

6. Under the When to send tab, set the following values:

Inserted: **[checked]**
Updated: **[checked]**

Conditions:
Active | is | true **AND**
Priority | is | 1 – Critical **AND**
Category | is | Security **AND**
Short description | contains | infinity
OR Short description | contains | Infinity

Your **When to send tab** should look like this:



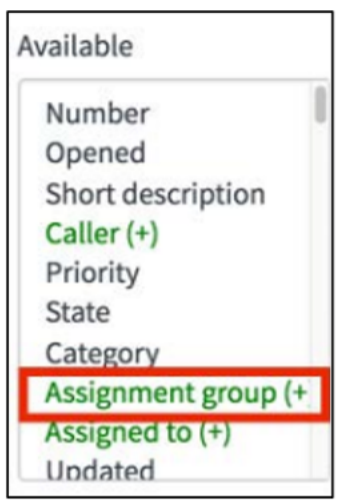
***NOTE:** The **Security** category choice was previously defined for this exercise.*

7. Select the **Who will receive** tab.

8. Click the **Unlock** icon (closed lock) for **Users/Groups in fields**:

Instead of selecting a user by name, thus hardcoding that particular user to the notification, select a field that contains the data of a user account. Doing so will require dot-walking tables.

9. Highlight the **Assignment group (+)** field (formatted in green) under the **Available** bucket:



***NOTE**: The **(+)** icon indicates a field is a reference field, which can then be used to dot-walk from one table to another.*

---

10. With **Assignment group (+)** highlighted, click the **Expand Item** icon (+) between the **Available** and **Selected** buckets:

***NOTE**: Doing so successfully dot-walks from the Incident table into the User table.*

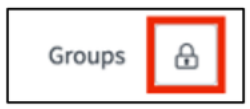
---

11. Scroll down in the **Available** list to find and then highlight **Manager [+]**:

12. Click the **Add Item** icon (>).

    This adds **Assignment group.Manager** to the **Selected** bucket or, in other words, the notification will be sent to the manager of the incident's assignment group.

13. Click the **Edit Groups** icon (closed lock) on the **Groups** field:



14. Search for and select **Infinity Security**.

15. Click the **What it will contain** tab.

16. Type the following into the **Subject** field:

    IMPORTANT! P1 Infinity Security Incident Created

17. Add a dynamic value placeholder to the subject:

    a) From **Select variables**, expand the **Fields** list by clicking the + icon

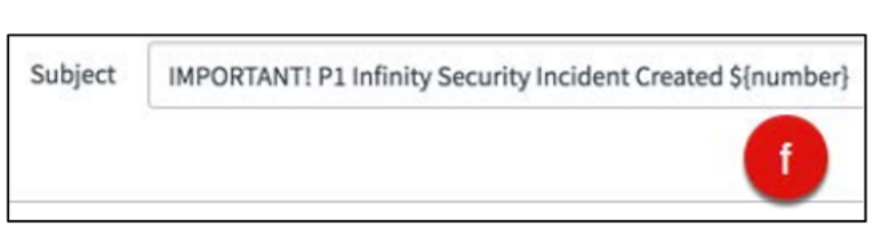
    b) Scroll down and select the **Number** field variable

c) Highlight **${number}** from the **Message HTML** field, then copy the text to your clipboard

d) Click your cursor into the **Subject** field at the end of the text

e) Press the **spacebar** on your keyboard once

f) Paste the **${number}** text from your clipboard



***NOTE**: The **${number}** placeholder is added to the end of the subject text and is dynamic, meaning it will automatically populate with the incident number value of the record that triggers the notification. Additional placeholders can be found and added from the **Select variables** section to the subject and Message HTML fields.*

---

18. Update the **Message HTML** field to include the following text, replacing any existing text:

    Critical Incident ${URI_REF} has been created with an Infinity Security category.

***NOTE**: The **${URI_REF}** placeholder includes an upper-case "i" after the "UR" text. Check to see how this placeholder behaves after the notification is generated.*

---

19. Click **Submit** to save the notification.

B. Test and Verify the Notification

Create an incident to trigger the notification, then check to verify it was sent.

***NOTE**: Email is not enabled for the ServiceNow Lab Instance.*

---

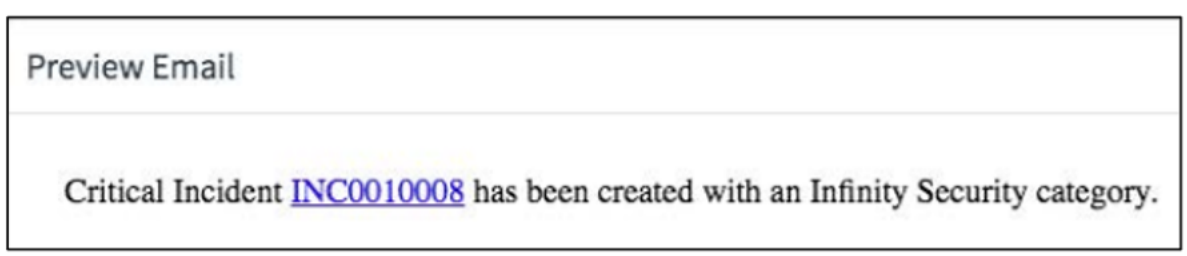
1. Impersonate **Buster Wubbel**.
2. **Incident > Create New**.
3. Fill out the record as follows:

   Caller**: Buster Wubbel**
   Category**: Security**
   Impact**: 1 - High**
   Urgency**: 1 - High**
   Priority**: 1 – Critical** (autofills)
   Assignment group**: Service Desk**
   Short description**: Testing P1 Infinity Security Notification**

4. **Submit** the incident.

   Because email is disabled on the instance, impersonate **System Administrator** to check the instance's email logs.

5. Impersonate **System Administrator**.
6. **System Mailboxes > Outbound > Outbox**.
7. Locate the record with the Subject **IMPORTANT! P1 Infinity Security Incident Created INC#######**, then click on the **Created** timestamp to open the record.
8. After reviewing the **Recipients** list, scroll down further, then finally click the **Preview HTML Body** related link to display a preview of the message received by the recipients:



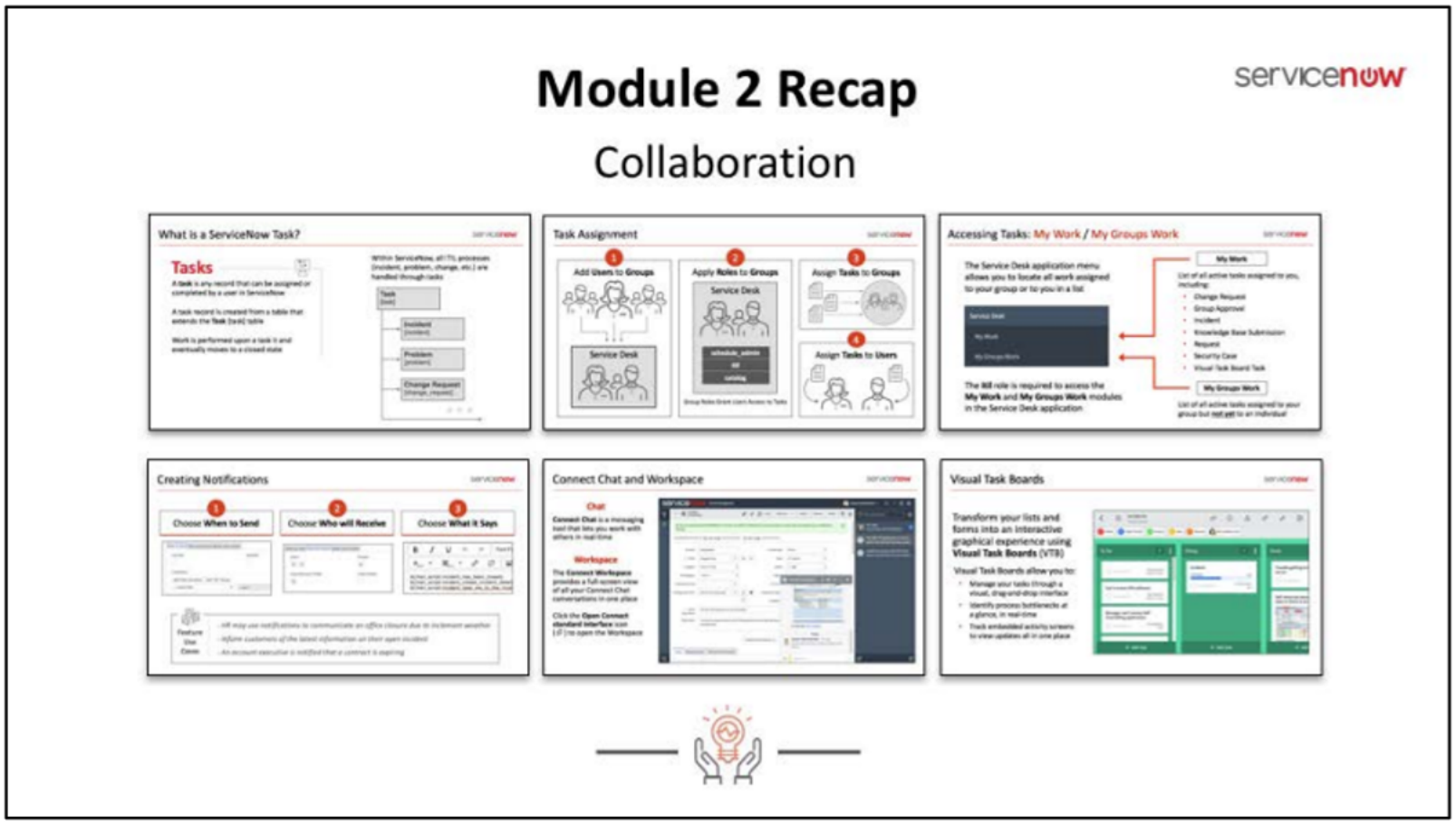
***NOTE:** The **${URI_REF}** placeholder renders as a direct link to the incident record.*

LAB VERIFICATION

Outbox: Infinity Security P1 Notification

Recipient List

Great work! You have successfully created and tested a new email notification.



For these selected topics, discuss:

Why would you use these capabilities?

When would you use these capabilities?

How often would you use these capabilities?

servicenow

1 User Interface & Navigation

2 Collaboration

3 **Database Administration**

4 Service Automation

5 Intro to Scripting & Application Tools

Module 3 – 3.1 Data Schema

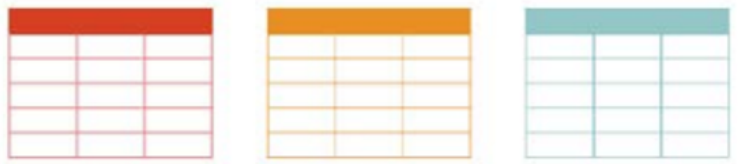
servicenow

Objectives

- Database Structure and Hierarchy
  - Tables
  - Records
  - Fields
- Reference Fields
- Table Relationships
- Table Types
  - Base
  - Core
  - Custom
- Schema Map

ServiceNow Database

Everything in ServiceNow is built on a **MariaDB** database, containing tables, which you access through the ServiceNow Graphical User Interface (GUI)



The **Database** contains **Tables**, **Tables** contain **Records**, **Records** contain **Fields** – a **Field** is an individual piece of data in a record

Data in ServiceNow is stored and managed according to a structure that administrators can view and configure. A table is a collection of records in the database. Each row on a table list corresponds to a record in a table, and each column on a table list corresponds to a field on that table.

Instance information is stored in database **Tables**, which contains **Records.** A **Table** is a data structure or database component and data is stored in tables. The individual pieces of data in a record are called **Fields**. When looking at data in a list view, data in these fields can be modified using the record List Editor functionality.

Lists and forms provide a friendly user interface (UI) for managing tables and records.

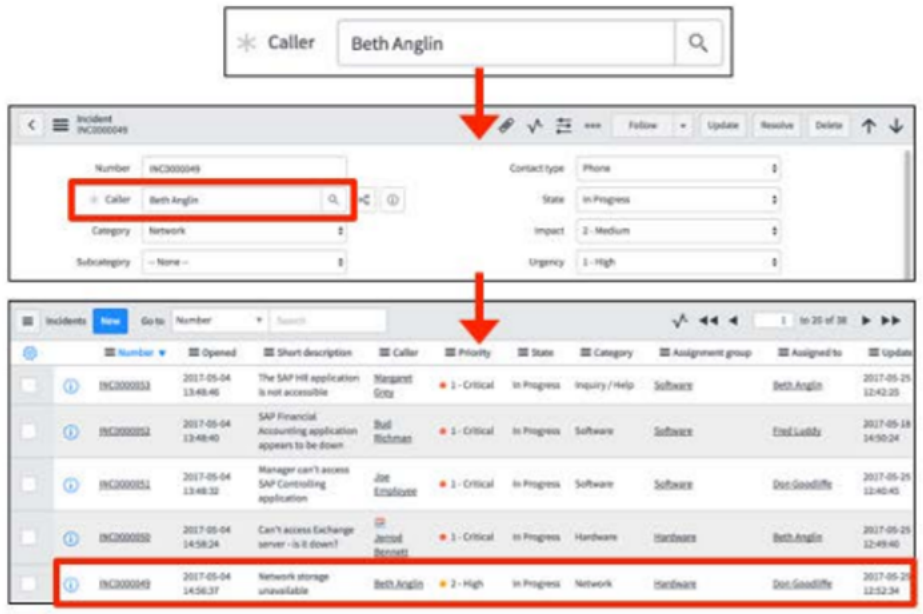
Record Hierarchy

servicenow

Information is documented and saved in **fields**

Many fields make up a record, which can be viewed and edited using a **form**

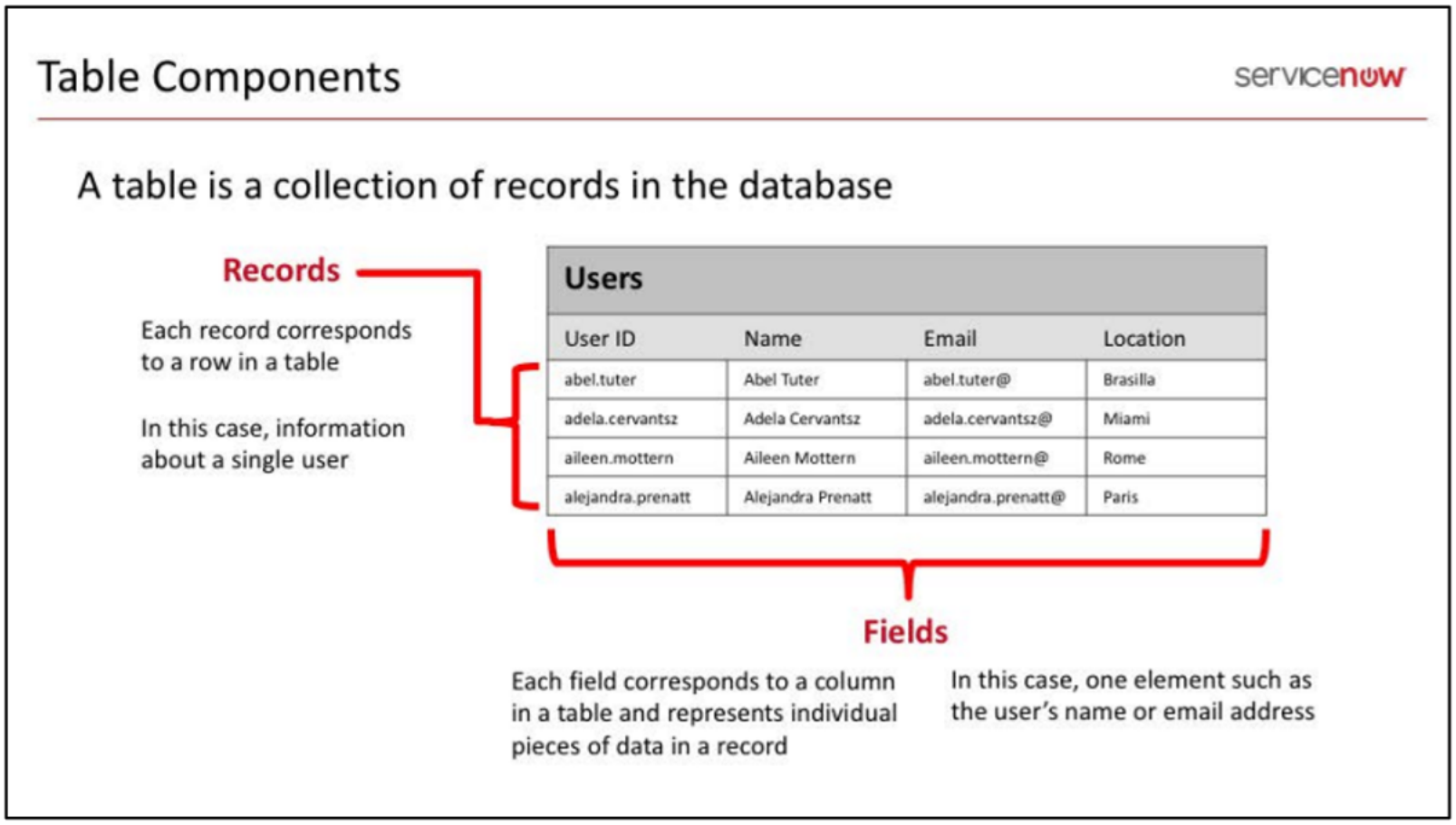
One or more records make up a searchable, sortable **list**



Data in ServiceNow is entered into individual **fields,** which are displayed on a **form.** Once saved, these forms make up **lists.** A list typically comprises forms with a common theme, such as a record type, an assignment group, or a priority.

You can then sort or search in these lists to locate a particular form, which contains data in its fields.

NOTE: ServiceNow often uses the term **record** to describe all the data saved within a particular form. The form is simply the way that the user can view and modify that record.

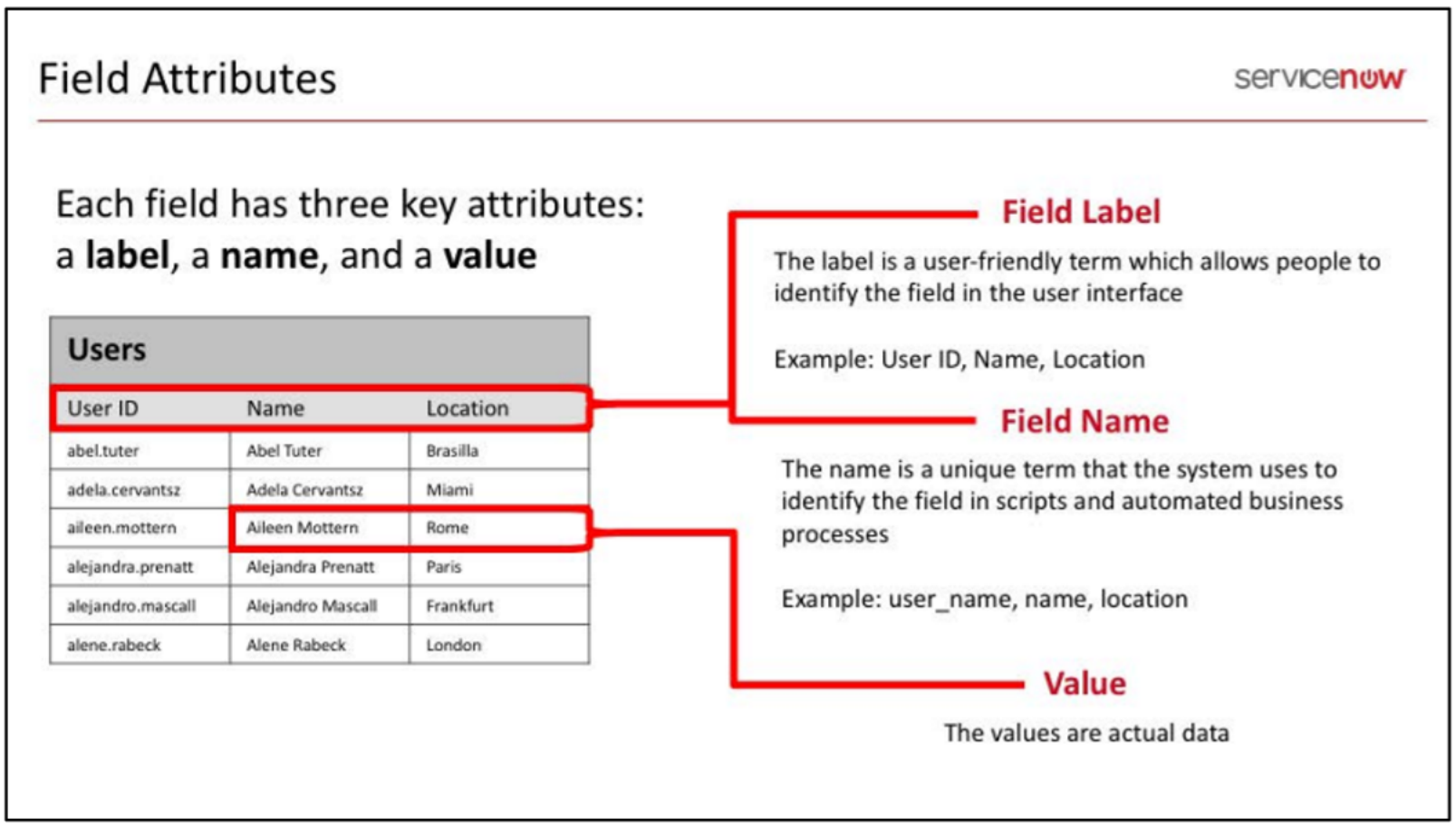


Records are identified by a 32-character, globally unique ID, called a **sys_id**.

Record numbers are automatically incremented, and the number format per table in the system can be changed by visiting the **Number Maintenance** application.

For example, change the default incident record number prefix from "INC" to "IN."

Fields are available in a variety of different types, including: Choice, Date/Time, Journal, Reference, and more. Field types define how a field is interacted with through the interface, as well as the type and format of data it can store.



Each field has three key attributes: a **label**, a **name**, and a **value**.

The field **label** in this example includes User ID, Full Name, and Location

The **name** is a unique term that the system uses to identify the field in scripts and automated business processes. For example, the name attribute of the Location field on the User table is sys_user.location. Do not confuse the name attribute with the Name field on the user table, which is a label. The name attribute of the Name field on the User table is sys_user.name.

NOTE: The Name field is a combination of the First name and Last name fields of the user record.

Values are the actual data, such as this user's name, Aileen Mottern, or her location, Rome. In some cases, the value may be empty, or null.

Reference Fields

servicenow

Reference fields are identified with the **reference lookup icon** (🔍)

The reference lookup icon opens a dialog box for locating a record to reference, presented as a list of the referenced (target) table

If a record is specified in the reference field on the source table, you can hover over the **reference icon** (ⓘ) to preview the referenced record (on the target table)

Source Table
Incident

Number	Caller	Category
INC0000015	Fred Luddy	Software
INC0000016	Bow Ruggeri	Hardware
INC0000017	Joe Employee	Inquiry / Help

🔍 Reference

Target Table
User

Name	User ID	Department
Bow Ruggeri	bow.ruggeri	Development
Fred Luddy	fred.luddy	Development
Joe Employee	employee	Sales

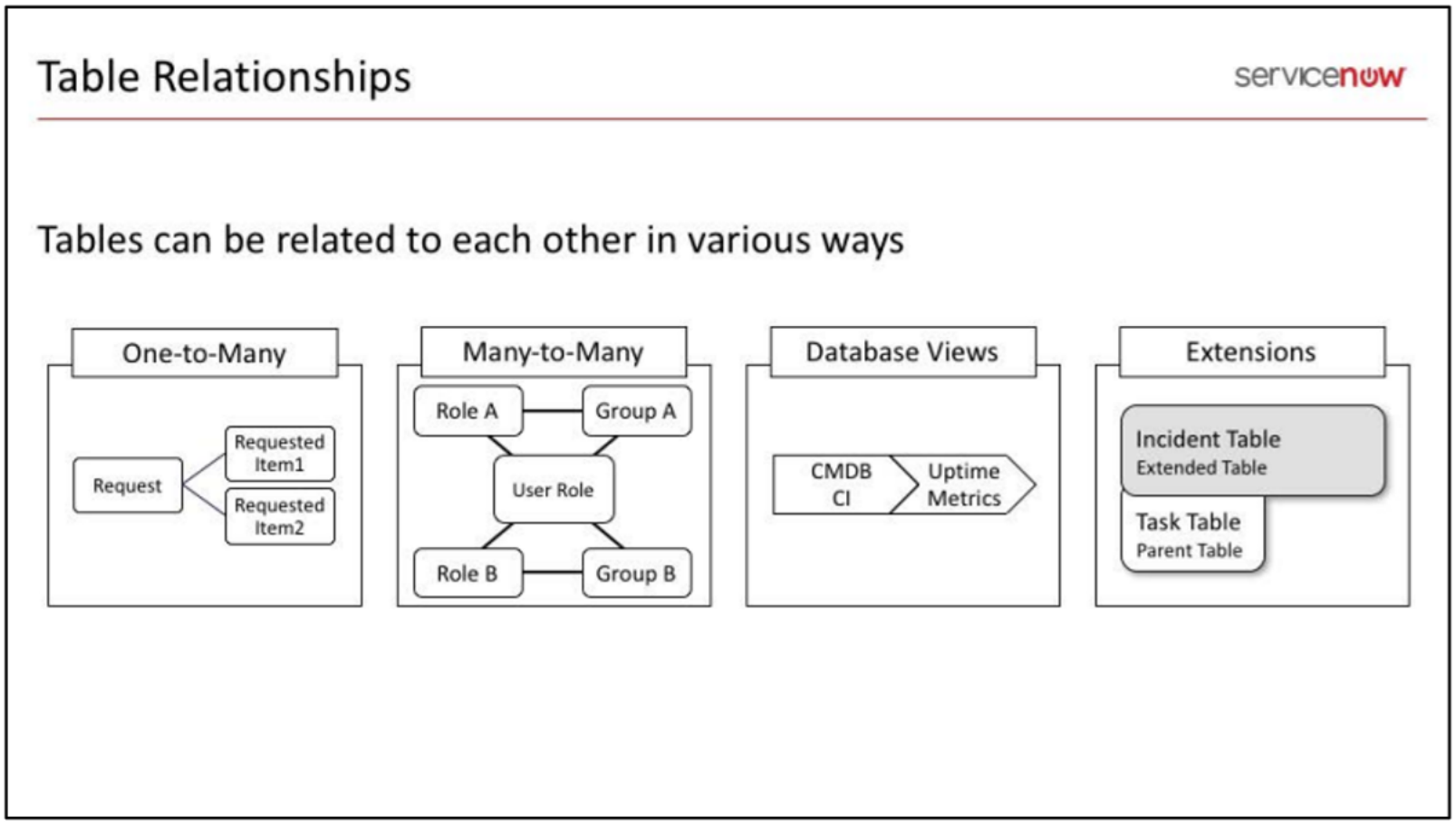
A reference field stores a unique system identifier (known as the sys_id) of a record on another table which is what establishes the reference relationship. For example, the Caller field on the Incident table is a reference to a record on the User table.

When you define a reference field, the platform creates a relationship between the two tables. Adding a reference field to a form makes the other fields in the referenced table available to the form.

Administrators can create new reference fields and configure several options for reference fields.

NOTE: A reference field can refer only to records from one other table. To add a field that can refer to records on any table in the platform, regardless of a shared reference, use the Document ID element type.

Additionally, wildcard searches can be used in reference fields.



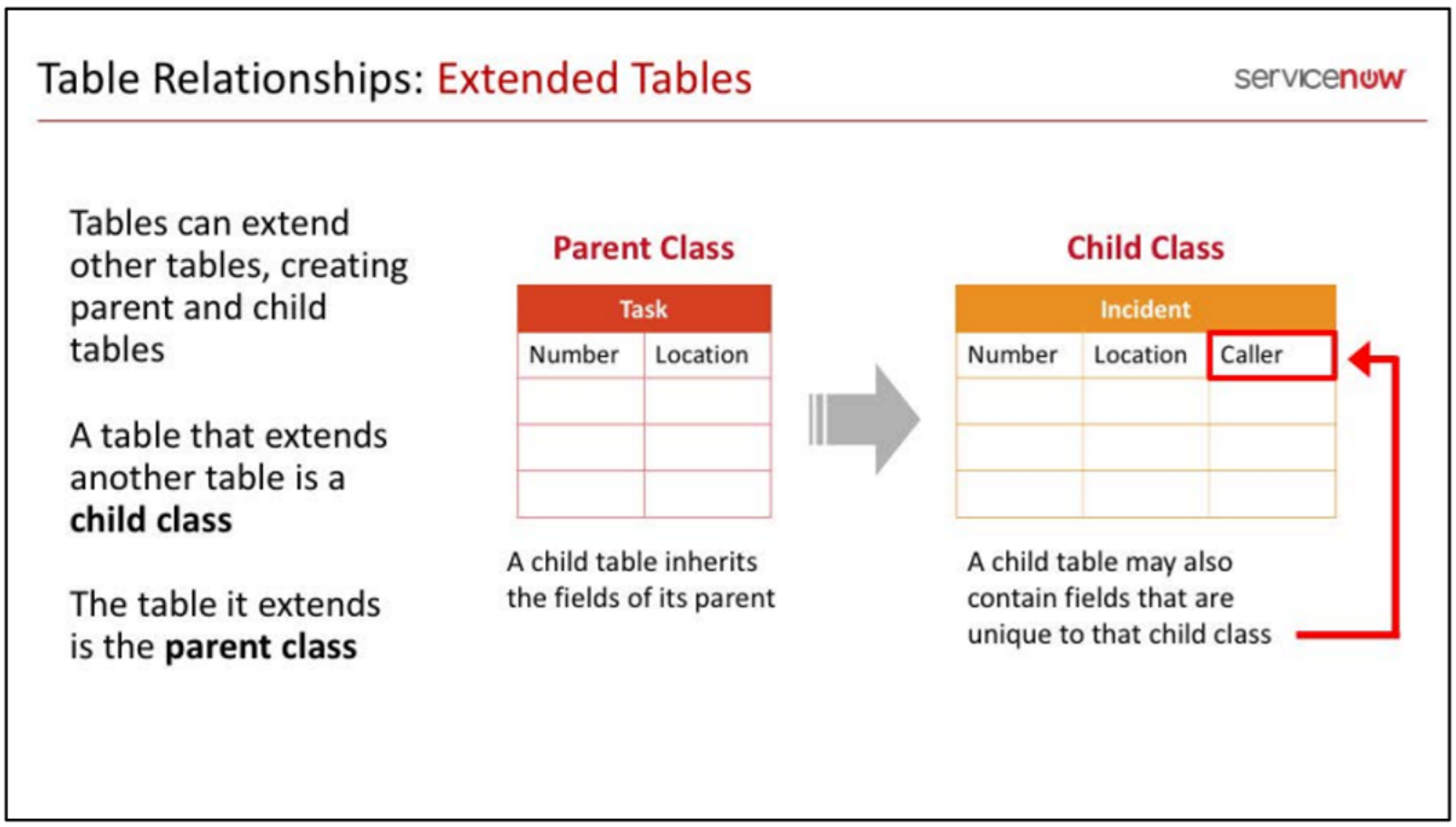
One-to-Many: Within a table, a field can hold a reference to a record on another table. There are three one-to-many relationship fields:

1. **Reference Fields** - Allows a user to select a record on a table defined by the reference field. Example: Caller field on the Incident table allows a user to select any record on User table.
2. **Glide List** - Allows a user to select multiple records on a table defined by the glide list. Example: The Watchlist field on the Incident table allows the user to select any record or records on the User table.
3. **Document ID Fields** - Allows a user to select a record on any table in the instance. Example: Document field on the Translated Text table.

Many-to-Many: Two or more tables can be related in a bi-directional relationship, so that the related records are visible from both tables in a related list. Example: software vendors can sell multiple products and products can be sold by multiple vendors.

Database Views: Two tables can be joined virtually using the Database Views Plugin to allow for reporting on data that might be stored in more than one table. Database Views are read-only. Create Database Views by navigating to **System Definition > Database Views**.

Extensions: A table can extend another table. The extended table includes unique fields plus all of the fields and their properties from the parent table.



In ServiceNow, you can create a new table that stands alone or that extends another table.

The Task table and Configuration table are examples of parent classes that are extended to child classes.

For example, child tables extended from the Task table include Change Request, Incident, and Problem. Child tables extended from the Configuration table include Database, Hardware, and Software.

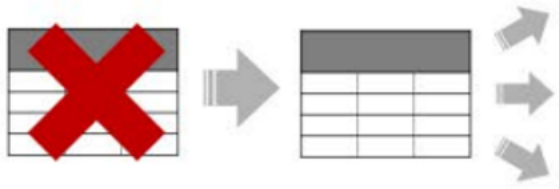
Extending a table incorporates all of the fields of the original table and allows for unique fields to be created on the new table. The inheritance of the fields of the original table is used to create subcategories of data. Examples include the Incident, Problem, and Change Request tables, which are all subcategories of the **Task [task]** table.

Using the **Dictionary overrides** feature provides the ability to define a field on an extended table differently from the field on the parent table. Examples include overriding the default values, field dependencies, or read-only status of a field.

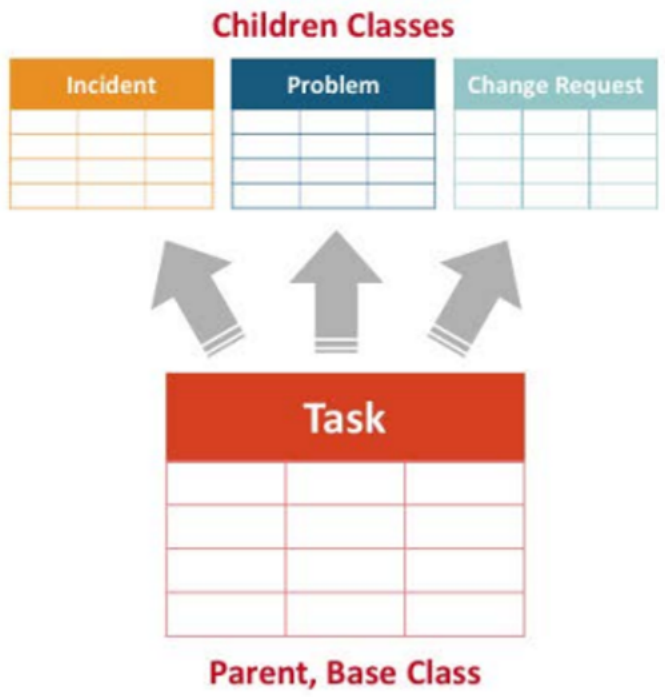
Table Types: Base Tables

servicenow

If a table is extended but itself is not extending another table, it is called a **base table**



The task table is such a table, making it both a parent and a base class

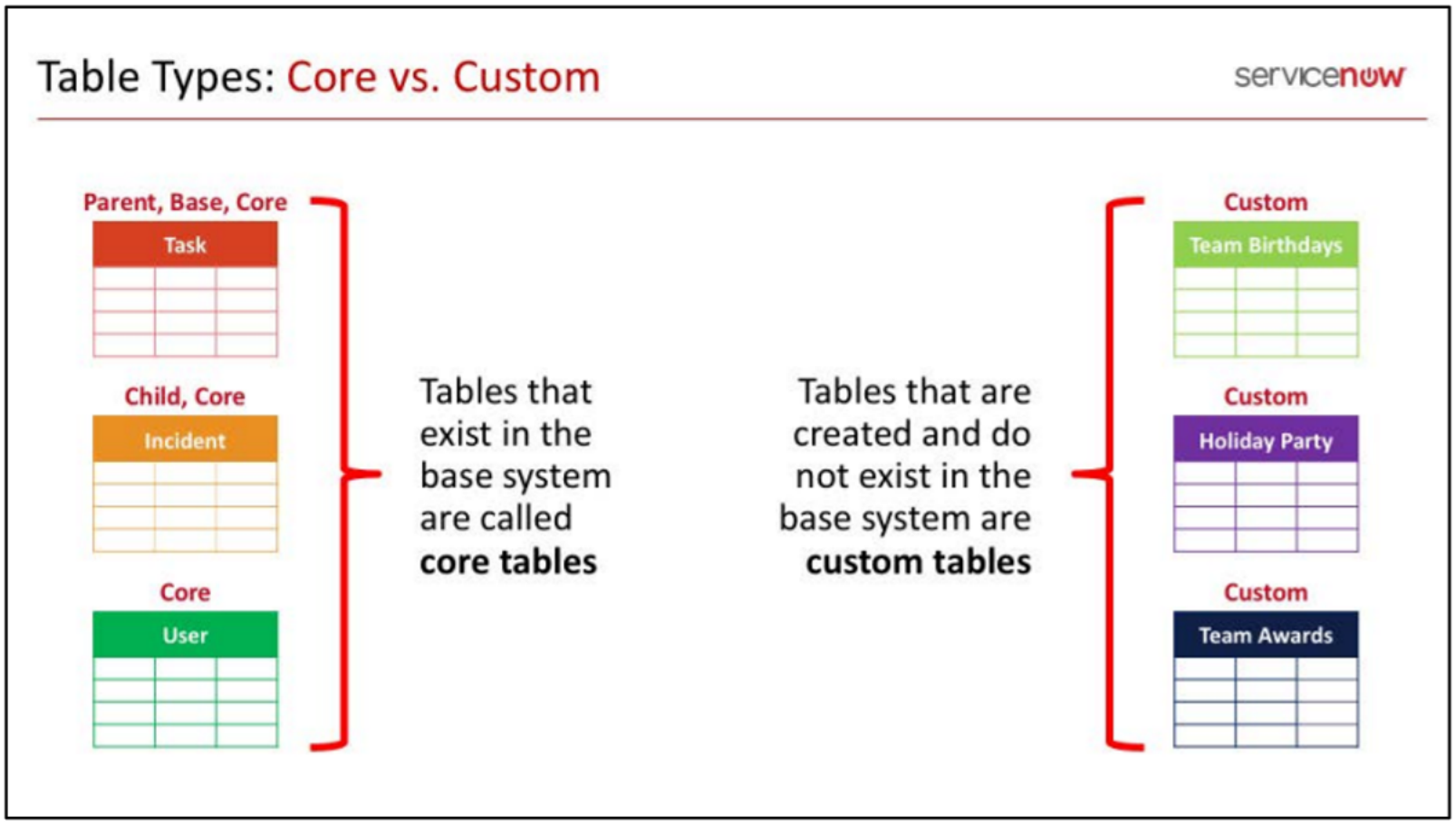


If a field is on a base parent table such as the **Task [task]** table, for example, a different label can be defined for each extended table, such as Incident or Problem. To add a different label for an extended table, navigate to **System Definition > Language File**, then create a new entry for the extended table.

Every child table is a specialization of the previous base table or previous child table. The **Task [task]** table provides a series of standard fields used on every table that extends it.

To extend a table, select the table to extend in the Extends Table field on the table record.

NOTE: This option is available only when you are creating a table but not all tables are extensible.



Although custom tables are not in the base system, they can still interact with existing core tables.

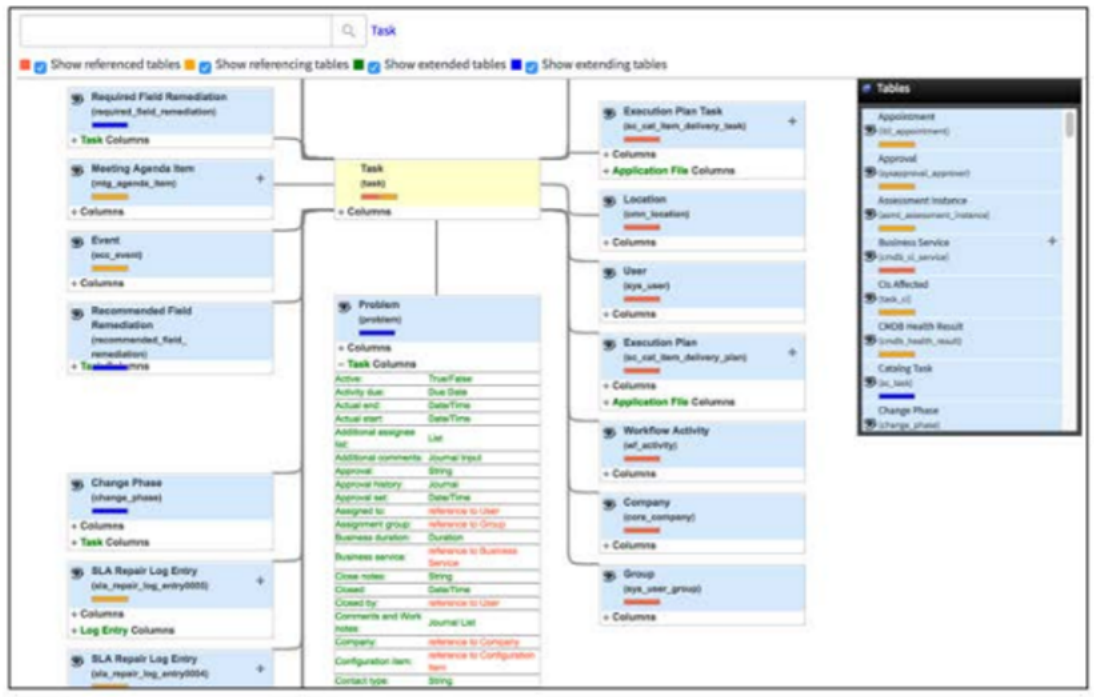
For example, a reference field on a custom table can access data stored on a core table. By doing so, a relationship between the tables is created which makes them related tables. This relationship is not exclusive between just a custom table and a core table. Related tables can be a combination of multiple core tables and/or multiple custom tables.

Schema Map

The **Schema Map** provides a graphical representation of other tables related to a specific table

Relationships can be filtered by extension or reference classes

This tool is available to **admin** users only



In this example map, you will see the Task table as the focus of the map (highlighted in yellow).

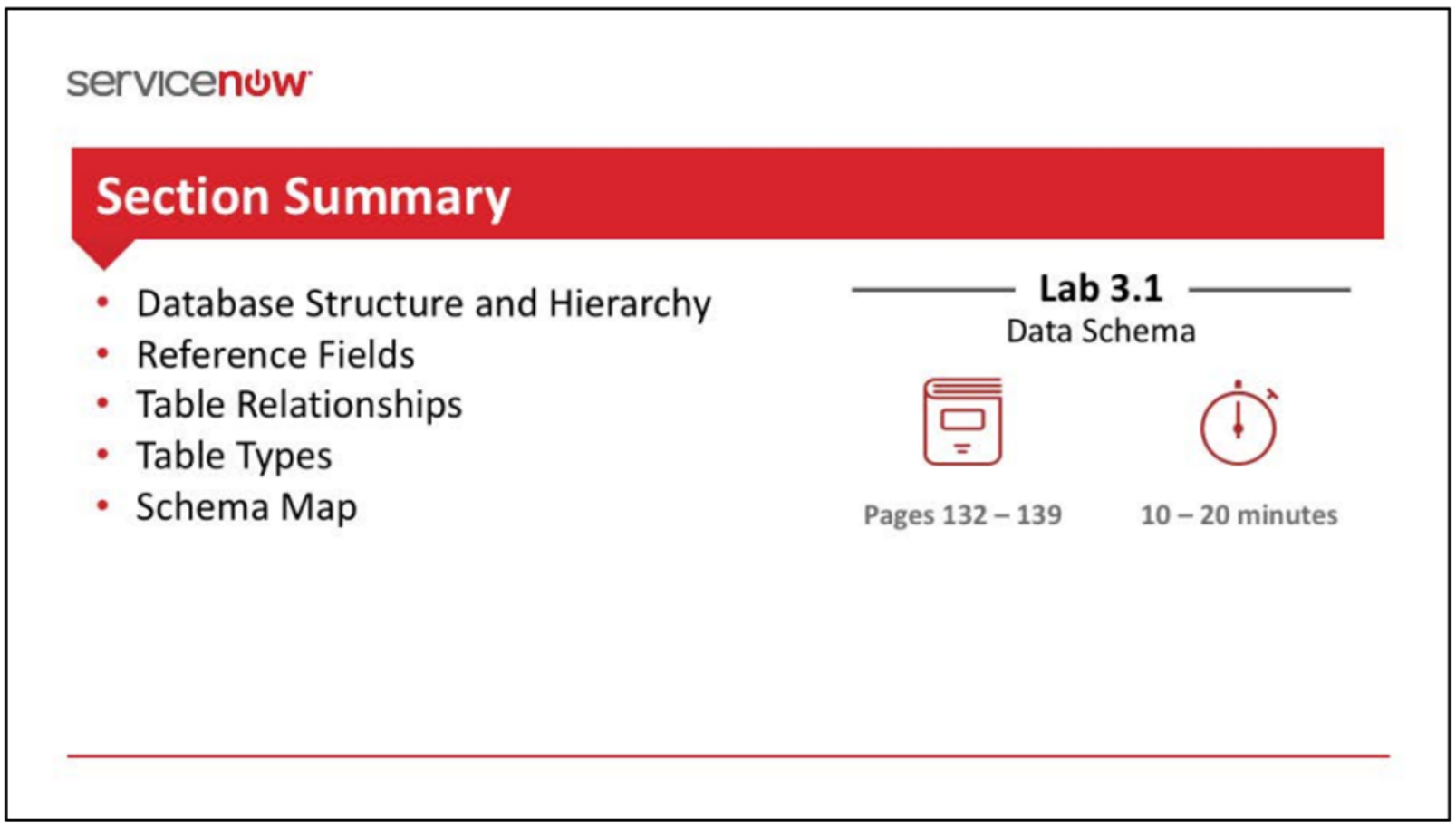
Tables with blue bars, including Problem and Change Phase, are tables that **extend** the Task table.

Demonstrated with the Problem table, you can use the Schema Map to identify what columns (fields) originate on Problem, and which columns are inherited from the Task table. Additionally, you can see what field type they are.

Tables with red bars, including Location and User, are tables that are **referenced** by the Task table.

A series of filters at the top of the Schema Map allow you to show/hide tables based on criteria such as whether they are referenced by the Task table, reference the Task table, are extended by the Task table, or extend the Task table.

The **Tables** window on the far right of the screen provides a summary of all the tables presented and their relationships.



Lab 3.1 – Data Schema:

- Create a new table
- Configure the table form view
- Update the application menu and create new modules

LAB

Data Schema

3.1

10 – 20 minutes

Lab Goal

This lab will show you how to do the following:

- Create a new table
- Configure the table form view
- Update the application menu and create new modules

With an active procedure of testing Infinity devices, Cloud Dimensions needs a method for managing inventory; tracking how many devices have been issued and to whom.

The various teams involved with Infinity testing have come up with a solution but will need the help of the system administrator to implement it.

Their plan is to have this information accessed through an application menu with a series of modules.

The primary data point being tracked will be the Infinity devices but information about the users, such as name and email, will also be available.

A. Create a New Table

1. As **System Administrator**, navigate to **System Definition > Tables & Columns**.

   The Tables & Columns module provides a clean interface for browsing a list of existing tables in the database. Selecting a table name will display its contents: columns (fields) and their attributes.

   This interface provides an easy way to navigate between multiple tables, without having to open individual records to see and compare table content. Please note that all data displayed on this page is read only – it may not be edited, unless you select a table and click the **Edit Table** button.

2. Navigate to **System Definition > Tables** to create a new table.

3. From the list header, click the **New** button:

4. Fill out the top of the Table form with the following changes:

   Label: **Infinity**
   Name: **u_cmdb_ci_hardware_infinity** (auto fills with u_infinity)
   Extends table: **Hardware [cmdb_ci_hardware]**
   New menu name: **Infinity** (auto fills)

NOTE: *The Name field automatically populates with* ***u_infinity****. The table name can be changed, as long as it starts with the* ***u_*** *prefix indicating it is a custom table. It is best practice to rename the table to indicate it is a custom CMDB CI table.*

---

5. **Submit**.

B. Add Fields to the Infinity Form

1. Use the Application Navigator filter field to navigate to **Infinity > Infinities**:

NOTE: *During the creation of the Infinity table, the* ***Create module*** *checkbox was selected. As a result, this automatically created the new module* ***Infinities*** *which is a pluralized form of the table name, Infinity.*

---

2. An empty List (No records to display) is displayed with default fields.

NOTE: *In a future lab, you modify the Infinities list layout before importing device records from spreadsheets.*

---

3. Click **New** to open a form displaying default fields.

4. Open the **Form Context Menu**, select **Configure > Form Layout**.

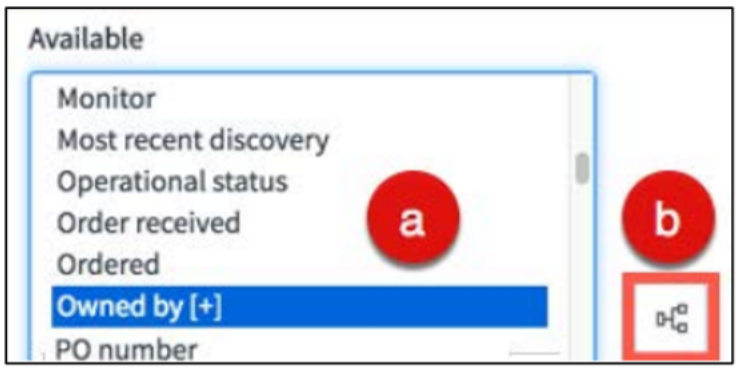
5. Remove the **Assigned to**, **Category** and **Fault count** fields, keeping **Name**, **Asset tag**, **Installed**, and **Status** in the Selected list.

6. Add the **Owned by** and **Support group** fields to the Selected list from the Available list.

7. Using dot-walking, add **Owned by.Email** to the Selected list:

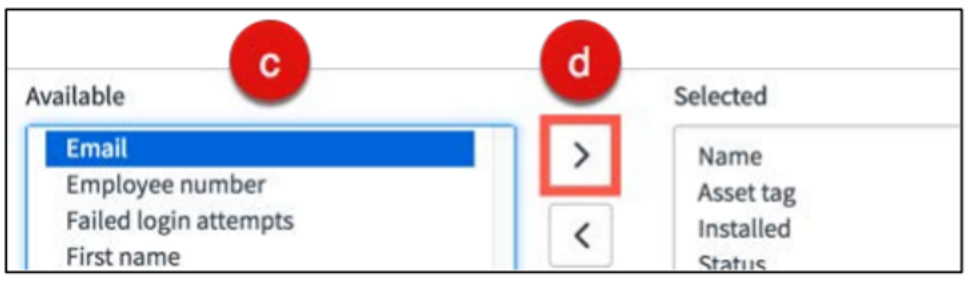
   a) From the Available list, locate and select **Owned by**

   b) Click the **Expand selected reference field** icon

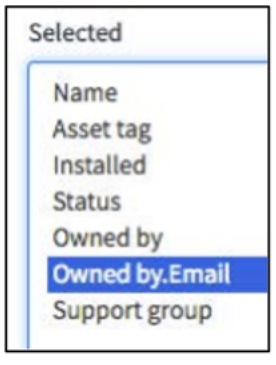


   c) From the Available list, scroll down and select **Email**

   d) Select **Add**



You should now see **Owned by.Email** under the Selected list:



NOTE: *The* ***Owned by.Email*** *field's value represents the email of the device's owner, and will automatically populate when a value is entered into the Owned by field, as long as the user record includes an email address..*

8. Next, in the **Create new field** section, enter:

   Name: **Device Number**
   Type: **String** (autofills)
   Field length: **Small (40)** (autofills)

9. Click **Add**.

10. Add two additional fields:

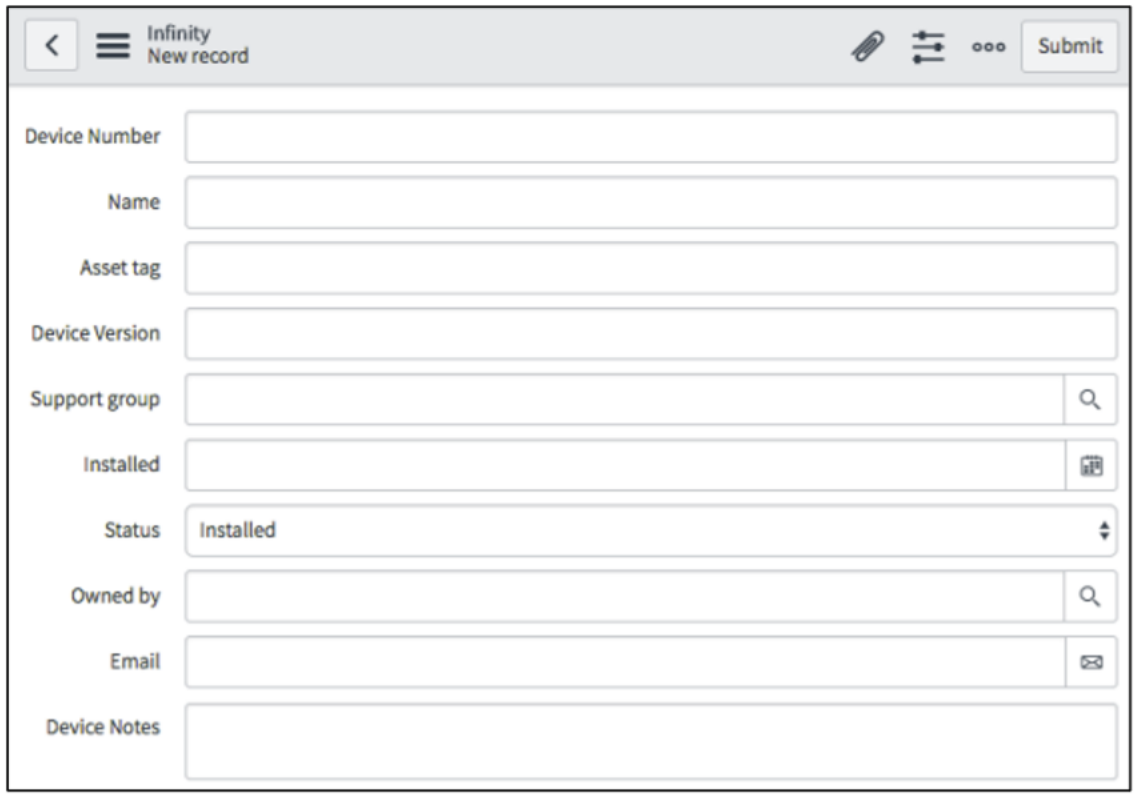
Name	Type	Field Length/Reference
Device Version	String	Small (40)
Device Notes	Journal	

11. After adding the new fields, rearrange the fields under the Selected list to look like this:

12. Click the **Save** button.

13. Close the **Saving Form Section** window.

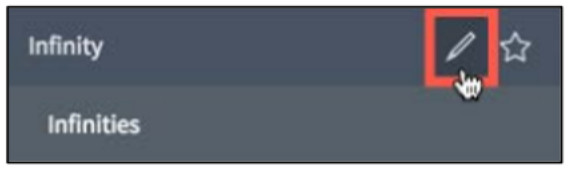
14. The **Infinity** New record form displays:



C. Update the Infinity Application Menu

The new Infinity application menu and its modules will be used to create new Infinity device records from the form you have just designed, as well as display a list of all devices.

1. In the Application Navigator filter field, type **Infinity**.
2. Hover your curser over **Infinity** and click the **Edit Application** icon (pencil):



3. This brings up the Application Menu record for **Infinity**.

NOTE: *You could alternately access this record by navigating to* ***System Definition > Application Menus*** *and searching for Infinity in the list.*

---

4. Update the **Title** to **Infinity Inventory**.
5. **Save** the record.

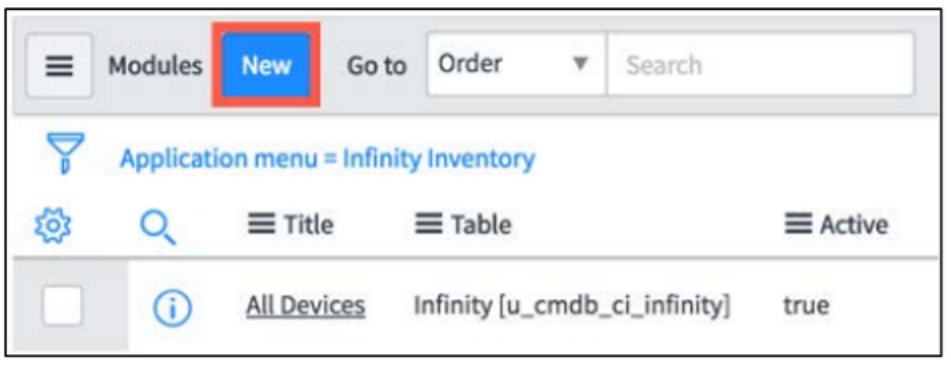
6. From the Modules section, open the **Infinities** record:

7. Update the record as follows:

   Title: **All Devices**
   Order: **200**

8. Click **Update**.

9. Next, click **New** from the Modules section:



10. Fill out the form:

    Title: **Add Inventory**
    Order: **100**

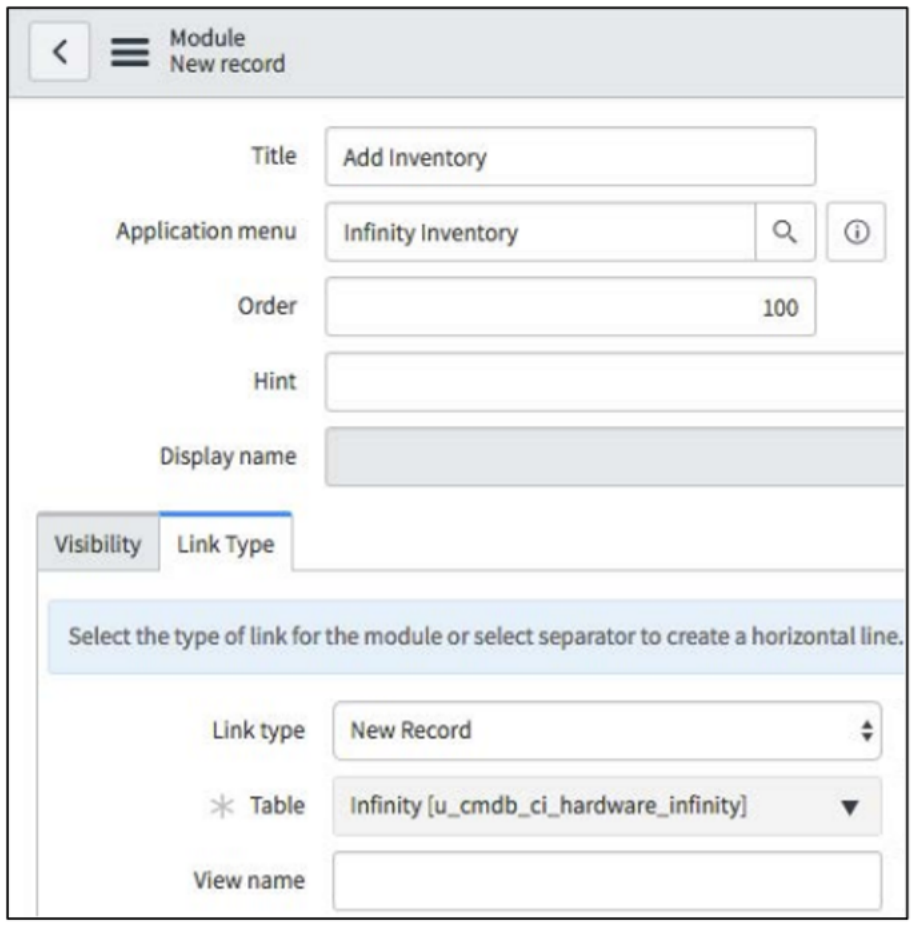
11. Click the **Link Type*** tab and fill out the fields as shown:

    Link type: **New Record**
    Table: **Infinity [u_cmdb_ci_hardware_infinity]**

NOTE: *The* ***Table*** *field appears to be read-only (indicated by gray), but clicking on the table name will open the drop-down menu.*

---

12. Your screen should look similar to this:



13. Click **Submit**.

Knowledge Check

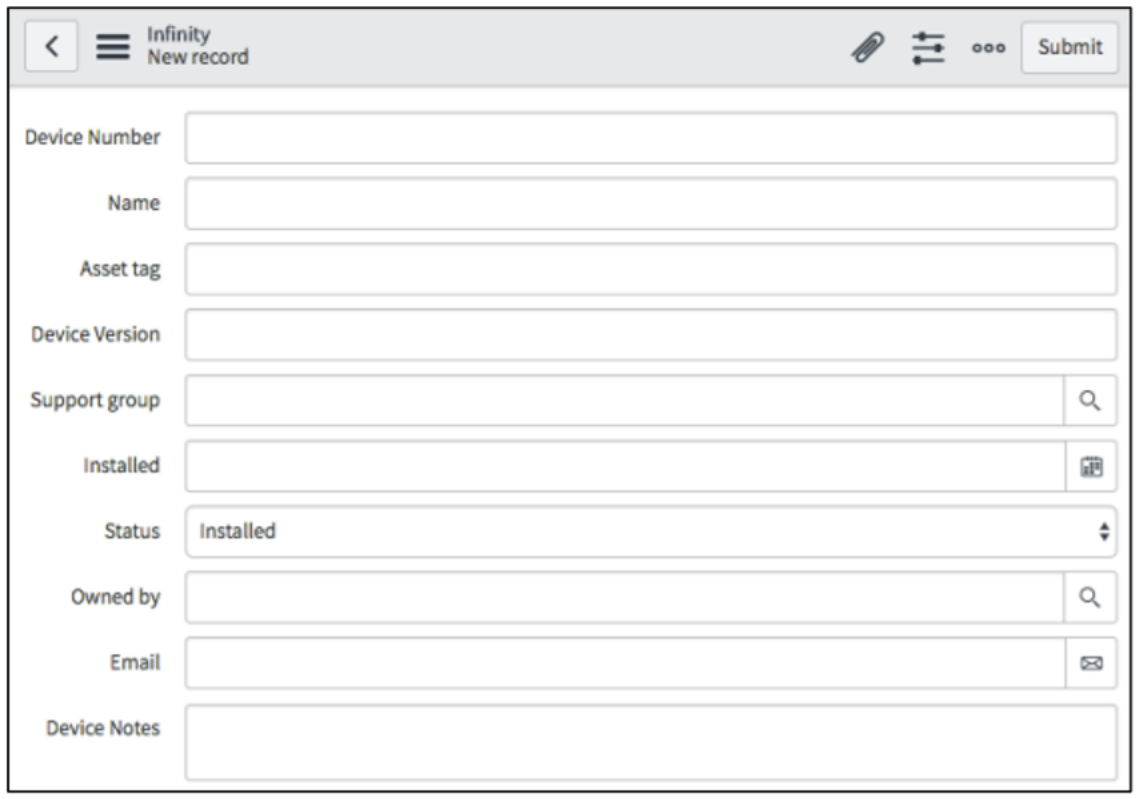
Before creating any table in ServiceNow, the question to always start with is: should the table be created from scratch or by extending an existing table?

