ScienceLogic

Microsoft Teams SyncPack

Version 1.0.0

Table of Contents

Chapter

1

Introduction to the Microsoft Teams SyncPack

Overview

This chapter describes how you can use the "Microsoft Teams" SyncPack to automatically create channels and trigger event messages from SL1 to Microsoft Teams. The integration is uni-directional between SL1 and Microsoft Teams.

This SyncPack uses the "Microsoft Teams Automation" PowerPack

This chapter covers the following topics:

What Can I Do with this SyncPack?

The "Microsoft Teams" SyncPack lets you automate the creation of channels and messages in Microsoft Teams based on SL1 events. The integration is bidirectional, between SL1 and Microsoft Teams.

You can configure this SyncPack to perform the following integrations:

- Create a channel in Microsoft Teams to post event notifications from SL1.
- Synchronize an event update made in Microsoft Teams with the event status in SL1.
- Synchronize other event activities and status changes between teams and SL1.

For more information about how to configure these applications, see ***Configuring Applications for the Microsoft Teams SyncPack***.

Contents of the SyncPack

This section lists the contents of the "Microsoft Teams" SyncPack.

PowerFlow Applications

- ***Create Channel For Event***. This application creates a channel in Microsoft Teams to post event notifications from SL1.
- ***Created Channel Event Notification MS Teams***.
- ***Major Event Notification To MS Teams***.
- ***Outage Event Notification***.
- ***Sync Event Status From Teams to SL1***. This application synchronizes an event update made in Microsoft Teams with the event status in SL1.

For more information about how to configure these applications, see ***Configuring Applications for the Microsoft Teams SyncPack***.

Configuration Object

- ***Sample Microsoft Teams Config***. This configuration object can be used as a template after the SyncPack is installed on the PowerFlow system. The configuration object includes the following:
    - Details for connecting to the SL1 API, including the URL, username, and password
    - Details for connecting to the Microsoft Teams API, including the base URL, private instance URL, and API keys
    - Details for connecting to an HTTP Proxy when communicating with Microsoft Teams, including the hostname or IP address, username, and password

- Mapping between SL1 events and Microsoft Teams alerts including, device ID format, heartbeat and schedule name, default user, and maintenance object format information

Steps

The following steps are included in this SyncPack:

- Call User on Teams
- Create Teams Channel
- Create SL1 Payload
- Delete Event From SL1
- Fetch Channel ID From Name
- Fetch Microsoft Email
- Fetch Teams User Input Data
- Post Event Card To Teams
- Post Event Details To Channel
- Post Update to SL1

Chapter 2

Installing the Microsoft Teams SyncPack

Overview

This chapter describes how to install the "Microsoft Teams" SyncPack, and how to install the "Microsoft Teams Automation" PowerPack.

This chapter covers the following topics:

Prerequisites for the SyncPack

This SyncPack requires the following:

- The latest "Microsoft Teams Automation" PowerPack
- The latest "SL1 Notifications" SyncPack
- Administrator access to the SL1 Administration Portal
- Administrator access to Microsoft Teams
- A secured SSL/TLS-enabled connection to PowerFlow

For the latest System Requirements, see the ***SL1 PowerFlow SyncPack Release Notes***.

The following table lists the port access required by PowerFlow and this SyncPack:

Source IP	PowerFlow Destination	PowerFlow Source Port	Destination Port	Requirement
PowerFlow	SL1 API	Any	TCP 443	SL1 API Access
PowerFlow	Microsoft Teams API	Any	TCP 443	Microsoft Teams API Access

Downloading and Installing the Microsoft Teams SyncPack

A SyncPack file has the **.whl** file extension type. You can download the SyncPack file from the ScienceLogic Support site.

Downloading the SyncPack

NOTE: If you are installing or upgrading to the latest version of this SyncPack in an offline deployment, see ***Installing or Upgrading in an Offline Environment*** to ensure you install any external dependencies.

To locate and download the SyncPack:

1. Go to the ScienceLogic Support Site at https://support.sciencelogic.com/s/.
2. Click the **[Product Downloads]** tab and select *PowerPacks & SyncPacks*.
3. In the ***Search*** field, search for the SyncPack and select it from the search results. The **Release Version** page appears.
4. On the **[Files]** tab, click the down arrow next to the SyncPack version that you want to install, and select *Show File Details*. The **Release File Details** page appears.
5. Click the **[Download File]** button to download the SyncPack.

After you download the SyncPack, you can import it to your PowerFlow system using the PowerFlow user interface.

Importing the SyncPack

To import a SyncPack in the PowerFlow user interface:

1. On the **SyncPacks** page (⚙) of the PowerFlow user interface, click **[Import SyncPack]**. The **Import SyncPack** page appears.
2. Click **[Browse]** and select the **.whl** file for the SyncPack you want to install. You can also drag and drop a **.whl** file to the **Import SyncPack** page.
3. Click **[Import]**. PowerFlow registers and uploads the SyncPack. The SyncPack is added to the **SyncPacks** page.
4. You will need to activate and install the SyncPack in PowerFlow. For more information, see the following topic.

