Lenovo Validated Design: Deploying Workloads at the Edge Using SUSE Edge 3.3

Last update: **30 September 2025**

Version 1.0

Ease of deployment using SUSE Edge 3.3 and ThinkEdge servers

Provides a replicable deployment and management methodology using SUSE Elemental

Streamline deployment using Rancher Prime Continuous Delivery

Lenovo ThinkEdge servers provide data protection and security

Ameaza Rodrigues

Billzheng Sun

Alex Arnoldy

LENOVO PRESS

Table of Contents

1 Introduction

As organizations increasingly shift toward distributed computing models, the need for scalable, efficient, and secure deployment of infrastructure at the edge has become critical. Edge environments demand solutions that minimize manual intervention, accelerate time-to-value, and ensure consistent operations across diverse and often remote locations.

This document outlines the streamlined approach to deploying Kubernetes workloads in Edge environments at scale using SUSE Edge 3.3 on Lenovo ThinkEdge servers. Leveraging SUSE's robust platform, we will explore the advantages of zero-touch server, O/S, and application deployment, which eliminates complex on-site, manual configuration and decouples the steps of deployment. This design also allows friction-free logistics in delivering and replacing Edge server hardware. The strengths of this solution enable organizations to rapidly and efficiently provision their edge infrastructure on Day Zero, as well as manage through Day One operations and beyond. We will also explore the security features of Lenovo ThinkEdge servers (SE350 V2 and SE360 V2) that guarantee smooth operations and data protection at the remote locations.

We will also delve into the architectural components of SUSE Edge 3.3 that facilitate automated deployment, centralized management, and secure operation, empowering businesses to deploy and maintain distributed Kubernetes environments and applications with unprecedented ease. This guide will showcase how SUSE Edge 3.3 simplifies the complexities of edge computing, providing a reliable and scalable foundation for your critical edge applications.

Background

The Edge computing market is exploding. The key to the massive increase in popularity of Edge computing is that data can be collected and processed at the source; reducing latency and providing real-time insights – as compared to processing all collected data in the cloud. Gartner predicts that by the end of 2025, over 75% of enterprise-managed data will be processed outside the datacenter or cloud, with the edge computing market growing by over $500 billion by 2030. Organizations are unlocking tremendous value deploying workloads at the Edge.

Depending upon where the edge infrastructure is located, we can segment the Edge landscape into three logical areas: Near, Far and Tiny.

Far Edge

The Edge computing infrastructure that is deployed in a location furthest from the datacenter is called Far Edge. It is on the outermost boundary of the network and closest to the end-users or data sources. Examples of Far Edge include:

- Commercial sector: Retail (shopping malls), Hospitality (hotels), Banking (local branch offices), Education (schools), Healthcare (hospitals)
- Industrial sector: Agriculture, Oil and Gas (drilling location), Manufacturing (factory floor),
- Transportation (aircraft, trains), Energy (wind turbines), Utilities (electricity, water facilities)

Near Edge

Edge computing infrastructure that is in-between the datacenter and the Far Edge is called Near Edge. It is *nearest* to the centralized services, and therefore it is called Near edge. For example, Cell tower-based compute, Telecom Central Offices, and Campus compute facilities.

Tiny Edge

The fixed-function devices like microcontroller enabled sensors, actuators, fixed function devices, etc. represent the Tiny edge. These devices are typically within the same network as a Far edge service making it a subsegment within the Far Edge defined above. Note that in the Far edge, the IP space and infrastructure are typically owned, operated and managed by the end-user organization. Whereas the Tiny edge is the set of devices running with this space.

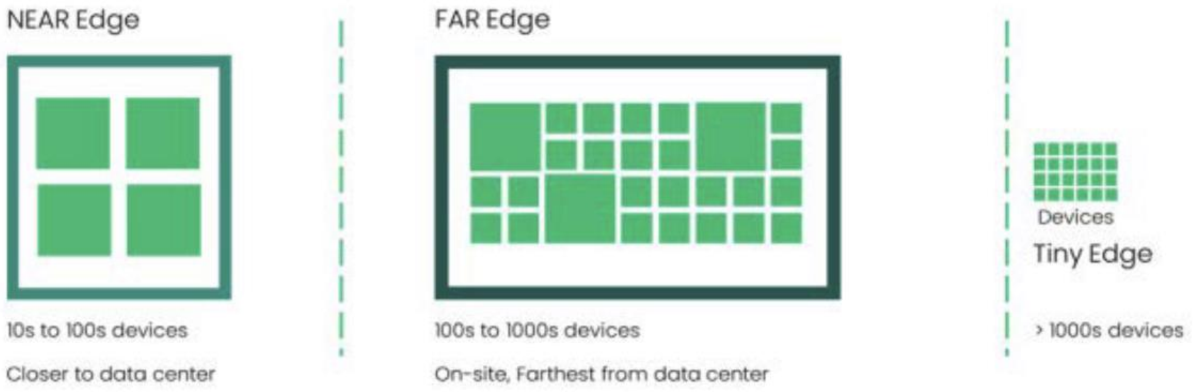


Figure 1: Edge Computing Categories

For Near Edge computing, computing resources (servers, storage and network equipment) are typically deployed in either regional datacenter or on-premises IT rooms. Given that the deployment environment of Near Edge is like a traditional datacenter, the current practices of management and deployment of IT infrastructure for datacenters can be leveraged for Near Edge deployment.

However, in Far Edge deployment, computing infrastructure is deployed outside the datacenter, which lacks necessary facilities and personnels with the required IT skills on-site to manage the IT infrastructure. This shift in deployment practices brings lots of challenges to IT managers and administrators, especially on security, as it removes edge computing resources from the physical access and network security protection provided by a datacenter.

In this document, we will describe the major challenges faced by IT administrators for Edge deployments, discuss features in Lenovo's next generation of ThinkEdge servers and SUSE Edge 3.3 infrastructure software. We will address the security concerns in Edge deployments with SUSE Security. We will deploy Edge applications at any scale and automatically remedy application software and configuration drift with Rancher Prime Continuous Delivery. Finally, we'll show how SUSE Elemental allows for friction-free logistics for deploying and maintaining Lenovo ThinkEdge servers across thousands of Edge locations.

The target audience for this Lenovo Validated Design (LVD) is Edge solution architects. Some experience with Linux and Kubernetes may be helpful, but it is not required.

Motivation

In today's distributed computing landscape, the need for resilient workloads, robust security, and simplified management at the edge is paramount. Traditional cloud-centric models often struggle to meet the demands of latency-sensitive applications and geographically dispersed operations. This necessitates a shift towards edge computing, where processing occurs closer to the source of data. However, deploying and maintaining edge infrastructure presents unique challenges. This document addresses these challenges by focusing on three key areas:

- **Resilient Workloads:** Ensuring continuous operation of critical applications at the edge, even in the face of network disruptions or hardware failures. SUSE Edge 3.3 provides the tools and technologies to build and maintain highly available application deployments.
- **Security Needs at the Edge:** Securing edge locations, which are often physically vulnerable and may lack the same level of protection as traditional data centers. SUSE Edge 3.3 incorporates security best practices and features to protect edge devices and data. Complementing this, Lenovo Edge servers bring an additional layer of protection with their hardened hardware design and built-in security features.
- **Low/Zero-Touch Deployment and Management:** Simplifying the complexities of deploying and managing edge infrastructure at scale. SUSE Edge 3.3 enables zero-touch provisioning, centralized management, and automated updates for infrastructure and applications, significantly reducing operational overhead.

By addressing these critical aspects, SUSE Edge 3.3 empowers organizations to confidently deploy and manage Edge Kubernetes workloads, unlocking the full potential of edge computing for a wide range of use cases.

Scope

The scope of this document is to show the ease of deploying multi-node Edge clusters at scale. While SUSE Edge can deploy cloud-native clusters through a variety of methodologies, this document seeks to show the easiest method, both in terms of on-site labor as well as in the logistical supply chain. To meet this goal, we will focus on operations occurring after the management infrastructure is in place. Only a minimal description of the management infrastructure components will be provided.

2 Business Challenges and Key Benefits

Deploying business applications at the Edge provide very specific challenges that aren't encountered in datacenter and cloud operations. These challenges include:

Lack of technical expertise

As it's simply not possible to have expertise available for every technology in use at every Edge location, specialized tooling is required to deploy and maintain hardware and software infrastructure as well as the business applications. The goal should be to create a reliable means for maintaining Edge locations with no more technical expertise on-site than is required to operate a modern mobile phone.

Logistical supply chains

Relative to the simple logistics of maintaining datacenter equipment, maintaining equipment at hundreds or thousands of Edge locations can very easily become prohibitively complex and expensive. Each specialized step in this operation equates to a significant increase in complexity and cost. The goal at this layer is to create as homogeneous a design as possible. From cable length to the process for installing base software, bespoke components and processes can sink a large-scale Edge design.

Security

Most of the compute infrastructure used today was originally designed with limited consideration for physical security. Because of this, traditional datacenters must use multi-layered and multi-modal security standards to protect against physical attacks against the compute assets contained within. However, most Edge locations can provide only the most minimal in terms of physical security of the location's compute assets.

To make up for this, Edge hardware and software must be designed under the expectation that direct, physical attacks against at least one location is guaranteed. It is critical to implement hardware layer security measures as well as advanced software attack detection/isolation capabilities and application configuration drift detection and remediation. Additionally, a zero-trust security model should be implemented for both applications and the underlying software infrastructure.

Maximize compute resources

In edge computing, the efficiency of resource utilization is paramount. A small amount of wasted compute resources, multiplied across thousands of edge locations, can quickly escalate into a crippling cost drain on the entire project. To maximize efficiency, it is crucial to enable the same compute resources to support a variety of applications for different purposes. This approach avoids deploying unique compute platforms for each application, which would otherwise lead to significant waste, especially during periods of low utilization. By optimizing resource allocation and minimizing waste, the economic viability and operational effectiveness of edge computing initiatives can be greatly enhanced.

Connectivity

Typically, only the outbound network is available at Edge sites. It may not be easy or even possible for IT admin to connect to Edge servers in an Edge site remotely from corporate IT locations as they are often behind NAT or firewall. Network connection from Edge site to public cloud is not considered to be reliable.

3 Architectural Overview

The core of this design is deployed at the corporate operations center. It is centered around SUSE Rancher Prime with two key components: Elemental and Rancher Continuous Delivery.

Management Cluster

Elemental provides the bulk of the capabilities for onboarding downstream edge nodes with an operating system as well as installing and configuring the Kubernetes cluster software. Elemental is also the component that will upgrade the operating system and Kubernetes software as updates become available. Elemental is powerful in its capabilities as well in that it can be managed via command-line, Web User Interface, and/or through API calls. This makes it ideal for virtually any size organization and edge project.

SUSE Rancher Prime Continuous Delivery (CD) is a continuous delivery engine based on the open-source project Fleet. It leverages application configurations maintained in any git-compatible repository. Rancher Prime CD will automatically deploy applications to downstream Kubernetes clusters based on labels that are applied to those clusters. Not only does this allow applications to be maintained and updated en masse, but it allows new value to be uncovered by deploying applications both where and when they are needed. Hidden ROI can be captured by running applications after hours on systems that would otherwise only be utilized during business hours.

Downstream Edge Clusters

Due to the extensive capabilities built into the Management Cluster, the downstream edge clusters are free to completely focus on running the applications that run the business. Due to full application portability, applications are deployed by SUSE C/D based on the business need at the location and time. This means, for example Point of Sale systems run the required software during business hours but can then be utilized as compute platforms for after-hours applications such as in-store video analysis and inventory reconciliation.

Edge clusters consist of one or more Lenovo server nodes (SE350 V2 and SE360 V2). Single-node and two-node clusters are cost-effective ways of providing a specific quantity of compute power at a particular location but are not highly available. Three-or-more cluster nodes create a highly available compute platform.