Why do you think the Configuration Item table extended instead of creating a new table?

LAB VERIFICATION

New Table - Application Menu and Modules

Add Inventory Module (Form View)



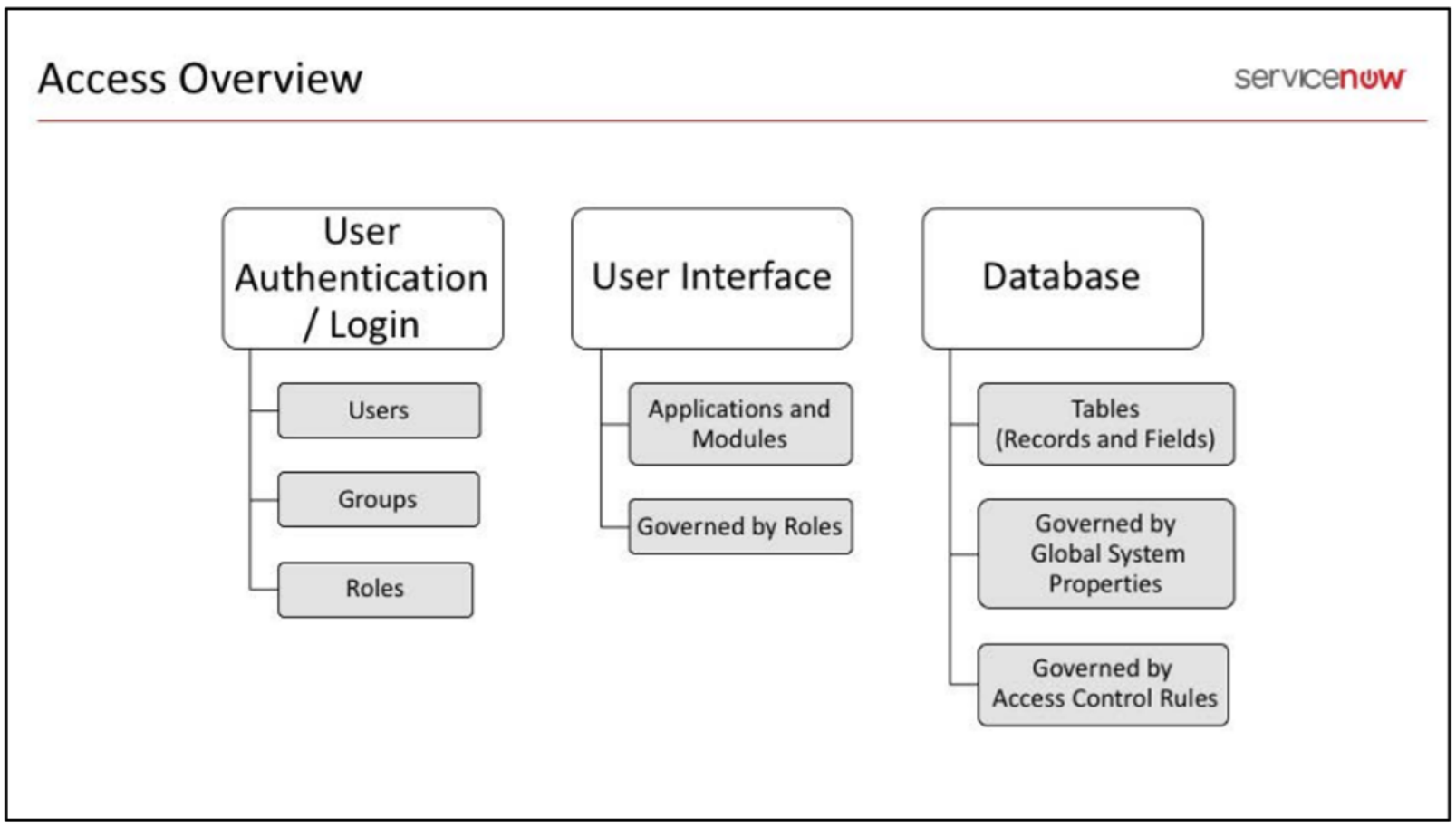
Cloud Dimensions now has its first custom table, thanks to you!

Module 3 – 3.2 Data Security

servicenow

Objectives

- User Permissions Summary
- Access Control Rules
  - What is an Access Control?
  - Access Control List (ACL)
  - Rule Types
  - Requirements
  - Using the Wildcard
  - Evaluation Workflows



ServiceNow provides several levels of security before an end user has the capability to perform CRUD (Create, Read, Update, Delete) operations on a table:

- **User Authentication/Login:** Users, Groups, and Roles
- **Application and Modules Access**: Controlled by roles configured at the Application and Module level
- **Database Access**: Access to tables and their records and fields are controlled via globally defined system properties as well as table and field level Access Controls. If a row level rule and a field level rule are in conflict, both rules must be true before an operation is allowed.

There are three security modules typically used by the System Administrator:

- **System Properties > Security**
- **System Security > Access Control (ACL)**
- **System Security > High Security Settings**

What is an Access Control?

servicenow

Access Control

An **Access Control** is a security rule defined to restrict the permissions of a user from viewing and interacting with data

It is executed when attempting to access any ServiceNow table and may be set at the:

row-level and/or **column-level**

(access to the record)

(access to the field)

These rules restrict ServiceNow-specific and CRUD operations

C reate

R ead

U pdate (write)

D elete

Most security settings are implemented using Access Controls.

In addition to restricting CRUD operations, Access Control rules can restrict ServiceNow-specific operations on tables and fields.

ServiceNow operation examples include:

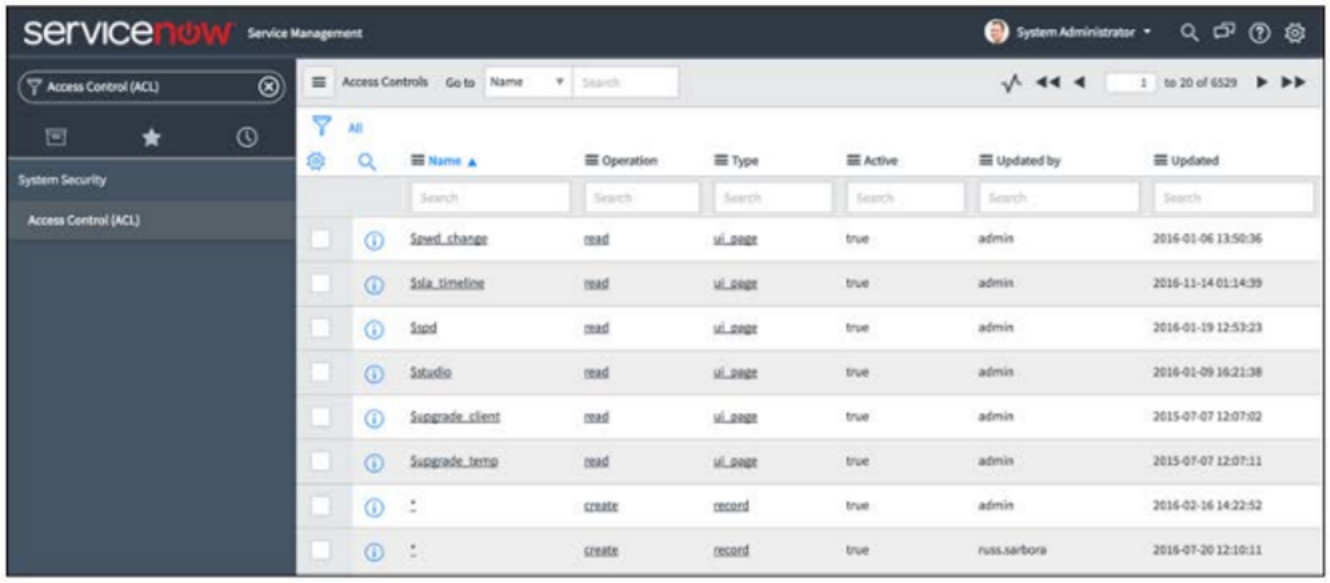
- **execute**: user cannot execute scripts on a record or UI page
- **Edit_ci_relations**: user cannot define relationships between Configuration Item [cmdb_ci] tables
- **Save_as_template**: controls the field that should be saved when a template is created
- **Report_on**: user cannot create reports on the object
- **Personalize_choices**: user cannot right-click a choice list field and select Configure Choices

Access Control List (ACL)

servicenow

The **Access Control List (ACL)** contains all of an instance's Access Control rules

Users with the appropriate permissions can modify rules and their definitions



All users with the admin role have special access to all platform features, functions, and data because admins can override Access Controls and pass all role checks so grant the admin privilege carefully! With this said, in order for a user to create or update Access Control roles, they must have the **security_admin** role.

Access Control Definition: Rule Types

servicenow

table.None	table.field	table.*
		
No specific field selected - this rule applies to the whole table including its records	This rule applies to only one field on a record and in this case, the Caller field on an incident record	Wildcard – this rule applies to every field on a record without a table.field rule

Each Access Control specifies the table or type of record (including fields), operation being secured, and unique object identifier. Access Control rules are defined for and applied to a specific table so that the rule is within the context of the table and the type of data stored.

Access Control Definition: Requirements

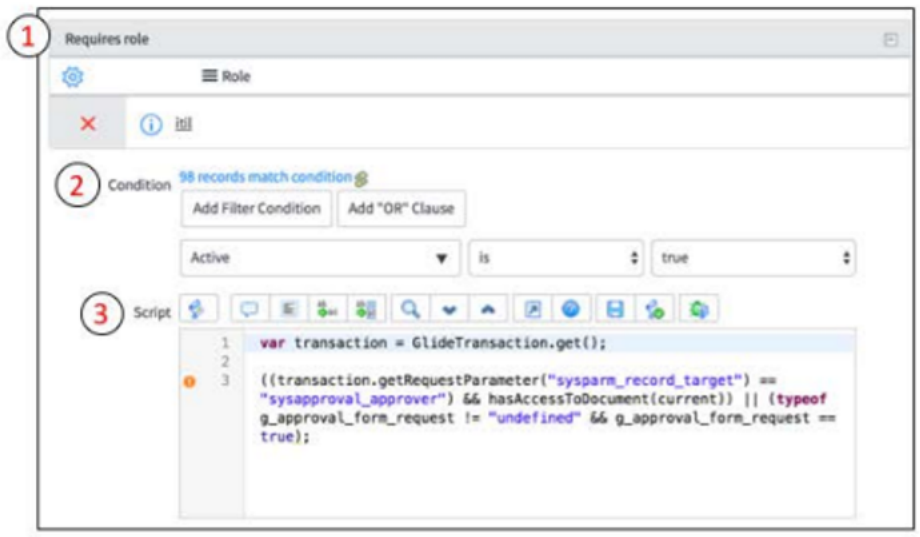
servicenow

Each Access Control rule specifies

- The object being secured
- The permissions required to access the object

Access Control rules require users to pass a set of requirements in order to gain access to particular data:

1. Roles
2. Conditional Expressions
3. Scripts



Access Control Rules can be defined with the following three components, which relate to the set of requirements a user must pass in order to gain access to particular data:

1. **Roles:** If you put one or more roles there, then only users with at least one of those roles are allowed to perform the requested operation.
2. **Conditional Expressions:** You will see a condition widget where you can add conditional expressions to your ACL. For example, you might have a conditional expression that says “Active = true" if you want this rule to evaluate to true only for database records that meet a specific condition.
3. **Scripts:** If the Advanced checkbox is selected, you then have an opportunity to apply security based on user defined script. Your script has access to the current record and has responsibility for setting a global "answer" variable to allow, or deny, access to the requested resource/operation. A script is evaluated *in addition to* other conditions you set on the rule.

   All must evaluate to true.

Access Control Examples

servicenow

Rule Example 1

Rule Name: **incident**

Rule Operation: **read**

Definition:

Script

```
1  current.opened_by == gs.getUserID() ||
   current.caller_id == gs.getUserID() ||
   current.watch_list.indexOf(gs.getUserID()) > -1;
```

Description:

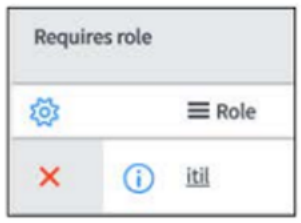
You can read an incident if you opened it, are the caller of record, or are on the watch list

Rule Example 2

Rule Name: **incident**

Rule Operation: **read**

Definition:



Description:

You can read an incident if you have the **itil** role

In the first rule example, the access control's definition includes only a script requirement which roughly translates to: is the user that opened this record the same user attempting to access it, or is the user that is listed as the caller the same user attempting to access it, or is the user that is listed on the record's watch list the same user attempting to access it?

If the user meets any single one of these criteria, they are able to read (or view) the incident record. If they do not meet a single one of these criteria, then they are not able to read the incident record.

To further illustrate this, think of the following scenario: Joe Employee called in to the Service Desk to report an issue. The support agent receiving Joe's call created an incident and added Joe's name to the Caller field. Later, when Joe attempts to view the record – possibly to see any progress made with it – he will have no problem doing so, because of this rule.

Now, while still thinking of this scenario, the agent that opened the incident record would technically be able to view Joe's record because of that same rule. However, what happens if they reassign the incident to another fulfiller user?

Luckily, the second rule example would allow that other fulfiller the chance to view the record. This rule states that in order to read an incident, you have to have the itil role.

In the case of these two rules, because they are applied on the same object level (reading an incident record), passing either rule will grant access to the user.

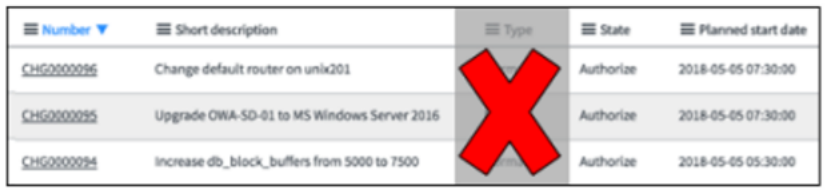
Using the Wildcard

servicenow

The selection of the wildcard (*) symbol from the Namedrop-down list on the Access Control form means that the rule in question applies to all fields on the selected table ***except*** for those with explicit rule

change_request.None read Access Control for itil_admin and itil roles
change_request.* read Access Control for itil_admin role
change_request.type read Access Control for itil role

itil_admin role is able to see:



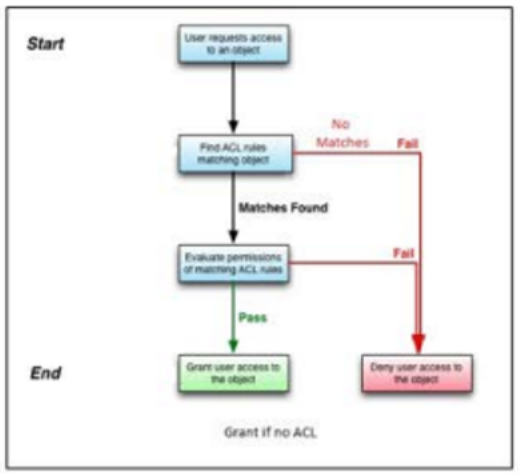
itil role is able to see:

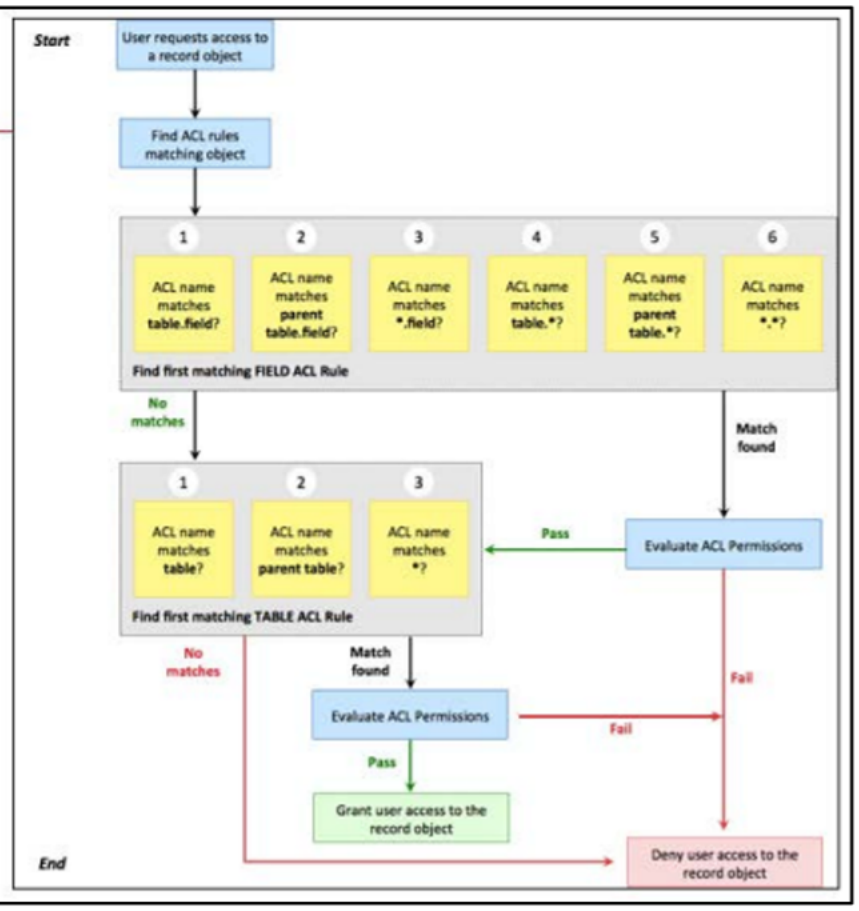
One of the real benefits of using the wildcard type of rule is to reduce the amount of rules required to control access, which also results in less required maintenance.

For example, taking the same series of rules above – it could accomplish the same end results without a wildcard rule. However, in order to do so, it would require six separate rules. That is a significant more amount of rules to manage, should changes be needed in the future.

Access Control Evaluation

One of the most powerful items behind Contextual Security is that it can apply either to a particular field or an entire row, using this process:



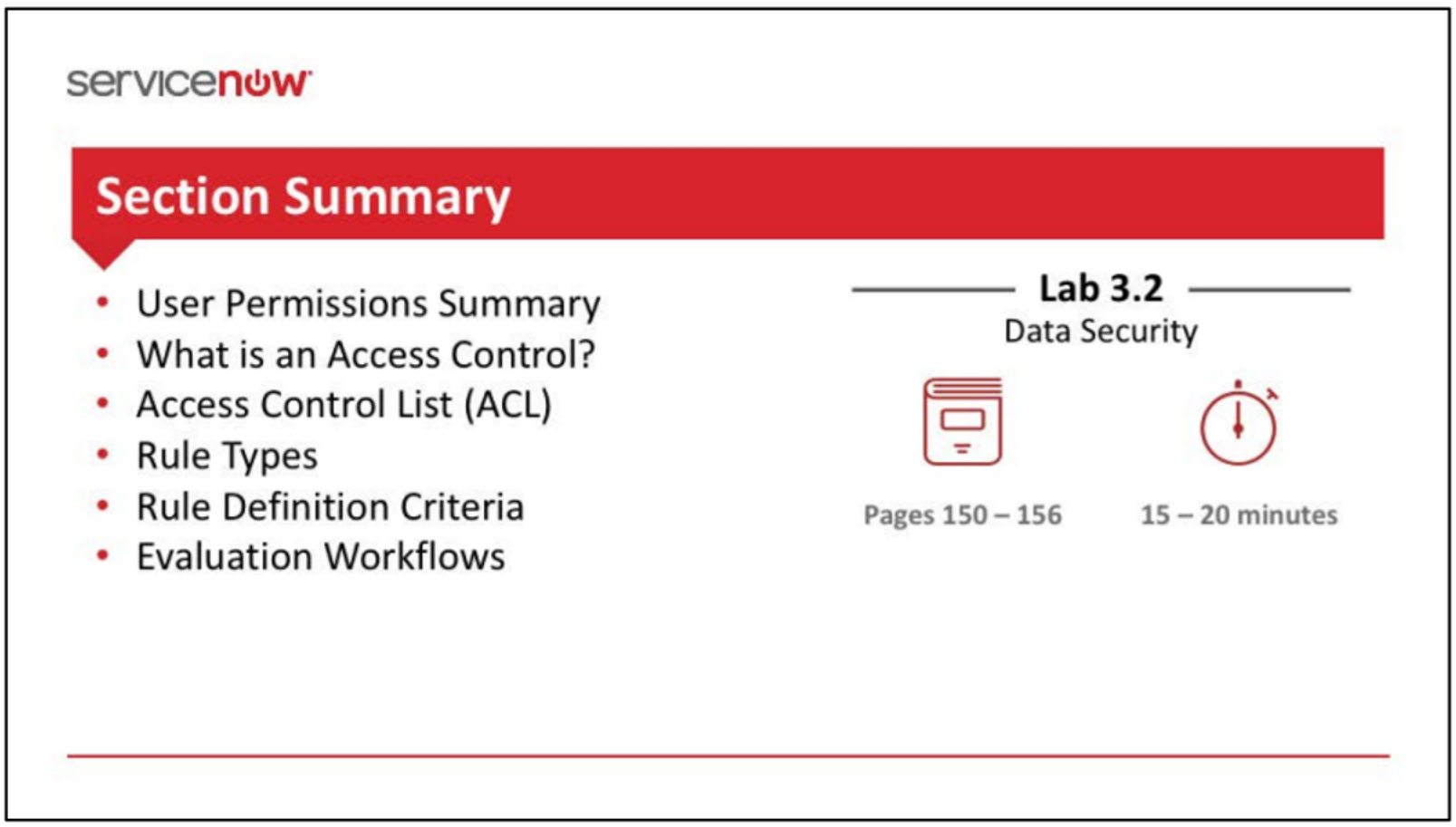


Record ACL rules are processed in order of most specific to most general: Match the object against **field ACL rules.** Match the object against **table ACL rules.**

This processing order ensures that users gain access to more specific objects before gaining access to less specific ones. A user must pass both field and table ACL rules in order to access a record object. If a user fails a field ACL rule but passes a table ACL rule, the user is denied access to the field described by the field ACL rule. If a user fails a table ACL rule, the user is denied access to all fields in the table even if the user previously passed a field ACL rule.

In most cases there is not an individual field ACL rule for every field in the table the users is trying to access. If no field ACL rule matches the record object, the user must pass the table ACL rule. Since the base platform includes wildcard table ACL rules that match every table, the user must always pass at least one table ACL rule. The base platform provides additional table ACL rules to control access to specific tables.

Table ACL rules are processed in the following order: Match the table name. For example, incident. Match the parent table name. For example, task. Match any table name (wildcard). For example, *. Just like with field ACL rules, the platform grants the user access to the record object secured by the ACL rule and stops searching for matching ACL rules the first time a user passes a table ACL rule's permissions. A user who passes the table ACL rule for Incident has access to all fields in the Incident table. A user who passes the table ACL rule for task has access to all fields in the Task table as well as the fields in extended tables. A user who passes the table ACL rule for any table has access to all fields in all tables.



Lab 3.2 – Data Security:

- Update a role
- Provide application menu and module access for a specified role
- Create an Access Control rule to grant data permission

LAB

Data Security

3.2

15 – 20 minutes

Lab Goal

Lab Dependency: Requires the completion of Lab 3.1.

This lab will show you how to do the following:

- Update a role
- Provide application menu and module access for a specified role
- Create an Access Control rule to grant data permissions

After much deliberation, Cloud Dimensions management have decided to limit access to Infinity device data to only those teams actively supporting the product.

The result is a requirement to restrict access to the Infinity Inventory application menu and modules, as well as controlling which fields the users can update.

As the system administrator, you will act upon the requirements provided by Cloud Dimensions management to successfully secure Infinity device data.

A. Modify the Infinity Table Role

1. **System Security > Users and Groups > Roles**.
2. Search for and open the **u_infinity_user** role record.

***NOTE**: This role was automatically created when the Infinity table was added to the database, along with the creation of four table Access Control rules. All of these are optional during the table creation process and can be avoided through settings.*

---

3. Change the name to **u_infinity_support**.
4. Type into the **Description** field: **Support role for the Infinity**.
5. From the **Form Context Menu**, select **Insert**.

***NOTE**: Two infinity roles now exist which will allow for more granular access provisioning in the future:*

B. Confirm and Modify Access to the Infinity Inventory Application

1. **System Definition > Application Menus**.
2. Use any method to find and open the **Infinity Inventory** application menu.
3. Notice the value next to the Roles field, **u_infinity_user**:

4. Click the **Edit User Roles** icon (pencil) next to the Roles field:

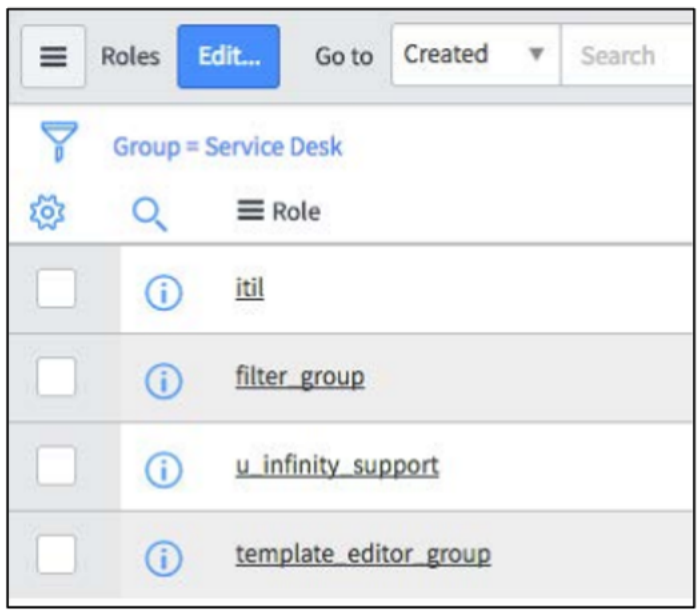
5. Add the **u_infinity_support** role from Available to Selected.
6. Click the **Done** button.
7. **Update** the Application Menu record.

Update the Infinity Support Group

1. **User Administration > Groups**.
2. Locate and open the **Service Desk** group record.

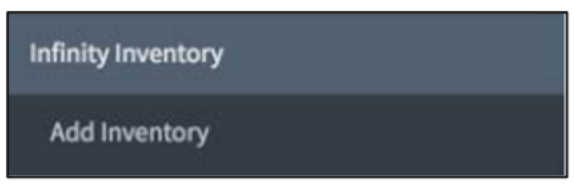
3. Select the **Roles** tab, if needed, then click **Edit…**
4. Add the **u_infinity_support** role to the Roles List.
5. **Save**.

   The Service Desk Roles list should look like this:



Test the Visibility Settings

1. Impersonate **Rita Center**.
2. Confirm **Rita Center** has been denied access to the **Infinity Inventory** application and to the modules it contains.
3. Impersonate **Kevin Edd**.
4. Expand the **Infinity Inventory** application menu to notice only **one** module displays:



   A module is missing from the application menu. Which one?

5. Click on the **Add Inventory** module.

   Although **Kevin Edd** has inherited the **u_infinity_support** role because he is a member of the **Infinity Support** group, which has access to the **Infinity Inventory** Application, the role does not currently have the rights for creating new data.

As a result, the Infinity New record page is blank when Kevin attempts to access it.

Access Control rules can be created to allow users with a certain role access to work with a table's data, but first let us update Rita and Kevin's group permissions so Service Desk and Infinity Customer Support group members are granted access to the application and all of its modules.

Update Roles and Groups

1. As System Administrator, **User Administration > Roles**.
2. Locate and open the **u_infinity_support** record.
3. Under the **Contains Roles** section, click **Edit…**
4. Add **u_infinity_user** to the **Contains Roles List**.
5. Click **Save**.

***NOTE**: You have added the u_infinity_user role and its permissions under the u_infinity_support role. This means all users with the u_infinity_support role now automatically inherit the permissions found with u_infinity_user.*

---

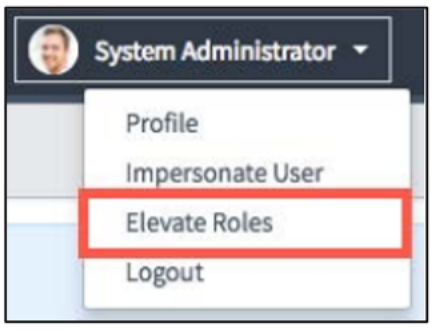
6. **User Administration > Groups**.
7. Open **Infinity Customer Support**.
8. Under **Roles**, click **Edit…**
9. Add the **u_infinity_user** role.
10. **Save**.
11. Impersonate **Rita Center**.
12. Confirm the **Infinity Inventory** application menu is accessible and its two modules display.

***NOTE**: Both Rita Center and Kevin Edd can now access both of the Infinity Inventory modules. Additionally, they can now create new records and update existing ones.*

C. Create an Access Control Rule

Now that the group permissions have been set and only the right groups of users can access the application, create an Access Control rule that allows *only* the System Administrator to update the **Asset tag** field on the Infinity record, and no other role.

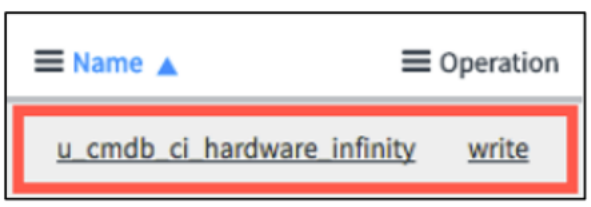
1. Impersonate the **System Administrator**.
2. Open the **User menu**.
3. Next, choose **Elevate Roles**:



4. Select the checkbox next to **security_admin**:

5. Click **OK**.
6. Notice the page refreshes and there is now an unlocked pad icon next to your name on the User menu:

7. **System Security > Access Control (ACL)**.
8. Filter the list of rules to find the four **u_cmdb_ci_hardware_infinity** rules.
9. Open the rule with the **write** operation:

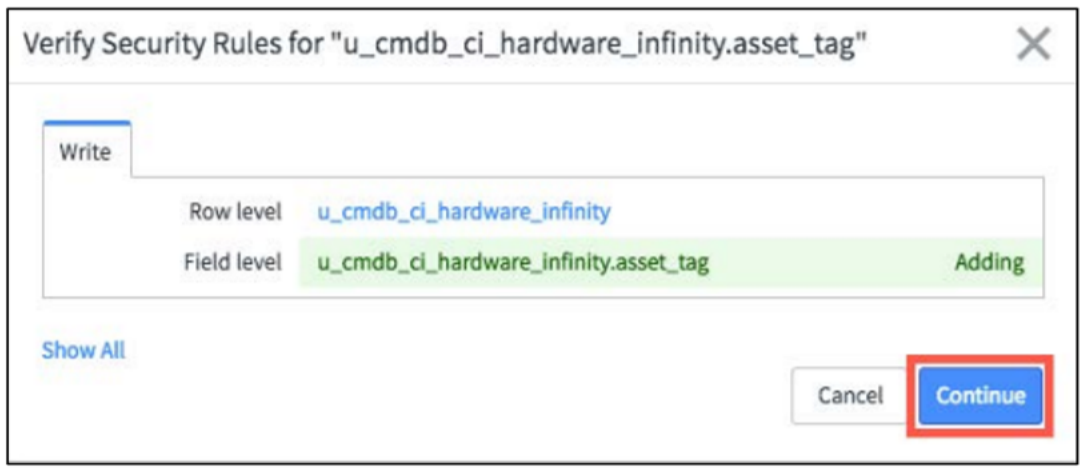


10. From the **Name** field, select the drop down next to the drop down with the value **Infinity [u_cmdb_ci_hardware_infinity]**:

11. From the drop down, select **Asset tag**.

12. Open the **Form Context Menu** and select **Insert and Stay**.

13. From the **Verify Security Rules** window, click **Continue**:



14. Scroll down to the **Requires role** section.

15. Double click on ***Insert a new row…***

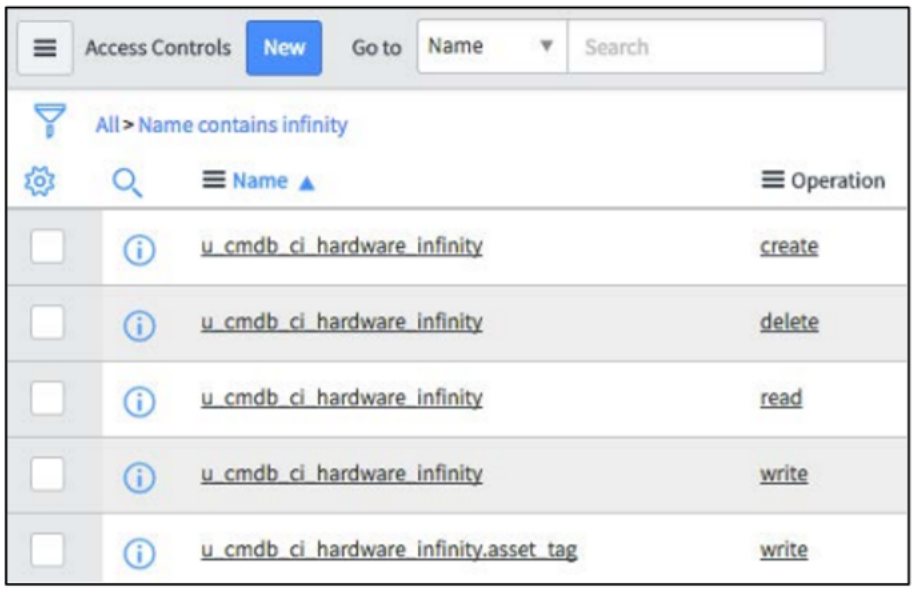
16. Type **admin** and click the save icon to add the role.

17. **Update**.

***NOTE**: The **u_cmdb_ci_hardware_infinity.asset_tag** rule was created and added to the Access Control List.*

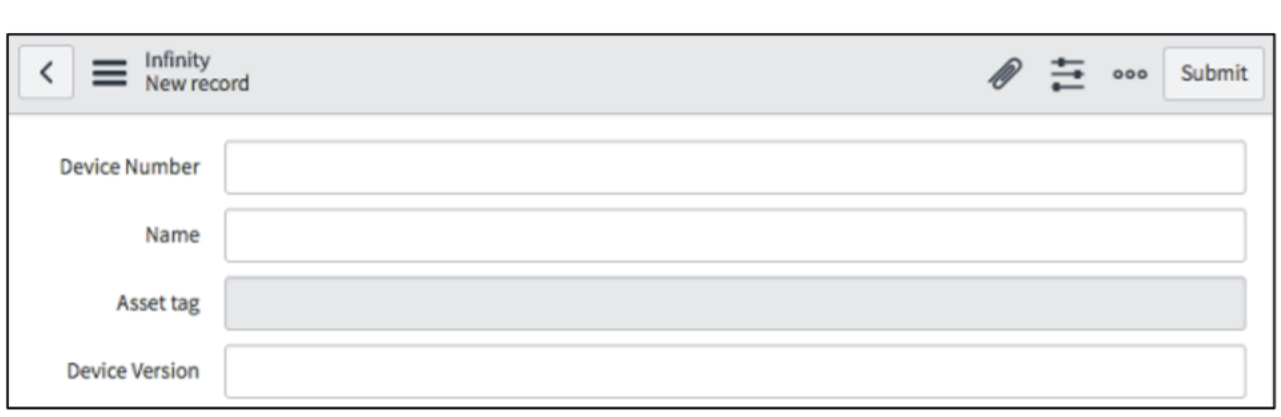
LAB VERIFICATION

Access Control List – Infinity Rules



Test New Security Settings

1. As Rita Center, **Infinity Inventory > Add Inventory**:



NOTE: *All fields but* ***Asset tag*** *and* ***Email*** *should be editable. These same fields would also be uneditable for existing records, for all users in Service Desk or Infinity Customer Support groups. As system administrator, all fields on the form are editable.*

With this lab, application and module access was adjusted, and an Access Control rule was created to limit permissions to table data.

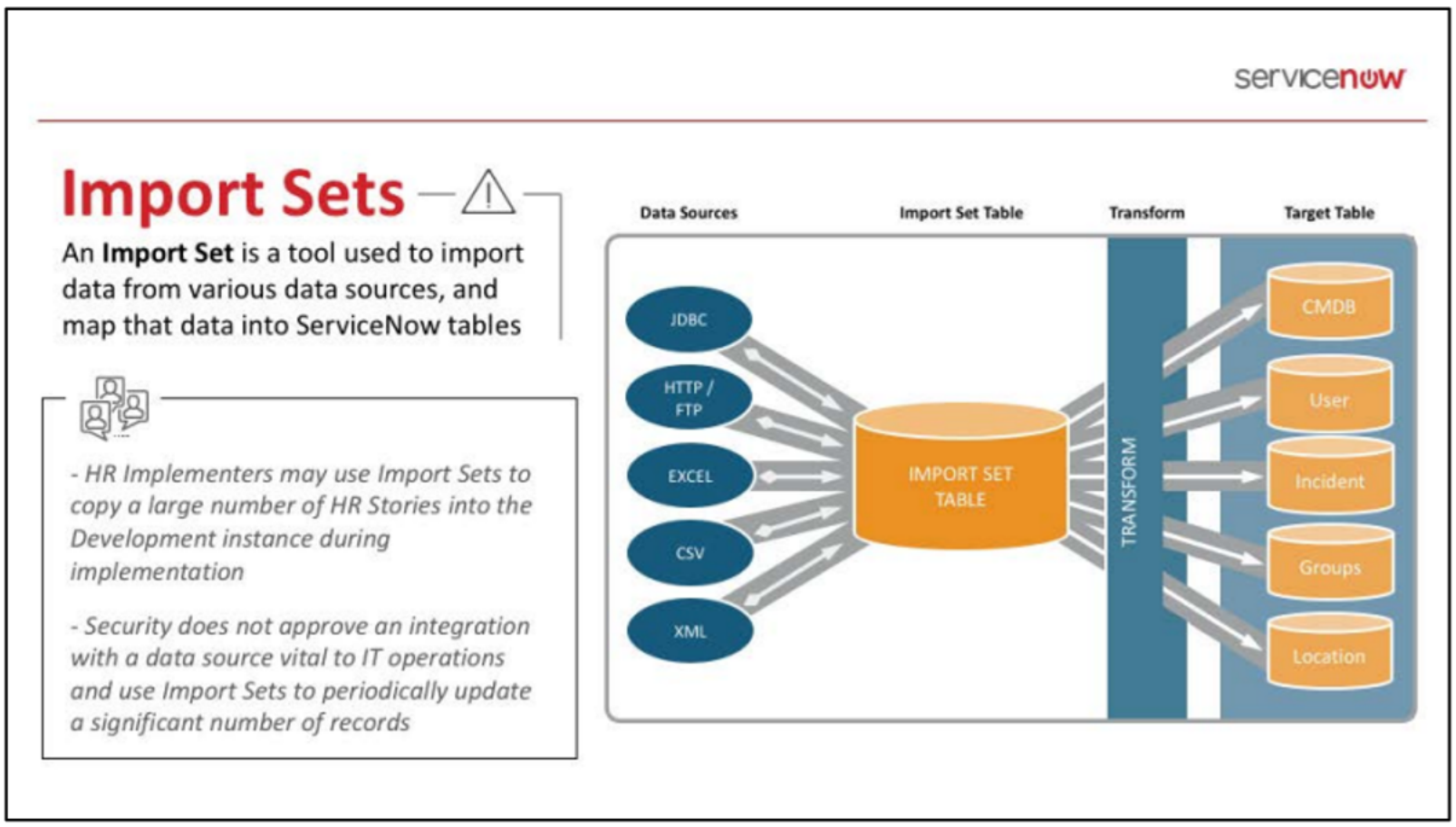
This is not an easy topic nor lab, so well done completing it!

Module 3 – 3.3 Import Sets

servicenow

Objectives

- What are Import Sets?
- Import Set Table
- Transform Maps
- Coalesce Fields



Import Sets provide a mechanism to pull data into ServiceNow. Import Sets store data in Import Set tables. Any user logged in with the **admin** or **import_admin** role can manage all aspects of Import Sets.

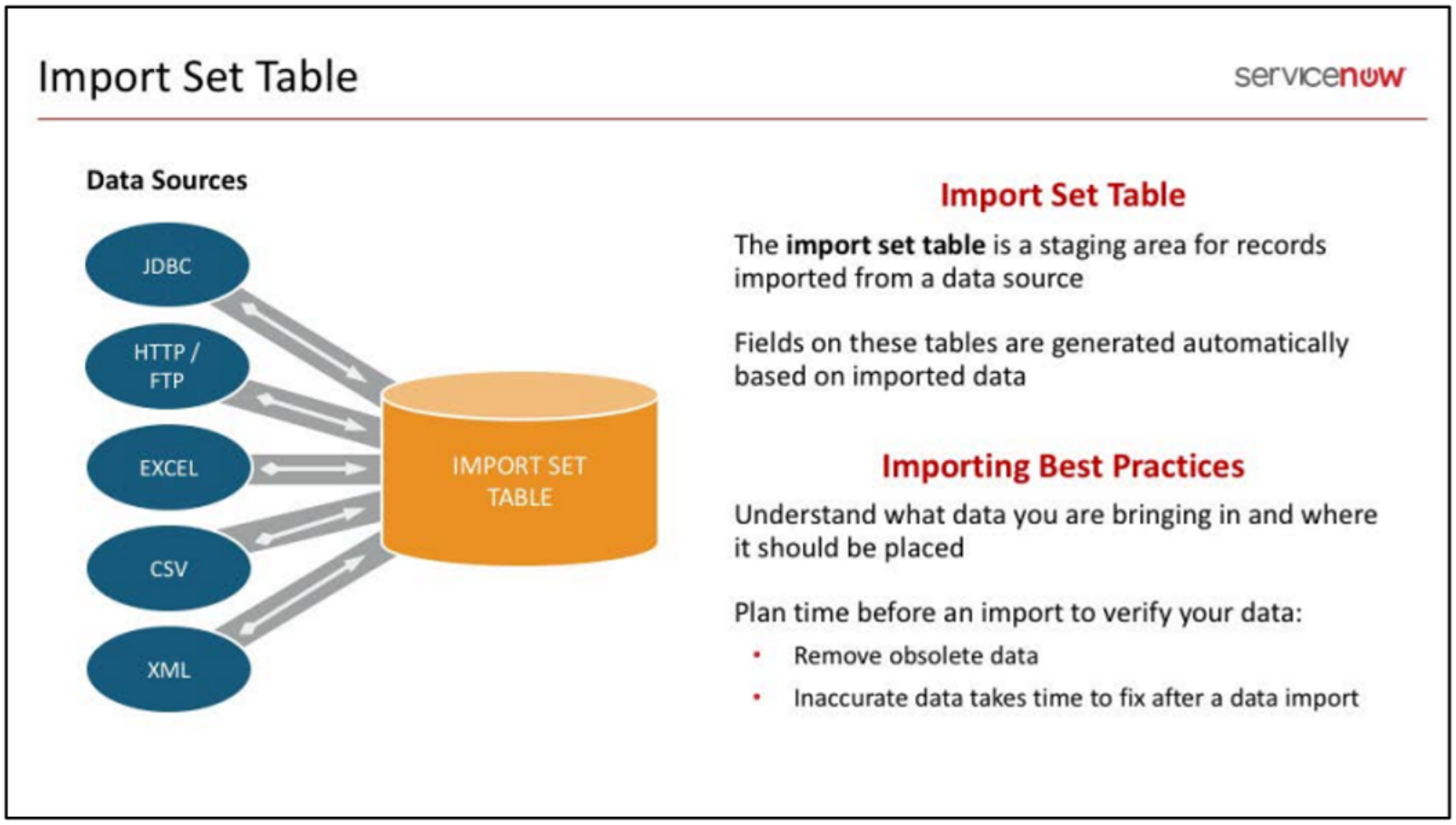
Data Sources are records in ServiceNow that contain information regarding an Import Set data source. You can import a file from a local source (i.e. XML, CSV, Excel) or from a network server by providing a path and authentication information. A data source can come from a file, an LDAP connection, or a JDBC connection.

The **Import Set Table** acts as a staging area for records imported from a data source.

Transform Maps provide a guide for moving data from Import Set tables to "Target" tables; field mapping provides direct field-to-field data moves.

A Transform Map is a set of field maps that determine the relationships between fields in an Import Set and fields in an existing ServiceNow table (such as Incidents or Users). Once defined, existing Transform Maps can be reused for mapping data from an Import Set to a ServiceNow table. The Transform Map Module enables an administrator to define destinations for imported data on any ServiceNow tables. Transform mapping can be as simple as dragging and dropping to specify linking between source fields on an Import Set table and destination fields on any ServiceNow table.

The **Target Table** is an existing table in where the data will be placed, post-transformation.



Before importing any data, it is important to understand what data you are bringing in and where that data should go.

You should verify the data before you import it since bad data will complicate things later in the import and transform processes. Extra time spent planning and examining data before import will save time and potential problems later.

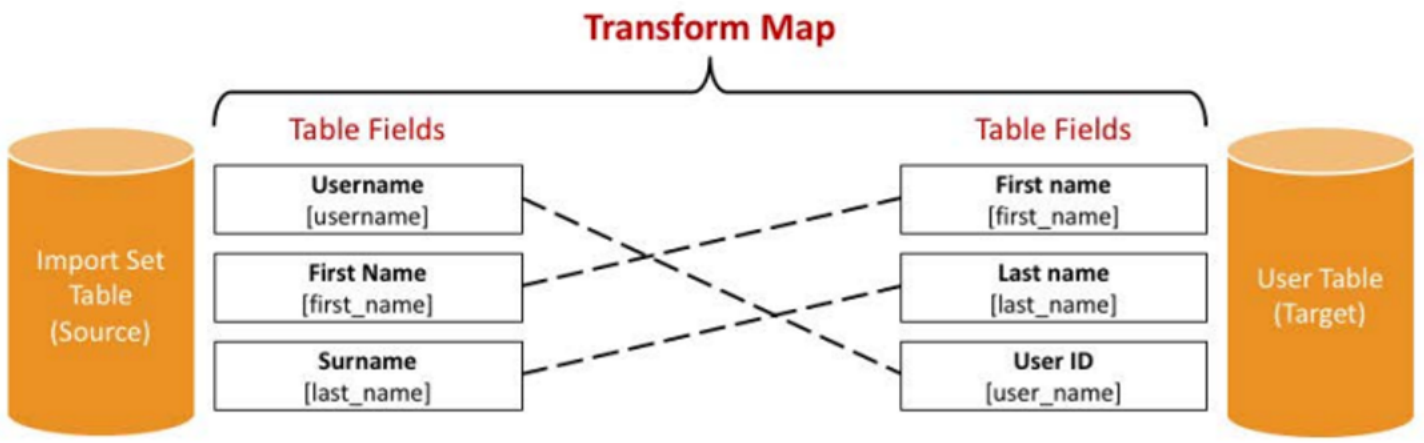
Data should not be imported in extremely large chunks. Creating an extremely large Import Set can cause extensive delays. The imported file label is used to determine the name for the Import Set table that data will be loaded into.

NOTE: It is also possible to choose an existing Import Set table to use for loading data from the same source, or data that has the same field/column designations. When an existing Import Set table is chosen, the table fields are added when the incoming source of data contains fields/columns that do not exist.

Transform Maps

servicenow

A **transform map** is a set of field maps that define the relationships between fields in an import set table and fields on a target table



Transform mapping is flexible; the specification can be as simple as having the application auto-match field names from source and destination, or mapping can use advanced logic and leverage the full power of the ServiceNow scripting environment. A single Import Set field can also be mapped to multiple fields on a Target table. Any table is a potential destination for transformation of an Import Set, and any field within a table can serve as a potential destination for transformation from a field within an Import Set.

Select the table where you want transformed data to be placed. You can select only tables within the currently selected application scope, the global scope, or tables that grant write access to other applications. Name and Source table are set based on the label which was assigned to the Import Set. It is necessary to assign a target table into which the data can be transferred.

Automatic Mapping Utility: The simplest mapping method is where all of the field names of the Import Set match the name of the fields on the Target table where the data will be transformed. In this case, simply click Auto Map Matching Fields in the related list in the Table Transform Maps form and confirm proper matching. If there are any discrepancies in terms of how fields were automatically matched, these can easily be corrected using the Mapping Assist utility. When all fields are matched properly, click Transform in the related lists to begin transforming data onto the destination table.

Mapping Assist Utility: The Mapping Assist utility provides a visually intuitive environment for specifying mapping between Import Set fields and Target table fields. With the Mapping Assist utility it is possible to map a single source field (field on an Import Set table) to multiple destination fields (fields on a Target table).

Coalesce Fields

servicenow

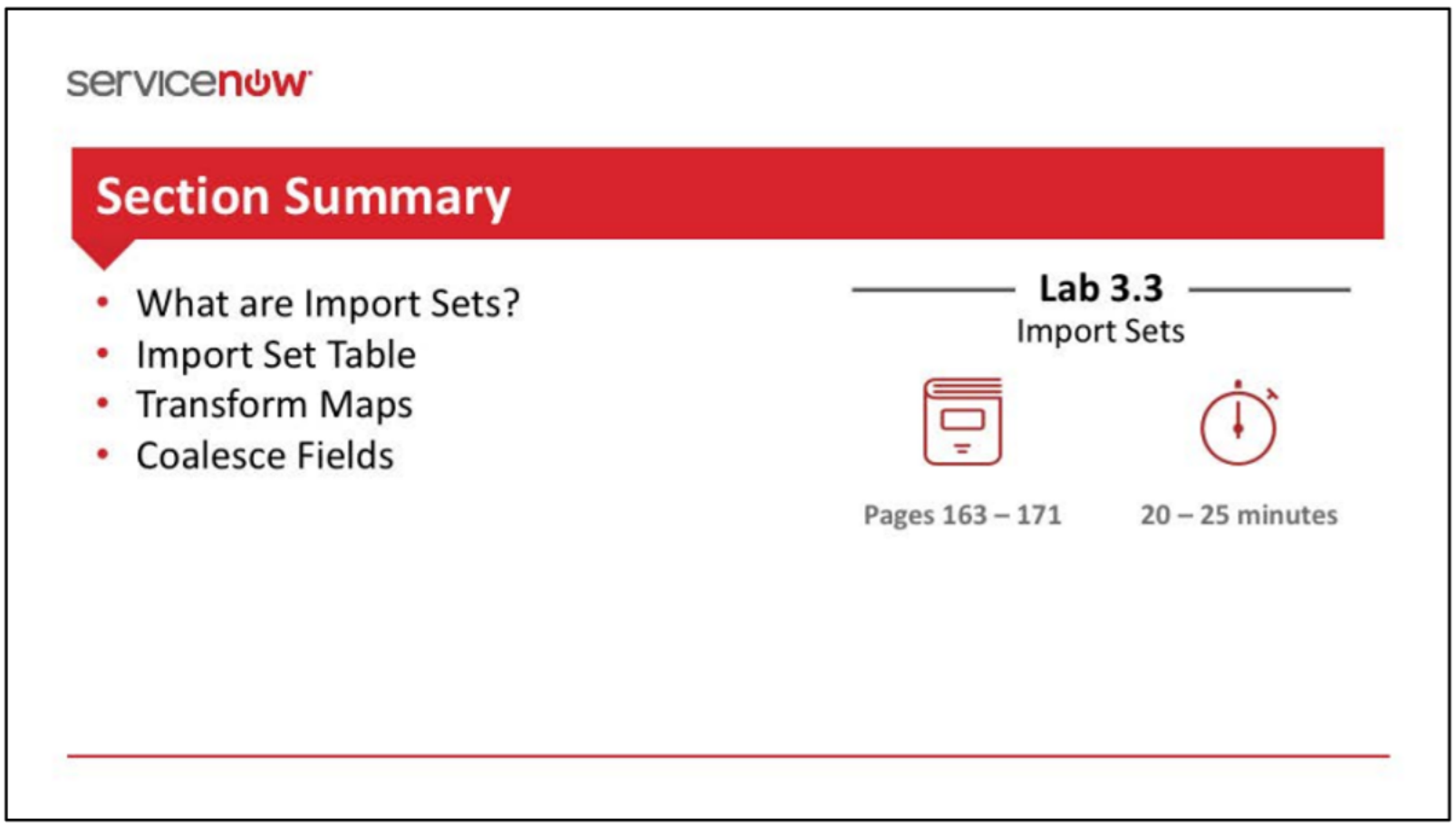
Coalescing a field (or set of fields) means the field will be used as a unique key during imports

If a match is found using the coalesce field(s), the existing record will be updated with the information being imported

If a match is not found using the coalesce field(s), then a new record will be inserted into the database

There are several possible configurations you can use to coalesce data in Import Sets:

- **No coalesce**: If no coalesce is defined, all imported rows are treated as new records. No existing records are updated. If the import is executed again, duplicate records will be created.
- **Single-field coalesce:** You can coalesce on a single field to update an existing record. If a target table record exists with the same value in the coalesce field as the staging table record, the target table record is updated using the Import Set record values.
- **Multiple-field coalesce:** You can coalesce on multiple fields to update an existing record. If a target table record exists with the same values in all coalesce fields as the staging table record, the target table record is updated using the staging table record values. All coalesce field values between the target and staging tables must match to coalesce with multiple fields.
- **Conditional coalesce:** You can use a script to determine if a staging table row should coalesce to a target record. Most conditional coalesce scripts are defined in the source script field of a field map for the sys_id field. To update a target record using the staging table record values, the script must return the sys_id of the target table record.



Lab 3.3 – Import Sets:

- Modify a list layout as preparation
- Create an Import Set Table and Transform Map
- Transform multiple imports
- Clean up import data

LAB

Import Sets

3.3

20 – 25 minutes

Lab Goal

Lab Dependency: Requires the completion of Lab 3.1.

This lab will show you how to do the following:

- Modify a list layout as preparation
- Create an Import Set Table and Transform Map
- Transform multiple imports
- Clean up import data

In this lab, you will use Import Sets to load data that has been collected outside of ServiceNow into the Infinity table.

The data will represent asset registration by Cloud Dimensions employees and partners, and include information about the user and their registered Infinity device.

This lab has three parts:

1. An initial load, with all of the foundational work required.
2. An incremental load.
3. A data cleanup.

In the **initial load**, you will:

- Gather Excel data files
- Organize a list layout for the Infinity table
- Create a new Import Set by importing data from an Excel spreadsheet to a staging table, then validate the data
- Create a Transform Map based on the staging table: use automapping and mapping assist to establish the mapping between the source and the target tables
- Complete the transform and verify the resulting data

In the **incremental load**, you will:

- Upload a second Excel spreadsheet, reusing the staging table and Transform Map
- Identify a coalesce (key) field to ensure that existing records are updated from the new imported data and not duplicated

In the **cleanup**, you will clean up the Import Set Table's data.

A. Part 1: Initial Load

Gather Excel (.xlsx) Data for Lab

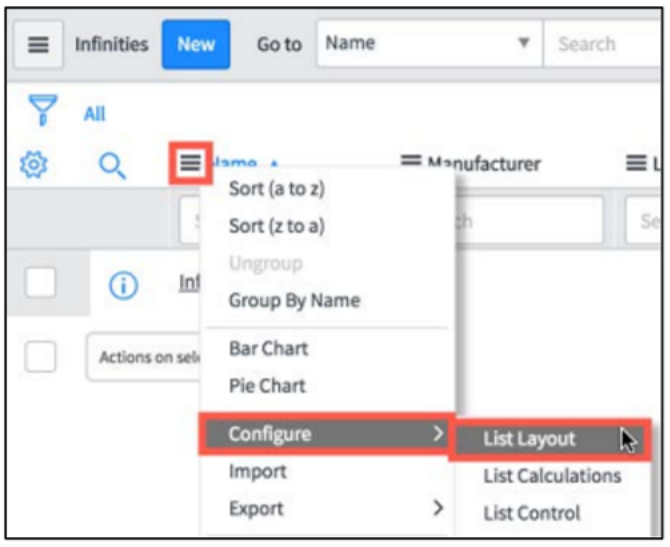
Before you begin, download the two Infinity MS Excel Knowledge files: **infinity-data.xlsx** and **infinity-updates.xlsx**, and save them to your desktop.

Open both the **infinity-data.xlsx** and **infinity-updates.xlsx** files to review their contents; columns, fields, and data types.

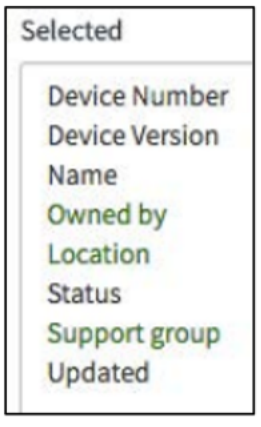
Modify the Infinity All Devices List Layout

1. As the System Administrator, navigate to **Infinity Inventory > All Devices**.

2. Open the **Column Context Menu** from any field, then select **Configure > List Layout**:

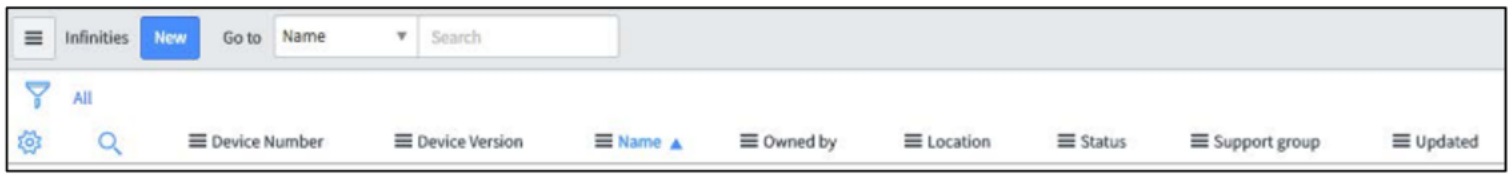


3. Using the slushbucket, organize the **Selected** field list to include the following fields in the order seen here:



4. Click **Save**.

5. Your **All Devices** list header should look like this:



Create a New Import Set

1. **System Import Sets > Load Data**.

   ***NOTE**: The following steps (process) can be completed by any user with the role **import_admin** or **import_set_loader** and **import_transformer**.*

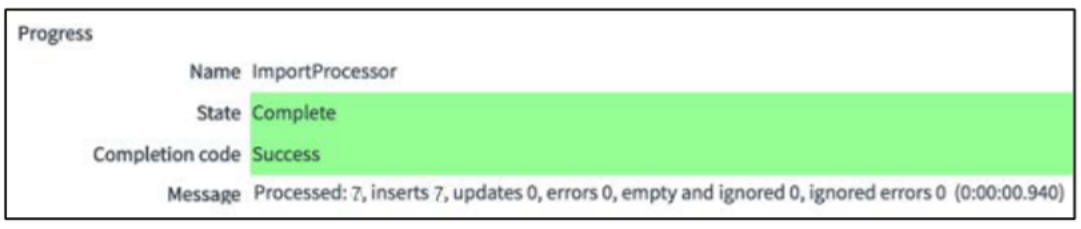
   ---

2. Populate the Import Set form:

   Import set table: **Create table** (auto selected)
   Label: **Infinity Imports**
   Name: **u_infinity_imports** (automatically populates)
   Source of the import: **File** (auto selected)
   File: Choose file, then select **infinity-data.xlsx**

3. Click **Submit**.

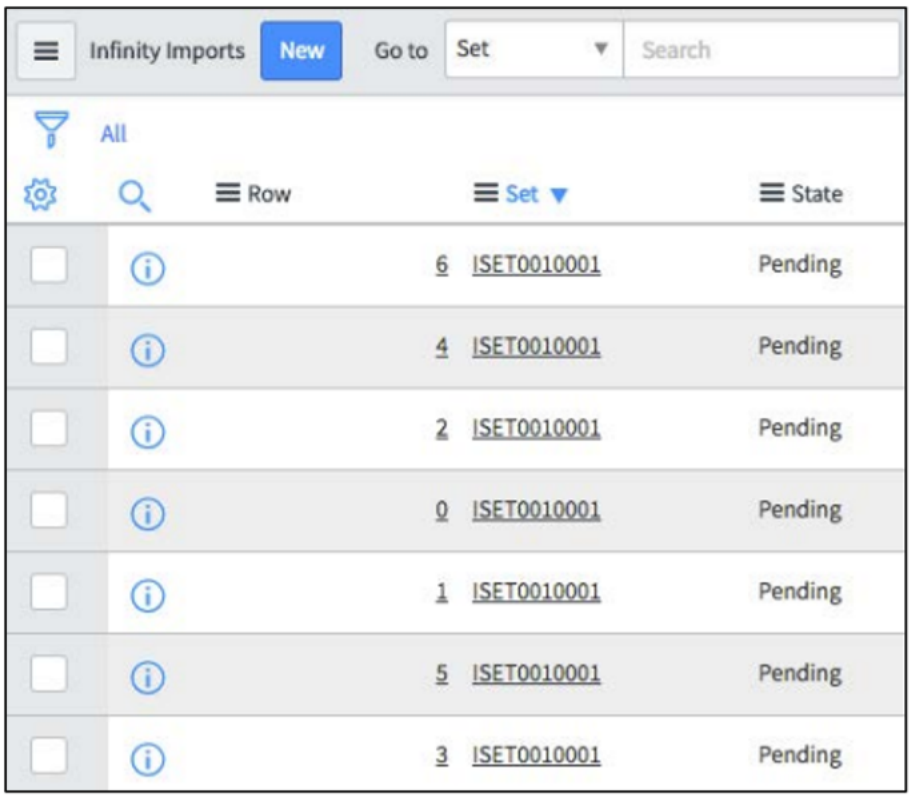
4. Review the Progress screen, you should see **7 inserts**:



Validate Data in Import Set

1. To verify the data in the new Import Set, in the **Next Steps…** section of the **Progress** screen, click the **Loaded data** link:

2. Confirm the seven **Infinity Imports** records loaded correctly:



NOTE: *You may notice a different order.*

Create Transform Map

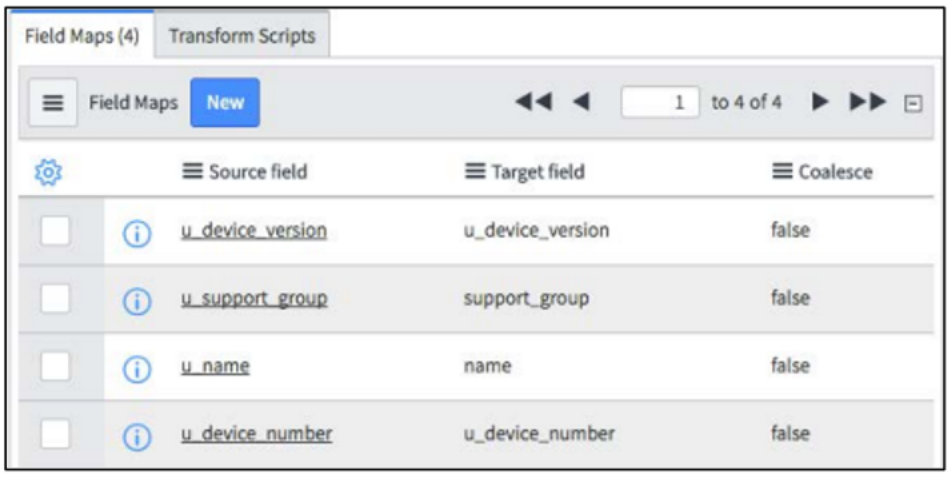
1. **System Import Sets > Create Transform Map**.
2. Fill out the form as shown:

   Name: **Infinity Assets**
   Source table: **Infinity Imports [u_infinity_imports]**
   Target table: **Infinity [u_cmdb_ci_hardware_infinity]**

3. **Save**.
4. Scroll to Related Links, then click **Auto Map Matching Fields**.

