Deploying and Managing SUSE Edge on Lenovo ThinkEdge SE360 V2

Planning / Implementation

In this guide, we will be using SUSE Edge solutions for deployment and management. SUSE provides several key components to help manage and deploy edge devices, including Edge Image Builder (EIB), SUSE Linux Enterprise Micro (SLE Micro) operating system, and SUSE Rancher services. These key components are introduces below:

- SUSE Edge solution

  For the SUSE Edge solution, the primary components include Edge Image Builder (EIB), SLE Micro OS, and Rancher services. EIB allows for the creation of customized images tailored to specific environmental needs, including addressing the requirements of the ThinkEdge SE360 V2 system. These images can be deployed using Lenovo's XClarity and managed via the Rancher server for comprehensive server management. For detailed reference documents on SUSE Edge, please refer to SUSE Documentation.

- Edge Image Builder (EIB)

  Edge Image Builder (EIB) is a tool designed by SUSE to streamline and accelerate the process of generating Customized, Ready-to-Boot (CRB) disk images. These images can bootstrap machines and are effective even in fully isolated environments. When using Lenovo Edge systems, the EIB tool can generate SLE Micro images tailored to your needs for deployment. For detailed usage instructions, please refer to the SUSE documentation and GitHub resources.

- SUSE Linux Enterprise Micro (SLE Micro)

  SUSE Linux Enterprise Micro OS is a product designed by SUSE specifically for edge devices. It is a lightweight and secure edge operating system that combines the enterprise-hardened components of SUSE Linux with the features of a modern, immutable operating system, providing a simple and reliable infrastructure platform. For detailed usage instructions, please refer to the SUSE documentation

- SUSE Rancher

  Rancher is a Kubernetes management tool from SUSE that deploys and runs clusters anywhere and on any provider. Rancher can provision Kubernetes from a hosted provider, configure compute nodes, and then install Kubernetes onto them or import existing Kubernetes clusters running anywhere. For detailed information, please refer to the SUSE documentation.

In this paper, we will demonstrate how to utilize SUSE Edge components, including the Edge Image Builder (EIB), to create a customized SLE Micro 6.1, and use its customized image to simplify the bootstrapping process of machines. Additionally, we will use the Lenovo XClarity Controller to further deploy customized ISO images for remote machine deployment. Using SUSE Rancher, we will import the already deployed Lenovo Edge System SE360 V2 for resource management.

To create a boot image supporting SE360 V2 based on SLE Micro OS, the following are required:

- One host machine with SUSE Edge Image Builder installed
- One edge system to be deployed

Installing SUSE Edge Image Builder

This example installs on openSUSE Tumbleweed. The steps are as follows:

1. Install the podman tool.

```
#zypper install -y podman
```

```
SE450:~ # sudo zypper install -y podman
Loading repository data...
Reading installed packages...
Resolving package dependencies...
```

Figure 1: Installing the Podman Tool

2. Obtain the EIB image.

```
# podman pull registry.opensuse.org/isv/suse/edge/edgeimagebuilder/containerfile-sp6/suse/edge-image-builder:1.1.0
```

```
localhost:~/eib # podman pull registry.opensuse.org/isv/suse/edge/edgeimagebuilder/containerfile-sp6/suse/edge-image-builder:1.1.0
Trying to pull registry.opensuse.org/isv/suse/edge/edgeimagebuilder/containerfile-sp6/suse/edge-image-builder:1.1.0...
Getting image source signatures
Copying blob c669a40a73a3 done   |
Copying blob bcb014eb0029 done   |
Copying config 4f621fd63e done   |
Writing manifest to image destination
4f621fd63e2cea413e1445ec10678a1f33a620b574f3ce3f6f26843e492736d6
```

Figure 2: Obtaining the EIB Image

Creating a Custom Image for Boot Installation

We will proceed with the following steps to create a Custom Image for Boot Installation:

- Creating an Image Configuration Directory
- Creating an Image Definition File
- Generating a Custom Image File

Creating an Image Configuration Directory

Since EIB runs as a container, we need to set up a directory on the host to specify the required configurations and allow EIB to access necessary files and supporting items during the build process. This directory must follow a specific structure. We create this directory in the home directory and name it "eib":

```
# export CONFIG_DIR=$HOME/eib
# mkdir -p $CONFIG_DIR/base-images
```

In the previous step, we created the "base-images" directory to host the SLE Micro 6.1 ISO files. Now, we ensure that the downloaded ISO files are copied to the configuration directory:

```
# cp SL-Micro.x86_64-6.1-Default-SelfInstall-GM.install.iso /root/eib/base-imag
es/
```

At this point, the configuration directory should look like this:

```
├── eib
│   └── base-images
│       └── SL-Micro.x86_64-6.1-Default-SelfInstall-GM.install.iso
```

Creating an Image Definition File

The definition file describes the configurable options supported by Edge Image Builder. We can find a complete example of the options here and refer to the upstream image-building guide provided by SUSE for more flexible environment settings.

