Lab Guide

SLM Best Practices for IT

Brian Dunbar & Mary Beth Grant

Default Login / Password:

admin / Knowledge17

itil / Knowledge17

employee / Knowledge17

Lab Sponsored by Microsoft Azure



This
Page
Intentionally
Left
Blank

Lab Goal

Lab 1.0
Set Up an SLA

This lab explains how to define a Service Level Agreement (SLA) based on a customer agreement. Subject areas include the following:

- Learn how schedules and workflows are used to define SLAs
- Configure the conditions in which the SLA should run
- Create a P1 incident with VIP users to show how SLAs run against records

The goal of this lab is to create an SLA that states all Priority 1 incidents opened by VIP users are to be resolved in 2 hours during business hours. The SLA allows tracking of whether incidents opened by VIP users are completed in a timely manner, ensuring that "VIP" service is provided.

Log on to the Training Instance

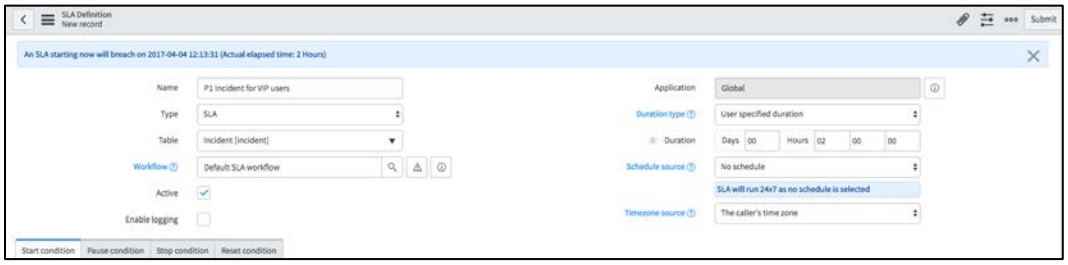
1. Navigate to the unique instance URL provided.
2. Log in using the **admin** credentials.

Create an SLA Definition

1. Navigate to **Service Level Management > SLA Definitions**.
2. From the list, click **New** to create a new SLA definition.

3. Update the form fields as follows:

   Name: **P1 Incident for VIP Users**
   Type: **SLA**
   Table: **Incident**
   Duration: **2 hours**
   Schedule source: **No Schedule** (for now)



4. Right-click on form banner and select **Save**.

Create SLA Start, Pause, and Stop Conditions

1. On the **Start Condition** tab build out the following condition to start the SLA for all Priority 1 Incidents where the caller is a VIP:

2. In the first condition statement, select **Field = Active**, **Operator = Is**, and **Value = true**.

3. Select **AND** to add another condition.

4. In the second condition statement, select **Field = Priority**, **Operator = Is**, **and Value = 1 – Critical**.

5. Select **AND** to add another condition.

6. In the third condition statement, dot walk to the VIP field on the Caller (user) record: select **Field = Show Related Fields**. (At bottom of list)

7. Go back and select **Field = Caller => User Fields**,

8. Go back and select Field = VIP, Operator = Is, and Value = True.

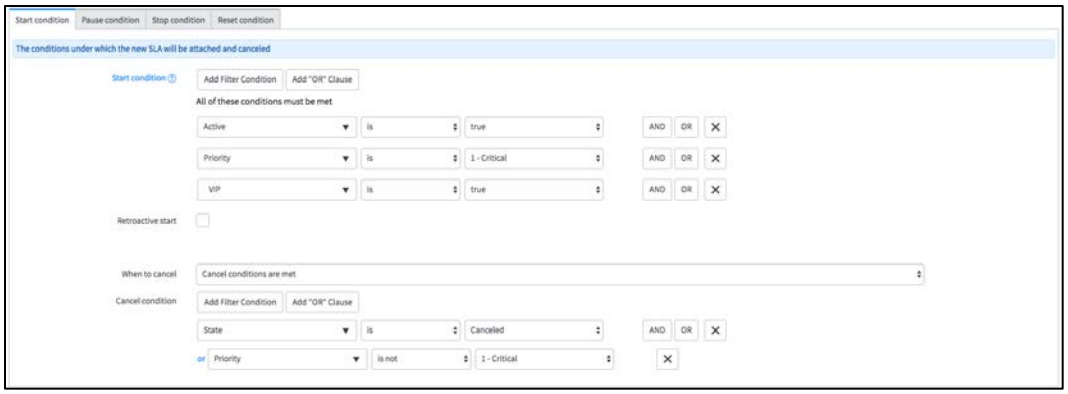
9. You want this SLA to cancel when the state of the incident is set to cancelled or if the priority changes. In the **Cancel condition,** set the following cancellation in the next steps.

10. In the first condition statement, select **Field = State**, **Operator = Is**, and **Value = Cancelled**.

11. Select **OR** to add another condition.

12. In the second condition statement, select **Field = Priority**, **Operator = Is Not**, **and Value = 1 – Critical**.

13. The tab should look like the example.

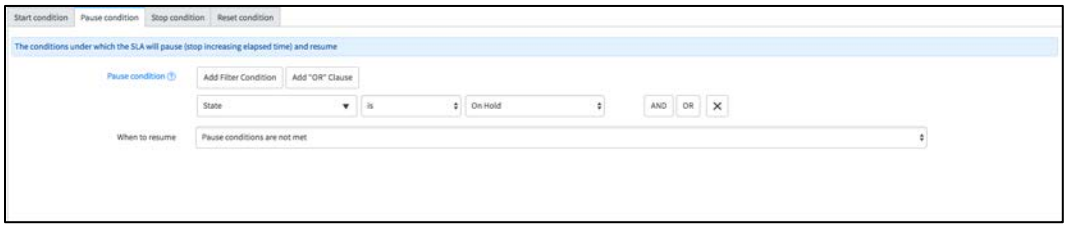


14. Next, configure the Pause Condition. Select the **Pause Condition** tab.

15. You want this SLA to pause during certain states of the Incident to pause the timer when next steps are out of our control. In this condition, select the following: **Field = State**, **Operator = is**, and **Value = On Hold**.

16. You want this SLA to resume when pause conditions are not met. On form field **When to resume**, select **Pause conditions are not met**.

17. The Pause Condition tab should look like the example.



18. Next, configure the Stop Condition. Select the **Stop Condition** tab.

19. This SLA completes when the Incident is placed in a Resolved state. In the condition, select the following: **Field = State**, **Operator = is**, and **Value = Resolved.** **ProTip 1!** (See Appendix A)

20. The Stop Condition tab should look like the example.



21. Click **Update**.

Lab Goal

Lab 1.2 Create a Schedule for SLAs

This lab explains how to define a Schedule in the system that is used in SLAs. Subject areas include the following:

- Create a Schedule
- Create a Schedule Entry
- Associate a Schedule with an SLA

The goal of this lab is to create a Schedule that allows the SLA to run only during business hours, which are 9am to 5pm.

Create a Schedule and Associate to an SLA

1. The SLA with the VIP users only applies during business hours (9am-5pm). Create a schedule to associate to the SLA definition.

2. Navigate to **System Scheduler > Schedules > Schedules**.

3. From the list, click **New** to create a new schedule.

4. In the Schedule form update the fields as follows:

   Name: **9 to 5 Weekdays**
   Description: **This schedule defines the business hours for the K17 lab**

5. The Schedule record should look like the example.

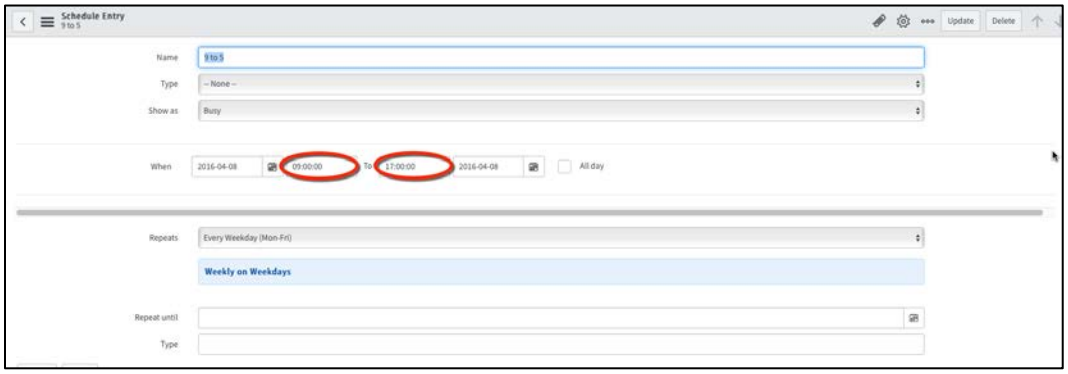


6. Right-click on form banner and select **Save**. **ProTip 2!**

7. Next, define this schedule on the related list called **Schedule Entries**. Select **New**.

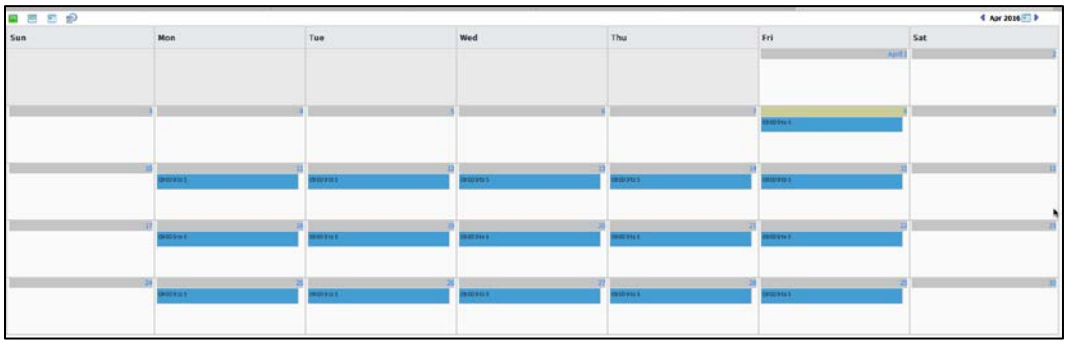
8. Fill out the Schedule Entry record as follows:

   Name: **9 to 5**
   When: **09:00:00 and 17:00:00 (see screenshot below)**
   Repeats: **Every Weekday (Mon-Fri)**



9. Select **Submit** to save this Scheduled Entry.

10. On the Schedule records, select the **Related Link > Show Schedule** to open a new tab that shows a calendar view of the schedule just created.



11. Close the tab.

12. Navigate back to the SLA Definition to associate the newly created schedule: **Service Level Management > SLA > SLA Definitions**.