NOTE*: A verification message displays at the top of your form.*

5. Verify that four fields auto mapped: **Name**, **Device Number**, **Device Version**, and **Support Group**:



***NOTE**: Your field order may be different.*

---

6. In the Related Links list, click **Mapping Assist**.

7. From the Source: Infinity Imports staging table, move **Device Owner** and **Owner Location** into the Field Map.

8. From the Target: Infinity table, move the **Owned by**, and **Location** into the Field Map:

Field Map	
Device Version	Device Version
Support Group	Support group
Name	Name
Device Number	Device Number
Device Owner	Owned by
Owner Location	Location

9. Click the **Save** button, then verify that there are now six mapped fields: **Name**, **Device Number**, **Device Version**, **Support Group**, **Owned by**, and **Location**.

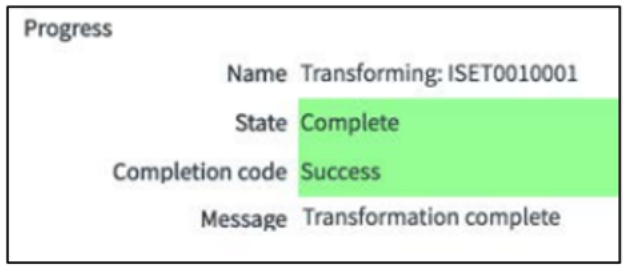
Run the Transform

1. In Related Links, click **Transform**.

2. Verify the **Infinity Assets – u_cmdb_ci_hardware_infinity** map is selected.

3. Click the **Transform** button.

4. The **Progress** screen displays the transformation confirmation messages:



Verify Infinity Inventory Import

1. **Infinity Inventory > All Devices**.

2. Your screen should show **seven total records**.

B. Part 2: Incremental Load

Import Additional Data using an Existing Import Set Table

You will now work with the second spreadsheet you downloaded at the start of this lab: **infinity-updates.xlsx**. For the incremental load, you will use the same Import Set Table (Infinity Imports) and Transform Map as this spreadsheet is in the same format as the first.

Upload the Data

1. **System Import Sets > Load Data**.

2. For **Import set table**, select the **Existing table** radio button.

3. Fill out the form as shown:

   Import set table: **Infinity Imports [u_infinity_imports]**
   Source of the import: **File** (auto selected)
   File: Choose file, then select **infinity-updates.xlsx**

4. Click **Submit**.

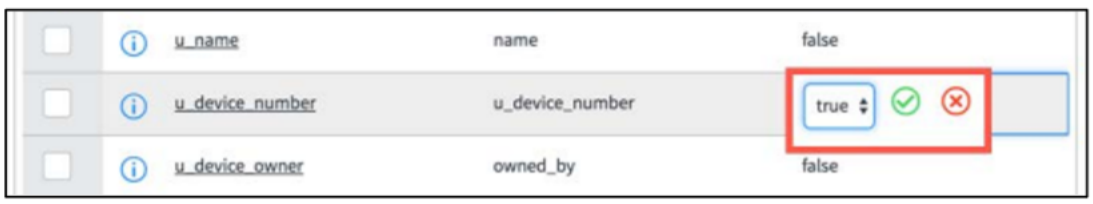
5. You should see **19 inserts**.

   The import of the Infinity data to the staging table is complete, but you are not ready to run the transform yet because you need to add a coalesce. Rather, tell the system what the key field is to ensure that existing records are updated rather than adding duplicate records by the imported data.

Define the Coalesce Field

1. **System Import Sets > Administration > Transform Maps**.
2. Open the **Infinity Assets** Transform Map.
3. Scroll to the **Field Maps** Related List.

   Set the device number (**u_device_number**) field as the coalesce, as it contains a unique value. If a match is found for this field, the record will be updated instead of creating a new record.
4. In the **u_device_number** row, in the **Coalesce** column, double-click the word **false**, then select **true** from the list:



5. Click **Save** (green checkmark) to update the value to true.

***NOTE**: A system message displays at the top of the form.*

Prepare and Run the Transform

1. Under the Related Lists section, click the **Transform** link.
2. In the Selected maps box, verify you are using the correct Transform Map: **Infinity Assets – u_cmdb_ci_hardware_infinity**
3. Click the **Transform** button.
4. Verify the transformation complete message.

Verify Infinity Inventory Import Updates

1. **Infinity Inventory > All Devices**.
2. There are **19 total records** that show.

   Notice that some of the records have an updated time from the **initial upload** (infinity-data.xlsx) and some have an updated time from the **incremental upload** (infinity-updates.xlsx).

Also notice that in the **infinity-updates.xlsx** spreadsheet, in record **CDE0100102**, there was no **Owner Location** information. Compare this to the same record in the **infinity-data.xlsx** spreadsheet, which had Owner Location information.

If you had **Copy empty fields** checked and active in the Transform Map, it would have removed the location data for this record. Since Copy empty fields was not active, the data from the original import still remains in the table.

C. Part 3: Clean Up Import Set Tables

1. **System Import Sets > Import Set Tables > Cleanup**.
2. Add the **Infinity Imports [u_infinity_imports]** table to the **Delete these tables** box.
3. The checkbox for **Delete related transform maps** should be unchecked.
4. The checkbox for **Delete data only (preserve table structure)** should be selected.

***NOTE**: This will remove the data collected in the Infinity Imports staging table.*

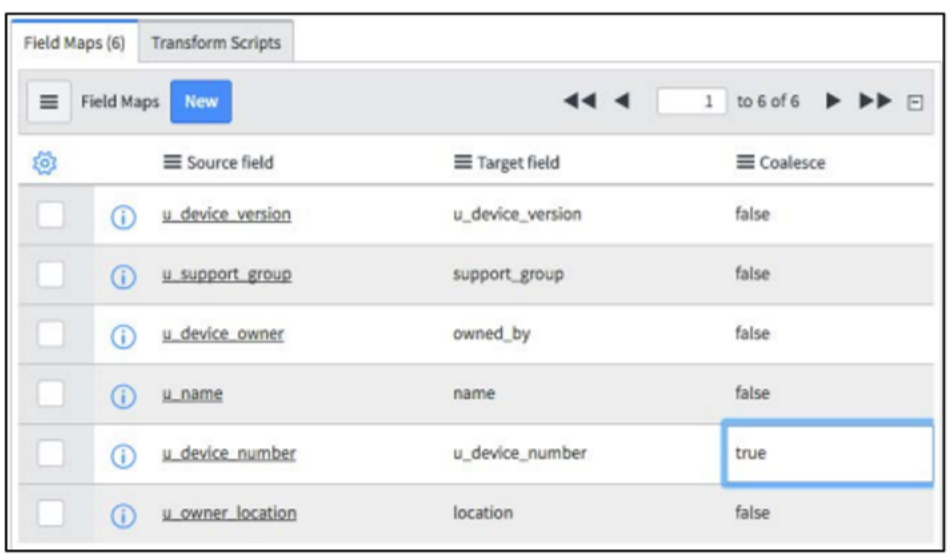
---

5. Click the **Cleanup** button – you should see a Cleanup completed verification message and actions taken displayed in an Import Log.

***NOTE**: If you wanted to delete the Import Set table and any reference to it, including the Infinity Assets Transform Map, you would have checked the **Delete related transform maps** checkbox.*

LAB VERIFICATION

Transform Map



Imported Infinity Inventory Data

Infinities | New | Go to | Name | Search | 1 to 19 of 19

All

Device Number	Device Version	Name ▲	Owned by	Location	Status	Support group	Updated
CDE0100107	AP01		Megan Burke	3121 High Point Road, Greensboro, NC	Installed		2018-02-27 18:30:24
CDE0100106	AP01	Infinity Alpha Prototype	Kevin Edd	153 South Sierra Avenue, Solana Beach, CA	Installed	Service Desk	2018-02-27 18:26:51
CDE0100105	AP01	Infinity Alpha Prototype	Winnie Reich	9249 Cicero Avenue, Oak Lawn, IL	Installed	Service Desk	2018-02-27 18:26:51
CDE0100102	TD02	Infinity Testing Device	Kevin Edd	153 South Sierra Avenue, Solana Beach, CA	Installed	Service Desk	2018-02-27 18:26:51
CDE0100101	TD01	Infinity Testing Device	Kevin Edd	153 South Sierra Avenue, Solana Beach, CA	Installed	Service Desk	2018-02-27 18:26:51
CDE0100100	TD01	Infinity Testing Device	Kevin Edd	153 South Sierra Avenue, Solana Beach, CA	Installed	Service Desk	2018-02-27 18:30:24
PAR0100099	TD01	Infinity Testing Device	Jon Floyd	10335 Lee Hwy, Fairfax, VA	Installed	Infinity Customer Support	2018-02-27 18:26:51
CDE0100099	TD01	Infinity Testing Device	Winnie Reich	9249 Cicero Avenue, Oak Lawn, IL	Installed	Service Desk	2018-02-27 18:26:51

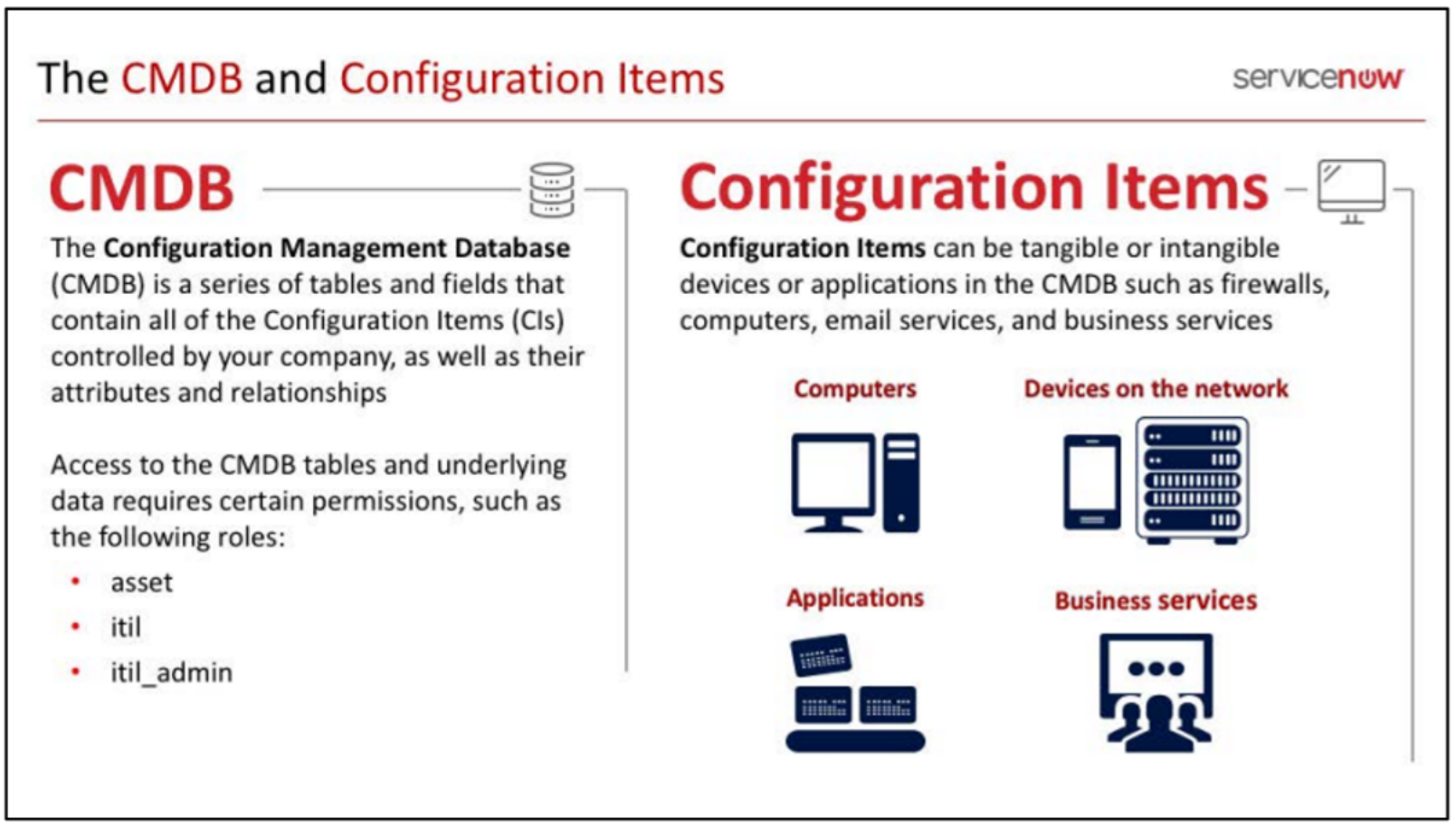
Congratulations, you have completed the Import Sets lab!

Module 3 – 3.4 CMDB

servicenow

Objectives

- What is a CMDB?
- What are Configuration Items (CIs)?
- Using the CMDB
- Relationship Editor
- Dependency View
- Implementation Considerations



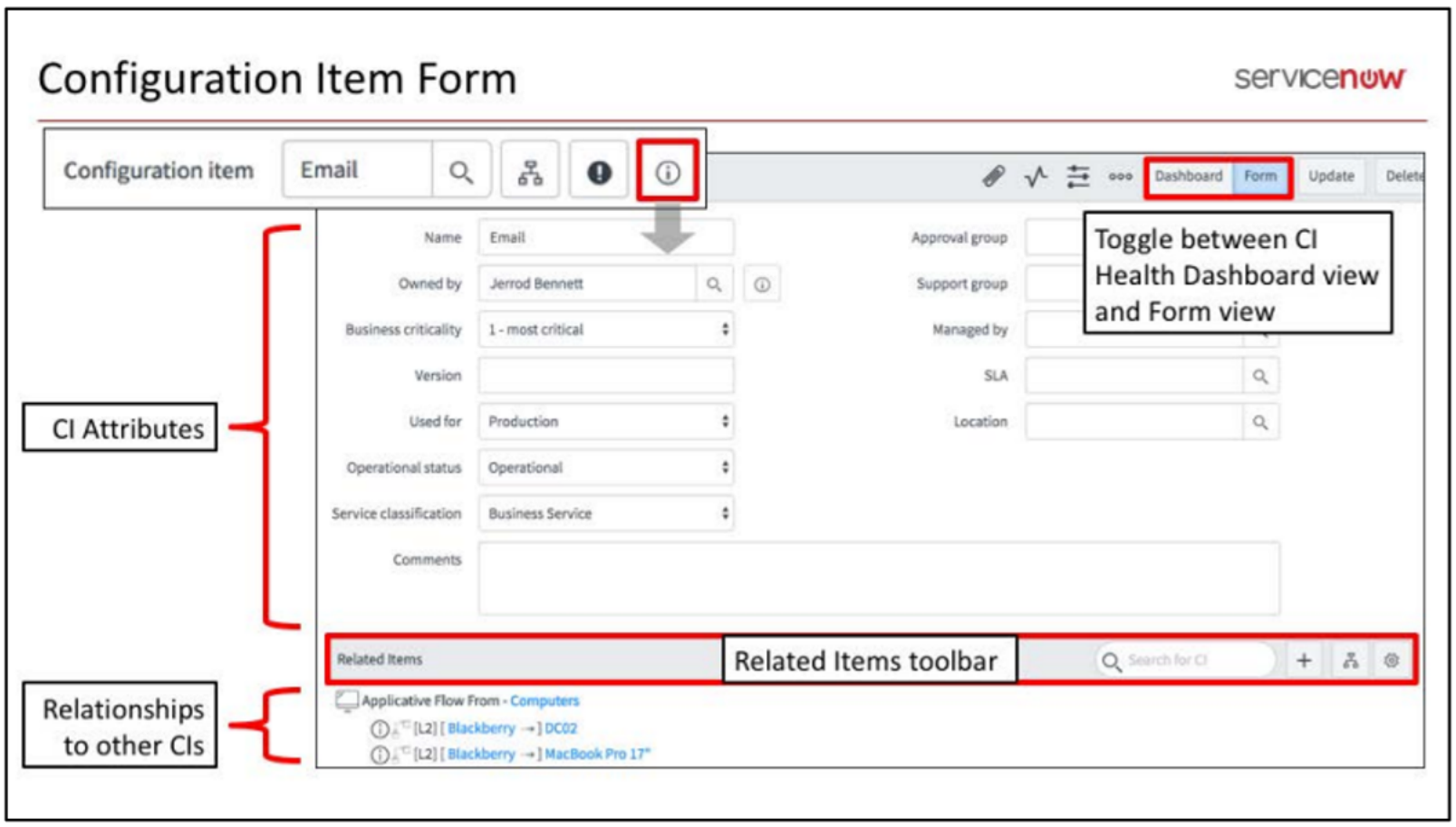
The Configuration Application provides core functionality for the Configuration Management Database, including modules for hardware and configuration items. This functionality is part of the CMDB plugin, which is activated in a base install.

ServiceNow provides a logical model of your company infrastructure by identifying, controlling, maintaining, and verifying the CIs that exist.

ServiceNow's CMDB, in contrast to a static list, tracks not only the CIs within your platform, but also the relationships between those items.

Two key CMDB tables are **Configuration Item [cmdb_ci]** which contains CI data, and **CI Relationship [cmdb_rel_ci]** which contains CI relationship data.

A Configuration Item is any component that needs to be managed in order to deliver Services. CIs typically include business services and their underlying components, such as business applications and hardware.



Click the reference icon to the right of the Configuration item field to be redirected to the selected CIs record in ServiceNow.

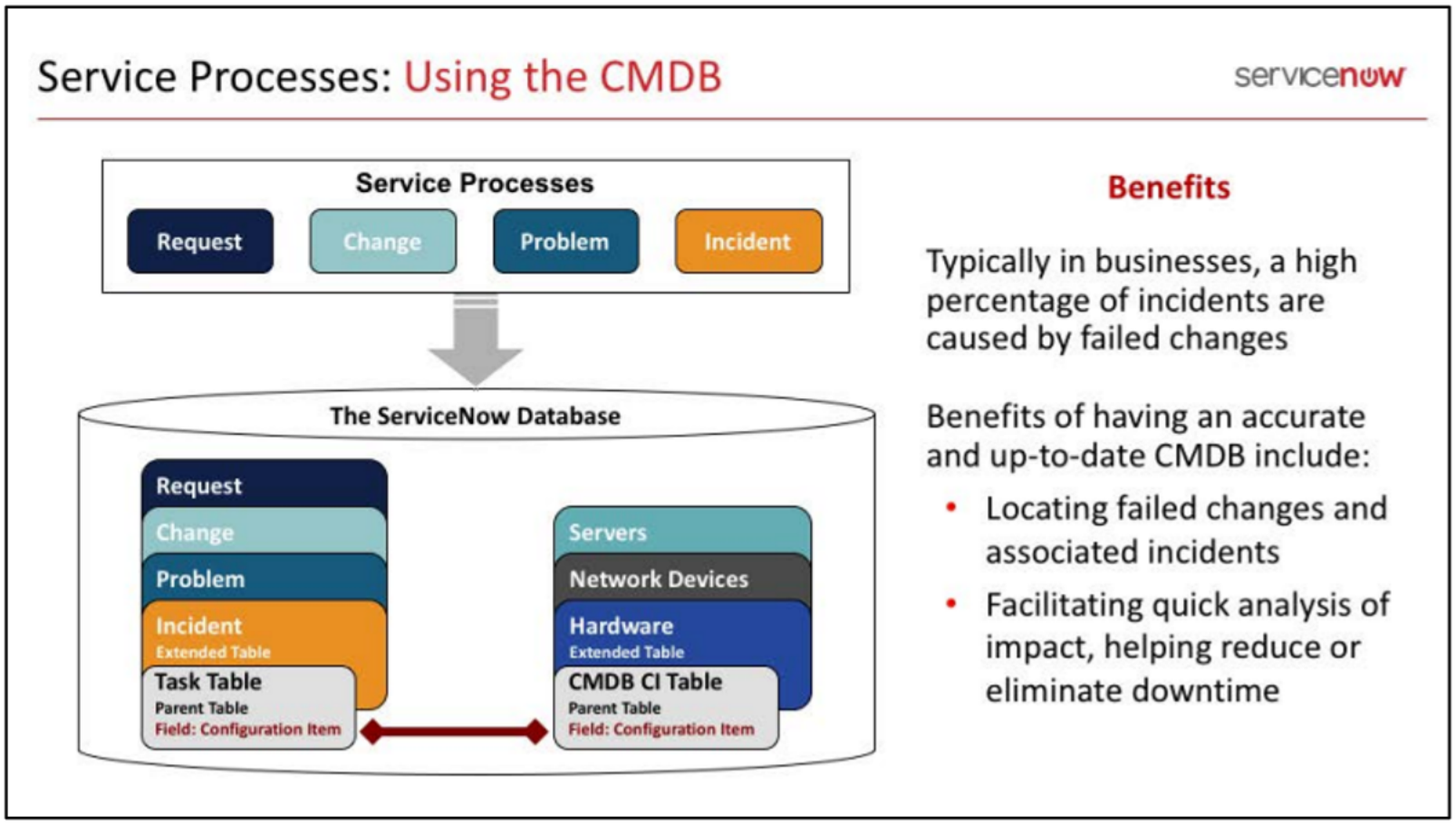
A CI record contains all of the relevant attribute data about an item such as name, version, descriptions, ownership, etc., which are documented in fields on the form.

Toggle between the CI Health Dashboard view and the Form view using the options in the form title bar.

The form also contains information about the relationships between CIs. Search for CIs, add new relationships, view the CI Map, or adjust the relationship view settings using the options in the Related Items toolbar.

ServiceNow relationship rules use separate tables to define the relationships between specific CI **base classes** and **dependent classes**. When you extend a table in the CMDB, you must create a new relationship rule in **Configuration > Suggested Relationships**.

You can view relationships between the current CI and other CIs. An advanced feature is the Related List in CI records which displays additional components contained by that particular CI.



The Configuration Management Database is a repository of information related to all of the components of an information system. Although repositories similar to CMDBs have been used for years in IT, the origin of the CMDB stems from the Information Technology Infrastructure Library (ITIL). CMDBs help monitor and discover what system components are needed for effective and efficient business processes and IT service management.

All service management processes relate to and involve the CMDB.

For example: someone calls with an issue, and you want to do a root cause analysis, the CMDB gives you insights to effectively troubleshoot.

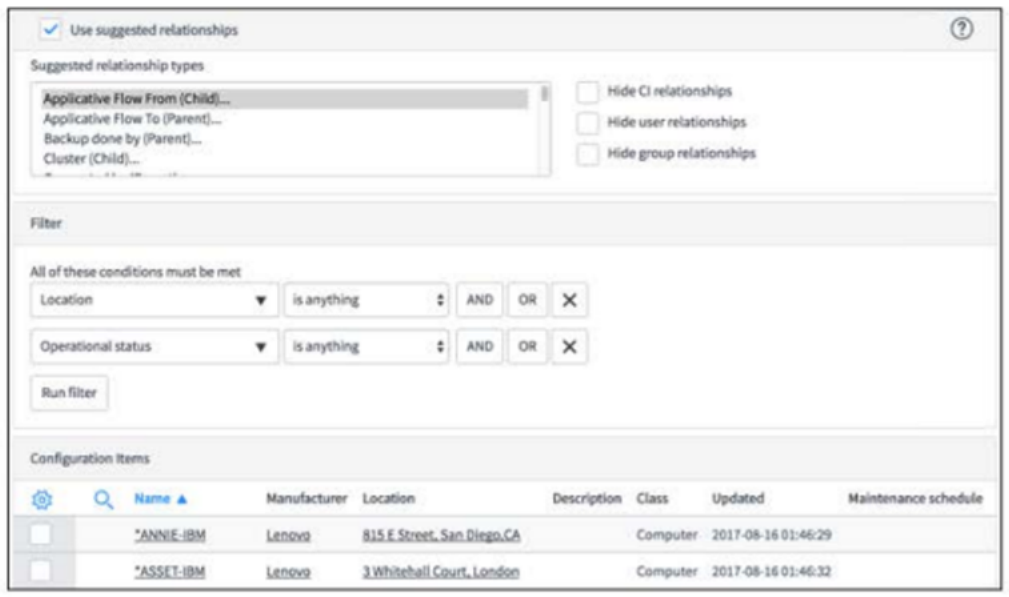
CI Relationship Editor

servicenow

The **CI relationship editor** uses a concept of suggested relationships to help users see reasonable relationships between configuration items

Examples:

- A database *runs on* a server
- A rack *provides power for* a server



Use the Configuration Item relationship editor to create CI relationships. Use the CI relationship editor to create CI relationships. It is access from the Related Items toolbar on a Configuration Item form.

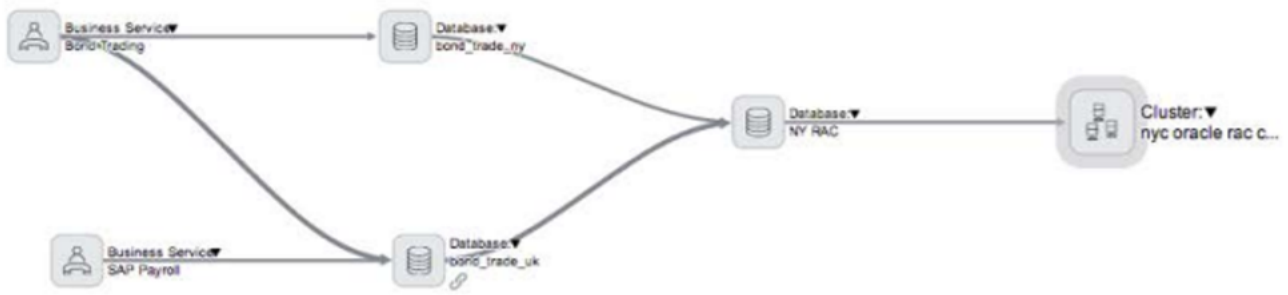
When you use the relationship editor, the CI from which the editor was launched is designated as the base CI. You can then select one or more CIs as a second CI for the relationship. Depending on the selected relationship type, the base CI can become the parent CI or the child CI in the new relationship

Configuration Items: Dependency View

servicenow

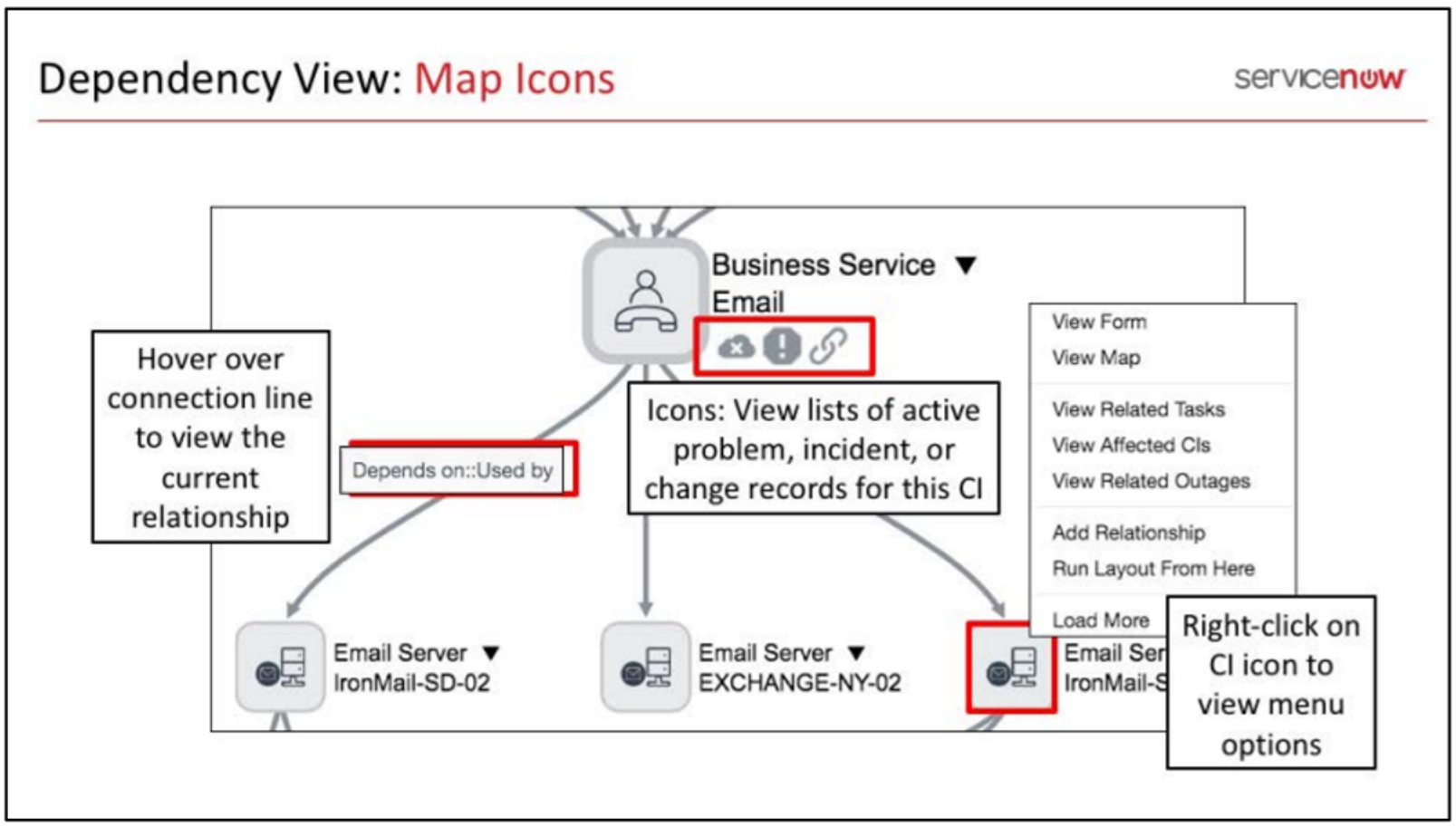
Dependency Views graphically display an infrastructure view for a configuration item and the business services that it is part of and that it supports

Dependency Views indicate the status of configuration items, and allow access to the CI's related alerts, incidents, problems, changes, and business services



The maps generated by Dependency Views are based on D3 and Angular technology, providing a modern interactive graphical interface to visualize configuration items and their relationships.

Use the Dependency View to view other configuration items “upstream” that feed data into an email service, for example, and then “downstream,” where you can view all of the other CIs that the email service is dependent on.



In a Dependency View Map, icons and glyphs indicate whether a CI has an active, pending issue. You can investigate the tasks that are connected to a CI to get more details. The map collapses and expands clusters to make them easier to view.

Understanding the dependencies and other relationships among the CIs will enhance the operational delivery of incident, change, and problem management processes.

CMDB: Implementation Considerations

servicenow

Questions to consider before implementing a CMDB:

1. How is data entered or imported then managed?
   Consider people, process, and technology

2. Where is data stored?
   Identify and extend tables accordingly

3. What data is necessary?
   Store only the data being used and omit unused data fields

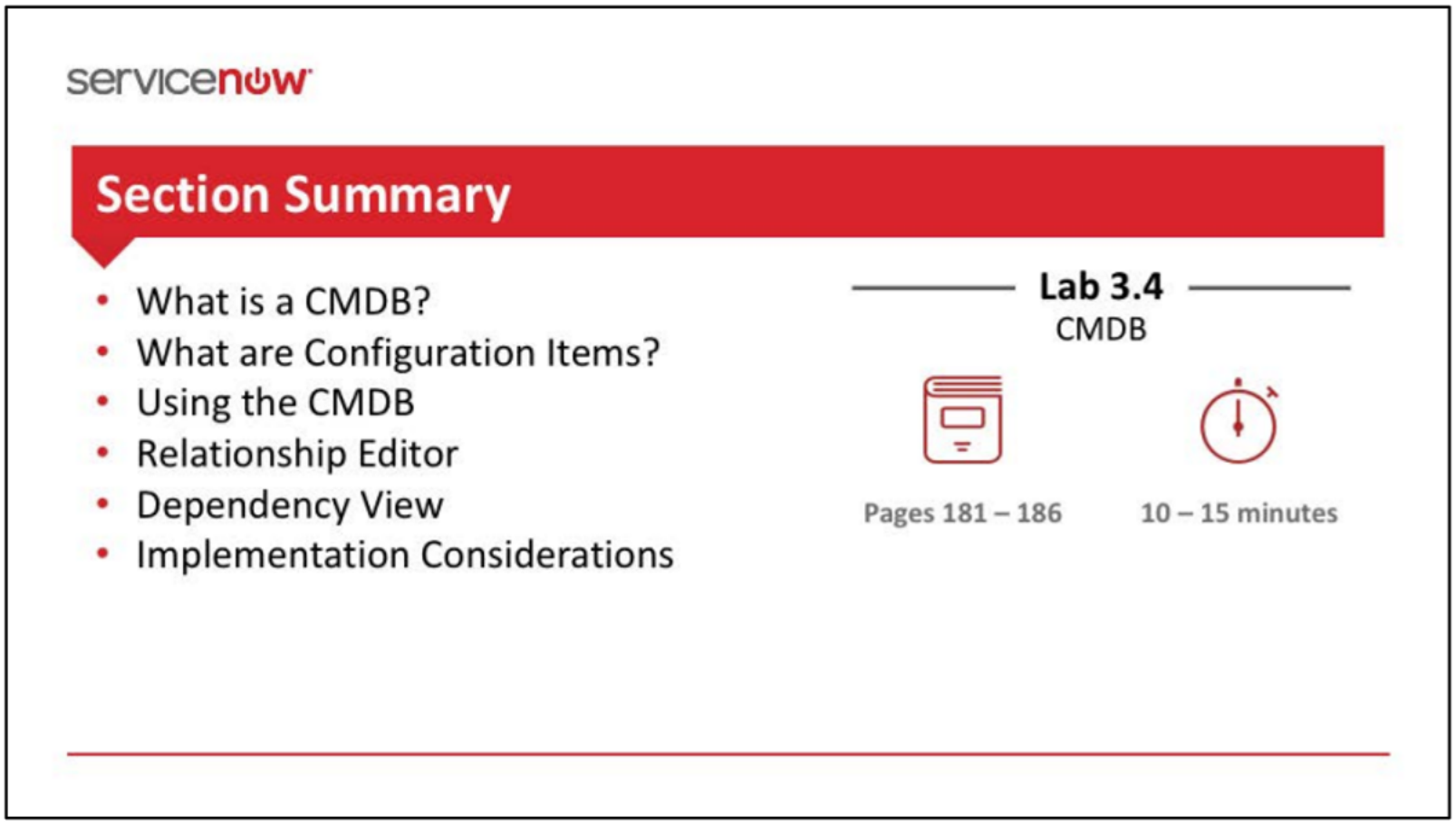
4. When should imports or refreshes of CI data happen?
   Schedule imports to maintain up-to-date data

A core component in IT best practices, the CMDB serves to maximize the alignment of human capital, technology, and business processes. Setting up a CMDB is a company commitment with measurable financial benefits once everything is set up. You need to keep data current and plan refreshes.

Review the base instance tables when implementing the CMDB. A table you want to create may already exist. Also, prepare and draw out your CMDB schema beforehand so you know what tables you are going to use, which ones you are going to turn off, and what relationships you are going to allow.

Methods for populating the CMDB include: Import Sets, integrating with external CMDBs, and manual input.

Additionally, tools in ServiceNow such as **Help the Help Desk** and **Discovery**, a licensed offering, are efficient methods for gathering configuration item data.



Lab 3.4 – CMDB:

- Create a new CI class in the CMDB
- Define CI relationships

LAB

CMDB 3.4

10 – 15 minutes

Lab Goal

Lab Dependency: Requires the completion of Lab 3.1.

This lab will show you how to do the following:

- Create a new CI class in the CMDB
- Define CI relationships

All Infinity devices rely upon a media server, to process and stream data, which will require tracking in the CMDB.

The media server can be categorized under the existing CI Server Class.

With both the Infinity and media server being tracked in the CMDB, a relationship between the two CI Classes will be established – this relationship could be used by Cloud Dimensions to potentially identify an impact of a change mangement request or outage.

A. Explore the CI Class Manager

To begin, explore the CI Class Manager interface. The CI Class Manager displays the entire CI Class hierarchy in a tree-view format, consolidating class definitions into a central location. It enables an easier method for viewing, modifying, or extending CI Classes.

A CI Class represents a type of Configuration Item or essentially a table collecting certain data, such as Applications, Computers, Printers, Servers, etc.

In the last lab, a new CI Class was defined for the Infinity – categorizing it as a Hardware CI because of the cmdb_ci_hardware table extension.

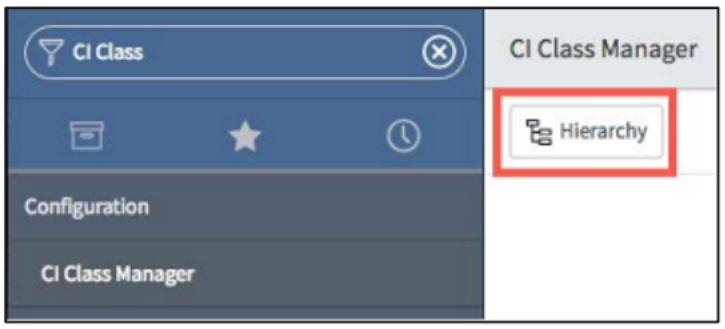
1. Impersonate **Darrel Tork**.

NOTE: *This user is the Configuration Manager of Cloud Dimensions. They were provided the* ***itil*** *role, in order to access the* ***CI Class Manager****. If your company does not have a Configuration Manager, this responsibility could fall onto the system administrator.*

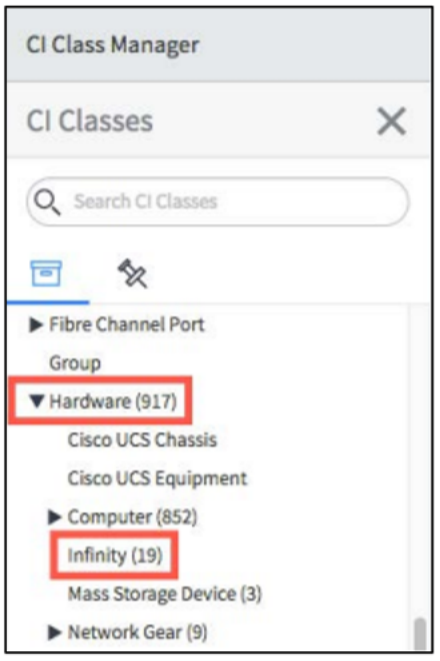
---

2. Navigate to **Configuration > CI Class Manager**.

3. Select the **Heiarchy** button:



4. From the CI Classes menu, scroll down to locate and expand the **Hardware** section, to find the **Infinity** class:



***NOTE**: This confirms that the Infinity table and its consequential data records are tracked in the CMDB, even though they are accessed from their own application menu and respective modules.*

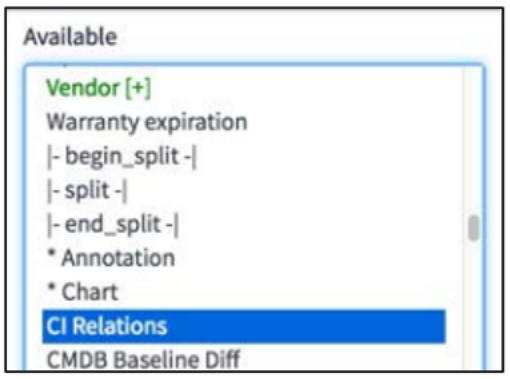
B. Add a New Infinity Device to the CMDB

Unlike existing CI Classes, the **CI Relationships** section is not visible on the new Infinity CI Class. The form needs to be configured to display this section

1. **Infinity > Add Inventory**.

2. From the **Form Context Menu** select **Configure > Form Layout**.

3. Scroll down the list of Available fields to locate **CI Relations**:



***NOTE**: This field displays after the alphabetically sorted list of available fields, as it is a formatter rather than a standard field. It will not display on the form layout until the record has been saved.*

---

4. Add the **CI Relations** field to the bottom of the Selected list.

5. Click the **Save** button.

6. Fill out the Infinity New Record form as shown:

   Device Number: **CDE0100100**
   Name: **Infinity Beta Prototype**
   Device Version: **BP01**
   Support Group: **Service Desk**
   Installed: **[today's current date and time]**
   Owned by: **Darrel Tork**
   Email: **darrel.tork@cloudd.com** (auto-fills)

7. **Submit**.

C. Create a New Suggested CI Relationship

Switching back to system administrator, you will create a new suggested CI relationship between Infinity and the Infinity Media Server, using Darrel's new incident.

The Infinity Media Server is responsible for sending content to Infinity devices.

With a CI relationship defined, tools like the CI dependency view can be used by Infinity Support agents to identify the level of impact when issues occur.

1. Impersonate **System Administrator**.

2. **Configuration > Relationships > Suggested Relationships**.

3. Click **New**.

4. Fill out the form as shown:

   Base class: **Infinity**
   Relationship: **Receives data from (parent)**
   Dependent class: **Server**

5. **Submit**.

6. **Configuration > Servers > All**.

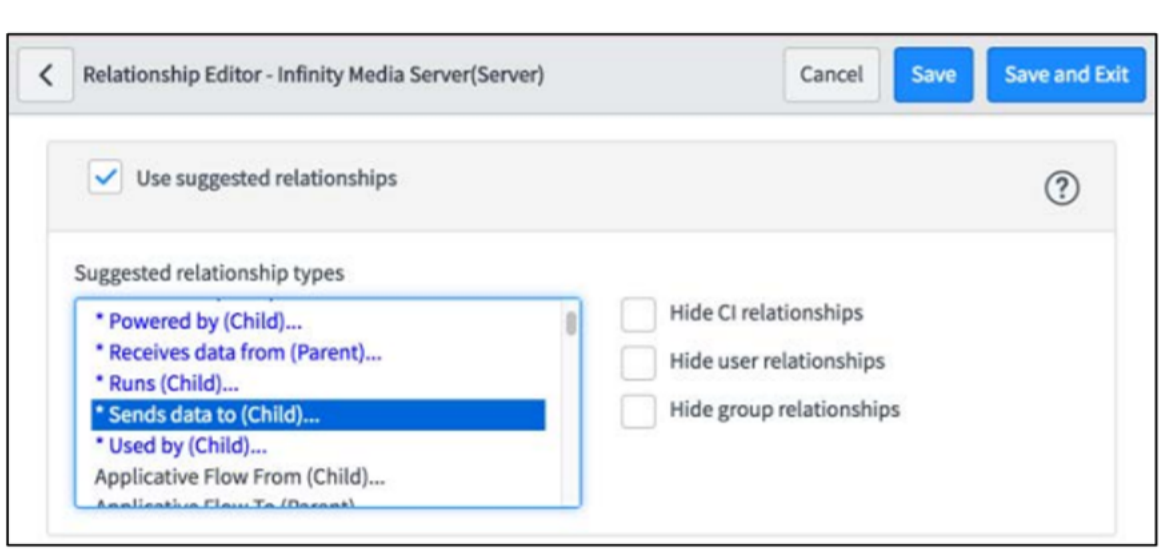
7. **New**.

8. Name: **Infinity Media Server**.

9. **Save**.

10. Scroll down to the Related Items section, then click the **Add CI relationship** icon:

11. In the Suggested relationship types field, select *** Sends data to (Child)...**

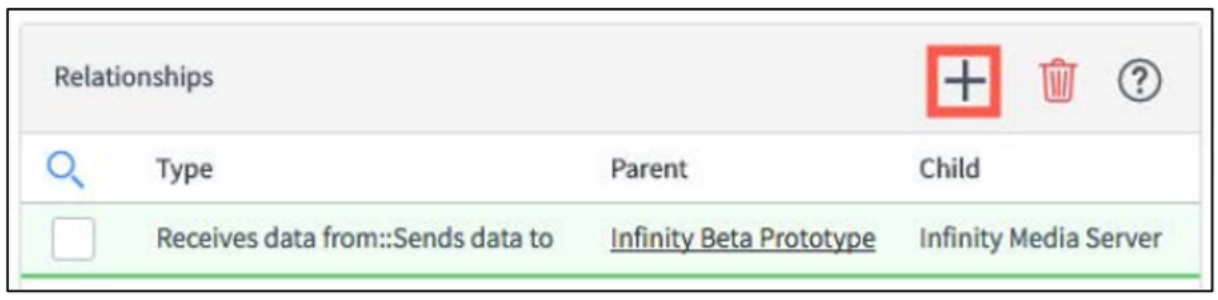


12. From the **Configuration Items** section, use the Updated field to sort the records by last updated.

13. Use the checkbox to select the **Infinity Beta Prototype** record:

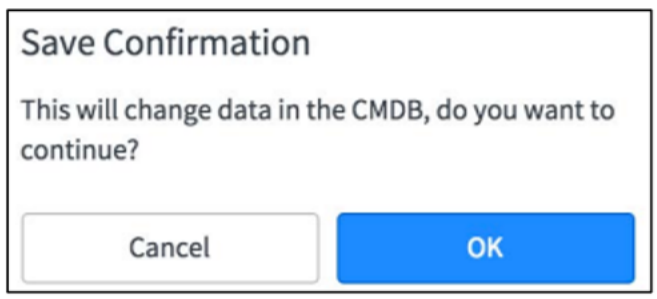
***NOTE**: While only the Sends data to (Child) relationship was created, suggested relationships also accounts for the converse relationship, Receives data from (Parent).*

14. Scroll down to the Relationships section and add **Infinity Media Server** by selecting the **Create new relationships** icon (+):



15. Click the **Save and Exit** button.

16. A Save Confirmation pop-up message may display, click **OK**:

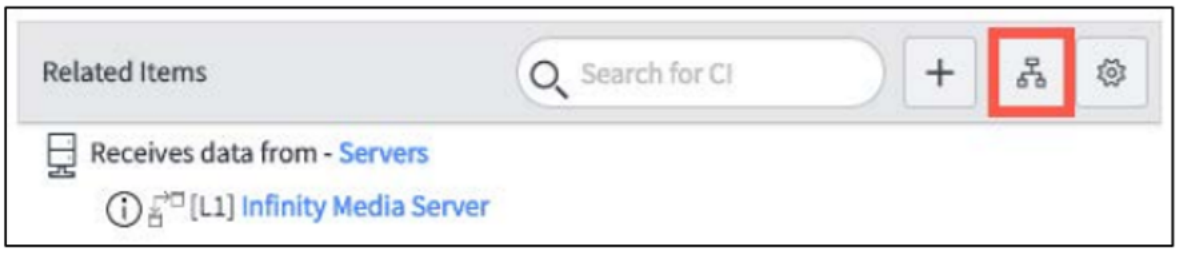


LAB VERIFICATION

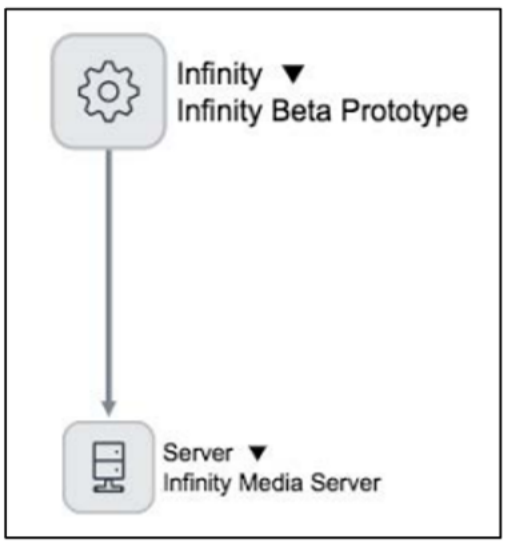
New Infinity Media Server CI and Relationships

1. **Infinity Inventory > All Devices**.
2. Locate and open the **Infinity Beta Prototype** record; **CDE0100100**.
3. Verify the Related Items section looks like this:

4. Open the Dependency View by clicking the **Show dependency views** icon from the Related Items toolbar:

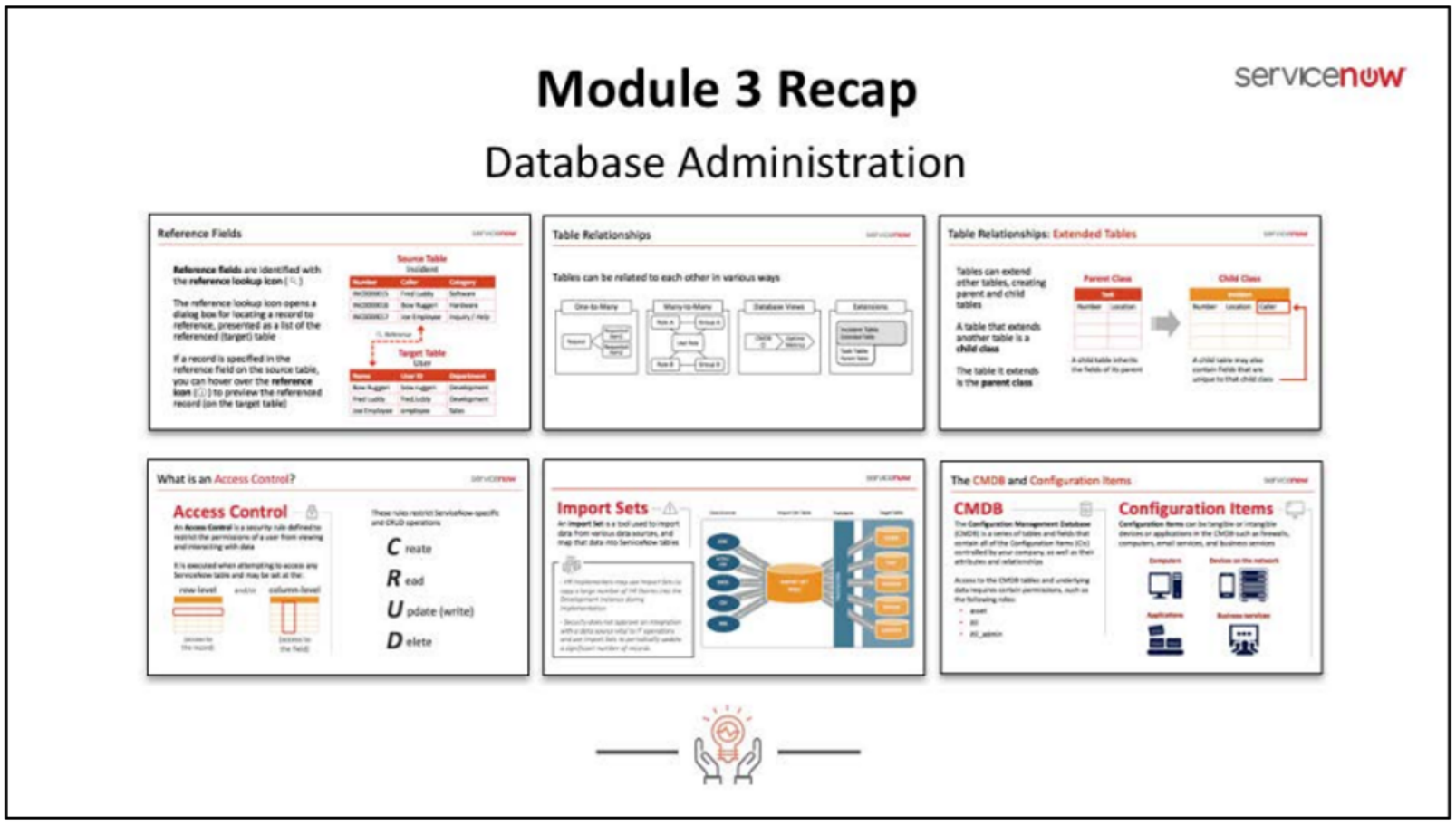


5. The Infinity Testing Device Dependency View displays in a new browser tab/window:



***NOTE**: This displays an example relationship between the Infinity device and Infinity Media Server CI Classes.*

That concludes the CMDB lab. Excellent work so far!

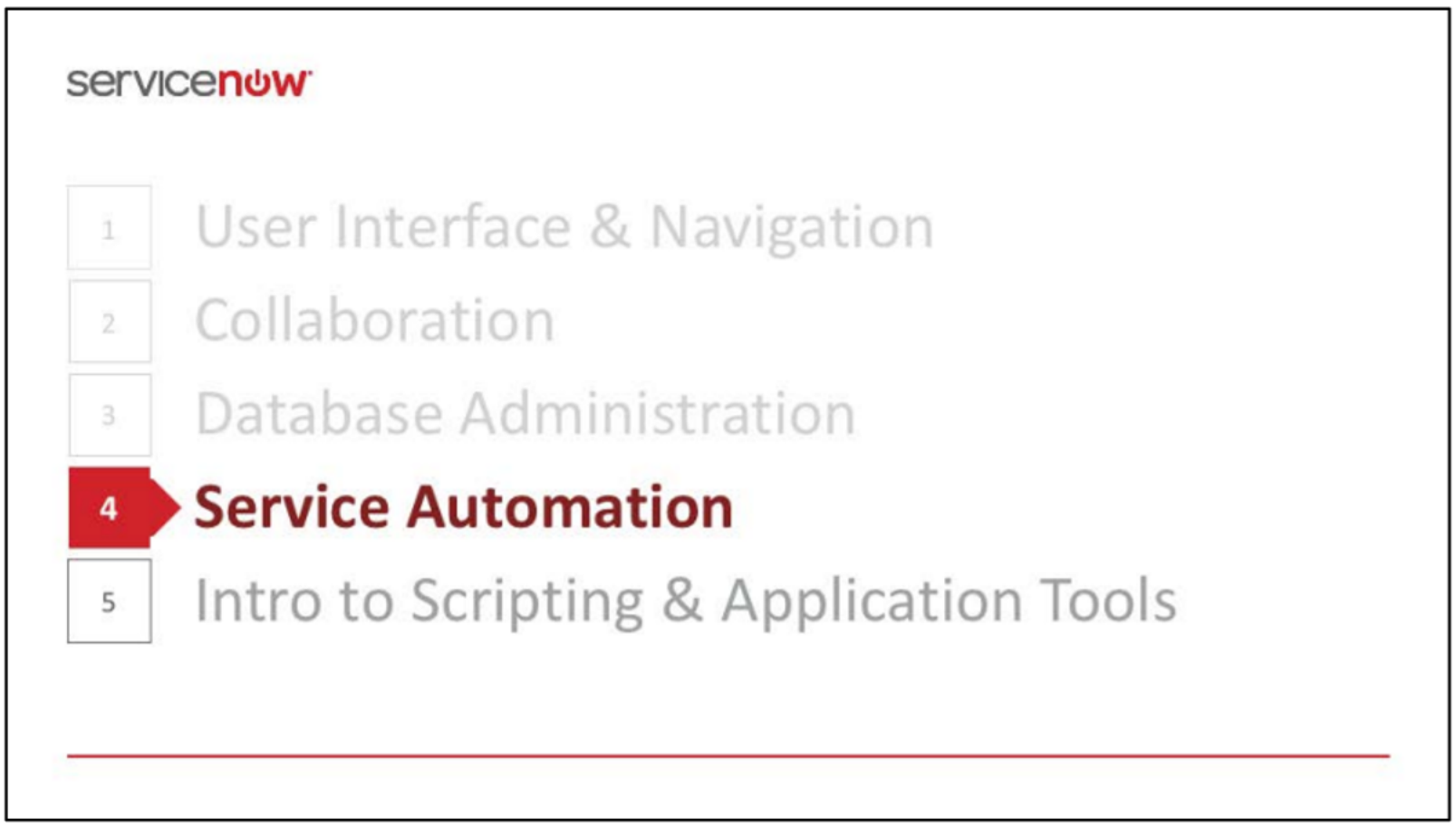


For these selected topics, discuss:

Why would you use these capabilities?

When would you use these capabilities?

How often would you use these capabilities?



Module 4 – 4.1 Service Catalog

servicenow

Objectives

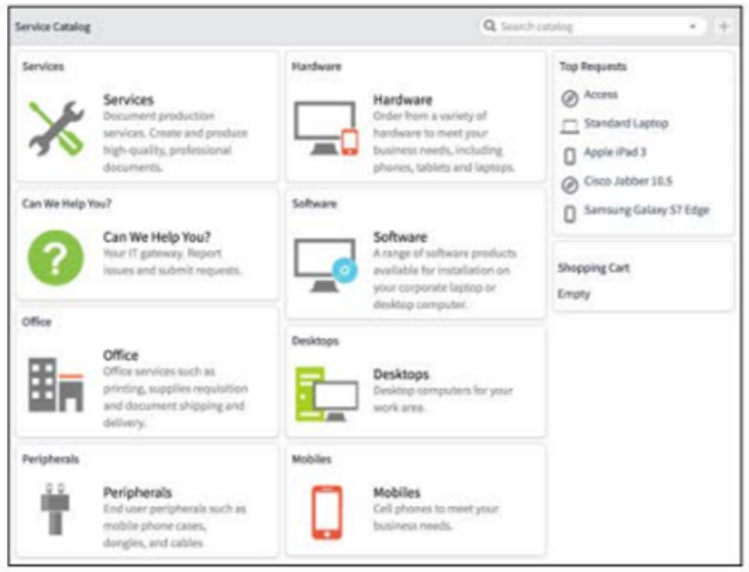
- What is the Service Catalog?
- Major Components
  - Catalog Items
  - Variables/Variable Sets
  - Record Producers
  - Order Guides
  - Workflows
- Service Catalog Item Request Output
- Service Catalog Security: User Criteria

servicenow

Service Catalog

The **Service Catalog** is a robust ordering system for services and products offered by various departments for users:

- Categories organize catalog items
- "One stop shopping" for users
- Access to the Service Desk
- Help and Training Portal
- Multiple catalogs are supported

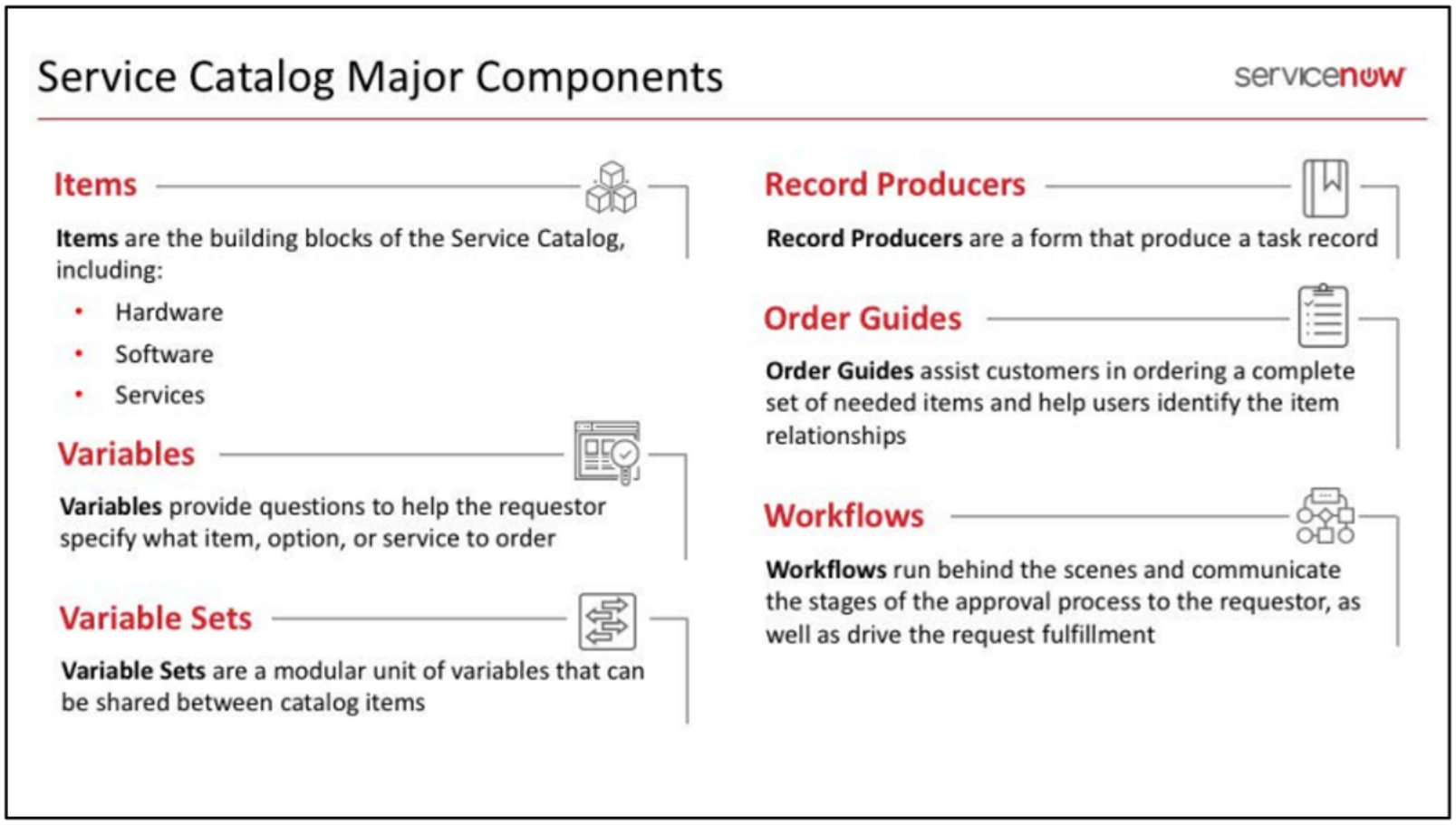


The IT Service Catalog lets users see a list of things they may need (to create a request for), or would like to have from IT. Administrators and users with the catalog admin role can define catalog items, including formatted descriptions, photos, and prices.