> NOTE: You cannot edit the content package in a SyncPack published by ScienceLogic. You must make a copy of a ScienceLogic SyncPack and save your changes to the new SyncPack to prevent overwriting any information in the original SyncPack when upgrading.

Installing the SyncPack

To activate and install a SyncPack in the PowerFlow user interface:

1. On the **SyncPacks** page of the PowerFlow user interface, click the **[Actions]** button (⋮) for the SyncPack you want to install and select *Activate & Install*. The **Activate & Install SyncPack** modal appears.

   > NOTE: If you try to activate and install a SyncPack that is already activated and installed, you can choose to "force" installation across all the nodes in the PowerFlow system.

   > TIP: If you do not see the PowerPack that you want to install, click the Filter icon (≡) on the **SyncPacks** page and select *Toggle Inactive SyncPacks* to see a list of the imported PowerPacks.

2. Click **[Yes]** to confirm the activation and installation. When the SyncPack is activated, the **SyncPacks** page displays a green check mark icon (✓) for that SyncPack. If the activation or installation failed, then a red exclamation mark icon (❗) appears.
3. For more information about the activation and installation process, click the check mark icon (✓) or the exclamation mark icon (❗) in the **Activated** column for that SyncPack. For a successful installation, the "Activate & Install SyncPack" application appears, and you can view the Step Log for the steps. For a failed installation, the **Error Logs** window appears.

4. If you have other versions of the same SyncPack on your PowerFlow system, you can click the **[Actions]** button (⋮) for that SyncPack and select *Change active version* to activate a different version other than the version that is currently running.

Installing the Microsoft Teams Automation PowerPack

The "Microsoft Teams Automation" PowerPack includes run book automations and run book actions that create channels and trigger event updates between SL1 and Microsoft Teams.

NOTE: The "Microsoft Teams Automation" PowerPack requires SL1 version 11.1.0 or later. For details on upgrading SL1, see the ***SL1 Platform Release Notes***.

TIP: By default, installing a new version of a PowerPack overwrites all content from a previous version of that PowerPack that has already been installed on the target system. You can use the ***Enable Selective PowerPack Field Protection*** setting in the **Behavior Settings** page (System > Settings > Behavior) to prevent new PowerPacks from overwriting local changes for some commonly customized fields. (For more information, see the ***System Administration*** manual.)

To download and install the PowerPack:

1. Search for and download the PowerPack from the **PowerPacks** page (Product Downloads > PowerPacks & SyncPacks) at the ScienceLogic Support Site.
2. In SL1, go to the **PowerPacks** page (System > Manage > PowerPacks).
3. Click the **[Actions]** button and choose *Import PowerPack*. The **Import PowerPack** dialog box appears.
4. Click **[Browse]** and navigate to the PowerPack file from step 1.
5. Select the PowerPack file and click **[Import]**. The **PowerPack Installer** modal displays a list of the PowerPack contents.
6. Click **[Install]**. The PowerPack is added to the **PowerPacks** page.

NOTE: If you exit the **PowerPack Installer** modal without installing the imported PowerPack, the imported PowerPack will not appear in the **PowerPacks** page. However, the imported PowerPack will appear in the **Imported PowerPacks** modal. This page appears when you click the **[Actions]** menu and select *Install PowerPack*.

Chapter

3

Configuring Applications for the Microsoft Teams SyncPack

Overview

This chapter describes how to set up the run book automations in SL1 and how to configure the PowerFlow applications in the "Microsoft Teams" SyncPack

This chapter covers the following topics:

Workflow for Configuring the SyncPack

The following workflows describe how to configure Microsoft Teams, SL1, and PowerFlow to work with the "Microsoft Teams" SyncPack.

Configuring Microsoft Teams

1. See the steps in *Configuring Microsoft Teams*, below.

Configuring SL1

1. *Create a SOAP/XML credential to access PowerFlow*
2. *Edit the Microsoft Teams run book actions*
3. *Enable the Microsoft Teams run book automations*

Configuring PowerFlow

1. *Create and align a PowerFlow configuration object*
2. *Configure the Microsoft Teams PowerFlow applications*
3. *Schedule the PowerFlow applications*

Configuring Microsoft Teams

To configure Microsoft Teams to work with PowerFlow, you must register an application in Azure Active Directory and perform some configuration steps for that application before using the "Microsoft Teams" SyncPack. To configure the application:

1. Follow the steps in the Microsoft documentation to register your application in Azure Active Directory.
2. Note the "app_id(client_id)", "client_secret", and "tenant_id" of your application.
3. Grant scoped permissions to the web API, specifically the following:
   - Calls.Initiate.All
   - Call.InitiateGroupCall.All
   - Channel.Create
   - ChannelMessage.Send
   - Channel.ReadBasic.All
   - offline_access
   - openid
   - User.Read
   - User.Read.All