13. Select **P1 Incident for VIP Users**.

14. On the SLA Definition record from the **Schedule Source** field select **SLA Definition**.

15. Next from the **Schedule** field select **9 to 5 Weekdays**.

16. Select **Update**.

Lab Goal

Lab 1.3
SLA in Action

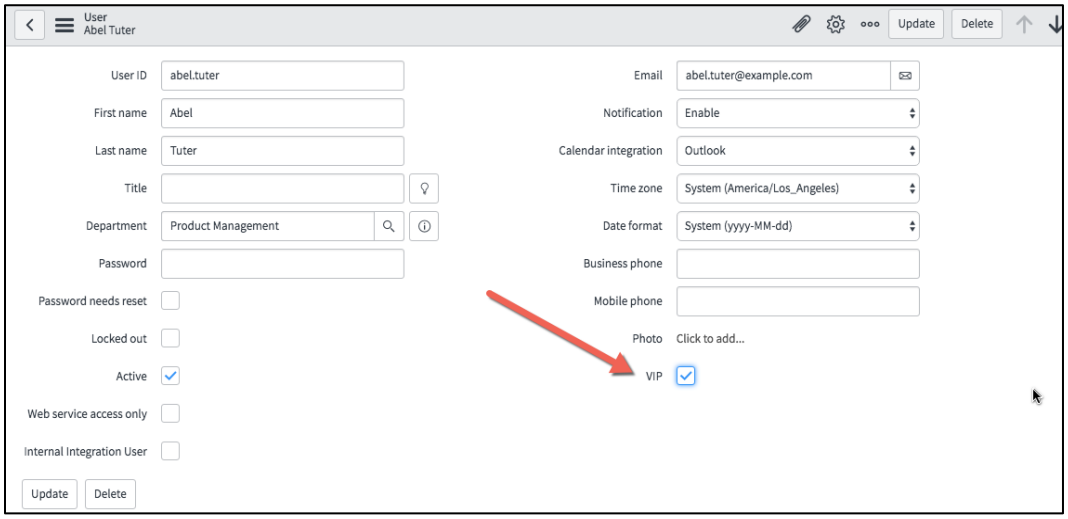
This lab shows how SLAs are used in Incident Management. Subject areas include the following:

- Observe SLA being triggered
- Observe SLA Pause
- Observe SLA Complete

The goal of this lab is to observe the SLA just created running against an incident. This serves as a test of the work completed in Lab 1.1.

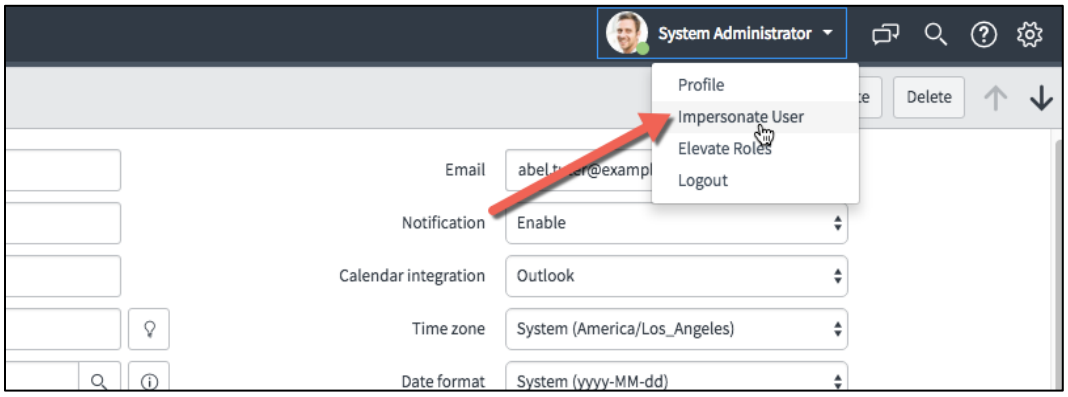
Update a User to a VIP

1. Navigate to **User Administration > Users**.
2. Search for a user with the Last name **Tuter**. Click the **abel.tuter** User ID.
3. Check the **VIP** box in Abel's user record. Click **Update**.



4. Next, impersonate Abel Tuter to create an Incident as a VIP.

5. Click **System Administrator** in the upper-right corner. Click **Impersonate User**.

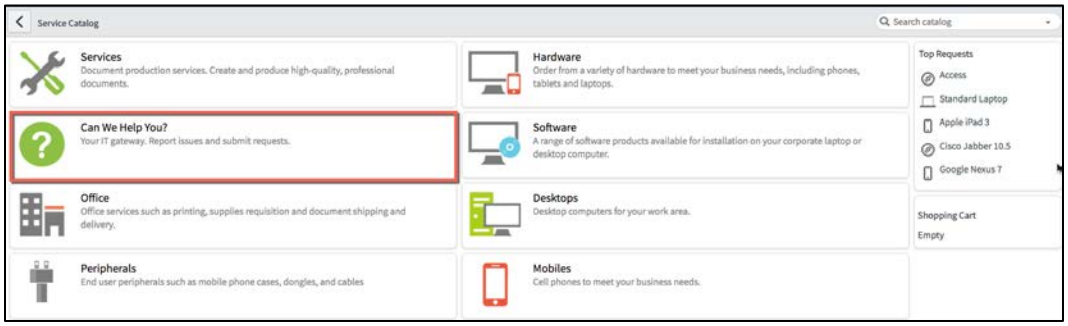


6. Search for **abel** and select **Abel Tuter**.

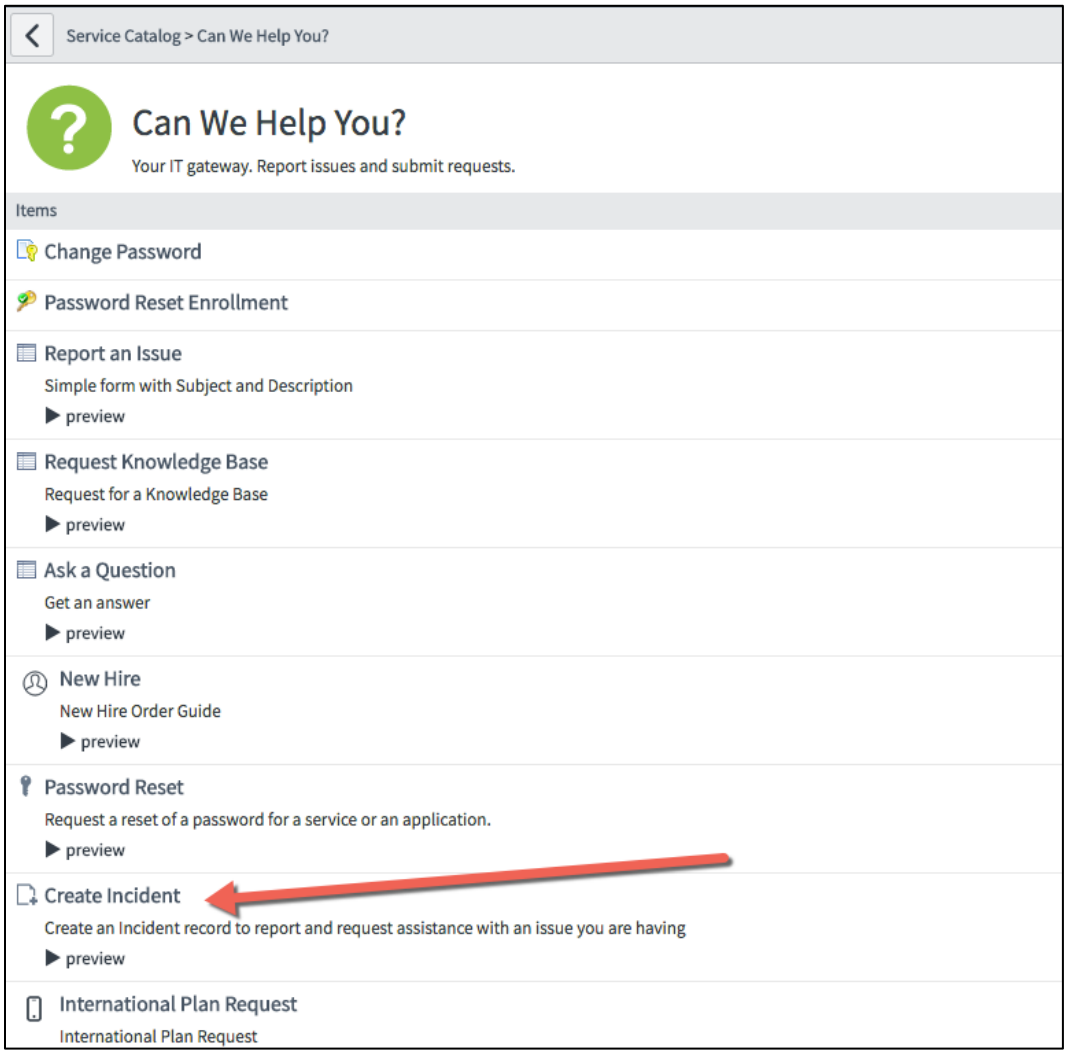
Create an Incident

1. Navigate to **Self Service > Service Catalog**.

2. Select the **Can We Help You?** Catalog.



3. Select **Create Incident**.



4. Enter data as follows:

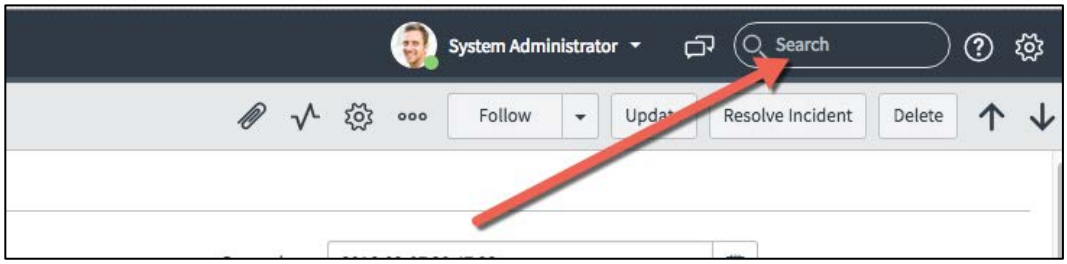
   Urgency**:** leave as **2-Medium**
   Describe your issue: K1**7 Incident**

5. Click **Submit**.

6. Make a note of the **Incident number**.

Observe in Progress, Paused, and Completed SLA on an Incident

1. Impersonate **Admin** to update the Incident status.
2. Click **Abel Tuter** in the upper-right corner. Click on **Impersonate User**.
3. Select **System Administrator** from the Recent Impersonations list.
4. Click the Search magnify glass in the upper-right corner.



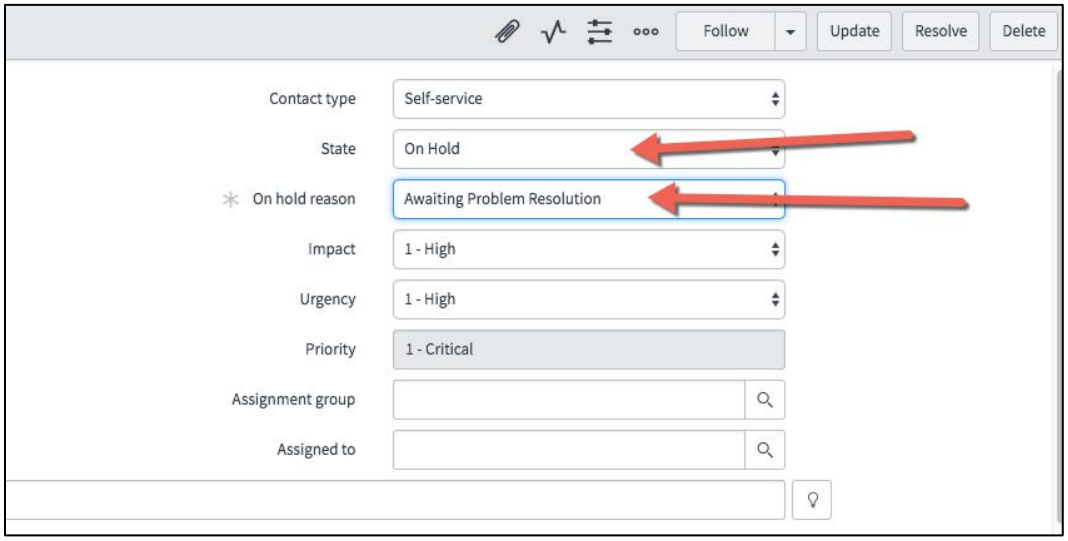
5. Enter the **Incident number** from **Step 6** and press **Enter**.
6. Scroll to the Task SLAs related list at the bottom of the Incident record and observe no SLA is attached, because the start conditions are not met.



