Writing Sample: Edited Text

<This document contains an example of how I edited text written by developers. The original version appeared in an internal confluence page. The edited version appeared as the last installation step in the Install Guide.>

Edited Version

Step 12: Enable Container Restart

You must perform one additional configuration to make sure that containers automatically restart when the Podman host reboots. This is because Podman was not originally designed to bring up a suite of containers at once. For details, see the Red Hat documentation.

To ensure that all containers restart after a Podman host reboot, use the Hansen-provided script, `generate_systemd_config.sh`, to automatically restart containers. The script uses this logic:

1. Create systemd unit files for each container in the host.
2. Each system unit file sets up an always-restart policy for its corresponding container.

Using the Container Restart Script

This section describes the steps you need to take to use `generate_systemd_config.sh` to configure containers to automatically restart.

Prerequisites

- You must have permission to work with systemd services.
- All containers must be up running.

Things to remember

- If you delete a container from the host you must also disable and delete the corresponding systemd unit file.

Procedure

1. In the Sigma Admin container, go to `<SigmaAdmin>/podman/tools/.`
2. To generate systemd files, open a command line and run the following command:
   `generate_systemd_config.sh <SIGMAADMIN_USER> <SIGMAADMIN_PASS> <SIGMAADMIN_URL> <CURRENT_DOCKER_HOSTNAME>`
   Where
   `SIGMAADMIN_USERC` is the user name you use to log in to the Sigma Admin server.
   `SIGMAADMIN_PASS` is the password you use to log in to the Sigma Admin server.
   `SIGMAADMIN_URL` is the URL you use to log in to the Sigma Admin server.
   `CURRENT_DOCKER_HOSTNAME` is the hostname of the current Podman host.

   The script creates systemd unit files for each container in <CURRENT_DOCKER_HOSTNAME> host.

3. To copy systemd files to the `/etc/systemd/system/ directory` and enable them, run the following commands for each container:

```
sudo cp <container_name>.service /etc/systemd/system
systemctl enable <container_name>.service
systemctl is-enabled <container_name>.service
```

   All containers will automatically restart if the Podman host reboots.

Original Version

The following image shows the original text:

Issue that containers not started automatically after host reboot

Redhat has suggested a workaround solution to start container as a systemd service. The details is in the link below.

Porting containers to systemd using Podman Red Hat Enterprise Linux 8 | Red Hat Customer Portal

After confirming with Redhat, we have added a helper script in SigmaAdmin to help setup the systemd unit files.

The script can be found in the Bootstrap SigmaAdmin, It is also available for copy from the sigmaadmin Docker container in the same location. `<SigmaAdmin>/podman/tools/generate_systemd_config.sh`

Note that the user running the script will need elevated permissions to work with systemd.

This script can be used to generate the systemd configuration files for all sigmaadmin managed containers on a host. The systemd unit file can be used to control the container. We will use it here to setup the always restart policy for all the sigmaadmin managed containers

Prerequisite : The containers should be up and running before running the script"
Usage : `generate_systemd_config.sh <SIGMAADMIN_USER> <SIGMAADMIN_PASS> <SIGMAADMIN_URL> <CURRENT_DOCKER_HOSTNAME>`

SIGMADMIN_URL,SIGMADMIN_USER, SIGMADMIN_PASS is the URL and user credentials to connect to sigmaadmin server.

For example, https://sigmaadmin_host:1000 sadmin sadmin

CURRENT_DOCKER_HOSTNAME is the hostname used in the docker ud of the current host where this script is running

The script will produce systemd unit file for each container in the <CURRENT_DOCKER_HOSTNAME> that are managed by sigmaadmin. Please copy the generated systemd files and enable them usually by

```
sudo cp <container_name>.service /etc/systemd/system
systemctl enable <container_name>.service
systemctl is-enabled <container_name>.service
```

If the container is permanently deleted from this host, you should also disable and remove the corresponding systemd unit file.