HPE Swarm Learning Installation and Configuration Guide

Revision history

Part number	Publication date	Edition	Summary of changes
10-191041-Q222	April 2022	1	New manual
10-191041-Q322	July 2022	2	Content related to Swarm Learning 1.1.0 is updated: • **Environment variables** section updated for `SL_WAIT_FOR_FULL_QUORUM_SECONDS`, `SWCI_TASK_MAX_WAIT_TIME`, and `SWOP_KEEP_CONTAINERS` • **Installing HPE Swarm Learning Management UI (SLM-UI)** section updated. • **Running Swarm Learning using CLI** section updated for primary and secondary APLS IP and Port information. • **Starting SWCI nodes** section updated with a note on `swci-init` script file. • **Upgrading Swarm Learning** section added.

Table Continued

Part number	Publication date	Edition	Summary of changes

10-191041-Q223	April 2023	3	Content related to Swarm Learning 2.0.0 is updated:
			
			• Updated **Prerequisites**.
			
			• **Installing HPE Swarm Learning Management UI (SLM-UI)** updated with SLM-UI feature.
			
			• **Running SLM-UI Installer** updated for SLM-UI Installer.
			
			• Added the following sections:
			   ◦ **Running Swarm Learning examples using SLM-UI**
			   ◦ **Adding a Swarm Host**
			   ◦ **Creating a Project**
			   ◦ **Adding Swarm Nodes**
			   ◦ **Creating a task**
			   ◦ **Creating Task Runners**
			   ◦ **Creating a Contract**
			   ◦ **Executing a Task**
			   ◦ **Managing the Global Settings**
			   ◦ **Starting SLM-UI manually**
			
			• **Running Swarm Learning** updated for DNS, SL node and SN node and Machine Learning parameters.
			
			• **Starting Sentinel node** updated for `sn-api-service`, `--sentinel-ip` and `sn-p2p-service` parameters.
			
			• **Starting Swarm Learning Node** updated for `sn-api-service` and `sl-fs-service` parameters.
			
			• **Starting SWOP node** section updated with `sn` parameters.
			
			• **IP address of host systems** and **Exposed port numbers** sections updated with new content.
			
			• **Environment variables** section updated with `SL_MAKE_ME_ADMIN` and `SL_LEADER_FAILURE_BASE_TIMEOUT`.
			
			• **Versioning and upgrade** section updated with new content.

Table Continued

Part number	Publication date	Edition	Summary of changes

10-191041-Q323	October 2023	4	Content related to Swarm Learning 2.1.0 is updated:
			
			• Added the following sections:
			  ◦ **Centralized Swarm diagnostic**
			  ◦ **Running Swarm Learning with SE Linux**
			  ◦ **Running Swarm Learning with Podman**
			  ◦ **Uninstalling Swarm Learning using SLM-UI**
			
			• **Prerequisites** section updated with Docker and Podman versions.
			
			• **Installing the License Server** updated with new content.
			
			• **Creating a Project in SLM-UI** section updated with new screenshot.
			
			• **Adding Swarm nodes** section updated with new screenshot.
			
			• **Starting Sentinel node** updated with new parameter – `v \<blockchain path on host machine \>:/platform/swarm/SMLNODE`.
			
			• **Environment variables** section updated with new notes, `SL_RAM_INTENSIVE` and `SL_LEADER_FAILURE_BASE_TIMEOUT`.
			
			• **Running SLM-UI Installer** updated with new note content.
			
			• **Running Swarm Learning examples using SLM-UI** section updated with new content.
			
			• **Starting SLM-UI manually** section updated with new content.
			
			• **Versioning and upgrade** section updated with new content.

10-191041-Q224	February 2024	5	Content related to Swarm Learning 2.2.0 is updated:
			
			• Added the following sections:
			  ◦ **Running Swarm Learning with SPIRE**
			  ◦ **SPIFFE**
			  ◦ **SPIRE**
			  ◦ **Executing a task**
			  ◦ **Targeted execute**
			
			• **Installing Swarm Learning using SLM-UI** section updated with new content.

Part number	Publication date	Edition	Summary of changes
			__Creating a Project in SLM-UI__ updated with new screenshot. __Execute task__ section updated with new content. __Adding Swarm Nodes__ and __Running Swarm Learning using CLI__ sections updated with new notes.

Contents

Introduction

HPE Swarm Learning is a decentralized, privacy-preserving Machine Learning (ML) framework. Swarm Learning framework utilizes the computing power at, or near, the distributed data sources to run the ML algorithms that train the models. It uses the security of a blockchain platform to share learnings with peers safely and securely. In Swarm Learning, training of the model occurs at the edge, where data is most recent, and where prompt, data-driven decisions are mostly necessary. In this completely decentralized architecture, only the insights learned are shared with the collaborating ML peers, not the raw data. This tremendously enhances data security and privacy.

There are various types of Swarm nodes - SN, SL, SWOP, and SWCI. For more information about the architecture of Swarm Learning, how these nodes work, how model training happens, see the *HPE Swarm Learning User Guide*.

Prerequisites

HPE recommends that you run each Swarm Network node, and Swarm Learning node on dedicated systems to get the best performance from the platform. The recommended requirements for each system are as follows:

NOTE: Configuration of the ML user nodes are driven by the complexity of the ML algorithm. GPUs may be needed.

Hardware

- Any x86-64 hardware.

- System memory of 32 GB or more. Hard disk space of 200 GB or more.

- Qualified with HPE Edgeline, Proliant DL380, and Apollo 6500.

Network

- A minimum of one or a maximum four open TCP/IP ports in each node. All swarm nodes **must be able to access** the ports of every other node. For more information on port details that must be opened, see **Exposed ports**.

- Stable internet connectivity to download Swarm Learning package and Docker images.

Operating systems

- Linux - Qualified on Ubuntu 22.04, RHEL 8.5, SLES 15.

- For Swarm SLM-UI installer, any x86-64 hardware running Linux, Windows, or Mac.

Container hosting platform

- HPE Swarm Learning is qualified with Docker 24.0.5 and Podman 3.4.4. Configure Docker or Podman to use IPv4. For more details on podman, see **Running Swarm Leaning with Podman**.

- Configure Docker to run as a non-root user. For more details, see **Manage Docker as a non-root user**.

- Configure network proxy settings for Docker. For more details, see **HTTP/HTTPS proxy**.

Machine Learning framework

Qualified with Keras 2.9.0 (TensorFlow 2 backend) and PyTorch 1.5 based ML models implemented using Python3.

NOTE: Python version must be between 3.6 to 3.9.

Multi system cluster requirements

Synchronized time across all systems using NTP.

NOTE: 'Qualified' in this section means that HPE has qualified the product with the respective versions. Swarm Learning may work with other versions as well.

HPE Swarm Learning Installation

Procedure

1. **Installing the License Server**

2. **Installing HPE Swarm Learning Management UI (SLM-UI)**

3. **Installing Swarm Learning using SLM-UI**

Installing the License Server

Procedure

1. After purchasing Swarm Learning from HPE, you will receive an email with a download link **Access Your Products**.

2. From the email, click **Access Your Products**. You are redirected to **MY HPE SOFTWARE CENTER**.

3. If you have the HPE Passport account, enter the credentials and **Sign In**. If you do not have it, create the HPE Passport Account and **Sign In**.
   After signing in, you should see the Software Notification Message Receipt page listing the products.

4. Download APLS container and run it using the following procedures.

   a. Login to the HPE docker registry using your HPE Passport email id and password `hpe_eval`.

      ```
      docker login hub.myenterpriselicense.hpe.com -u <HPE-PASSPORT-EMAIL-ID> -p
      hpe_eval
      ```

   b. Enable Docker content trust.

      ```
      export DOCKER_CONTENT_TRUST=1
      ```

   c. Pull the image with a tag.

      ```
      docker pull hub.myenterpriselicense.hpe.com/hpe_eval/autopass/apls:9.14
      ```

   d. Configure Data persistence.

      In order to retain configurations and installed licenses across containers, HPE recommends you to create a volume to persist the `/hpe` directory. This directory contains the following details:

Image Directory	Subdirectories	Description

`/hpe`	`AutoPass/LicenseServer/data`
	 `..data/conf`
	 `..data/log`
	 `..data/database`
		License server configuration directory. Contains database, logs and configuration files required to persist setup across containers transactions such as restarts, deletion or upgrades to new image tags.

      HPE recommends you to create the volume using the docker volume create command and assign a volume name such as `apls-volume`, as follows:

      ```
      docker volume create apls-volume
      ```

   e. Create and run the APLS container.