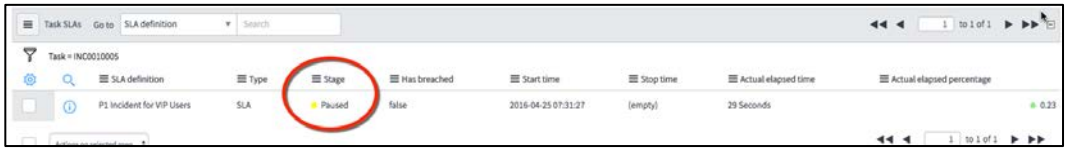
7. Update the Impact to **1 – High** and Urgency to **1 – High.** Right-click the Incident header and select **Save**.
8. Scroll to the Task SLAs related list and **observe** the P1 for VIP Users SLA created earlier is attached and **In progress**.



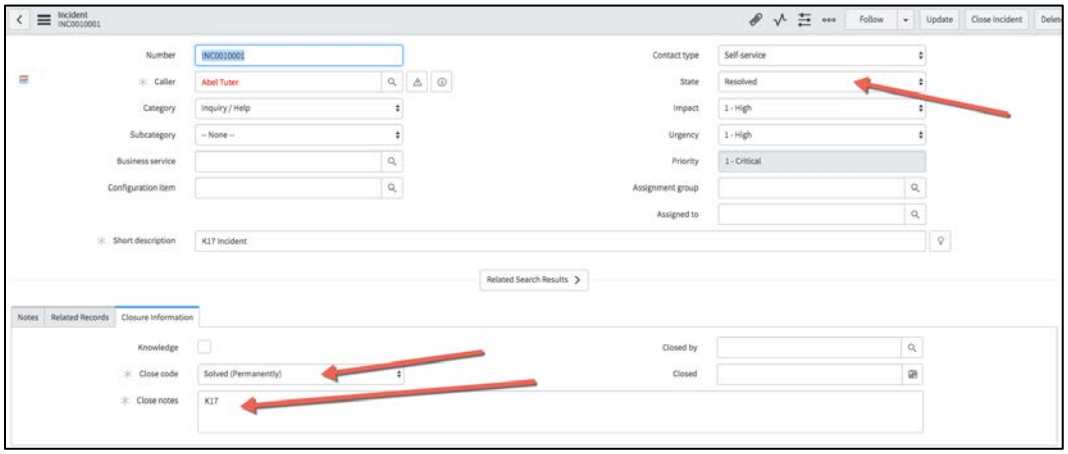
9. Update the State of the Incident to **On Hold**. In the **On hold reason** field, select **Awaiting Problem Resolution**. Right-click the Incident header and select **Save**. **ProTip 3!**



10. Scroll to the Task SLAs related list and **observe** the P1 for VIP Users SLA created earlier is now **Paused**.



11. Update the State of the Incident to **Resolved**, navigate to the Closure Information tab and select close code of **Solved (Permanently)** type **K17** in the Close Notes field. Right-click the Incident header and select **Save**.



12. Scroll to the Task SLAs related list and **observe** the P1 for VIP Users SLA created earlier is now **Completed**.



Lab Goal

Lab 2.0
Set Up an OLA

This lab explains how to define an Operational Level Agreement (OLA) based on an internal agreement. Subject areas include the following:

- Learn how schedules and workflows are used to define OLAs
- Configure the conditions in which the OLA should run
- Create a P1 incident with VIP users to show how OLAs run against records

The goal of this lab is to create an OLA that states all Priority 1 Problem records need to be updated daily. The OLA ensures that the most critical problem records are being worked regularly.

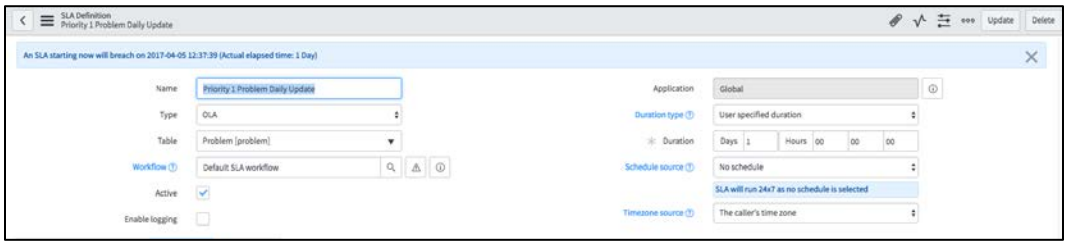
Create an OLA Definition

1. Navigate to **Service Level Management > SLA Definitions**.

2. From the list, click **New** to create a new SLA definition.

3. Update the from fields as follows:

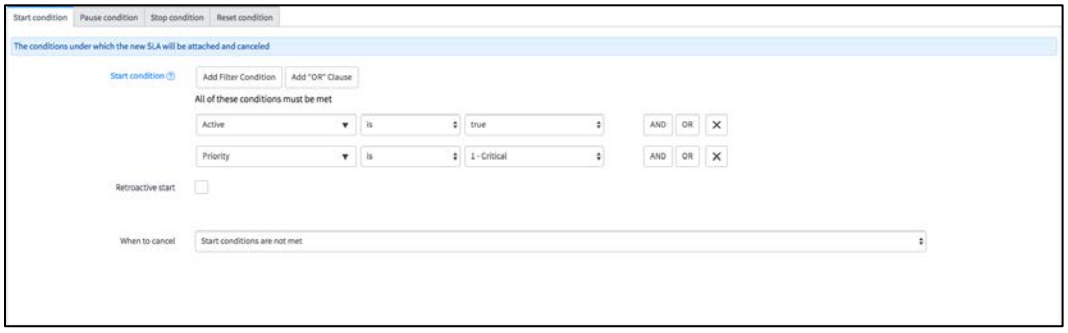
   Name: **Priority 1 Problem Daily Update**
   Type: **OLA**
   Table: **Problem**
   Duration: **1 Day**
   Schedule Source: **No Schedule**



4. Right-click on form banner and select **Save**.

Create OLA Start, Stop, and Reset Conditions

1. On the **Start Condition** tab build out the following condition to start the OLA for any Problem record that is a **Priority 1**.
2. In the first condition statement select **Field = Active**, **Operator = Is**, and **Value = true**.
3. Select **AND** to add another condition.
4. In the second condition statement select **Field = Priority**, **Operator = Is**, and **Value = 1 – Critical**.
5. Set the **When to cancel** field to **Start conditions are not met.**
6. The tab should look like the example.



7. Next, configure the Stop Condition. Select the **Stop Condition** tab.
8. This OLA completes when the Problem is placed in any state other than open. In the condition, select the following: **Field = State**, **Operator = is not**, and **Value = Open**.
9. The Stop Condition tab looks like the example.



10. Next, configure the Reset Condition. Select the **Reset Condition** tab.

11. This OLA completes when the Problem is placed in any state other than open; however, you are tracking to make sure the problem is updated every day. Reset conditions allow for an SLA/OLA to be marked as complete and generate a new SLA/OLA based on the reset conditions. In the reset condition, select the following: **Field = Updated**, **Operator = on**, and **Value = Today**. The Reset Condition tab should look like the example. **ProTip 4!**



12. Right-click on form banner and select **Save**.

Lab Goal

Lab 2.1 Create a Workflow

This lab explains how to create a workflow that is associated with an SLA or OLA. Subject areas include the following:

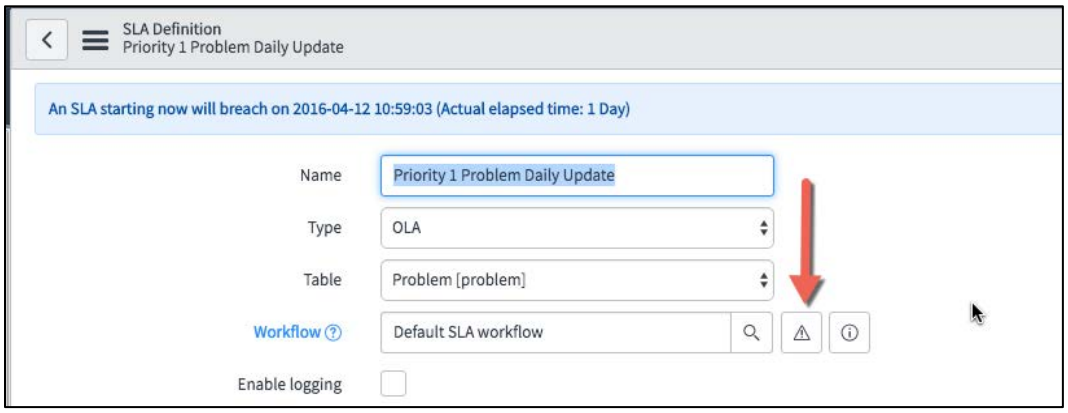
- Learn how to create a workflow
- Configure the workflow
- Associate the workflow with an OLA or SLA

The goal of this lab is to create a workflow that provides regular reminders to problem management staff about updates that are due on Priority 1 Problem records.

Create a Workflow

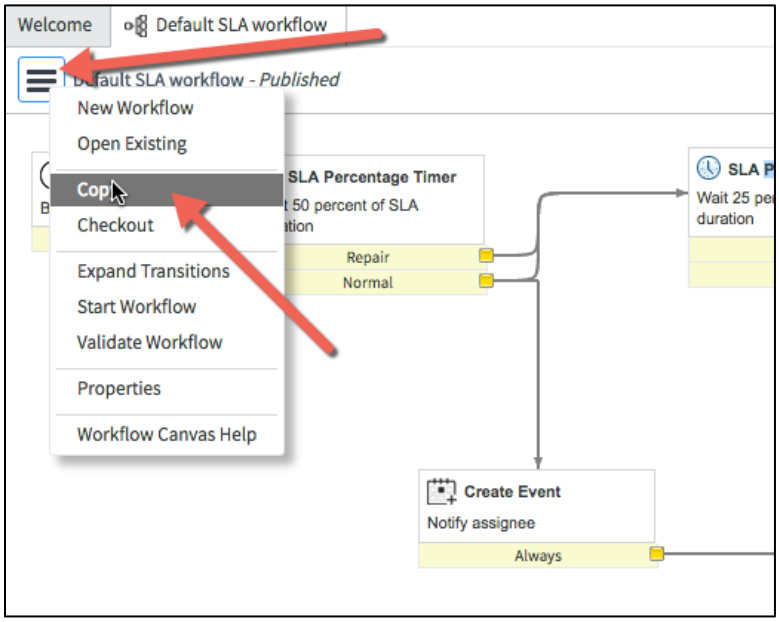
1. Navigate to the OLA record **Priority 1 Problem Daily Update** and open it.

2. Currently the OLA has the Default SLA workflow associated. Click the ⚠ icon next to the workflow as shown below to show the current workflow in a new tab.