4. Follow these steps to authorize a user and retrieve the "refresh_token". If a refresh token is generated **before** enabling multi-factor authentication (MFA), the same token will not work once MFA is enabled. In this case you must generate a new refresh token once MFA is enabled and update it in your PowerFlow configuration.
5. Create incoming webhooks in Microsoft Teams and ensure the channel privacy setting is "standard" and not "private".
6. Note the URL of your webhook(s).
7. Create a bot here (only required fields are needed).
8. After creating your bot, click ***My bots*** from the top menu and select your bot.
9. Add Microsoft Teams as a featured channel and enable calling.
10. Create an intermediary URL:
    - In PowerFlow, navigate to the **API Keys** page (🔑).
    - Click **[Create API Key]** and copy the key once it is generated.
    - Create the following URL, inserting your PowerFlow IP address and API key where indicated, and paste it in to your PowerFlow configuration: `https://<Your PF IP Address>/api/v1/applications/sync_event_status_from_teams_to_sl1/run?PF-APIKEY=<API Key>`

Configuring SL1

The following topics cover how to set up your SL1 instance to work with the "Microsoft Teams" SyncPack.

Creating a SOAP/XML Credential to Access PowerFlow

You will need to create a SOAP/XML credential so that the action policies included in the "Microsoft Teams Automation" PowerPack can access your PowerFlow system.

The PowerPack includes the "PowerFlow Microsoft Teams" credential, which you can use as a template for the SOAP/XML credential.

To define a SOAP/XML credential using the example credential:

1. In SL1, go to the **Credentials** page (Manage > Credentials).
2. Click the wrench icon (🔧) for the "PowerFlow Microsoft Teams" credential. The **Credential Editor** modal window appears.
3. Complete the following fields:
    - ***Name***. Type a new name for the credential.
    - ***All Organizations***. Toggle on (blue) to align the credential to all organizations, or toggle off (gray) and then select one or more specific organizations from the ***What organization manages this service?*** drop-down field to align the credential with those specific organizations.
    - ***URL***. Type the URL for your PowerFlow system.

- ***HTTP Auth User***. Type the username for your PowerFlow system.
- ***HTTP Auth Password***. Type the password for your PowerFlow system.

4. Click **[Save & Close]**.
5. Take note of the SL1-assigned ID number for the new credential on the **Credentials** page, in the **ID** column. You will need the ID number when editing the input parameters of the run book actions included in the PowerPack, below.

Editing the Microsoft Teams Run Book Actions

The "Microsoft Teams Automation" PowerPack includes seven run book actions that use the "Run Integration Service Application" action type to trigger the PowerFlow applications that send and receive data to and from Microsoft Teams. You can specify the credential ID in a JSON structure that you enter in the ***Input Parameters*** field in the **Action Policy Editor** modal.

After you edit the action and trigger the event policy, the new event log will be added to the respective device on the **Event Console** page.

To edit the actions included in the PowerPack:

1. In SL1, go to the **Actions** page (Registry > Run Book > Actions).
2. Locate the automation action that you want to use, and then click its wrench icon (🔧). The **Editing Action** page appears.
3. In the ***Input Parameters*** field, change the values of the following parameters:
    - ***credential_id***. Change the value to the credential ID that you noted earlier when creating a credential for your PowerFlow system in the previous procedure. This parameter is required.
    - ***include_event***. Leave the value as "true".
    - ***application_name***. Leave the default application value.
    - ***params***. Leave the default parameter value.
4. Make sure the ***Action State*** is set to *Enabled*, and then click **[Save]**.

Enabling the Microsoft Teams Run Book Automations

The "Microsoft Teams Automation" PowerPack includes seven run book automation policies that you will need to enable.

These policies update the Microsoft Teams alert or SL1 event with the state of the associated event or alert, create a maintenance object, or populates SL1 with on-call details. When the event is first detected in SL1 or Microsoft Teams, the "triggered" state is sent. When the event is acknowledged in SL1 or Microsoft Teams, the "acknowledged" state is sent. When the event is cleared in SL1 or Microsoft Teams, the "resolved" state is sent. When the on-call policy is triggered, on-call details are added to the SL1 event notes. When a maintenance object is required, a maintenance object is created in Microsoft Teams.

The following table shows the automation policy, its aligned events, and the automation action that runs in response to the events.