All cluster nodes are installed with the same, Elemental-generated SUSE Linux Micro operating system. Therefore, after similarly equipped nodes have been prepared, they can be shipped to any edge location, providing a friction-free logistics experience. This stands in contrast to the tightly coupled logistics of other edge architectures where servers are assigned, early in the build process, for specific roles and/or for specific edge locations. This type of tight coupling of hardware to destination and role leaves the success of the project vulnerable to supply-chain and logistics problems.

The auto-installing SUSE Linux Micro (SL Micro) operating system can be installed, or re-installed, with no special equipment or technical skills. Thus, the nodes can be installed at the Lenovo factory, the customer’s designated central location, or at an edge location. Standardized installation processes can be augmented on an ad-hoc basis when needed to resolve unexpected issues.

The installation of the SL Micro operating system also causes the node to register itself with the Elemental service running on SUSE Rancher Prime. This registration adds the node to the Elemental inventory of unallocated nodes. Again, un-installed servers can be shipped to edge locations to be installed on-site;

however, in our example scenario, the SL Micro operating system is installed at the Lenovo factory, then shipped to any edge location that needs them.

Once the edge nodes arrive at the edge location, the only technical skills required to implement them are the ability to mount them and plug in the network and power cables. When the node powers on, it announces to SUSE Elemental that it is available to be used in the Kubernetes cluster assigned to that location. Changing the node’s label can be done in several ways but is performed from the central SUSE Elemental interface.

Very often, complex deployments consist of operations that are interrelated and tightly coupled. This can lead to delays and cost overruns during implementation. For example, the application experts must wait until the orchestration experts have completed their work. By the same token, the orchestration experts must wait until the hardware infrastructure has been installed. Obviously, no technical work can begin until all the edge and networking hardware is available at the edge location. Scaling these challenges to hundreds or thousands of edge locations can quickly make a project unfeasible.

A huge advantage of SUSE Elemental is that it intelligently decouples the primary phases of an edge location implementation. The application, orchestration, and operating system experts are free to complete their work independently, and at any time; even before the hardware (or even the edge location itself) has been built!

The key to this decoupling is using Kubernetes labels on each resource to identify where it is and what it should be doing. For example, labeling compute nodes tells SUSE Elemental where they are located, and which Kubernetes cluster should be installed on them.

Labeling a downstream cluster under Rancher Prime Continuous Delivery determines what applications will be deployed to the cluster. Changing that label immediately causes the currently running applications to be removed and different applications to be put into service. This can be done manually or automatically. Automatic application transitions can occur based on things such as the time of the day (e.g. normal business hours versus after hours), activity of other applications (e.g. replace instances of a lower priority application due to a surge in activity of a higher priority one), or based on the input of an AI agent (e.g. open points of sale closer to where customers are in a store).

Example SUSE Edge Installation

For this design, a central management cluster contains all the vital functions of Edge Kubernetes cluster management (SUSE Rancher Prime), Edge node operating system onboarding and management (SUSE Elemental), and application deployment, updating/rollback, and removal (Rancher Prime Continuous Delivery). The specifications of this management cluster vary based on the needs of the project; thus, the design and deployment of this cluster are not covered in this paper.

The focus of this paper is to show how SUSE Edge decouples the stages of the platform and application deployment to ensure new downstream clusters and applications can be deployed on-time and within budget, even in the face of multitudes of unexpected setbacks. As well, it will show how the SUSE Edge automation, through Elemental and Continuous Delivery, significantly reduces the amount of work and technical debt required to deploy multiple single-node downstream Edge clusters, and a multi-node Edge cluster. While the options for automation will be called out, for uniformity the procedures will be performed using the SUSE Rancher Prime user interface.

4 Enhanced Security at the Edge

ThinkEdge Server Security Features

The ThinkEdge SE350 V2 and SE360 V2 servers are ideal for scenarios where application and data security are paramount. They are shipped with following security features:

Data Protection and Self Encrypted Drives

The key focus of ThinkEdge security is data protection. There are many potential threats to data that are unique to edge environments. Attackers could steal the whole ThinkEdge server or just pull out the disk drive in the front panel. To protect user's data, ThinkEdge servers are equipped with Self-Encrypting Drives (SED). The SED drive has encryption hardware built into the drive controller, which will automatically encrypt all data as it is written to the drive and decrypt all data as it is read from the drive. Data stored on SEDs are always fully encrypted by a data encryption key (DEK), which is stored on the drive's hardware and cannot be accessed by the host operating system or unauthorized users. Because SEDs use hardware-based full disk encryption, both the encryption and decryption processes occur in the disk hardware. This separation from the host operating system makes hardware encryption more secure than software encryption. Moreover, unlike software encryption, hardware encryption does not require extra CPU resources. If a SED is physically stolen or lost, it becomes practically impossible to obtain intelligible information from the SED.

While SED drives use Data Encryption Key (DEK) to encrypt data in disk drive, Authentication Key (AK) is used to unlock the drive and manage the access to DEK in the SED drive. ThinkEdge servers carefully protect the SED AK by storing it inside a secure processor. The ThinkEdge servers only allow access to the SED AK after the system is properly authorized. Once the system is authorized, SED AK unlocks the access to DEK in SED drives, so encrypted data in SED drive can be decrypted by DEK for access. If an attacker steals the SED drive, given that the data in the SED drive is encrypted, the attacker cannot read the content in the drive.

For ThinkEdge SE350 V2 and SE360 V2, SED Encryption needs to be enabled to encrypt the data stored in the SED drives. SED Encryption is offered as a CTO (configure to order) option, or it can be enabled later in the XClarity Controller. SED encryption can't be disabled after it is enabled in manufactory or in XClarity Controller. If SED Encryption is not enabled, data will not be encrypted in SED Drive.

System Lockdown and activation

If security sensors (Chassis Intrusion Detection or motion detection) are enabled in XClarity Controller, ThinkEdge server will be locked down when tamper events (open top cover, unexpected movement of the server) are detected. When the ThinkEdge server is locked down, the system can't be rebooted. Access to SED AK stored in the security chip is blocked, therefore user data stored in the SED drive can't be accessed even if the whole system is stolen.

Users can also tell whether the server is in lockdown mode from the Activation LED in the server's front panel. If the Activation LED is blinking, the server is in a locked down model and needs to be activated.

Under Additional Configuration in System Lockdown Mode, users can choose whether Host OS will be shutdown with system lockdown. This option is enabled by default. Uncheck this option if Host OS needs to be kept running when the system is locked down. If this option is not enabled, Host OS will not be shut down when the system is locked down in the event of tamper detection, but the system can't be rebooted.

Activation of Server in Lockdown mode

When the ThinkEdge server is locked down in the event of tamper detection, the user needs to activate the server to allow the system to reboot and restore the access to encrypted data stored in SED drives. ThinkEdge servers are shipped with two options in terms of how the system is managed in lockdown mode, which are defined by System Lockdown Control in CTO configuration in https://dcsc.lenovo.com/ (Data Center Solution Configurator) when the system is ordered.

The ThinkEdge server ordered with Feature Code BYBR (System Lockdown Control is managed by ThinkShield Key Vault Portal) will be shipped in lockdown mode from manufacturing. When the server arrives at a deployment site, it will not be powered on and complete the Power On Self-Test until it has been successfully activated with ThinkShield Key Vault Portal.

5 Secure Hardening Recommendation

In the previous chapters, we discussed security features of Lenovo ThinkEdge servers (ThinkEdge SE350 V2 and SE360 V2). In this chapter, we will give recommendations on security hardening of the server and firmware (UEFI and XClarity Controller for system management).

ThinkEdge server – Hardening UEFI

UEFI configuration can be done by pressing the F1 key after the system is rebooted. The LXCE OneCLI command line tool can also be used to configure UEFI when applicable.

Enable Secure Boot

Secure boot is functionality built into UEFI's specification. When Secure Boot is enabled and properly configured, it protects computers against attacks and infections from malware that installs rootkits and boot kits.

Secure Boot detects when software like the boot loader and key operating system files and other things like option ROMs have been tampered with. It does this by validating each component's digital signature. Any component whose digital signature verification fails is not loaded during the boot process. Depending upon the OS and drivers you are using on the server it may not always be possible to enable secure boot.

Configure the Trusted Platform Module

Onboard Trusted Platform Module (TPM) is a component of most modern computer systems. It is classified as a secure crypto processor, to enable advanced cryptographic functionality in the operating system and applications. TPM 2.0 is supported in ThinkEdge SE350 V2 and SE360 V2.

Set Boot Mode to UEFI

Boot Mode determines which mode the system used to boot. Setting boot mode to UEFI is the most secure value for Boot Mode. When set to UEFI the system runs UEFI drivers and boots a UEFI OS loader.

Review and update boot order

The boot order determines the order the system searches for bootable media as part of the boot process. The system follows the order specified until it finds a device that is bootable. Once it does it boots the system from that device.

Remove any unnecessary boot options that are not required. Systems normally will contain a network boot option such as PXE boot or HTTPS boot as part of the boot order. Network boot is typically used for initial deployment of the host operating system. After initial deployment of the host operating system, network boot options should be removed from the boot order.

Review and remove unnecessary boot options

Verify the boot options and that all are required and remove the boot options that are not necessary. Carefully consider removing USB Storage from BootOrder if you do not need to boot from a USB device.

Review and update boot priority for each device type

Verify the priority/order for each type of boot device to ensure the correct device of each type receives the highest priority.

Disable booting using the network stack

The Network Stack controls whether the system uses any network option for booting. If you do not require any network boot setting, then disable the entire network stack. This is not only the most secure setting it also helps speed up the boot process.

Disable PXE Boot

PXE boot allows a system to boot from a server on a network that supports PXE booting instead of from a local hard drive. If you are not able to disable booting using the network stack, disable PXE boot if you do not need it.

Disable IPMI over Keyboard Controller Style (KCS) Access

The IPMI over KCS channel allows a host user, who is an administrator user on the host, full access to the IPMI commands supported by XCC without any form of XCC authentication. If you are not running any tools or applications on the server that access the XClarity Controller through the IPMI protocol, it is highly recommended that you disable the IPMI-over-KCS access for improved security of XCC.

HTTPS boot allows a system to boot from a server on a network that supports HTTPS booting instead of from a local hard drive. If you are not able to disable booting using the network stack, disable HTTPS boot if you do not need it.

Set an administrator password

Setting an administrator password deters unauthorized users from changing configuration settings. When an administrator password is set, you are prompted to enter a valid password each time you try to access the Setup Utility program. The Setup Utility program cannot be accessed until a valid password is entered.

Given that ThinkEdge servers are deployed outside a data center, it is at risk of being accessed by none authorized person. It is very important to set an administrator password for UEFI, for ThinkEdge servers shipped with Feature Code BYBQ (System Lockdown Control is managed by XClarity Controller).

For the ThinkEdge server shipped with Feature Code BYBQ, when it is locked down in the event of tamper detection, it can be activated in the XClarity Controller Web Console. If the UEFI administrator password is not set, the attacker can easily go into the UEFI Setup Utility and change the password of XClarity Controller to factory default, thus obtaining the access to XClarity Controller to activate the server.

If both the power-on password and administrator password are set, you can enter either password. However, you must use your administrator password to change any configuration settings.

ThinkEdge server – Hardening XClarity Controller (XCC)

The XCC hardening secures the out-of-band management interface that controls server monitoring, remote access and firmware updates.

Harden the network settings of XCC

Do not connect the Lenovo XClarity Controller (XCC) network interface to untrusted networks. Restricting XCC network access to only trusted networks reduces its attack surface and makes it more difficult for attackers to exploit any weaknesses it might have.

Configure IPv4

Select the proper method for determining the IPv4 address of the XCC interface. For example, if you do not wish the IPv4 address to be obtained from a DHCP server then do not select that option.