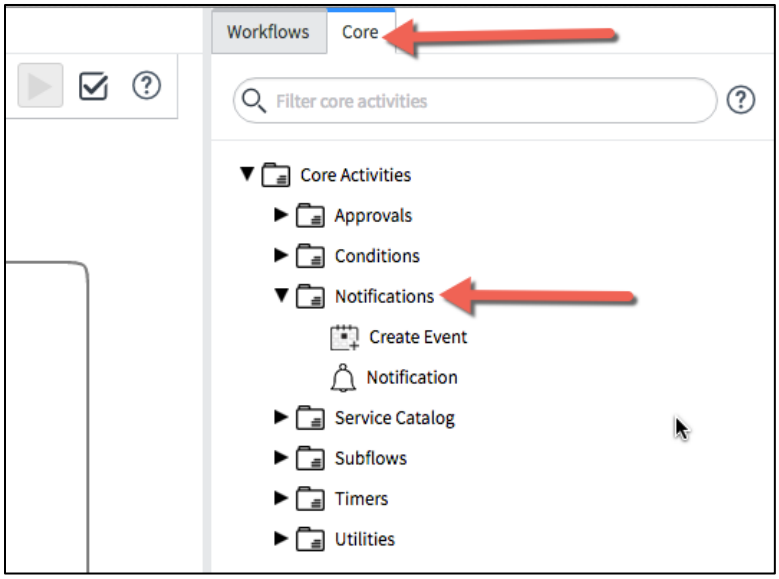
3. Observe the Default SLA Workflow. It is configured to send 3 notifications: the first after 50% of the SLA has elapsed, the second after 75%, and the third after 100% has elapsed. As a reminder to the Problem Management team you want to send a notification right away to remind them about daily updates.

4. Instead of creating a workflow from scratch, copy and modify the Default SLA Workflow. In the workflow editor select the hamburger menu ☰, from the list select **Copy** as shown. **ProTip 5!**

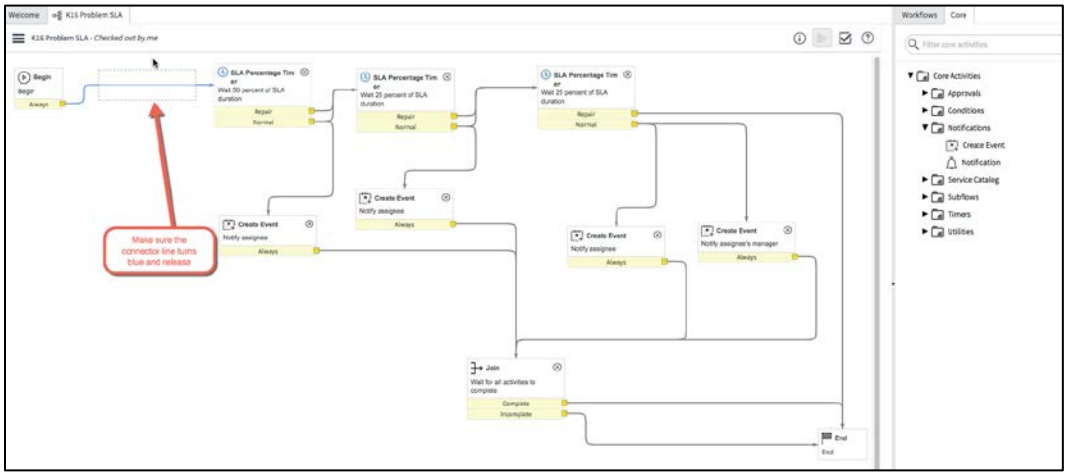


5. In the **Workflow Name** field, type **K17 Problem OLA** and select **OK**.

6. A copy of the workflow is created and checked out. Next, add a new notification to the workflow. On the left navigation pane of the workflow editor, select the **Core Tab,** which lists all the core workflow activities that are available and select **Notifications** to show the list of available timers as shown in the example.



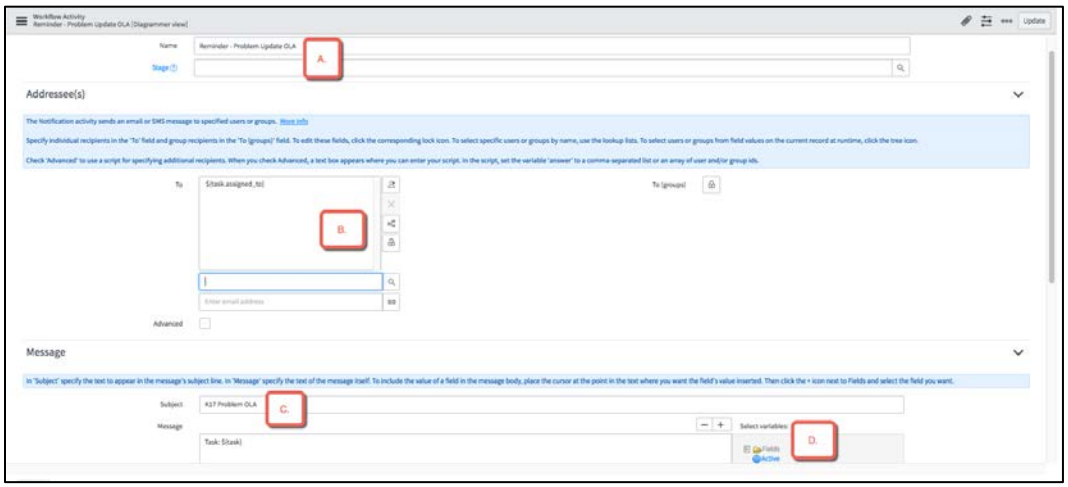
7. Next, drag a **Notification** on the workflow as shown in the example.



8. An activity record now opens a pop-up. Complete the form as follows:

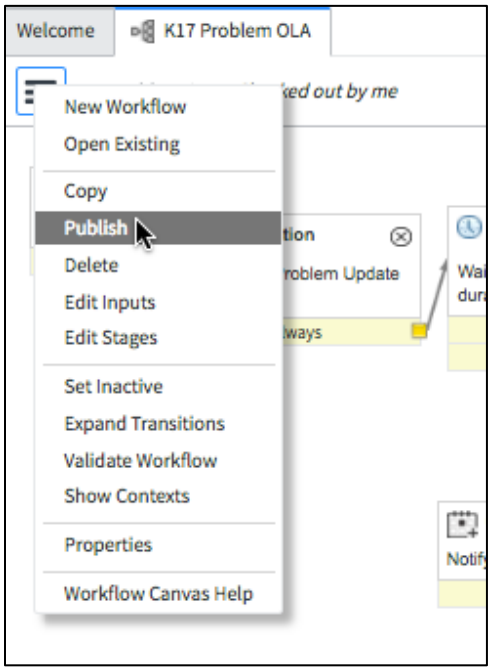
   Name: = **Reminder - Problem Update OLA**

   To: = **Click on lock** to unlock the field, then on the **Branch Icon**  . Then select the plus sign (**+**) next to Task, and select **Assigned To**
   Subject: = **K17 Problem OLA**
   Message: Select from variables as shown in screenshot. Select **Task**.



9. Select **Submit**.

10. Next**, publish** the workflow: from the h**amburger** menu ☰ select **Publish**.



11. Close the workflow tab.

12. Navigate to the new OLA: **Service Level Management > SLA Definition > Priority 1 Problem Daily Update**.

13. In the SLA definition, update the **Workflow** field to the newly created workflow **K17 Problem OLA**.



14. Click **Update** to update the SLA Definition Record.

Lab Goal

Lab 2.2
OLA in Action

This lab shows how OLAs are used in Problem Management. Subject areas include the following:

- Observe OLA being triggered
- Observe OLA Reset
- Observe OLA Workflow
- Observe OLA Completion

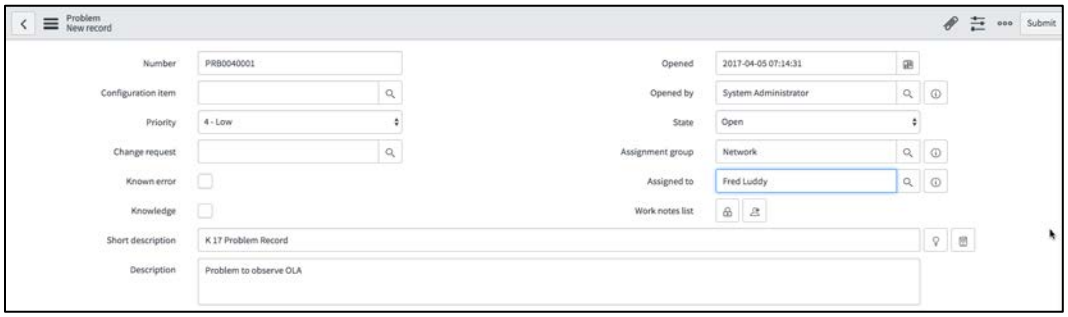
The goal of this lab is to observe the new OLA running against a problem record to test the work completed in Labs 2.0 and 2.1.

Observe OLAs

1. To observe the new OLA, create a Problem Record. Navigate to **Problem > Create New**.

2. On the Problem Record complete the following fields:

   Short Description: **K17 Problem Record**
   Description: **Problem to observe OLA**
   Assignment Group: **Network**
   Assigned to: **Fred Luddy**



3. Right-click on form banner and select **Save**.

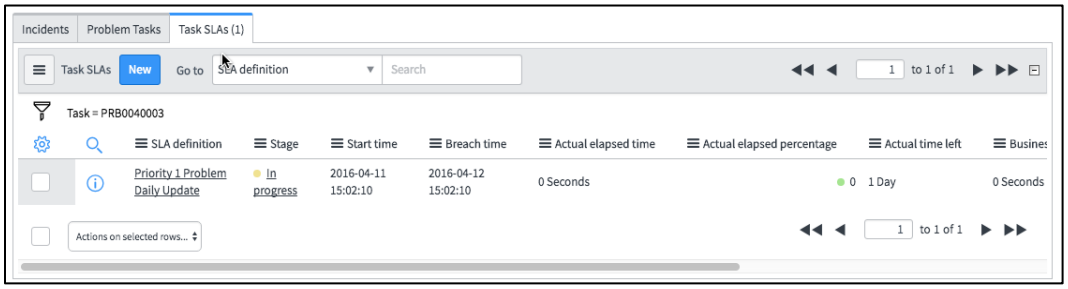
4. Scroll to the bottom of the record and **observe** no SLA is running, since this is not a Priority.



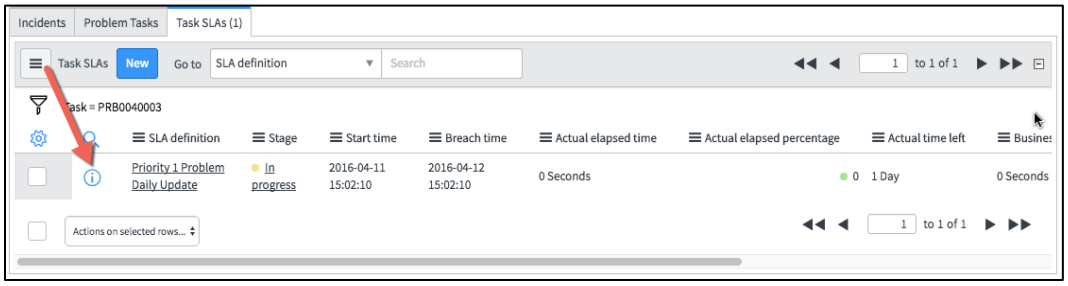
5. The OLA applies to Priority 1 Problem Records. Change the Priority from **4-Low** to **1-Critical**.