To run the APLS Docker container, user can use `docker cli` using the following instructions:

Docker CLI

```
docker run -d \
--name apls \
-v apls-volume:/hpe \
-p 5814:5814 \
--restart unless-stopped \
hub.myenterpriselicense.hpe.com/hpe_eval/autopass/apls:9.14
```

NOTE: In case the APLS container does not work, then user can choose to install APLS software using the APLS installer. User can select the AutoPass License Server (APLS) Installer link under 'Additional Notes' and download the **APLS** software. To install the APLS software on a host machine (Linux or Windows), see *AutoPass License Server User Guide*, which is part of the downloaded APLS software.
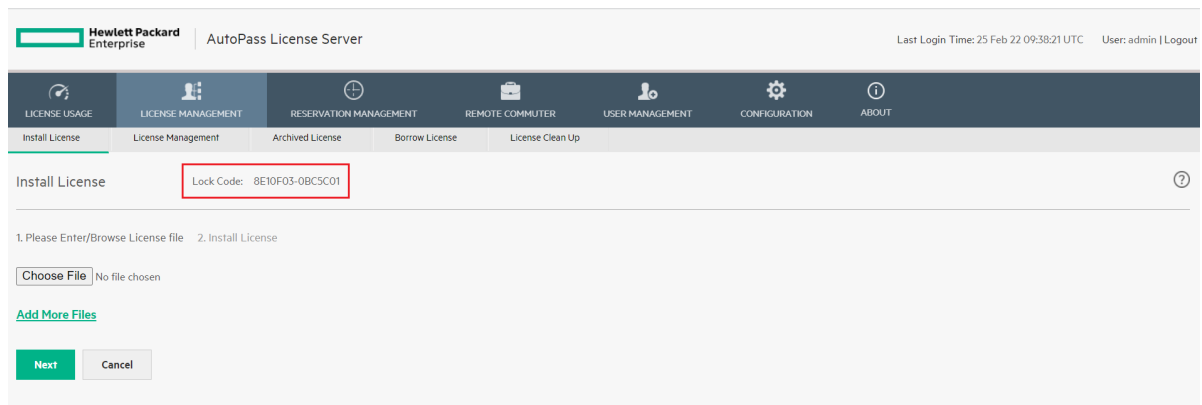
5. From a browser, access the APLS management console using the URL `https://<localhost>:5814` on the host machine where you installed the license server.

   The default user name is *admin*, and the password is *password*.

   NOTE: These instructions assume that the host IP of license server is *<localhost>* and the external port is *5814*. Host IP is the IP of the system where the license server is running. Modify these values to match the actual IP and external port on your system.

   If the web browser cannot connect to the APLS management console, check your network proxy settings and firewall policies. Consider techniques like port forwarding to work around firewall policies. If necessary, work with your network administrator to diagnose and resolve connectivity problems.

6. In the APLS web GUI, go to **License Management** -> **Install License** and note down the lock code.

7. Go to the Software Notification Message Receipt page and click **Access Your Products**.
   You will be navigated to the **MY HPE SOFTWARE CENTER** home page. After signing in with your HPE Passport credentials, you will see the **Activate** page.

8. Activate the license:

   a. Select the number of licenses to activate and click **Next**.

      NOTE: You can select the number of licenses to be installed on the host machines. For example, if you have 5 licenses, you can install 2 on Windows, and 3 on Linux machines.

   b. Designate yourself or for another user for activation. Click **Next**.

   c. Enter the lock code you got from the **Install Licenses** page in the HPE Serial Number field and click **Activate**.

9. Once you activate the licenses, you will see the **Download Files** page. Select the keys and the software and download them.

10. Install and manage the Swarm Learning license:

    a. Open the APLS management console.

    b. Select **License Management** -> **Install License**.

    c. Select **Choose** file to upload the license file that you downloaded and click **Next**.

    d. Select the required feature IDs and click **Install Licenses**.

For more information on installing and managing licenses, see the chapter *HPE AutoPass License Server License Management* in *AutoPass License Server User Guide*.

Installing HPE Swarm Learning Management UI (SLM-UI)

About this task

Installing Swarm Learning is a two-step process.

1. Using SLM-UI Installer, you can install the SLM-UI on one host.

2. Using SLM-UI, you can install Swarm Learning in multiple hosts.

Procedure

1. Navigate to the **MY HPE SOFTWARE CENTER** home page.

2. Perform the following actions after signing in with your HPE Passport credentials:

    a. Go to **My Activations** and select your ordered product.

    b. Go to **Action** pull down and then select **Download/Re-download** page.

    c. Select and download listed software files.

- The tar file containing docs and scripts.

- The signature file for the above tar file.

- The docker digest hash file (JSON).

- Download the Swarm Learning SLM-UI installer for your platform, Mac, Windows, or Linux.

Running SLM-UI Installer

Procedure

1. Run the SLM-UI Installer Web App. This is launched in a web browser.

   > ⓘ **IMPORTANT:** For Mac users, HPE recommends you to run the downloaded SLM-UI installer from the terminal window only.
   >
   > When you run the SLM-UI installer on Windows, web browsers such as Google Chrome and Microsoft Edge, are only supported. Internet Explorer browser is not supported.

   The installer has a few configurable options. To change the default options, run the installer from a command prompt. Use the following optional flags to customize the configuration or behavior of the installer:

   -port

   Defines the port for the application to run. The default value is 30302.

   Example, `-port 30355`

   -logs

   If enabled, displays the detailed message. To enable, use the command, `-logs verbose`.

   -version

   Defines the version of SLM-UI to be installed. Default value: 2.0.0.

   Example, `-version 2.0.0`

   -timeoutDuration

   Defines installer timeout duration for individual installation tasks. The default value is 300 seconds.

   Example, `-timeoutDuration 600`

Overview Requirements Docker Registry Database Install Next Steps Summary

What is Swarm Learning?

Swarm Learning is a decentralized, privacy-preserving Machine Learning framework. This framework utilizes the computing power at, or near, the distributed data sources to run the Machine Learning algorithms that train the models. It uses the security of a block chain platform to share learnings with peers in a safe and secure manner. In Swarm Learning, training of the model occurs at the edge, where data is most recent, and where prompt, data-driven decisions are mostly necessary. In this completely decentralized architecture, only the insights learned are shared with the collaborating ML peers, not the raw data. This tremendously enhances data security and privacy.

Installing Swarm Learning is a two-step process.
1. Firstly, using this SLM-UI Installer, you should install Swarm Learning Management UI (SLM-UI) on one host.
2. Using SLM-UI, you can install Swarm Learning on multiple hosts and run the examples (details in 'Next Steps' tab)

Next

2. Click **Next** in the **Overview** screen.

3. Review the **Requirements** and ensure that you have the hosts identified with required configuration, and click **Next**.

4. In the **Docker Registry Access**, enter your HPE Passport credentials and click **Next**.

5. In the **Database**, click **Next** if you want to use the default values.

NOTE:

If you want to configure the database with customized values, then make sure that those customized field values are valid.

6. Install the SLM-UI by providing the following details.

HPE SLM-UI Installer

Overview — Requirements — Docker Registry — Database — Install — Next Steps — Summary

Install SLM-UI

Swarm Installation Location ⑦

/opt/hpe/swarm-learning

The immediate parent directory should have *write permissions* for the Linux user specified. If the directory exists, the contents would be overwritten.

Upgrade ? ⑦

☐

You are Installing SLM-UI with latest version by un-selecting the checkbox. Please read hint carefully before proceeding!!

Host Server Name or IP Address ⑦

IP Address

SSH Port Number (Optional) ⑦

22

Username

Username

Password

Password

Additional Options +

Back   Run

7. Click **Run**.
A success message is displayed as **SLM-UI install successful**.

8. Click **Next**.

9. Review **Next Steps** and click **Next**.

Next Steps

1. Install License Server: ⌄

2. Generate Swarm Learning Licenses: ⌄

3. Install Licenses: ⌄

4. SLM-UI: ⌄

Back   Next

10. Review the **Summary** screen, which displays all the installed hosts. Click **Finish**.

This concludes the installation of SLM-UI installer. The SLM-UI will be launched in a separate browser window.

Sign In
to Swarm Learning

Email

example@my.com

Password

Enter your password

Sign In

Need an account?

User can login into the SLM-UI using default account (Username: `admin`; Password: `admin`).

After logging in to the SLM-UI, user can run the examples. User can also create the custom account.

Starting SLM-UI manually

SLM-UI containers are automatically started by the SLM-UI installer. But, if it is not running on the host, users need to manually start it using the following commands.