Configure IPv6

Select the proper Method for determining the IPv6 address of the XCC interface. If your organization does not use IPv6 then you should disable it. If you are using IPv6, then select the address configuration method used by your organization.

Disable or configure Ethernet over USB

Ethernet over USB is used for in-band communication between the host server and XCC. This feature provides an in-band channel for applications on the host server to communicate with XCC and vice-versa. This means that a user logged in to the host can use applications on XCC that can communicate over this channel.

To prevent applications that are running on the server from accessing XCC via this interface, you should disable the USB in-band interface. If you disable the USB in-band interface, you cannot perform an in-band update of the XCC firmware, the UEFI firmware, the embedded provisioning tool, and certain adapter firmware by using the XClarity Essentials in-band update utility.

SNMP Configuration

SNMP can be used to manage and monitor XCC using the SNMP protocol. If you enable SNMP, then only enable those items that you need. For example, if you do not need SNMPv1 traps then ensure it is disabled.If you do need to use an SNMP Agent, then enable the SNMPv3 Agent only.

Disable unnecessary services

The following table shows the network services that are available within XCC. To reduce XCC’s attack surface, disable any service that your organization does not require. Certain services are required by XCC and cannot be disabled. Those are noted below.

Block List and Time Restriction

Block Lists and Time Restrictions are used to further restrict access to XCC. Use the Block List settings and configure the IPs and MAC addresses of those systems that do not need access to XCC. Use the Time Restrictions and configure the times that XCC cannot be accessed.

Configure XCC SSL certificate

The SSL certificate is the server certificate used by the XCC WebUI, the Redfish Service and the CIM

Service. By default, XCC will generate a self-signed certificate for the server. Self-signed certificates typically cause errors or warnings in browsers that the server cannot be trusted.

When possible, use a valid CA signed certificate as it is preferred and is considered a security best practice. To do this, use XCC to generate a certificate signing request, get it signed by the CA of your choosing and then upload the CA signed certificate into XCC.

Disable IPMI over Keyboard Controller Style (KCS) Access

The IPMI over KCS channel allows a host user, who is an administrator user on the host, full access to the IPMI commands supported by XCC without any form of XCC authentication. If you are not running any tools or applications on the server that access the XClarity Controller through the IPMI protocol, it is highly recommended that you disable the IPMI-over-KCS access for improved security of XCC.

Port	Service	When to disable	How to disable in XCC	OneCLI setting
427	SLP	Used by Lenovo XClarity Administrator (LXCA) and other Lenovo tools to discover devices on the network. If you do not use LXCA or another Lenovo tool that uses SLP, disable this protocol	BMC Configuration → Network → Service Enablement	IMM.SLPPortControl "Closed"
546	DHCPv6 Client	Disabled if you do not use DHCPv6 for IPv6 Interface Configuration	BMC Configuration → Network → Ethernet Configuration	IMM.IPv6DHCP1 "Disabled"
623	IPMI over LAN	Disable IPMI over LAN if you are not using any tools or applications that access the XClarity Controller through the network using the IPMI protocol.	BMC Configuration → Network → Service Enablement	N/A
1900	SSDP	Used by Lenovo XClarity Administrator (LXCA) and other Lenovo tools to discover devices on the network. If you do not use LXCA or another Lenovo tool that needs SSDP,	Enablement	N/A

		disable this protocol		
5989	DHCPv6 Client	Used by LXCA and other Lenovo tools to configure the server. If you do not use LXCA or another Lenovo tool that needs CIM-ov		IMM.CIMXMLOverHTTPS_E nable "Disabled" IMM.CIMOverHttpsPortControl "Closed”

Table 1: Network Services Available through XCC

When using XClarity Essentials Utility tools on the host, the IPMI-over-KCS interface to the XClarity Controller is required. It is still recommended that you disable the IPMI-over-KCS interface and only re-enable it when you need to use XClarity Essentials Utility tools on the host then disable it after you’ve finished using XClarity Essentials.

Disable Bluetooth Button on Front Panel

ThinkEge SE360 V2 server can be activated with ThinkShield Mobile App with bluetooth connection. And there's a button on the front panel where users can press to turn on / off support for Bluetooth connection. Disable this setting if you’re not going to use ThinkShield Mobile App to activate the server with Bluetooth connection.

Prevent System Firmware Down-Level

The Enable Prevent System Firmware Down-level option prevents all system firmware including XCC, UEFI, and LXPM from being downgraded to an older revision.

Enabling this setting prevents an attacker from installing a previous version of firmware that contains known vulnerabilities that can then be exploited. This option should be enabled unless for some reason there is an organizational requirement to allow firmware to be downgraded to an older version.

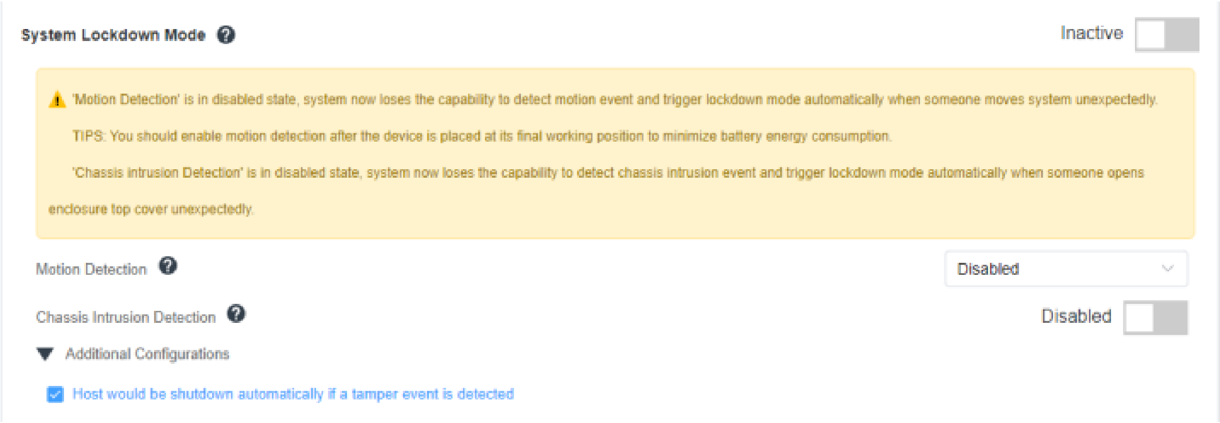
System Lockdown Mode

ThinkEdge server is shipped with security sensors for tamper detection. The ThinkEdge server will be locked down in the event of tamper detection if security sensors are enabled, which are configured under System Lockdown Mode in XClarity Controller Web Console.

There are two type of security sensors:

- Motion detection: for unexpected movement of the server
- Chassis intrusion detection: trigger the system lockdown when the top cover is opened

It’s highly recommended to enable these two options for Edge server deployed in Edge site. Another configuration option under System Lockdown Mode is to determine whether Host OS will be shutdown as part of system lockdown. It is recommended to leave this option as default.



System Lockdown Mode ? Inactive

⚠ 'Motion Detection' is in disabled state, system now loses the capability to detect motion event and trigger lockdown mode automatically when someone moves system unexpectedly.

TIPS: You should enable motion detection after the device is placed at its final working position to minimize battery energy consumption.

'Chassis intrusion Detection' is in disabled state, system now loses the capability to detect chassis intrusion event and trigger lockdown mode automatically when someone opens enclosure top cover unexpectedly.

Motion Detection ? Disabled

Chassis Intrusion Detection ? Disabled

▼ Additional Configurations

☑ Host would be shutdown automatically if a tamper event is detected

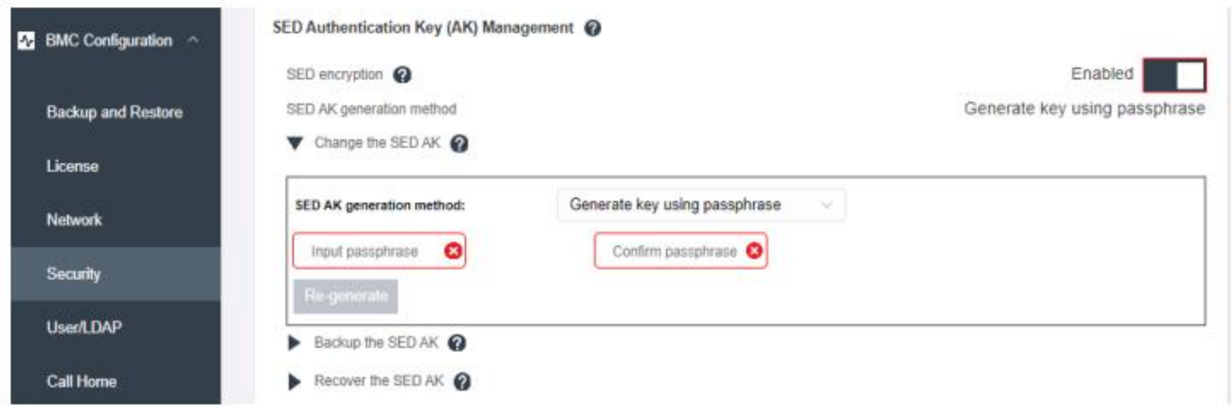
Figure 2: Tamper detection and the System lockdown Mode

SED AK Management

As discussed in previous chapters, ThinkEdge SE350 V2 and SE360 V2 are shipped with SED drives. But data will not be encrypted unless SED Encryption is enabled.

When SED Encryption is enabled, SED AK (Authentication Key) is used to manage the access to DEK (Data Encryption Key) in SED drives. SED AK can be generated from a passphrase or randomly by XClarity Controller. If you want to share the SED drives between multiple ThinkEdge servers, generating SED AK from a passphrase is recommended.

It's highly recommended to backup SED AK when SED encryption is enabled, particularly when the AK is generated randomly. SED AK will be exported to an external file during SED AK backup. In case of system failure, or after replacement of the motherboard, users may need to restore the SED AK to obtain access to the data stored in the SED drive. If SED AK is lost, users will not be able to read the encrypted data in the SED drive.



SED Authentication Key (AK) Management ?

SED encryption ? Enabled

SED AK generation method Generate key using passphrase

▼ Change the SED AK ?

SED AK generation method: Generate key using passphrase

Input passphrase ⊗ Confirm passphrase ⊗

▶ Backup the SED AK ?

▶ Recover the SED AK ?

Figure 3: SED AK Management Dashboard

6 Edge Deployment

A Study in What-if Scenarios

To show the power of SUSE Edge deployment decoupling, we have chosen an imaginary scenario where scheduling problems plague our Edge location deployment. Many other software providers use heavy helpings of Marketing fluff to transform their datacenter products into Edge focused ones. However, no matter the extra shine, they are still rigid and require a strict adherence to the process of installation and deployment. Any hiccups in the timing of hardware, licenses, consultants, and/or administrators will result in significant delays and increased costs.

The Schedule

In a perfect world, our Lenovo SE360v2 cluster nodes would arrive at the Edge location first. After that, the infrastructure consultants would install the SUSE Linux operating system. Then, Kubernetes specialists would install and configure the Kubernetes clusters. Finally, the application specialists would install and configure the cloud-native business applications.

But who needs “PERFECT”?!?! In our imaginary scenario, nothing goes to plan. Our application experts finished their work in record time and were the first to be ready to bring their work to our Edge project. Our Kubernetes experts were delayed and completed their work almost a week later than expected. Worst of all, due to supply-chain issues, our Edge nodes were the last pieces of the puzzle to arrive at our pilot Edge location.