6. Right-click on form banner and select **Save**.

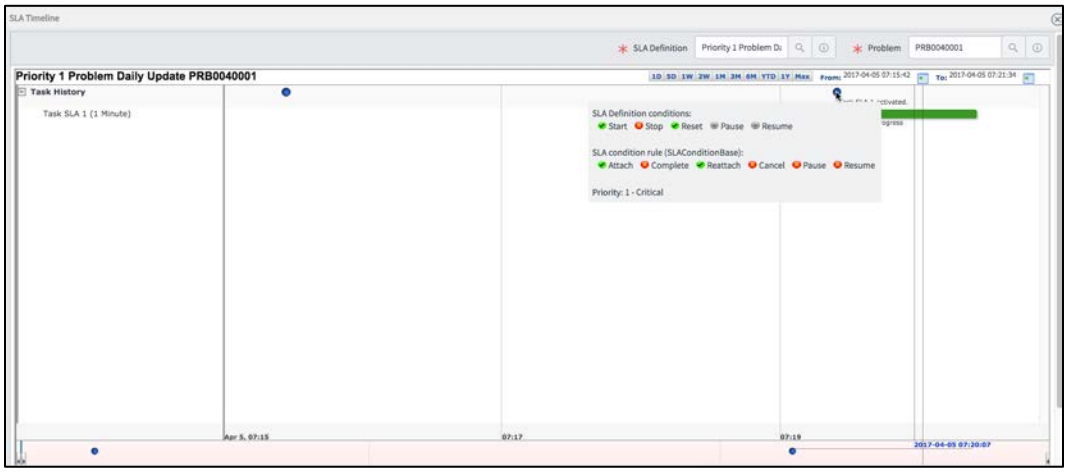
7. Scroll to the bottom the record and **observe** that an SLA is running.



8. Next, observe the new OLA running: click the  next to the OLA Record running to open the Task SLA Record.



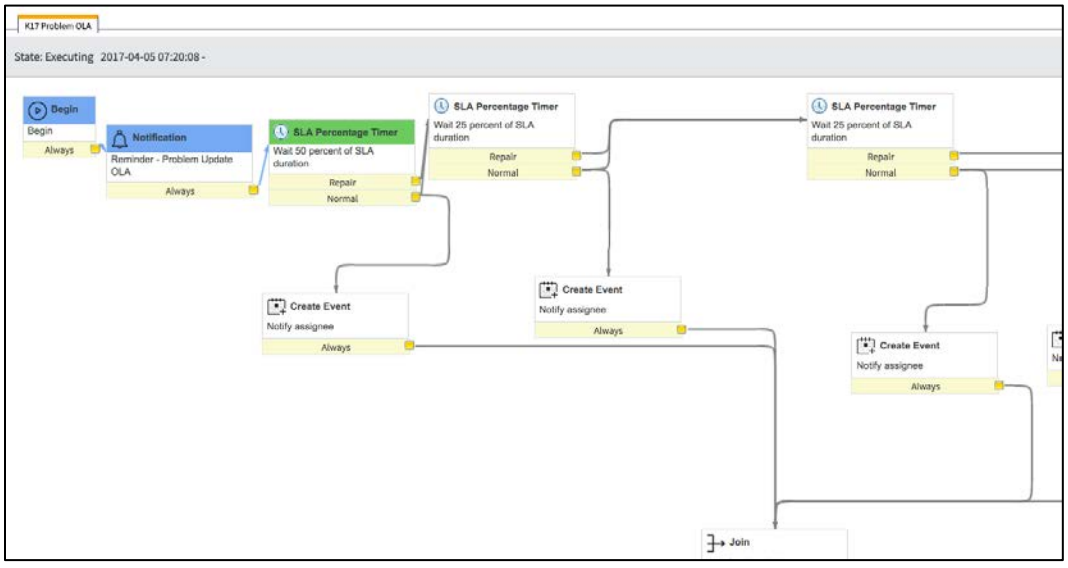
9. On the Task SLA form under related links select **Show SLA Timeline**.



10. Hover over the time and observe the OLA data provided. This view can be very helpful in debugging and understanding how the OLA behaves.

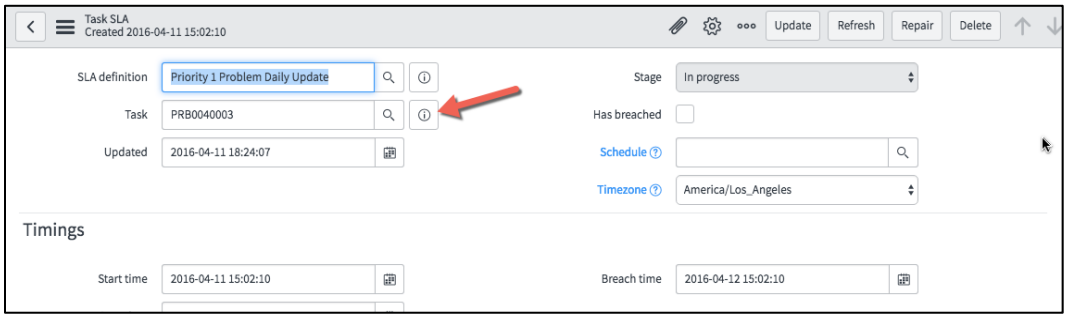
11. Close the pop-up window.

12. Next select **Show Workflow** under the related links section.



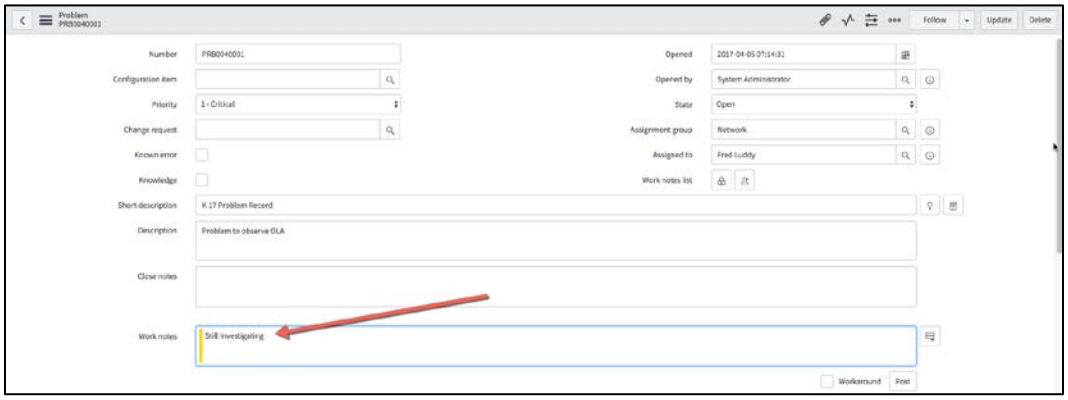
13. This opens a new tab that shows the workflow running against this Task SLA record. Observe the workflow and note that it matches the new one.

14. Close the workflow tab, and navigate back to the Problem Record as shown in the example.



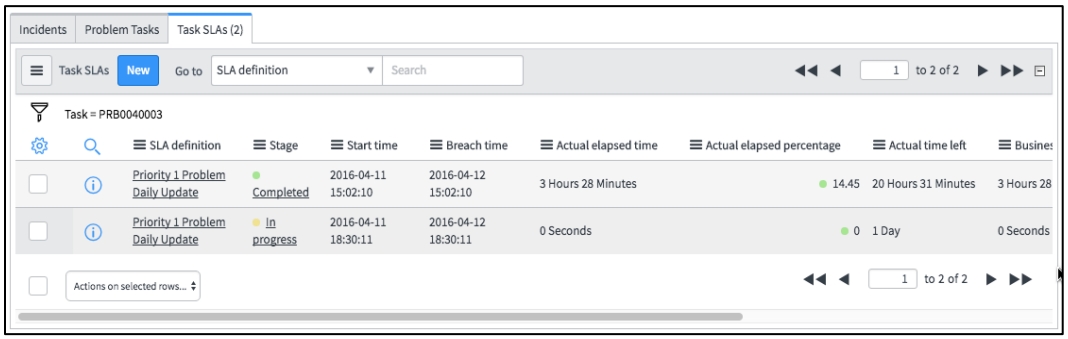
15. Observe the OLA reset. Any update made to the Problem Record completes the running SLA and creates a new one.

16. On the Problem Record in the **Work Notes** field, add the note **Still Investigating**.



17. Right-click the form banner and select **Save**.

18. Scroll to the bottom of the Problem Record. Observe the old OLA is completed and a new one is created.



19. Close the Problem Record by changing the **State = Closed/Resolved**. Right-click the form banner and select **Save**.

20. Scroll to the bottom of the Problem Record and observe both OLAs are now completed.

21. Finally, navigate to **System Mailboxes > Outbound > Outbox**

22. Observe the email from the workflow that was created in the lab to send an email to the assignee in this case Fred Luddy.



Lab Goal

Lab 3.0
Refine SLAs – Retroactive Start

This lab shows how to refine SLAs using the following features:

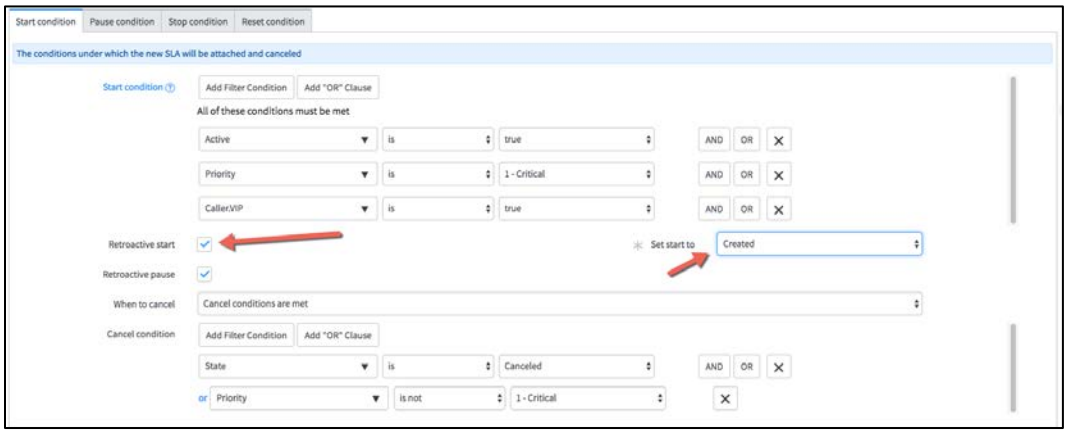
- Set Retroactive start times
- Set Retroactive pause times
- Observe Retroactive start in action

The goal of this lab is to take add additional detail to the SLAs created in the first two labs to account for specific business needs.

The ability to retroactively set a start time allows for proper accounting for time elapsed if the priority on an incident changes.

Update an SLA to Use Retroactive Start Times

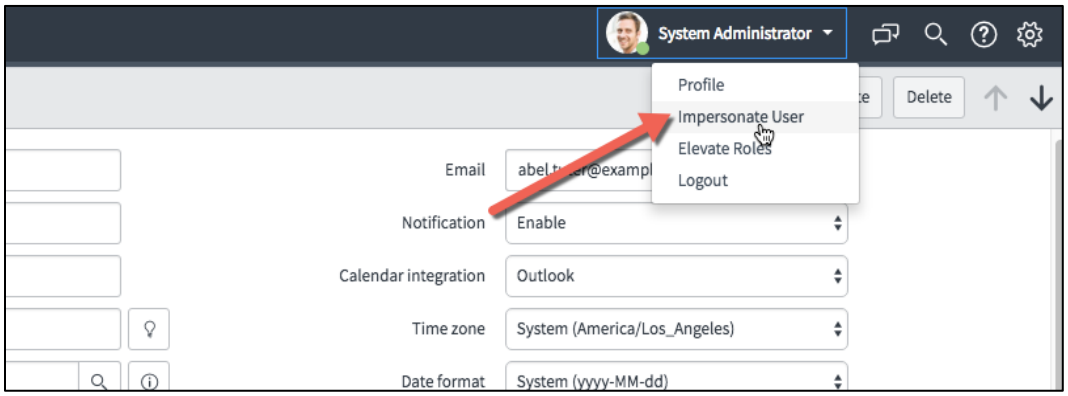
1. Navigate to **Service Level Management > SLA > SLA Definitions**.
2. Select **P1 Incident for VIP Users**.
3. On the SLA Definition record, change the **Start Condition** to **Retroactive Start** by checking the check box, which also checks **Retroactive Pause**.
4. Select the **Set start to** parameter to **Created**.