Below, we will build a custom image with SE360 V2 as the endpoint device.

Setting the Objectives

1. Create a customized SLE Micro OS 6.1.
2. Install the OS on a VROC RAID1 disk.
3. Configure SE360 V2 to use a fixed IP so that Rancher can obtain the information for import. If there are other network ports, use the DHCP IP address.

Creating the Definition File

1. Create a main definition file.
    - Generate an x86_64 image file.
    - Output the image file as eibimage-eib110.iso.
    - Define kernel parameters "auto=1" and "rd.kiwi.install.pass.bootparam," in kernelArgs.
    - Specify the services to be started: sshd.
    - Create the root and a regular user "conie", defining their home directories and passwords.
    - Install additional software, wget.
    - Enter the SUSE registration code to add additional software packages, wget, to the image ISO. For SUSE registration code, please contact SUSE.
    - Configure the Kubernetes version.

   The main definition file should look like this:

```
apiVersion: 1.0
image:
  imageType: iso
  arch: x86_64
  baseImage: SL-Micro.x86_64-6.1-Default-SelfInstall-GM.install.iso
  outputImageName: eibimage-eib110.iso
operatingSystem:
  isoConfiguration:
    installDevice: /dev/md126
  time:
    timezone: Asia/Taipei
  kernelArgs:
    - rd.auto=1
    - rd.kiwi.install.pass.bootparam
  systemd:
    enable:
      - sshd
  keymap: us
  users:
    - username: root
      createHomeDir: true
      encryptedPassword: $6$KPTiD9HrOvaBhCaL$tWD1z7peYm6e33xyquofodIKqWxj3MqM8UBUGo/RKnlNS9iC5iKr.JKVJAkN0s5adzb68run7eDwOzZYWoK.n/
    - username: conie
      createHomeDir: true
      encryptedPassword: $6$KPTiD9HrOvaBhCaL$tWD1z7peYm6e33xyquofodIKqWxj3MqM8UBUGo/RKnlNS9iC5iKr.JKVJAkN0s5adzb68run7eDwOzZYWoK.n/

  packages:
    packageList:
      - wget
    sccRegistrationCode: A[redacted]
kubernetes:
  version: v1.30.5+k3s1
  network:
  nodes:
    - hostname: host1.local
      type: server
  manifests:
    urls:
      - https://k8s.io/examples/application/nginx-app.yaml
~
```

Figure 3: Main definition file

2. Create a network directory and a network definition file, “local.yaml”, in the eib/network/ folder.
    - Set a fixed IP address for the SE360 V2 specific MAC address.
    - Define other networks as DHCP.

The network definition file should look like this:

```
routes:
  config:
  - destination: 192.168.0.0/24
    metric: 100
    next-hop-address:
    next-hop-interface: eth0
    table-id: 254

dns-resolver:
  config:
    server:
    - 192.168.0.254
    - 8.8.8.8

interfaces:
  - name: eth0
    type: ethernet
    state: up
    mac-address: 08:3a:88:fb:80:c8
    ipv4:
      enabled: true
      dhcp: true
      auto-dns: true
      auto-gateway: true
      auto-routes: true
    ipv6:
      enabled: true
      autoconf: true
      dhcp: true
      auto-dns: true
      auto-gateway: true
      auto-routes: true

  - name: eth1
    type: ethernet
    state: up
    mac-address: 08:3a:88:fb:80:c9
    ipv4:
      enabled: true
      address:
        - ip: 192.168.0.103
          prefix-length: 24
    ipv6:
      enabled: true
      autoconf: true
      dhcp: true
      auto-dns: true
      auto-gateway: true
      auto-routes: true
```