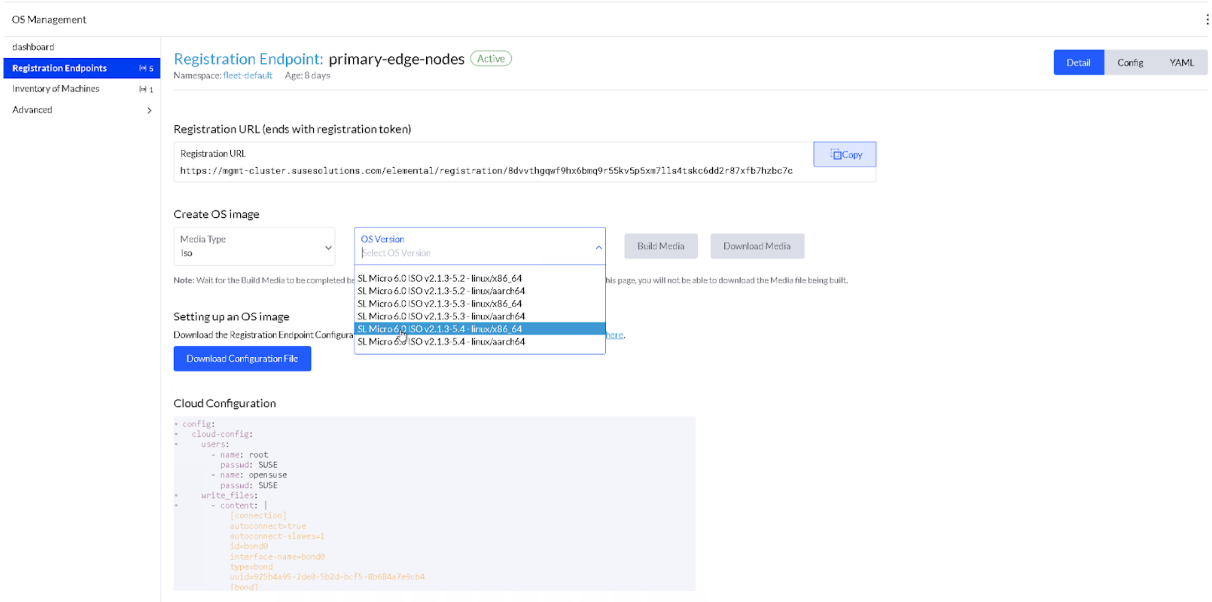
While this level of disruption would throw any other project into disarray and undoubtedly lead to cost overruns, not so with SUSE Edge. Since Elemental allows significant decoupling of deployment steps, we can tolerate unforeseen changes to our plan. Rancher Prime Continuous Delivery allows the tested application definitions and deployment modes to be ready, even if there are no RKE2 or K3s Kubernetes clusters on which to deploy them. As well, Elemental allows defining the exact, production Kubernetes clusters without being dependent on actual nodes to be available at the time.

Creating a Registration Endpoint in OS Management

Deploying an Edge server with Elemental begins with establishing a Registration Endpoint—a configuration that defines how devices will be registered and managed during installation. This is done through the OS Management dashboard, which provides both a user-friendly interface and YAML-based configuration options.

Add Machine Registration Endpoint

In the web interface, a registration endpoint can be configured in the OS Management Dashboard by clicking on the “Create Registration Endpoint” button. The endpoint information can be entered manually in the form or by editing the YAML file, for a bulk or scripted setup.



Registration Endpoint: primary-edge-nodes Active

Namespace: fleet-default Age: 8 days

Registration URL (ends with registration token)

Registration URL
`https://mgmt-cluster.susesolutions.com/elemental/registration/8dvvthgqwf9hx6bmq9r55kv5p5xm7lls4tskc6dd2r87xfb7hzbc7c`

Create OS image

Media Type: Iso

OS Version: Select OS Version

- SL Micro 6.0 ISO v2.1.3-5.2 - linux/x86_64
- SL Micro 6.0 ISO v2.1.3-5.2 - linux/aarch64
- SL Micro 6.0 ISO v2.1.3-5.3 - linux/x86_64
- SL Micro 6.0 ISO v2.1.3-5.3 - linux/aarch64
- SL Micro 6.0 ISO v2.1.3-5.4 - linux/x86_64
- SL Micro 6.0 ISO v2.1.3-5.4 - linux/aarch64

Note: Wait for the Build Media to be completed be his page, you will not be able to download the Media file being built.

Setting up an OS image

Download the Registration Endpoint Configura here.

Cloud Configuration

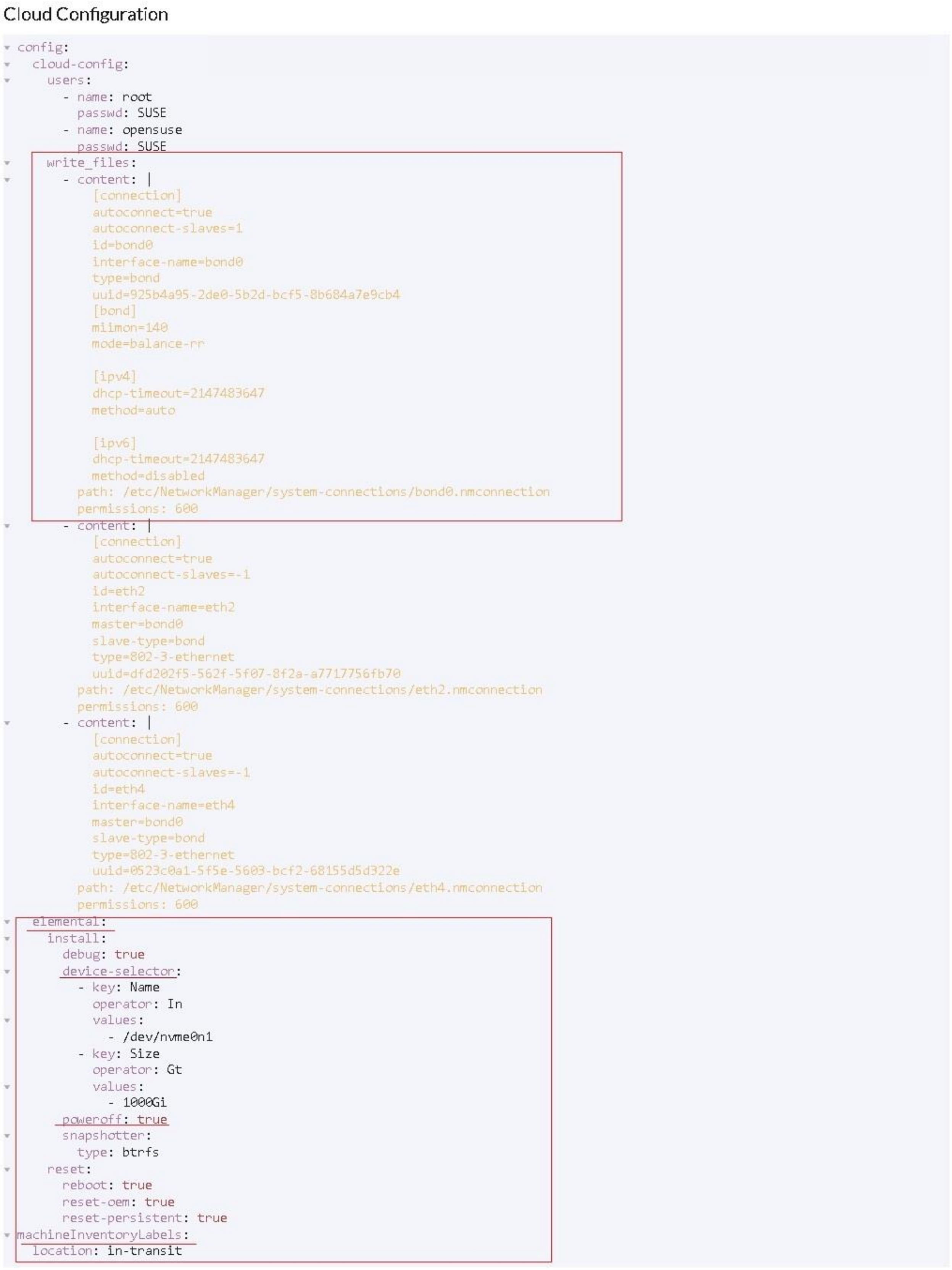
```
config:
  cloud-config:
    users:
      - name: root
        passwd: SUSE
      - name: opensuse
        passwd: SUSE
    write_files:
      - content: |
          [connection]
          autoconnect=true
          autoconnect-slaves=1
          id=bond0
          interface-name=bond0
          type=bond
          uuid=925b4a95-2de0-5b2d-bcf5-8b684a7e9cb4
          [bond]
```

Figure 4: Registration Endpoint Dashboard

Key Cloud Configuration Fields

Here are the main fields that were configured:

- users: This field configures users with their respective usernames and passwords.
- write files: This field can be used for virtually any custom configuration. In our example (Fig. 4), this field is used to configure network interfaces.
- elemental: The device-selector field contains a list of rules to dynamically pick a storage drive during installation. Setting the poweroff field to “true" powers off the node after a successful automated installation. This means the person installing the O/S, at a staging location or at the Edge location, doesn’t have to monitor the installation. Rarely will the installer even attach a keyboard/mouse and monitor to the node. Simply put, when the node powers off, it’s ready for the next step in the process. This helps scalability and significantly lowers the technical debt required for O/S installations.
- machineInventoryLabels: When nodes are booting up for the first time, they connect to Rancher Manager and a Machine Inventory is created for each node. Within the MachineInventory label, the location label for the node will identify where it is in the build and deployment cycle and when it is ready to be installed with an RKE2 Kubernetes cluster. In our example, the field indicates that the node is “in-transit”.



Cloud Configuration

```
config:
  cloud-config:
    users:
      - name: root
        passwd: SUSE
      - name: opensuse
        passwd: SUSE
    write_files:
      - content: |
          [connection]
          autoconnect=true
          autoconnect-slaves=1
          id=bond0
          interface-name=bond0
          type=bond
          uuid=925b4a95-2de0-5b2d-bcf5-8b684a7e9cb4
          [bond]
          miimon=140
          mode=balance-rr

          [ipv4]
          dhcp-timeout=2147483647
          method=auto

          [ipv6]
          dhcp-timeout=2147483647
          method=disabled
        path: /etc/NetworkManager/system-connections/bond0.nmconnection
        permissions: 600
      - content: |
          [connection]
          autoconnect=true
          autoconnect-slaves=-1
          id=eth2
          interface-name=eth2
          master=bond0
          slave-type=bond
          type=802-3-ethernet
          uuid=dfd202f5-562f-5f07-8f2a-a7717756fb70
        path: /etc/NetworkManager/system-connections/eth2.nmconnection
        permissions: 600
      - content: |
          [connection]
          autoconnect=true
          autoconnect-slaves=-1
          id=eth4
          interface-name=eth4
          master=bond0
          slave-type=bond
          type=802-3-ethernet
          uuid=0523c0a1-5f5e-5603-bcf2-68155d5d322e
        path: /etc/NetworkManager/system-connections/eth4.nmconnection
        permissions: 600
  elemental:
    install:
      debug: true
      device-selector:
        - key: Name
          operator: In
          values:
            - /dev/nvme0n1
        - key: Size
          operator: Gt
          values:
            - 1000Gi
      poweroff: true
      snapshotter:
        type: btrfs
    reset:
      reboot: true
      reset-oem: true
      reset-persistent: true
machineInventoryLabels:
  location: in-transit
```