1. Initially, run `<swarm-learning>/slm-ui/scripts/run-postgres -pw" supersecretpassword"`. (`supersecretpassword` is a default database password. User can change this default database password using external tools like pgAdmin).

2. Then, run `<swarm-learning>/slm-ui/scripts/run-slm-ui -pw" supersecretpassword"`. Most of the other arguments of the above commands are optional. If you have changed the defaults, then you can use the following arguments to specify them.

The following arguments are the optional arguments for `run-postgres`:

Argument	Description
`-d|--data <dir>`	Directory where DB data will be persisted to default: `/opt/hpe/swarm-learning/slm-ui/data`
`-u|--user <string>`	Postgres database username. default: `postgres`
`-pw|--password <string>`	Postgres database password. This is a mandatory argument.
`-po|--port <number>`	Postgres database port. default: 5432
`-n|--network <string>`	Docker network where database and SLM-UI communicate. default: `slm-ui-network`
`-h|--help|/h`	help

The following arguments are the optional arguments for `run-slm-ui`:

Argument	Description
`-nw|--network <name>`	Docker network where SLM-UI and database communicate. default: `slm-ui-network`
`-l|--log <dir>`	Directory where log files will be saved to. default: `./logs`
`-pu|--public-cert <name>`	Public Certificate Name.
`-pr|--private-cert <name>`	Private Certificate Name.

Table Continued

Argument	Description
`-ca|--ca-cert <name>`	Certificate Authority Certificate Name.
`-h|--help|/h`	help

Installing Swarm Learning using SLM-UI

Procedure

1. In the SLM-UI **Hosts** tab, click **Add Host**.

   User can install Swarm Learning on any host or VM using this step.

   NOTE:

   a. User needs to repeat these steps for all the hosts on which the Swarm Learning needs to be installed.

   b. The host needs to have a Docker container runtime. Podman as a container runtime is not supported using SLM-UI.

2. Enter the Host address or Range of IPs.

   Multiple host addresses can be added by providing comma seperated IPs or range of IPs. Range is supported only for last octet. For example,

   Format 1: 127.1.1.11-20, 127.2.1.1

   Format 2: 127.1.1.11-20, 127.2.1.15-20

   Format 3: 127.1.1.1, 127.2.1.1, 128.1.2.2

   NOTE: The Host installation popup displays the status of the host installation. If any of the host is failed to install, retry the host installation process.

3. Select the Swarm version number from the drop-down menu.

4. Enter the Swarm Install Path.

   The default installation directory is `/opt/hpe/swarm-learning`. This directory path must have write permission.

5. Enter the Host username and password.

6. Enter the HPE Passport username.

7. Click **Create** to create the Swarm host.

Versioning and Upgrade

Upgrading Swarm Learning

To upgrade to a latest version:

- Delete the existing host from the SLM-UI. Click the remove icon in host page to delete the host.

  This step deletes all swarm artifacts (docs, examples, lib, and scripts), but not the user created artifacts like workspace folder or SLM-UI project folder, and hence preserving user generated artifacts.

  This step is not applicable, if you are running any older version of Swarm Learning (version < 2.0.0).

- Add the host again with new Swarm version on the same install location that was used earlier. If user selects the same installation location, the system overwrites the earlier version artifacts. If user needs both older and newer versions of Swarm Learning on the same host, they need to set a new installation location. User must restart the training with new images if needed. For more information, see **Installing Swarm Learning using SLM-UI**.

SLM-UI can handle multiple hosts each with their own version of Swarm and hence supporting a manual rolling upgrade, where users can upgrade one host at a time. However on a given host machine, if multiple versions are installed, only the last installed version is seen active. All nodes in a Swarm Learning running cluster must be running the same version of the product.

For example, Hosts A, B, C can have Swarm 1.1.0 version and a training can be started using these hosts. Simultaneously, Hosts E and D can have Swarm 2.0.0 version installed and an independent training can be started within the E and D hosts. SLM-UI can support both these multi-version configurations and trigger the training.

Upgrading SLM-UI

If user checks the **Upgrade** checkbox, all the previous projects and the created host artifacts are retained. While adding new hosts using SLM-UI, user gets the flexibility to choose the latest installed version or the previously installed version of Swarm. It also displays all projects, hosts, license servers, etc., which were added previously.

The following image shows the SLM-UI installer with **Upgrade** option:
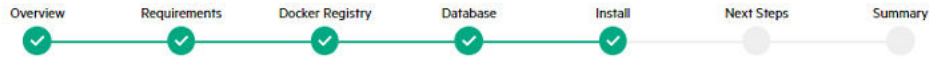
Overview   Requirements   Docker Registry   Database   Install   Next Steps   Summary

Install SLM-UI

Swarm Installation Location ⑦

    /opt/hpe/swarm-learning

The immediate parent directory should have *write permissions* for the Linux user specified. If the directory exists, the contents would be overwritten.

Upgrade ? ⑦

☐

You are installing SLM-UI with latest version by un-selecting the checkbox. Please read hint carefully before proceeding!!

Host Server Name or IP Address ⑦

    IP Address

SSH Port Number (Optional) ⑦

    22

Username

    Username

Password

    Password

Additional Options +

Back   Run

Uninstalling Swarm Learning using SLM-UI

In the **Hosts** tab, user needs to click the delete icon next to each of the hosts where user wants to uninstall the product. This stops all Swarm Learning components on that host, removes the docker container images, and deletes the `docs`, the `examples`, and the `scripts` directories installed under `swarm-learning`.

⚠️ **CAUTION:**

- This command deletes all user created artifacts under the `examples` directory.

- If needed, any log output produced by the containers must be saved before invoking the script. Logs are not available after the script is executed.

- Also, the output files that have been written under the `examples` directory by previous runs may need to be saved.

Running Swarm Learning examples using SLM-UI

About this task

> **TIP:**
>
> User can get a quick intuition and overview of SLM-UI by looking at this **video**.

> Before you start using the SLM-UI, HPE recommends you reading the *HPE Swarm Learning User Guide* to understand about the architecture of Swarm Learning, how these nodes work, how model training happens, and Swarm Learning Concepts section.

Perform the following steps to run the SL examples using SLM-UI.

1. For ease of use, users can extract the `examples` folder from the documentation tar ball on their laptops.

2. Users must generate x.509 certificates for their own environment. HPE-provided `gen-cert` utility under `swarm-learning/examples/utils` can be used for generating self-signed certificates just for the purposes of running the examples. Run the `gen-cert` utility on one of the Linux host to generate the self-signed certificates. Ensure that these certificates are copied to your laptop under the `examples` folder.

3. User can use Linux remote desktop to display the SLM-UI on the local linux browser (instead of the remote laptop browser). With this, the SLM-UI's file open dialog box shows all local Linux files to the user. This makes it easier to upload the training artifacts like certs, SWOP profile, Task yaml, etc.

   Alternatively, user can use cross-platform file sharing to mount the artifacts directory from the Linux machine on his laptop.

4. User can use the provided template files and the generated certificates for the examples and import them into the SLM-UI installer.

5. User can use the certificate names which are hardcoded inside the provided template yaml files. In case, user changes the certificate names, then they need to make the corresponding changes in the SWOP yaml file.

NOTE: HPE recommends that users use their own certificates in actual production environment.

Procedure

1. **Creating a Project**
2. **Adding Swarm Nodes**
3. **Creating a Task**
4. **Creating Task Runners**
5. **Creating a Contract**
6. **Executing a Task**
7. **Managing the Global Settings**
8. **Centralized Swarm diagnostic**

Creating a Project in SLM-UI

Procedure

1. In **My Projects**, click **Create Project**.

   Create Project
   Upload your project files here

   Project name*
   [Enter Project Name]

   Swarm Version*
   [v2.0.0-rc2-Jira654 ⌄]

   Project files

   Project in SLM-UI is a logical representation of a particular Swarm training.

   Projects help to view deployment topology and monitor the progress for the given Swarm training. They define what all Swarm nodes (and associated host nodes) a training will run, the model being used, the x.509 certificates, SWOP and Task yaml files for a particular training. Multiple Projects can be defined in a single instance of SLM-UI.

   Project artifacts are created under the `swarm-learning/slm-ui/projects/<project number>` automatically once the project is saved.

   NOTE: HPE recommends that users avoid editing these files directly on the host machine.

2. Enter the project name.

3. Select the Swarm version number from the drop-down menu.

4. Upload the following files:

   a. Upload your Certificate and CA Certificate files to be used with your swarm nodes.

   b. Upload your Certificate and CA Certificate files to be used with SWCI node used by the SLM-UI.