Figure 4: Network definition file

Generating a Custom Image File

Execute the podman command with the configuration file to generate a custom image.

```
# podman run --rm -it --privileged -v $CONFIG_DIR:/eib registry.opensuse.org/isv/suse/edge/edgeimagebuilder/containerfile-sp6/suse/edge-image-builder:1.1.0 build --definition-file iso-definition_eib11.yaml
```

```
localhost:~/eib # podman run --rm -it --privileged -v $CONFIG_DIR:/eib registry.opensuse.org/isv/suse/edge/edgeimagebuilder/containerfile-sp6/suse/edge-image-builder:1.1.0 build --definition-file iso-definition_eib11.yaml
Setting up Podman API listener...
Downloading file: dl-manifest-1.yaml 100% |████████████████████████████████| (498/498 B, 3.7 MB/s)
Generating image customization components...
Identifier ................... [SUCCESS]
Custom Files ................. [SKIPPED]
Time ......................... [SUCCESS]
Network ...................... [SUCCESS]
Groups ....................... [SKIPPED]
Users ........................ [SUCCESS]
Proxy ........................ [SKIPPED]
Resolving package dependencies...
Rpm .......................... [SUCCESS]
Os Files ..................... [SKIPPED]
Systemd ...................... [SUCCESS]
Fips ......................... [SKIPPED]
Elemental .................... [SKIPPED]
Suma ......................... [SKIPPED]
Populating Embedded Artifact Registry... 100% |████████████████████████████████| (1/1, 2439 it/s)
Embedded Artifact Registry ... [SUCCESS]
Keymap ....................... [SUCCESS]
Configuring Kubernetes component...
Downloading file: k3s_installer.sh
Kubernetes ................... [SUCCESS]
Certificates ................. [SKIPPED]
Cleanup ...................... [SKIPPED]
Building ISO image...
Kernel Params ................ [SUCCESS]
Build complete, the image can be found at: eibimage-eib110.iso
```

Figure 5: Generating a Custom Image File

Deploying Custom SLE Micro 6.1 to remote ThinkEdge SE360 V2

In this section, we will be using the XClarity Controller in the ThinkEdge SE360 V2 to perform the deployment. The following steps will guide you through the process of connecting to the XClarity Controller Web and remotely mounting the custom ISO for installation and deployment.

Connecting to ThinkEdge SE360 V2 XClarity Controller Web and Remotely Mounting Custom ISO

Notes: For more information about the Lenovo XClarity Controller, please refer to this link. https://pubs.lenovo.com/xcc/