Automation Policy Name	Aligned Events	Automation Action
Microsoft Teams: Create Channel	(All Events)	Microsoft Teams: Create Channel
Microsoft Teams: Create Channel Notification - Acknowledge	(All Events)	Microsoft Teams: Created Channel Notification - Acknowledge
Microsoft Teams: Create Channel Notification - Clear	(All Events)	Microsoft Teams: Create Channel Notification - Clear
Microsoft Teams: Event Acknowledged	(All Events)	Microsoft Teams: Acknowledge
Microsoft Teams: Event Cleared	(All Events)	Microsoft Teams: Clear
Microsoft Teams: Event Triggered	(All Events)	Microsoft Teams: Add/Update
Microsoft Teams: Outage Remediation	(All Events)	Microsoft Teams: Outage Notification

To enable the run book automations:

1. In SL1, go to the **Automation** page (Registry > Run Book > Automation).
2. Locate a Microsoft Teams automation policy and click its wrench icon (🔧). The **Automation Policy Editor** page appears.
3. Update the following fields:
   - ***Policy State***. Select *Enabled*.
   - ***Policy Priority***. Select *High* to ensure that this PowerFlow automation policy is added to the top of the queue.
   - ***Available Actions***. If it is not already selected, select the "Run Integration Service Application: <*name*>" action that corresponds with the Ansible automation policy you selected in step 2, and click the arrows to move it to ***Aligned Actions***.

WARNING: ScienceLogic highly recommends that you do not make changes to the ***Policy Type***, ***Repeat Time***, or ***Align With*** fields or the *And event is NOT acknowledged* setting.

4. Click **[Save]**.
5. Repeat steps 2-4 for the remaining Microsoft Teams run book automation policies.

Configuring PowerFlow

The following topics cover how to set up your PowerFlow instance to work with the "Microsoft Teams" SyncPack.

Creating a Configuration Object

A ***configuration object*** supplies the login credentials and other required information needed to execute the steps for a PowerFlow application. The **Configurations** page (⚙) of the PowerFlow user interface lists all available configuration objects for that system.

You can create as many configuration objects as you need. A PowerFlow application can only use one configuration object at a time, but you can use (or "align") the same configuration object with multiple applications.

For this SyncPack, you can make a copy of the "Sample Microsoft Teams Config" configuration object, which is the sample configuration file that was installed with the *Microsoft Teams* SyncPack.

> **TIP:** The "Sample Microsoft Teams Config" configuration object contains all of the required variables. Simply update the variables from that object to match your SL1 and Microsoft Teams settings.

To create a configuration object based on the "Sample Microsoft Teams Config" configuration object:

1. In the PowerFlow user interface, go to the **Configurations** page (⚙).
2. Click the **[Edit]** button for the "Sample Microsoft Teams Config" configuration object. The **Configuration** pane appears.
3. Click **[Copy as]**. The **Create Configuration** pane appears.
4. Complete the following fields:
    - ***Name***. Name of the configuration object that will display on the **Configurations** page.
    - ***Description***. A brief description of the configuration object.
    - ***Author***. User or organization that created the configuration object.
    - ***Version***. Version of the configuration object.
5. In the ***Configuration Data Values*** fields, update the default variable definitions to match your PowerFlow configuration:
    - ***sl1_host***. The hostname or IP address of the SL1 system the alerts will synchronize to.
    - ***sl1_user***. The username for your SL1 system.
    - ***sl1_password***. The password for your SL1 system.
    - ***refresh_token***. The refresh token you received in Azure Active Directory.
    - ***client_id***. ID of registered app in Azure Active Directory.
    - ***client_secret***. Secret code given for the corresponding app in Azure Active Directory.
    - ***tenant_id***. Tenant ID of the Azure Active Directory.
    - ***team_id***. ID of your team in Microsoft Teams.
    - ***ms_teams_webhook_url***. The URL of the webhook you created in Azure Active Directory.

- ***microsoft_user_ids***. Microsoft Teams user ID used for group calling. For a group call, enter multiple user IDs in your PowerFlow configuration separated by commas. For example:

```
{
  "encrypted": false,
  name":"microsoft_user_ids",
  "value": "dce672e6-c430-43f9-b17e-bfe07e51816d,30127c40-12b3-436b-8523-115d3b814ea9"
}
```

- ***default_ack_user***. The default user to acknowledge events and alerts.
- ***intermediary_url***. URL endpoint of the intermediary between Microsoft Teams and PowerFlow for sending user interaction data.

6. Click **[Save]**. You can now align this configuration object with one or more applications.

Aligning a Configuration Object and Configuring PowerFlow Applications

With this SyncPack, any status changes made to an SL1 event is sent to Microsoft Teams to update the corresponding alert. Any status changes to the Microsoft Teams alert are synced back to the corresponding SL1 event. You will need to align the Microsoft Teams applications with the relevant configuration object in PowerFlow, and, if needed, update any other fields on the **Configuration** pane for the applications.

To run this SyncPack, you must "align" the configuration object to run with the following PowerFlow applications:

- Create Channel For Event
- Created Channel Event Notification MS Teams
- Major Event Notification To MS Teams
- Outage Event Notification
- Sync Event Status From Teams to SL1

To align the configuration object with the relevant PowerFlow applications:

1. On the **Applications** page of the PowerFlow user interface, open one of the PowerFlow applications listed above and click **[Configure]** (⚙). The **Configurations** pane for that application appears.