   Certificates

   Upload your Certificate and CA Certificate files to be used with your swarm nodes

   Certificate upload
   [Upload]

   CA Certificate upload
   [Upload]

   Upload your Certificate and CA Certificate files to be used with SWCI node used by the SLM-UI

   SWCI Public Certificate upload
   [Upload]

   SWCI Private Certificate upload
   [Upload]

   SWCI CA Certificate upload
   [Upload]

   c. Upload the Machine Learning model file.

Machine Learning Model

Model upload

[Upload]

---

d. Upload the SWOP Profile schema file.

SWOP Profile

SWOP profile upload

[Upload]

Click on the open icon to view the uploaded file.

e. Upload the SWCI tasks file.

SWCI TASKS

SWCI task upload

[Upload]

Click on the open icon to view the uploaded file.

NOTE:

- For standard examples, this artifacts are available as part of tar file. User can extract the examples on their laptop.

- In a two node example, the network names should be `host-1-net` for the sentinel node and `host-2-net` for the non-sentinel node, respectively.

5. Click **Save Project** to create the Project.

My Projects

Manage your projects here

Project name	Created	Author		
FD	6/14/2023	admin	🗑	📄
Mnist2Node	6/21/2023	admin	🗑	📄

Adding Swarm Nodes

About this task

The following procedure can be used to add the SN node and the SWOP node.

Procedure

1. In the **Projects** tab, click the created Project name which is displayed under **Project name** field.

   The system displays the **Project Nodes**, **Tasks**, **Task Runners**, **Contracts** and **Settings** tabs.

2. Click **Add nodes** under **Project Nodes** tab.

3. Add the Swarm Node by providing the following details.

Add Swarm Node

Please fill out the form to add a new Swarm node.

Swarm Node Type*

| sn | ⌄ |

Swarm Node Name*

| Swarm Node Name |

Is Sentinel

| ☐ This is a sentinel node |

Sentinel Address*

| Sentinel Address |

Host Address*

| Select Host Address | ⌄ |

Public Cert*

| Public Cert | ⌄ |

Private Cert*

| Select Private Cert | ⌄ |

Use Project CA Path

| ✔ Use Project CA Path |

Exposed SN Api Port

| 30304 |

Exposed P2P Port

| 30303 | ⬍ |

Docker Network

| Docker Network |

Show advanced

[Create] **Cancel**

NOTE:

a. In a two node example, the network names must be `host-1-net` for the sentinel node and `host-2-net` for the non-sentinel node, respectively as this are as specified in the associated SWOP profile.

b. The node launch may fail if the correct Cert is not provided for each sub-type of the certificate option.

c. For SWOP nodes, user can specify the correct proxy server ENV variables as part of **Show Advanced** > **Node parameters**. Otherwise, ML application may not build and/or model related data files would NOT be downloaded correctly.

Docker Network

| Docker Network |

Hide advanced

SN Api Service

| SN API Service |

SN P2P Service

| SN P2P Service |

Sentinel Api Service

| Sentinel Api Service |

Node Parameters

| --env value --env value |

Create   **Cancel**

For example,

```
-e https_proxy=http://web-proxy.XYZ.net:8080 -e http_proxy=http://web-proxy.XYZ.net:8080
```

4. Click **Create** to add the swarm node.

NOTE: To visualize the Swarm Network topology through SLM-UI unambiguously and also to effectively utilize the new feature of **Targeted Execute**, HPE recommends to give unique container names for each Swarm container.

Creating a task

Procedure

1. In the **Tasks** tab, click **Create new task**.

2. Create a task by providing the following details.

Create Task

Please fill out the form to Create new Task.

SN Name*

Select SN ⌄

Task Name*

Select task file ⌄

Create Cancel

3. Click **Create** to create the task. The created task is displayed under the Task List.

Creating Task Runners

Procedure

1. In the **MyProjects**, click the **Project name** link.

2. In the **Task Runners** tab, click **Create new task runner**.

   If you are running concurrent Swarm training, you need to create multiple task runners. If you are running a single training, the default task runner would be good enough and this step is optional.

3. Create new Task Runner by providing the following details.

Create new Task Runner

Please fill out the form to add task runner.

Task runner name*

Enter Task Runner Name

Create Cancel

4. Click **Create**. The Task Runner id created and displayed under **Task Runner** list.

Task Runner list   Create new task runner

Name	Created	State	
defaulttaskbb.taskdb.sml.hpe	2023-03-21T07:17:51.000Z	COMPLETE	Reset

Creating a Contract

Procedure

1. In the **MyProjects**, click the **Project name** link.

2. In the **Contracts** tab, click **Create new contract**.

   If you are running concurrent Swarm training, you need to create multiple contracts. If you are running a single training, the default contract would be good enough and this step is optional.

3. Create new contract by providing the following details.

Create new Contract

Please fill out the form to add contract.

Contract name*

Enter Contract Name

Create Cancel

4. Click **Create**. The contract is created.

Contracts

Create new contract

Name	Type	Created	Completed Peers	Min Peers	Peer Count	
defaultbb.cqdb.sml.hpe	TRAINING	2023-03-21T07:17:38.000Z	2	2	2	Reset

Executing a task

Once you create the task, there are two options (**Execute** and **Targeted Execute**) available to execute the task as follows.

Task list

Create new task

Task name	Task type	Author	Finalized		
ROOTTASK	ROOT_TASK	HPE	✓	Execute	Targeted Execute
user_env_tf_build_task	MAKE_USER_CONTAINER	HPESwarm	✓	Execute	Targeted Execute
swarm_mnist_task	RUN_SWARM	HPESwarm	✓	Execute	Targeted Execute

Execute task

About this task

User can use the **Execute Task** to initiate execution of various types of Swarm tasks on a task runner and see the progress of the execution of the task.

Procedure

1. In the **Task List**, click **Execute**.

2. Execute the task by providing the following details.

The following screenshot shows typical training task's Execute panel.

Execute Task

Please fill out the form to start a training run.

> ⚠ Executing a task will reset any training contracts and task runners that it uses.

Task Runner*

| Select Task Runner | ⌄ |

Contract*

| Select Training Contract | ⌄ |

> ⓘ ML model code must be updated if a non-default contract is selected.

Annotation/Description of the experiment

| Enter Unique Description |

Select to save the experiment

| ✅ Save experiment |

Execute Task on 'All Peers'

| ☐ All Peers |

Or Execute Task on selected # of peers*

| 1 |

> ⓘ Either 'All Peers' or '# of Peers' can be selected.

Create Cancel

User can select either **All Peers** or **# of Peers** to execute the task. If user selects **All Peers** option, it schedules the task on all the SWOPs listening on the selected Task Runner.

3. Enter unique description and select **Save Experiment** to save a particular execution of an experiment.

NOTE: The **Contract*** field is displayed only if the **Task Type** is set as **RUN_SWARM**.

---

4. Click **Create** to execute the task.

Targeted execute

About this task

User can use **Targeted Execute** to initiate execution of various types of Swarm tasks on a specific SWOP or to add a specific SWOP to an already executing task runner. This targeted execute command helps in retrying a failed task on a particular SWOP.

Procedure

1. In the **Task List**, click **Targeted Execute**.

2. Execute the task by providing the following details.

   The following screenshot displays the typical training task's Execute panel.

Targeted Execute

Please fill out the form to execute task on a given SWOP Target

New experiment or Join a running experiment?

☐ New Experiment/Training

Task Runner*

Select Task Runner ⌄

Contract*

Select Training Contract ⌄

ⓘ ML model code must be updated if a
non-default contract is selected.

Annotation/Description of the experiment

Enter Unique Description

Select to save the experiment

✓ Save experiment

SWOP ID*

Select Target SWOP ⌄

Create   **Cancel**

---

a. Enter unique description and select **Save Experiment**.

The save experiment will be enabled only if it is a new experiment.

b. Select SWOP ID to execute the task on that SWOP.

Targeted Execute on a SWOP can be used to add the selected SWOP to an already executing Task runner or to initiate a new execution of the selected SWOP.

c. Select **New Experiment/Training** checkbox if this is a new training task targeting SWOP.

NOTE: The **Contract*** field is displayed only if the **Task Type** is set as **RUN_SWARM**.

If the **Task Type** is selected other than **RUN_SWARM**, Targeted Execute panel is shown as follows.

Targeted Execute

Please fill out the form to execute task on a given SWOP Target

Reset Task Runner?