1. Connect to the remote ThinkEdge SE360 V2 and remotely mount the customized ISO.

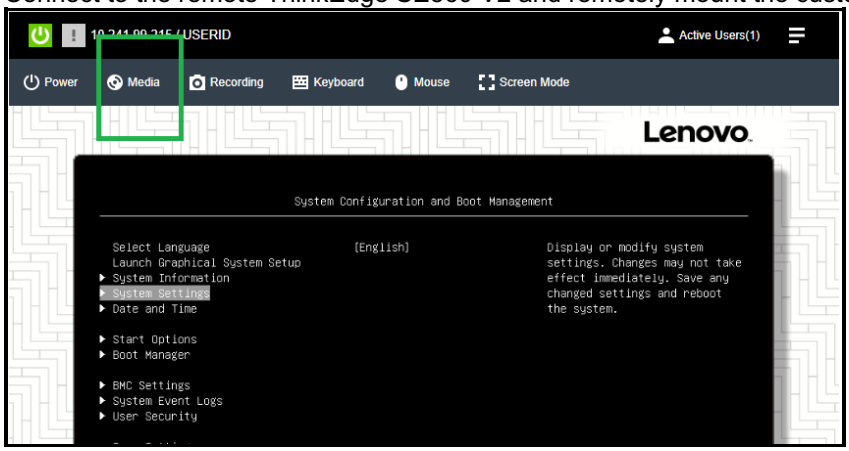

Figure 6: Select the Media option

2. Ensure the customized ISO is mounted.

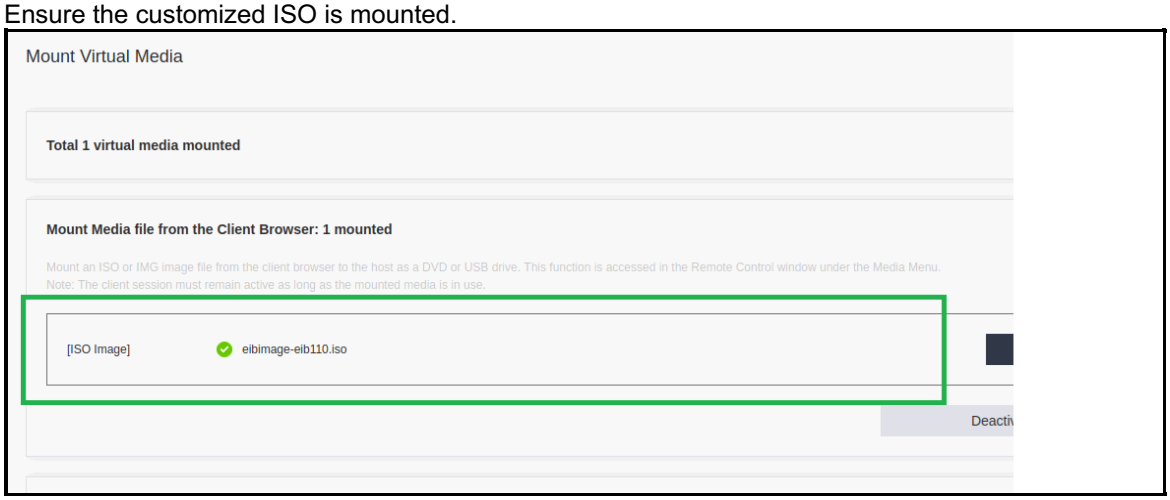

Figure 7: Mount the customized ISO

3. Perform the customized ISO installation, automating any interactive modes until the installation is complete.

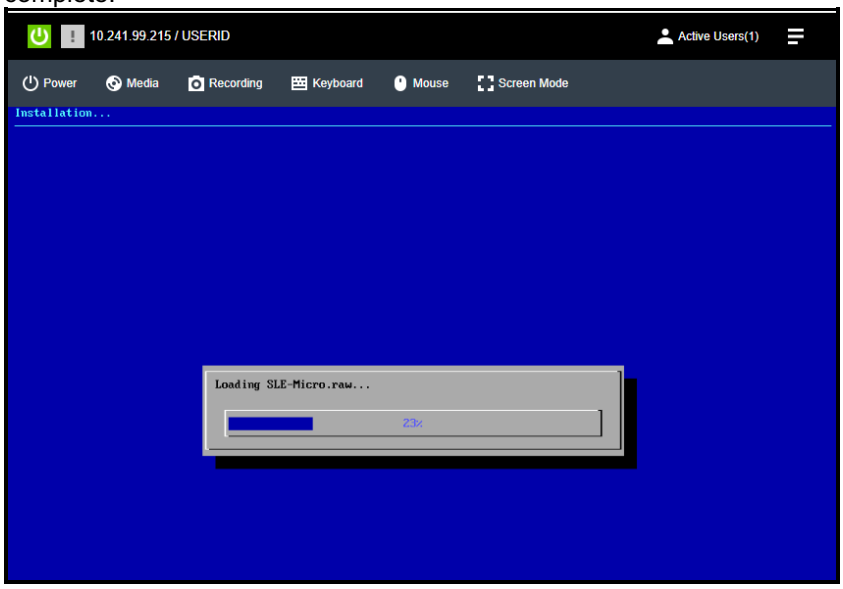

Figure 8: Perform the customized ISO installation

4. Confirm the OS is fully installed on the ThinkEdge SE360 V2.

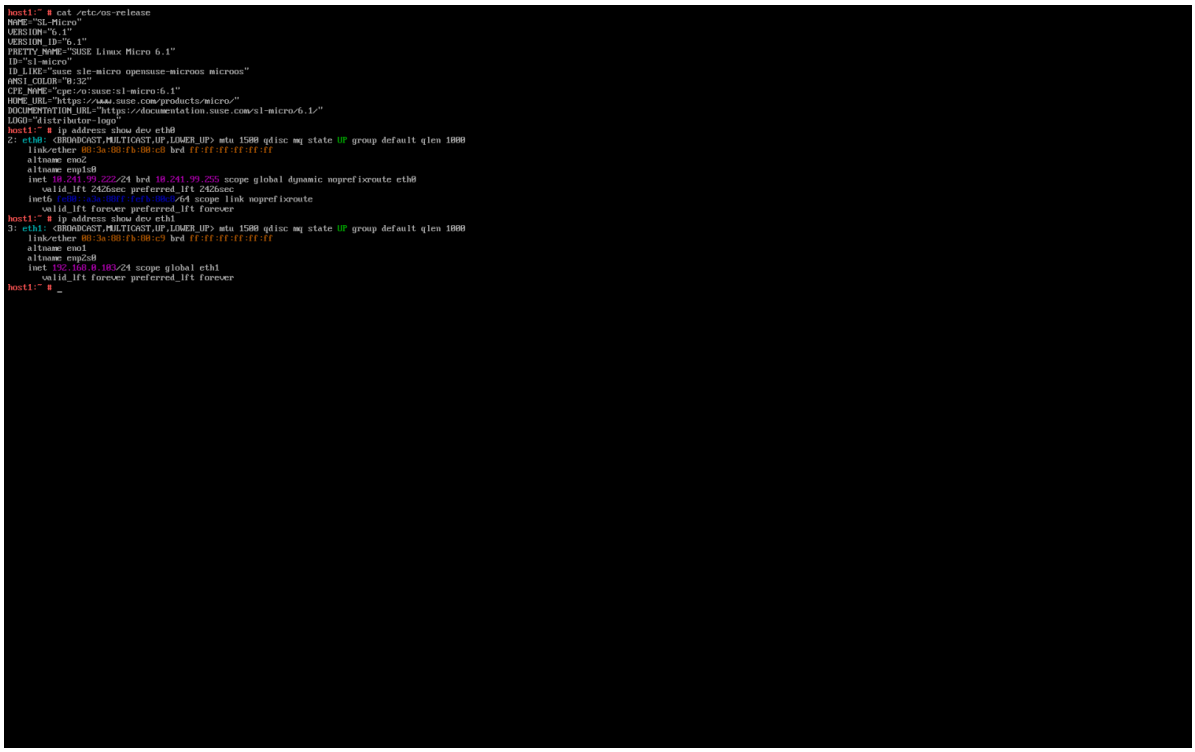

Figure 9: Confirm the OS is fully installed

Installing Rancher

This example will use SUSE OS for installation. If you want to use other operating systems to install Rancher software, you can refer to the SUSE Rancher documentation.