Categories define the organization for Service Catalog items. Categories organize service catalog items into logical groups. Categories can have a parent-child relationship, such as **IT** and **Laptops**. A child category is a subcategory of its parent category. Each Catalog Item, Order Guide, Record Producer, Content Item, and subcategory appears as a single item within the category.

In Service Catalog, Employee Self Service (ESS) users can order pre-defined, bundled goods and services from the IT organization, or other departments. ITIL users or Administrators see additional choices; Administrators can view all the Service and Item categories. **Top Requests** is a dynamic category displaying the five most ordered items.

The ServiceNow platform supports multiple Service Catalogs. System Administrators and Catalog Administrators can manage multiple Service Catalogs and provide services to different teams within the organization. Examples include: IT Services, Human Resources, and Facilities Management. Views can be defined for groups that view a Catalog, and Catalog Items can be shared by multiple Catalogs



Items: In the Service Catalog, users locate a category for an item or service they want to order, and then click the subcategory link.

Variables: Provide options gather specific information related to the customer's needs. Questions that define item options can be added to ask the end user ordering the catalog item. Variables can affect the order price. Service Catalog variables are flagged as "Global" by default and will display in all the execution tasks of a requested item. A variable is defined once and can be used in multiple places. The Service Catalog allows you to attach variables to a catalog item.

Variable Sets: You attach variables either to a Catalog Item or to an Execution Plan.

Record Producer: An interface used as an alternative to lists and forms. Each Record Producer focuses on a specific process or task and can be used anywhere in the ServiceNow platform. In the Service Catalog, Record Producers are presented in categories along with catalog items. Users can use Record Producers to create an incident, request an emergency change, and more. This enables the Service Catalog to be used as a complete front-end UI.

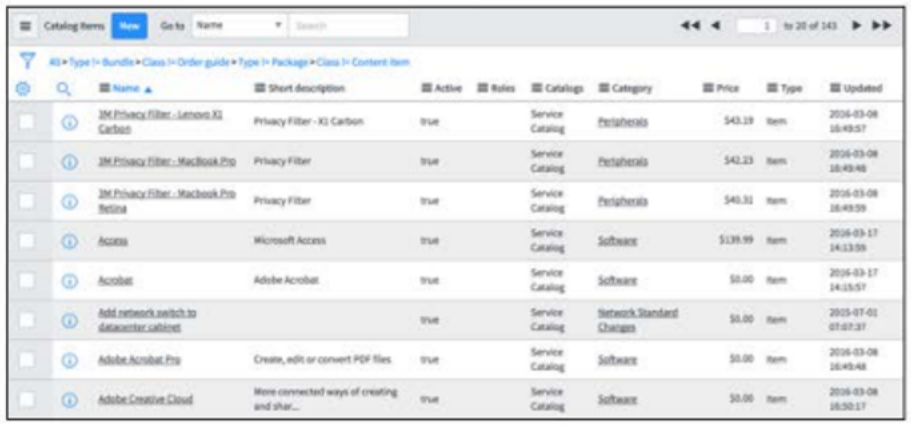
Order Guides: Order Guides provide the ability to order multiple, related items as one request. Questions can be used to present item options; present users with only **relevant questions and choices** at the appropriate time in the ordering process.

Workflows: When you create a new service catalog item, you can create a new corresponding workflow at the same time.

Service Catalog Items and Variables

servicenow

To create a new item or modify an existing item, navigate to **Service Catalog > Catalog Definitions > Maintain Items**



Once an item is published to the Service Catalog, users will be able to order it

Variables

- Global by default
- Define the questions to ask the end user ordering the catalog item
- Question choices can define the available options and may affect the order price

Examples:

- Which monitor size?
- Who is the hiring manager?
- What is the budget code?

Service Catalog variables are global by default and provide options to tailor a catalog item to the customer's needs. For example, a computer may be available with different operating systems.

The Service Catalog lets you attach variables either to a catalog item or to an execution plan.

Common Variable Types

- **Multiple Choice:** Creates radio buttons for user-defined question choices.
- **Select Box:** Creates a choice list of user-defined question choices.
- **Single Line Text:** Creates a single-line text input field.
- **Reference:** Specifies a record in another table, similar to a reference field.
- **Checkbox:** Creates a checkbox that may be selected or cleared; list checkboxes in order under a label to create an options question.

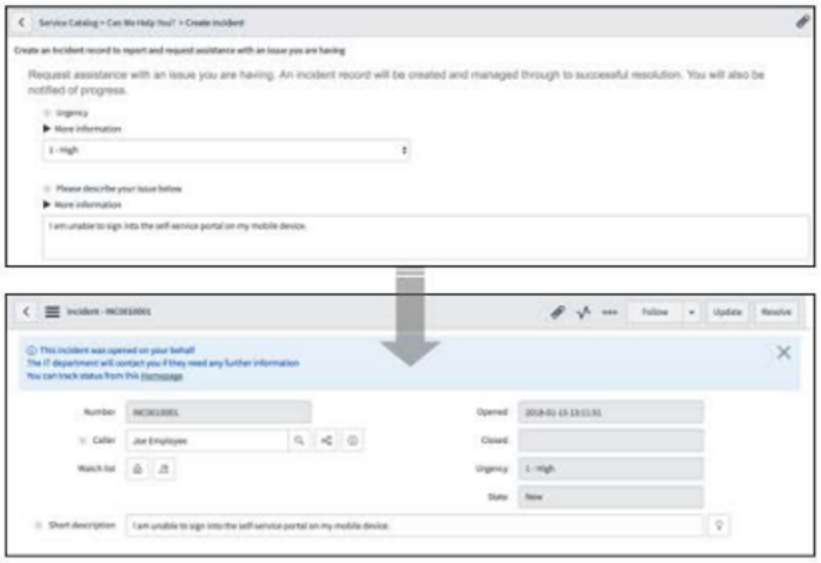
Functionally, a **Variable Set** is just a container, so it has only two fields: **Name** and **Description**. From the Application Navigator, select **Service Catalog > Catalog Variables > Variable Sets**, and create a new variable set. After you save the variable set, you will get a Related List at the bottom where you can add as many variables as you want.

Used by items and variables, the Order field establishes the sequence for displaying information. For example, an item with 100 in the Order field, displays first in the list. An item with 200 in the Order field will display second.

Record Producers

servicenow

Record Producers appear as simplified forms, allowing users to provide information that is translated into records being added or modified in the database



- When employees use the Service Catalog on the HR Service Portal to submit a request for Direct Deposit setup, a record producer transfers the request into an HR case

- Incident management may use a record producer for users who need to open an incident easily via the web

- Facilities management uses a record producer for users to open requests for printer service

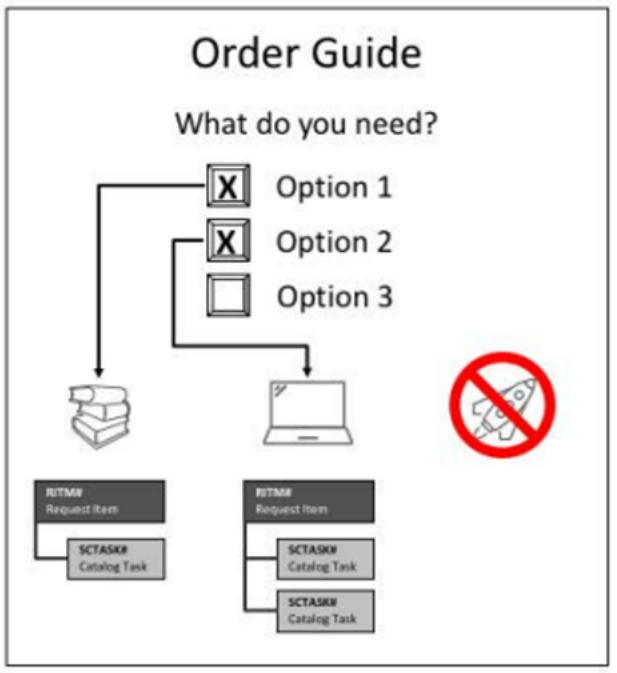
A Record Producer focuses on a specific process or task and can be used anywhere in the ServiceNow platform. In the Service Catalog, Record Producers are presented in categories along with catalog items where each table has its own record-identifying designation.

Order Guides

servicenow

Define an **Order Guide** to assist customers in ordering a complete set of needed items and to help users see item relationships

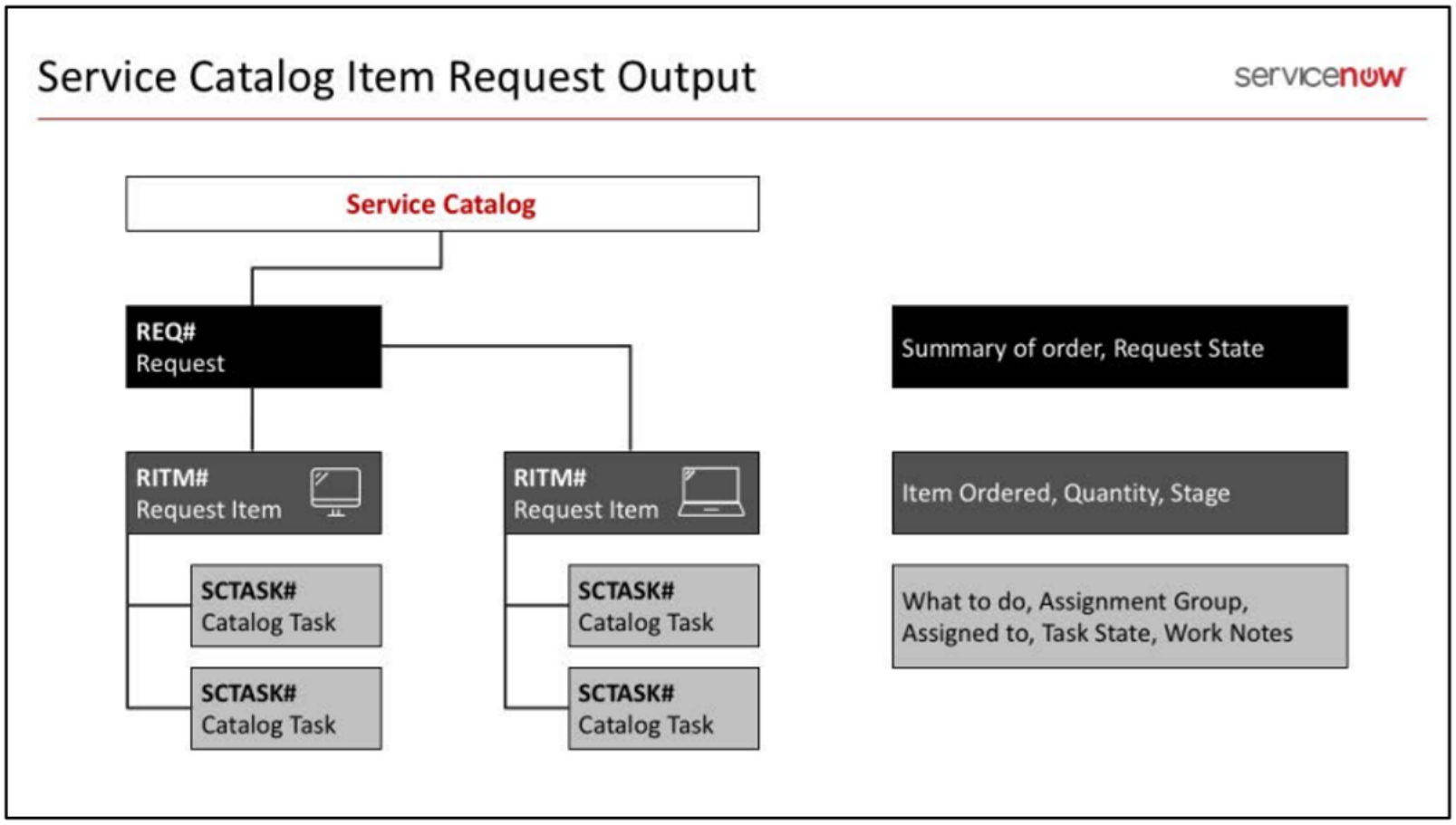
Questions can be used to present item options; present users with only **relevant questions and choices** at the appropriate time in the ordering process



Items are the building blocks of the Service Catalog. Once you have built a complete item with variables and a delivery workflow, you can do a number of things with it, such as add it to an Order Guide.

Order Guides provide the ability to order multiple, related items as one request. Remember that variables are presented by the **Order** field number.

Use an Order Guide to assist users in determining what items they need.



For Catalog Items, a request, an item, and a task are all created when an order is placed, each on a corresponding table:

REQ# Request [**sc_request**] table: A request number generated to keep track of an order.

RITM# Requested Item [**sc_req_item**] table: Within a request generated from a catalog order, each discrete item ordered is given a specific "Requested Item Number" known as an RITM (number).

SCTASK# Catalog Task [**sc_task**] table: In the **Catalog Tasks** section for an ordered item, the different tasks display for what has to be done to get the item ready for delivery to the user, for example; the Assignment group, the Due date, Work start, and Work end dates.

In this example, a manager orders two new computers for the team.

The first requested item is a desktop computer with two catalog tasks associated to it: order equipment and configure equipment.

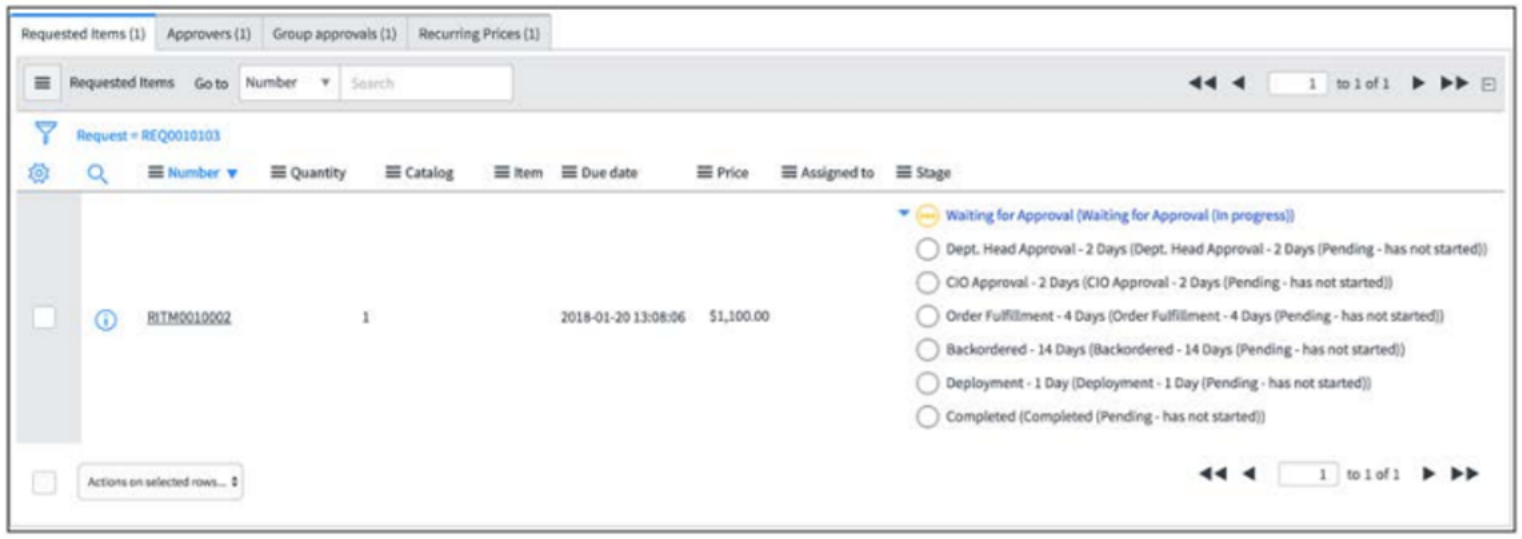
The second requested item is a laptop computer with two catalog tasks associated to it: order equipment and configure equipment.

Although the requested items share similarly named catalog tasks, these tasks are tied directly to their respective item because the required steps for completion may be different from item to item.

Progress Stages for a Requested Item

servicenow

While viewing the requested item, you can expand the workflow stages which provide summary-level feedback about the progress or state of an item in the delivery process



After an request has been submitted, users are able to easily track it by navigating to **Self-Service > My Requests** and opening the record associated with the request.

Workflows attached to an item may indicate the progress or state of an item in the delivery process with one of the following stages:

- **Waiting for approval** (In Progress)
- **Approved**
- **Pending** (has not started)
- **Fulfillment** (In Progress)
- **Deployment/Delivery**
- **Completed**

Additionally, workflows can have multiple rounds of approval actions as it relates to Service Catalog requests. Each approval action can share the same stage value or their own separate stages.

Stages can be grouped into a Stage Set for convenience of applying related stage values from workflow to workflow.

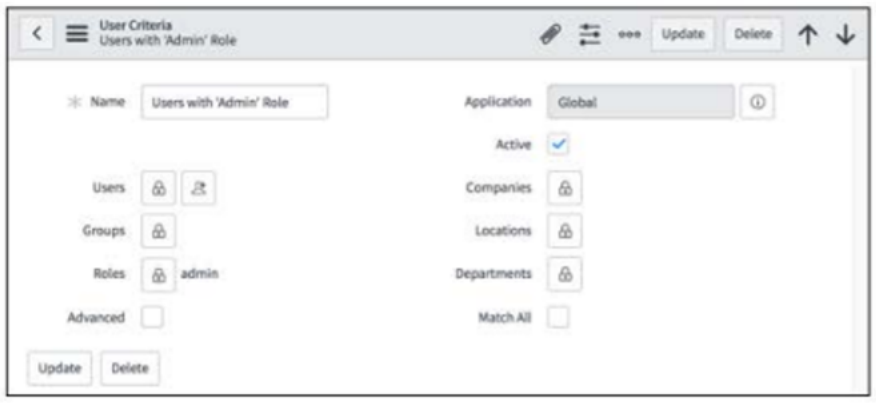
Service Catalog Security: User Criteria

servicenow

User Criteria defines conditions that are evaluated against users to determine which users can access Service Catalog items

You can apply several user criteria records to a single catalog item or category

User Criteria is applied to an item or category



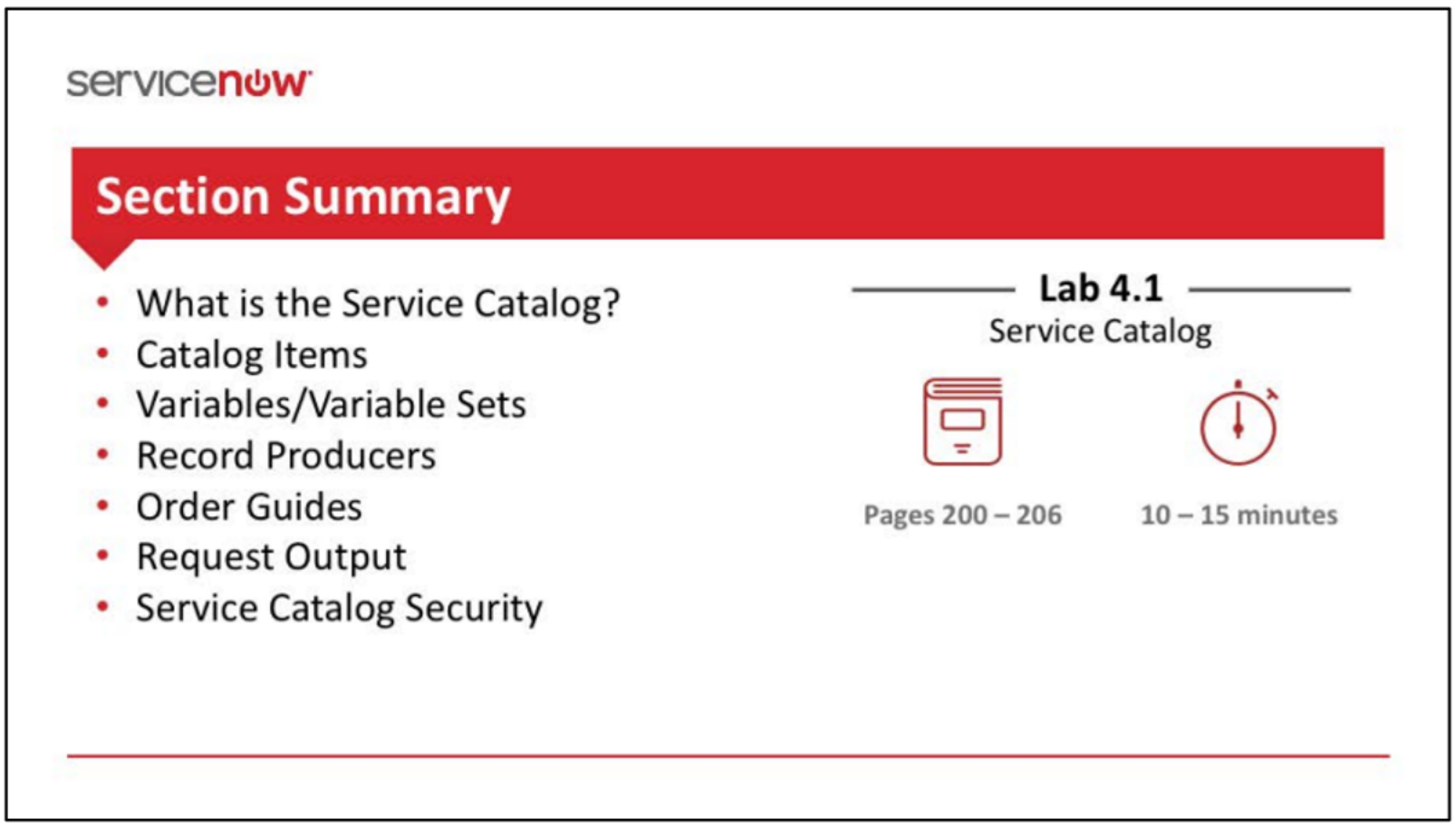
To apply user criteria to an item or category, open the respective record and navigate to the **Available For** or **Not Available For** related lists.

NOTE: These related lists are not on the form by default and must be added by configuring the form.

Next, click **Edit** to add an existing user criteria record, or click **New** to create a new one.

Save the record to associate the user criteria record with the item or category.

NOTE: The **Not Available For** settings override the **Available For** settings. A user on the **Not Available For** list cannot access an item or category, even if that user is also on the **Available For** list.



Lab 4.1 – Service Catalog:

- Create an Infinity Service Catalog item
- Add item variables
- Validate your work with “Try It”

LAB

Service Catalog

4.1

10 – 15 minutes

Lab Goal

This lab will show you how to do the following:

- Create an Infinity Service Catalog item
- Add item variables
- Validate your work with "Try It"

Up until now, the employee ordering and fulfillment process for an Infinity has been entirely "off the books" – unofficial, to say the very least.

With the availability of the Service Catalog, Cloud Dimensions would like to improve the process and ensure every employee has a chance to receive an Infinity device. Additionally, tracking orders and inventory in one convenient location is appealing.

Infinity is offered to employees in either Crimson or Silver, with optional, additional specifications to choose from.

Required Resource: CloudDimensions-Infinity-Logo.png

A. Create New Service Catalog Item

1. Impersonate **Asset Manager**.

   NOTE: *This user account is provided the* ***catalog_admin*** *role, which grants them access to the* ***Maintain Items*** *module in order to complete Service Catalog administration tasks.*

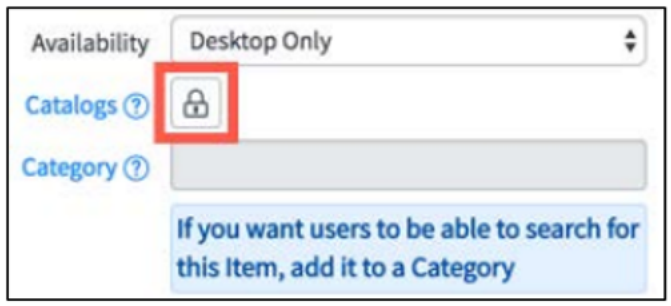
2. **Service Catalog > Catalog Definitions > Maintain Items**.

3. Click **New**.

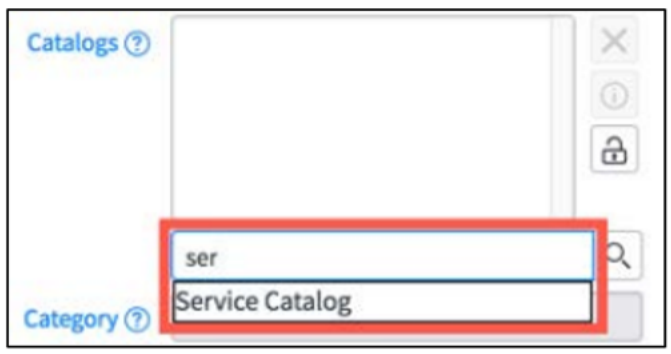
4. Fill out the top of the form as shown:

   Name: **Infinity**
   Price: **$395.99 USD**

5. Click the **lock** icon (Edit Catalogs) next to Catalogs:



6. Start typing **ser** then select **Service Catalog**:



7. In the Category field, select **Hardware**.

NOTE: *Category is used to determine where a catalog item appears within the Service Catalog. From the left navigation pane, under Service Catalog > Maintain Categories, additional categories can be created and categories can be created hierarchically.*

---

8. Add a **Short Description** and **Description** as shown:

   Short description: **VR, but without the glasses**

   Description:
   The Infinity is a portable holographic projector (PHP) that is capable of projecting immersive environments around you. What are you waiting for? Get Infinity!

9. In the Picture field, select **Click to add…**

10. Choose File: **CloudDimensions-Infinity-Logo.png**

11. Click **OK**.

12. Click **Update**.

Verify the Service Catalog Item Creation

1. **Self-Service > Service Catalog**.

2. Click the **Hardware** category header.

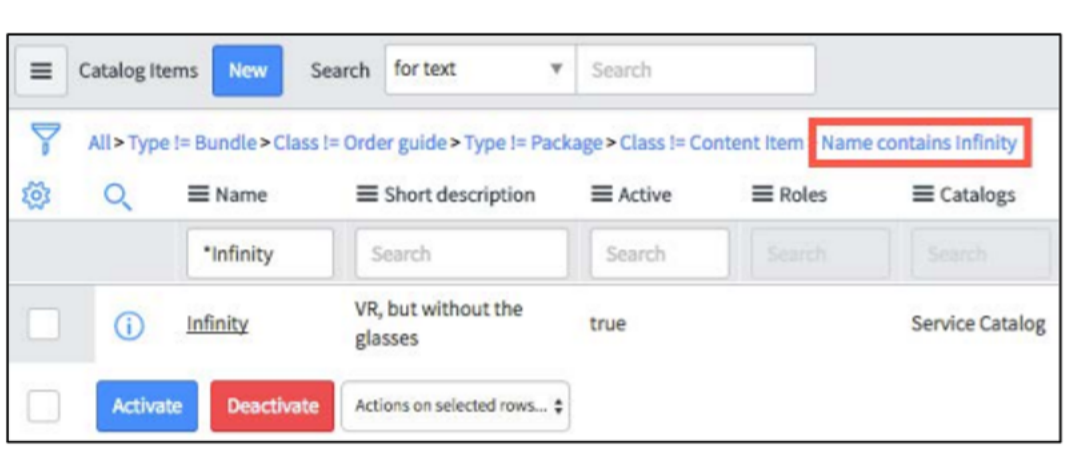
3. Select the **Infinity** item from the list to open the order screen.

Add a Memory Variable to the Service Catalog Item

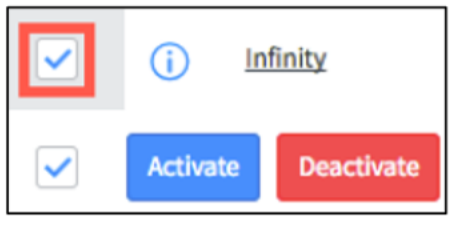
From this form, a user could order an Infinity, but the form does not yet allow users to specify options for the device. Let us fix that!

1. **Service Catalog > Catalog Definitions > Maintain Items**.

2. Filter the list by adding a “Name contains Infinity” condition:



3. Select the **Infinity** record:



4. Click the **Deactivate** button.

NOTE: *This button deactivates the catalog item(s) selected on the list, making them inaccessible to users in the Service Catalog.*

---

5. Open the **Infinity** record.

6. From the **Variables** tab, select **New**:

7. Fill out the form as shown:

   Type: **Multiple Choice**
   Mandatory: **[select checkbox]**

8. Under the **Question*** tab, enter the Memory question as shown:

   Question: **How much memory do you want in your Infinity?**
   Name: **memory**

***NOTE**: The corporate-approved version of the Infinity is available in both 256GB and 512GB models.*

---

9. Click the **Default Value*** tab.

10. Enter the Default value: **256**.

11. **Save** the form, instead of Submit, to define the Infinity memory Question Choices.

12. Scroll to the **Question Choices** section then click the **New** button.

13. Fill out the **Question Choices** form as shown:

    Order: **100**
    Text: **256GB**
    Value: **256**

14. **Submit**.

15. In the **Question Choices** section, click **New.**

16. Fill out the form for the second memory choice:

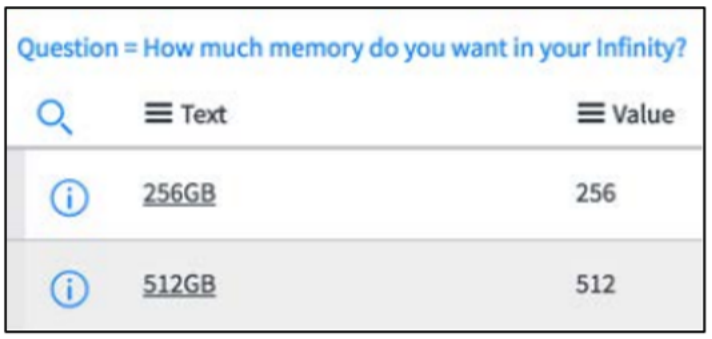
    Price: **$100.00**
    Order: **200**
    Text: **512GB**
    Value: **512**

    ***NOTE**: The Price field is utilized by the 512GB choice, as it adds $100 to the overall price. Additionally, the order value is set to 200 which will place the 512GB memory option second in the choice list.*

    ---

17. **Submit**.

18. You should now see both question choice values:



19. Click **Update** to return to the Infinity item record.

B. Add Color Variables to Service Catalog Item

With the first variable defined to provide end users with a choice of memory options, you will now create a second question for choosing a color choice by configuring a second variable.

1. In the **Variables** related list, click **New**.

2. Complete the form as shown:

   Type: **Select Box**

   In the **Question*** tab,
   Question: **What color Infinity would you like?**
   Name: **color**

   In the **Default Value** tab,
   Default value: **crimson**

3. **Save**.

4. Scroll to the **Question Choices** section then click the **New** button.

5. Fill out the form for the first color choice:

   Text: **Crimson**
   Value: **crimson**

6. **Save**.

7. Change the following field values for the second color choice:

   Order: **200**
   Text: **Silver**
   Value: **silver**

8. Open the **Form Context Menu**, then select **Insert** to return to the color variable record.

NOTE: *You have just defined the values for the What color would you like? variable, using question choices.*

Another option for providing value choices is to use reference tables or fields from the database. See the ***Choice table*** *and* ***Choice field*** *options under the* ***Type Specifications*** *section on the Variable form.*

docs.servicenow.com defines all of the possible variable types.

---

9. Click **Update** to return to the Infinity item record.

LAB VERIFICATION

Test the Catalog Item

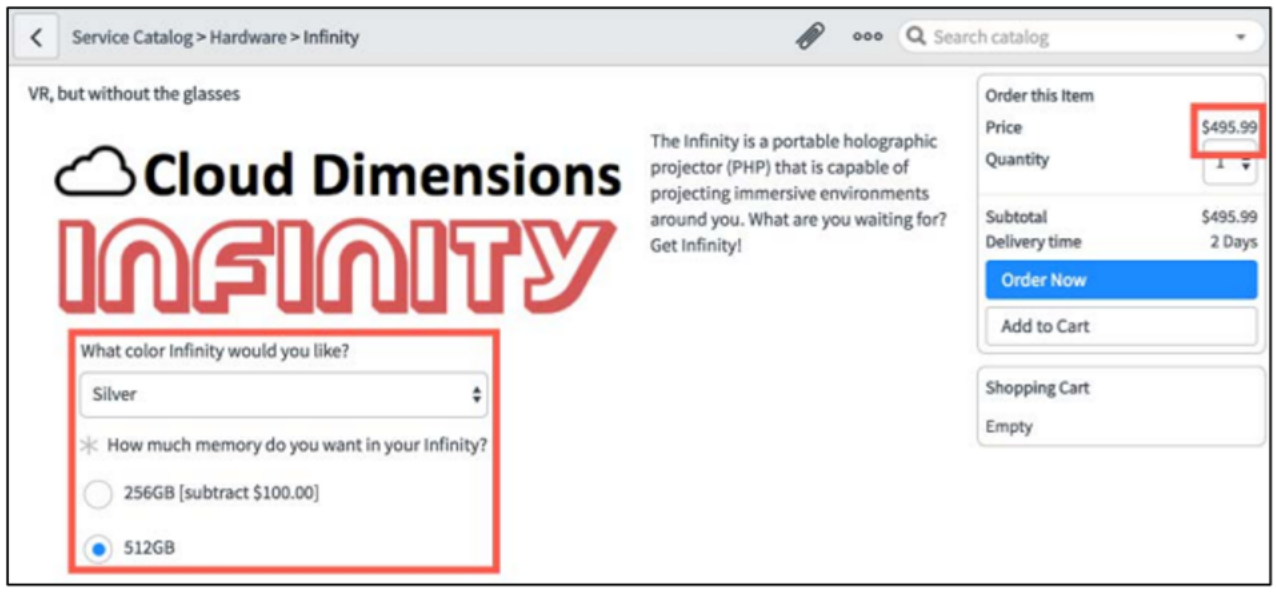
1. From the top of the **Infinity** Catalog Item form, click the **Try It** button to view the new variable options added.

NOTE: *The* ***Try It*** *button is only available if the item is active.*

---

2. Change the Infinity color to **Silver**.

3. Choose **512 GB** and notice how the item **Price** changes:



Congratulations! With this labs success, adding a workflow to the item will be no problem!

Module 4 – 4.2 Flow Designer & Workflow

servicenow

Objectives

- What is the Flow Designer?
- What is a Workflow?
  - Using Workflows
  - Workflow Execution
- Workflow Editor
- Workflow Activities, Transitions, Conditions, and Stages
- Workflow States
  - Versions
  - Contexts

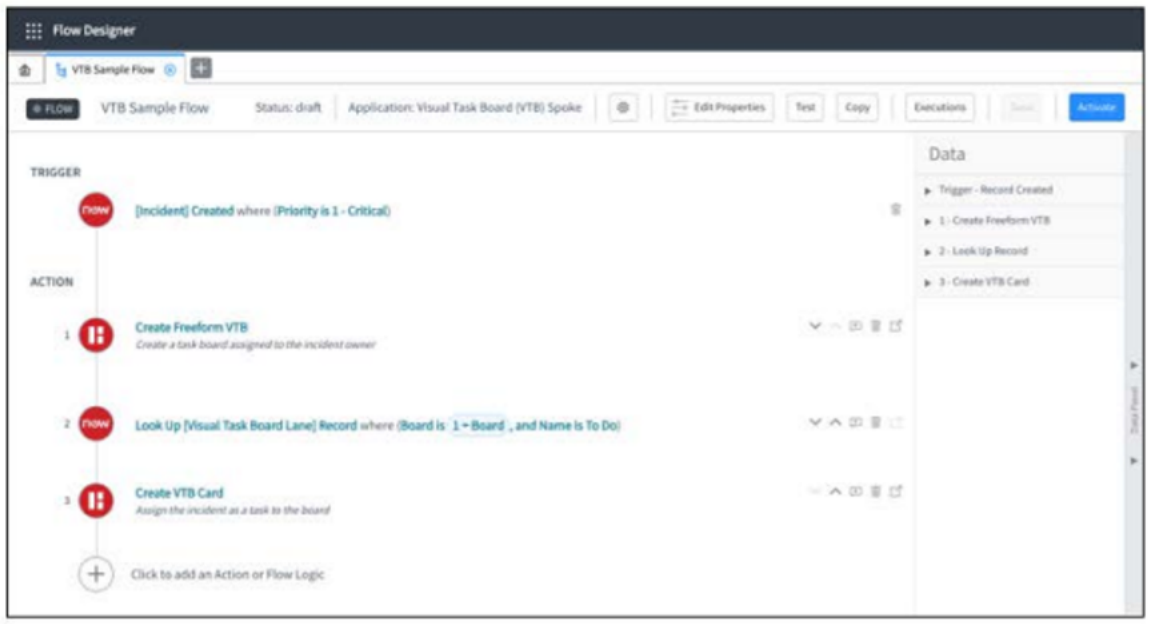
What is the Flow Designer?

servicenow

The **Flow Designer** is an interface for building and enabling rich process automation capabilities, known as **flows**

Flows automate business logic for a particular application or process such as approvals, tasks, notifications, and record operations

Flows include a **trigger** (an activity that initiates the flow) and **actions** (operations executed by ServiceNow)



Launch the Flow Designer by navigating to **Flow Designer > Designer** in the Application Navigator. The following roles may be required to work with the Flow Designer in one capacity or another: **flow_designer**, **flow_operator**, and **action_designer**.

Example actions available include:

- Ask for Approval – create approvals on any record, including rules for an approval, rejection, or cancellation, and a due date
- Create Record – create a record on any table with configurable fields and field values
- Look Up Record – look up a single record on any table, confirming whether or not it exists and using this in conjunction with other actions
- Add User to Task Conversation – add a user to a task record conversation (accessible via Connect Chat)
- Create Flexible VTB – create a data driven Visual Task Board (VTB), including default lanes and other configuration details

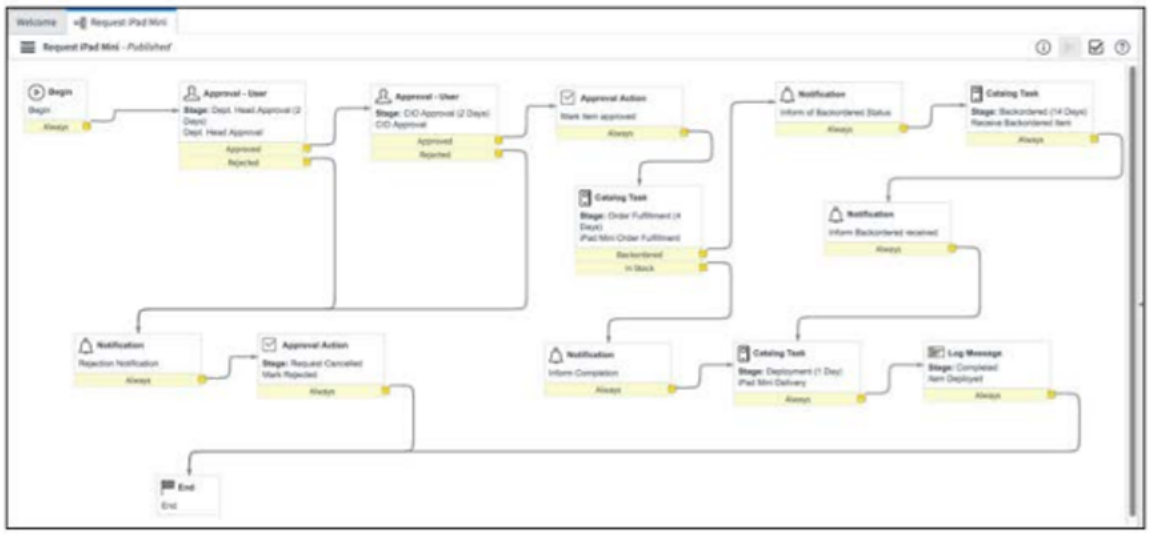
In addition to these core actions, new application-specific core actions can be created by activating the associated spoke.

The natural-language descriptions of flow logic helps non-technical users understand triggers, actions, inputs and outputs, allowing all subject matter experts the ability to develop and share reusable actions with flow designers.

What is a Workflow?

servicenow

A **workflow** is an automated multi-step process occurring across the platform



This is a workflow **model** which represents the entire structure of the workflow and the definition of the processing a workflow can accomplish

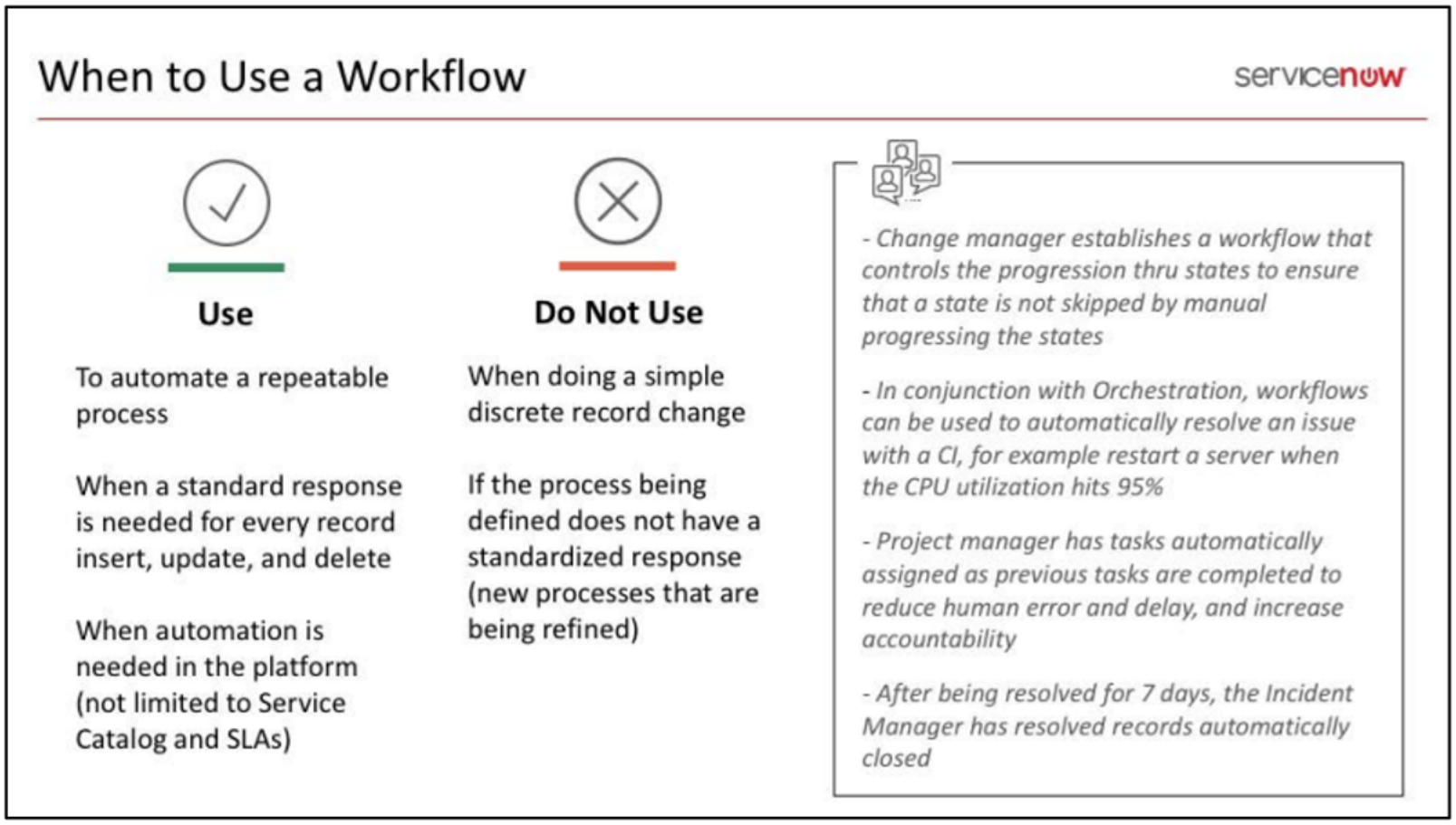
Each workflow consists of a sequence of activities, such as generating records, notifying users of pending approvals, or running scripts.

The workflow starts when a triggering event occurs. Common triggers include a record being inserted into a specific table, or a particular field in a table being set to a specified value. For example, you might create a workflow that runs whenever a user requests approval for an item they want to order from the catalog. When an activity completes, the workflow transitions to next activity. An activity might have several different possible transitions to various activities, depending on the outcome of the activity. Continuing the example above, if the user's request is approved, the activity might transition to an activity that notifies someone to order the item; if the user's request is denied, the activity might transition to notifying the user that their request has been denied.

The graphical Workflow Editor provides a drag-and-drop interface for creating workflows and represents workflows visually as a type of flowchart. It shows activities as boxes labeled with information about that activity and transitions from one activity to the next as lines connecting the boxes.

An activity is processed for each step in the workflow (the “behavior” is determined when the activity begins).

When this action completes, the workflow checks each of the activity's conditions. For each matching condition, the workflow follows the transition to the next activity. When the workflow runs out of activities, the workflow is complete.



You can create a workflow with the Workflow Editor to automate a multi-step process.

NOTE: You must have the **workflow_admin** or **workflow_creator** role to use the Workflow Editor.

Workflow Execution

servicenow

Workflows use conditions and a table association to determine when to start, execute, continue, and end

Scenarios when workflows are started:

- Schedule or SLA launch
- Execute on table interaction (conditions are met)
- Invoke from a Script Include or Business Rule
- Associate with a Catalog Item in the Service Catalog
- Call a workflow from a workflow

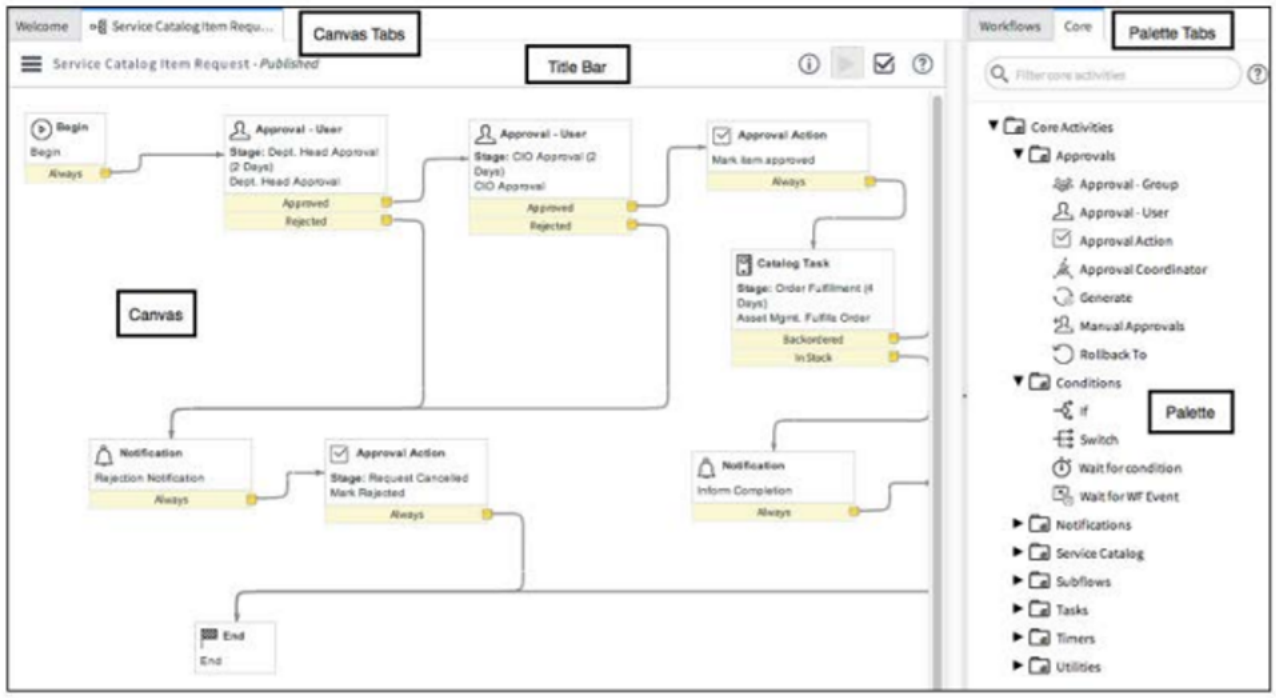
Only published workflows can be instantiated. Unpublished workflows can not be started.

Notes about workflow execution:

- If the processing conditions are met, a workflow can be automatically started on a table record insert or update operation.
- Workflows that are built for the **Requested Item** [**sc_req_item**] table can be manually associated with a Service Catalog Item. Once associated, the workflow will start automatically. This is also true for SLA Workflows.
- The **startFlow(workflowid,current,operation,vars)** method is one method that can be used to call a workflow from a Script Include or Business Rule.
- Workflows that are called from other workflows are called subflows to that workflow.

What is the Workflow Editor?

servicenow



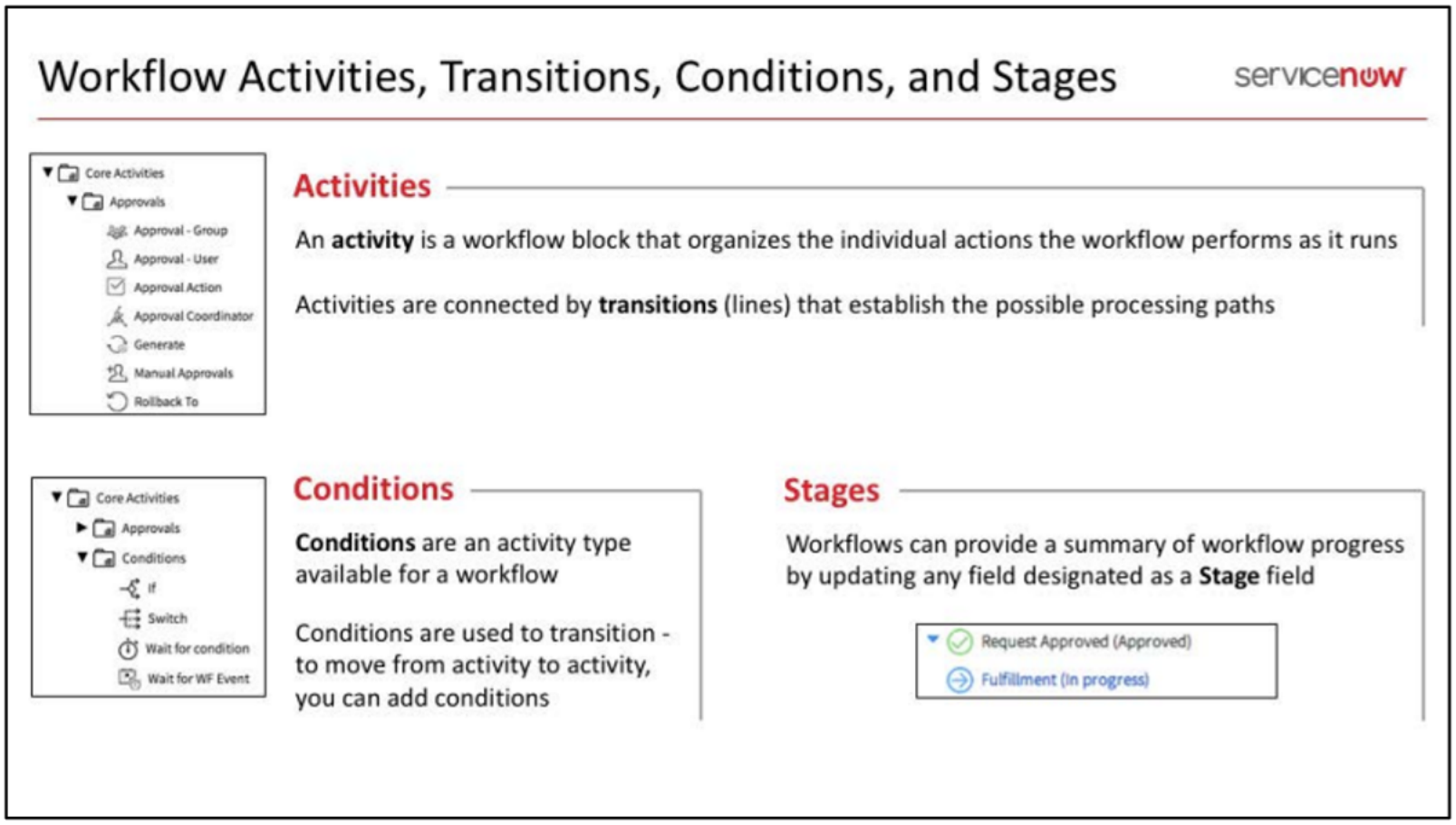
The Workflow Editor is an interface for creating and modifying workflows by arranging and connecting activities to processes. Activities can be added, removed, or rearranged. Transitions can be drawn or changed, and the workflow powers the process as defined visually. All workflows need to have a **Begin** and an **End.**

The Workflow Editor user interface is divided into the following areas:

- **Canvas Tabs**: Contains tabs for accessing workflows being edited or created
- **Title Bar**: Displays the workflow name and status. Provides a menu and controls for configuring, testing, and validating workflows.
- **Canvas**: Provides the working surface for creating new workflows or editing existing ones
- **Palette Tabs:** Contains all available workflow activities and existing workflows you can use as subflows, displayed in the **Palette**. Drag activities and subflows to the canvas to create new workflows or edit existing ones.

Workflows need to be checked out before they can be edited. When a workflow is checked out, changes only apply to the user who has the workflow checked out. Other users can continue to use the published workflow. After the changes are complete, the workflow can be published so that it is available to all users.

To design a workflow model, drag an activity into the Workflow Editor and connect it to other activities by drawing transitions. You can also copy an existing workflow to add, delete, and connect activities. In the Workflow Editor, scrollbars are used to view parts of the workflow and the **Expand Transitions** command to redraw lines.

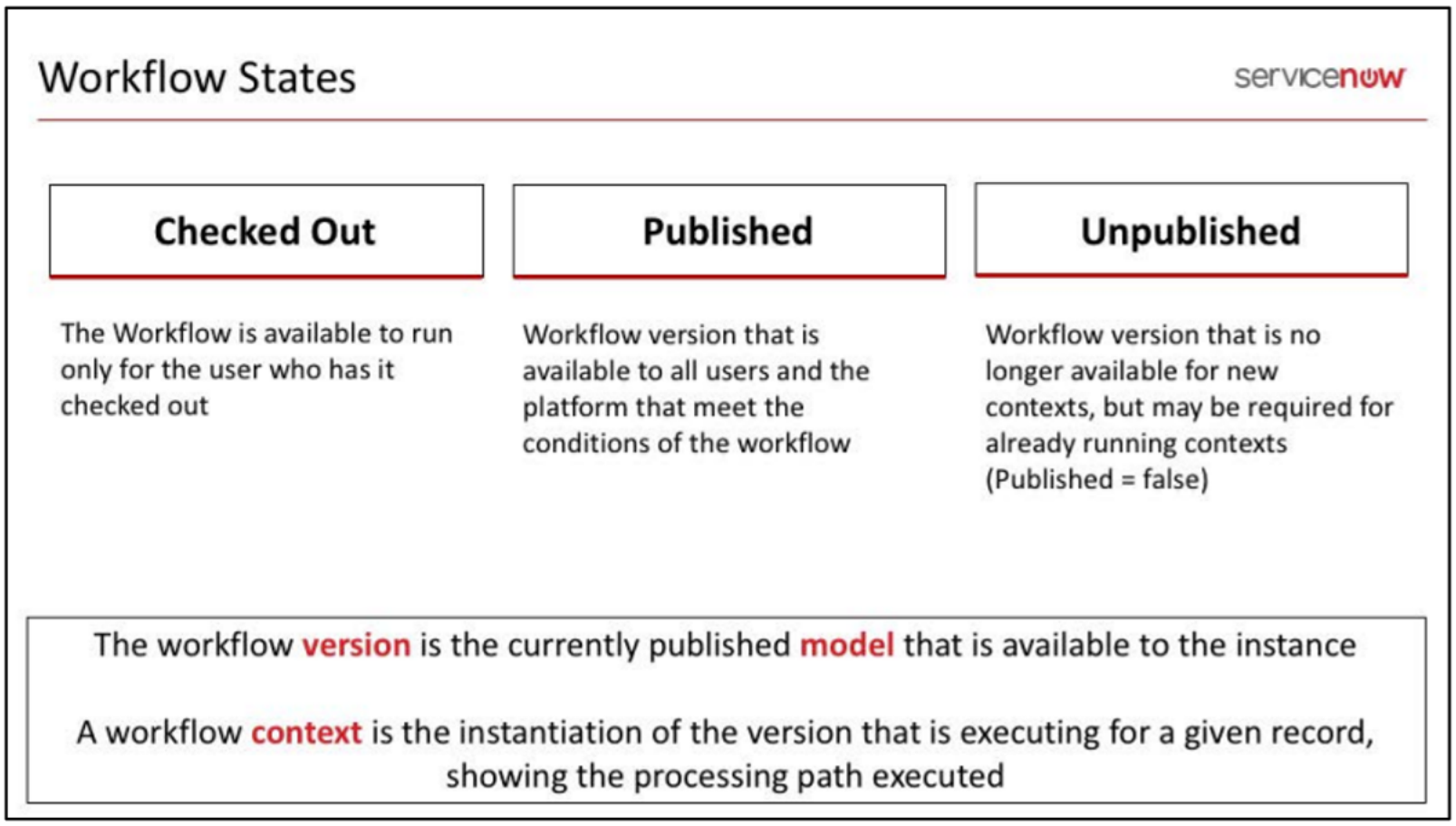


Activities are the organizational element that defines the individual actions a workflow performs as it runs. A **transition**, or line, establishes the processing paths or order in which activities are executed.

The three basic workflow activities are Approval, Notification, and Task. **Approvals** allow approvals to be generated and managed, while driving a record to fulfillment. **Notifications** allow users to be notified of events that occur during the workflow. **Tasks** allow task records to be created and modified. These activities are only available when the workflow is defined to run on a table which extends Tasks. Other activities include **Timers** which pause the workflow for a set period of time and **Utilities** which provide useful controls over the path of the workflow. Plugins can add specialized activities to the Core Activities list and custom activities can be created.

Conditions are the start of a transition to the next activity in the workflow model, and all activities (except End) use conditions to route processing. Add conditions to activities by right-clicking the activity header. Approved, Rejected, Always, Skip and Error are common conditions.

Stages show workflow progress, but work a little bit differently for workflows that run against **Service Catalog Request Items** [**sc_req_item**]. Still, add a designated stage field to any table (for example, **Incident [incident]**) for progress summaries.



The **version** is the published model of the workflow. The published version is available to the platform to use for creating new workflow contexts.

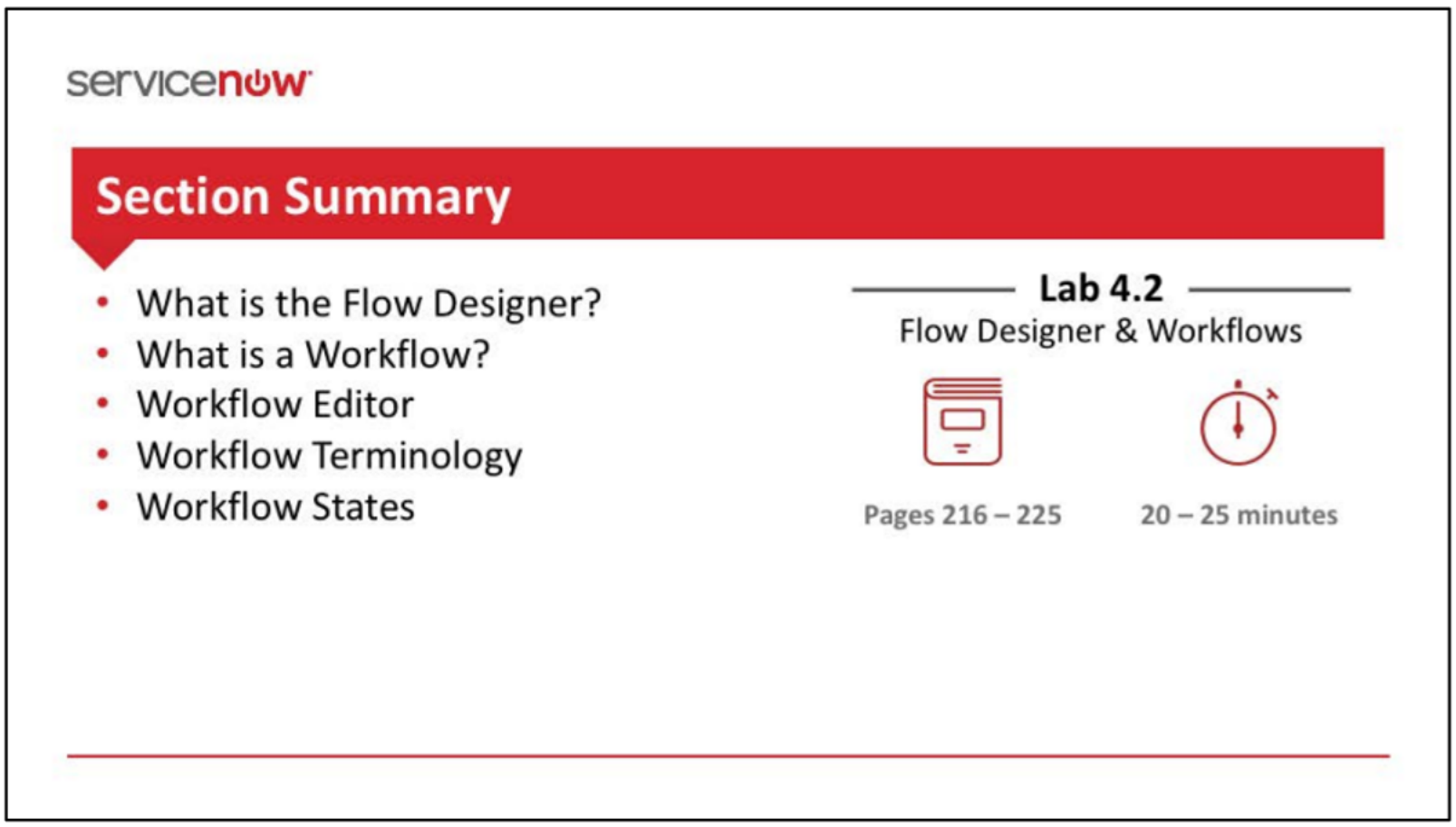
A workflow version maintains these properties:

- The workflow table association
- The conditions, published status, and permissions of the workflow
- The workflow model, which is the workflow in its entirety as the set of workflow activities and their transitions (lines)

When a workflow version record is updated and the State changes to **published**, a table insert action occurs and adds the version record to the platform's current Update Set. Only published workflows are captured in Update Sets.