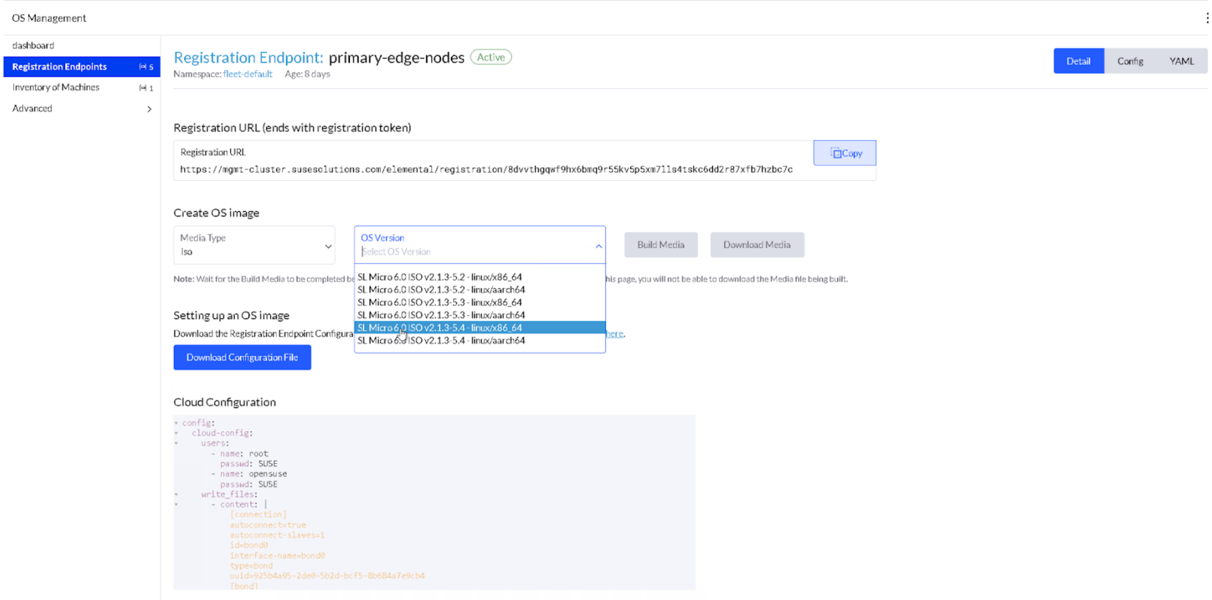
Figure 5: Cloud Configuration fields in the YAML file

Finalizing the Endpoint

Once all fields are configured the endpoint needs to be finalized. In order to do that click Create and the new registration endpoint should appear in the dashboard with a status of Active, indicating it's ready to be used for device registration.

Create Elemental OS seed image

Once the Registration Endpoint is created and marked "active", the next critical task is to prepare a seed image. A seed image contains the necessary configurations to automatically register and deploy Edge nodes as a part of a cluster. The lightweight bootable media is designed to streamline the onboarding of edge devices by embedding essential configuration elements, including the registration URL for the management endpoint and the server certificate to enable secure communication. When a device is booted using this image, it automatically registers with the OS Management system, initiates the installation of the SUSE Linux Enterprise (SLE) Micro operating system, and reboots to become an active member of the Kubernetes clusterThe seed image is defined as a Kubernetes resource and built using the Build Media function in the OS Management dashboard.



Registration Endpoint: primary-edge-nodes Active
Namespace: fleet-default Age: 8 days

Registration URL (ends with registration token)

Registration URL
https://mgmt-cluster.susesolutions.com/elemental/registration/8dvvthgqwf9hx6bmq9r55kv5p5xm7lls4tskc6dd2r87xfb7hzbc7c

Create OS image

Media Type: Iso

OS Version: Select OS Version

- SL Micro 6.0 ISO v2.1.3-5.2 - linux/x86_64
- SL Micro 6.0 ISO v2.1.3-5.2 - linux/aarch64
- SL Micro 6.0 ISO v2.1.3-5.3 - linux/x86_64
- SL Micro 6.0 ISO v2.1.3-5.3 - linux/aarch64
- SL Micro 6.0 ISO v2.1.3-5.4 - linux/x86_64
- SL Micro 6.0 ISO v2.1.3-5.4 - linux/aarch64

Note: Wait for the Build Media to be completed be his page, you will not be able to download the Media file being built.

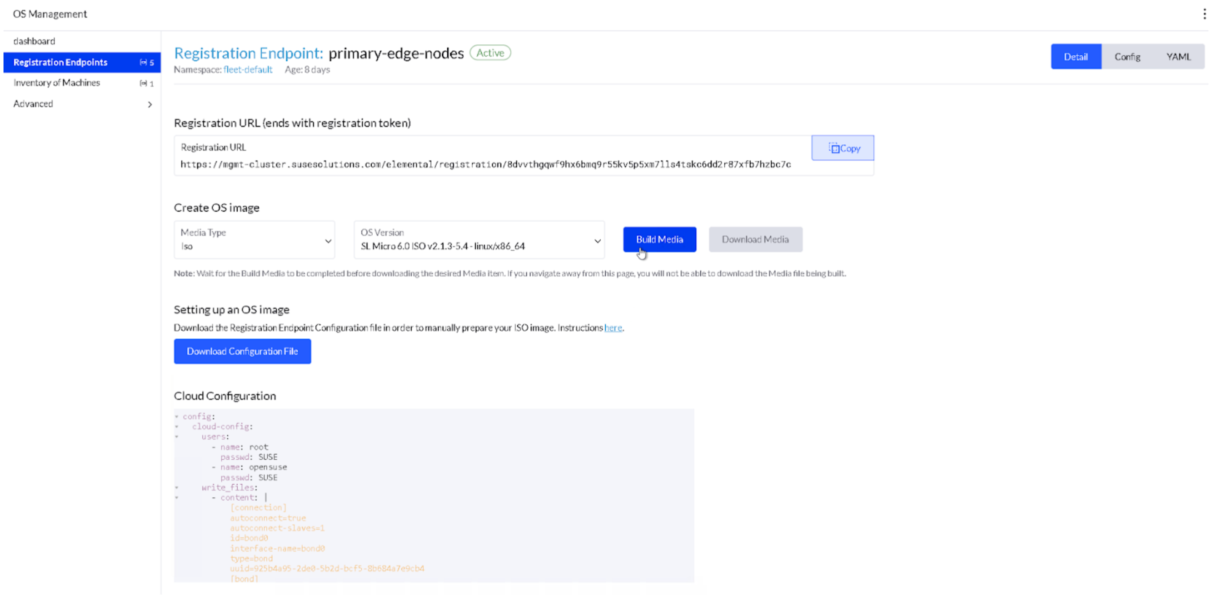
Setting up an OS image

Download the Registration Endpoint Configura here.

Cloud Configuration

```
config:
  cloud-config:
    users:
      - name: root
        passwd: SUSE
      - name: opensuse
        passwd: SUSE
    write_files:
      - content: |
          [connection]
          autoconnect=true
          autoconnect-slaves=1
          id=bond0
          interface-name=bond0
          type=bond
          uuid=925b4a95-2de0-5b2d-bcf5-8b684a7e9cb4
          [bond]
```

Figure 6: Select the SL Micro version for the Seed Image



Registration Endpoint: primary-edge-nodes Active

Namespace: fleet-default Age: 8 days

Registration URL (ends with registration token)

Registration URL: https://mgmt-cluster.susesolutions.com/elemental/registration/8dvvthgqwf9hx6bmq9r55kv5p5xm7lls4tskc6dd2r87xfb7hzbc7c

Create OS image

Media Type: Iso

OS Version: SL Micro 6.0 ISO v2.1.3-5.4 - linux/x86_64

Note: Wait for the Build Media to be completed before downloading the desired Media item. If you navigate away from this page, you will not be able to download the Media file being built.

Setting up an OS image

Download the Registration Endpoint Configuration file in order to manually prepare your ISO image. Instructions here.

Cloud Configuration

```
config:
  cloud-config:
    users:
      - name: root
        passwd: SUSE
      - name: opensuse
        passwd: SUSE
    write_files:
      - content: |
          [connection]
          autoconnect=true
          autoconnect-slaves=1
          id=bond0
          interface-name=bond0
          type=bond
          uuid=925b4a95-2de0-5b2d-bcf5-8b684a7e9cb4
          [bond]
```

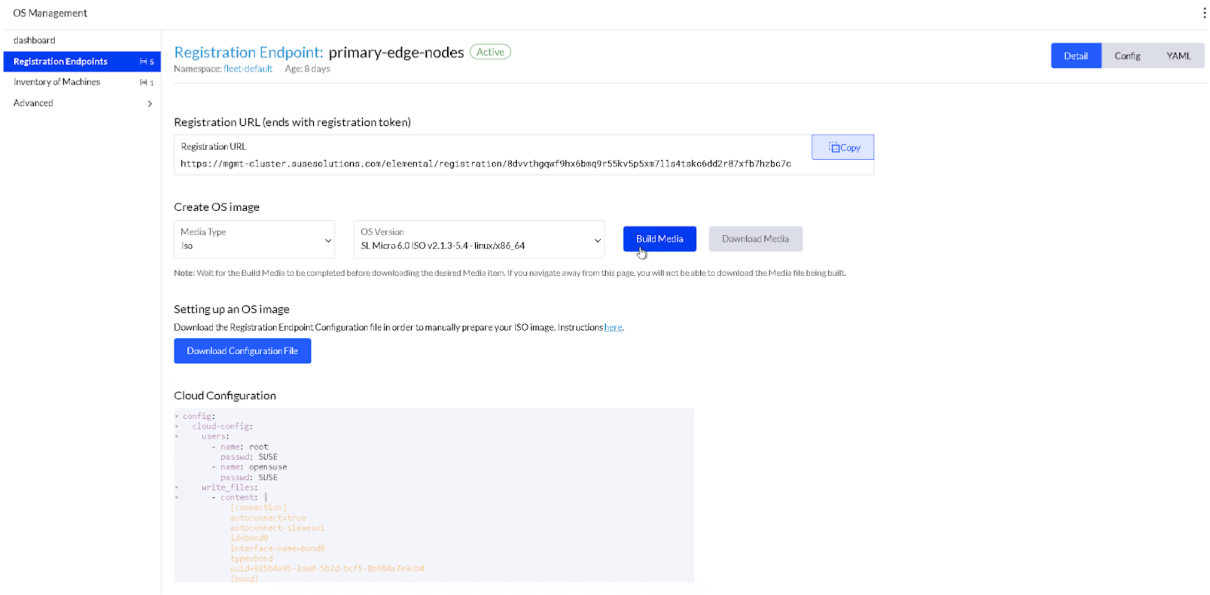
Figure 7: Building the Seed Image

Steps to Build the Image

A seed image includes the initial registration configuration, so it can be auto registered, installed and fully deployed as part of a cluster. The contents of the file are nothing more than the registration URL that the node needs to register and the proper server certificate, so it can connect securely. This seed image can then be used to provision an infinite number of machines.

The seed image can be built using the Build Media button, but first the choice between ISO or RAW image has to be made. ISO needs two devices (a device with ISO and another disk to install Elemental) whereas the RAW image allows booting from a single device and directly installs the operating system on the device. The RAW image only contains a boot and a recovery partition, and it boots first into recovery mode to install Elemental.

Deploying with the Seed Image



Registration Endpoint: primary-edge-nodes Active

Namespace: fleet-default Age: 8 days

Registration URL (ends with registration token)

Registration URL

```
https://mgmt-cluster.susesolutions.com/elemental/registration/8dvvthgqwf9hx6bmq9r55kv5p5xm7lls4tskc6dd2r87xfb7hzbc7c
```

Create OS image

Media Type: Iso

OS Version: SL Micro 6.0 ISO v2.1.3-5.4 - linux/x86_64

Note: Wait for the Build Media to be completed before downloading the desired Media item. If you navigate away from this page, you will not be able to download the Media file being built.

Setting up an OS image

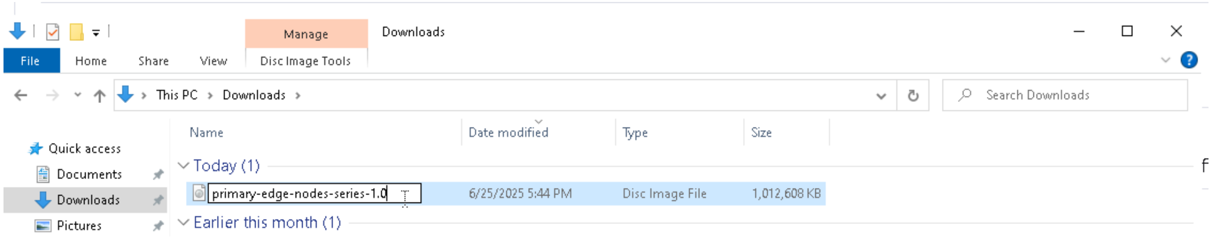
Download the Registration Endpoint Configuration file in order to manually prepare your ISO image. Instructions here.