1. Install Docker:

```
#zypper install docker
#systemctl enable docker
```

2. Install the Rancher container. Once the installation is complete, you can access the Rancher interface.

```
# docker run --privileged -d --restart=unless-stopped -p 80:80 -p 443:443
rancher/rancher
```

Access the Rancher interface and import the node host1

After installing SUSE Rancher, if you are unsure about the login password, you can use the following command to query it:

1. Obtain the Rancher container ID.
2. Retrieve the initial password.

```
# docker ps
CONTAINER ID   IMAGE             COMMAND           CREATED         STATUS
PORTS
           NAMES
e7932b12c399   rancher/rancher   "entrypoint.sh"   8 minutes ago   Up 8 mi
nutes   0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443-
>443/tcp   relaxed_chatterjee
# docker logs e7932b12c399 2>&1 | grep "Bootstrap Password:"
2025/01/13 07:48:25 [INFO] Bootstrap Password: 5h9t6kfgbft5qqnh9fk9wvdjkcv
tkn2nt7v8qxdwr2s5sswch62cx6
```

If the Rancher password is forgotten, reset the password as follows:

1. Ensure Rancher is running properly: Open the main node shell and run `docker ps` to confirm the Rancher container is running.

```
# docker ps
CONTAINER ID   IMAGE             COMMAND           CREATED        STATUS
PORTS
NAMES
e7932b12c399   rancher/rancher   "entrypoint.sh"   20 hours ago   Up 20 ho
urs   0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443
/tcp   relaxed_chatterjee
```

2. Reset the password.

```
# docker exec -ti e7932b12c399 reset-password
New password for default admin user (user-ws6vx):
OfBZdkPvc6WKdyCL2ldq
```

import the node host1:

1. Access the Rancher web interface and log in: The default username is admin.

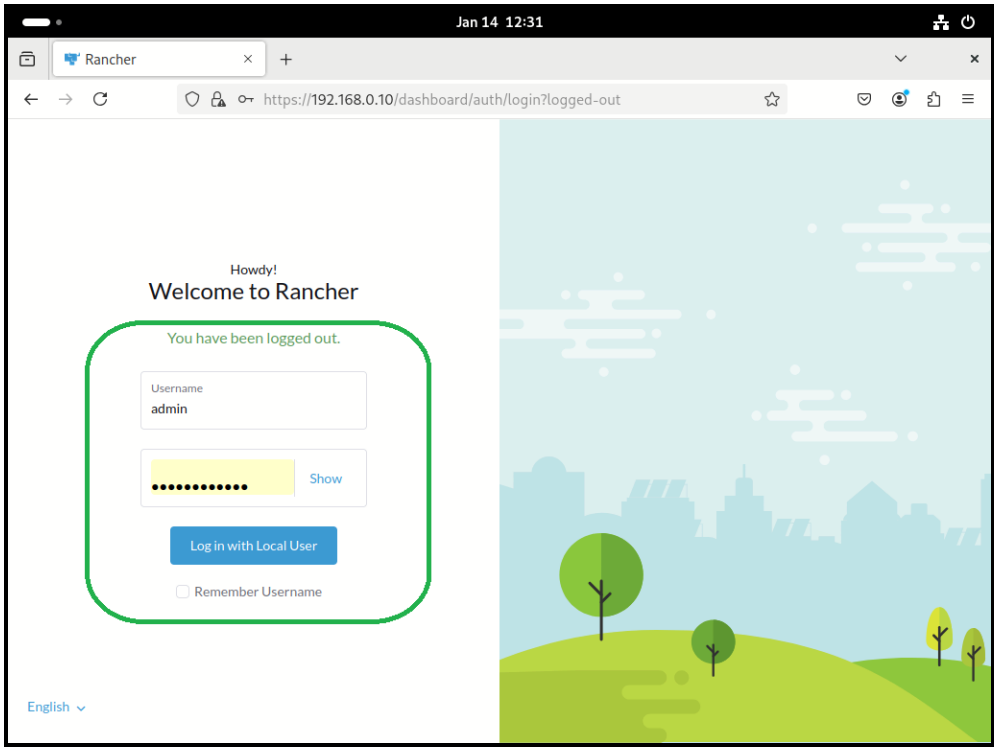

Figure 10: Login Rancher web interface

2. During login, import the SE360 V2 and follow the prompts to enter the information. At this point, the added node, with a cluster name as hakuba.