NOTE: There can only be one active published version of a workflow at a time and past versions are kept as unpublished so there is a historical record. Additionally, executing workflows will not be affected when a new version is published.

Contexts can use different versions of the same workflow, and all of these contexts can be running at the same time.



Lab 4.2 – Flow Designer & Workflow:

- Copy and edit an existing workflow to create a new workflow
- Validate and publish the workflow
- Associate the workflow to a Service Catalog item
- Test the workflow
- View the workflow history

Flow Designer & Workflows

LAB
4.2

20 – 25 minutes

Lab Goal

Lab Dependency: Requires the completion of Lab 4.1.

This lab will show you how to do the following:

- Copy and edit an existing workflow to create a new workflow
- Validate and publish the workflow
- Associate the workflow to a Service Catalog item
- Test the workflow
- View the workflow history

Now that Cloud Dimensions has opened up enrollment for testing the Infinity across the entire organization and the Infinity has been created in the Service Catalog, it is time to implement procurement automation through a workflow.

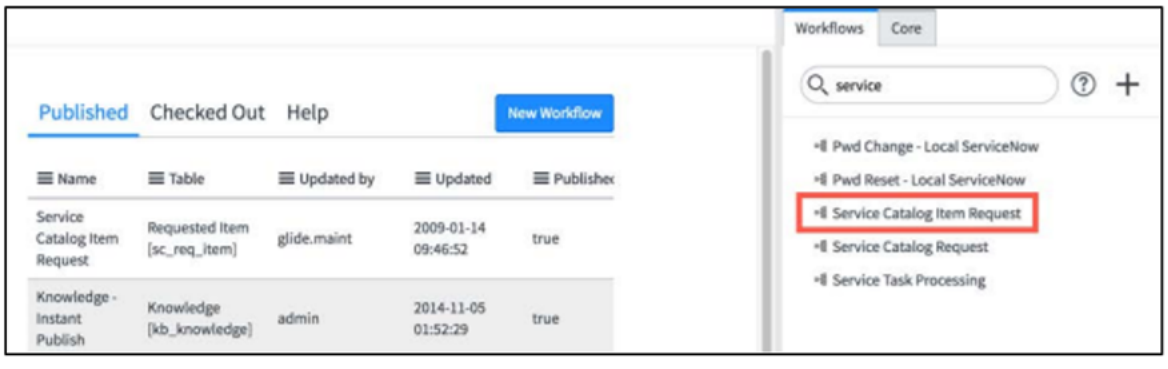
There is a baseline workflow, **Service Catalog Item Request**, that the system administrator will copy and modify to fit the processes in place at Cloud Dimensions, as it relates to order fulfillment and deployment of employee Infinity devices.

A. Copy and Modify the Service Catalog Item Request Workflow

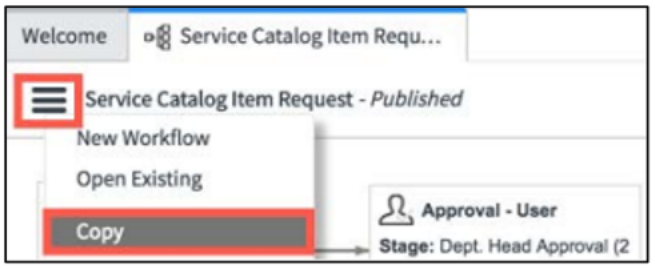
1. As **System Administrator**, navigate to **Workflow > Workflow Editor.**

   NOTE: *The Workflow Editor opens and displays in a separate tab or browser window.*

2. From the Workflows tab on the right, select **Service Catalog Item Request**:



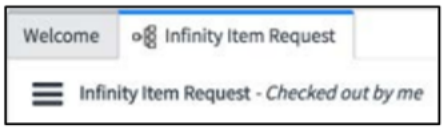
3. From the **Workflow Actions** menu, select **Copy**:



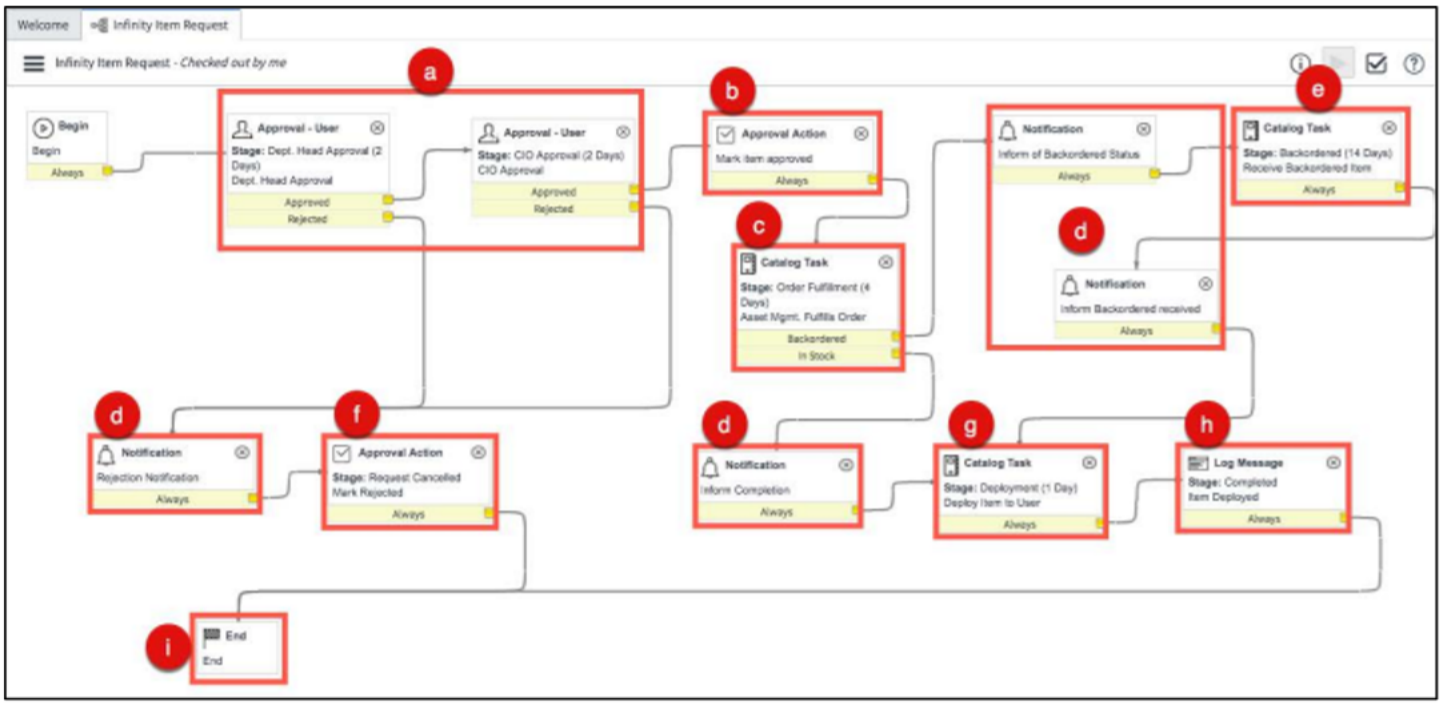
4. Name your workflow **Infinity Item Request**.

5. Click **OK**.

   Verify the workflow tab displays the title **Infinity Item Request**:

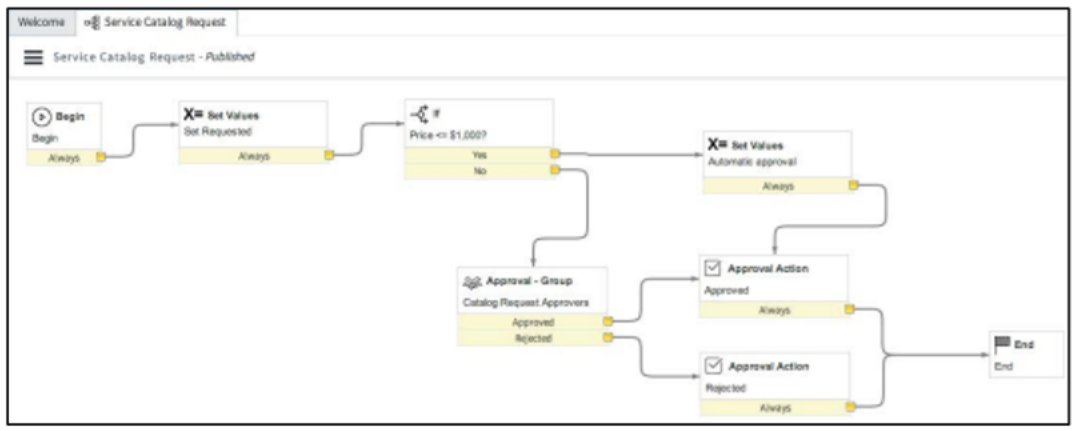


Workflow Overview (What the Checked Out Workflow Looks Like):



a) Sequential Approval Tasks
b) Set the Approval field to Approved
c) Assigned to the Procurement group to backorder or procure
d) Notify the Requested for user
e) Assigned to the Procurement group to check if backorder was received
f) Set Approval field to Rejected
g) Assigned to the Field Services group to deliver
h) Writes to the Workflow Log
i) Automated workflow stop

NOTE: *All new Service Catalog requests have an initial approval rule that is evaluated by a separate Workflow,* ***Service Catalog Request****, shown here:*



If the request amount is under 1,000 it is automatically approved. This automated approval then instantiates any Workflow associated to requested items of the request.

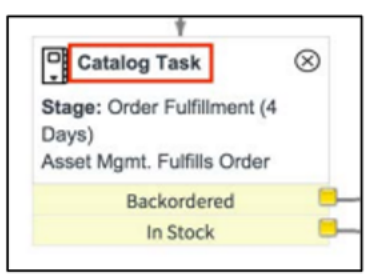
In the **Service Catalog Item Request Workflow**, which you copied and will now modify in this lab, there are two User Approval Activities: Department Head and CIO. These are approval tasks for the item the Workflow is associated with (Infinity for example). A successful Department Head approval then triggers the CIO approval which, when successful, triggers the next Workflow activity, Mark item approved, and so on.

When the item is marked approved, then the first Catalog Task, Order Fulfillment, triggers. You will customize the Order Fulfillment activity in the lab. If the Infinity is in stock, a notification is sent. The second Catalog Task, Deployment, will end the Workflow after the task is completed successfully.

Customize the Workflow

Begin by customizing the two Catalog Tasks: add a description of the steps involved for Order Fulfillment and Deployment, define the groups responsible, and customize the names to align with Cloud Dimensions processes.

1. Double-click the Catalog Task with the Stage: **Order Fulfillment (4 Days)** and the Name: **Asset Mgmt. Fulfills Order**:



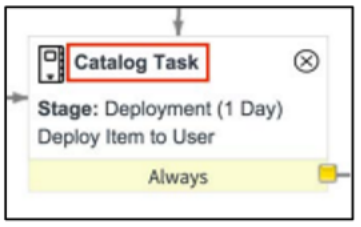
2. Modify the Name to: **Infinity Order Fulfillment**.

3. Under the **Populate task variables** section, change the **Procurement** to **Service Desk**.

4. Modify the **Short description** field, overwriting existing values with the following values: **Tasks to fulfill an Infinity**

5. Click **Update**.

6. Double-click the Catalog Task with the Stage**: Deployment (1 Day)** and the Name: **Deploy Item to User**:



7. Modify the Name to: **Infinity Delivery**.

8. Change the Fulfilment group to **Service Desk**.

9. Modify the form fields, overwriting existing values with Infinity values shown below:

   Short description: **Deliver Infinity**
   Instructions: **Deliver Infinity to requester or requester's manager**.

10. Click **Update**.

Validate and Publish the Infinity Item Request Workflow

1. Validate the Workflow by clicking the **Validate** icon (Checkmark):

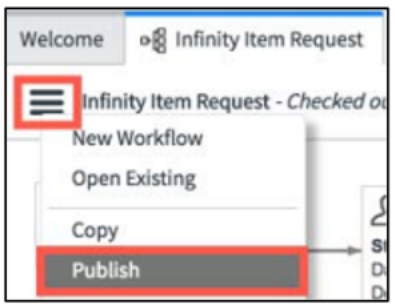
2. A Workflow Validation Report displays.

3. Close out of the Workflow Validation Report window.

NOTE: *If you have warnings or critical errors, correct them before continuing.*

---

4. Next, publish the Workflow to make it available for other users to checkout; from the **Workflow Actions** menu (which was used to copy a Workflow before), select **Publish**:



5. The workflow state updates in the header to ***Published***.

B. Associate Workflow to Service Catalog Item

In this next step, associate the Infinity Item Request workflow to the Infinity Service Catalog item created in the last lab.

1. Close the **Workflow Editor** tab (or window) to return to the main ServiceNow interface.
2. **Service Catalog > Catalog Definitions > Maintain Items.**
3. Search for and open the **Infinity** catalog item.
4. In the Workflow field, select **Infinity Item Request**.
5. **Update**.

C. Test Workflow – Order an Infinity

In this next step, you will impersonate **Joe Employee** and request an Infinity from the Service Catalog.

1. Impersonate **Joe Employee**.
2. **Self-Service > Service Catalog**.
3. Click the **Hardware** Category, then select **Infinity**.
4. You can request either **color**, and, if desired, additional **memory**.

***NOTE**: If your order amount is greater than 1,000, additional approvals (not documented here) are required.*

---

5. Click the **Order Now** button to make the request, and initiate the workflow.

6. Verification that the request was **submitted** is displayed at the top of the form.

7. Record the Request number here: **REQ_____**.

8. Click the Infinity item description and record the Requested Item number here: **RITM_____**.

Test the Workflow – User Approval

Next, review the progress of the Infinity Item Request Workflow, then approve the Department Head and CIO User Approval activities by impersonating the System Administrator. You could impersonate the individual users but this is quicker.

1. Impersonate **System Administrator**.

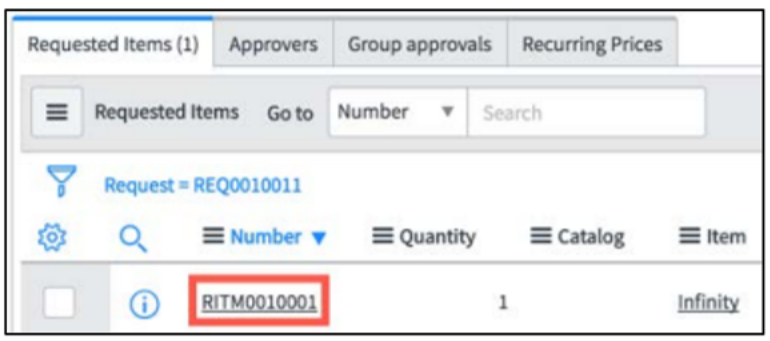
2. **Service Catalog > Open Records > Requests**.

3. Open the Infinity Request created with Joe Employee's account.

***NOTE**: The REQ record is the parent record of the request.*

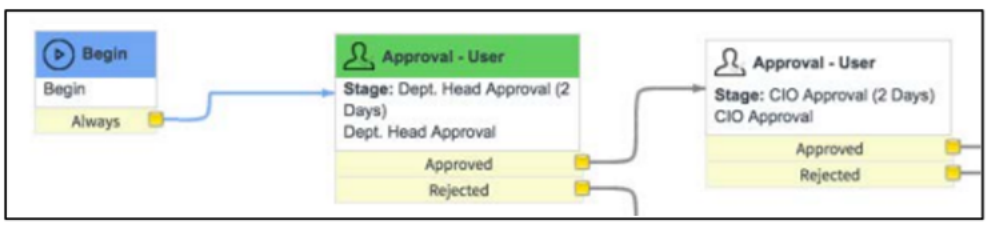
---

4. In the Requested Items tab, click on the Requested Item Number (**RITMnnnnnnn**):



5. Scroll down to Related Links and click **Show Workflow**.

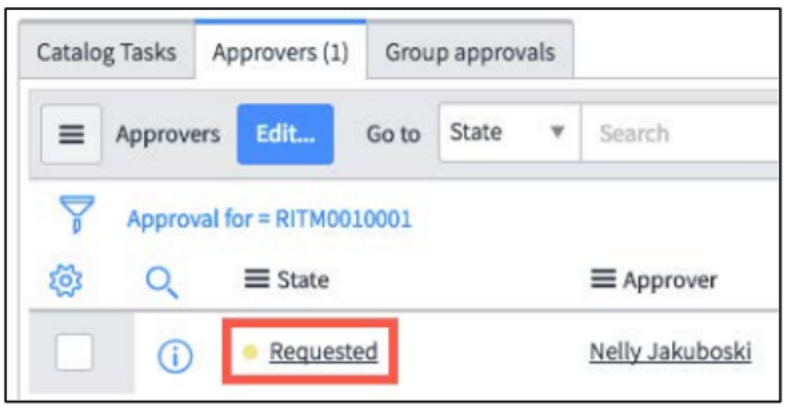
   This opens up summary of the running workflow context for the Infinity request:



   Blue indicates activities that have been completed and green indicates the activity that the workflow is currently at.

6. Next, return to the main instance tab/window and select the **Approvers** tab on the **RITM** form.

7. Click **Requested** from the State column:



8. Click the **Approve** button to approve the request.

***NOTE**: You have approved this request as the system administrator, but for this workflow, you could also have impersonated Nelly Jakuboski to approve.*

---

9. Now, **repeat steps 6-8** to complete the second approval, CIO Approval, on Bow Ruggeri's behalf.

   TIP: From the RITM Approvers tab, you can also right-click on the Requested record, then select **Approve** from the context menu as opposed to opening the approval record.

10. Impersonate **Joe Employee**.

11. **Self-Service > My Requests**.

12. Open your Infinity Request.

13. Scroll down to the Requested Items section and expand the **Stage** field by clicking on the ">" icon:

Notice the current Stage is set at **Order Fulfillment**.

Test the Workflow – Complete Fulfillment and Deployment Tasks

Next, impersonate Kevin Edd to approve the Order Fulfillment task since he belongs to the **Service Desk** group and is responsible for completing the next phase in the Workflow.

1. Impersonate **Kevin Edd**.
2. **Service Desk > My Groups Work**.
3. Locate and open the Service Catalog Task associated with the Fulfillment workflow activity.

   HINT: Search the Short description column for **Infinity**.
4. As Kevin Edd, assume the four items listed in the Description field are complete – click the **Close Task** button.

Challenge

View the progress of the Stage field for Joe Employee's Infinity request:

Test the Workflow – Complete the Deployment

Service Desk is also responsible for delivering the Infinity once it is received so, as Kevin Edd, close the deployment task.

1. If needed, impersonate **Kevin Edd**.
2. **Service Desk > My Groups Work**.
3. Locate and open the Service Catalog Task associated with the Deployment workflow activity.

   HINT: Search the Short description column for **Infinity**.
4. Assume the Infinity has been delivered – click the **Close Task** button.

Test the Workflow – Verify Request Complete

1. **Service Catalog > Open Records > Requests**.
2. Confirm that the **REQ** is no longer in the list.

LAB VERIFICATION

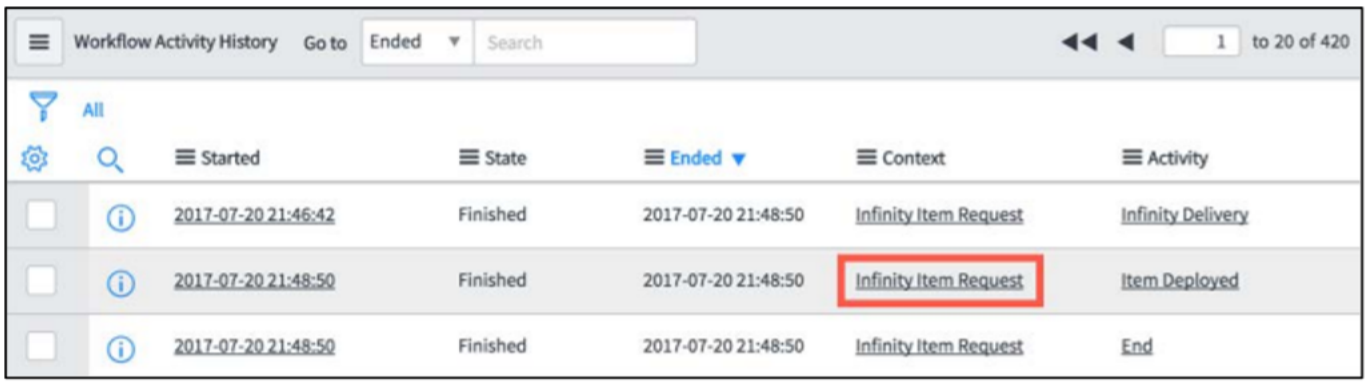
Final Verification: View Workflow History

1. Impersonate **System Administrator**.
2. **Workflow > Live Workflows > History**.
3. Click the **Ended** column header to date sort most recent at the top.

***NOTE**: The history for all requests and their workflow steps are displayed in a single list view. The list can be personalized to add other fields or a filter can be created to view only steps associated with a particular workflow or workflow activity.*

---

4. Click the **Infinity Item Request** title in the **Context** column where the **Activity** is Item Deployed:

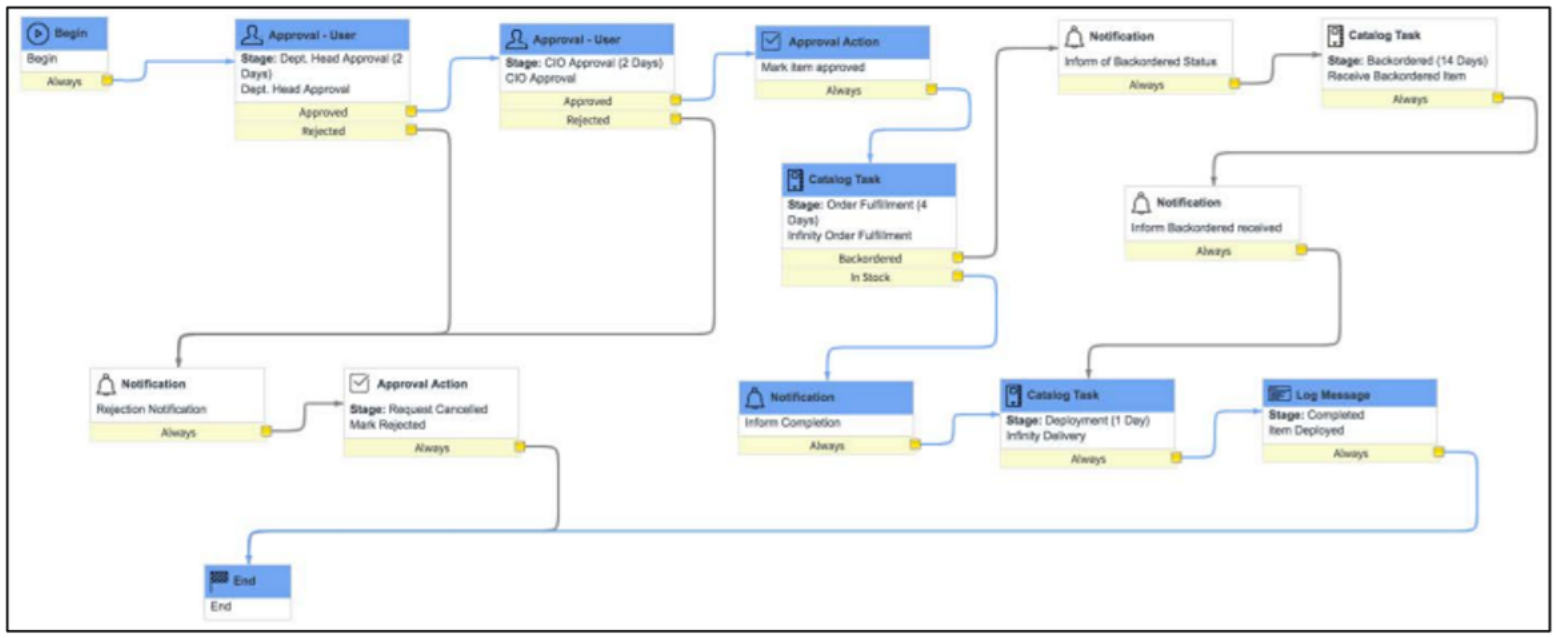


***NOTE**: This workflow context shows all of the workflow steps completed for the Infinity request; ordered, approved, fulfilled, and delivered.*

---

5. Select the **Workflow Activity History** tab and notice the nine Activities.

6. Under Related Links, click **Show Workflow** to show which path the workflow executed for the Infinity catalog item request:



Impressive!

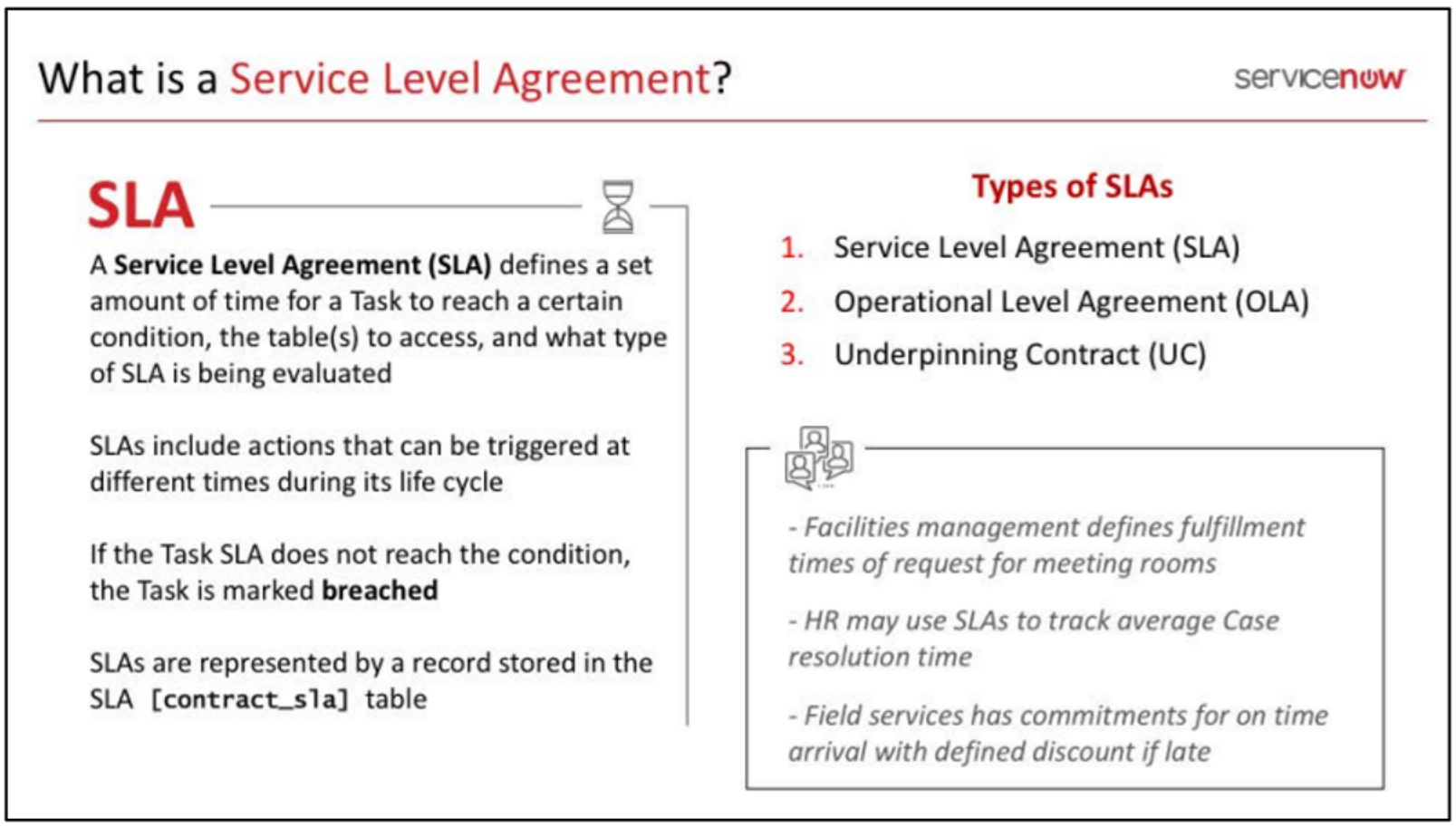
In this lab you took an existing workflow and modified it to fit within a Service Catalog item fulfillment and delivery process.

Module 4 – 4.3 Service Level Agreements

servicenow

Objectives

- What is a Service Level Agreement?
  - SLA Types
  - Definition and Conditions
- Default SLA Workflow



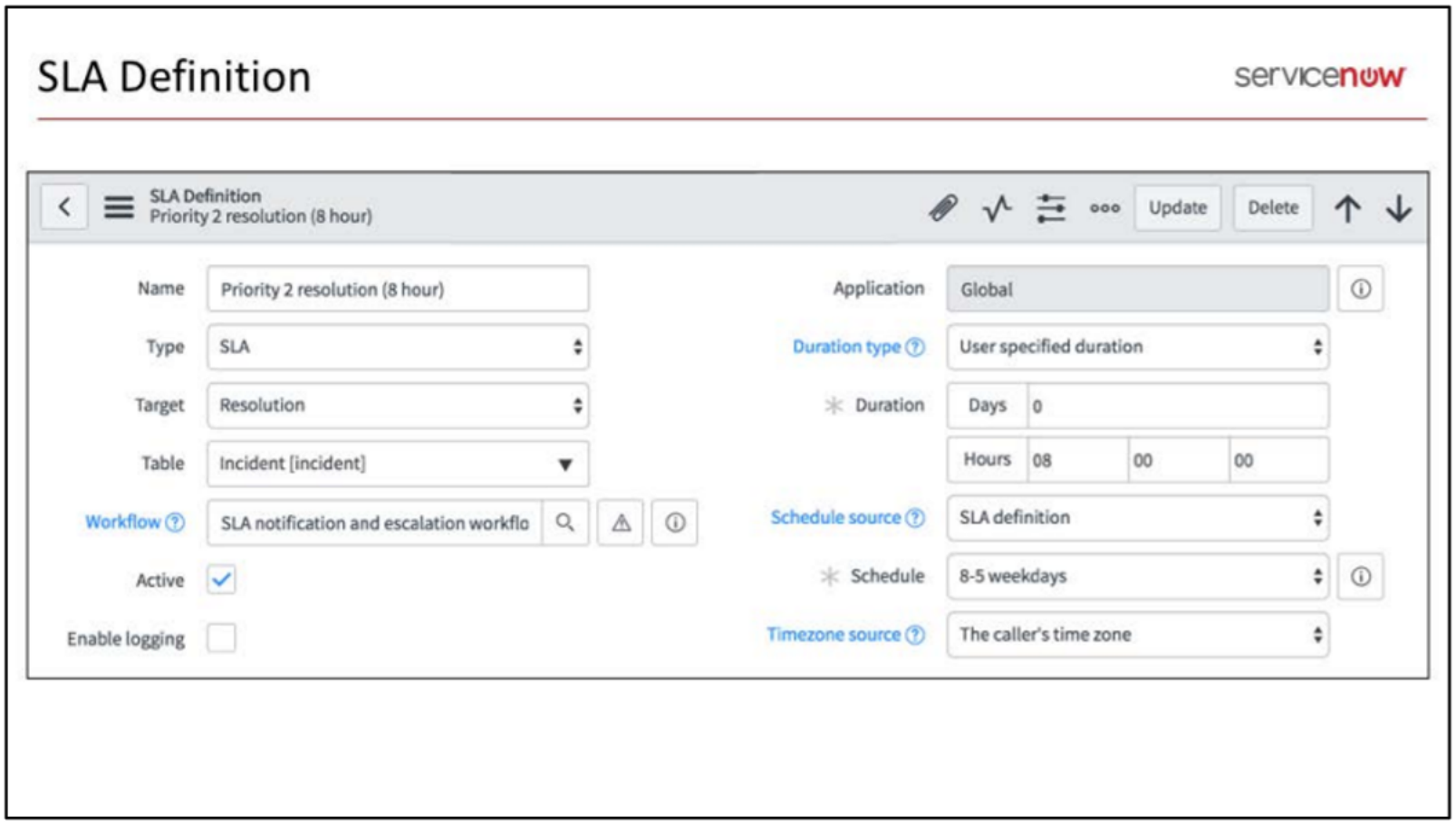
Several major components work together to power the Service Level Agreements plugin:

- **SLA Definition**: The record which defines the conditions that trigger the SLA
- **Task SLA**: The individual instances of the SLAs associated with particular tasks
- **SLA Workflow**: Workflow powers events or actions based on the SLA definition and is designed to be used over and over
- **SLA Automation**: The Business Rule and Scheduled Job that automate the SLA
- **SLA Conditions and Script Include**: A Script Include and reference record that can be used to customize the transitions between different SLA states

The **Task SLA [task_sla]** table stores each of the individual SLAs attached to particular tasks. Unlimited SLAs can be running against a record but SLAs can only run against a table that extends the Task table.

Types of SLAs: While each type of Service Level Agreement may involve different stakeholders, their basic structure in the tool is the same, they track things you want tracked. The only difference between SLAs, OLAs, and Underpinning Contracts is the Type field on the Task SLA form. These SLA types basically all behave the same way.

- An **Operational Level Agreement (OLA)** defines how departments work together to meet the service level requirements documented in an SLA
- An **Underpinning Contract (UC)** is a type of SLA that defines and monitors the guarantees established with an outside supplier; it is a tool for supplier management



When defining a Service Level Agreement, there are many important fields including:

- **Duration type:** You can choose a specific duration from the list, or you can define your own. The SLA performs the calculations and sets a day and time as the deadline for the SLA. Typically, you would apply **User specified duration**.
- **Duration**: When **User specified duration** is selected from the **Duration Type** list, an administrator can define the number of days and hours of the timer for the SLA.
- **Schedule**: Defines what is calculated as hours in a work day and number of days in a work week to use for the SLA. Before you begin to look at how to calculate conditions for the scheduling of the SLAs, you can check to see what platform schedules have been set up for hours per work day and the number of days in the work week.
- **Timezone source**: Specify the time zone for the SLA. The SLA definition's time zone is used when creating Task SLAs if the **Use the following time zone for SLA** property is selected in **Service Level Management > Properties > SLA Engine**. The time zone can be definition, schedule, location, or configuration item

SLA Definition: Conditions

servicenow

Start

Enables you to define the conditions under which the SLA will be attached

Retroactive start, when activated, works with the **Set start to** field and calculates the SLA start time

For example, an incident is received saying that email is down; an hour later the incident is updated with the email server being offline, but the start time is the same for both actions

Pause

Enables you to define the conditions under which the SLA will suspend increasing elapsed time

Use the **When to resume** drop-down list to choose the condition under which the SLA will resume increasing elapsed time

Stop

Enables you to define the conditions under which the SLA completes

If all of the specified stop conditions match, then the task SLA will complete regardless of whether it is breached

Reset

Enables you to define the conditions under which the running SLA will be completed and a new SLA will be attached

For a new SLA to be attached, the start condition must match

While defining an SLA, you can set up to four SLA conditions: Start, Pause, Stop, and Reset.

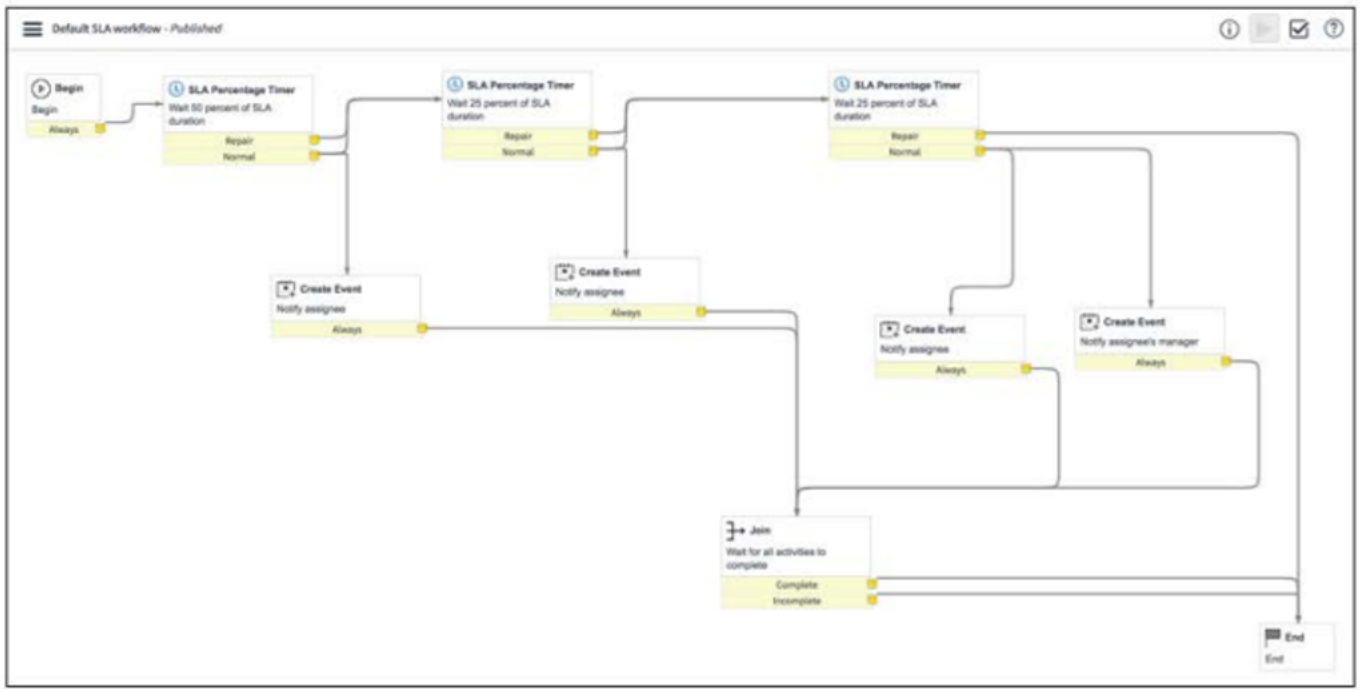
SLA Definition Evaluation - Every task in the platform is evaluated in the following order:

- Process new SLAs to determine if a new SLA record must be attached to a task
- Process existing SLA records attached to a task

SLA Condition Evaluation:

- **Attach** if start condition matches and both the stop and cancel conditions don't match
- **Complete** if the stop condition matches
- **Pause** if the pause condition matches
- **Resume** if the pause condition doesn't match or resume condition matches
- **Reattach** if both the reset and the start conditions match
- **Cancel** if the start condition doesn't match or cancel conditions matches

Default SLA Workflow



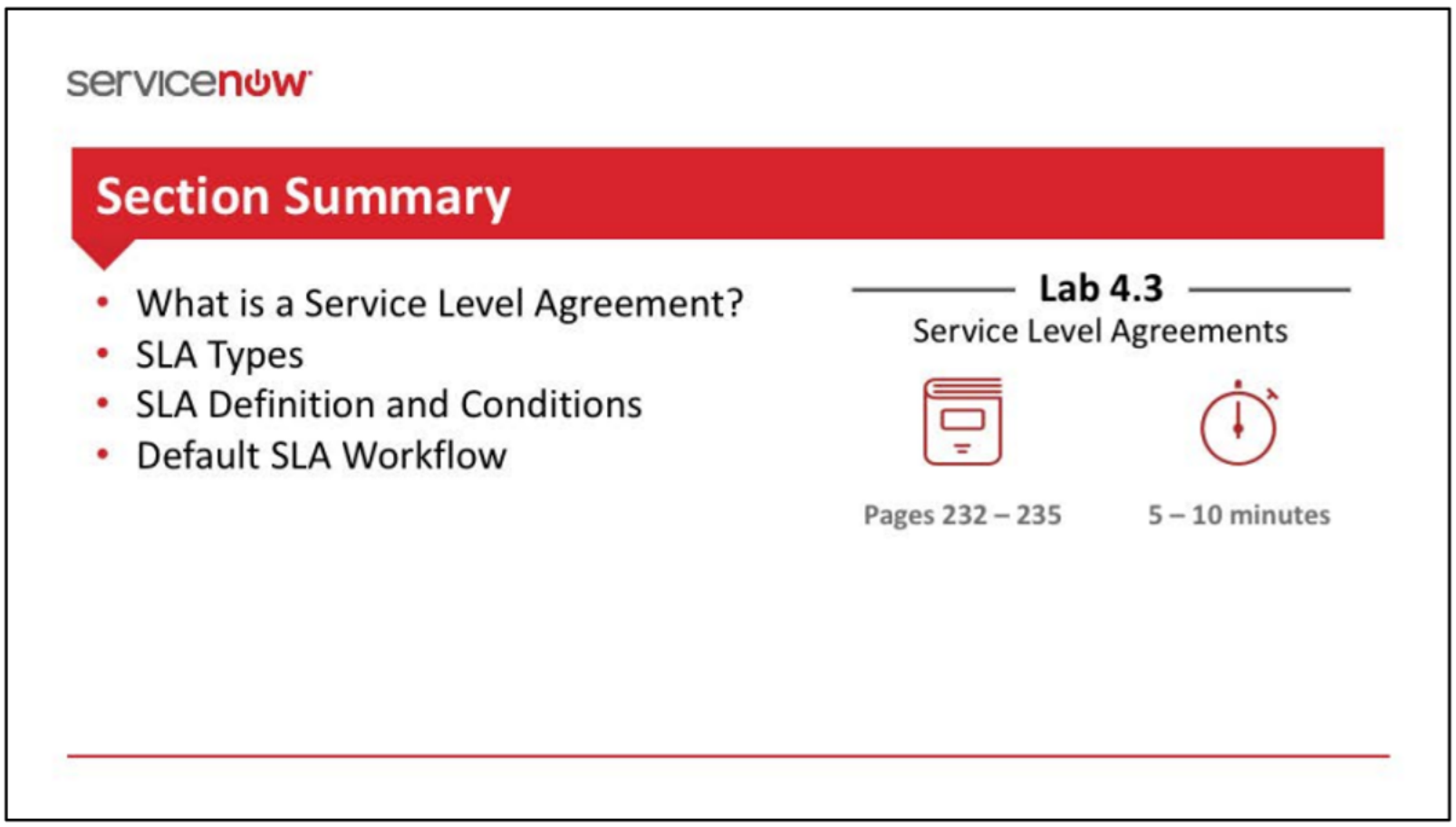
As you saw earlier, ServiceNow also provides Workflows for Service Catalog Requests, Service Catalog Item Requests, and other types of Workflows, such as Routine Change and Emergency Change.

The **Default SLA Workflow** is designed to be used with multiple service level agreements of any types. This workflow will create events that send out notifications automatically after a task reaches 50%, 75%, and 100% of its allotted SLA duration.

There is also the **SLA Notification and Escalation Workflow** which will create events that send out notifications automatically. When a task reaches 50% of its allotted SLA duration, for example, the workflow will trigger a notification to be sent to the assignee and the user listed in the Supported by field on the configuration item. At 75% and 100%, a notification is sent to the assignee and the assignee's manager.

SLAs allow an IT service desk to track if their representatives are providing a specific level of service, and run reports on the success rates of the SLA actions.

For example, notify the manager when the SLA reaches 75% of its allotted time. Most commonly used to ensure incidents are being resolved within a certain amount of time.



Lab 4.3 – Service Level Agreements:

- Create an SLA for Infinity security incidents
- Test the SLA

LAB

Service Level Agreements 4.3

5 – 10 minutes

Lab Goal

This lab will show you how to do the following:

- Create an SLA for security incidents
- Test the SLA

Buster Wubbel and Winnie Reich have agreed upon a reasonable duration for how long it should take to resolve all security incidents, including Infinity incidents.

They have asked the system administrator to create a new SLA that meets their requirements.

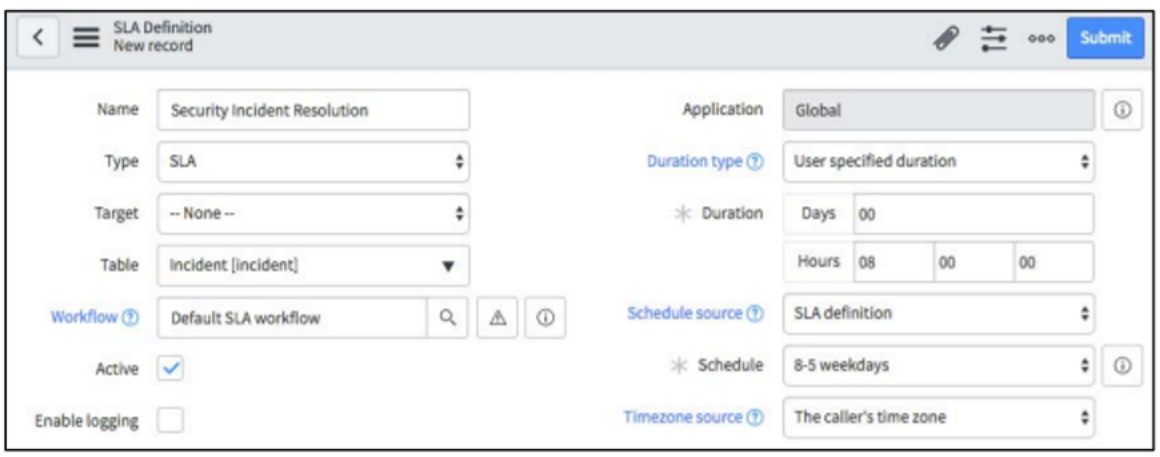
A. Define an SLA for Security Incidents

1. **Service Level Management > SLA> SLA Definitions**.

2. Click **New**.

3. Complete the form as shown:

   Name: **Security Incident Resolution**
   Table: **Incident [incident]**
   Duration: **Days 00 Hours 08:00:00**
   Schedule: **8-5 weekdays**



4. Add the **Start conditions** as shown:

   Active | is | true **AND**
   Category | is | Security

5. Select the **Retroactive start** checkbox.

***NOTE**: Doing so will display the **Set start to** field.*

---

6. Choose **Created** for the **Set start to** field.

7. Change the value of the **When to cancel** field to **Start conditions are not met**:

8. Select the **Stop condition** tab.

9. Add the following **Stop condition**:

   State | is | Closed **OR**
   State | is | Resolved

***NOTE**: This SLA will take effect for any incident submitted with a category of Security, and it will track time until the incident reaches the state of Closed or Resolved.*

---

10. Click **Submit**.

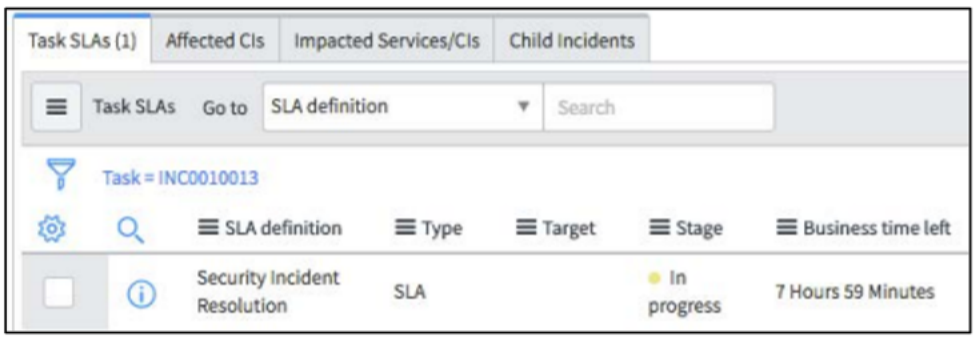
B. Test the Service Level Agreement

1. Impersonate **Winnie Reich**
2. **Incident > Create New**.
3. Fill out the form as follows:

   Caller: **Winnie Reich**
   Category: **Security**
   Assignment group: **Service Desk**
   Short description: **Testing Security INC SLA**

4. **Save**.
5. Scroll down to the **Task SLAs** section.

   Notice the Security Incident Resolution SLA has triggered:



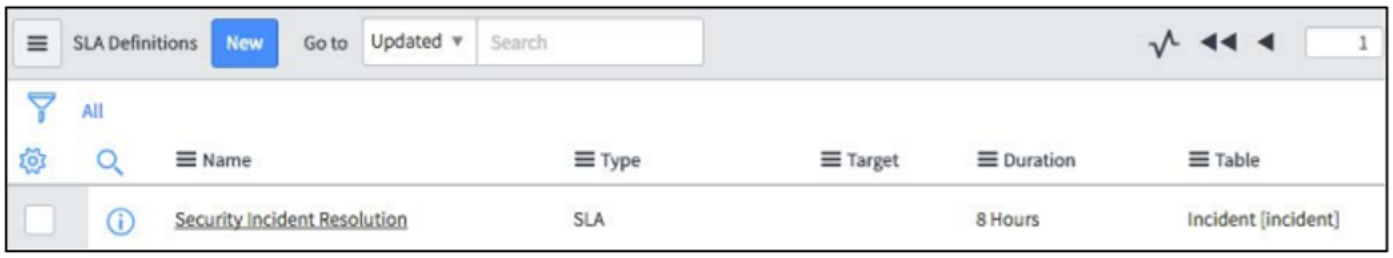
6. Scroll back to the top of the form and update the **State** field to **Resolved**.
7. **Save** the record.
8. From the Task SLAs section, select the **Preview** icon (circle with an "i") for Security Incident Resolution:

9. Next, click the **Open Record** button on the record preview window.
10. Under **Related Links** on the Task SLA record, click **Show Workflow**.

NOTE: *This Workflow context shows the lifecycle of the SLA for the incident*

LAB VERIFICATION

Service Level Agreement Definition



Well done, you have completed the Service Level Agreement lab!

Module 4 – 4.4 Knowledge Management

servicenow

Objectives

- What is Knowledge Management?
- Knowledge Base Architecture
- Knowledge Base Workflows
- Knowledge Article Security: User Criteria

What is Knowledge Management?

servicenow

Knowledge Management allows users to create, edit, and view Knowledge articles to share information across the organization in a centralized location

Knowledge articles exist within a single Knowledge Base, which is managed by one or more Knowledge Managers

Administrators and those with the **knowledge_admin** role can create multiple Knowledge Bases

- HR administrators can limit access to Knowledge articles with User Criteria, for example Benefits for EMEA employees are only visible to employees who reside in EMEA

- Include HR policies, calendars, and detailed instructions for reporting violations to reduce security cases

- In conjunction with Event Management, Knowledge Base articles containing resolution instructions can be generated from Events to fix an issue with a CI

With Knowledge Management, each organization can have their own Knowledge Base with flexible controls over who can see the information and who can help contribute.

Users can browse and search Knowledge using a Knowledge Homepage or use contextual search from other ServiceNow applications, so users can help themselves troubleshoot their issue.

To view Knowledge content, navigate to **Self-Service > Knowledge** to displays knowledge articles organized by Knowledge Base and Category, as well as Featured Content, and popular articles (Most Useful and Most Viewed).

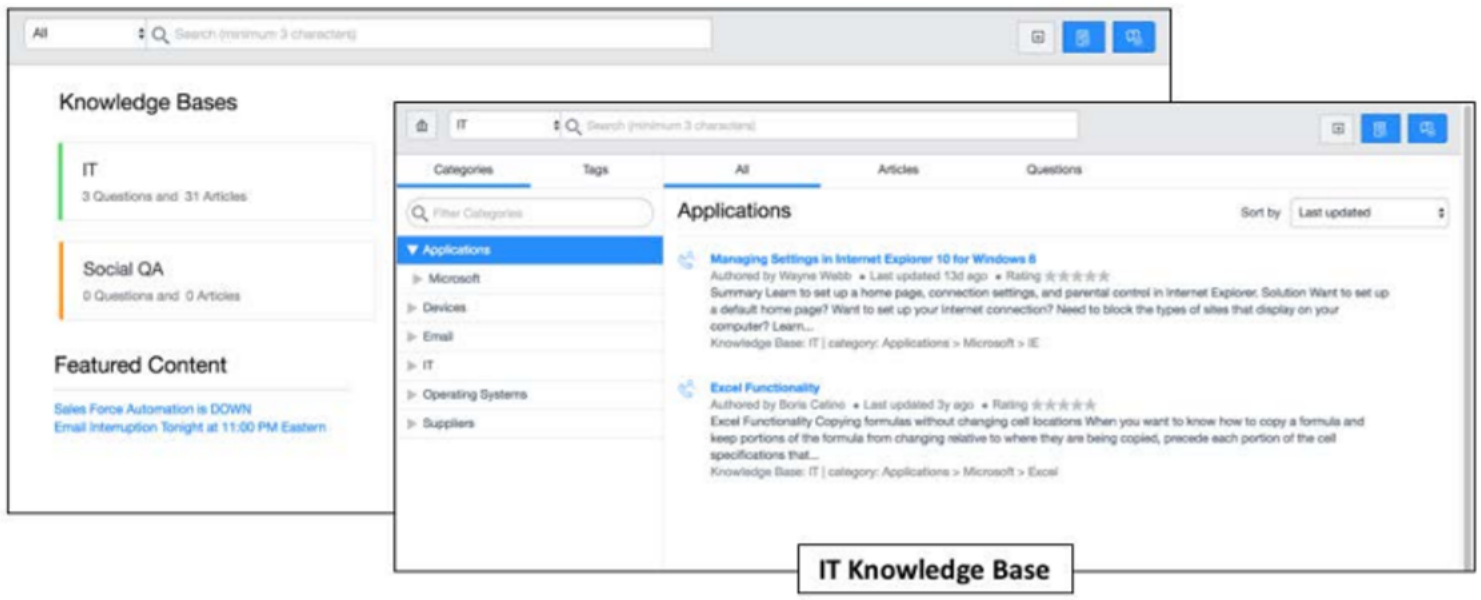
From the Knowledge homepage you can browse and search for articles, then submit feedback on those articles if desired.

NOTE: You must have at least one ServiceNow role to contribute content.

Knowledge Base Architecture

servicenow

The Knowledge homepage displays knowledge articles and social questions (Q&A) organized by **knowledge base** and **category**



From the homepage, users with the correct permissions can import a Word document to a knowledge base using the **Import Articles** button, create a new article using the **Create an Article** button, or ask a question using the **Post a Question** button.

Administrators can create multiple Knowledge Bases and assign them to individual managers responsible for controlling the behavior and organizational scheme of each knowledge base. Each Knowledge Bases can have unique lifecycle workflows, user criteria, category structures, and management assignments.

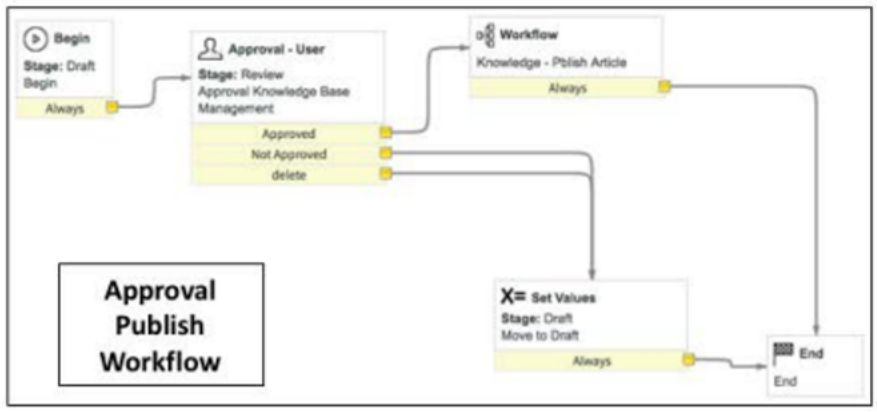
Category Hierarchy:

- Knowledge articles within a Knowledge Base are grouped by category
- Category groups can help you define the Knowledge Base taxonomy, and can help users find articles within a Knowledge Base
- Knowledge Managers can define knowledge categories to pre-populate the list of available categories, and knowledge contributors can select categories, and add or edit categories, if enabled, for a Knowledge Base

Knowledge Base Workflows

servicenow

The publishing and retirement processes for a Knowledge article are controlled by workflows defined for the Knowledge Base that the article belongs to



You can assign different workflows to each Knowledge Base

You can use one of the default workflows, or create your own workflow to define custom publishing and retirement processes for different types of knowledge

The Knowledge Base Workflows available in the ServiceNow baseline instance include:

- **Knowledge – Approval Publish:** Requests approval from a manager of the Knowledge Base before moving the article to the published state. The workflow is canceled and the article remains in the draft state if any manager rejects the request.
- **Knowledge – Instant Publish:** Immediately publishes a draft article without requiring an approval.
- **Knowledge – Instant Retire:** Immediately retires a published article without requiring an approval.
- **Knowledge – Retire Knowledge:** Moves a knowledge article to the retired state.

NOTE: This is only a selection of the base instance workflows to choose from, as designed for Knowledge Base management.

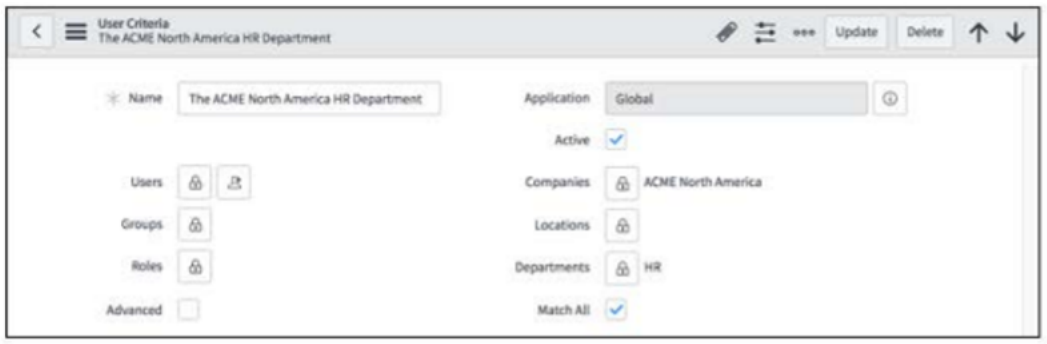
Knowledge Security: User Criteria

servicenow

User Criteria defines conditions that are evaluated against users to determine which users can create, read, write, and retire knowledge articles

You can apply several user criteria records to knowledge content

User Criteria is applied at the Knowledge Base level



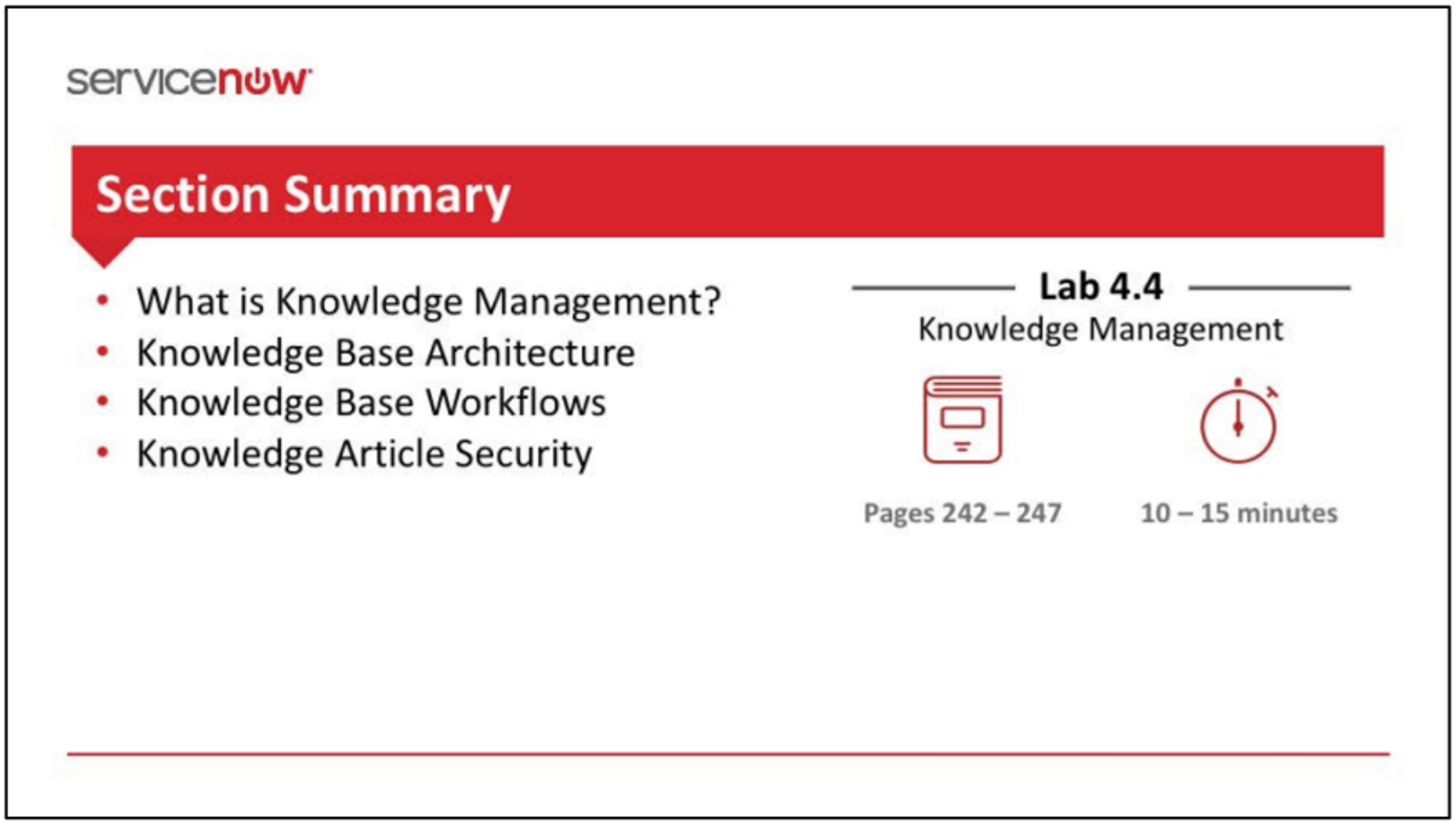
Knowledge Bases use User Criteria records to determine which sets of users can read or contribute knowledge within that Knowledge Base. If a knowledge base has no user criteria selected, articles within that Knowledge Base are available to all users.