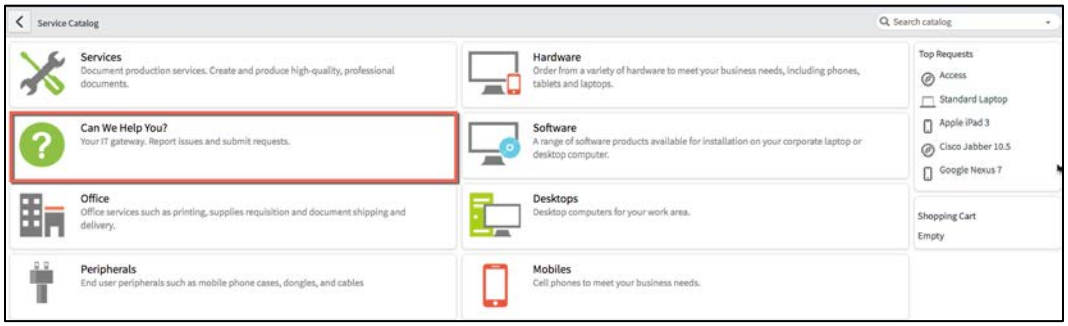
5. Select **Update**.

Create an Incident

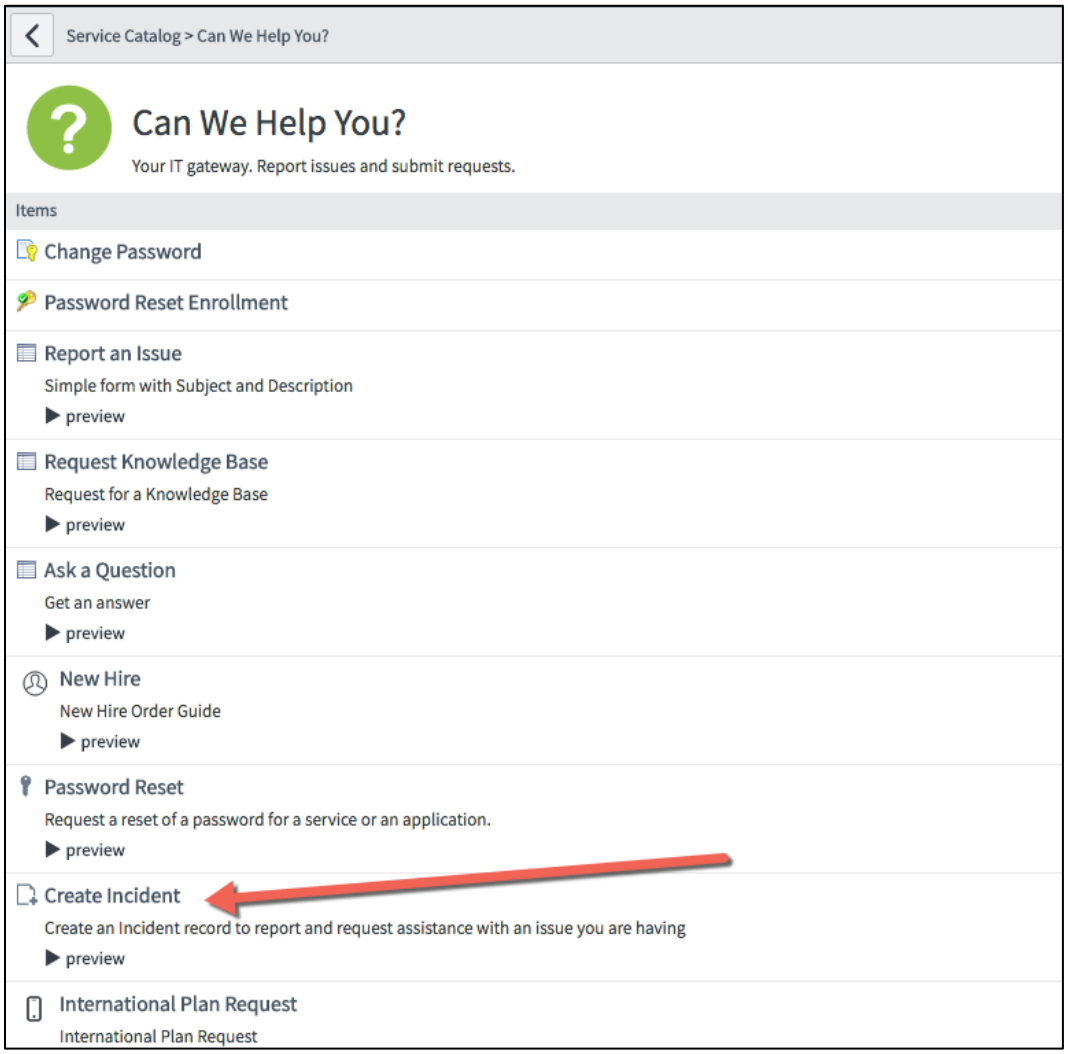
1. Impersonate Abel Tuter to create an Incident as a VIP.
2. Click **System Administrator** in the upper-right corner. Click **Impersonate User**.



3. Search for **abel** and select **Abel Tuter**.
4. Navigate to **Self Service > Service Catalog**.
5. Select the **Can We Help You?** Catalog.



6. Select **Create Incident**.



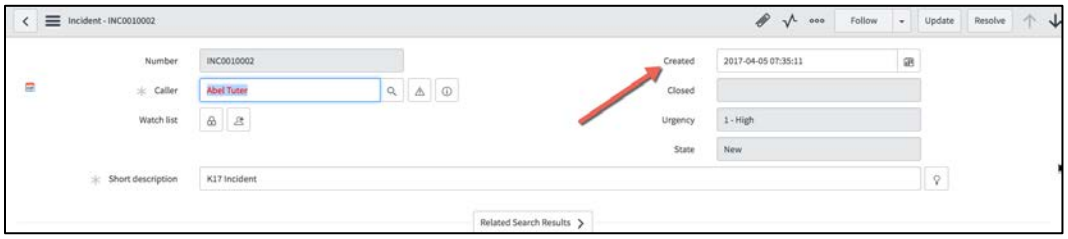
7. Enter data as follows:

   Urgency**:** leave as **2-Medium**
   Describe your issue: K1**7 Incident**

8. Click **Submit**.

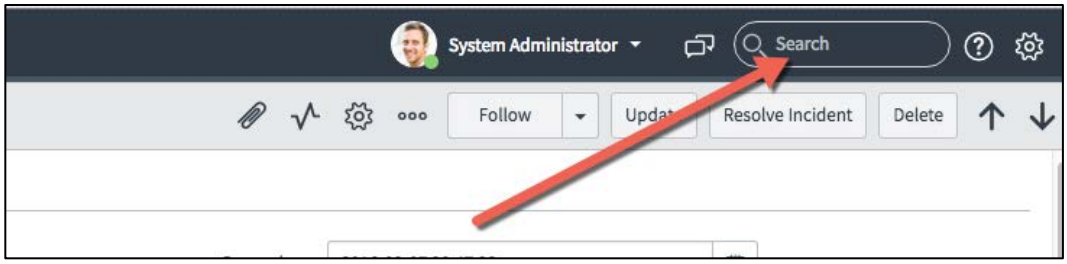
9. Make a note of the **Incident number**.

10. Also, make a note of the **Created** date/time.



Observe Retroactive Start SLA on an Incident

1. Impersonate **Admin** to update the Incident status.
2. Click **Abel Tuter** in the upper-right corner. Click on **Impersonate User**.
3. Select **System Administrator** from the Recent Impersonations list.
4. Click the Search magnify glass in the upper-right corner.



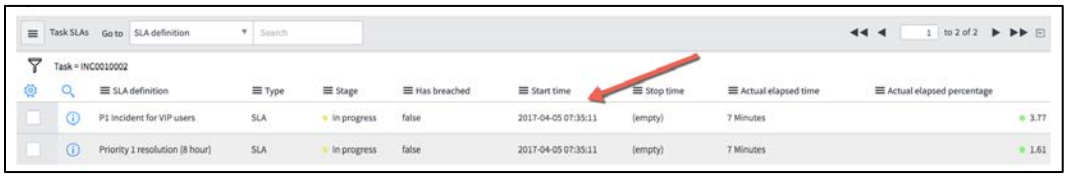
5. Enter the **Incident number** from **Step 9** and press **Enter**.
6. Scroll to the Task SLAs related list at the bottom of the Incident record and observe no SLA is attached, because the start conditions are not met.



7. Update the Impact to **1 – High** and Urgency to **1 – High.** Right-click the Incident header and select **Save**.

8. Scroll to the Task SLAs related list and observe the P1 for VIP Users SLA created earlier is attached and **In progress**.

9. Note also that the Start Time is the time the incident was created, not the current time.



Lab Goal

Lab 3.1 Using Time Zones

This lab shows how to refine SLAs using the following features:

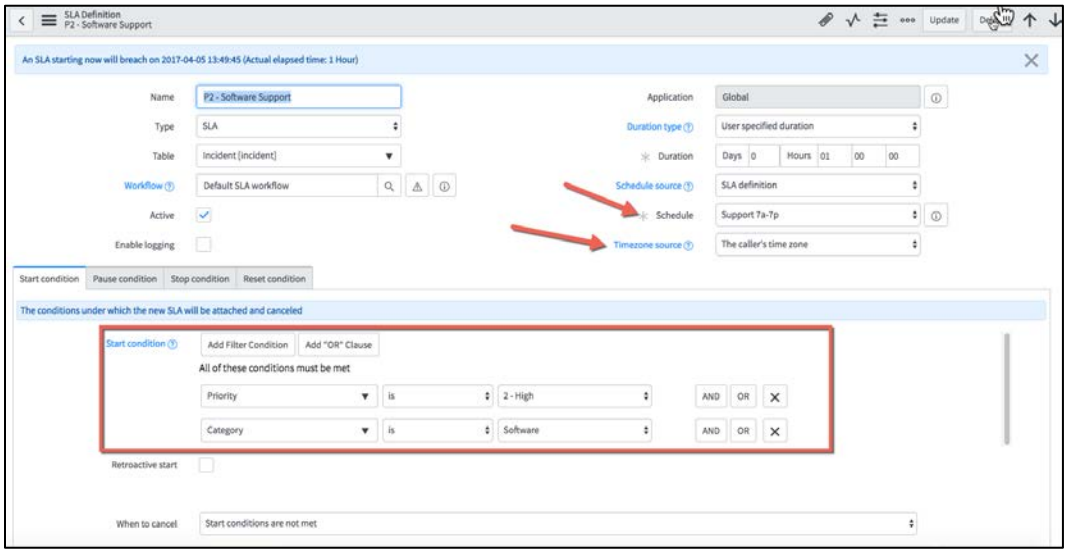
- Setting SLA based on caller's time zone
- Observe time zone based SLA

The goal of this lab is to take add additional detail to the SLAs created in the first two labs to account for specific business needs.