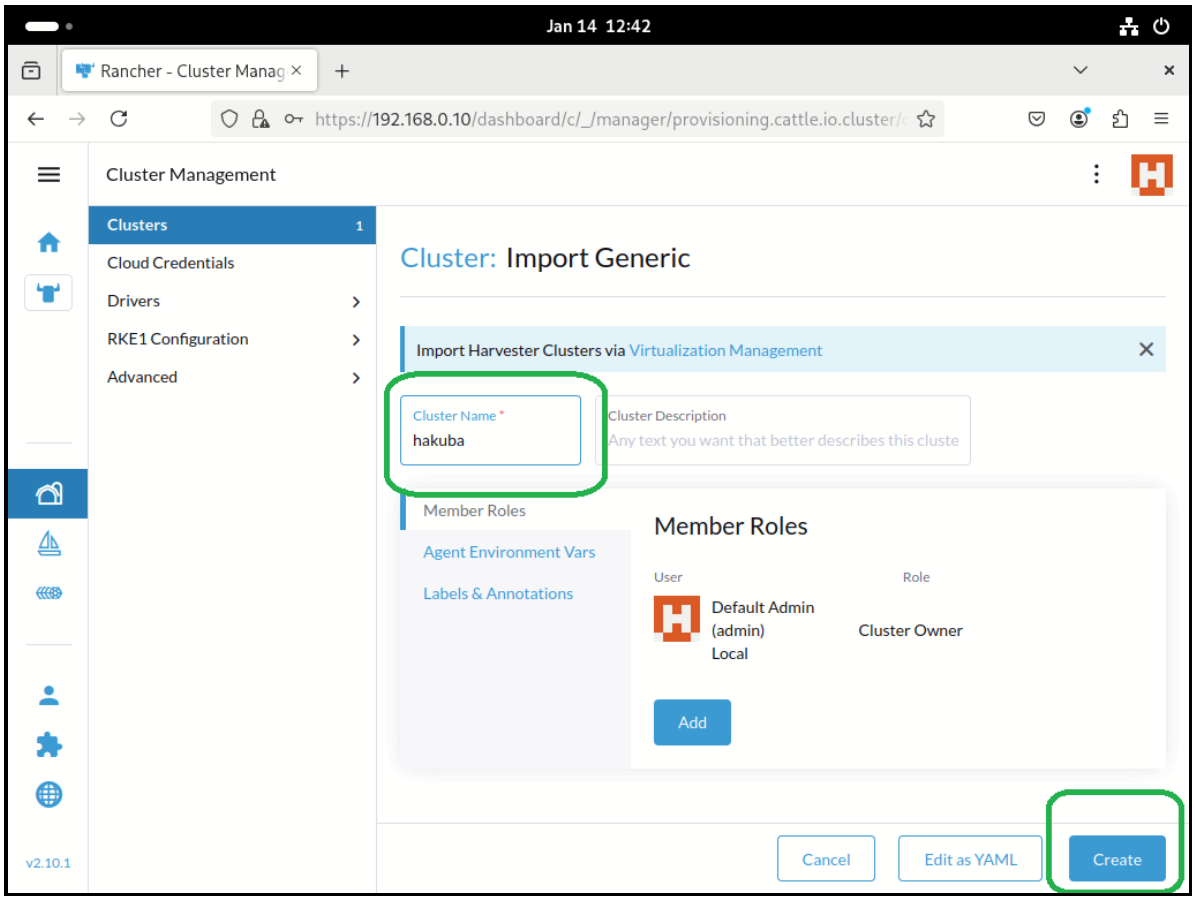

Figure 11: Import one node into Rancher web interface

3. Execute the commands sequentially on host1 as instructed by Rancher.

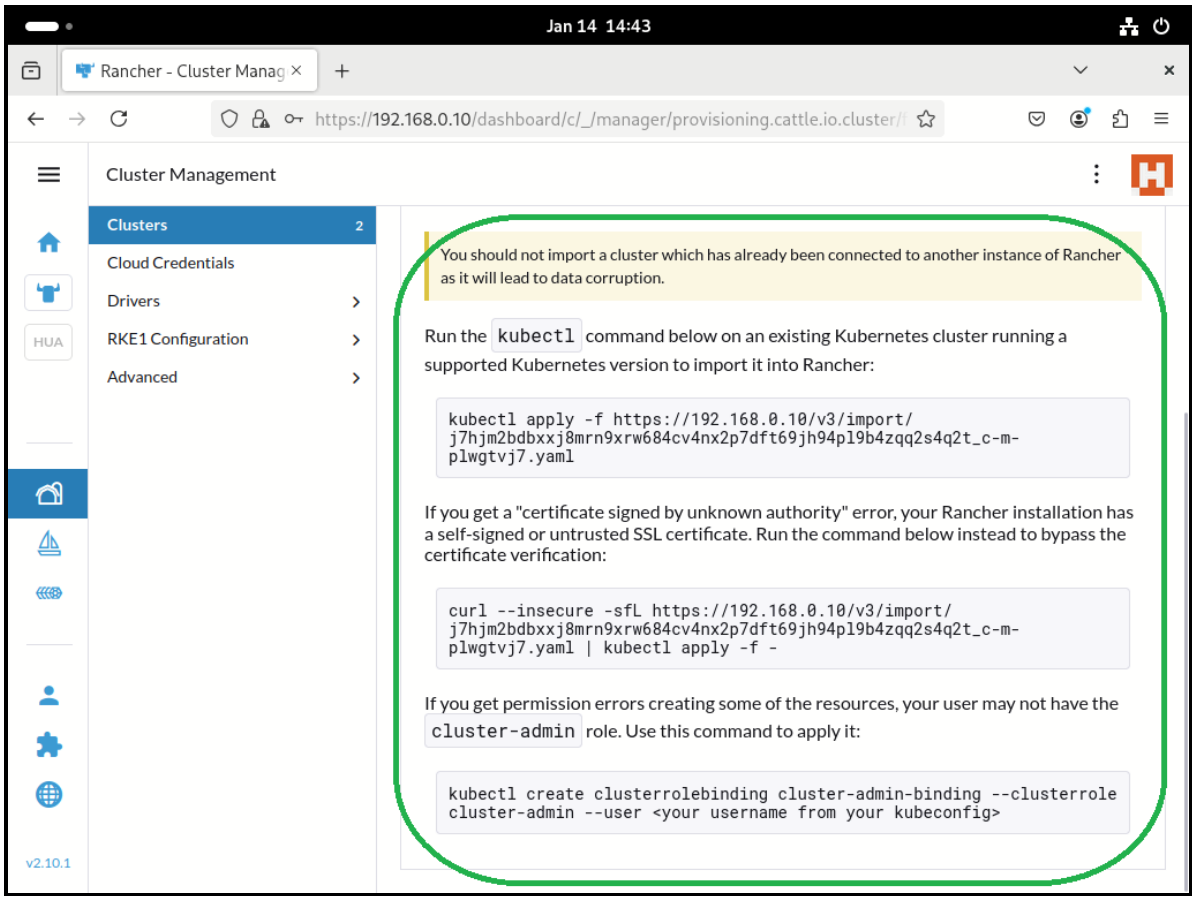

Figure 12: The commands as instructed by Rancher

4. Move to the node that needs to be added and follow Rancher's instructions to execute the commands.

```
# curl --insecure -sfL https://192.168.0.10/V3/import/j7hjm2bdbxxj8mrn9xrw
684cv4nx2p7dft69jh94pl9b4zqq2s4q2t_c-m-plwgtvj7.yaml | kubectl apply -f -
clusterrole.rbac.authorization.k8s.io/proxy-clusterrole-kubeapiserver unch
anged
clusterrolebinding.rbac.authorization.k8s.io/proxy-role-binding-kubernetes
-master unchanged
namespace/cattle-system created
serviceaccount/cattle created
clusterrolebinding.rbac.authorization.k8s.io/cattle-admin-binding created
secret/cattle-credentials-e97d362 created
clusterrole.rbac.authorization.k8s.io/cattle-admin created
Warning: spec.template.spec.affinity.nodeAffinity.requiredDuringScheduling
IgnoredDuringExecution.nodeSelectorTerms[0].matchExpressions[0].key: beta.
kubernetes.io/os is deprecated since v1.14; use "kubernetes.io/os" instead
deployment.apps/cattle-cluster-agent created
service/cattle-cluster-agent created
```

5. Moving to Rancher confirms that the node has been successfully imported. Then you can use Rancher to manage the node's resources.