2. From the ***Configurations*** drop-down, select the configuration object you want to use.

> NOTE: The values and other parameters that appear in the **Configuration** pane with a padlock icon (🔒) are populated either by the configuration object you aligned with the application or by the Run Book Action. Do not modify these values. If you encounter an error, make sure your Run Book Action is configured properly.

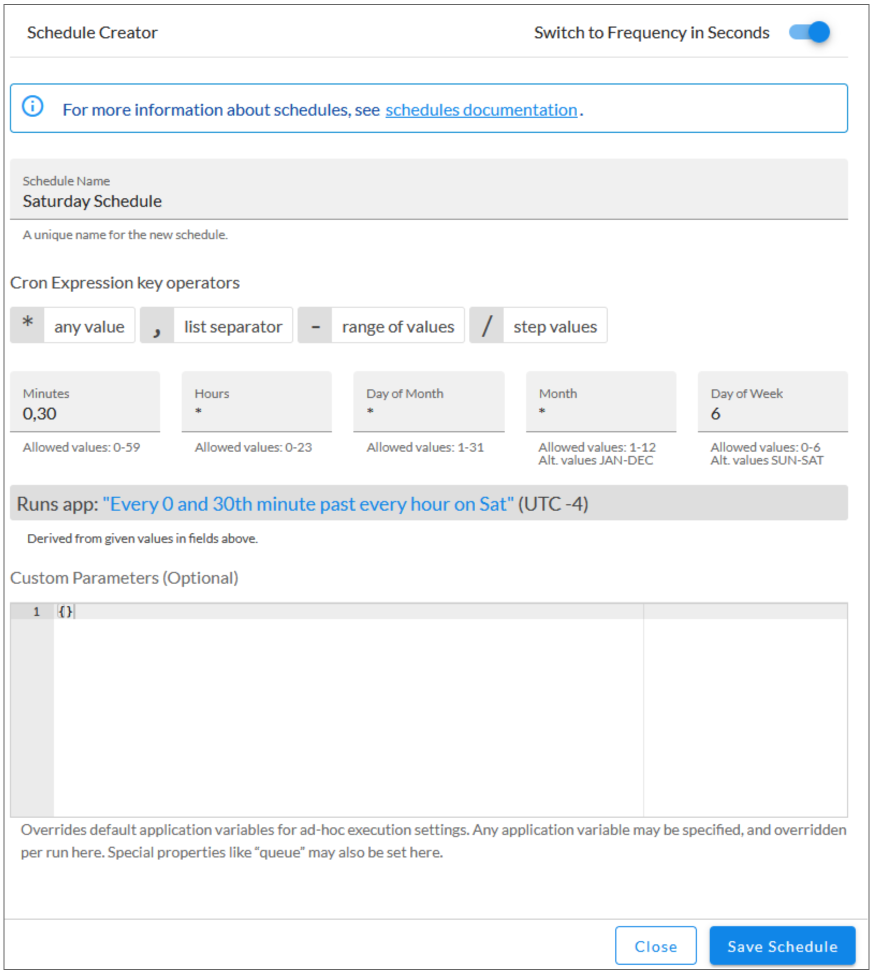
3. Click **[Save]** to align that configuration with the application.
4. Repeat this process for the other PowerFlow applications.

Scheduling PowerFlow Applications

You can create one or more schedules for a single application in the PowerFlow user interface. When creating each schedule, you can specify the queue and the configuration file for that application.

To create a schedule:

1. On the **Applications** page (☰), click the **[Schedule]** button for the application you want to schedule. The **Scheduler** window appears.
2. In the **Schedule List** pane, click the down arrow icon (⌄) next to an existing schedule to view the details for that schedule.
3. In the **Schedule Creator** pane, complete the following fields for the default ***Frequency*** setting:
    - ***Schedule Name***. Type a name for the schedule.
    - ***Frequency in seconds***. Type the number of seconds per interval that you want to run the application.
    - ***Custom Parameters***. Type any JSON parameters you want to use for this schedule, such as information about a configuration file or mappings.
4. To use a cron expression, click the ***Switch to Cron Expression*** toggle to turn it blue. If you select this option, you can create complicated schedules based on minutes, hours, the day of the month, the month, and the day of the week:


Schedule Creator

Switch to Frequency in Seconds

For more information about schedules, see schedules documentation.

Schedule Name: Saturday Schedule

A unique name for the new schedule.

Cron Expression key operators

Minutes	Hours	Day of Month	Month	Day of Week
0,30	*	*	*	6
Allowed values: 0-59	Allowed values: 0-23	Allowed values: 1-31	Allowed values: 1-12 Alt. values JAN-DEC	Allowed values: 0-6 Alt. values SUN-SAT

Runs app: "Every 0 and 30th minute past every hour on Sat" (UTC -4)

Derived from given values in fields above.