An SLA can be associated specifically to the time zone of either the Caller or of the impacted Configuration Item, depending on how the support organization agrees to engage.

Associate SLAs to Time Zones

1. Navigate to **Service Level Management > SLA > SLA Definitions**.
2. Select **P2 – Software Support**.
3. **Observe** the SLA Definition record, the **Schedule** field is set to a schedule of 7am to 7pm, the **Timezone source** field is set to **the caller's time zone**. Also, the **Start Conditions** are set to Priority = **2** and Category = **Software**. **ProTip 6!**



4. Using the caller's time zone source calculates the business duration of the SLA specific to that caller's time zone. To see this in action, create two incidents: one for a caller in New York and another for a caller in Hong Kong. Then you can see how this setting works.

5. Navigate to **Incident > Create New**

6. On the Incident Record complete the following fields:

   Caller: **Joe New York**
   Category: **Software**
   Short description: **NY Incident**
   Impact: **1 - High**
   Urgency: **2 - Medium**

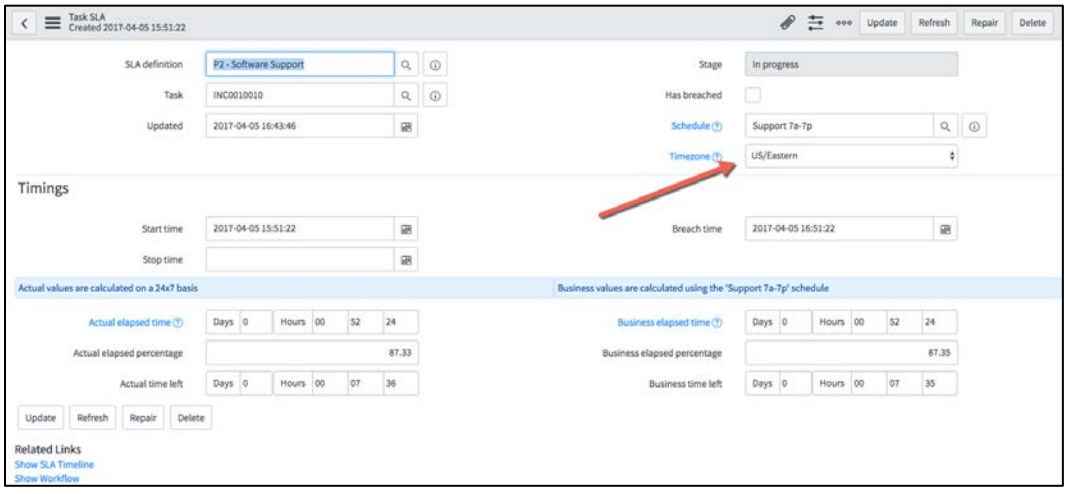
7. Right-click the form banner and select **Save**.

8. Scroll to the Task SLAs related list and observe the P2 – Software Support the **Breach time** is one hour from now local time.



9. Look at the Task SLA: click the ⓘ next to the SLA Record running to open the Task SLA Record.

10. Observe the **Timezone** field is set to US/Eastern.



11. Next, create an Incident for a caller from Hong Kong.

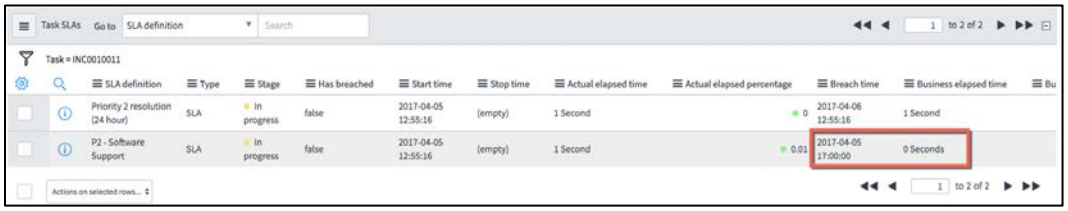
12. Navigate to **Incident > Create New**.

13. On the Incident Record complete the following fields:

    Caller: **Joe Hong Kong**
    Category: **Software**
    Short description: **HK Incident**
    Impact: **1 - High**
    Urgency: **2 - Medium**

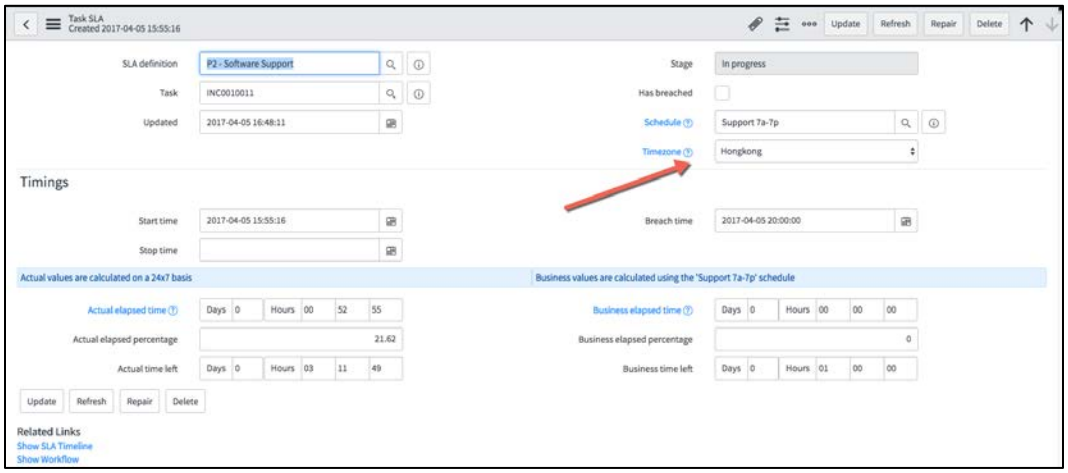
14. Right-click the form banner and select **Save**.

15. Scroll to the Task SLAs related list and observe that the P2 – Software Support the **Breach time** is now set for one hour from 7am Hong Kong time and business elapsed time is zero because the SLA is not yet in the schedule.



16. Look at the Task SLA: click the ⓘ next to the SLA Record running to open the Task SLA Record.

17. Observe the **Timezone** is set to **HongKong**.



18. Note the use of US-Pacific Time. This is because the Java Application Server FLX is in San Diego.

Lab Goal

Lab 3.2
Dynamic SLA Schedule

This lab shows how to refine SLAs using the following features:

- Set SLA based on task field schedule attribute
- Observe dynamic schedule based SLA

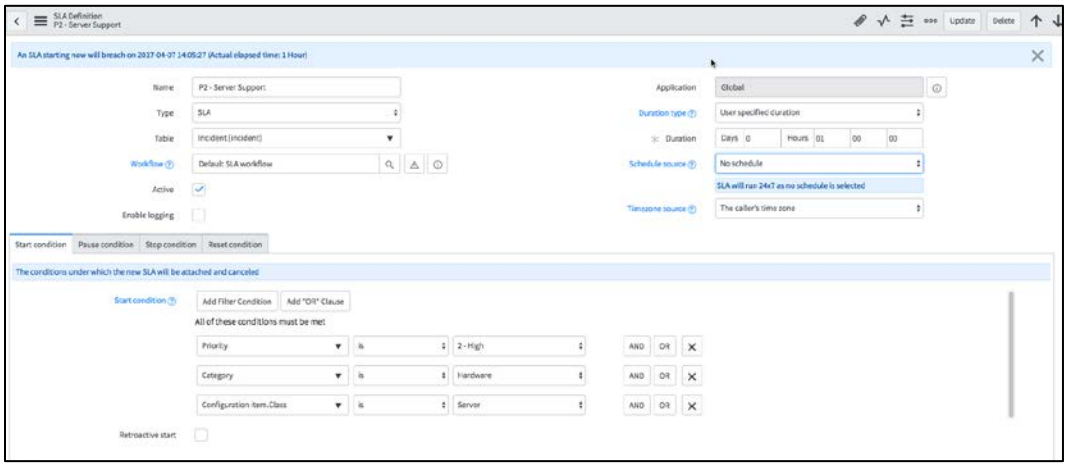
The goal of this lab is to take add additional detail to SLAs to account for specific business needs. Setting a dynamic schedule for SLAs leveraging data from the associated task can eliminate the need to create multiple SLA definitions. In this lab, you create a dynamic SLA that looks at the CI schedule to determine SLA response objectives.

Set SLA Schedule Using Attributes from the Associated Task

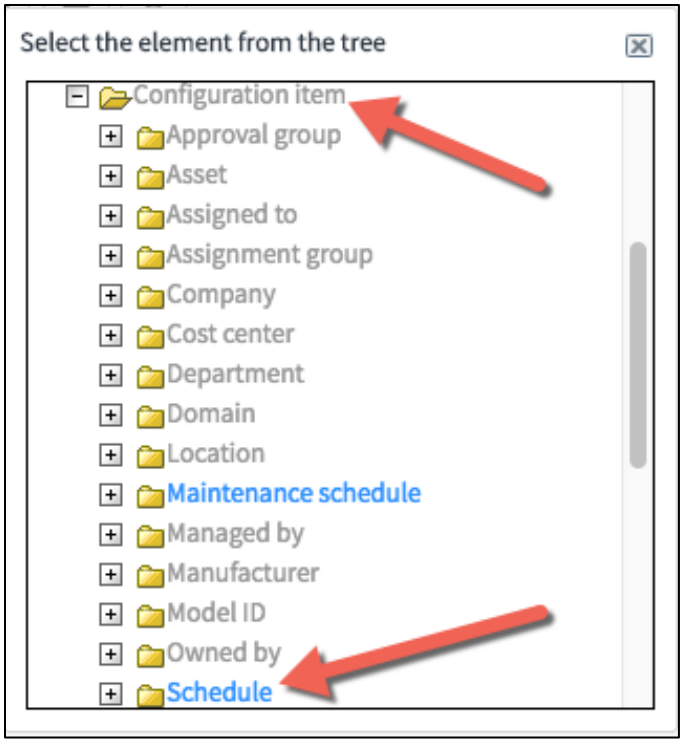
1. Navigate to **Service Level Management > SLA > SLA Definitions**.

2. Select **P2 – Server Support**.

3. Observe the SLA Definition record, the **Schedule** field is set to **No Schedule**, the **Timezone source** field is set to **the caller's time zone**. Also the **Start Conditions** are set to **Priority = 2**, **Category = Hardware** and **Configuration Item.Class = Server**. **ProTip 7!**



4. You want SLAs for Server support to only run during support hours for that specific CI. In the **Schedule Source** field. select **incident** field, then on **Schedule Source** field select **Configuration Item.Schedule** as shown below (it is likely already populated as such, but open the control to see how the list works).