☐ Reset Task Runner

Task Runner*

Select Task Runner ⌄

SWOP ID*

Select Target SWOP ⌄

[Create] **Cancel**

---

    d. Select **Reset Task Runner** only if this is the first task on the taskrunner.

3. Click **Create** to execute the task.

Managing the Global Settings

The Global settings feature allows the user to create multiple custom user accounts, configure swarm settings and configure License server settings. It also displays the version number of swarm software.
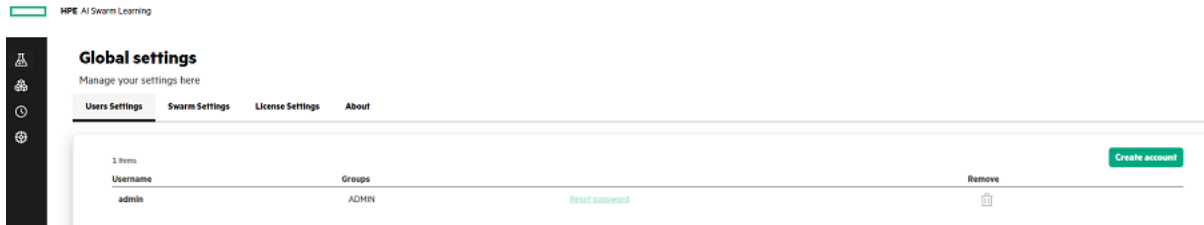
Configuring the User Settings

Procedure

1. In the **User Settings**, click **Create Account**.

2. Enter the Username and the Password.

You can also generate the password by clicking the **Generate Password** link.

3. Click **Create** to create the User account.
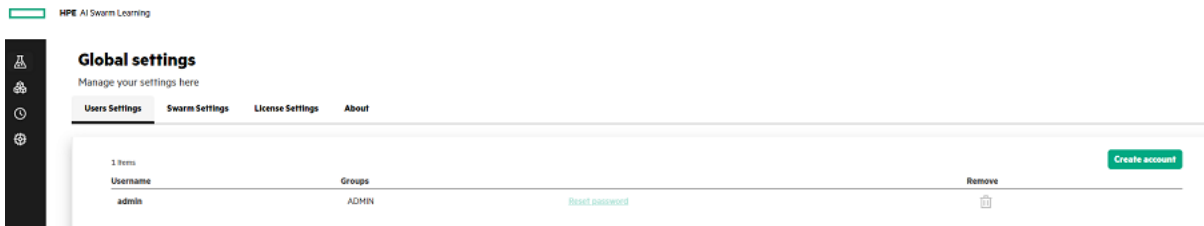
Configuring the Swarm Settings

Procedure

1. In the **Swarm Settings**, click **Add Version**.

2. Enter the Swarm Version number.

3. In the **Swarm Bundle**, select the Swarm version files from your local drive or drag and drop the Swarm version files.

4. Select the **Set as default** checkbox if you want to set this Swarm version as the default version number in the SLM-UI.

5. Select the **Set as evaluation** checkbox if you are using the community version of Swarm Learning.
    For more information, see **Versioning and upgrade** section.

6. Click **Create** to create the Swarm version.

Configuring the License Settings

Procedure

1. In the **License Settings**, click **Update License Server**.

2. Enter the License Server Address and the License Server Port.

3. Click **Update** to update the License server.
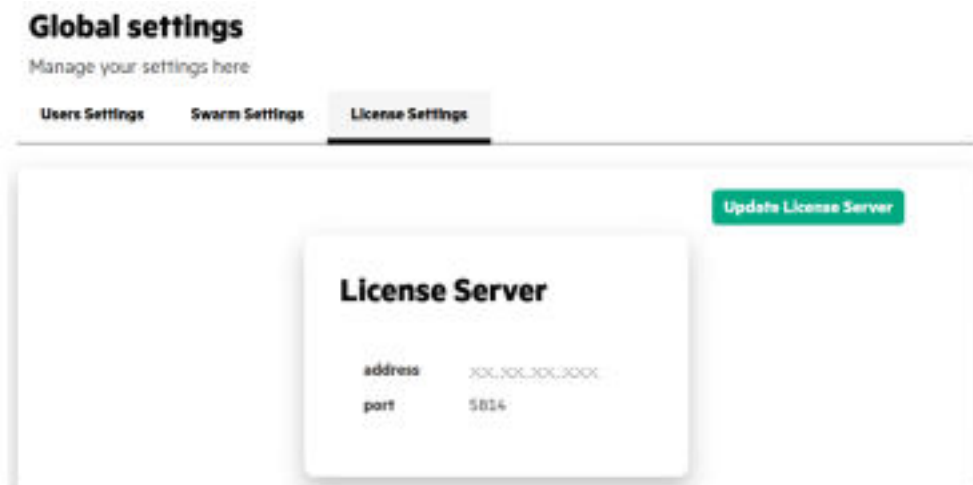
Centralized Swarm diagnostics

About this task

Centralized Swarm diagnostics utility can be used to collect and upload the logs from all the hosts associated with a project. This can be sent to HPE to report Swarm issues.

Centralized Swarm Diagnostics

This utility will collect and upload the logs from all the hosts associated with this project to the below host/VM.

Host Address*ⓘ

Select Host Address ⌄

Create   Cancel

Procedure

1. In the **Projects** tab, click **Collect Log**.

My Projects
Manage your projects here

Create project

Project name	Created	Author		
FD	6/14/2023	admin	🗑	📄
Mnist2Node	6/21/2023	admin	🗑	📄

2. Select **Host Address** from the Host Address drop-down menu.

Centralized Swarm Diagnostics

This utility will collect and upload the logs from all the hosts associated with this project to the below host/VM.

Host Address*⑦

Select Host Address ⌄

Create   **Cancel**

NOTE: The sshpass utility must be installed in the VM. *(apt install sshpass)*.

---

3. Click **Create** to complete the Log Collection.

● **Log Collection Completed!** ✕
From hosts : XX.XX.XX.117, XX.XX.XX.139
All logs copied to Host XX.XX.XX.117 @
logs/Swarm_Diag_1_1687348965794.tar.gz

Running Swarm Learning using CLI

> **TIP:** Before you start running the following commands, HPE recommends you to read the *HPE Swarm Learning User Guide* to understand about the architecture of Swarm Learning, how these nodes work, how model training happens, and so on.
>
> For examples of how to provide options to the various run commands, see the *Examples* chapter in *HPE Swarm Learning User Guide.*

> **IMPORTANT:**
>
> - Ensure that network proxy settings are configured correctly and the containers are able to communicate to each other.
>
> - Ensure that Docker is configured to run as a non-root user by adding your current user ID as part of the Docker group.
>
> - Ensure that the system time is synchronized across the systems by using NTP.
>
> - To understand the Swarm Network topology through `list nodes` SWCI command unambiguously and also to effectively utilize the new SWCI commands to `ASSIGN TASK`, HPE recommends to give unique container names for each Swarm container.

Start and run Swarm Learning in the following order. Make sure that License Server is started and the licenses are installed.

1. The Sentinel Node

2. Start the Swarm Network node before starting any of the associated Swarm Learning nodes.

3. After the training is completed, stop all the containers using the script `stop-swarm` on all nodes.

The scripts in the `swarm-learning/scripts/bin` directory is used to start these components. To run the scripts, a bash shell and a Linux environment is required.

> **NOTE:** The default directory where Swarm Learning is installed is `/opt/hpe/swarm-learning`. If the user has changed the default installation directory, all the run commands can be found in that location.

All start scripts take the following common options for configuring the Docker run command that is used to start the container.

> **NOTE:** These options do not apply to the `swarm-learning/scripts/bin/stop-swarm` script. These options are similar to those of the Docker run command.

Parameter name	Description	Default value
`--hostname <name>`	The host name assigned to the docker container.	`--name`, if it is specified. Otherwise, Docker assigns a host name.
`--name <name>`	The name assigned to the docker container.	Docker assigns a random name to the container.
`--network <network name>`	The docker network that the container should belong to.	Docker's default bridge network.