User Criteria definitions include:

- **canRead**: users who can read all Knowledge Base articles
- **cantRead**: users who cannot read, create, or modify articles in the Knowledge Base
- **canContribute**: users who can read, create, and modify articles in the Knowledge Base
- **cantContribute**: users who cannot create or modify articles in the Knowledge Base

To implement User Criteria, navigate to **Knowledge > Knowledge Bases** and select a Knowledge Base. Next, access the **Can read** or **Can contribute** related lists to select or create User Criteria records.

When creating User Criteria, the **Match All** check box to determine whether all elements from each populated criteria field must match. If selected, only users who match all criteria are given access. If cleared, the user must meet one or more of the set criteria to be given access. By default, this check box is cleared so that any condition met provides a match.



Lab 4.4 – Knowledge Management:

- Create a Knowledge Base article by importing a Word document
- Approve the article for publishing
- Define, apply, and test user criteria on the knowledge base

LAB

Knowledge Management 4.4

10 – 15 minutes

Lab Goal

This lab will show you how to do the following:

- Create a Knowledge Base article by importing a Word document
- Approve the article for publishing
- Define, apply, and test user criteria on the knowledge base

After a few positive rounds of Infinity testing, the product has been greatly improved and the testing audience is ready to be expanded.

Human Resources (HR) has volunteered to lead an initiative within Cloud Dimensions to advertise open enrollment for Infinity testing using the Knowledge Base.

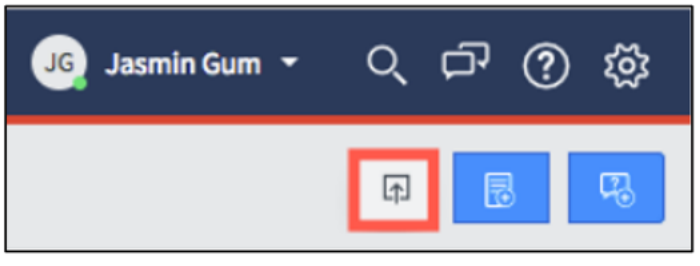
HR representatives will be granted the appropriate authoring permissions by the system administrator so that they may create, review, and publish articles.

The system administrator will also assist with ensuring the article remains secure and accessible only by Cloud Dimensions employees.

Required Resource: Infinity_Open_Enrollment.docx

A. Create a New IT Knowledge Base Article

1. Impersonate **Jasmin Gum**, a member of the Human Resources group.
2. **Self-Service > Knowledge**.
3. Click the **Import Articles** button:



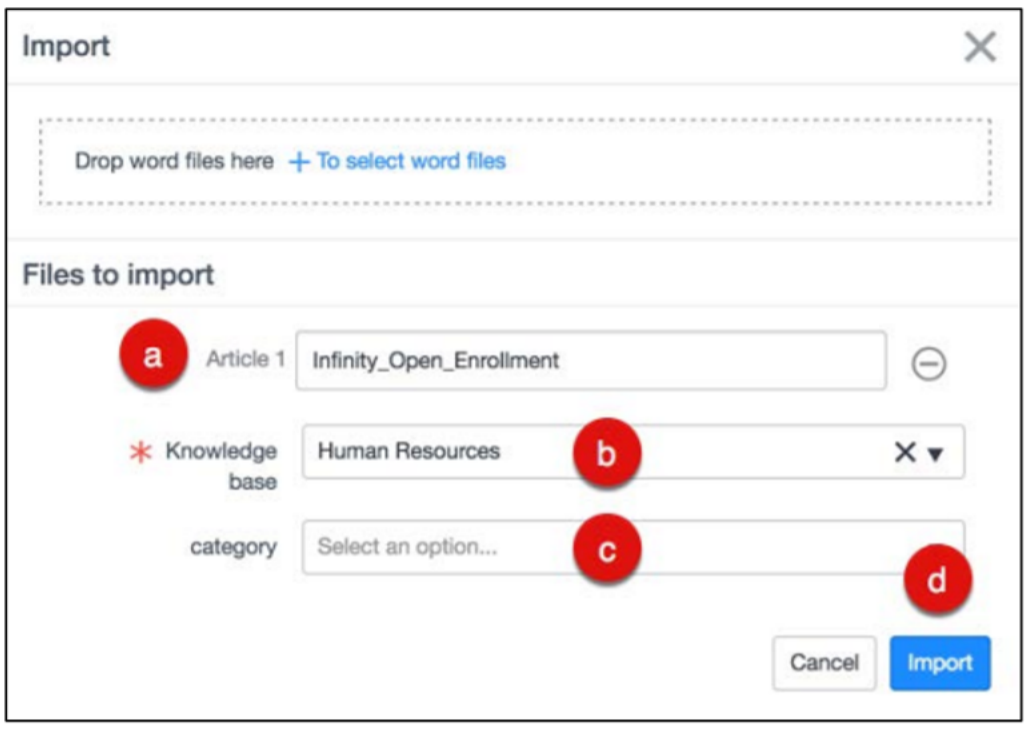
4. Next, click **+ To select word files** and find the **Infinity_Open_Enrollment.docx** file you have downloaded, then fill out the form as shown:

   a) Article 1: **Infinity_Open_Enrollment** (auto fills)

   b) Knowledge base: **Human Resources**

   c) category: **[leave empty]**

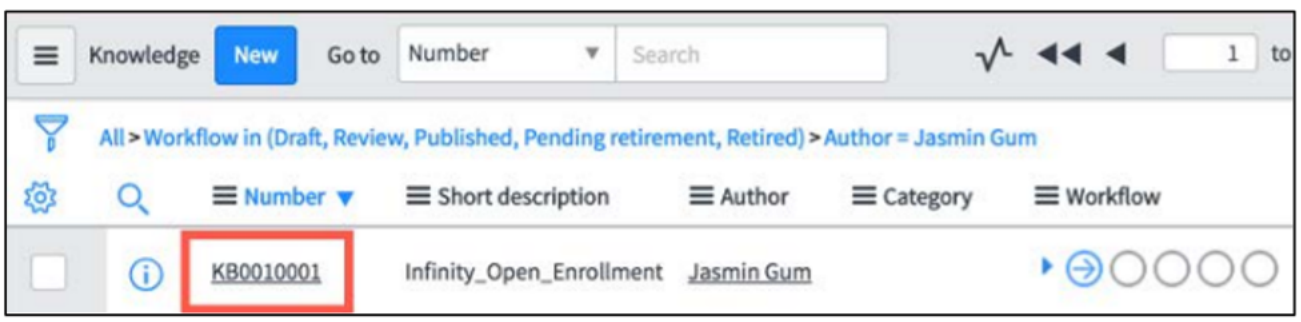
   d) Click the **Import** button



NOTE: *You should receive an "Import completed" message on the bottom of your screen.*

5. **Self-Service > My Knowledge Articles**.

6. Click the knowledge article **Number** for Infinity_Open_Enrollment:



7. Update the **Short description** to **Infinity Testing Open Enrollment**.

8. Click the **Publish** button from the header.

NOTE: *The default publish workflow for the Human Resources Knowledge Base is* ***Knowledge – Approval Publish****. This means after an author clicks the* ***Publish*** *button on their article, it goes into a review state. Other uses with the correct permissions can view the article and determine if any changes are needed before approving and publishing the article.*

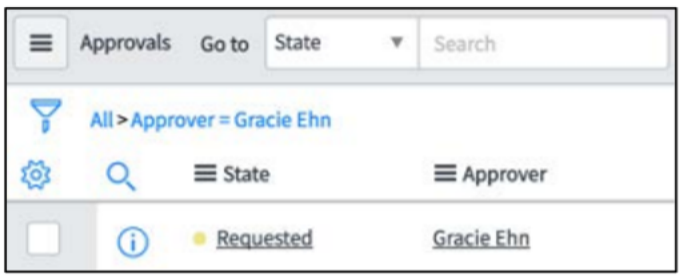
B. Approve the Article for Publishing

1. Impersonate **Gracie Ehn**, manager of Jasmin Gum.

NOTE: *Gracie Ehn is also a member of Human Resources and acts as the review board for new content submitted to be published.*

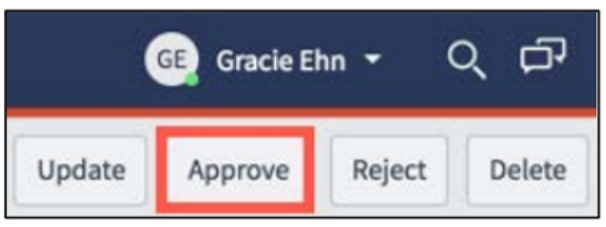
2. **Service Desk > My Approvals**.

3. Locate and open the requested approval record for the Infinity Testing Open Enrollment knowledge article:



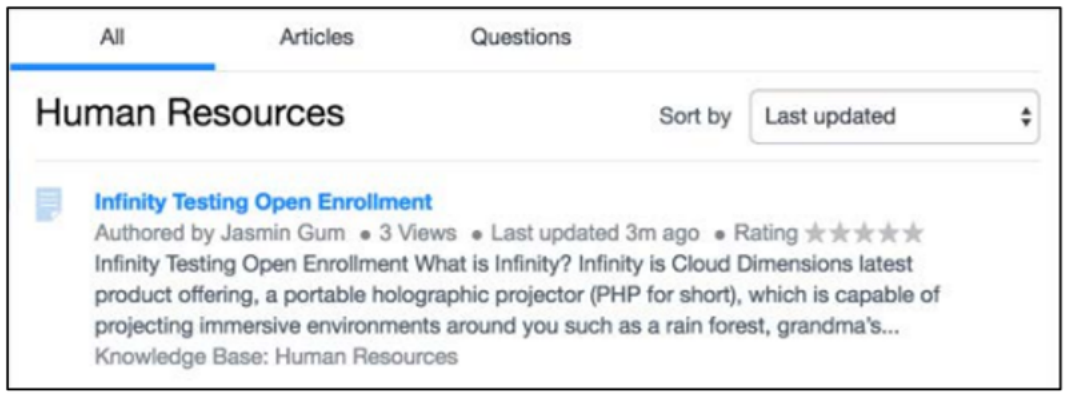
4. Scroll down to see a summary of the item being approved.

5. Assume the content looks good, then click **Approve** from the form header:



6. **Self-Service > Knowledge**.

7. Open the **Human Resources** Knowledge Base to confirm the article appears:



C. Create and Apply User Criteria

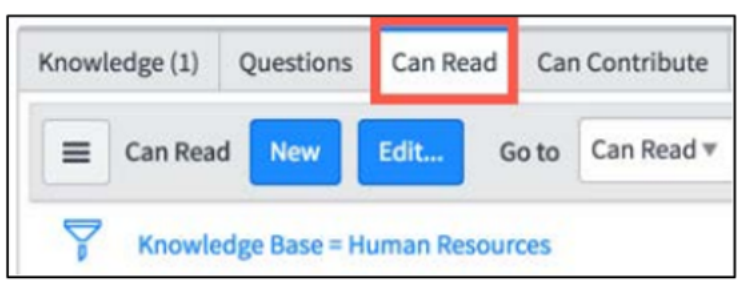
The Human Resources Knowledge Base is currently public to all users that log into the Cloud Dimensions instance and access the **Self-Service > Knowledge** module.

The system administrator will create user criteria and apply it to the Knowledge Base to appropriately control who can view the content.

1. Impersonate **System Administrator**.
2. **Knowledge > Administration > User Criteria**.
3. **New**.
4. Fill out the form as follows:

   Name: **Cloud Dimensions Employees**
   Companies: **Cloud Dimensions**

5. **Submit**.
6. **Knowledge > Administration > Knowledge Bases**.
7. Open the **Human Resources** record.
8. Scroll down and click the **Can Read** tab:



9. Click **Edit...**

10. Add **Cloud Dimensions Employees** to the Can Read List.

11. **Save**.

D. Test User Criteria

1. Impersonate **Jon Floyd**.

NOTE: *Remember, Jon Floyd works for a partner company of Cloud Dimensions. We will use his user account to verify the Human Resources Knowledge Base is unavailable to access because of the user criteria applied to it.*

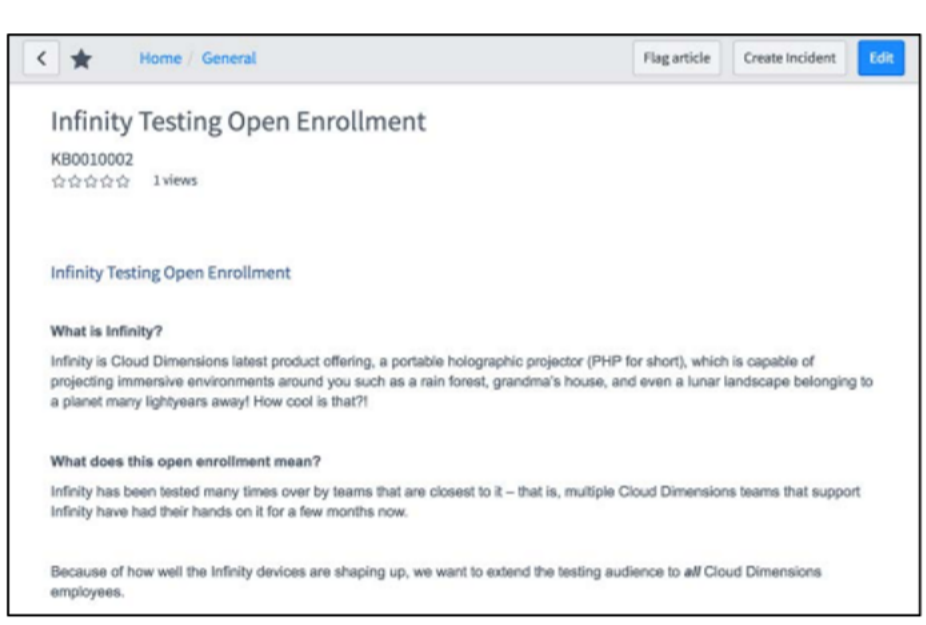
---

2. **Self-Service > Knowledge**.

   The Human Resources Knowledge Base does not appear.

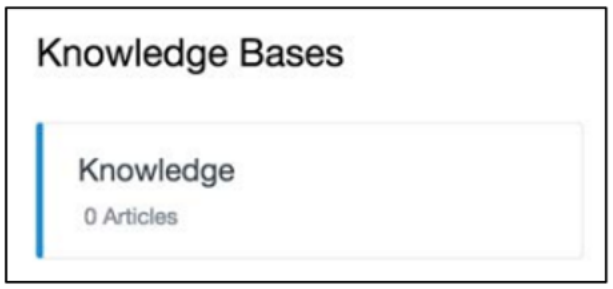
LAB VERIFICATION

Human Resource Knowledge Base Article



User Criteria – Employee Perspective

User Criteria – Non-Employee Perspective



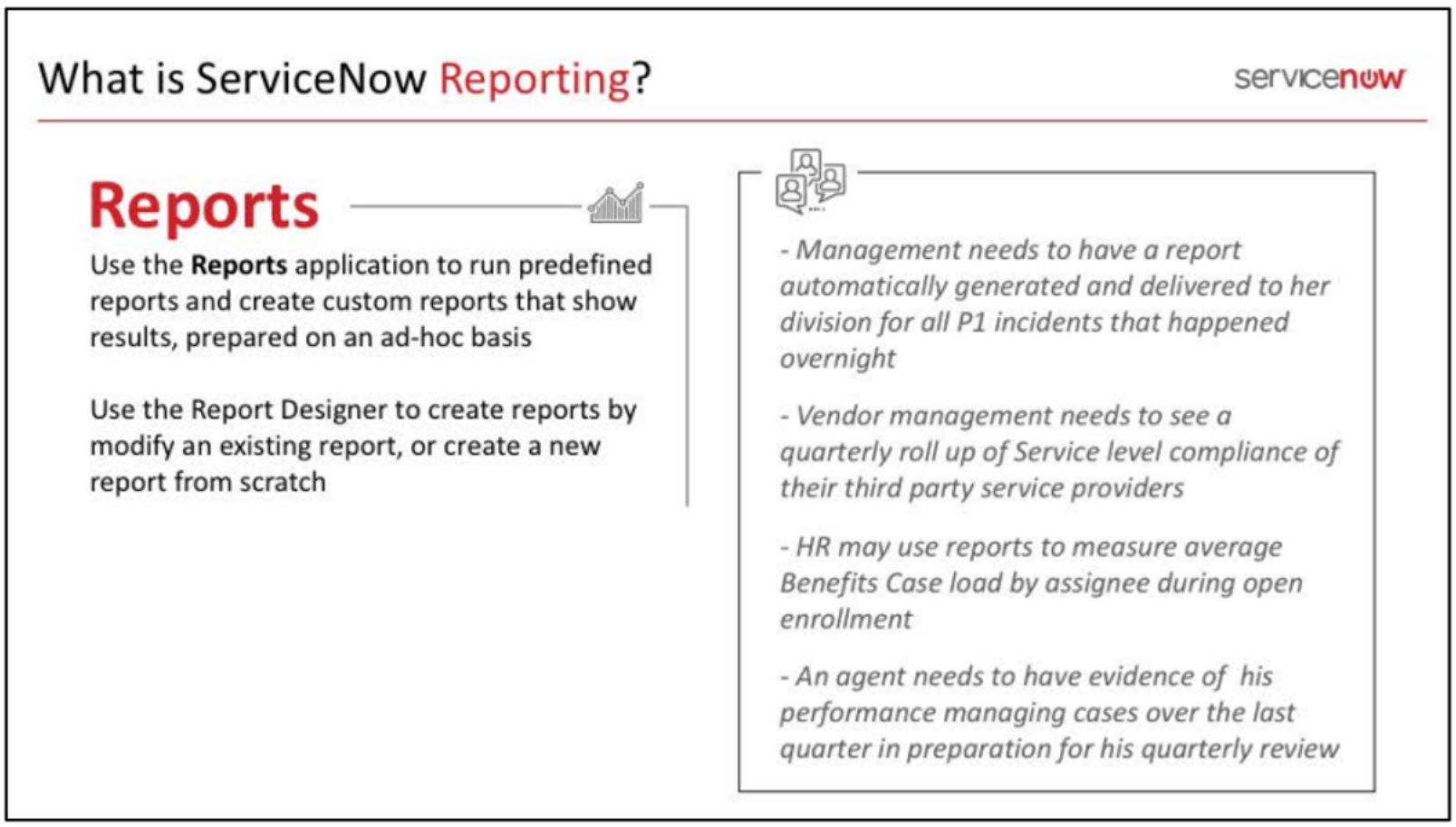
That does it for the Knowledge Management Lab!

Module 4 – 4.5 Reporting

servicenow

Objectives

- What is ServiceNow Reporting?
- Report Types
- Creating and Editing Reports
- Report Designer
  - Actions
  - Options
  - Distribution
- Metrics
- Performance Analytics
- Dashboards



The **Reports** application allows you to graphically view and analyze data from your ServiceNow environment. Reports can take many forms, including bar charts, pie charts, dials, lists, pivot tables, donuts, and more. Reports can be run manually or scheduled to be run automatically.

ServiceNow reports are interactive; you can drill down into the report gauges to view and manipulate the underlying data.

There are a range of predefined reports that pertain to applications and features like Incident Management and Service Catalog requests, including Key Performance Indicator (KPI) reports. If none of the predefined reports meet your needs, you can create your own reports by navigating to the **Reports > View/Run** module. Alternatively, you can simply click most column context menus in any list to generate a report directly from the data in that list.

ServiceNow Reporting can easily answer such questions as:

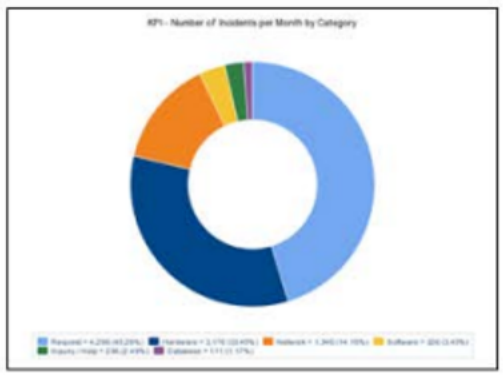
- Did I meet an SLA?
- How many incidents did my team close in a month?
- Metric Reporting: What was the average time from Incident open to Incident closed for each Service Desk team?

For more information, navigate to **Reports > Getting Started.**

Report Types

servicenow

The ServiceNow base instance comes with over 25 standard report types



Donut

Pareto

Heatmap

One report variation, a Pareto chart, named after Vilfredo Pareto, is a type of chart that contains both bars and a line graph, where individual values are represented in descending order by bars, and the cumulative total is represented by the line. Pareto Charts are useful to show the significance of factors for a given question/process. Pareto Charts use the rule that about 20% of input produces almost 80% of the outputs.

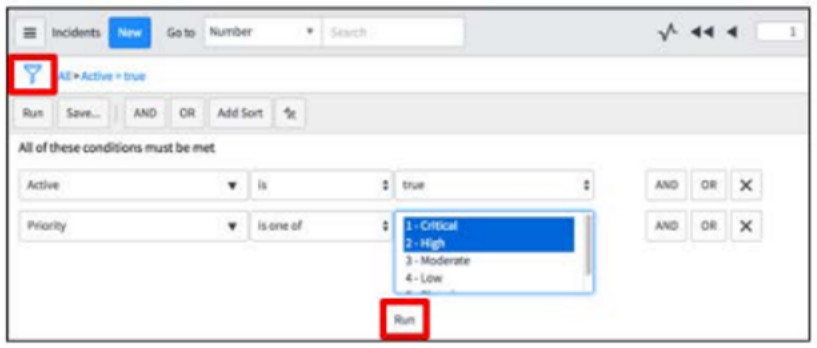
Other report types include: Speedometer, Dial, Single Score, Pie, Semi Donut, Bubble, Multi-Level Pivot Table, Line, Column, Area, Spline, Bar, Histogram, Horizontal Bar, List, Funnel, Calendar, Pyramid, Box, Trend, Control, Trendbox, Map, Pivot Table, and Text Analytics.

Create and Edit Reports

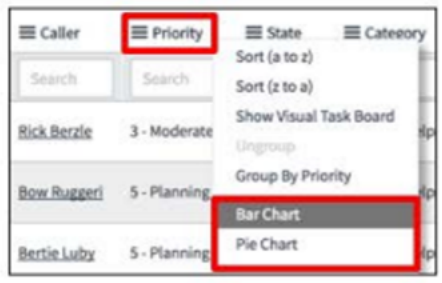
servicenow

Create a New Report from a List

1. Define and run a filter, displaying only the data to report on



2. Open the **Column Context Menu**, then choose **Bar Chart** or **Pie Chart**



Create a New Report or Edit an Existing Report

Navigate to **Reports > View / Run** to create a new report with the Report Designer, or open an existing report

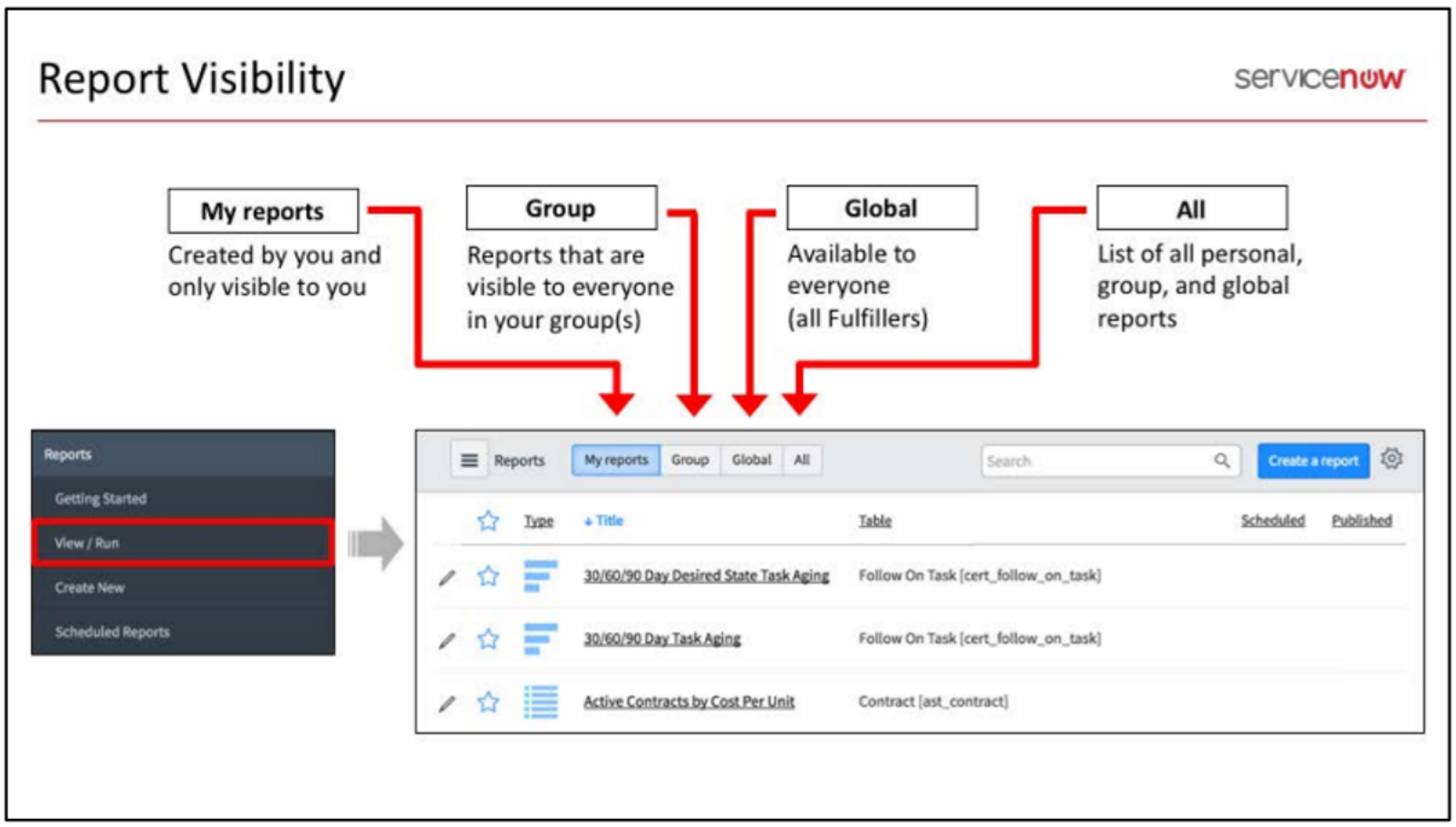
You can build reports from scratch, but it is often easier to start with a filtered list or an existing report. When you start with an existing report, reporting configuration choices will be provided for you. By modifying the provided options, you can configure the report to meet your needs. It is a best practice to copy a base report then edit your copy.

Use the ServiceNow Report Designer to:

- Leverage reporting visibility and available report types
- Use multi-level filters, filter operators, and sort order to refine reports
- View, create, edit, and schedule reports
- Work with reporting roles
- Use Related Tables (dot-walking and Database Views)

Advantages to modifying an existing report:

- You can start with a report that already has the basic information and make minor changes to get what you need
- Browsing existing reports helps you learn which of the tables are relevant to the work you do
- Helps you learn different uses for the various report types
- You can leverage ITIL best practices by using Key Performance Indicator (KPI) reports



The **Reports > View / Run** module contains a library of reports which you can run and use to create your own custom reports. Many of these reports came with the platform and others were created by your reporting administrators specifically for your company.

The Reports page contains different sections for reports which are visible to different audiences:

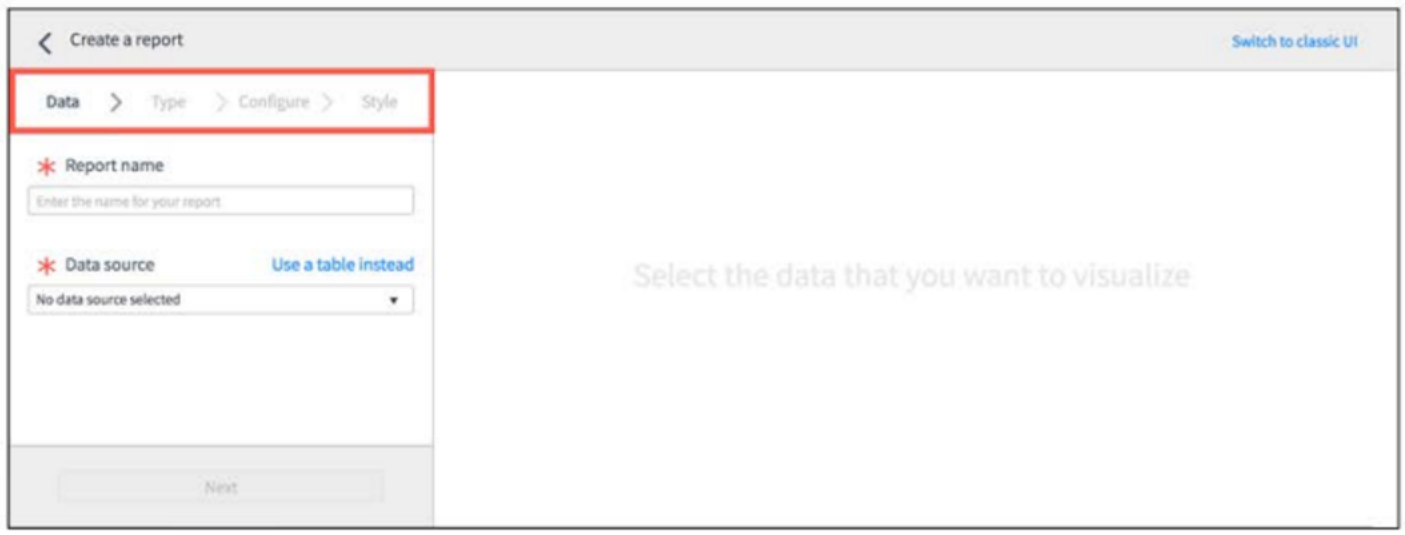
- **My reports**: Visible only to the report creator (Me)
- **Group**: Visible to one or more specific users and/or groups (Groups and Users)
- **Global**: Visible to all users (Everyone)
- **All**: A list of all Global, Group, and personal reports

Report Designer: Creating a Report

servicenow

Report Designer is an interface used for creating or modifying ServiceNow reports

It features four sections which provide reporting configuration options: Data, Type, Configure, and Style

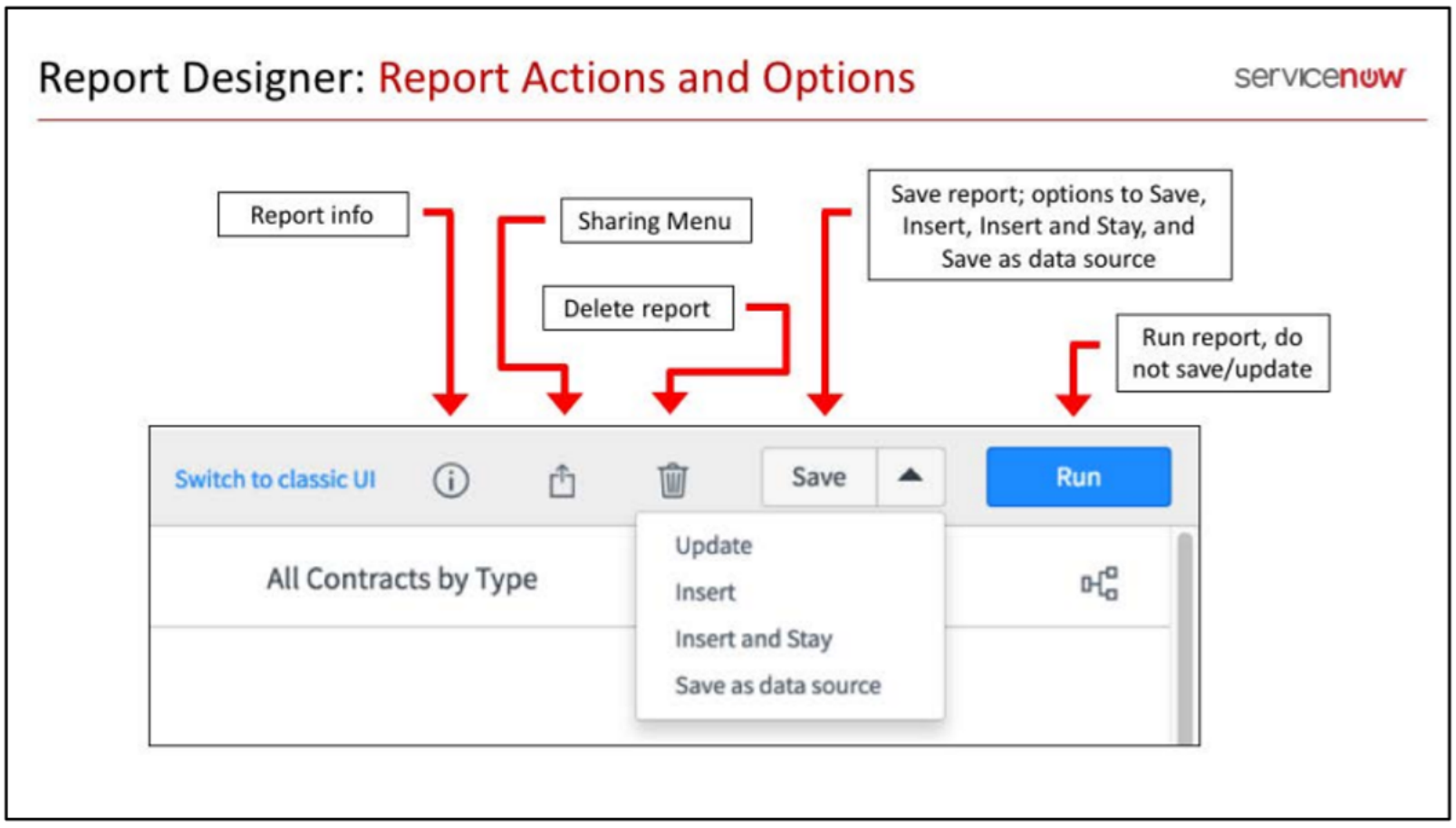


Users with the admin role can access the **Reports > Administration > Properties** module to enable the Report Designer for all users with the permissions to create/edit Reports. Alternatively, users will be able to revert back to the classic reporting interface by clicking on the **Switch to classic UI** link towards the top-right of the Report Designer interface.

Each section of the Report Designer provides different configuration options:

- **Data:** Provide a name for the report, as well as select the source from where your data comes from. You can choose a data source, which is a predefined data set used for creating reports; or a ServiceNow table.
- **Type:** Select the visualization of your report by choosing a report type. There are 27 different types to choose from!
- **Configure:** Do things like group the data by a specific field(s) and run calculations against the data.
- **Style:** Adjust the look of your report, from coloring to titles, as well as making adjustments to the report legend.

NOTE: Every time you make an adjustment through these controls, remember to click the **Run** button in the top-right to redraw the report with your changes.



The Report actions available to you depend on your role. Many of the Report actions are easily understood; therefore only some of them are detailed here:

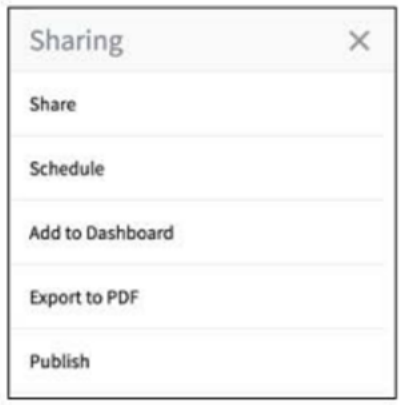
- **Update**: Overwrite report, return to the report list
- **Insert**: Save a duplicate copy of the report, return to the report list
- **Insert and Stay**: Save a duplicate copy of the report, remaining on the report
- **Save as report source**: Allows you to create a pre-defined data set that can be used for creating reports

Report Designer: Report Distribution

servicenow

Share

To make a report visible to a particular group or user, use the **Share** option to select Groups and/or Users



Publish

Steps to publish and view a report:

1. With desired report displayed click the **Sharing** menu icon, then click **Publish**
2. Click the **Copy report link** icon from the report header to copy the URL to your clipboard
3. **Open URL** in browser

From the **Sharing** menu, you have the following options:

- **Share:** Allows you to specify who can see the report. Options include Me, Everyone, and Groups and Users. Admin role is required for Everyone and Group sharing.
- **Schedule:** Create a scheduled email of the report
- **Add to Dashboard:** Add directly to a Dashboard on a homepage you choose, or within a Performance Analytics tab
- **Export to PDF:** Convert the report to a PDF which can be generated immediately or sent as an email
- **Publish:** Create a public URL for this report. Users may need to log into ServiceNow to view the report and have an appropriate role in order to view all of the data.

When distributing a report, sharing has the ability to make the report visible to authenticated users within ServiceNow.

Publishing a report makes it available to users outside of ServiceNow but does not necessarily share the underlying data. Access Control rules restrict visibility to the underlying data but not to all reports. For this reason, it is recommended to be cautious when publishing reports for external visibility.

Reports and Metrics

When you **report** on a table (for example, Incident or Problem), information about the current state of platform data displays

A **metric** is used to measure and evaluate the effectiveness of IT service management processes

- Metrics measures data over time to show past history
- Metrics can gather data as the data is updated

System tables are, by default, restricted from the reporting module. These tables include, but are not limited to the following: syslog, syslog_transaction, sys_attachment, and sys_email.

Compared to reports, **metrics** are used to measure and evaluate the effectiveness of IT service management processes. A metric could measure the effectiveness of the incident resolution process by calculating how long it takes to resolve an incident.

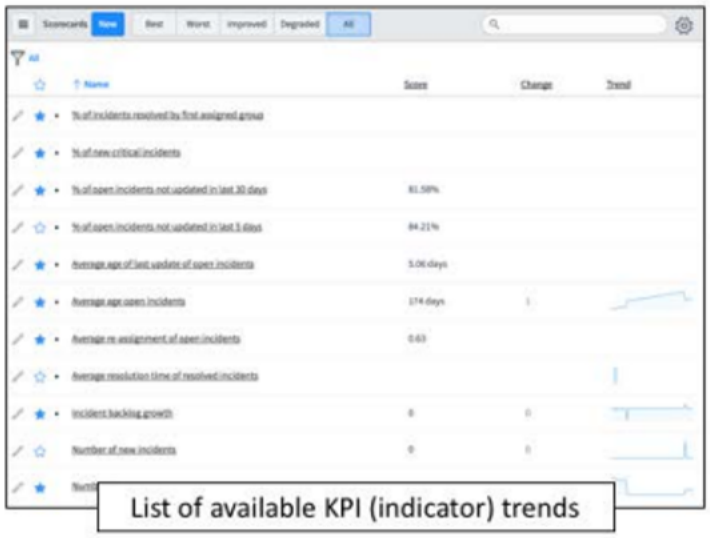
Sometimes a metric can be easily obtained from the data. For example, to find the number of incidents that were created today, a report will count the number of incident records in the Incident table with a created date of today. Other times, metrics need to be gathered as data is updated. For example, determining how long an incident was assigned to a certain group requires collecting information about assignment changes and calculating the duration of each assignment.

The Metric Definition plugin provides a declarative way of defining metrics, and, once defined, the data for the metric is gathered, and occurrences of the metric are calculated and stored.

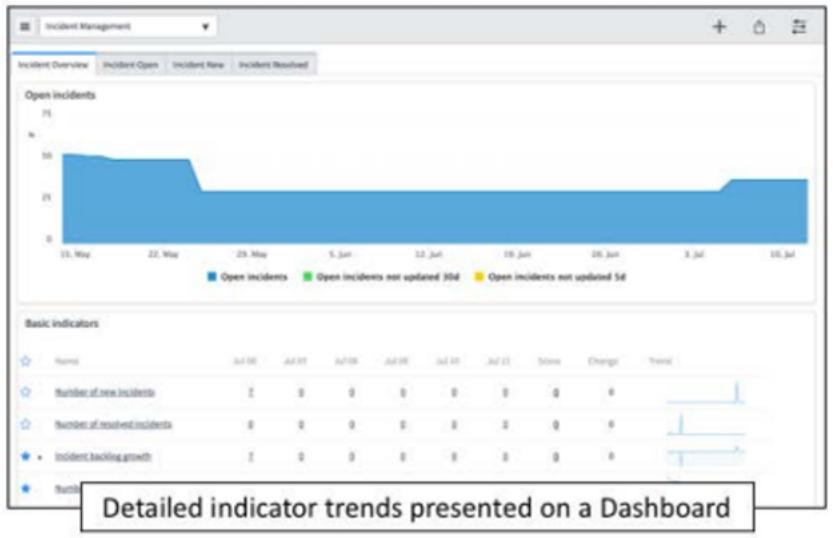
Performance Analytics

servicenow

Performance Analytics enables you to track and aggregate data over time through elements called **scorecards** and **indicators**



List of available KPI (indicator) trends



Detailed indicator trends presented on a Dashboard

Performance Analytics enables you to track and aggregate data over time, such as to measure how many tickets are resolved each week per assignment group. Performance Analytics is enabled for the Incident table by default. To track data for other tables and applications, you must license Performance Analytics.

Performance Analytics samples source data on a daily basis to build a trend over time.

Performance Analytics for Incident Management comes with several predefined elements that you can use to assess organizational performance, including:

- **Indicators:** define the metrics to track based on an indicator source, and specifies an aggregation such as to count the number of new tickets
- **Data collection jobs**: automatically collect scores for automated indicators and breakdowns
- **Scorecards**: display scores for a single indicator and allows you to perform detailed analysis of the metric, such as comparing scores over time
- **Dashboards**: contain actionable data visualizations that help you improve your business processes and practices

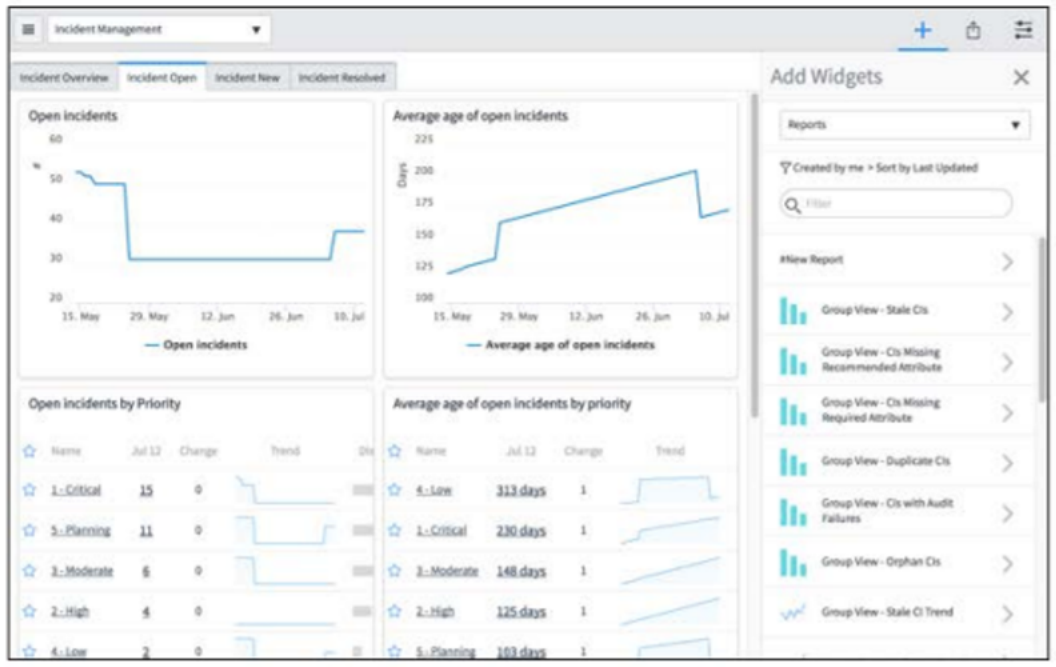
Performance Analytics for Incident Management is a limited version of Performance Analytics that is included in the base system, enabling you to become familiar with the functionality.

Performance Analytics premium allow you to create indicators and other configuration records such as breakdowns, and collect data for tables other than Incident.

Dashboards Overview

Dashboards enable you to display multiple Performance Analytics, reporting, and other widgets on a single screen

Use dashboards to create a story with data that can be shared

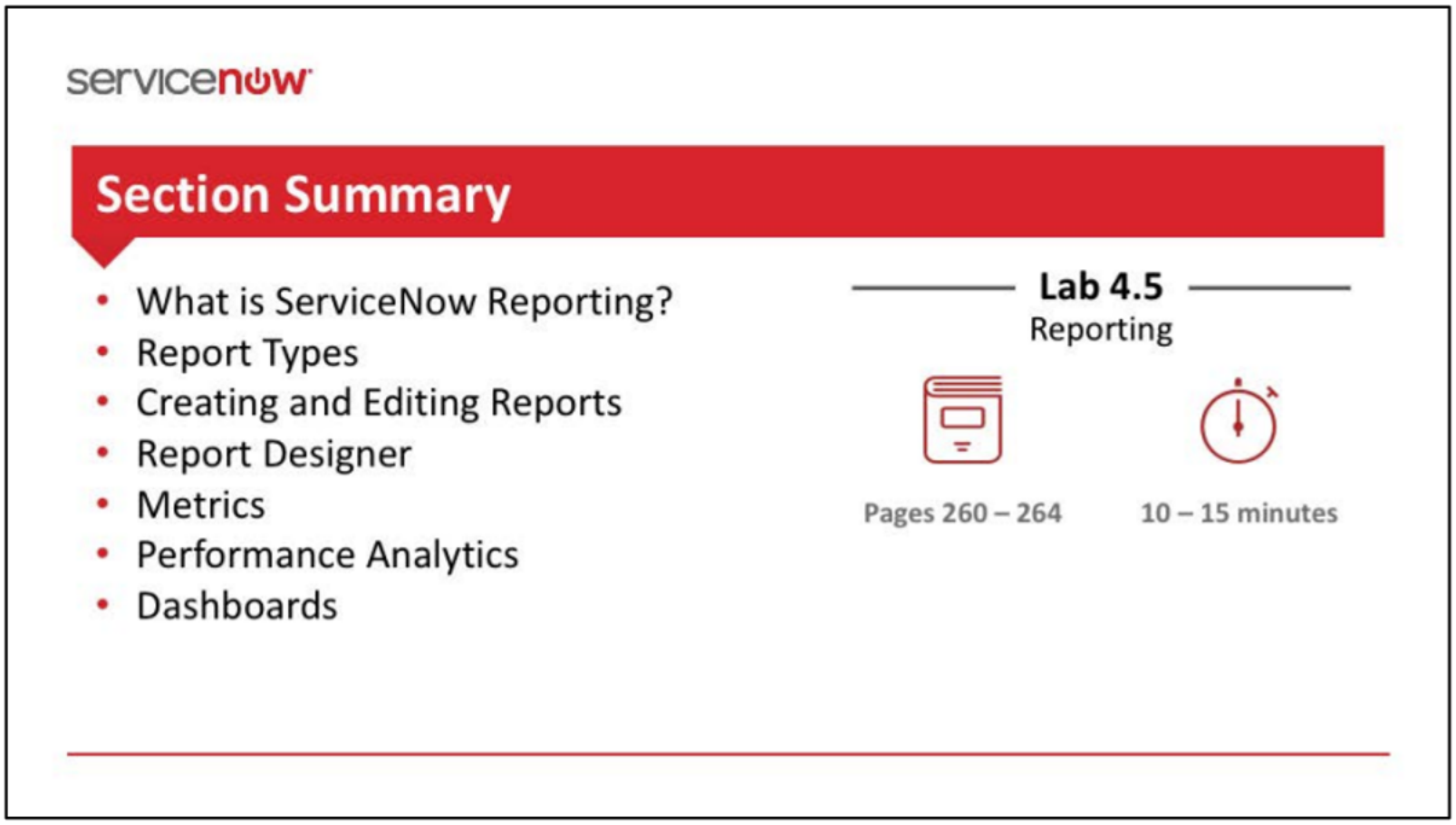


Dashboards may be responsive or non-responsive, but are set as non-responsive by default. Responsive dashboards require the activation of the Performance Analytics plugin on upgraded instances.

With dashboards you are able to:

- Create and edit Performance Analytics reports and other widgets directly from the dashboard
- Use the Add Widgets pane to quickly find and preview widgets, then add them to the dashboard
- Easily share dashboards with other users from the integrated sharing pane
- Use quick layouts to snap widgets into a predefined layout, then adjust the layout as desired
- Set dashboards as your homepage so you can quickly access information that you use frequently

NOTE: There is a lot you can do with dashboards so it is encouraged that you to find more information at the ServiceNow product documentation site.



Lab 4.5 – Reporting:

- Create a report for a group
- Share the report to a group

LAB

Reporting

4.5

10 – 15 minutes

Lab Goal

Lab Dependency: Requires the completion of Lab 1.3.

This lab will show you how to do the following:

- Create a report for a group
- Share the report to a group

Members of the Service Desk would like a report built which provides a high-level summary of all incidents assigned to their groip, organized by incident category and priority.

Winnie Reich will create a report and then have the system administrator share it to her team.

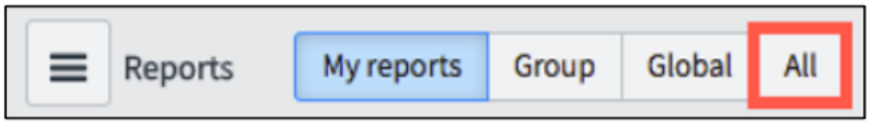
A. Create a Report

1. Impersonate **Winnie Reich**.
2. Open the list of Reports by navigating to **Reports > View / Run**.

NOTE: *Although the first part of the lab is completed as Winne Reich, any user with the right permissions can access the* ***Reports*** *application menu and modules to create and share repots.*

---

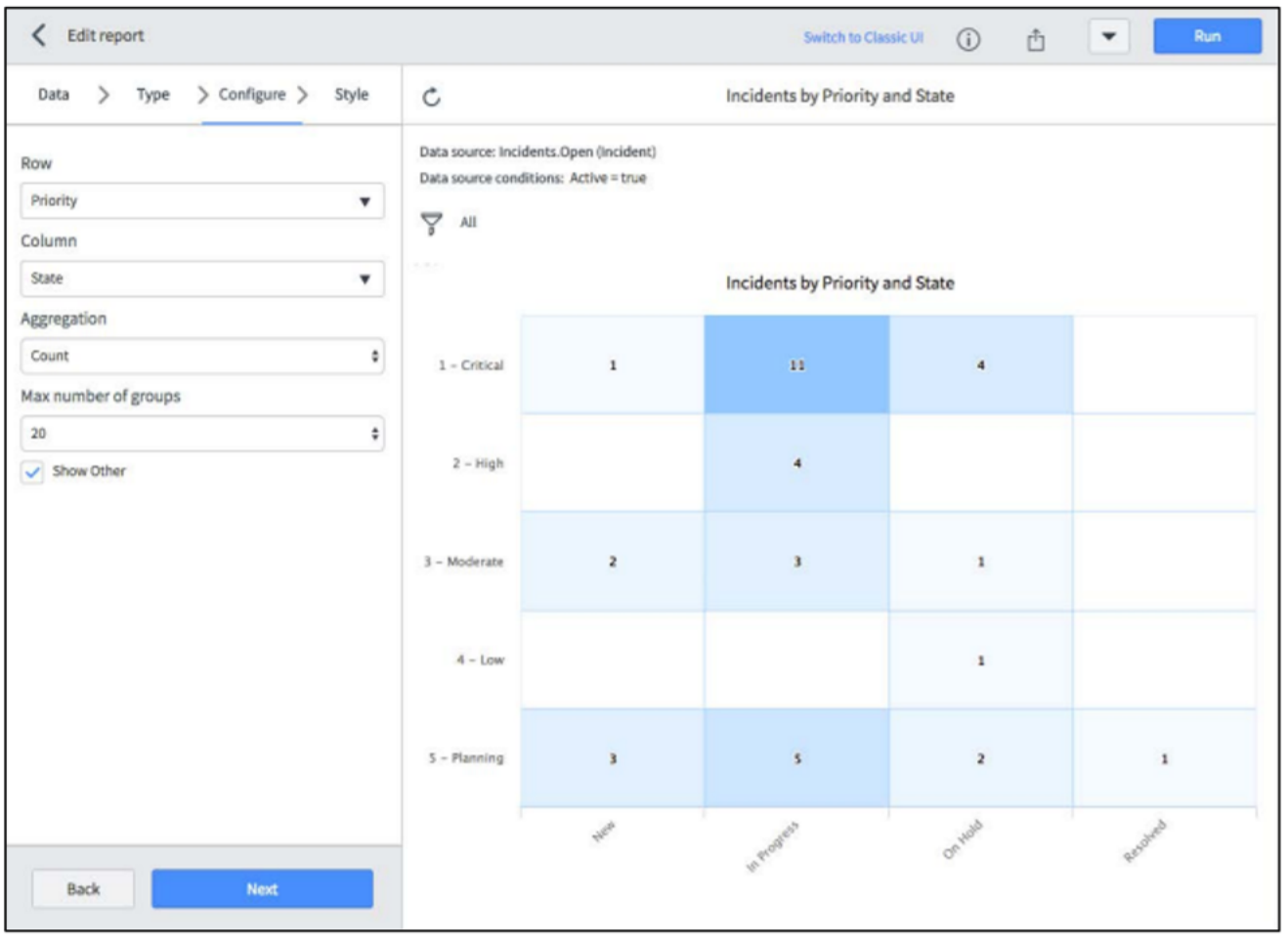
3. Click on **All** from the header options:



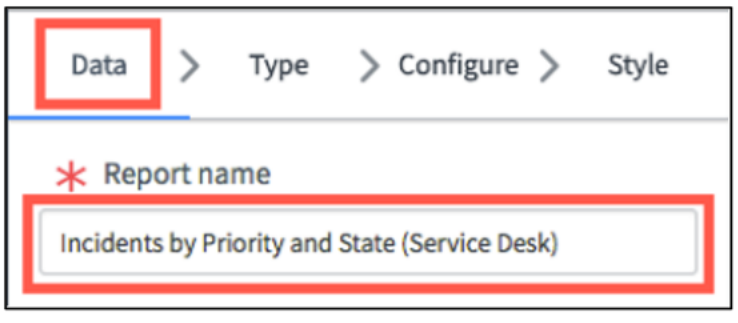
4. Use the search bar at the top-right to search for reports containing **incidents by priority** in their title.

5. Once the results display, click **Incidents by Priority and State** to open the report:

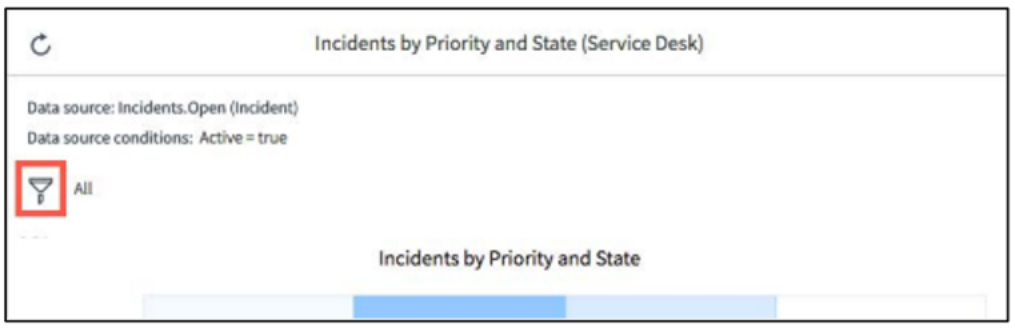
6. The **Report Designer** loads with the report displaying information:



7. Select the **Data** breadcrumb title from the panel on the left, then update the name of the report to **Incidents by Priority and State (Service Desk)**:



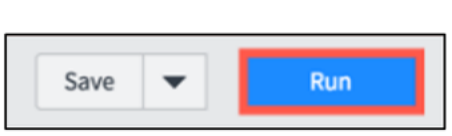
8. From the main reporting area, where the data is displayed, click the **Open condition builder** icon (funnel) to apply a filter on the data:



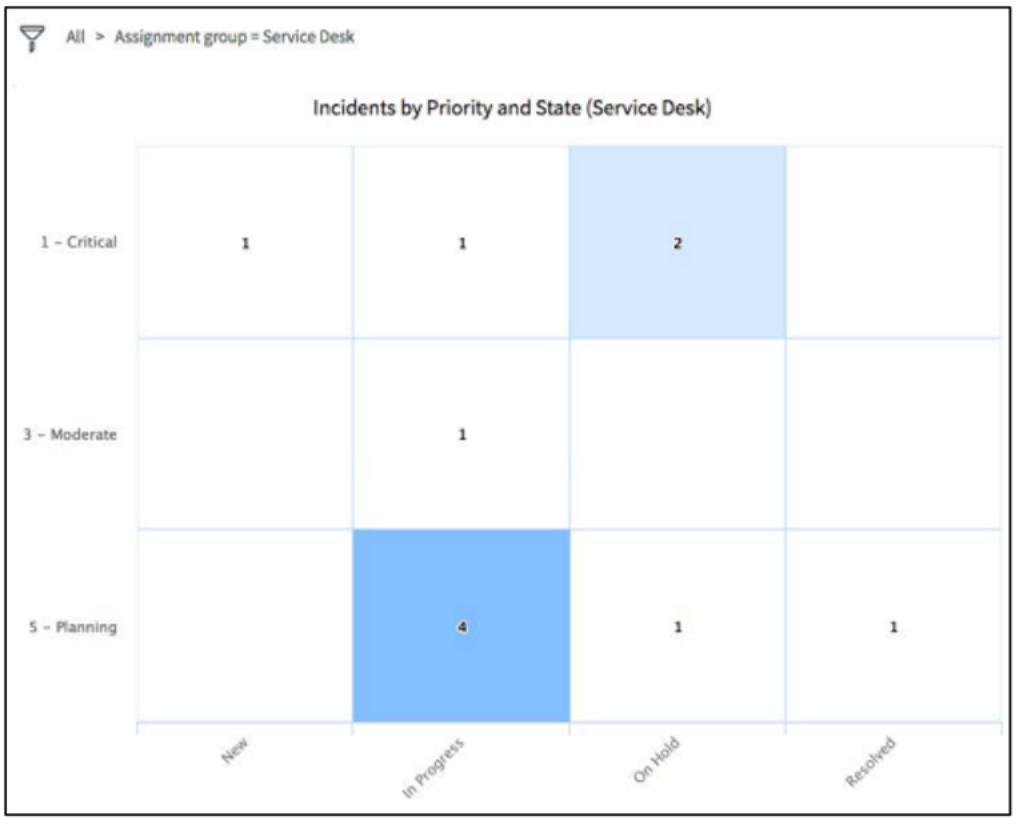
9. Use the dropdown menus to set a single filter condition:

   Assignment group | is | Service Desk

10. From the Report Designer header, click the **Run** button:



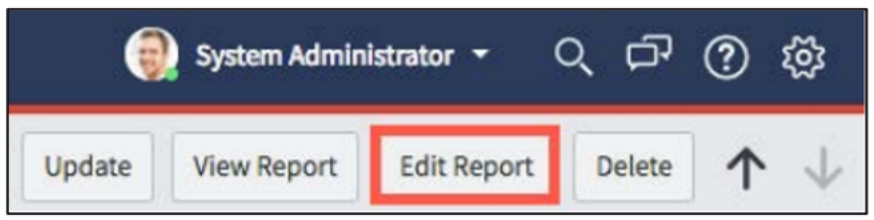
11. Notice the report's data has updated:



12. Open the **Save** menu by clicking the downward-facing arrow, then select **Insert** to save a copy of the new report you created:

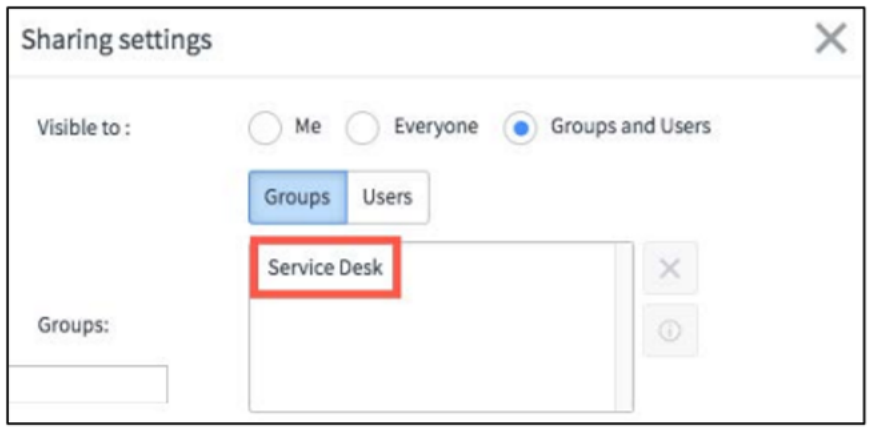
B. Share the Report

1. Impersonate **System Administrator**.
2. **Reports > Administration > All**.
3. Locate and open the record for the **Incidents by Priority and State (Service Desk)** report.
4. Click the **Edit Report** button:



5. From the Report Designer, open the **Sharing** menu:

6. Select **Share**.
7. Choose **Groups and Users**, then add the **Service Desk** to the Groups list:

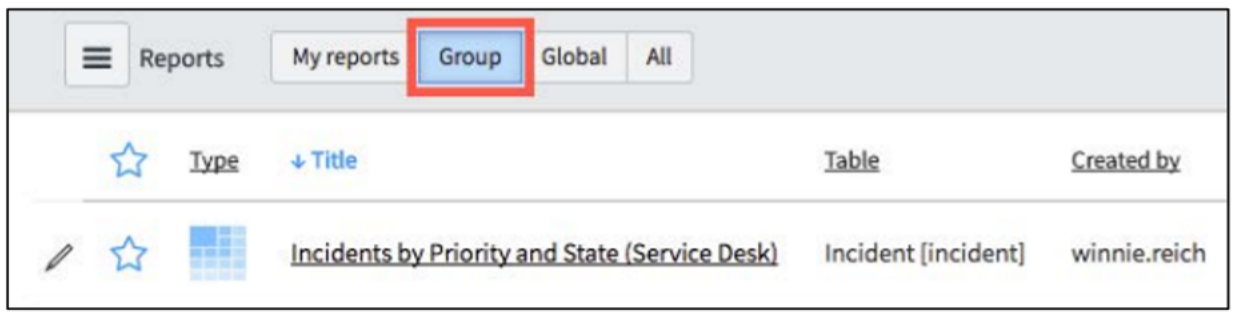


8. Click **OK** to close the Sharing settings window.
9. **Save** the report.

Lab Verification

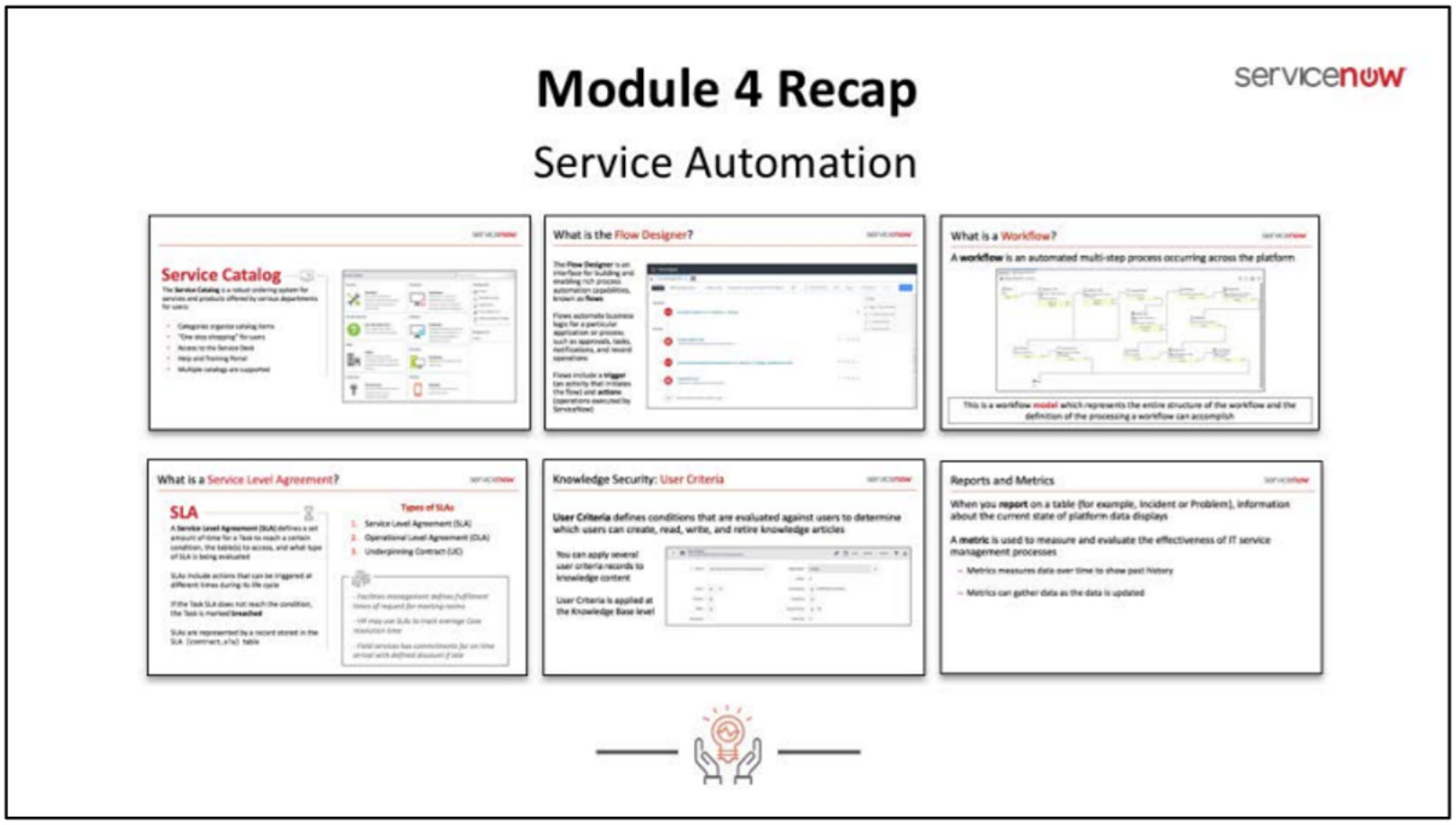
Access Report from Group Category

1. Impersonate **Kevin Edd**.
2. **Reports > View / Run**.
3. From the Reports list, select the **Group** category:



***NOTE:** The Infinity report is now available to all users in the **Service Desk** group.*

Wonderful! In this lab you have learned how to modify an existing report and share it.