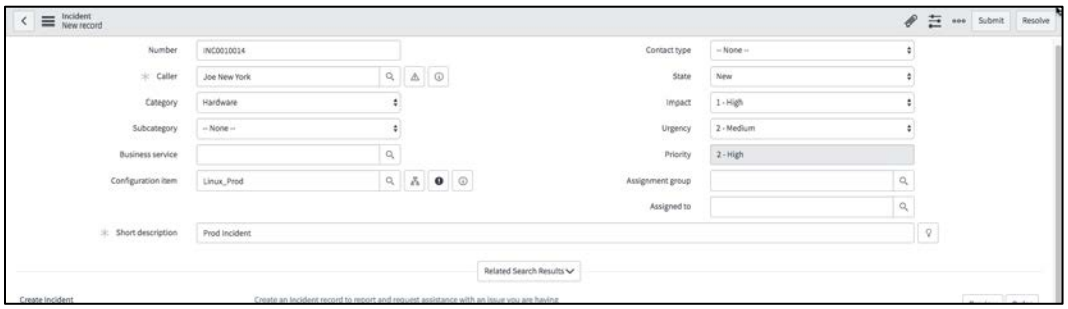


5. Click **Update** once complete to save the record.

6. Next, create two incidents: one for a Production Server that is supported 24/7 and the second for a Development Server that is supported 7am-7pm. Navigate to **Incident > Create New.**

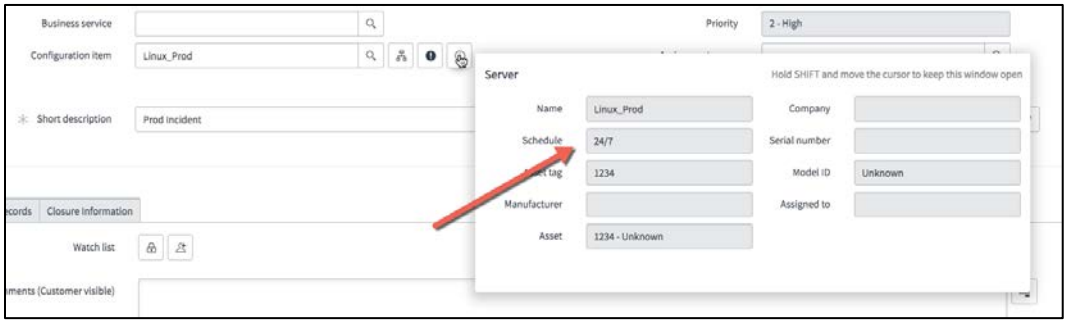
7. On the Incident Record complete the following fields:

   Caller: **Joe New York**
   Category: **Hardware**
   Short description: **Prod Incident**
   Impact: **1 - High**
   Urgency: **2 – Medium**
   Configuration Item: **Linux_Prod**



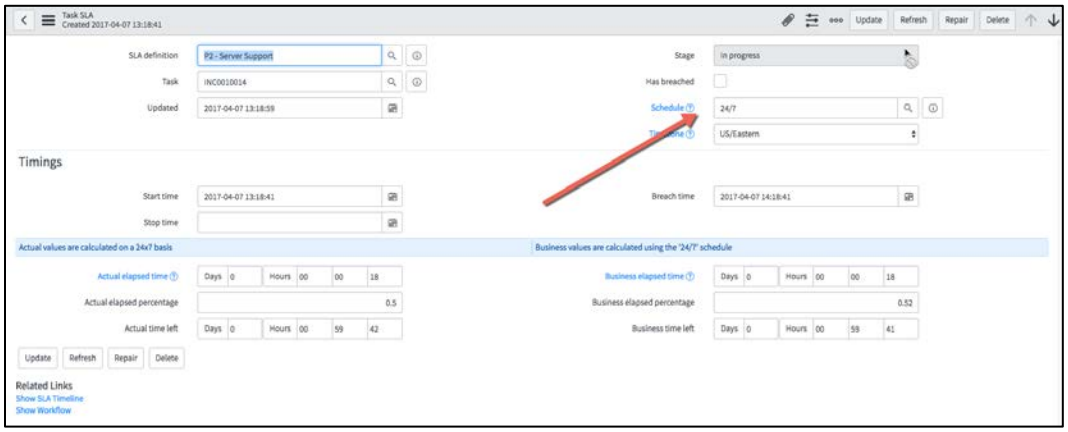
8. Right-click the form banner and select **Save**.

9. On the **Configuration Item** hover over the info button ⓘ and observe the CIs schedule is 24/7.



10. Scroll to the bottom of the form and in the Task SLA related list click the ⓘ next to the SLA Record running the P2 – Server Support SLA.

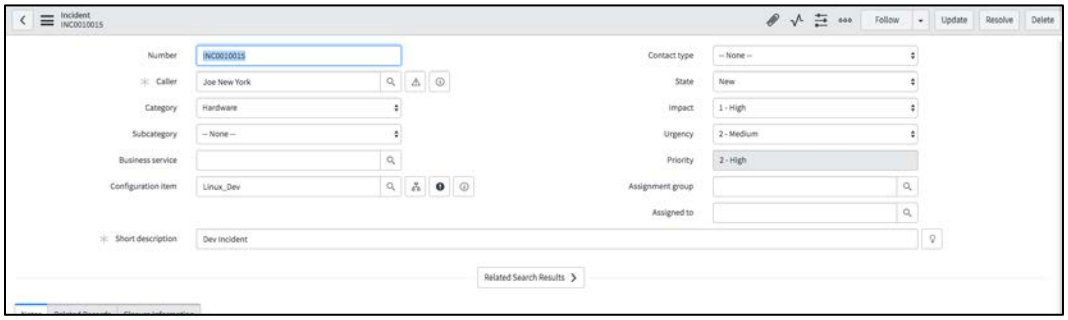
11. Observe the Schedule of this task SLA is **24/7**, which is derived from the CIs Schedule.



12. Next, create a second incident for a Development Server that is supported 7am-7pm. Navigate to **Incident > Create New.**

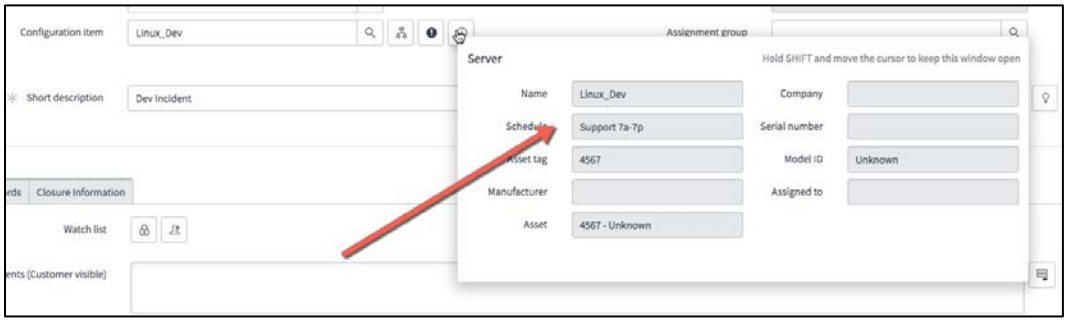
13. On the Incident Record complete the following fields:

    Caller: **Joe New York**
    Category: **Hardware**
    Short description: **Dev Incident**
    Impact: **1 - High**
    Urgency: **2 – Medium**
    Configuration Item: **Linux_Dev**



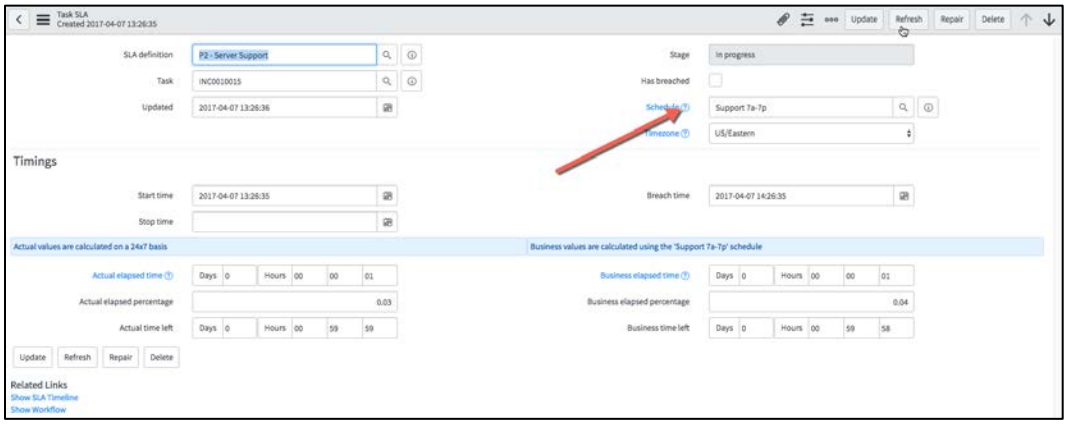
14. Right-click the form banner and select **Save**.

15. On the **Configuration Item** hover over the info button and observe the CIs schedule is **Support 7a-7p**.



16. Scroll to the bottom of the form and in the Task SLA related list click the next to the SLA Record running the P2 – Server Support SLA.

17. Observe the Schedule of this task SLA is **Support 7a-7p**, which is derived from the CIs Schedule.



Recommendation: SLAs are the most reported-on activity in the IT Service Management space. It is recommended that you learn more about reporting and creating dashboards to round out your experience with SLAs and Service Level Management.

Pro Tips

We included some Pro Tips along the way in the lab to extend your learning and challenge you to ask some questions when configuring SLAs.

Appendix A
Pro Tips

1. Make sure your Stop conditions (and any others, for that matter) match your process. For example, we had you stop the SLA when the State goes to Resolved. But what if your process allows the state to be moved directly to Closed (as you might for First Touch Resolution, when you gain agreement from the caller as to the completeness of the solution)? This condition would not account for that and you will have wild SLAs roaming free. You will see examples throughout this lab that don't follow the "closed loop" practice here. If you can point them out to the gurus, there might be some loot in it for you.
2. You'll notice we don't have a Type selected for this Schedule. Don't be like us. Define schedule types! If you don't, you'll have schedules popping up in places where they're not even used, like On-Call schedules showing up in your SLAs. Define a set of types as you start extending functionality, then use Reference Qualifiers on forms to present only the type of schedule that's appropriate for the form you're in.
3. The most commonly-abused, SLA-impacting status is On-Hold. Having processes around those items placed On-Hold is key. You might even consider an on-hold SLA workflow that requires notifications or automatic removal from On-Hold if certain conditions are met. At the very least, someone should monitor the use of the status (reporting) and follow up on those that are sitting for a significant amount of time or if something about it doesn't sound right.
4. Is this really a practice you want to implement? If you start forcing people to update records daily, all you're going to get is a never-ending list of Work Notes in the activity for the record of things like "blah" or "asdlkj". Is there value in that? What might be a better way to ensure that certain items get the daily attention they need? A "nag" notification might help, but we all know the first thing someone will do is set up an email filter to file those things away and never look at them. Maybe a daily task generation in the workflow might help? Even though this update requirement is far from best practice, it doesn't change the fact that this exact task is very common among our customers and prospective customers, so we've shown it to you anyway.
5. We recommend you always copy the stock, Out-of-the-Cloud workflows and edit the copy instead of just editing the OOTC flows. That way, if there's a mistake made or if you decide that you didn't really need to complicate your workflow as much as you did, you have the OOTC workflow to fall back on.

6. SLAs are meant to enforce good customer service levels. Using the caller's time zone makes a lot of sense when the issue is truly a customer-related one. It doesn't make as much sense when you're talking about, say, a piece of hardware in a data center that could be impacting multiple users across multiple time zones. Take care in defining these only for truly local situations (like PC support activities).
7. Make sure you use the right operator here, especially in the context of CI classes. If you are using subclasses of Server, you'll want to use **is a** instead of **is** to pull in the subclasses. Make sure you test with multiple server types to ensure that your CMDB data model matches the filter as you'd expect.
8. Bonus Tip! SLAs can get interesting because they can overlap, either intentionally or accidentally. Two things can help you here: Naming Convention and Testing. Make sure that you establish a naming convention early (including summary of conditions and the table/record type it applies for). That way, you can check for overlap without reading each SLA and manage accordingly. And unit testing as you're setting these up (in DEV, for Pete's sake!) is going to be vital to establish when the conditions stack up as expected.