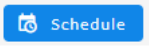
Custom Parameters (Optional)

```
{}
```

Overrides default application variables for ad-hoc execution settings. Any application variable may be specified, and overridden per run here. Special properties like "queue" may also be set here.

As you update the cron expression, the **Schedule** window displays the results of the expression in more readable language, such as *Runs app: "Every 0 and 30th minute past every hour on Sat*", based on 0,30 in the ***Minutes*** field and 6 in the ***Day of Week*** field.

5. Click **[Save Schedule]**. The schedule is added to the **Schedule List** pane. Also, on the **Applications** page, the **[Schedule]** button now displays with a dark blue background:



> **NOTE:** After you create a schedule, it continues to run until you delete it. Also, you cannot edit an existing schedule, but you can delete it and create a similar schedule if needed.

To view or delete an existing schedule:

1. On the **Applications** page, click the **[Schedule]** button for the application that contains a schedule you want to delete. The **Scheduler** window appears.

2. Click the down arrow icon (⌄) to view the details of an existing schedule.
3. To delete the selected schedule, click the Actions icon (⋮) and select**[Delete]**.

TIP: On the **Scheduler** window for a PowerFlow application, you can click the **[Copy as]** button from the **Schedule List** pane to make a copy of an existing schedule.

NOTE: When either multiple SL1 instance or multiple Microsoft Teams instances are involved with PowerFlow, you should create an individual configuration object for each SL1 or Microsoft Teams instance. Next, create an individual schedule for each configuration object. Each schedule should use a configuration object that is specific to that single SL1 or Microsoft Teams instance. Creating copies of a PowerFlow application from a SyncPack for the purpose of distinguishing between domains is not supported, and will result in issues on upgrades.

Chapter 4

Troubleshooting

Overview

This chapter contains troubleshooting resources and procedures to use with the "Microsoft Teams" SyncPack.

This chapter covers the following topics:

Initial Troubleshooting Steps

The first step taken when troubleshooting should be to ensure that there are no issues with the data platforms with which PowerFlow is interacting. There might be additional configurations or actions enabled on Microsoft Teams or SL1 that result in unexpected behavior. Perform the steps in the following sections to troubleshoot SL1 PowerFlow and Microsoft Teams. For detailed information about how to perform the steps below, see ***Resources for Troubleshooting***.

SL1 PowerFlow

1. Run `docker service ls` on the PowerFlow server.
2. Note the Docker container version and verify that the Docker services are running.
3. If a certain service is failing, make a note of the service name and version.
4. Run `docker service ps <service_name>` to see the historical state of the failing service and make a note of this information. For example: `docker service ps iservices_contentapi`
5. Make a note of any logs impacting the service by running `docker service logs <service_name>`. For example: `docker service logs iservices_couchbase`

Microsoft Teams

1. Make a note of the Microsoft Teams version and SyncPack version, if applicable.
2. Make a note of the Microsoft Teams application that is failing on PowerFlow.
3. Make a note of what step is failing in the application. Try running the application in debug mode, and capture any traceback or error messages that occur in the step log.

Resources for Troubleshooting

This section contains port information for PowerFlow and troubleshooting commands for Docker, Couchbase, and the PowerFlow API.

Useful PowerFlow Ports

- **https://<*IP of PowerFlow*>:8091**. Provides access to Couchbase, a NoSQL database for storage and data retrieval.
- **https://<*IP of PowerFlow*>:15672**. Provides access to the RabbitMQ Dashboard, which you can use to monitor the service that distributes tasks to be executed by PowerFlow workers. Use the following for login: guest guest for the login.
  - Username: guest
  - Password: guest
- **https://<*IP of PowerFlow*>/flower**. Provides access to Flower, a tool for monitoring and administrating Celery clusters.

NOTE: For version 2.0.0 and later of PowerFlow, port 5556 must be open for both PowerFlow and the client.

Helpful Docker Commands

PowerFlow is a set of services that are containerized using Docker. For more information about Docker, see the Docker tutorial.

Use the following Docker commands for troubleshooting and diagnosing issues with PowerFlow:

Viewing Container Versions and Status

To view the PowerFlow version, SSH in to your PowerFlow instance and run the following command:

```
docker service ls
```

In the results, you can see the container ID, name, mode, status (see the replicas column), and version (see the image column) for all the services that make up PowerFlow.

Restarting a Service

Run the following command to restart a single service:

```
docker service update --force <service_name>
```

Stopping all PowerFlow Services

Run the following command to stop all PowerFlow services:

```
docker stack rm iservices
```

Restarting Docker

Run the following command to restart Docker:

```
systemctl restart docker
```

NOTE: Restarting Docker does not clear the queue.