For these selected topics, discuss:

Why would you use these capabilities?

When would you use these capabilities?

How often would you use these capabilities?

servicenow

1 User Interface & Navigation

2 Collaboration

3 Database Administration

4 Service Automation

5 **Intro to Scripting & Application Tools**

Module 5 – 5.1 Scripting

servicenow

Objectives

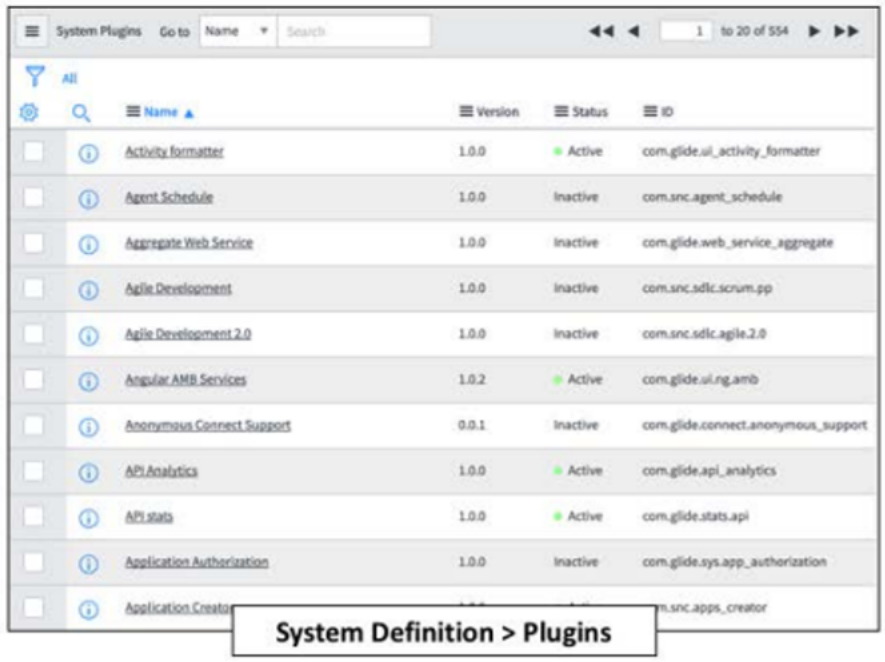
- What are Plugins?
- What is Scripting in ServiceNow?
  - Client/Server Side Scripting
- Script Types
  - UI/Data Policy
  - UI Action
  - Client Script
  - Business Rule
- Scripting Areas in ServiceNow

What are Plugins?

servicenow

Before adding script to ServiceNow, administrators should check the list of available plugins

Plugins provide additional optional functionality within a ServiceNow instance



System Definition > Plugins

System administrators have control over when to activate plugins.

Some plugins include demo data - sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when first installing the plugin on a development or test instance. Demo data can loaded after the plugin is activated by repeating this process and selecting the checkbox. If the plugin depends on other plugins, these plugins and their activation status are listed.

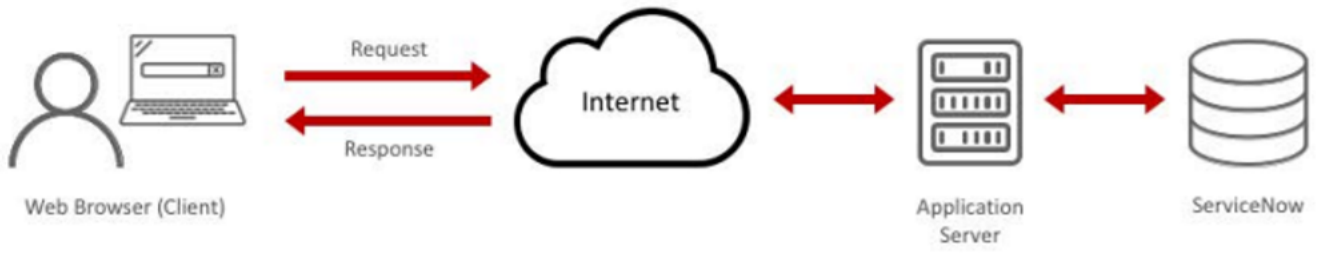
Most plugins are published, and system administrators can activate any published plugin. But, some plugins are available only by request due to operational considerations making the plugin only appropriate for certain deployments. In these cases, to activate the plugin, make a Service Catalog request to ServiceNow Technical Support using the Request Plugin Activation form at https://hi.service-now.com.

What is Scripting in ServiceNow?

servicenow

Scripting in ServiceNow or Platform Scripting is the customization of an instance and/or applications by using JavaScript

Dependent on how the script is executed, JavaScript may execute on the **client** or **server** side and fundamentally alter how the baseline instance functions and user experience



Client refers to an application or system that accesses a remote service or another computer system, known as a server. A **server** is the computer program running as a service; a physical computer dedicated to running one or more services, or a system running a database.

ServiceNow uses a Software as a Service (SaaS) model; the web browser is the client. The web browser is the only thing that is installed on the client. The application server and the database live at the Data Center. Client scripts run on the client browser. Server scripts run on the server (which includes the database).

Client to server round-trips take time and make the end-user wait for the round-trip to complete. Request + Response = Round trip.

Script Type: UI Policy and Data Policy

servicenow

A **User Interface (UI) Policy** is a rule that is applied to a form to dynamically change form information or the form itself

Once a UI Policy is saved, **UI Policy Actions** determine what happens on the form, including:

- Setting a field as mandatory – requiring a value in order to save the record
- Setting a field as hidden – no longer displaying a field on the form
- Setting a field as read-only – preventing a user from updating its value

UI Policies execute on the **client** side

A **Data Policy** is a rule that enforces data consistency by setting fields as mandatory and/or read-only

Data Policy controls are similar to UI Policies but UI Policies are only enforced on data entered into a form (passing through the UI)

Data Policies are applied to all data entered into the platform; form (UI), Import Sets, or Web Services

A Data Policy execute on the **server** side but can also run as a UI Policy on the client side

To immediately implement updates and changes to forms and lists, you can use UI Policies which allow you to add sophisticated controls without having to write scripts and define custom process flows for tasks.

Use a UI Policy to set fields on a form to:

- Mandatory or Optional
- Hidden or Visible
- Read-only or Editable

NOTE: To apply a UI Policy to all views, set the **Global** setting to **true**.

A Data Policy enforces requirements on field and record data when the data is imported into ServiceNow or when the data in an Import Set is submitted through an external system. Data Policies can be opted out for Web Services and Import Sets. A Data Policy is used to set mandatory and read-only states on form fields. Data Policies can be used on lists to make a field read-only; the field will appear to be editable, but the update will fail.

The purpose of a Data Policy is to standardize the same data across ServiceNow applications.

NOTE: UI and Data Policies are not about security, they are about managing the user experience.

Script Type: UI Action

servicenow

User Interface (UI) Actions add buttons, links, and context menu items on forms and lists, making the UI more interactive, customizable, and specific to user activities

For example: enabling a **Create Problem** button on an existing incident form

UI Actions can contain scripts that define custom functionality. UI Actions can be server or client side depending on the 'client' check box selection. This setting determines when a UI Action can appear.

UI Actions include:

- Form buttons
- Form context menu items (right-click the header)
- Form links (Related Links in a form)
- List buttons
- List context menu items (right-click a record)
- List choices (at the bottom of a list)
- List links (Related Links at the bottom of a list)

When Order 100 is specified, UI Actions with Order numbers greater than 100 will display after this UI Action, while UI Actions with Order numbers less than 100 will display before this UI Action, in the user interface.

NOTE: When the UI Actions **Active** box is checked, the UI Action is running and visible unless there is a condition met that specifies otherwise.

Script Type: Client Script

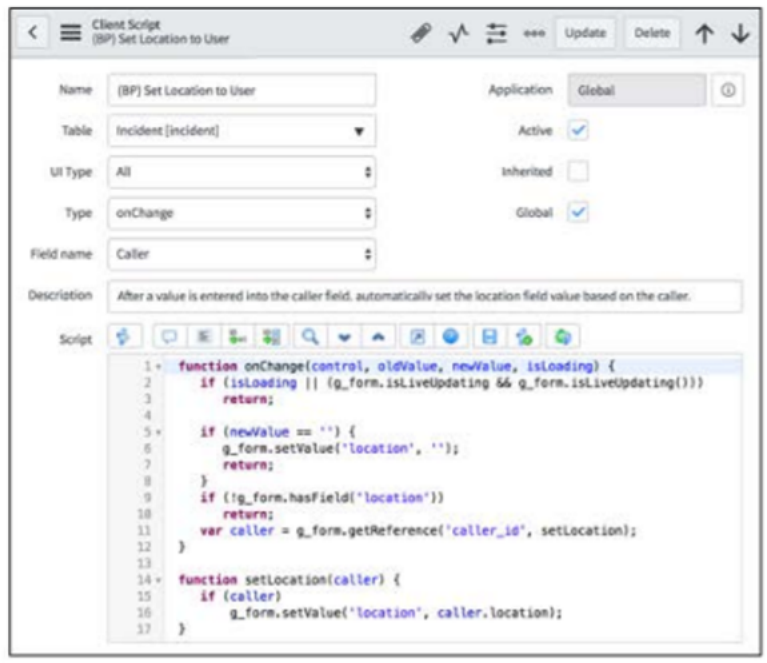
servicenow

Client Scripts make "real-time" changes to the appearance of the user interface, especially forms

Client Scripts can be created to do the following:

- Automatically update the location field to reflect the value (user) entered into the caller field
- Disable the attachment link of a closed record when the form is loaded so a user is unable to add or modify attachments
- Display a notice at the top of the page to confirm a catalog request was submitted

Client Scripts execute on the **client** side



Client Scripts allow for browser/form manipulation and verification such as making fields visible on a condition. An example of this would be an alert appearing when a user changes the priority of an incident. Client Scripts get executed on the browser, but you may also run a Client Script when a database lookup is needed; if you think you need database info, and you need the info frequently (such as every form load) then ask: Is it a field you can add to the form but hide? Is it something you really, truly need?

Several types of scripts are supported:

- **onCellEdit()**: runs when a cell on a list changes value through use of the list editor
- **onChange()**: runs when a particular field changes value
- **onLoad()**: runs when a form is loaded
- **onSubmit()**: runs when a form is submitted

Unlike onLoad() or onSubmit() scripts, onChange() scripts apply to a particular widget on a form, rather than to the form itself. They are fired when a particular value on screen changes. An onLoad() script runs when a form is first drawn and before control is given to the user to begin typing. Typically you use an onLoad() script to perform some client side manipulation of the document on screen. An onSubmit() script runs when a form is submitted. Typically you use an onSubmit() script to validate things on the form to make sure the submission makes sense. As such, onSubmit() scripts can potentially cancel a submission by returning false.

Script Type: Business Rule

servicenow

A **Business Rule** is configured to run when a record is displayed, inserted, updated, deleted, or when a table is queried

Business Rules execute on the **server** side

Business Rules can be set to run **before** or **after** the database action has occurred

The **When** setting determines when the Business Rule executes and has the following choices:

- **Before** a record is saved to the database
- **After** a record is saved to the database
- **Async** (queued); client and server work independently so the client is not waiting for the server
- **Display** before the record is displayed

Although there are multiple ways to control behaviors in the ServiceNow application, most customization of platform behavior is done using Business Rules. Business Rules are loaded and initialized at the beginning of each interaction between a user and the platform.

Every Business Rule includes what table to run against and timing (before or after insert and more), what conditions to evaluate, what script to run based on the evaluation, and if it is client-callable.

Business Rules are consistently applied to records regardless of how they are accessed-through forms, lists, or Web Services. This is one major difference between Business Rules and Client Scripts, which only apply when editing through the form.

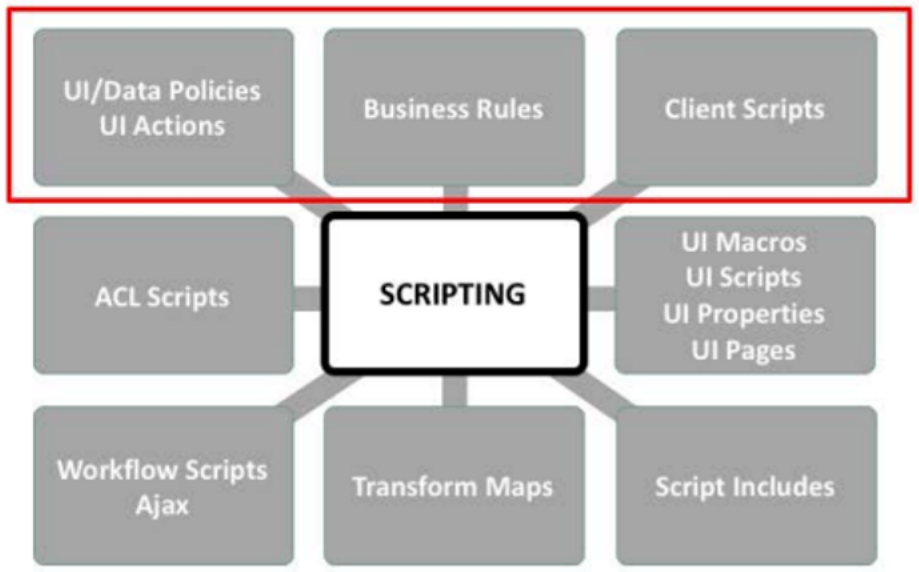
Unlike UI Policies, Business Rules are **NOT** real-time:

- They do not monitor fields on a form
- They monitor records as they are inserted or updated

Business Rules run on the server, but can be client-callable. If the **Client callable** setting is checked, the client can use AJAX to call the Business Rule.

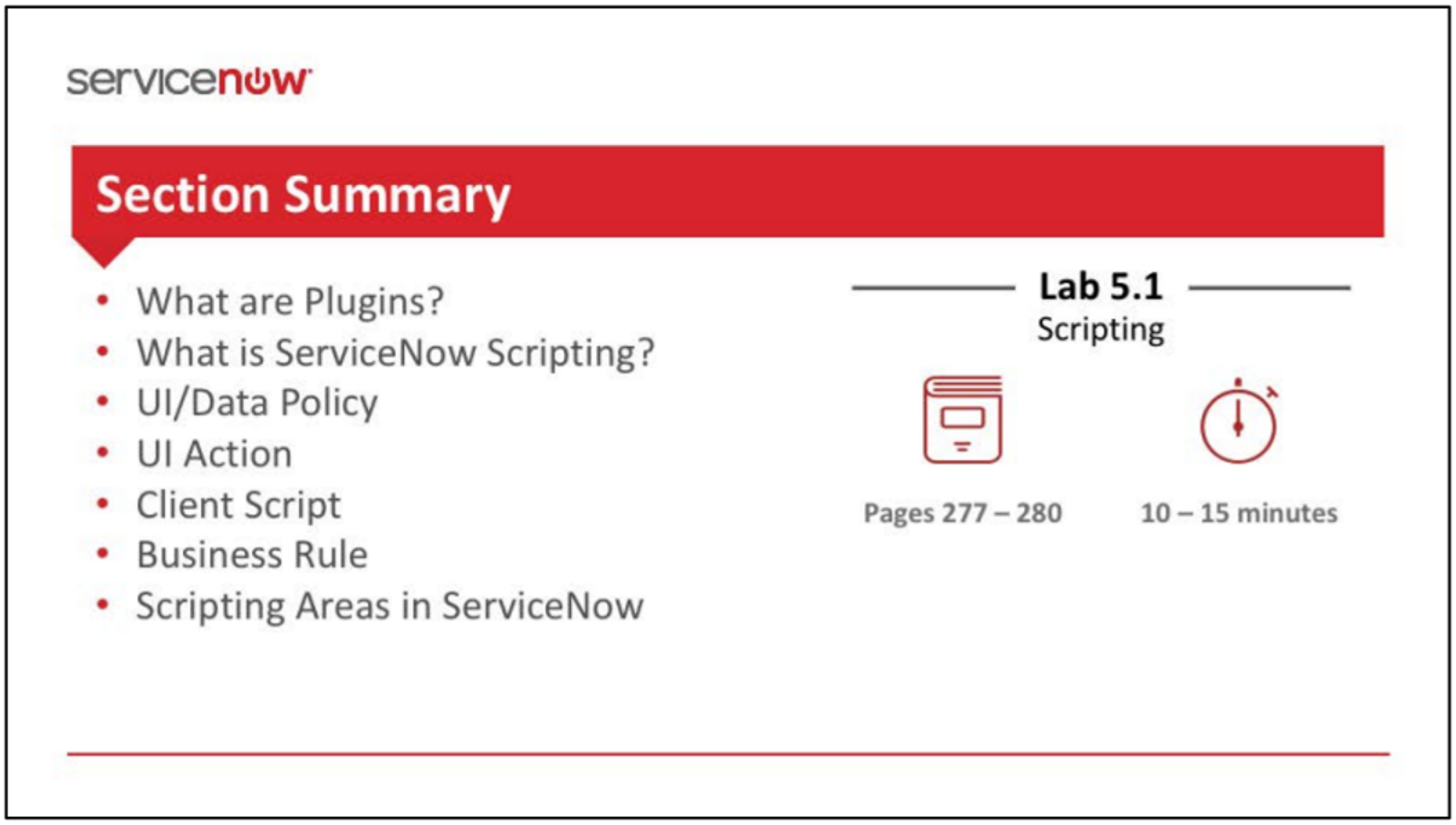
The primary objective of **display** Business Rules is to use a shared scratchpad object, "g_scratchpad", which is also sent to the client as part of the form. This is useful when you need to build client scripts that require server data that is not part of the record being displayed.

Scripting Areas in ServiceNow



ServiceNow has over 30 places where code can be inserted to change the behavior of the platform. JavaScript is used almost everywhere and it is a very flexible and powerful language commonly known for its inclusion in most modern web browsers. This has made it almost mandatory for web development these days, with its simple syntax allowing many people to quickly add simple logic to web pages with minimum effort. Taking advantage of this familiarity, ServiceNow uses JavaScript both on the server and on the client.

You can learn more about the Scripting in ServiceNow course at https://www.servicenow.com/services/training-and-certification/scripting-in-servicenow-training.html and additional information on scripting at our product documentation site (docs.servicenow.com).



Lab 5.1 – Scripting:

- Create a UI Policy with a UI Policy Action
- Create a Business Rule

Scripting

LAB

5.1

10 – 15 minutes

Lab Goal

Lab Dependency: Requires the completion of Lab 1.3.

This lab will show you how to do the following:

- Create a UI Policy with a UI Policy Action
- Create a Business Rule

Cloud Dimensions would like to continue configuring ServiceNow as much as possible by using baseline functionality and available features.

A. Create a UI Policy

One requirement for Cloud Dimensions' process improvement is to require important Infinity support data, with the goal to speed up resolution time. To achieve this, create a UI Policy and UI Policy Action to make the **Assigned to** field required for employee incidents.

1. Ensure you are logged into the instance as **System Administrator**.

2. **Incident > Create New**.

3. From the **Form Context Menu**, select **Configure > UI Policies**.

4. Click **New**.

5. Fill out the UI Policy form information as shown:

   Table: **Incident [incident]** (already selected)
   Short description: **Mandatory Assigned to if Employee = True**

   Under the **When to Apply** tab,
   Conditions:
   Employee | is | True

6. **Save**.

You have just defined a UI Policy and set the conditions on which it will be enforced. Next, indicate which field(s) to affect by the UI Policy.

Create a UI Policy Action

1. Scroll down to the **UI Policy Actions** section, then click **New.**
2. Enter the following information on the **UI Policy Action** form:

   Field name: **Assigned to**
   Mandatory: **True**

3. Click **Submit** to save the UI Policy Action.

NOTE: *A message may display indicating multiple UI Policies with the same order for this field exist. Basically, this means there are other policies that may run at the same time as this one, and that the end-result may not be predictable. You can view the other policies by selecting their names in the message – adjusting either policies accordingly.*

Confirm New UI Policy is Working

1. **Incident > Create New**.
2. Make sure the **Employee** field is on the form.

NOTE*: If you do not see the* ***Employee*** *field, switch to the* ***Infinity*** *form view.*

---

3. Select the **Employee** field.
4. Notice that the **Assigned to** field is now mandatory.
5. Uncheck the **Employee** field and notice that the Assigned to field is no longer mandatory.

B. Create a Business Rule

In this section of the lab, create a Business Rule to display an alert, “Your incident has been successfully submitted” to all users who submit an incident, improving overall user experience.

1. On an incident form, open the **Form Context Menu**.
2. Select **Configure > Business Rules**.
3. Click **New**.

4. Fill out the Business Rule form as shown:

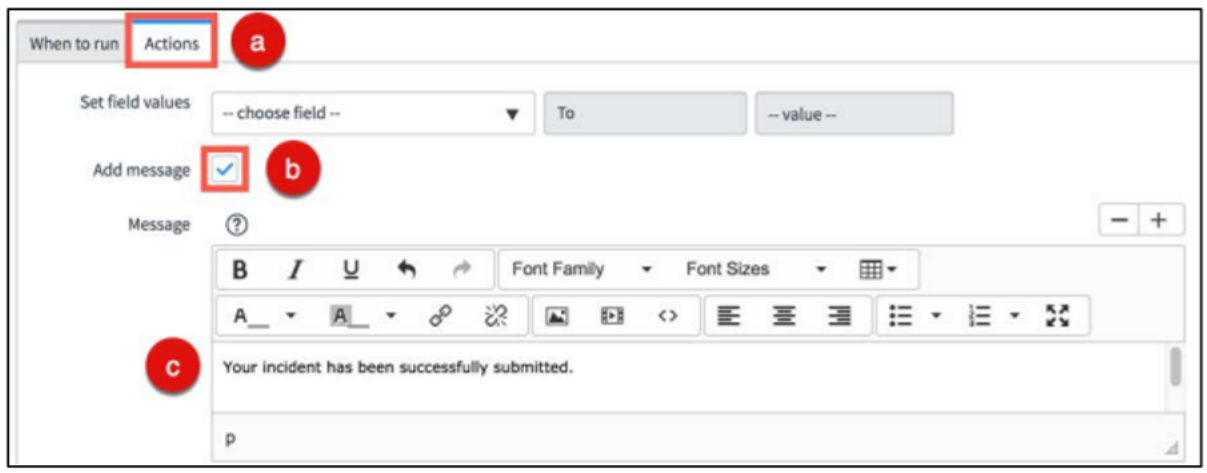
   Name: **Alert - Incident Submitted**
   Insert: **[check]**

5. Add a message:

   a) Click the **Actions** tab

   b) Check **Add message**

   c) Add message text: **Your incident has been successfully submitted.**



6. Click **Submit**.

Test Your Business Rule

1. **Incident > Create New**.
2. Fill out the incident form, including values for all mandatory fields.
3. Click **Submit**.
4. The new Business Rule displays your message in blue on top of the list:

Lab Verification

Incident UI Policy – Mandatory Assigned to if Employee = True

Business Rule – Confirmation Message

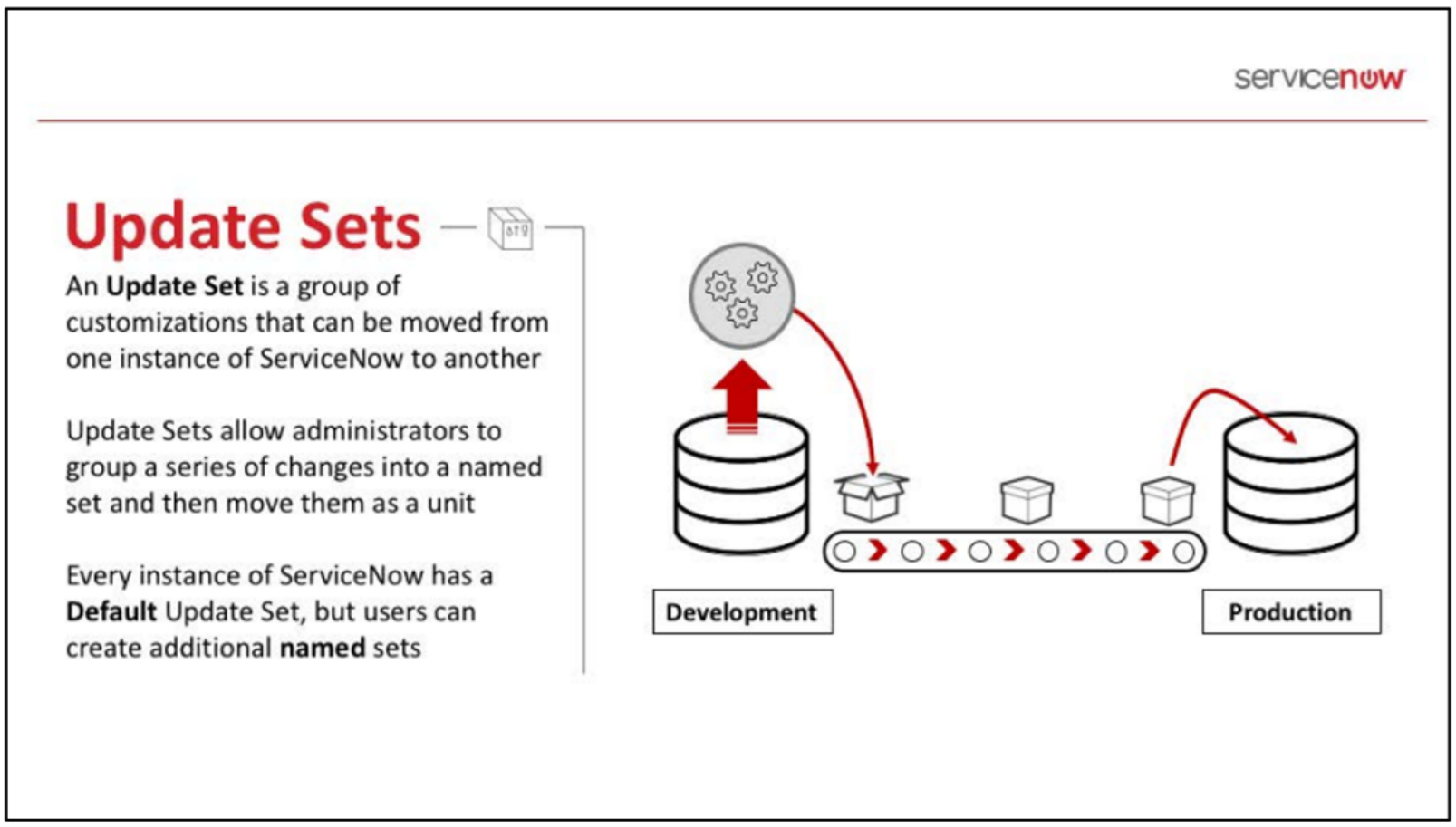
Good job, you have learned how to create a UI Policy and Business in this lab!

Module 5 – 5.2 System Update Sets

servicenow

Objectives

- What are Update Sets?
  - What is Captured in an Update Set?
  - Compare, Revert, and Merge
  - Best Practices
- Introduction to Integrations



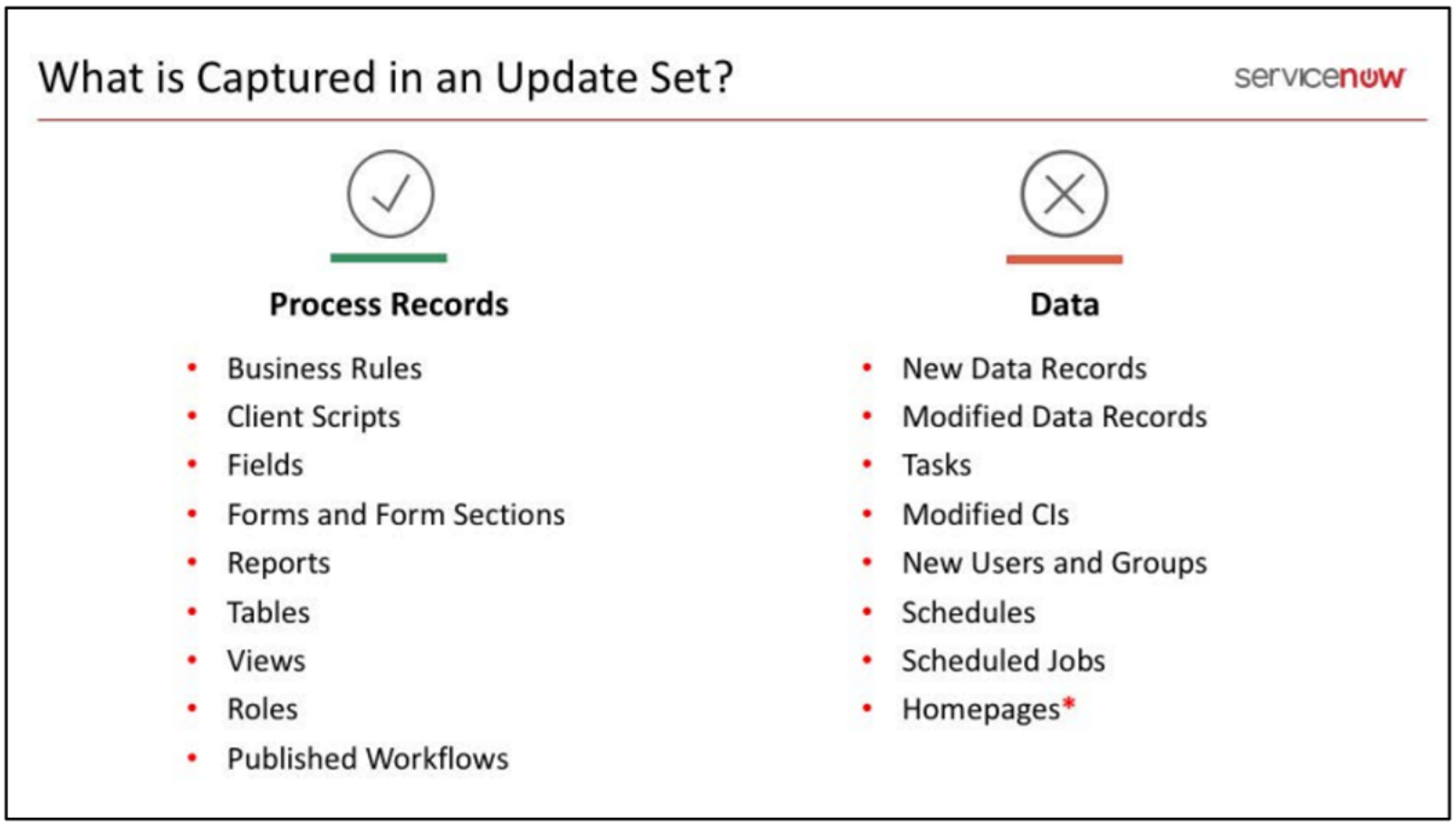
An Update Set example:

- A set of enhancements to Incident Management can be grouped in an Update Set called Incident Management 2.0.
- While Incident Management 2.0 is marked as the current Update Set, all process changes are tracked in it.
- Once the Update Set is marked as complete, it is ready to be moved to a test or production instance.

Basically an Update Set record is a “point in time” XML snapshot of process records. An Update Set works by writing changes from tracked tables to the **Customer Update** [**sys_update_xml**] table.

An Update Set is used to apply changes that have been checked and verified in another instance. When merging multiple Update Sets, if several Update Sets have modified the same object, (for example: the Incident form), the most recent change will be the one moved to the new, merged Update Set.

An Update Set is a container for configuration records. By navigating to **System Update Sets > Local Update Sets**, you can create a new Update Set or set an existing one as your current Update Set. Use an Update Set to migrate your code. When an Update Set is completed, you can transfer it to another instance to move customizations from development, through testing, and into production.



What is captured in an Update Set is typically a customization, or a change made to tables.

Homepages are **not** captured in an Update Set but can be manually added by navigating to **Homepage Admin > Pages**, right-clicking on a homepage record, then selecting **Unload Portal Page**.

NOTE: Data is not captured in an Update Set. Examples: a new incident or new change record would not be in an Update Set.

When completing work, you may want to move data records with your updates. These records can be useful for testing or training. Data (such as user records, CIs, or locations) can be moved using the **Export XML function**.

Compare, Revert, and Merge Update Sets

servicenow

Comparing Objects

Changes are tracked and you can customize objects on a table with the **update_synch** attribute, including:

- Tables
- Forms
- Fields
- Business Rules
- Client Scripts
- Views

Revert Compared Versions

You can compare versions before reverting a change

You can only revert back to the most recent base version

Merge Update Sets

Update Sets can be merged

If two users are working on separate Update Sets they can be combined into one Update Set for easy transfer

You can compare a version to the current version for any customizable object that a user has modified, such as a form layout or Business rule. The **Update Versions** [**sys_update_version**] table supports this feature.

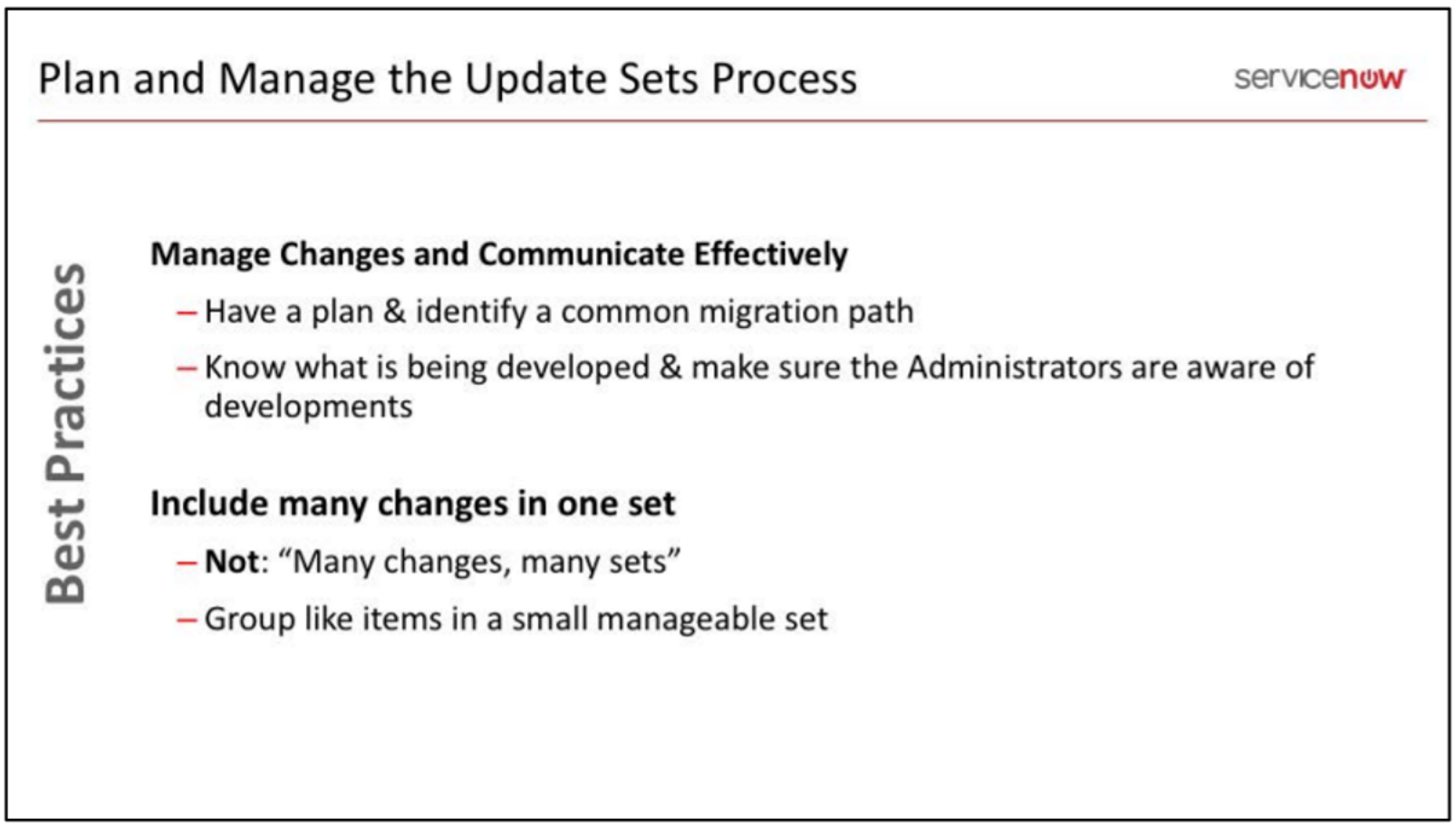
NOTE: Administrators can suppress versions for specific tables.

To revert changes:

1. View a list of versions for an object.
2. Right-click a version and select **Revert** to this version.
3. Click **OK** to confirm the action.
4. The selected version becomes the current version.

If both Update Sets have an update for the same object (for example, both Update Sets modify the Incident form), the most recent change will be the one moved to the new merged Update Set. The other update will be left in its original Update Set. Once a merge is performed, the other Update Sets remain, and if there were collisions, the duplicates remain where they were.

This provides a reference for what got moved and what did not. After merging and validating, it is a good idea to delete or empty the original Update Sets. The platform will not remove an update from an Update Set unless it was the one chosen for the merge.



The typical process of retrieving an Update Set includes verifying the Update Set is in a Complete state, Retrieve, Preview, and Commit.

Check that both instances are the same version since customizations may not work if they rely on code that has changed between versions.

Determine the changes to make in a single Update Set since ServiceNow recommends limiting Update Sets to a maximum of 100 records to reduce the number of potential conflicts and make it easier to identify and review changes.

Ensure that all platform records have matching **sys_id** fields since some platform records are created on an instance after provisioning and do not match between different instances, leading to problems with Update Sets. The best way to avoid this issue is to provision production and sub-production instances. You can clone the production instance onto the sub-production instance.

NOTE: Newest change will always overwrite older changes.

Introduction to Integrations

servicenow

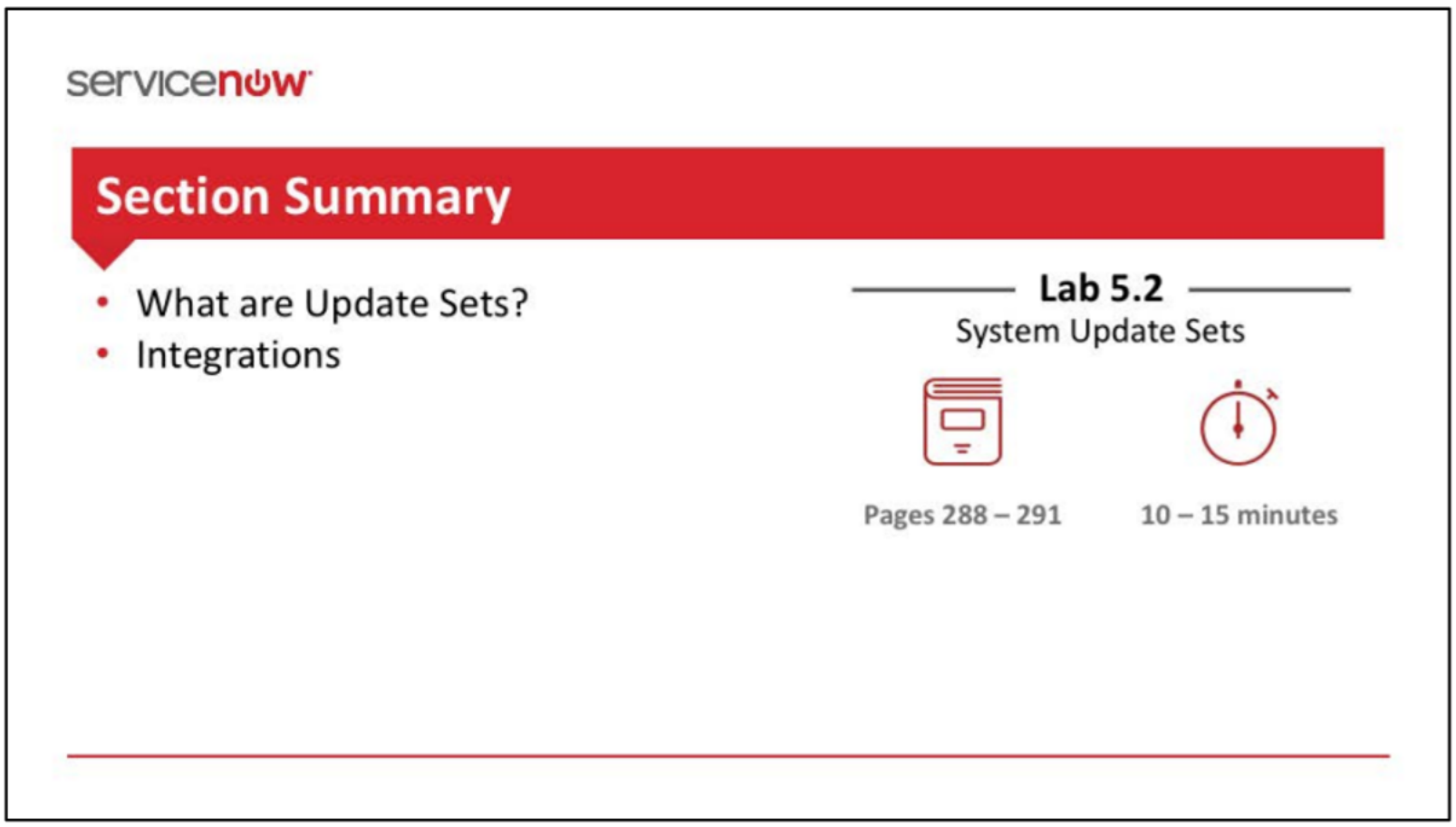
To share data between ServiceNow and an external system, ServiceNow integrates with many third-party applications and data sources

Standard integrations for ServiceNow include:

- Login (Single Sign-On)
- LDAP
- Communications
- Monitoring
- Discovery & Systems Management

The most common processes required for integration are the CMDB, Incident Management, Problem Management, Change Management, User Administration, and Single Sign-On.

A variety of techniques can be used, most notably Web Services, LDAP, Excel, CSV and email, as well as any industry standard technologies that use SOAP or REST WSDLs.



Lab 5.2 – System Update Sets:

- Review an Update Set
- Create another Update Set
- Make platform changes and capture them in the new Update Set

LAB

System Update Sets

5.2

10 – 15 minutes

Lab Goal

Lab Dependency: Requires the completion of Lab 1.2.

This lab will show you how to do the following:

- Review an Update Set
- Create another Update Set
- Make platform changes and capture them in the new Update Set

Update Sets are a useful tool for transferring configurations between ServiceNow instances.

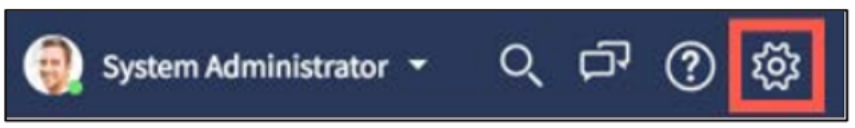
As Cloud Dimensions system administrator, review an Update Set to identify configuration changes that have been captured throughout class.

Finally, create another Update Set and capture additional changes that will be transfered in the next lab.

A. Review an Update Set

You may remember that we created an Update Set in Lab 1.2 which has been collecting updates we have made to the lab instance. We will review some of these updates.

1. Logged into the instance as System Administrator, open **Settings** from the banner frame:



2. Select **Developer**.

3. Next, toggle on the switch for **Show update set picker in header**:

4. Close out of the System Settings window and return to your main instance screen. You will now see a drop down menu in the banner frame, next to the user menu:

This is the **Update Set Picker** menu. It allows you to quickly select an Update Set to capture platform configuration changes.

NOTE: *If it is not displaying the* ***Default*** *Update Set, it should be displaying the* ***ServiceNow Fundamentals*** *Update Set.*

---

5. **System Update Sets > Local Update Sets.**
6. Locate and open the **ServiceNow Fundamentals** Update Set.
7. Notice how many total **Customer Updates** have been collected.

   How many updates are there?
8. Find the updates for all of the script types created in a previous lab: **UI Policy**, **UI Policy Action**, and **Business Rule**.

   What other items are captured in the Update Set?
9. Locate the incident records created during the many labs before.

   What items are not captured in the Update Set, but were created in class?

B. Create an Update Set

Now that you have seen the types of configuration changes Update Sets capture, create a new Update Set and make additional configuration changes.

1. **System Update Sets > Local Update Sets.**
2. Click **New** to open the **Update Sets** form.
3. Enter the following name for the new Update Set: **Problem Form Changes**.
4. In the Description field, enter the following description of this Update Set: **Added Updated and Updated by after Assigned to on Problem form**.
5. Click the **Submit and Make Current** button.

6. A brief confirmation message displays and the current Update Set is shown:

7. The current Update Set is also represented in the Update Set picker:

Modify the Problem Form

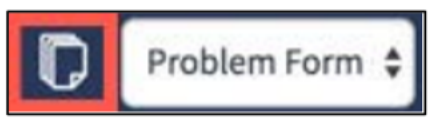
1. **Problem > Create New**.
2. From the **Form Context Menu** in the Problem header, select **Configure > Form Layout.**
3. Make the following layout changes:

   Add **Updated** after Assigned to
   Add **Updated by** after Updated

4. Click **Save**.

Mark Update Set Complete

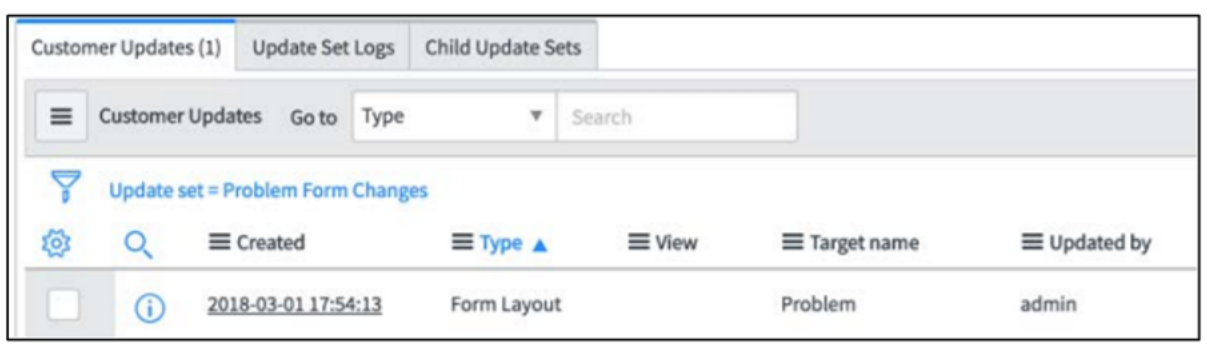
1. Click the **View current Update Set** icon, next to the update set picker, to open the update set record:



NOTE: *You could also navigate to* ***System Update Sets > Local Update Sets****, then locate and open the record.*

---

2. Notice on the **Customer Updates** tab that a new record is captured:



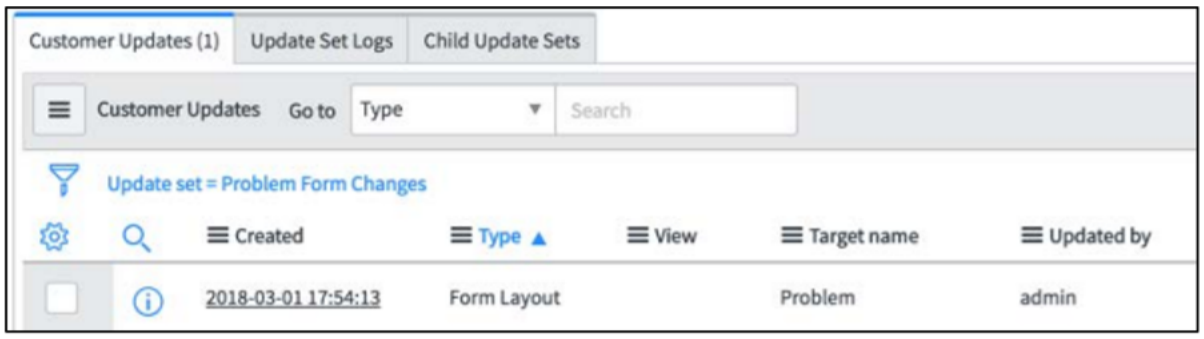
3. Change the Update Set State from In progress to **Complete.**

4. Click **Update**.

5. With the status of **Complete**, this Update Set is now ready to be retrieved by another ServiceNow instance.

Lab Verification

Problem Form Changes Update Set



Wow! You now know your way around Update Sets.

Module 5 – 5.3 Development

servicenow

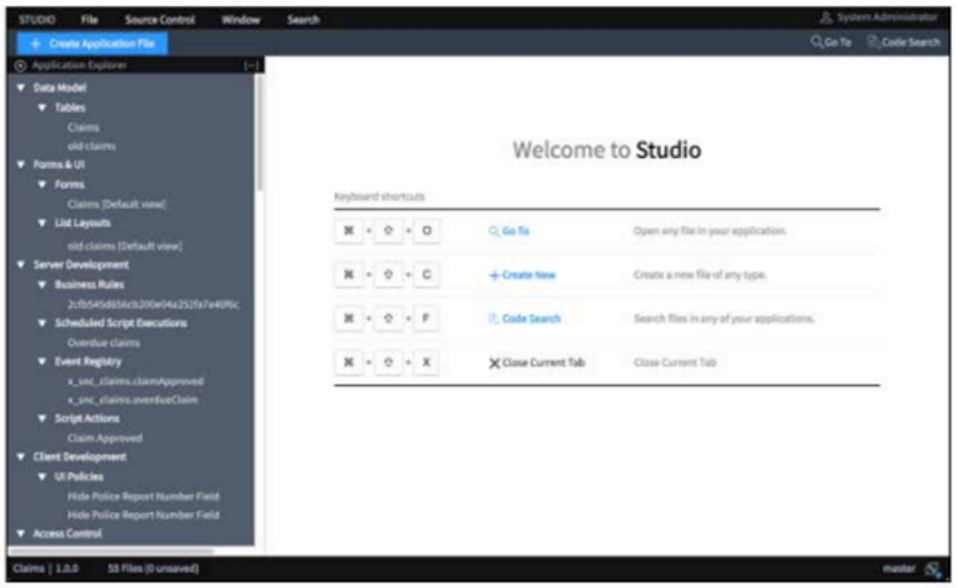
Objectives

- ServiceNow Studio
- Delegated Developers
- Application Administration
- Application Scopes
- Developer Documentation

ServiceNow Studio

ServiceNow **Studio** provides an IDE-like interface for application developers to create custom applications

Application developers can also access Studio to import or open applications



ServiceNow Studio provides an IDE-like interface (integrated development environment). It offers a simple way to identify and interact with application files, create files as you develop, and modify existing application files in a tabbed environment. Accessing Studio requires an admin or a delegated developer role.

With Studio, application developers can:

- See exactly what files comprise their application in the **Application Explorer**
- Add new files to their application using a single **Create Application File** interface
- Navigate to files using familiar search-by-name or by-type behavior with the **Go To** dialog
- Find code both within and outside an application using the **Code Search** tool
- Operate on multiple files at once using the tabbed interface
- Operate on multiple applications at once using multiple studio windows
- Publish the application to company instances or the ServiceNow Store
- View information about the current application from the **Status Bar**

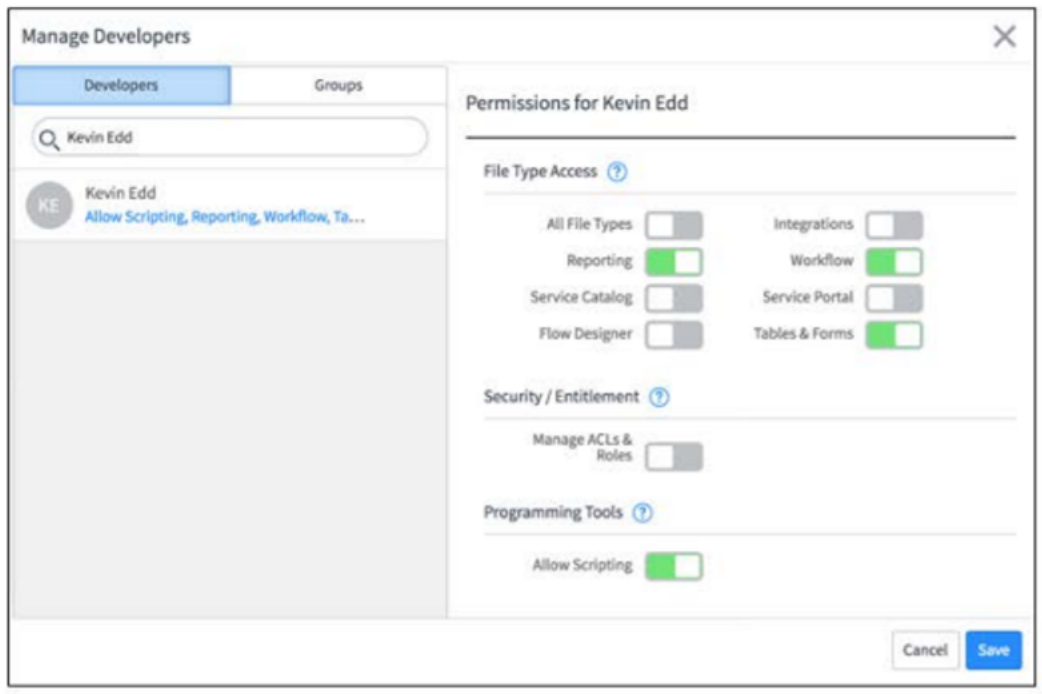
NOTE: Studio is not intended for global applications and can behave unexpectedly when editing them.

Delegated Developers

servicenow

Delegated developers are non-administrator users and groups which are assigned one or more permissions to develop applications

Each permission grants one or more delegated-development-specific roles to retain control over the system without having the admin role



Delegated developers can be granted the following permissions:

- **All File Types**: Grants the developer access to all application file types including some not granted by the other options
- **Integrations**: Grants the developer access to web service APIs, REST APIs, and data sources
- **Reporting**: Grants the developer access to reports and scheduled reports
- **Workflow**: Grants the developer access to the Workflow Editor and Activity Creator
- **Service Catalog**: Grants the developer access to catalog related file types such as catalog items, record producers, and variables
- **Flow Designer**: Grants the developer access to the Flow Designer design environment to create flows and actions. Script action steps require the **Allow Scripting** permission
- **Service Portal**: Grants the developer access to Service Portal editors and tools
- **Tables & Forms**: Grants the developer access to model and layout related file types such as table columns, form layout, and list layout
- **Manage ACLs & Roles**: Grants the developer access to security-related file types such as access controls and user roles
- **Allow Scripting**: Grants the developer write access to script fields such as those in business rules, client scripts, and Flow Designer script action steps

To manage delegated developers, navigate to **System Applications** > **Applications**, open the application record, then click on **Manage Developers**.

Application Administration

servicenow

Application administration allows organizations to protect sensitive application data by restricting how users acquire application-specific roles

Application developers and administrators can use application administration to:

- Prevent unauthorized users from accessing sensitive data such as financial records or personally identifiable information
- Restrict who can assign application roles
- Prevent admin users from assigning themselves access or bypassing existing access controls to a protected application

You can enable application administration from the application record and restrict the assignment of application roles from the user role record. Application developers should enable application administration after completing application development and before adding application records.

The application's administration role only allows users access to the application and does not include any other admin role. Someone must assign an application user an admin role before that user can perform typical administration tasks such as configuring form and list layouts, making changes to application tables and fields, and assigning the application admin role to new users

If you do not want the application administrator to have the admin role, the application administrator can make themselves a delegated developer. Once a delegated developer, the application administrator can perform a subset of administrative tasks without having the admin role.

Additionally, admin users can be prevented from:

- Assigning themselves a protected application role or to a group containing said role
- Overriding or bypassing existing access controls to a protected application by creating new access controls
- Impersonating or changing the password of users who have a protected application role
- Inheriting a protected application role
- Running scripts that access protected application records

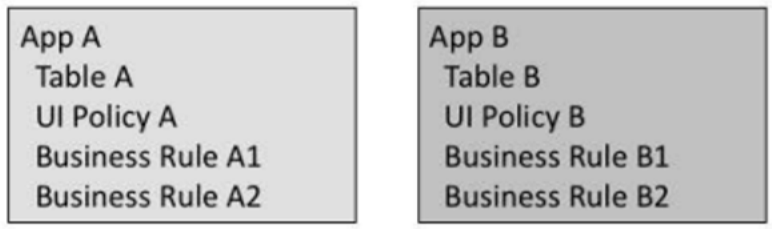
Application Scopes

servicenow

Application scoping protects applications by identifying and restricting access to available files and data

Applications developed prior to application scoping are in the **global scope**

All custom applications have a **private scope** that uniquely identifies them and their associated artifacts



Administrators can specify what parts of an application are accessible to other applications from the custom application record and each application table record.