Table Continued

Parameter name	Description	Default value
`--pull`	Pull the docker image from its repository before running it.	False, the image is not pulled from its repository, if it is already available locally
`--sudo`	Prefix the Docker commands with "sudo".	False, if the current user belongs to the docker group; true otherwise.
`-d, --detach`	Run the container in the background.	A pseudo-terminal is allocated if the launcher has an associated terminal; otherwise, the container is run in the background
`-i, --interactive`	Keep STDIN open even if not attached to a terminal.	STDIN is kept open if a pseudo-terminal is allocated to the container; otherwise, it is closed.
`-t, --tty`	Allocate a pseudo-terminal for the container.	A pseudo-terminal is allocated if the launcher has an associated terminal; otherwise, the container is run in the background.
`-e, --env var=val`	Set an environment variable inside the container.	
`-l, --label key=val`	Set metadata on a container.	
`-p, --publish host-port:container-port`	Publish a container port to the host.	
`-u, --user { name | uid } [ : { group | gid } ]`	User and group ID to use inside the container.	
`-v, --volume host-path:container-path`	Bind mount a volume.	
`-w, --workdir container-path`	Working directory inside the container.	
`--dns`	The IP address of the custom DNS server. If there are more than one custom DNS servers, then for each DNS, repeat the same argument with different IP address.	
`--rm`	Request Docker to automatically remove the container when it exits.	
`-h, --help`	This (helpful) message.	
`--primary-apls-ip <IP address or DNS name>`	The IP address on which the primary Autopass License Server is serving license requests.	None

Table Continued

Parameter name	Description	Default value
`--secondary-apls-ip <IP address or DNS name>`	The IP address on which the secondary Autopass License Server is serving license requests.	None
`--primary-apls-port <port numberw>`	The port number on which the primary Autopass License Server is serving license requests.	5814
`--secondary-apls-port <port number>`	The port number on which the secondary Autopass License Server is serving license requests.	The value assigned to `--primary-apls-port`
`--apls-pdf <path to license PD file>`	The path to the license PD file to be used.	None
`--cacert <path to certificates file>`	The path to the file containing the list of CA certificates.	None
`--capath <path to certificates directory>`	The path to the directory containing CA certificate files.	None
`--cert <path to certificate file>`	The path to the certificate file that provides the component's ID.	None
`--key <path to key file>`	The path to the private key file corresponding to the certificate.	None
`--socket-path <SPIFFE Workload API socket>`	Path, volume or container hosting the socket on which the SPIFFE Agent serves the Workload API.	None
`--host-ip <IP address or DNS name>` (Mandatory parameter)	The IP address or DNS name of the host system on which this Swarm Learning node is created.	
`--sn-ip <IP address or DNS name>`	The IP address or DNS name of the host system on which the Swarm Network (SN) node with which this Swarm Learning node must associate, is running.	
`--sn-api-port <port number>`	Host port for the API Server of the associated Swarm Network node	30304
`--sn-api-service <fqdn>:<port number>`	Fully Qualified Domain Name for the SN API Service of associated SN node. Here, Port number is optional.	
`--sl-fs-port <port number>`	Host port for this Swarm Learning node's File Server.	30305
`--sl-fs-service <fqdn>:<port number>`	Fully Qualified Domain Name and optional port for this Swarm Learning node's file service.	

User machine learning container parameters

Parameter name	Description	Default value
`--ml-image <ML image name>` (Optional parameter)	Name of the User's Machine Learning image.	
`--ml-entrypoint <entrypoint>` (Optional parameter)	Entry point to the Machine Learning container.	
`--ml-cmd <command>` (Optional parameter)	Command to the Machine Learning container.	
`--ml-w <directory path>` (Optional parameter)	Working directory of the Machine Learning container.	
`--ml-name <container name>` (Optional parameter)	Name of the Machine Learning container.	
`--ml-v <host-path:container-path>` (Optional parameter)	Bind mount a volume for the Machine Learning container.	
`--ml-e <environmental-variable-name=value>` (Optional parameter)	To pass environmental variable to the Machine Learning container.	
`--ml-user <uid:gid> -` (Optional parameter)	The access privilege with which the ML container needs to be spawned on the host. If `--ml-user` is not provided, then ML container would be spawned with current host user's `uid:gid`. If only `uid` of the host user is provided, then ML container would be spawned with specified host user's `uid` and primary `gid`. If `uid:gid` of the host user is provided, then ML container would be spawned with specified host user's `uid:gid`.	
For AMD GPUs, one may need to use the following parameters:	For more information, see **https://developer.amd.com/resources/rocm-learning-center/deep-learning/**.	

Table Continued

Parameter name	Description	Default value
`--ml-device`	Expose host devices to the container, as a list of strings.	None
`--ml-ipc`	Sets the IPC mode for the container.	None
`--ml-shm-size`	Size of `/dev/shm` (for example, 1G).	None
`--ml-group-add`	List of additional group names and/or IDs that the container process will run as.	None
`--ml-cap-add`	Add kernel capabilities.	None
`--ml-security-opt`	A list of string values to customize labels for MLS systems, such as SELinux.	None
`--ml-privileged`	Provides extended privileges to this container.	None

Starting Sentinel node

Use the `swarm-learning/scripts/bin/run-sn` script to start Sentinel and Swarm Network (SN) nodes. This script accepts the following parameters:

Parameter name	Description	Default value
`--host-ip <IP address or DNS name>`	The IP address or DNS name of the host system on which this Swarm Network node is created.	None
`--sentinel`	If this flag is passed, this node does the Blockchain initialization and make configuration information ready to be shared with other SN nodes. Also it does not expect sentinel node IP to be passed. If this flag is not passed this node is a regular SN node and needs a sentinel node IP for initialization.	
`--sentinel-ip <IP address or DNS name>`	Any running SN node's host IP address or DNS name can be passed as a value. If this parameter is not specified, this Swarm Network node makes itself as the sentinel.	None
`--sn-api-port <port number>`	The host port for this Swarm Network node's API Server.	None
`--sn-api-service <fqdn>:<port number>`	Fully Qualified Domain Name for the SN API Service of associated SN node. Here, Port number is optional.	

Table Continued

Parameter name	Description	Default value
`--sn-p2p-port <port number>`	The host port for this Swarm Network node's P2P communications.	None
`--sn-p2p-service <fqdn>:<port number>`	Fully Qualified Domain Name for the SN P2P Service of associated SN node. Here, Port number is optional.	
`-v \<blockchain path on host machine\>:/ platform/swarm/SMLNODE`	The host path where user wants to persist blockchain across SN restart.	By default, Blockchain data will not be preserved over SN restart.

Starting Swarm Learning node

Use the `swarm-learning/scripts/bin/run-sl` script to start a Swarm Learning (SL) node. This script accepts the following parameters:

NOTE:

- This script starts only one Swarm Learning node at a time. To launch multiple Swarm Learning nodes, you must invoke this script as many times as desired and on appropriate host systems.

- HPE recommends to use SWOP to automatically launch SL and ML nodes, which is a preferred way. For more information on launching, *HPE Swarm Learning User Guide*.

Parameter name	Description	Default value
`--host-ip <IP address or DNS name>`	The IP address or DNS name of the host system on which this Swarm Network node is created.	None
`--sn-ip <IP address or DNS name>`	The IP address or DNS name of the host system on which the Swarm Network node with which this Swarm Learning node must associate, while running.	None
`--sn-api-port <port number>`	The host port for this Swarm Network node's API Server.	None
`--sn-api-service <fqdn>:<port number>`	Fully Qualified Domain Name for the SN API Service of associated SN node. Here, Port number is optional.	
`--sn-docker-name <container name>`	Docker container name for the associated Swarm Network node.	None
`--sl-fs-port`	The host port for this Swarm Learning node's File Server.	None
`--sl-fs-service <fqdn>:<port number>`	Fully Qualified Domain Name and optional port for this Swarm Learning node's file service.	

Starting SWCI nodes

Use the `swarm-learning/scripts/bin/run-swci` to launch SWCI. The SWCI command prompt is displayed when the launch is successful. You can enter any command from a pre-defined set of commands. It supports a list of well-defined commands that are self-explanatory. There is a built-in online help, that lists all supported commands and further one can drill down and see help for each command.

```
SWCI:2 > help HELP
HELP [command:string]
Help without parameter lists all supported commands.
Help with command name show help content for the specified command.
SWCI:3 >
```

NOTE: You must launch the SWCI node after the SN nodes are started.

The run-swci script accepts the following parameters:

Parameter		Default value
`--usr-dir <dir>`	The host directory that must be used as the user directory by this SWCI node.	None
`--init-script-name <swci-init file>`	Name of the SWCI script file. This file must be located inside the user directory at the top level.	`swci-init`

NOTE:

- If you need to use the `swci-init` script file as-is (default), `--usr-dir` option must be specified and SWCI looks for this default script file under this user directory.

- If you want to run a script file with a different filename, you must explicitly specify the `--init-script-name` and `--usr-dir`.

- If the `--usr-dir` is not specified, the SWCI runs in an interactive mode.