Cloud Configuration

```
config:
  cloud-config:
    users:
      - name: root
        passwd: SUSE
      - name: opensuse
        passwd: SUSE
    write_files:
      - content: |
          [connection]
          autoconnect=true
          autoconnect-slaves=1
          id=bond0
          interface-name=bond0
          type=bond
          uuid=925b4a95-2de0-5b2d-bcf5-8b684a7e9cb4
          [bond]
```

Figure 8: Click Download Media to retrieve the image file.

Once the image is built, it is downloaded to retrieve the image file. Later, this image can boot any number of Edge devices. It is recommended to rename the downloaded image file.

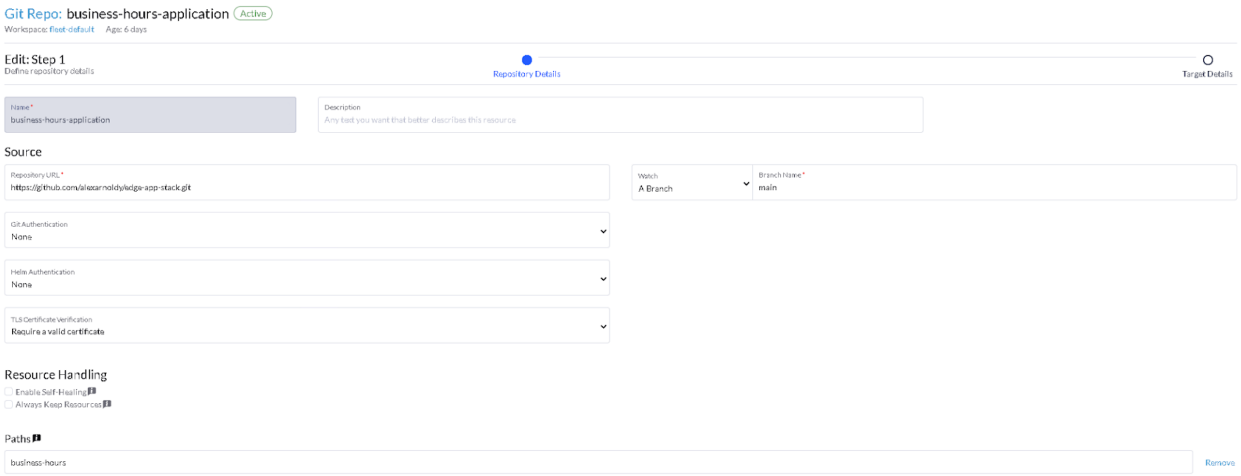


Name	Date modified	Type	Size
Today (1)			
primary-edge-nodes-series-1.0	6/25/2025 5:44 PM	Disc Image File	1,012,608 KB
Earlier this month (1)			

Figure 9: Rename the Seed Image once downloaded

Configure SUSE Rancher Prime: Continuous Delivery Application Repository

Once the seed image is built and ready, the next phase is defining the applications that will run at the Edge locations.   The applications are deployed using a GitOps model. First, Git compatible repositories (in our example we have two repositories named business-hours-application and after-hours-application) are configured in a public or private git repository. The deployment path is set, and the target cluster group is defined (business-hours-cluster-group or after-hours-cluster-group).



Git Repo: business-hours-application Active

Workspace: fleet-default Age: 6 days

Edit: Step 1

Define repository details

Name* business-hours-application

Description Any text you want that better describes this resource

Source

Repository URL* https://github.com/alexarnoldy/edge-app-stack.git

Watch A Branch

Branch Name* main

Git Authentication None

Helm Authentication None

TLS Certificate Verification Require a valid certificate

Resource Handling

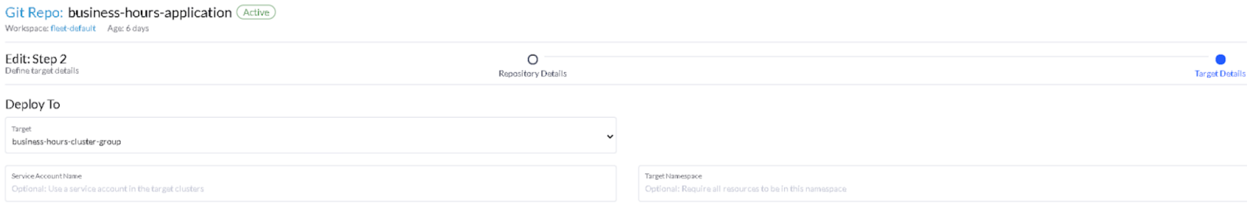
□ Enable Self-Healing

□ Always Keep Resources

Paths

business-hours

Figure 10: Configure the source Git repository details



Git Repo: business-hours-application Active

Workspace: fleet-default Age: 6 days

Edit: Step 2

Define target details

Deploy To

Target business-hours-cluster-group

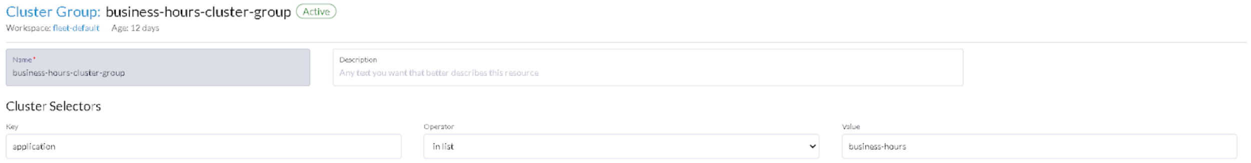
Service Account Name Optional: Use a service account in the target clusters

Target Namespace Optional: Require all resources to be in this namespace

Figure 11: Configure the Target Cluster Group

Then, these cluster groups are configured to dynamically select Kubernetes clusters labeled with “application: business-hours”, ensuring only the appropriate Edge clusters receive the desired application. This setup creates a seamless pipeline from Git to deployed application, enabling automated and scalable application delivery.

In the future, when the application code is updated, Rancher Prime Continuous Delivery will detect the changes and automatically update all the Edge clusters.



Cluster Group: business-hours-cluster-group Active

Workspace: fleet-default Age: 12 days

Name* business-hours-cluster-group

Description Any text you want that better describes this resource

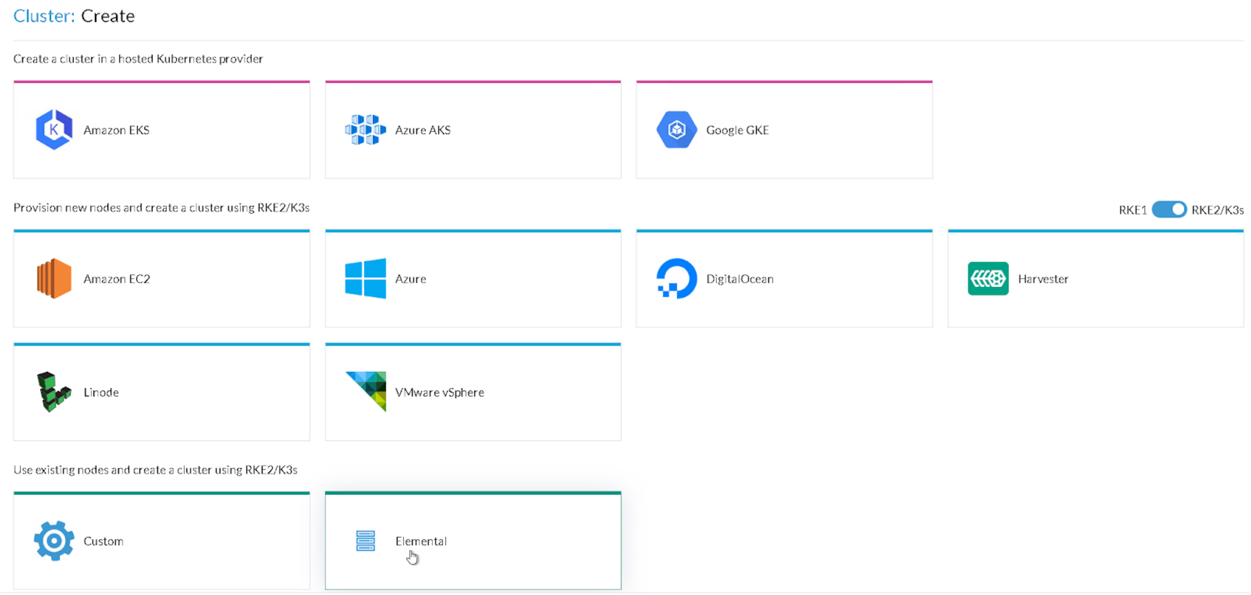
Cluster Selectors

Key	Operator	Value
application	in list	business-hours

Figure 12: Cluster Group: business-hours-cluster-group

Define RKE2 Cluster Configuration

Just after the application experts had completed their work, the Kubernetes experts finally became available to configure the RKE2 Kubernetes cluster definition that will be deployed at the Edge locations. Thus, the first step was to choose a provisioning method, which in this case is via Elemental.


Cluster: Create

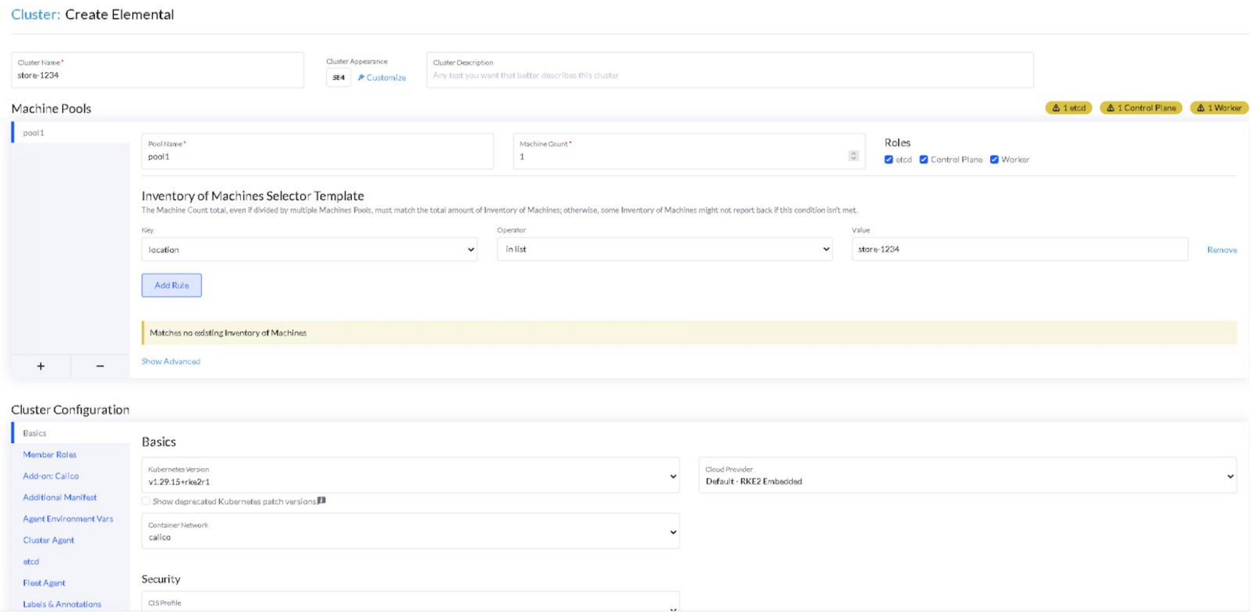
Create a cluster in a hosted Kubernetes provider

Provision new nodes and create a cluster using RKE2/K3s

Use existing nodes and create a cluster using RKE2/K3s

Figure 13: Create Elemental Cluster by clicking on the "Elemental" tile on the dashboard

Next, we defined an Elemental cluster for the Edge location. In our example, we have one cluster per Edge location. We name our RKE2 cluster based on the store number (e.g., store-1234). The cluster definition will include defined machine pools using selectors that target nodes by the "location" label that was defined in the Elemental operating system image. Our experts configured the cluster definition with a specific Kubernetes version and many other tunables, such as defining a CNI plugin (Calico in our example).