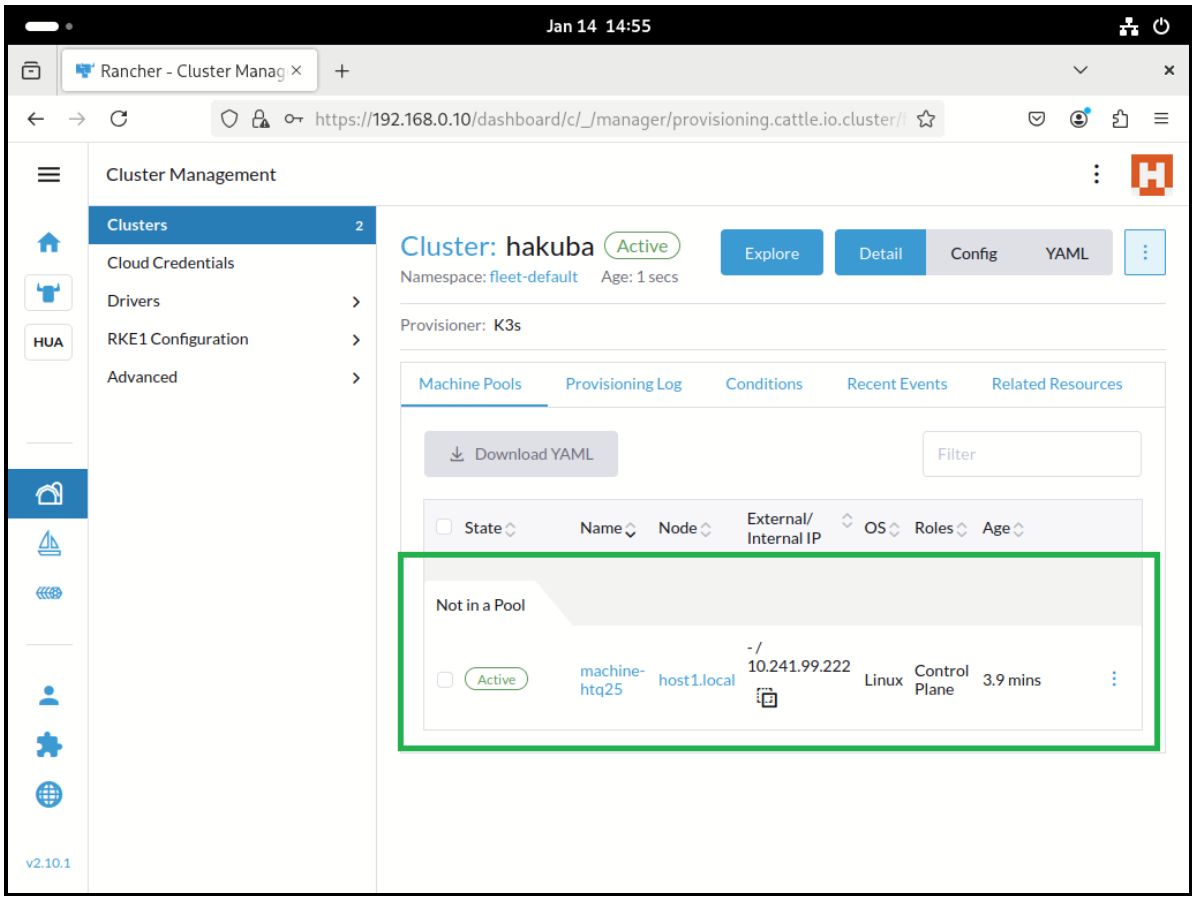

Figure 13: The system information in Rancher

References

For more information, see these resources:

- Lenovo XClarity Controller
  https://pubs.lenovo.com/lxcc-overview/
- Lenovo ThinkEdge SE360 V2 Server
  https://lenovopress.lenovo.com/lp1677-thinkedge-se360-V2-server
- SUSE Edge Documentation
  https://documentation.suse.com/suse-edge/3.1/html/edge/index.html
- The upstream building images guide
  https://github.com/suse-edge/edge-image-builder/blob/release-1.1/docs/building-images.md
- SUSE Rancher
  https://www.suse.com/products/rancher/

Author

Conie Chang is a Linux Engineer in the Lenovo Infrastructure Solutions Group, based in Taipei, Taiwan. She has experience in Red Hat and SUSE Linux OS.

Thanks to the following people for their assistance:

- Adrian Huang, Senior Linux Kernel Engineer
- David Watts, Lenovo Press

Related product families

Product families related to this document are the following:

- Edge Servers
- SUSE Linux Enterprise Server
- ThinkEdge SE360 V2 Server

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service. Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
8001 Development Drive
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary. Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk. Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document, LP2155, was created or updated on February 23, 2025.

Send us your comments in one of the following ways:

- Use the online Contact us review form found at:
  https://lenovopress.lenovo.com/LP2155
- Send your comments in an e-mail to:
  comments@lenovopress.com

This document is available online at https://lenovopress.lenovo.com/LP2155.

Trademarks

Lenovo and the Lenovo logo are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. A current list of Lenovo trademarks is available on the Web at https://www.lenovo.com/us/en/legal/copytrade/.

The following terms are trademarks of Lenovo in the United States, other countries, or both:
Lenovo®
ThinkEdge®
XClarity®

The following terms are trademarks of other companies:

Linux® is the trademark of Linus Torvalds in the U.S. and other countries.

Other company, product, or service names may be trademarks or service marks of others.