Starting SWOP nodes

Use the `swarm-learning/scripts/bin/run-swop` script to start SWOP nodes. This script accepts the following parameters:

Parameter name	Description	Default value
`--usr-dir <dir>` (Mandatory Parameter)	Host directory that should be used as the user directory by this SWOP node.	
`--profile-file-name <swop-profile file>` (Mandatory Parameter)	This file should be located inside the user directory, at the top-level itself.	

Table Continued

Parameter name	Description	Default value
`--sn-ip <IP address or DNS name>`	The IP address or DNS name of the host system with which this SWOP node must associate with the corresponding Swarm Network node.	None
`--sn-api-port <port number>`	The host port for this Swarm Network node's API Server.	None
`--sn-api-service <fqdn>:<port number>`	Fully Qualified Domain Name for the SN API Service of associated SN node. Here, Port number is optional.	None

Stopping Swarm Learning node

Use the `swarm-learning/scripts/bin/stop-swarm` script to stop all Swarm Network and Swarm Learning nodes that are running on a host system.

NOTE: This script does not operate across systems. It must be invoked on each host system to stop the Swarm Learning platform completely.

This script accepts the following parameters:

Parameter name	Description	Default value
`--all`	This parameter stops all components, SL, SN, SWCI, and SWOP.	None
`--sl`	This parameter stops Swarm Learning nodes.	None
`--sn`	This parameter stops Swarm Network nodes.	None
`--swci`	This parameter stops SWCI nodes.	None
`--swop`	This parameter stops SWOP nodes.	None
`--keep`	This parameter keep stopped containers - they are removed by default. This parameter is optional. When it is specified, the containers are stopped but not removed. In this case, the Log output from the containers are still available after the script has run. The leftover containers can be removed, either manually or by invoking this script again without the `--keep` parameter.	None
`--sudo`	This parameter specifies that sudo must be used when invoking Docker to stop or remove the Swarm Learning nodes.	None

Uninstalling the Swarm Learning package

Use the `swarm-learning/scripts/bin/uninstall` script to uninstall the Swarm Learning package. This script does not accept command line parameters. It should be run on every node where Swarm Learning package is installed.

On the host where it is run, the script stops all Swarm Learning components removes the docker container images, and deletes the "docs", "examples", and the "scripts" directories installed under `swarm-learning`.

> **△ CAUTION:**
>
> - This command deletes all user created artifacts under the "examples" directory.
>
> - If needed, any log output produced by the containers must be saved before invoking the script. Logs are not available after the script is executed.
>
> - Also, the output files that have been written under the "examples" directory by previous runs may need to be saved.

Running Swarm Learning with SPIRE

SPIFFE

The Secure Production Identity Framework for Everyone (SPIFFE)[1] standard provides a specification for a framework capable of bootstrapping and issuing identity to services that work across heterogeneous environments and organizational boundaries.

Components of SPIFFE

- SPIFFE ID: It standardizes an identity namespace. It is a URI that serves as the "name" of an entity.

- SPIFFE Verifiable Identity Document (SVID): It dictates that how an issued identity is presented and verified. An SVID is a document that carries the SPIFFE ID. SVID includes cryptographic properties that allow it to be proven as authentic and proven to belong to the presenter. The supported document types are an X.509 certificate or a JWT token.

- Workload API: It specifies an API through which identity may be retrieved and/or issued. The SPIFFE Workload API is the method through which workloads, or compute processes, and obtain their SVID(s). It is typically exposed locally (for example, via a Unix domain socket). The Workload API also delivers the CA bundles. These bundles are associated with trust domains outside of the issued SVID and are used for federation.

SPIRE

SPIRE[2] is a SPIFFE Run time Environment. SPIRE follows the SPIFFE standard and does the following:

- Node and workload attestation

- Securely issues SVIDs to workloads

- Verifies the SVIDs of other workloads

- Exposes the SPIFFE Workload API

- Attests running software systems

- Issues SPIFFE IDs and SVIDs to software systems

Components of SPIRE

SPIRE has two major components:

- SPIRE Server: It is responsible for authenticating agents and minting SVIDs.

- SPIRE Agent: It is responsible for serving the SPIFFE Workload API.

SPIRE usage

SPIRE can be used in three different ways:

---

[1] **https://github.com/spiffe/spiffe/blob/main/standards/SPIFFE.md**
[2] **https://github.com/spiffe/spire**

- Standalone - Installing SPIRE on a dedicated machine.

- Docker Compose/Docker - Installing SPIRE in a docker container.

- Kubernetes – Installing SPIRE in K8S via kubectl.

Although the SPIRE can be run in multiple ways, it is ideal to make use of the spire via the docker images itself because Swarm core components are docker images. Docker images are available for both spire-agent and spire-server via git hub container registry (ghcr) or chain guard registry (cgr). Both the docker images can be easily started by setting up the respective configuration files (`server.conf` and `agent.conf`) appropriately. User can try the quick start example[3] that uses docker-compose or can refer to the vanilla docker[4] way of spire execution. For more information on understanding the SPIRE and SPIFFE concepts in detail, see a book called Solving the Bottom Turtle[5].

Due to the federative nature of Swarm Learning, it has multiple hosts and multiple organizations involved in swarm training. For more information on how to actually run with SPIRE, see *CIFAR-10 using SPIRE* section in *HPE Swarm Learning User Guide*. The following are an overview of the steps.

To run Swarm Learning using SPIRE on a multi-host setup, HPE recommends to proceed with the following steps:

- Each host must have its own spire-agent container running.

- Each spire-agent must have seperate join token.

- Entry creation is specific to each swarm component. No need to create multiple entries in the SPIRE server for the same component.

To run Swarm Learning using a federated spire[6] setup, HPE recommends to proceed with the following steps:

- Each organization has its own SPIRE server started.

- The federation block of the `server.conf` holds the information of other SPIRE servers in the federation.

- Along with the SPIRE server API port (default 8081), an additional port (default is 8443) is needed for the federation service.

- A Bundle file is created at each SPIRE server and exchanged with other SPIRE servers for federation.

- Workload entries are specific to each SPIRE server. For example, the entry of SN workload must be done on each organization where SN exists.

NOTE: Workload is identified via the selector that is defined while creating an entry to the SPIRE server. Ensure that if an environment is used while creating an entry, then the same environment must be used in the `docker run` command. Also, ensure that if a label is used while creating the entry, then the same label must be used in the `docker run` command.

---

[3] **https://spiffe.io/docs/latest/try/spire101/**
[4] **https://github.com/spiffe/spire/blob/v1.8.4/doc/plugin_agent_workloadattestor_docker.md**
[5] **https://spiffe.io/pdf/Solving-the-bottom-turtle-SPIFFE-SPIRE-Book.pdf**
[6] **https://spiffe.io/docs/latest/architecture/federation/readme/**

Running Swarm Learning with Podman

- Install podman-base package `podman`.

- Install `podman-docker` package from system package manager. This package maps docker commands to respective podman commands.

- Enable `podman.socket`. It can be root or rootless. For more information on setting rootless podman socket, see **https://docs.podman.io/en/latest/markdown/podman-system-service.1.html**.

NOTE:

- Shorthand registry names may not work with podman. For more information, see *Short-name aliases* section in **www.redhat.com/sysadmin/manage-container-registries**.

- ML or user container running as non-root may fail due to permission issue while saving trained model file. User needs to precreate the required directory with full permissions.

- GPU based local training is not supported through Podman.

- SLM-UI is not supported with Podman.

Running Swarm Learning with SE Linux

TIP: User must consult their system administrator before running Swarm with SE Linux in their system or doing any security related changes on the system.

When SE Linux is enabled, it restricts access to various system resources. Before starting Swarm, user must apply appropriate security context labels to `workspace`. For more information, see *Platform specific SE Linux guide*.

To get access to various system resources, user needs to apply `svirt_sandbox_file_t` security context label to `workspace_path` using the `chcon` command.

```
chcon -Rt svirt_sandbox_file_t <workspace_path>
```

For more information, see **docker_selinux_security_policy**.

SN changes to support SE Linux

For SE Linux environment, `workspace` label set by `chcon` are sufficient.

SWOP changes to support SE Linux

SWOP needs access to host podman socket which is restricted by default. To get access to host podman socket, user needs to pass the following parameter in `run-swop` script:

```
--docker-socket-file="<podman socket path>"
--security-opt=label=type:container_runtime_t
```

- `docker-socket-file` parameter exposes `podman.socket` to SWOP container; and

- `container_runtime_t` label allows the container to interact with the `container.runtime` and access the socket.