Cluster: Create Elemental

Cluster Name* store-1234

Cluster Appearance SE4 Customize

Cluster Description: Any text you want that better describes this cluster

Machine Pools

Pool Name* pool1

Machine Count* 1

Roles ☑ etcd ☑ Control Plane ☑ Worker

Inventory of Machines Selector Template

The Machine Count total, even if divided by multiple Machines Pools, must match the total amount of Inventory of Machines; otherwise, some Inventory of Machines might not report back if this condition isn't met.

Key	Operator	Value	
location	in list	store-1234	Remove

Matches no existing Inventory of Machines

Cluster Configuration

Basics

Kubernetes Version v1.29.15+rke2r1

Cloud Provider Default - RKE2 Embedded

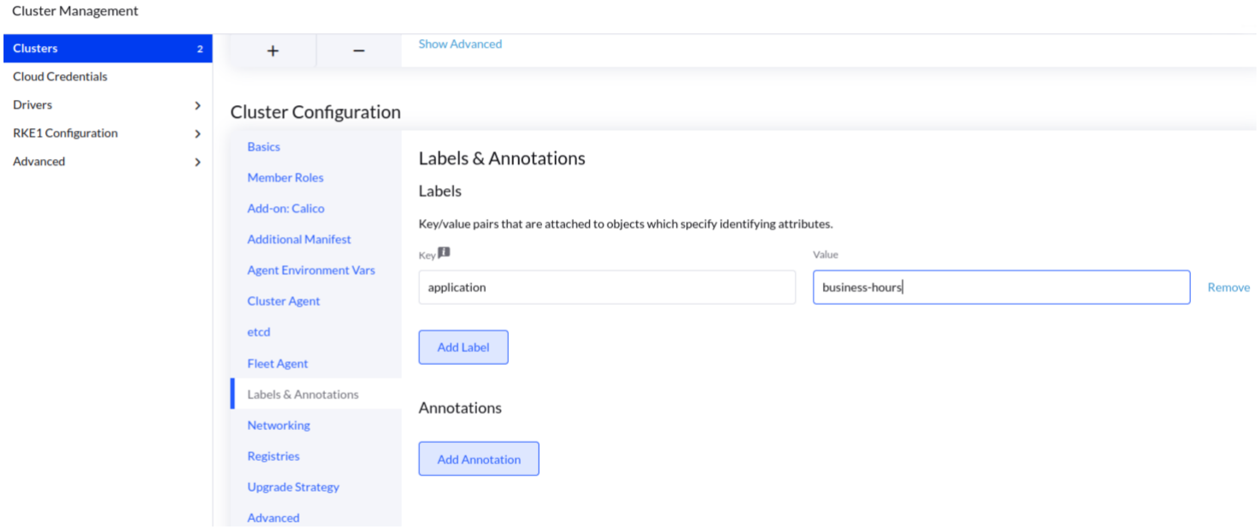
□ Show deprecated Kubernetes patch versions

Container Network calico

Security

CIS Profile

Figure 14: Configuring clusters and defining machine pools



Labels & Annotations

Labels

Key/value pairs that are attached to objects which specify identifying attributes.

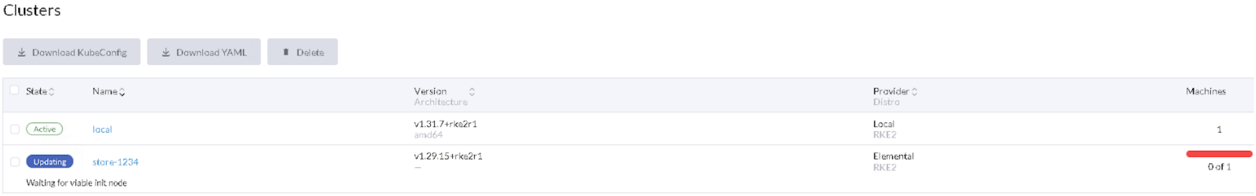
Key: application

Value: business-hours

Annotations

Figure 15: Applying the label key “application” and value “business-hours” to the newly defined RKE2 cluster

We then configured a label on the RKE2 cluster to signal Rancher Prime Continuous Delivery that it should automatically deploy an application to the cluster once it reaches an “active” state. In this case, the label has a key of “application” and a value of “business-hours”.



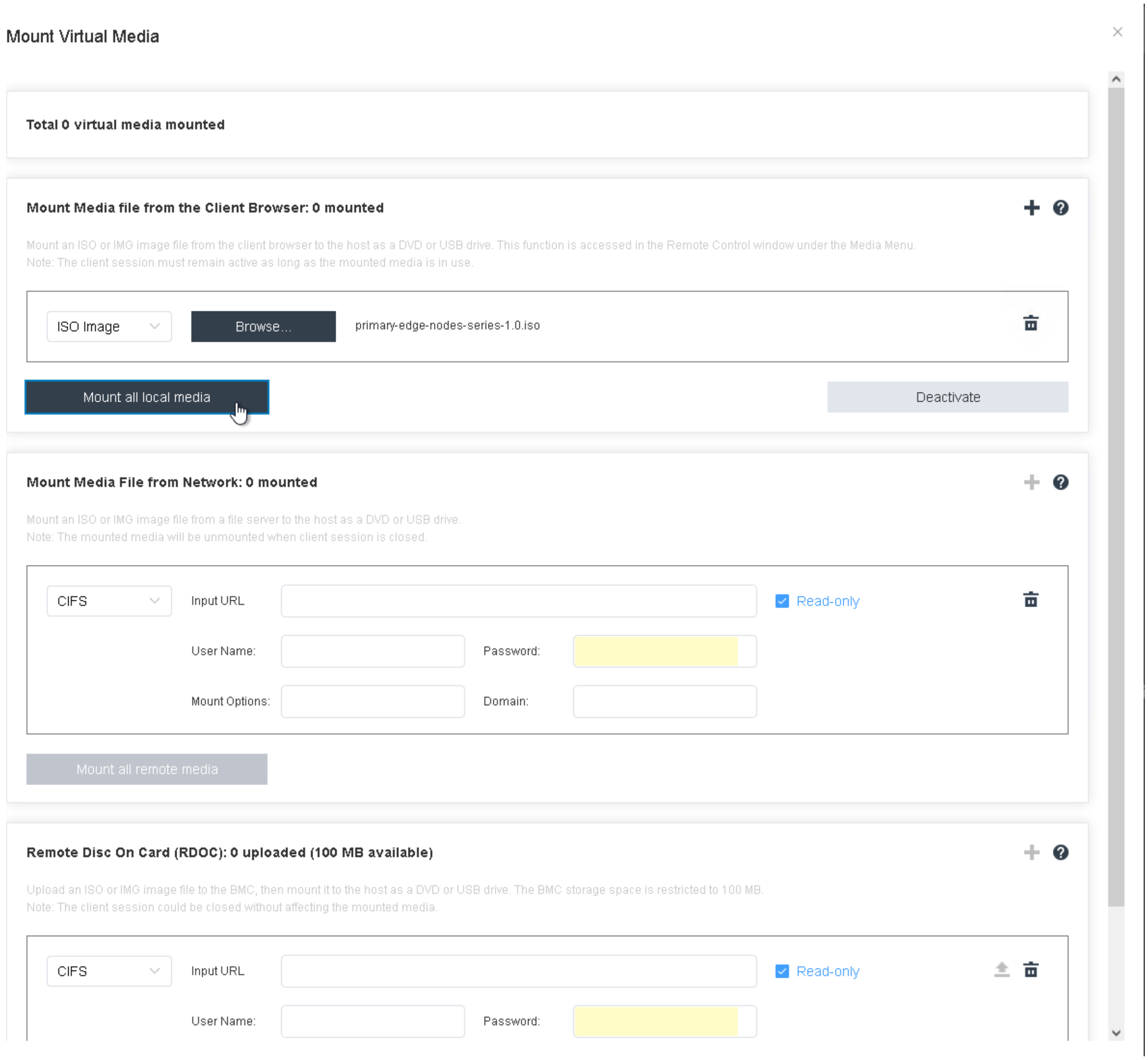
State	Name	Version Architecture	Provider Distro	Machines
Active	local	v1.31.7+rke2r1 amd64	Local RKE2	1
Updating Waiting for viable init node	store-1234	v1.29.15+rke2r1 —	Elemental RKE2	0 of 1

Figure 16: The new cluster is defined and waiting for an available node

The newly defined RKE2 cluster will remain in a state of "updating" until nodes with the appropriate node label become available.

Initial Edge Provisioning

After defining the RKE2 Cluster comes the provisioning stage. The Edge device begins its setup by mounting the ISO image (primary-edge-node-osdisk-1.0.iso) via the XClarity Controller Remote Console. This image contains the previously generated SUSE Linux Micro OS and registration configuration.



Mount Virtual Media

Total 0 virtual media mounted

Mount Media file from the Client Browser: 0 mounted

Mount an ISO or IMG image file from the client browser to the host as a DVD or USB drive. This function is accessed in the Remote Control window under the Media Menu.
Note: The client session must remain active as long as the mounted media is in use.

ISO Image | primary-edge-nodes-series-1.0.iso

Mount Media File from Network: 0 mounted

Mount an ISO or IMG image file from a file server to the host as a DVD or USB drive.
Note: The mounted media will be unmounted when client session is closed.

CIFS | Input URL: | ☑ Read-only
User Name: | Password:
Mount Options: | Domain:

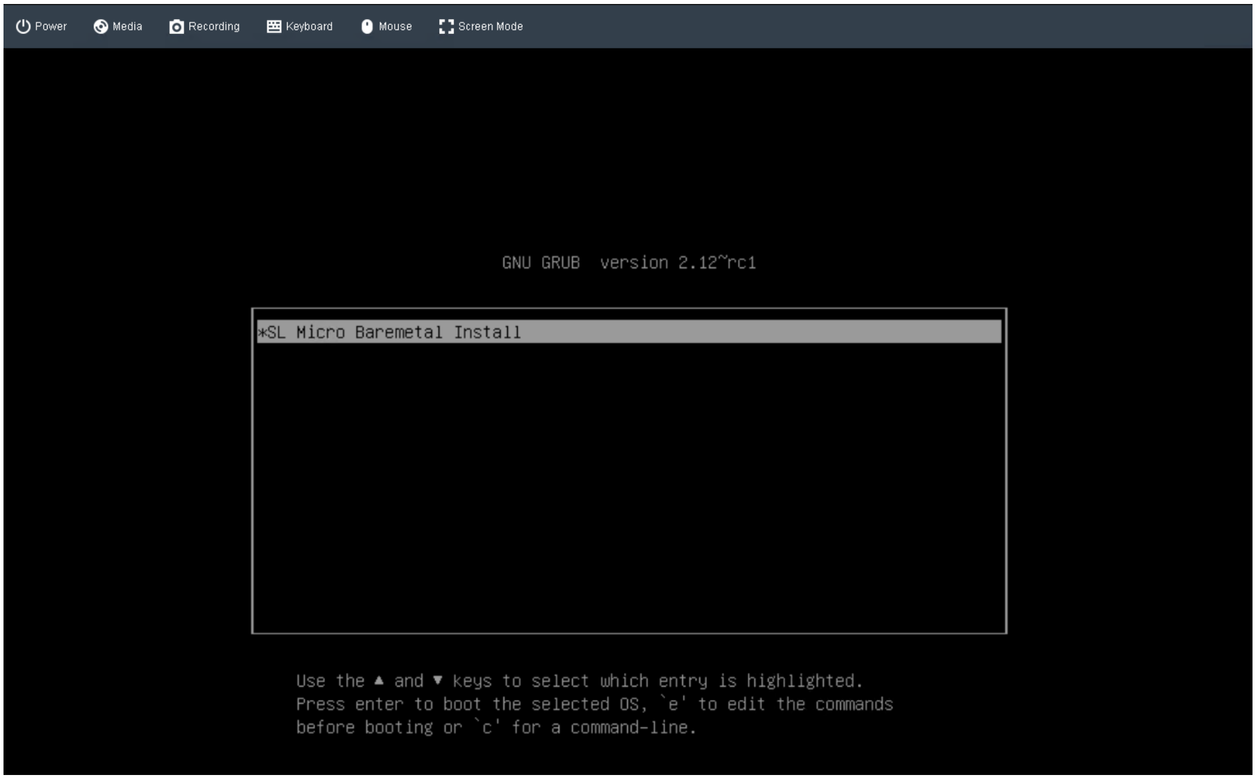
Remote Disc On Card (RDOC): 0 uploaded (100 MB available)

Upload an ISO or IMG image file to the BMC, then mount it to the host as a DVD or USB drive. The BMC storage space is restricted to 100 MB.
Note: The client session could be closed without affecting the mounted media.