Viewing Logs for a Specific Service

You can use the Docker command line to view the logs of any currently-running service in the PowerFlow cluster. To view the logs of a service, run the following command:

```
docker service logs -f iservices_<service_name>
```

Some common examples include the following:

```
docker service logs –f iservices_couchbase
```

```
docker service logs –f iservices_steprunner
```

```
docker service logs –f iservices_contentapi
```

Application logs are stored on the central database as well as on all Docker hosts in a clustered environment. These logs are stored at `/var/log/iservices` for both single-node or clustered environments. However, the logs on each Docker host relate only to the services running on that host. For this reason, using the Docker service logs is the best way to get logs from all hosts at once.

Clearing RabbitMQ Volume

RabbitMQ is a service that distributes tasks to be executed by PowerFlow workers.

The following error message might appear if you try to run a PowerFlow application via the API:

```
Internal error occurred: Traceback (most recent call last):\n File
\"./content_api.py\", line 199, in kickoff_application\n task_status =
... line 623, in _on_close\n (class_id, method_id),
ConnectionError)\nInternalError: Connection.open: (541) INTERNAL_ERROR -
access to vhost '/' refused for user 'guest': vhost '/' is down
```

First, verify that your services are up. If there is an issue with your RabbitMQ volume, you can clear the volume with the following commands:

```
docker service rm iservices_rabbitmq
```

```
docker volume rm iservices_rabbitdb
```

If you get a message stating that the volume is in use, run the following command:

```
docker rm <id of container using volume>
```

Re-deploy PowerFlow by running the following command:

```
docker stack deploy -c /opt/iservices/scripts/docker-compose.yml
iservices
```

NOTE: Restarting Docker does not clear the queue, because the queue is persistent. However, clearing the queue with the commands above might result in data loss due to the tasks being removed from the queue.

Viewing the Process Status of All Services

Run the following command:

```
docker ps
```

Deploying Services from a Defined Docker Compose File

Run the following command:

```
docker stack deploy -c <compose-file> iservices
```

Dynamically Scaling for More Workers

Run the following command:

```
docker service scale iservices_steprunner=10
```

Completely Removing Services from Running

Run the following command:

```
docker stack rm iservices
```

Diagnostic Tools

Multiple diagnostic tools exist to assist in troubleshooting issues with the PowerFlow platform:

- ***Docker PowerPack***. This PowerPack monitors your Linux-based PowerFlow server with SSH (the PowerFlow ISO is built on top of an Oracle Linux Operating System). This PowerPack provides key performance indicators about how your PowerFlow server is performing. For more information on the Docker PowerPack and other PowerPacks that you can use to monitor PowerFlow, see the *Using SL1 to Monitor SL1 PowerFlow* chapter in the ***SL1 PowerFlow Platform*** manual.
- ***Flower***. This web interface tool can be found at the `/flower` endpoint. It provides a dashboard displaying the number of tasks in various states as well as an overview of the state of each worker. This tool shows the current number of active, processed, failed, succeeded, and retried tasks on the PowerFlow platform. This tool also shows detailed information about each of the tasks that have been executed on the platform. This data includes the UUID, the state, the arguments that were passed to it, as well as the worker and the time of execution. Flower also provides a performance chart that shows the number of tasks running on each individual worker.
- ***Debug Mode***. All applications can be run in "debug" mode via the PowerFlow API. Running applications in debug mode may slow down the platform, but they will result in much more detailed logging information that is helpful for troubleshooting issues. For more information on running applications in Debug Mode, see ***Retrieving Additional Debug Information***.

- ***Application Logs***. All applications generate a log file specific to that application. These log files can be found at `/var/log/iservices` and each log file will match the ID of the application. These log files combine all the log messages of all previous runs of an application up to a certain point. These log files roll over and will get auto-cleared after a certain point.
- ***Step Logs***. Step logs display the log output for a specific step in the application. These step logs can be accessed via the PowerFlow user interface by clicking on a step in an application and bringing up the **Step Log** tab. These step logs display just the log output for the latest run of that step.
- ***Service Logs***. Each Docker service has its own log. These can be accessed via SSH by running the following command:

```
docker service logs -f <service_name>
```

Retrieving Additional Debug Information (Debug Mode)

The logs in PowerFlow use the following **loglevel** settings, from most verbose to least verbose:

- ***10***. Debug Mode.
- ***20***. Informational.
- ***30***. Warning. This is the default settings if you do not specify a loglevel.
- ***40***. Error.

WARNING: If you run applications with "loglevel": 10, those applications will take longer to run because of increased I/O requirements. Enabling debug logging using the following process is the only recommended method. **ScienceLogic does not recommend setting "loglevel": 10 for the whole stack with the docker-compose file.**

To run an application in Debug Mode using the PowerFlow user interface:

1. Select the PowerFlow application from the **Applications** page.
2. Hover over **[Run]** from and select *Debug Run* from the pop-up menu. PowerFlow executes the application in Debug Mode with a log level of 10.