NOTE: Paths in SWOP profile need to be labeled with `svirt_sandbox_file_t` security context using the `chcon` command.

SL or ML changes to support SE Linux

For SE Linux environment, `workspace` label set by `chcon` are sufficient.

Configuring Swarm Learning

This chapter provides information about various configurations involved in the Swarm Learning.

IP address of host systems

The `--host-ip` and `slhostip` IP addresses in the run scripts and the SWOP profile are the IP addresses of the host machine, where the respective containers are running on the host machine. Based on access, user can even use the FQDN of the host system.

By default, Swarm Learning framework uses a Docker bridge network. For improved isolation, users can even use a user-defined bridge network.

While using the user-defined bridge network, the options `--ip` and `ip` field of `slnetworkopts` in SWOP profile are the IP addresses of the container themselves. This case is specific to the reverse proxy examples or scenarios where user wants to use the fixed IP addresses for containers.

Exposed port numbers

Depending on the type of Swarm Learning components that are running on a host, some or all these ports must be opened to allow the Swarm Learning containers to communicate with each other:

- A Swarm Network peer-to-peer port on the hosts running Swarm Network nodes. By default, port 30303 is used.

- A Swarm Network API server port on the hosts running Swarm Network nodes. By default, port 30304 is used.

- Swarm Learning file server port on the hosts running Swarm Learning nodes. By default, port 30305 is used.

- A License Server API port on the host running the License Server. By default, port 5814 is used.

- (Optional). An SWCI API server port that is used by the SWCI node to run a REST based API service. By default, port 30306 is used.

NOTE:

1. If you use different ports other than the default port, you must open those ports accordingly. For instance, in our MNIST example, we are using ports 16000 and 18000 for the SL File server ports, which must be opened.

2. If you use a reverse proxy, you need to open only the SN peer-to-peer port (30303) for each SN node.

Environment variables

The environment variables are passed to containers or added to the environment variable through profile or configuration files.

NOTE: Environment variables starting with a Swarm component name (for example, SN_, SL_) are meant for those particular components. Environment variables starting without a Swarm component name are meant for all Swarm components.

The following environment variables are available to set and modify:

Environment variable name	Description
SWARM_LOOPBACK	Used to bypass Swarm Learning to help you quickly develop, integrate, and test your model code with Swarm Learning package.
	If SWARM_LOOPBACK is set to 'True', then all Swarm functionality will be bypassed, except parameter validation.
	This can help you to verify and test integration of the model code with Swarm **without spawning any Swarm Learning containers**.
LOGS_DIR	Sets the directory for Swarm components log, it is set usually by Docker file.
USR_DIR	Sets the directory for Swarm components, it is set usually by Docker file.
SN_ETH_PORT_EXT	Sets an Ethernet port for Swarm Network node.
SN_I_AM_SENTINEL	Sets a Swarm Network node to become the Sentinel node, only when it is set to true.
	Default value: False
SN_START_MINING	Starts mining on non-sentinel nodes. (Optional)
	Default value: False
SL_MAKE_ME_ADMIN	Determines whether an SL node can participate in leader election or not.
	Default value: True
	If SL_MAKE_ME_ADMIN is set to 'False', the corresponding SL node will not participate in leader election. If user doesn't want to make a slow node (with less compute power, network band width etc) as a leader, then this can be set to 'False'.
SL_LEADER_FAILURE_BASE_TIMEOUT	Sets the minimum timeout value (in seconds). If Swarm merging does not happen within this timeout, a new SL leader node is selected. The swarm training continues to run, regardless of SL leader node failures. This timeout will kickin after `min_peers` nodes have completed their local training.
	Default value: 600 seconds
	This variable may need tunning depending on the ML application complexity.

Table Continued

Environment variable name	Description
SL_WAIT_FOR_FULL_QUORUM_SECONDS	Sets the maximum time for an SL leader node to wait for full quorum after minPeers are ready for merge. This parameter lets you to maximize the number of peers participating in the merge process. Default value: 30 secs
SL_RAM_INTENSIVE	Optimizes the usage of RAM in the SL leader node for coordinate and geometric median merge methods. Unlike mean merge method, coordinate and geometric median merge methods involve memory intensive operations. If SL Leader node has limited hardware (RAM) configuration, then merging the intermediate model parameters using the median methods can result in memory issues. For such scenarios, user can set up the SL_RAM_INTENSIVE flag to 'False' for merging the model parameters layer by layer. This 'False' option is based on I/O operations and is time consuming, hence the default option is set to 'True'. User can pass this parameter in slenvvars option within SWOP profile. This option can be different for each SL node depending on its hardware capacity. Example: 'slenvvars : [SL_RAM_INTENSIVE : False]' Default value: True
SWCI_TASK_MAX_WAIT_TIME	Specifies a maximum timeout value for the completion of a task. This value must be set in minutes, and the default is 120 mins (2 hours).
SWCI_MODE	Enables SWCIs web interface instead of command line interface. Allowed values are CLI and WEB. Default value: CLI
SWCI_STARTUP_SCRIPT	This is a default start script of SWCI.
SWCI_WEB_PORT	Port on which SWCI API server listens when run in WEB mode. Default value: 30306
SWOP_PROFILE	Indicates default profile for SWOP.
SWOP_KEEP_CONTAINERS	SL, ML pair containers and host volumes created by SWOP are removed. This option can be enabled to retain the stopped containers for debugging. Default value: False
SWARM_ID_CACERT	Indicates user CA certificates file.
SWARM_ID_CAPATH	Indicates user CA certificates directory.
SWARM_ID_CERT	Indicates user certificates file.

Table Continued

Environment variable name	Description
SWARM_ID_KEY	Indicates user SSH key file.
SWARM_SPIFFE_WORKLOAD_API_SOCKET_PATH	Used for acquiring a SPIFFE identity. It points to the UNIX domain socket on which the SPIFFE agent is serving the SPIFFE workload API. For more information, **https:// spiffe.io/**.

GNU General Public License

This product includes code licensed under the GNU General Public License, the GNU Lesser General Public License, and/or certain other open source licenses. A complete machine-readable copy of the source code corresponding to such code is available upon request. This offer is valid to anyone in receipt of this information and shall expire three years following the date of the final distribution of this product version by Hewlett Packard Enterprise Company. To obtain such source code, send a check or money order in the amount of US $10.00 to:

Hewlett Packard Enterprise Company

Attn: General Counsel

1701 E Mossy Oaks Rd

Spring, TX 77389

U.S.A.

Support and other resources

Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:

  https://www.hpe.com/info/assistance

- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:

  https://www.hpe.com/support/hpesc

Information to collect

- Technical support registration number (if applicable)

- Product name, model or version, and serial number

- Operating system name and version

- Firmware version

- Error messages

- Product-specific reports and logs

- Add-on products or components

- Third-party products or components

Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.

- To download product updates:

  Hewlett Packard Enterprise Support Center

    https://www.hpe.com/support/hpesc

  My HPE Software Center

    https://www.hpe.com/software/hpesoftwarecenter

- To subscribe to eNewsletters and alerts:

  https://www.hpe.com/support/e-updates

- To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:

  https://www.hpe.com/support/AccessToSupportMaterials

  ⓘ **IMPORTANT:** Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HPE Account set up with relevant entitlements.

Remote support

Remote support is available with supported devices as part of your warranty or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which initiates a fast and accurate resolution based on the service level of your product. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

If your product includes additional remote support details, use search to locate that information.

HPE Get Connected

https://www.hpe.com/services/getconnected

HPE Tech Care Service

https://www.hpe.com/services/techcare

HPE Complete Care

https://www.hpe.com/services/completecare

Warranty information

To view the warranty information for your product, see the **warranty check tool**.

Regulatory information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at the Hewlett Packard Enterprise Support Center:

https://www.hpe.com/support/Safety-Compliance-EnterpriseProducts

Additional regulatory information

Hewlett Packard Enterprise is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements such as REACH (Regulation EC No 1907/2006 of the European Parliament and the Council). A chemical information report for this product can be found at:

https://www.hpe.com/info/reach

For Hewlett Packard Enterprise product environmental and safety information and compliance data, including RoHS and REACH, see:

https://www.hpe.com/info/ecodata

For Hewlett Packard Enterprise environmental information, including company programs, product recycling, and energy efficiency, see:

https://www.hpe.com/info/environment

Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, use the **Feedback** button and icons (at the bottom of an opened document) on the Hewlett Packard Enterprise Support Center portal (**https://www.hpe.com/support/hpesc**) to send any errors, suggestions, or comments. This process captures all document information.