For example, suppose that you create a conference room booking application in its own application scope. By default, the application can access and change its own tables and business logic but other not applications unless you give them explicit permission.

The application scope ensures:

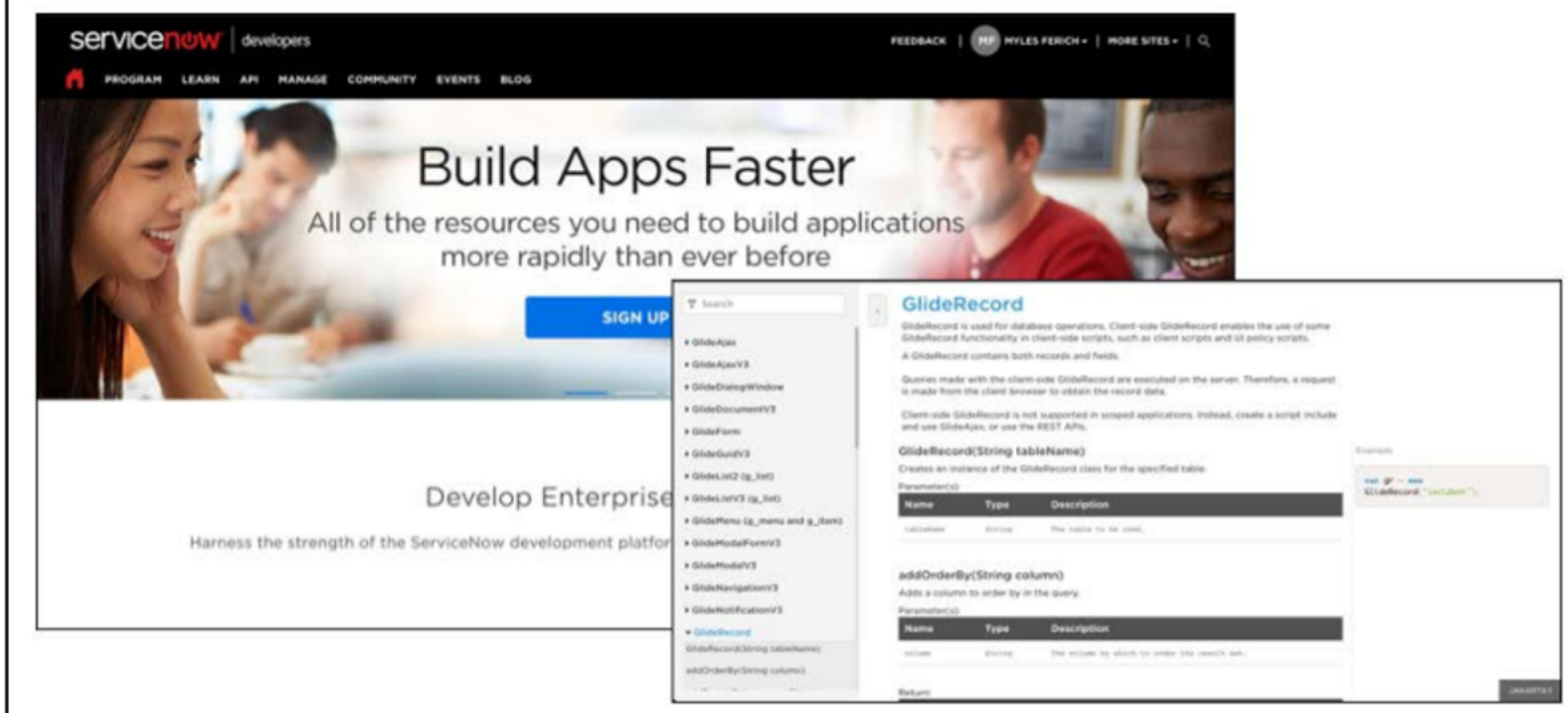
- The conference room booking application does not interrupt core business services
- Other applications do not interfere with its normal functioning

By default, all custom applications have a private scope that uniquely identifies them and their associated artifacts with a namespace identifier. The application scope prevents naming conflicts and allows the contextual development environment to determine what changes, if any, are permitted. Application developers specify an application scope when they create an application.

The global scope is a special application scope that identifies applications developed prior to application scoping or applications intended to be accessible to all other global applications.

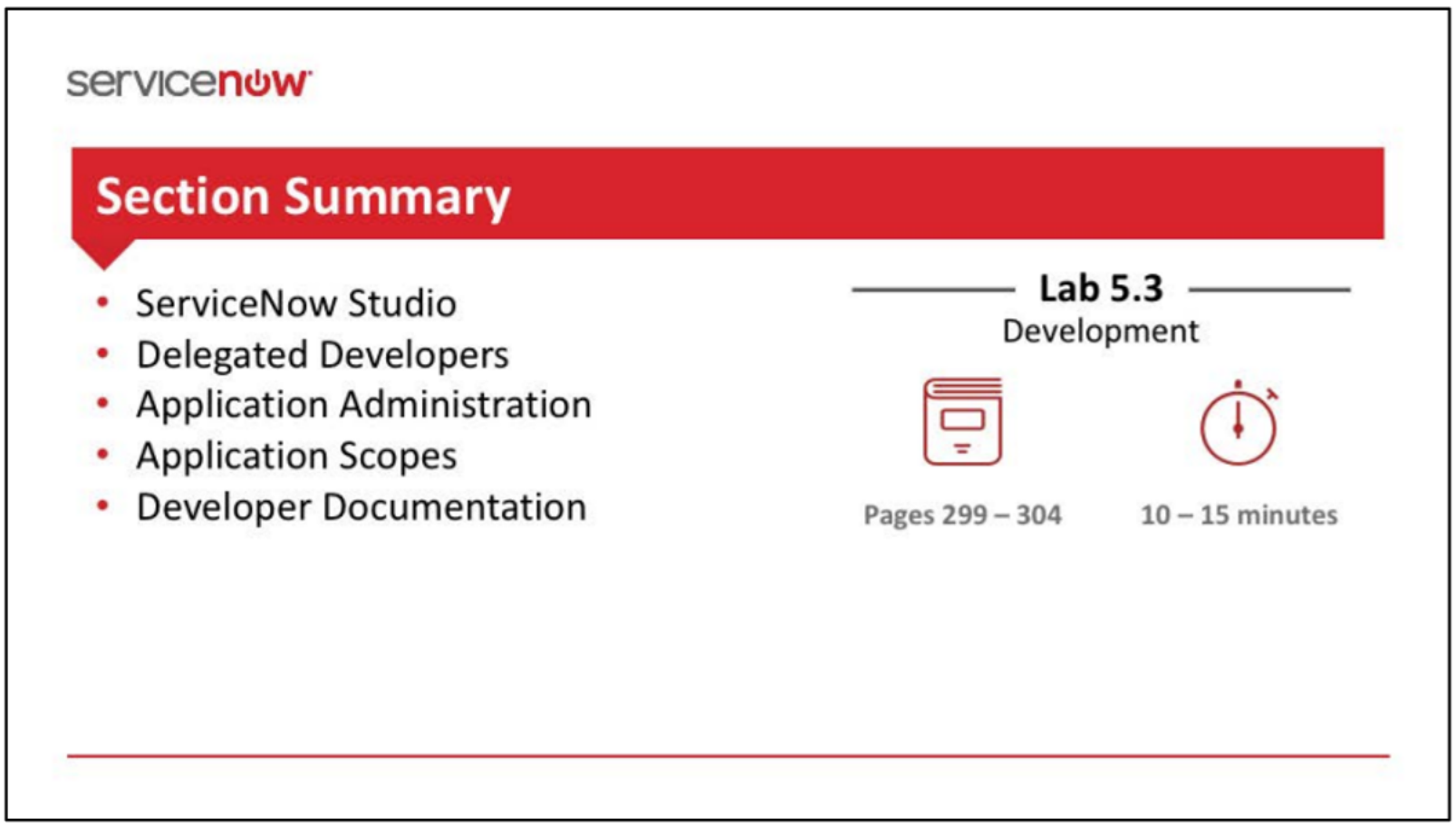
The system adds a namespace identifier to the front of application artifacts such as tables, scripts, and configuration records.

Documentation: Developer



developer.servicenow.com is a great resource for developers, but also anyone interested in developing applications within ServiceNow.

There are great resources here related to development, including: scripting API references, free training and documentation, and access to a free, personal developer instance.



Lab 5.3 – Development:

- Sign up for a developer instance
- Define an Update Source
- Retrieve an Update Set, committing platform changes to the developer instance

LAB

Development 5.3

10 – 15 minutes

Lab Goal

Lab Dependency: Requires the completion of Lab 1.2.

This lab will show you how to do the following:

- Sign up for a developer instance
- Define an Update Source
- Retrieve an Update Set, committing platform changes to the developer instance

To end class, you will be walked through the steps to sign up for a developer instance on the ServiceNow Developer Portal.

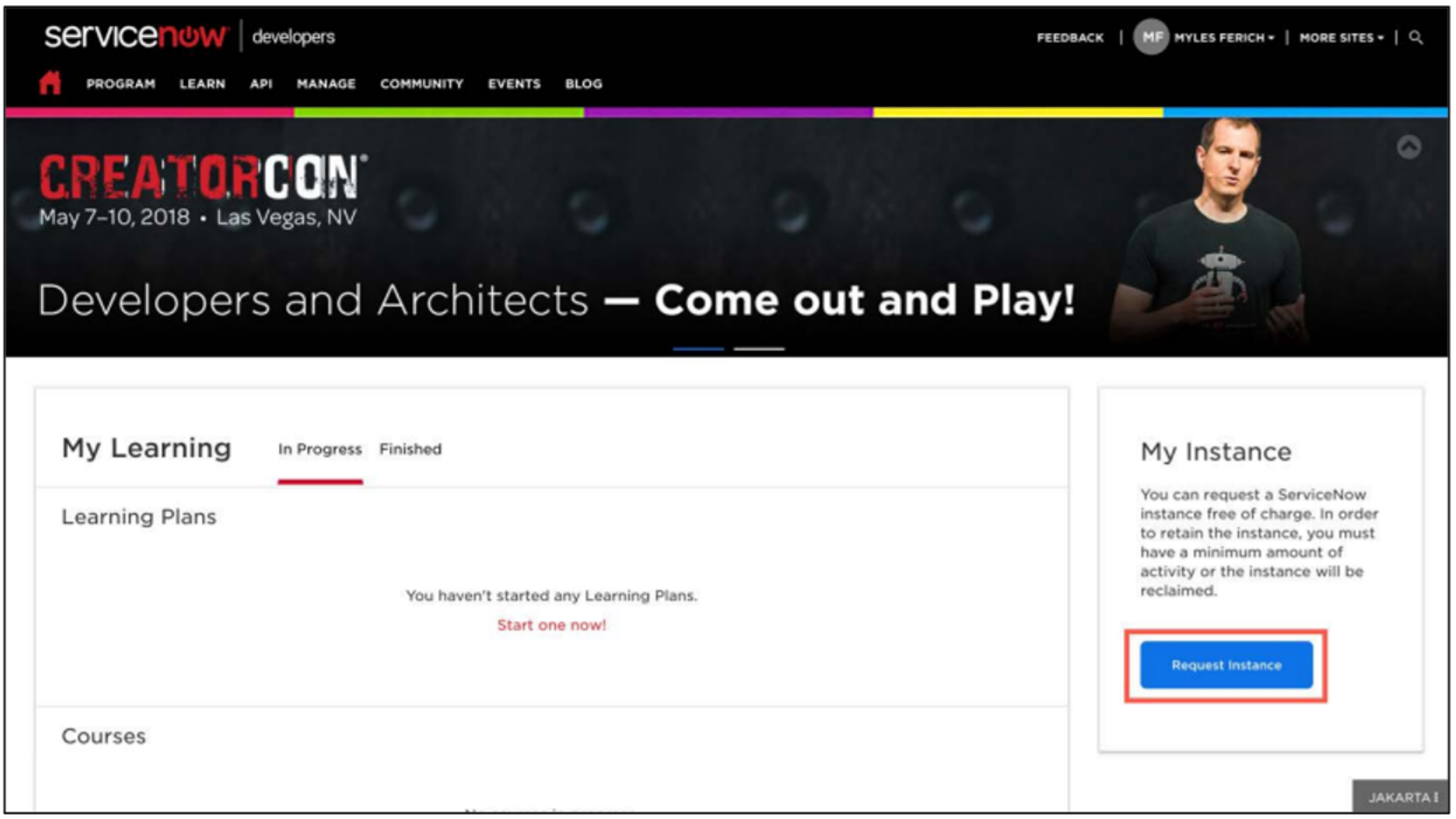
You will also practice the procedure of retrieving completed update sets from one instance of ServiceNow to another. This emulates the experience of taking configuration changes made to the platform of one instance and pulling them into another – just like pulling work from a development instance into production.

A. Register for the ServiceNow Developer Program

Even if you are not a developer or plan to create custom applications in ServiceNow, once you have access to a developer instance you will be able to use it to revisit the topics discussed in class, as well as have fun within *your very own* instance of ServiceNow!

1. Go to **https://developer.servicenow.com/**
2. Select **Register** from the upper right-hand menu, then fill out the form, then finally read and agree to terms of use before clicking **Submit**.
3. Look for and open an email from ServiceNow (signon@service-now.com).
4. Select the link in the email message to validate and activate your account.
5. Sign in with your username and password created in step 2, then read and accept the **ServiceNow Developer Agreement**.
6. Answer a few questions to maximize your experience, then click **Submit**.

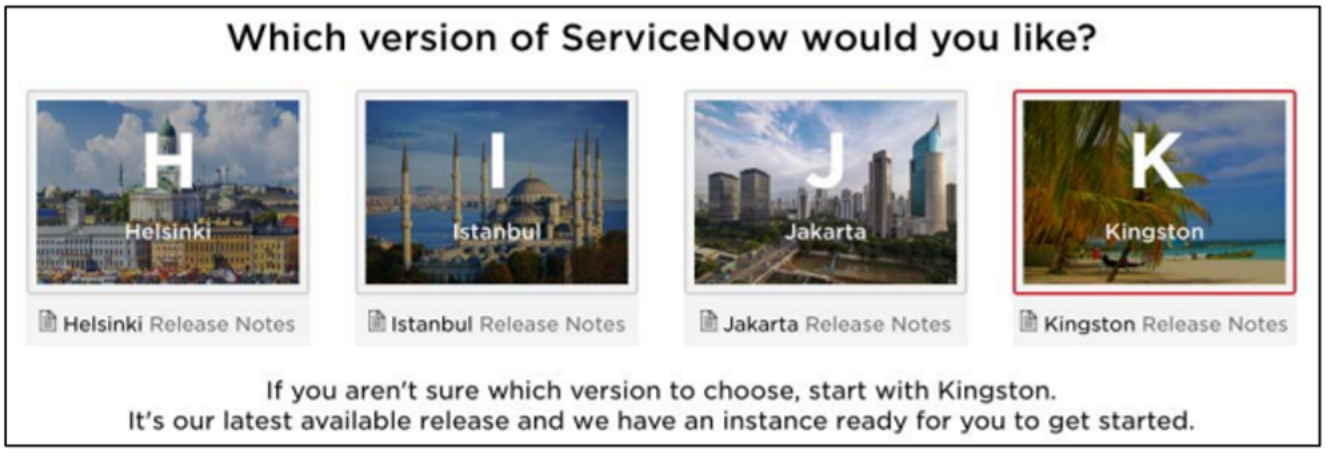
7. Under My Instance, click **Request Instance**:



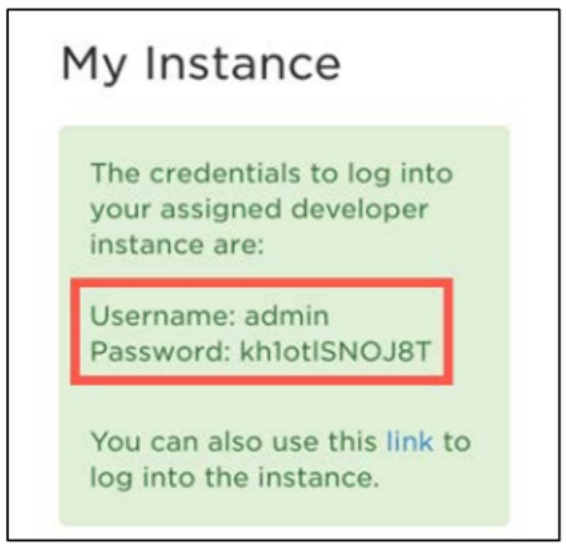
***NOTE**: If this page does not display, you can click on the ServiceNow logo or, from the main menu, select **Manage > Instance**.*

8. Complete the form to tell ServiceNow how you will use the personal developer instance, then click **I understand**.

9. You may choose any available version of ServiceNow you would like, but **Kingston** is recommended:



10. After selecting the ServiceNow version of your choice, the credentials to your personal developer instance will appear under the **My Instance** section:



***IMPORTANT**:*
Capture your username and password as it will not be available after the next step.

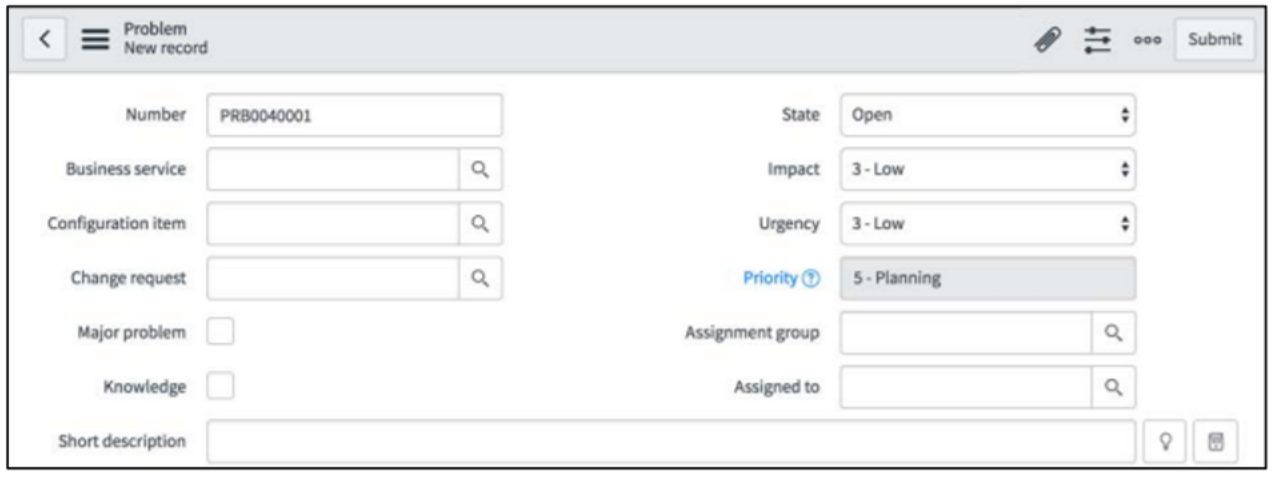
11. Under the instance log in credentials, click the **link** to log into the instance directly:

12. Use the credentials above on the login screen, then change the temporary password to successfully sign into your instance as system administrator.

B. Define an Update Source

1. Within your developer instance, navigate to **Problem > Create New**.

2. Observe that the **Updated** and **Updated by** fields are not displayed on the problem form yet:

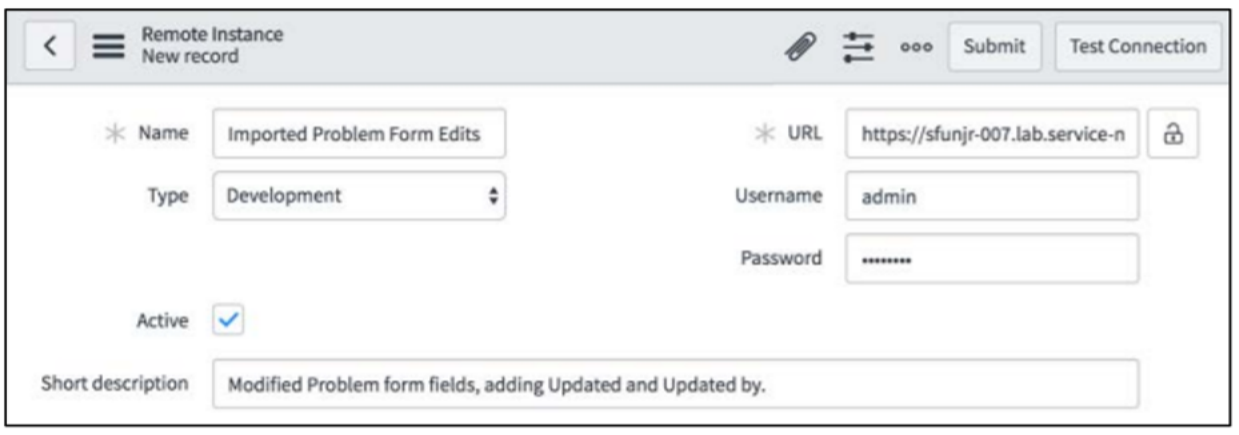


3. **System Update Sets > Update Sources**.
4. Click **New**.
5. Name: **Imported Problem Form Edits**
6. Click the padlock to the right of the URL field.
7. Input your course lab instance URL: https://instance-###.lab.service-now.com/
8. Enter this instance's admin **username** and **password**.

NOTE: *These are the system administrator credentials your instructor provided, and which you have been using during class.*

---

9. Type in a brief description of the Update Set into the Short Description field: **Modified Problem form fields, adding Updated and Updated by.**
10. Your form should look like this:



11. Click the **Test Connection** button.

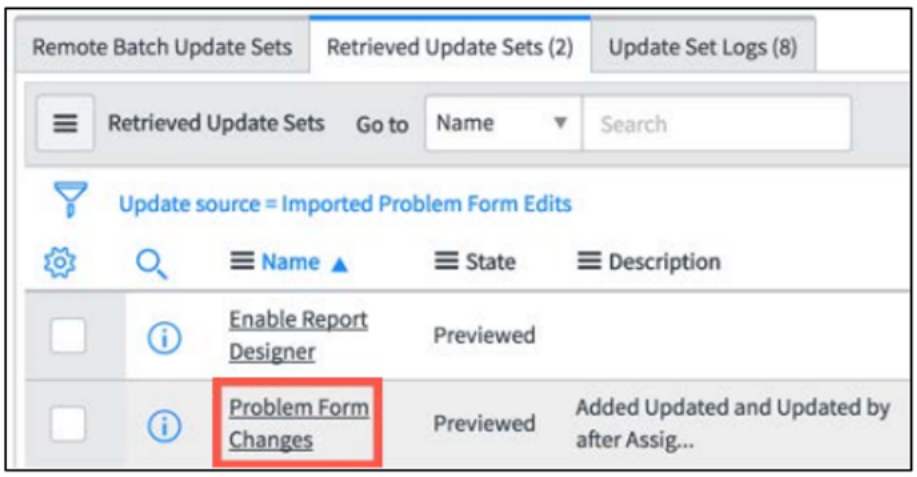
NOTE: *A brief* ***Connection is OK*** *message will display, which may also include language about varying version snapshots.*

---

12. **Save**.
13. Click **Retrieve Completed Update Sets**.
14. Close the **Retrieve Update Sets** progress pop-up once it has reached 100%.

Locate Retrieved Update Sets

1. From the **Retrieved Update Sets** tab, open the **Problem Form Changes** record:



2. From the form header, click the **Commit Update Set** button.

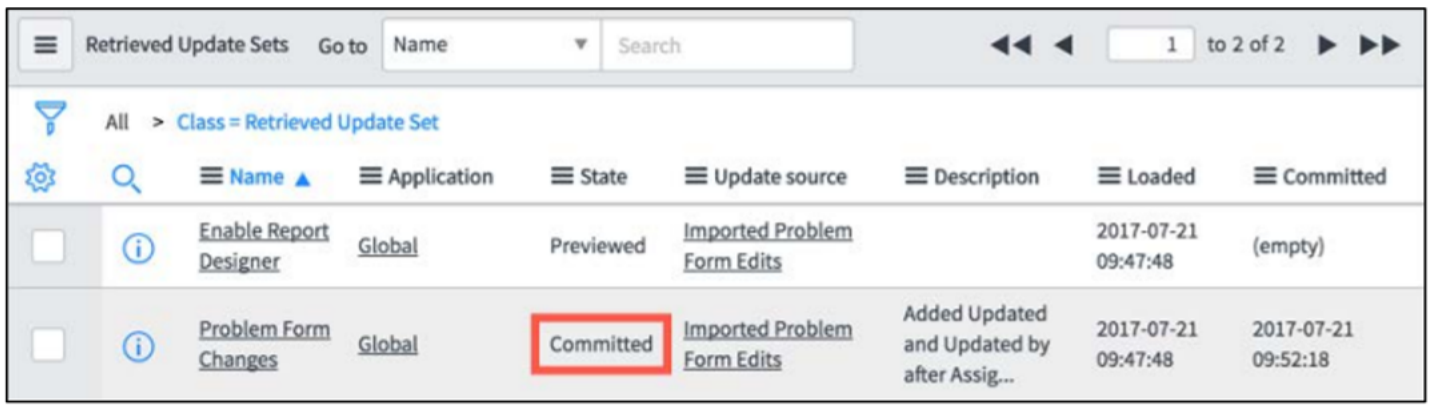
***NOTE**: This update set has already been previewed – checked to make sure its contents do not interfere with the target instance. If your developer instance is not running Kingston, you will need to click **Preview** and will get a collision error. You may accept the remote update and **Commit** to continue with the lab.*

---

3. Close the **Update Set Commit** pop-up when its succeeded.

4. **System Update Sets > Retrieved Update Sets**.

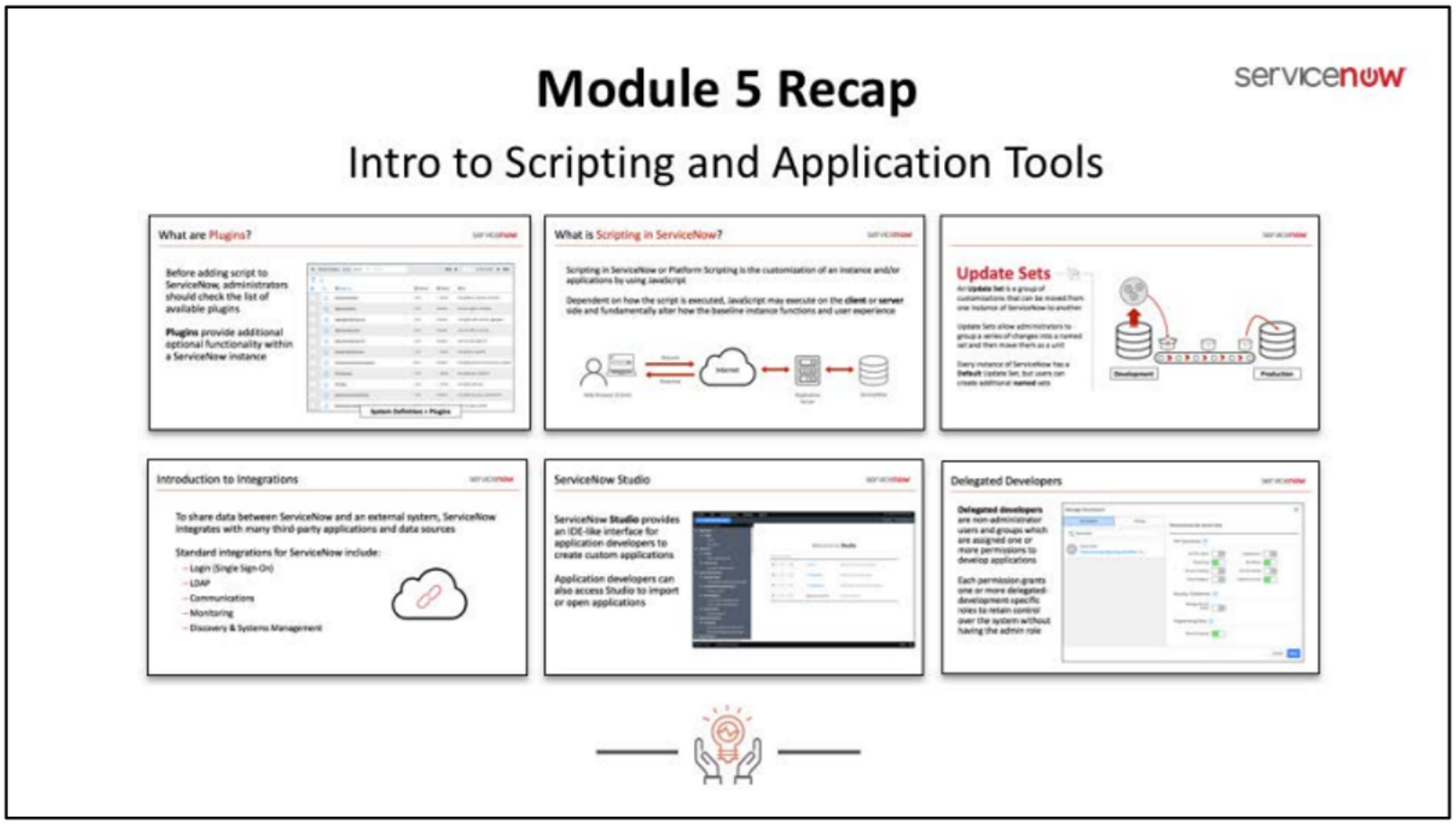
5. You should see the new Update Set with a **Committed** state:



Lab Verification

1. **Problem > Create New**.
2. You should now see the **Updated** and **Updated by** form fields:

Congratulations, you have completed the Development Lab, and the ServiceNow Fundamentals course!



For these selected topics, discuss:

Why would you use these capabilities?

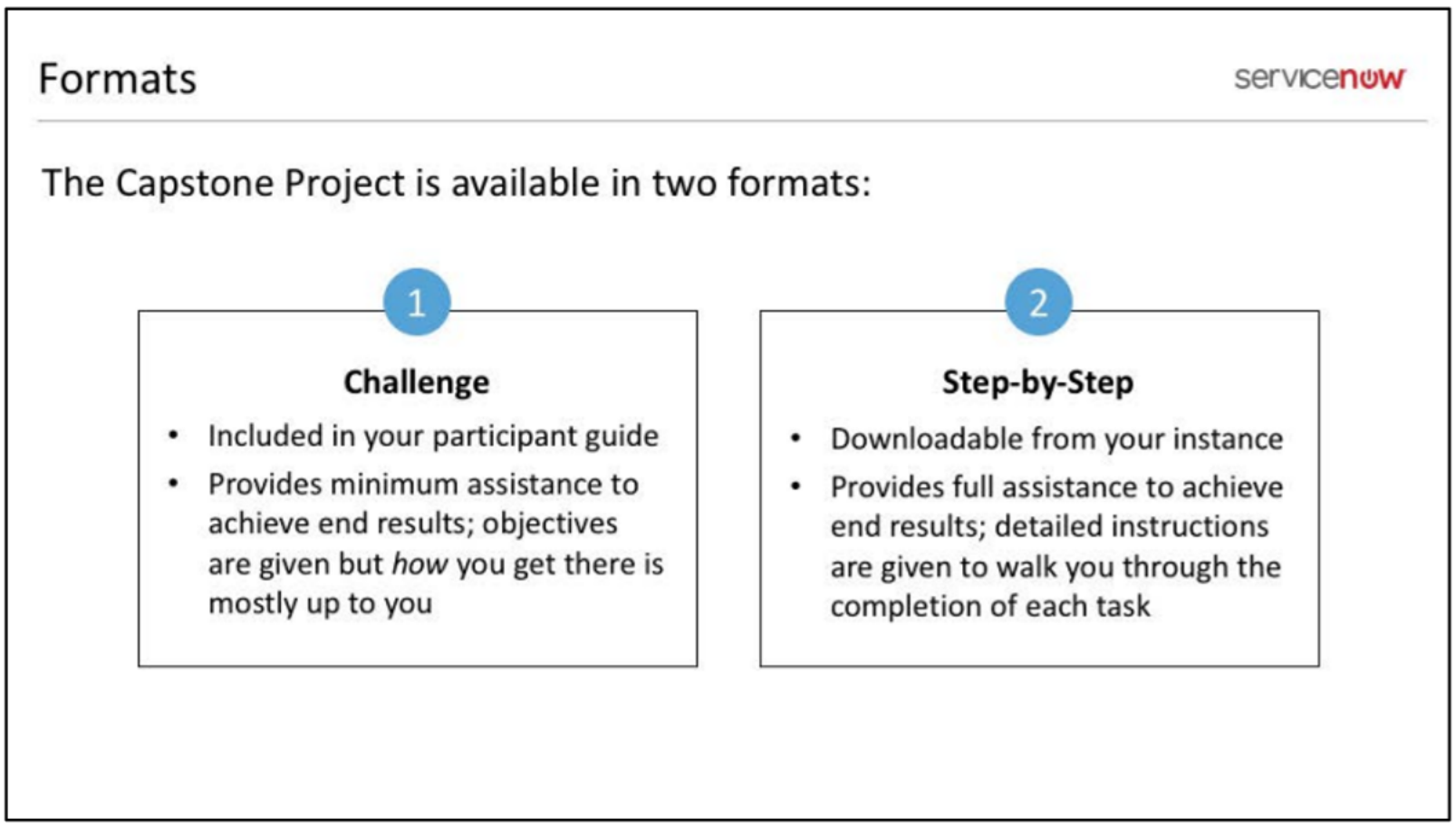
When would you use these capabilities?

How often would you use these capabilities?

The Capstone Project is a multifaceted **homework** assignment.

Specifically, eight typical implementation tasks which serve as a culminating project where you can safely apply your newfound ServiceNow Fundamentals knowledge in a less guided, more experiential, set of lab exercises.

As a prerequisite step, and as was with your last in-class lab, you will need a free ServiceNow Developer Instance.



The **Capstone Project: Challenge Format** is included in your participant guide. We strongly recommend that you try to solve each Capstone Project task using just the Challenge format. If you have difficulty completing a task, you can refer back to slides, notes, and labs.

If you find yourself still struggling, or prefer to complete the tasks "by the book" then no worries! Step-by-Step task solution guides are available. Locate and download these solution guides from your class lab instance. They are located under **Capstone Project** in the **ServiceNow Fundamentals Class** Knowledge Base.

Scenario for the Capstone Project

servicenow

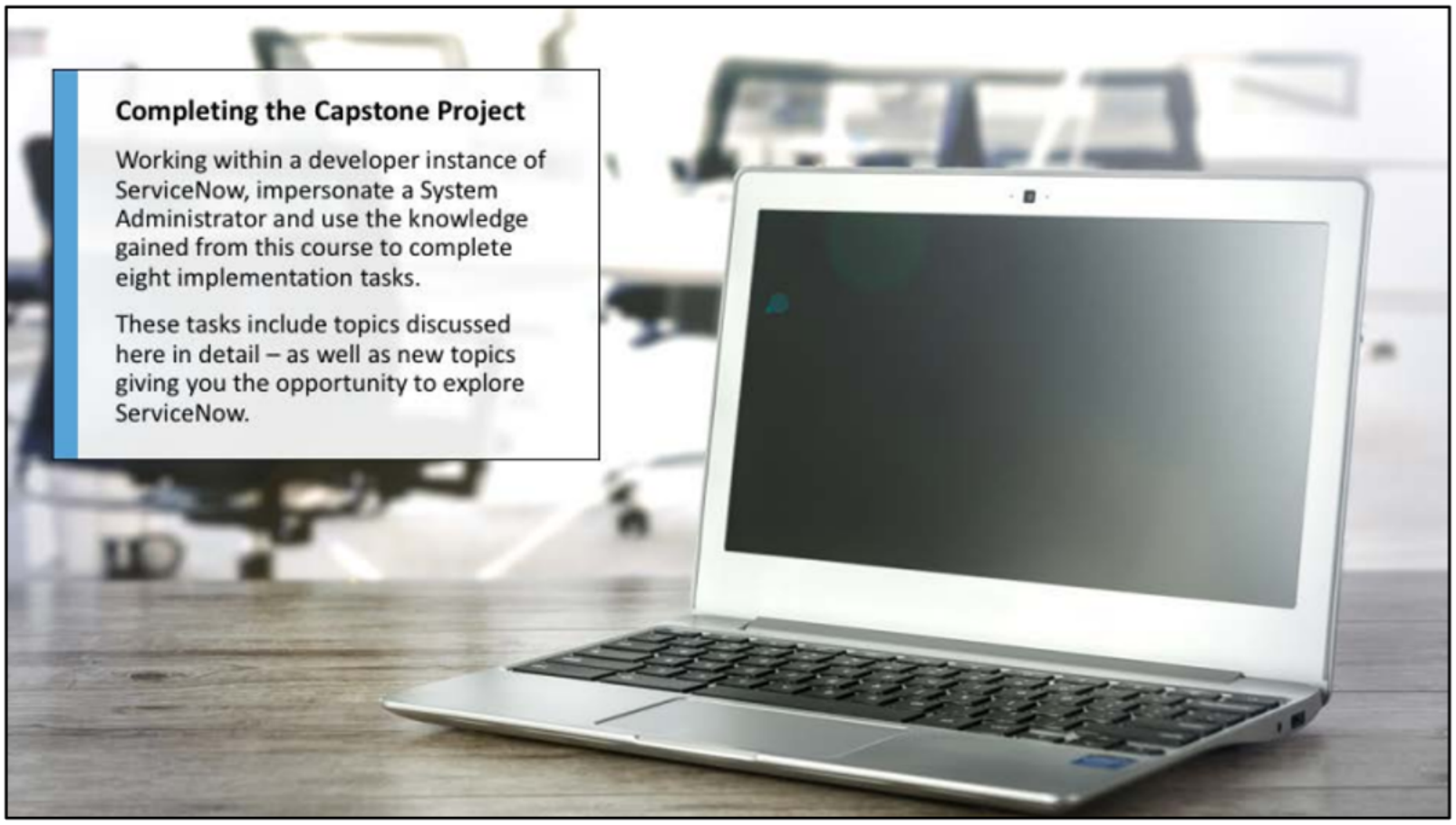
Cloud Dimensions has developed and tested a series of Infinity devices, used as an example throughout this course

After great success demonstrating said devices at various trade shows and conferences, Cloud Dimensions has officially begun production on a new product; the **Infinity Alidade**

Cloud Dimensions needs a software solution to support new business processes, which are still being developed, and requires that the solution be able to scale appropriately in order to support ongoing, rapid growth.

ServiceNow has proven to be the best solution to meet Cloud Dimensions current requirements and support plans for ongoing expansion. Therefore, Cloud Dimensions is ready to move forward with a ServiceNow implementation for their latest product, the **Infinity Alidade**.

A worldwide product launch date is aligned with the ServiceNow Go-Live date, so as a Cloud Dimensions System Administrator, you have been tasked with the responsibility to customize the Cloud Dimensions instance in support of technical needs for employees and customers alike!



The ServiceNow Fundamentals Capstone Project is meant to be completed on a developer instance of ServiceNow. Refer to **Lab 5.3 – Development** in the course for registering in the developer program.

Remember once you start the Capstone Project: this is suppose to be a fun exercise! It is meant to test the knowledge you gained in class and, most importantly, give you a chance to look around in the ServiceNow platform.

If you start with the **Challenge** format as suggested, remember to use all of the resources discussed in class, including docs.servicenow.com.

Enjoy!

ServiceNow Fundamentals
Capstone Project: Challenge Format

Capstone Tasks Overview

The Capstone has been divided into eight task categories to guide your deployment:

Task 1 – Customize Your Instance
Module 1: User Interface & Navigation

Task 2 – Update Incident Management
Module 1: User Interface & Navigation

Task 3 – User Administration
Module 1: User Interface & Navigation
Module 2: Task Management

Task 4 – Update Service Catalog
Module 4: Service Automation Platform
Module 6: Ongoing Maintenance

Task 5 – Update Knowledge Base
Module 4: Service Automation Platform

Task 6 – Configure the Mobile UI
Module 1: ServiceNow User Interface
Module 4: Service Automation Platform

Task 7 – Enhance Task Assignment and Communication
Module 5: Core Application Administration

Task 8 – Schedule a Report
Module 3: Data Management and Reporting

These tasks correlate to topics found in the ServiceNow Fundamentals materials; parent modules noted.

To successfully complete the Capstone Project, you must select the **Capstone Project** article in the **ServiceNow Fundamentals Class** Knowledge Base of your lab instance.

This will download a zip file contains file to your local machine titled **Capstone Project** which contains relevant task files, as well as the **Step-by-Step Solution** guides for every task.

Task 1: Customize Your Instance

Customize the instance to feature Cloud Dimensions branding styles that are familiar to both employees and customers.

Additionally, you will create an organization company record to include contact information and new welcome page content.

Configure Company Settings and Welcome Page

Use the **ITSM Guided Setup** to configure the following system properties:

- Page header caption: **Infinity Service Portal**
- Banner tab title: **Infinity Service Portal**
- Banner image for UI16: **cd_sp.jpg**
- Header background color: **#387bcc**

Create new Welcome Page Content text for all users on the login page to say:

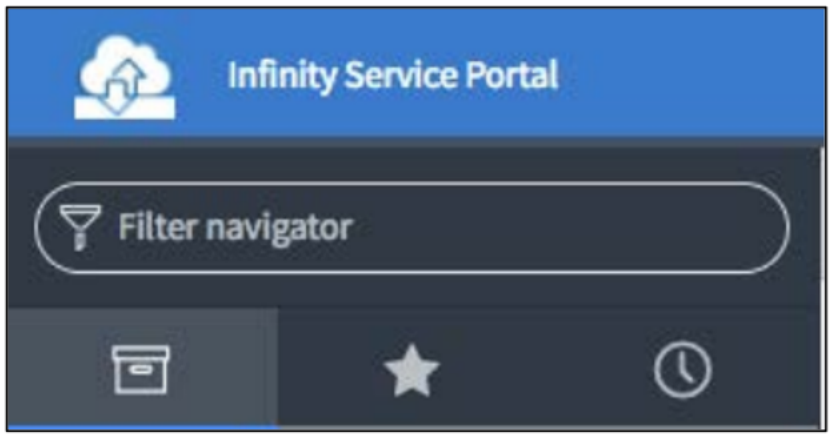
- Short description: **Welcome to Cloud Dimensions**
- Text: **Welcome to the home of Infinity! If you are an employee of Cloud Dimensions, please use your company login credentials to enter.**

Create New Organization Company Record

Use the **Organization** Application to create a company record with the following company information:

- Name: **Cloud Dimensions**
- Phone: **800-555-5555**
- Street: **3260 Jay Street**
- City: **Santa Clara**
- State / Province: **CA**
- Zip / Postal code: **95054**

TASK VERIFICATION



Welcome to Cloud Dimensions
Welcome to the home of Infinity! If you are an employee of Cloud Dimensions, please use your company login credentials to enter.

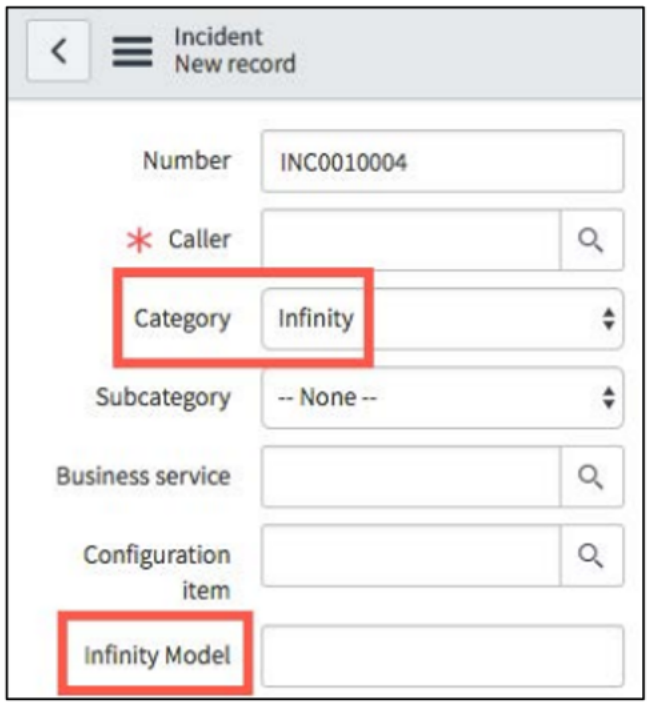
Task 2: Update Incident Management

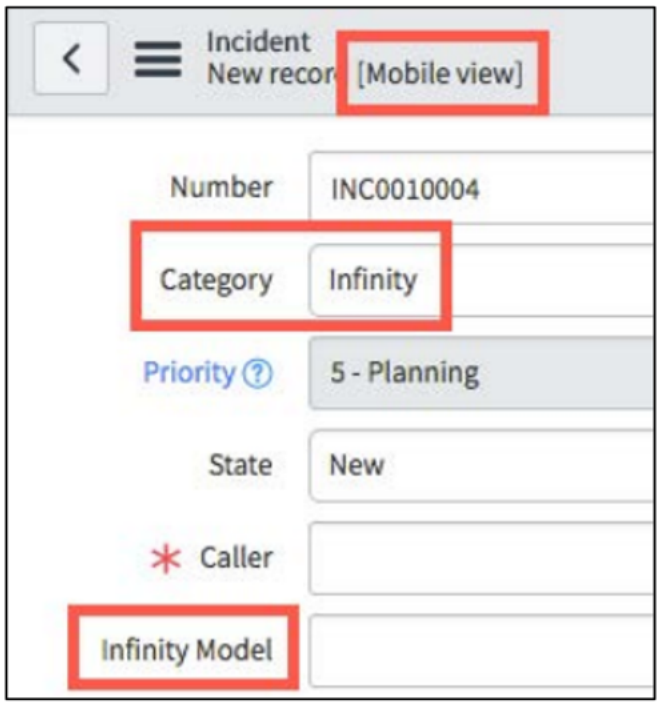
Modify the Incident form so that it can support a new process for troubleshooting technical issues reported by Infinity users.

Modify the Incident Form

1. Create a new field and add it to both the Default and Mobile View of the Incident form with the following properties:
   - Name: **Infinity Model**
   - Type: **String**
   - Field length: **Small (40)**
2. Configure the Default View and place the **Infinity Model** field beneath the **Configuration item** field
3. Configure the Mobile View and place the **Infinity Model** field beneath the **Caller** field
4. Modify the **Category** field on the Incident form to include a new **Infinity** choice.

TASK VERIFICATION





Task 3: User Administration

Create a new user group that is responsible for troubleshooting Infinity incidents and fulfilling Service Catalog Infinity requests.

Create Users, Groups, and Roles

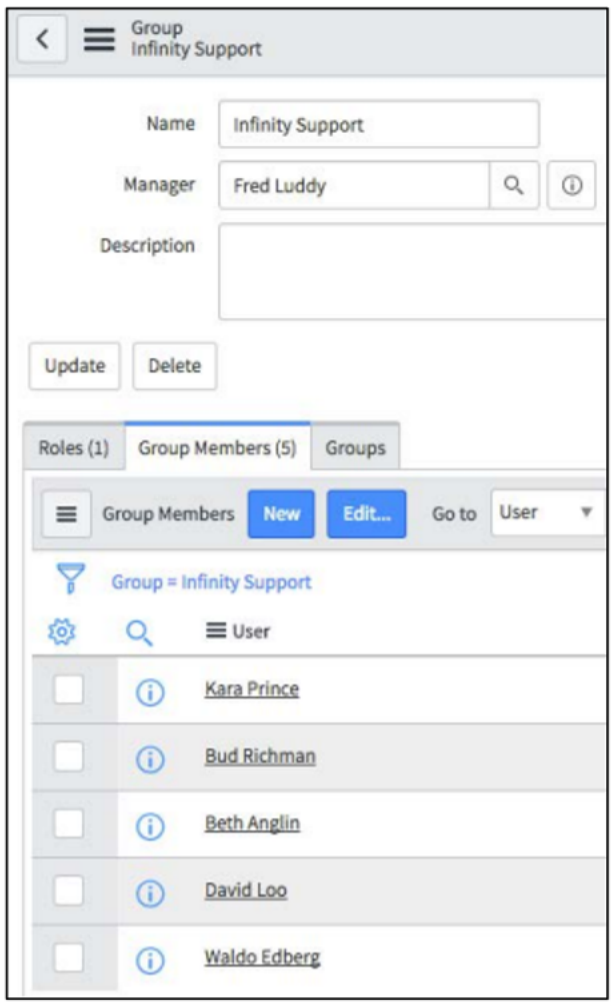
Under the existing **Service Desk** group, create a new child group called **Infinity Support** that includes the **itil** role and has Fred Luddy as the group manager.

Then **create or add** the existing users to the group:

- Beth Anglin
- Bud Richman
- David Loo
- Kara Prince
- Waldo Edberg

Additionally, set **Fred Luddy** as the Manager under Kara Prince's user record.

TASK VERIFICATION



Task 4: Update Service Catalog

Import an **Infinity Prototype** item into the Service Catalog to be requested, and develop a workflow to support and complete the fulfillment process.

Import Service Catalog Item

Import an Update Set **(cd_infinity_catalog_item.xml)** containing the Infinity Prototype Service Catalog item into the instance. Remember to **retrieve**, **preview**, and then **commit**!

Create a New Workflow

Develop a workflow to fulfill internal Infinity Service Catalog requests. It should feature the following activities in this particular sequence:

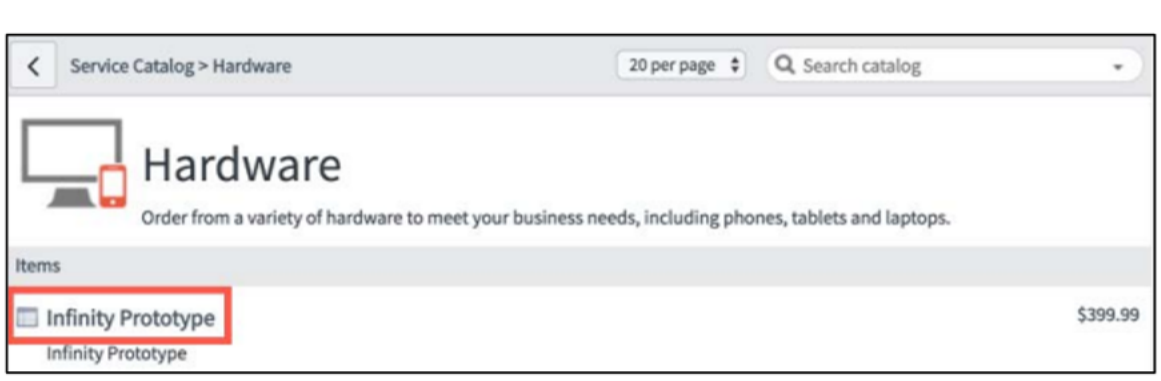
- Approval by the requester's manager
  - The **Approved** path marks the requested item as approved with an **Approval Action**

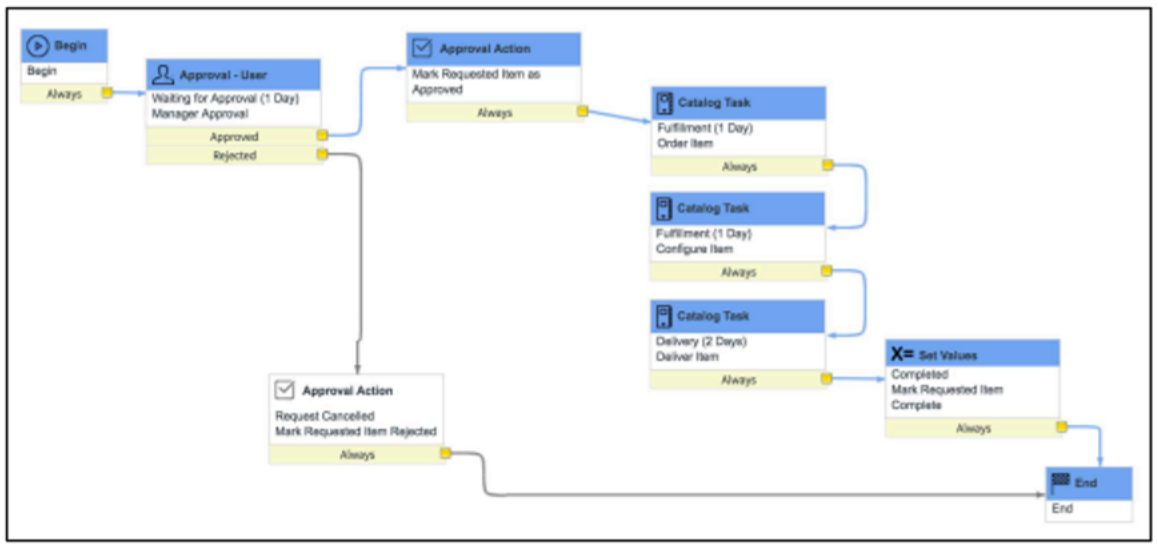
    - The **Rejected** path marks the requested item as rejected and then ends the workflow
- Continuing the Approved path are three Catalog Tasks:
    - **Catalog Task 1** details the steps for ordering the Infinity item and is assigned to the **Infinity Support** group
    - **Catalog Task 2** details the steps for configuring the Infinity and is assigned to the **Infinity Support** group
    - **Catalog Task 3** details the steps for delivering the Infinity and is also assigned to the **Infinity Support** group
- Upon completion of all three catalog tasks, mark the requested item as complete with a **Set Values** activity

  Validate and test the Workflow to successfully order, fulfill and deliver an Infinity.

  NOTE: Test by submitting a request for the Service Catalog item by a user with a manager and then impersonating every stakeholder involved, as determined by the activities created above, to ensure it would work in a real-life situation.

TASK VERIFICATION





Task 5: Update Knowledge Base

Populate the Knowledge Base with a new category that will contain two articles to support internal requesters and fulfillers. Additionally, you must abide by company security protocols and ensure the information is accessible by the appropriate parties.

NOTE: By default, all Knowledge Base articles must go through a Review Process before they are published. You will enable automatic publishing on the Knowledge Base level.

Enable Automatic Publish

Adjust the settings for the IT Knowledge Base to allow for Knowledge Base articles to instantly publish upon submission – using a publish workflow to bypass any review period.

Create a Requester Article

Create an article containing instructions for requesting an Infinity through the ServiceNow Catalog, before clicking the Publish button on the article form. This article must be available to all users and located in the IT Knowledge Base under the **Infinity** category. It should contain the following properties:

- **Short description:**
  Requesting an Infinity from the Service Catalog

- **Text:**
  To request an Infinity, navigate to **Self-Service > Service Catalog.** Then, click on the **Hardware** category and locate the **Infinity** item. You may also use the Service Catalog search field to locate the item.

  Click on the Infinity item name to open the ordering screen where you can customize your request. Once satisfied, click the **Order Now** button on the right-hand side.

Create a Fulfiller Article

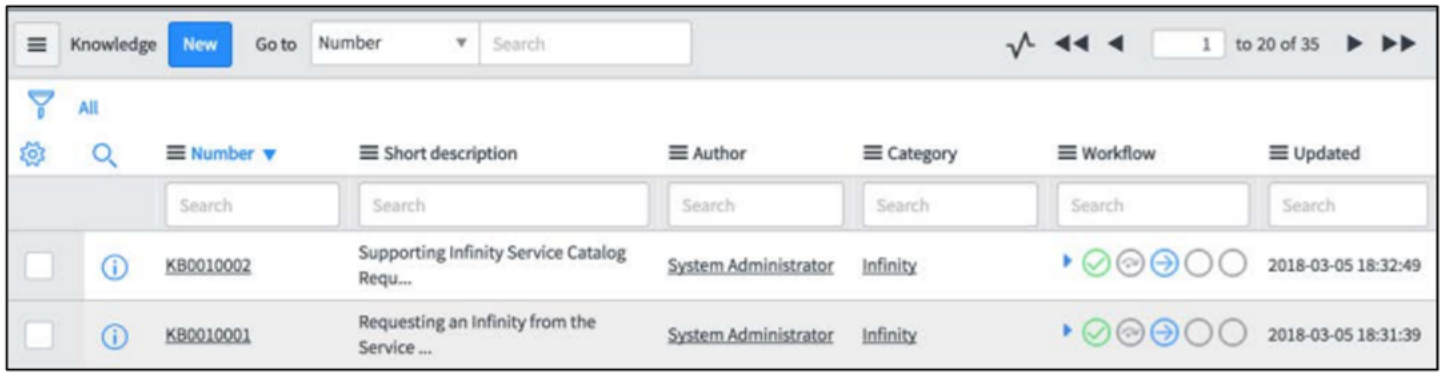
Create an article containing instructions for supporting inquiries about Infinity requests through the ServiceNow Catalog, before clicking the Publish button on the article form. This article must be available to all users with the **itil** role and located in the IT Knowledge Base under the **Infinity** category. It should contain the following properties:

- **Short description:**
  Supporting Infinity Service Catalog Requests

- **Text:**
  If the requester has a question about requesting an Infinity, redirect them to the other Knowledge Base article: **Requesting an Infinity from the Service Catalog**.

  If the requester has placed an order and would like to know about their request, please have them contact infinityorder@cloudd.com.

TASK VERIFICATION



Task 6: Configure the Mobile UI

Configure the Mobile UI to allow users to request an Infinity from the Service Catalog and create a custom application menu with a module to track and manage active Infinity incidents.

Publish a Service Catalog Item to the Mobile Interface

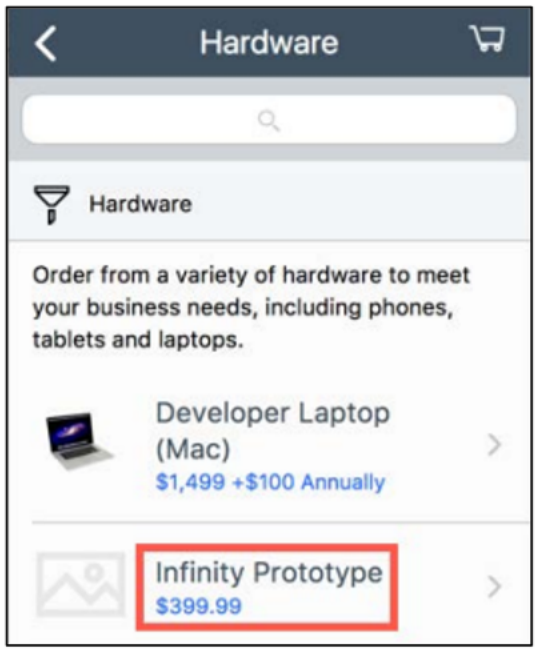
Update the **Infinity** Service Catalog item and set its availability to display in both the **Desktop and Mobile** User Interfaces.

Develop an Application Menu for the Mobile Interface

The application menu should be named **Infinity Incidents** and limited in access to only those users with the **itil** role. It should contain one module named **Active** that displays all incident records that are active and associated with the **Infinity** category.

Create a test incident by Kara Prince which has the Infinity category and a short description **My Infinity will not turn on.**

TASK VERIFICATION



Task 7: Enhance Task Assignment and Communication

Define an assignment rule that automatically assigns incidents to the Infinity Support group, if the category is Infinity.

Then, develop an email notification related to new critical Infinity incidents assigned to the Infinity Support group. Afterwards, test to ensure the email sends correctly.

Define an Assignment Rule

Define an assignment rule with the following details:

- Name: **Infinity Incidents**
- Condition: **Category | is | Infinity**
- Assignment group: **Infinity Support**

Create an Infinity Priority 1 Incident Notification

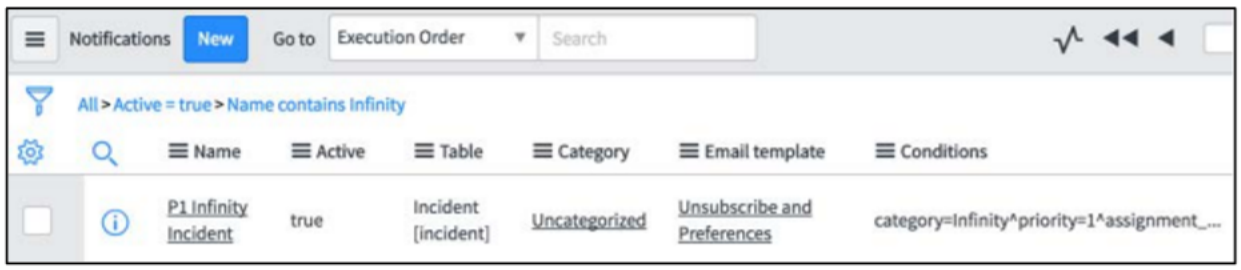
Create an email notification containing details about newly opened Priority 1 incidents that have **Infinity** as the category. This notification should go to the current **Infinity Support**

Manager only when a new Priority 1 Infinity incident is created and assigned to the Infinity Support group. The notification should contain the following properties:

- Name: **P1 Infinity Incident**
- Subject: **A new P1 Infinity Incident has opened: ${number}**
- The **Message HTML** text should contain a collection of dynamic information listing:
  - when the incident was opened
  - who opened the incident
  - the description of the incident

Verify the email is sending to the Infinity Support Manager by creating a new Priority 1 Infinity incident assigned to the Infinity Support group and checking the System Outbox.

TASK VERIFICATION



Task 8: Schedule a Report

Work with the Report Designer to create a report which displays the number of incidents that are active and tied to the Infinity category. Additionally, group the data by priority.

Schedule the report by sharing it with the Infinity Support group every Monday to coincide with their incident review meeting.

Create a Report

Set the following properties for the new report:

- Name: **Active Infinity Incidents by Priority**
- Data source: **Incident [incident] Table**
- Type: **Pie**
- Group by: **Priority**

Schedule the Report

Schedule the Report to run weekly, every Monday at 8:30am, and sent to the **Infinity Support** group.

Add the following schedule details which will appear within the email containing a copy of the report:

- **Subject**:
  Current Infinity Active Incidents Count

- **Introductory message**:
  Please find included the current count of all active Infinity incidents grouped by priority.

  This information will be discussed during the team's incident review meeting today at 9:00am.

TASK VERIFICATION