To run an application in Debug Mode using the API:

1. POST the following to the API endpoint:

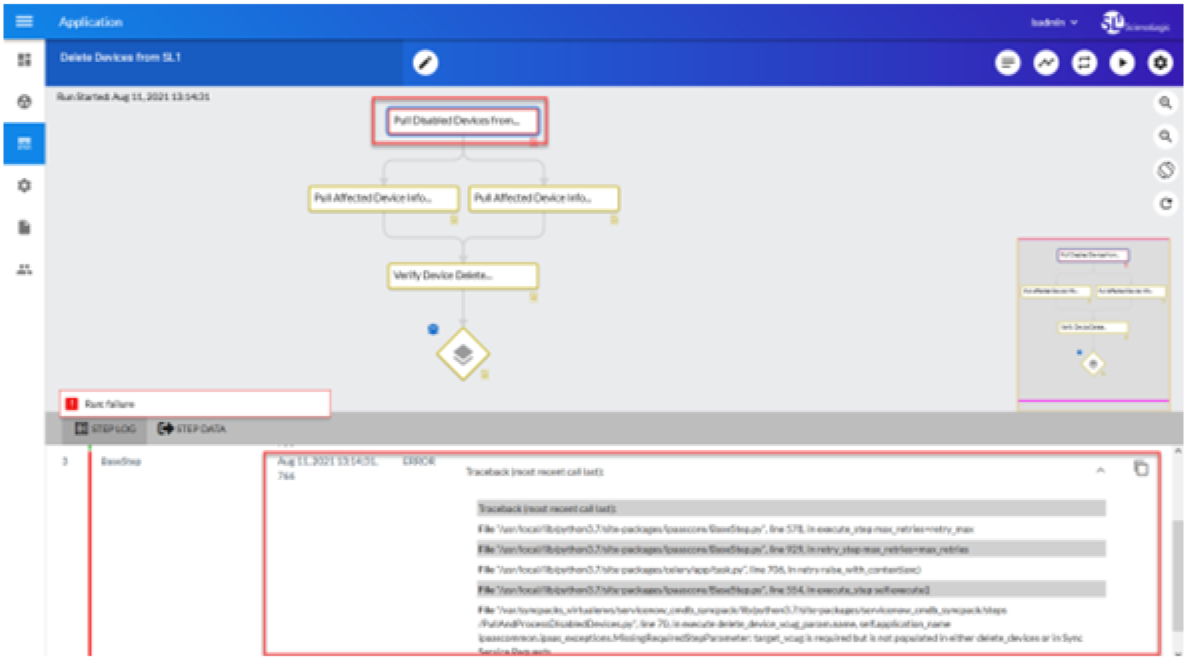
```
https://<PowerFlow>/api/v1/applications/run
```

2. Include the following in the request body:

```
{
  "name": "<application_name>",
  "params": {
    "loglevel": 10
    }
```

```
}
```

After running the application in Debug Mode, review the step logs in the PowerFlow user interface to see detailed debug output for each step in the application. This information is especially helpful when trying to understand why an application or step failed:



Run Started: Aug 11, 2021 13:14:31

```
3    BaseStep    Aug 11, 2021 13:14:31,766    ERROR    Traceback (most recent call last):

Traceback (most recent call last):
File "/usr/local/lib/python3.7/site-packages/ipaascore/BaseStep.py", line 578, in execute_step max_retries=retry_max
File "/usr/local/lib/python3.7/site-packages/ipaascore/BaseStep.py", line 929, in retry_step max_retries=max_retries
File "/usr/local/lib/python3.7/site-packages/celery/app/task.py", line 706, in retry raise_with_context(exc)
File "/usr/local/lib/python3.7/site-packages/ipaascore/BaseStep.py", line 554, in execute_step self.execute()
File "/var/syncpacks_virtualenvs/servicenow_cmdb_syncpack/lib/python3.7/site-packages/servicenow_cmdb_syncpack/steps
/PullAndProcessDisabledDevices.py", line 70, in execute delete_device_vcug_param.name, self.application_name
ipaascommon.ipaas_exceptions.MissingRequiredStepParameter: target_vcug is required but is not populated in either delete_devices or in Sync
Service Requests
```

You can also run an application in debug using curl via SSH:

1. SSH in to the PowerFlow instance.

2. Run the following command:

```
curl -v -k -u isadmin:em7admin -X POST "https://<your_
hostname>/api/v1/applications/run" -H 'Content-Type:
application/json' -H 'cache-control: no-cache' -d '{"name":
"interface_sync_sciencelogic_to_servicenow","params": {"loglevel":
10}}'
```

LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information on this Site, information on this Site may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described in this Site at any time without notice.

Copyrights and Trademarks

ScienceLogic, the ScienceLogic logo, and EM7 are trademarks of ScienceLogic, Inc. in the United States, other countries, or both.

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic's trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic's legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: legal@sciencelogic.com. For more information, see https://sciencelogic.com/company/legal.

ScienceLogic

800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010