CIFS | Input URL: | ☑ Read-only
User Name: | Password:

Figure 17: Mount ISO image via the XClarity Controller Remote Console

Upon power on, the node reaches the GRUB bootloader screen, then initiates the OS installation process. During installation the node will complete the initial registration with SUSE Elemental.



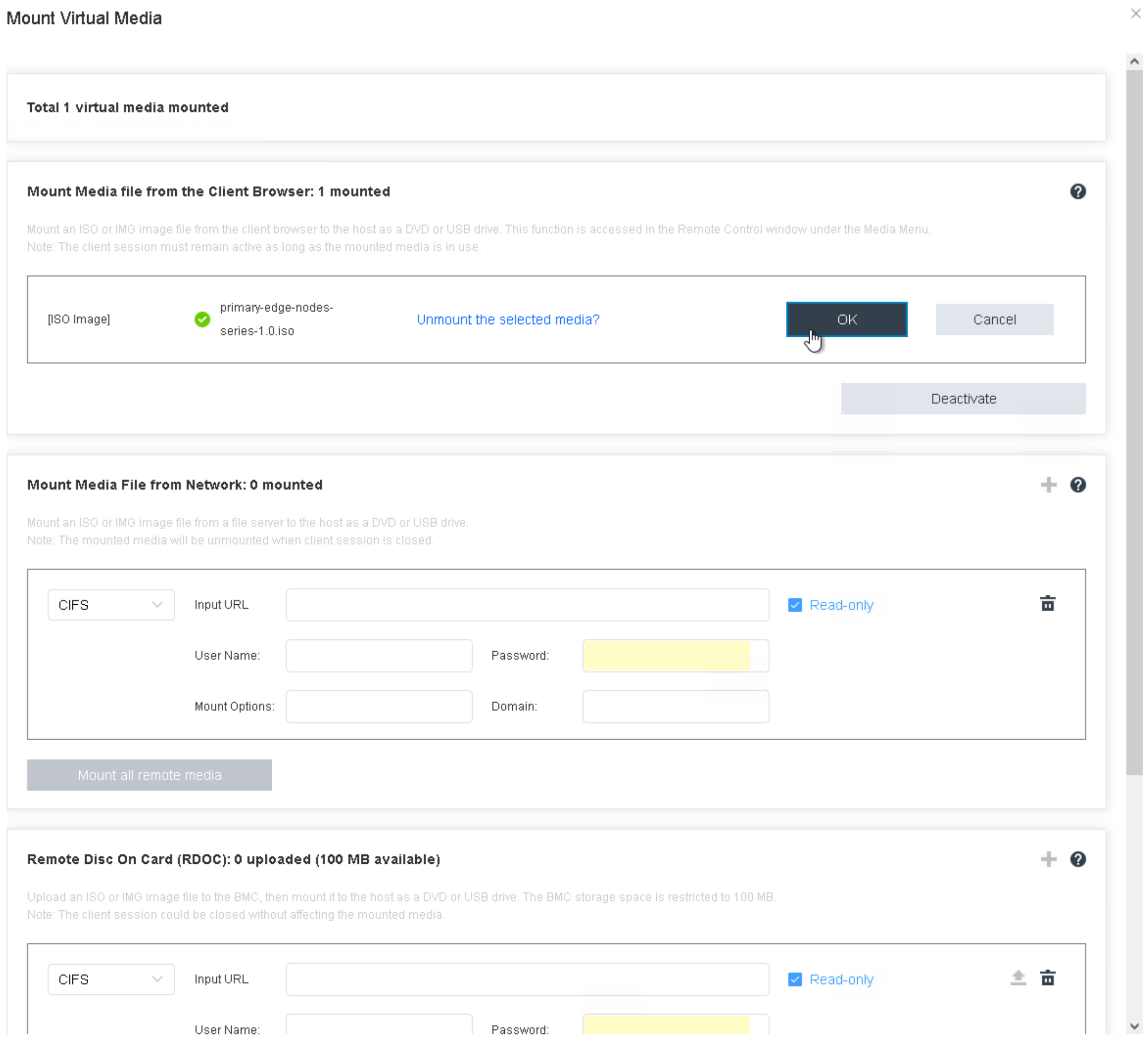
```
GNU GRUB  version 2.12~rc1

*SL Micro Baremetal Install

Use the ▲ and ▼ keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line.
```

Figure 18: OS Installation screen

After installation was completed, the node powered down. This was a signal to the technician that the installation completed successfully, and the node was ready for the next step in the process. There was no need for the technician to monitor the installation. In many examples, installation can be completed without attaching the keyboard/video/mouse. If a remote ISO image was used for the installation (as is the case in our example), the final step would be to unmount the ISO image. On the other hand, if the installation was done at a staging location, the node is now considered ready for shipping to the Edge location.



Mount Virtual Media

Total 1 virtual media mounted

Mount Media file from the Client Browser: 1 mounted

Mount an ISO or IMG image file from the client browser to the host as a DVD or USB drive. This function is accessed in the Remote Control window under the Media Menu.
Note: The client session must remain active as long as the mounted media is in use.

[ISO Image] primary-edge-nodes-series-1.0.iso Unmount the selected media?

Mount Media File from Network: 0 mounted

Mount an ISO or IMG image file from a file server to the host as a DVD or USB drive.
Note: The mounted media will be unmounted when client session is closed.

CIFS Input URL ☑ Read-only

User Name: Password:

Mount Options: Domain:

Remote Disc On Card (RDOC): 0 uploaded (100 MB available)

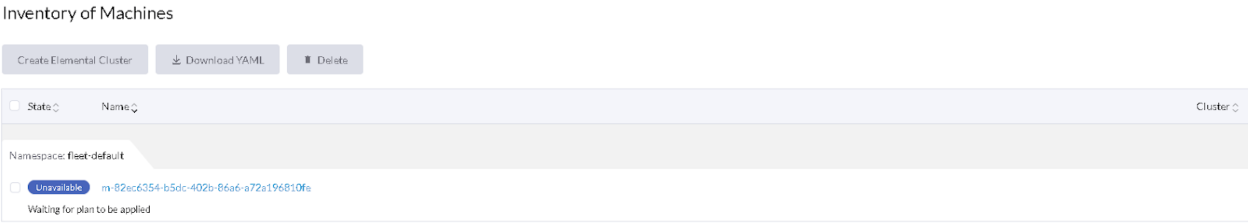
Upload an ISO or IMG image file to the BMC, then mount it to the host as a DVD or USB drive. The BMC storage space is restricted to 100 MB.
Note: The client session could be closed without affecting the mounted media.

CIFS Input URL ☑ Read-only

User Name: Password:

Figure 19: Unmount ISO image if a remote iso image was used for the installation

In the Rancher Prime web interface, we can see that the node has been registered with Elemental.



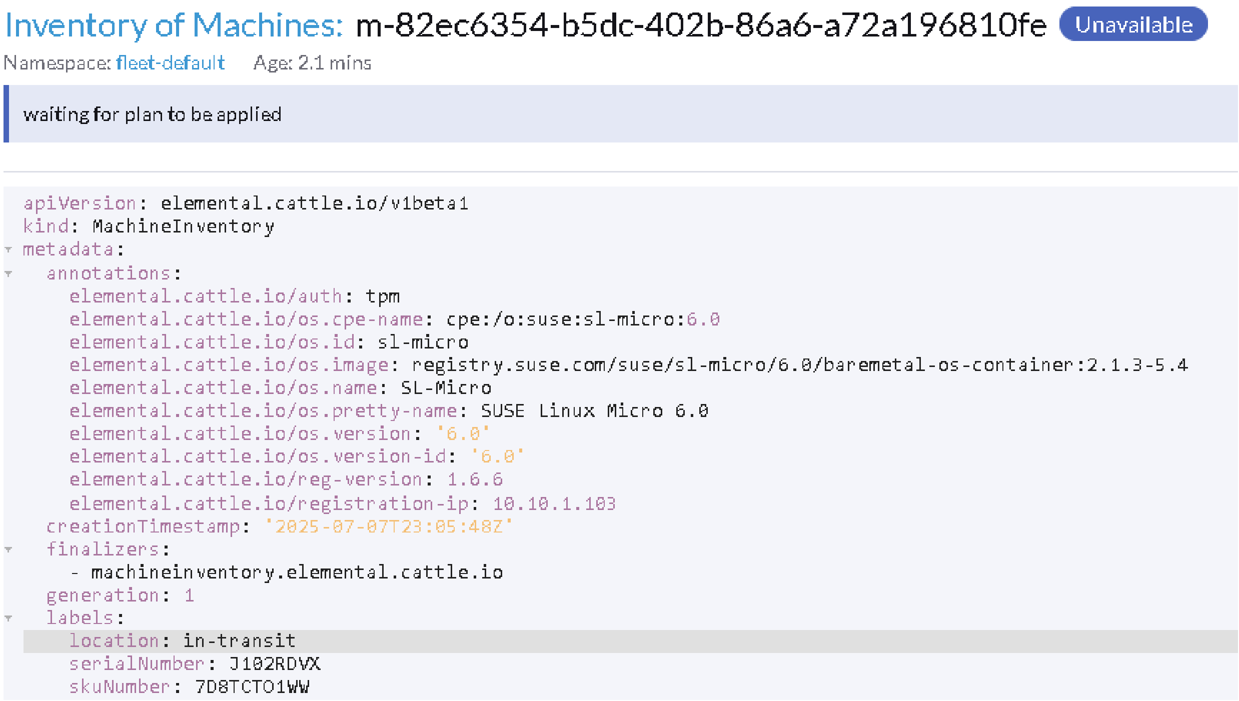
Inventory of Machines

□ State	Name	Cluster
Namespace: fleet-default		
□ Unavailable Waiting for plan to be applied	m-82ec6354-b5dc-402b-86a6-a72a196810fe	

Figure 20: Inventory of single node registered with Elemental in Rancher Prime web interface

The node is listed in SUSE Elemental under Inventory of Machines. It shows as Unavailable, with a status of "Waiting for plan to be applied," indicating successful registration but pending configuration.

We also see that the node has been configured with the location label and a value of “in-transit”. This helps to identify which nodes are at Edge locations and which nodes are still flowing through the build-configure-deployment process.



Inventory of Machines: m-82ec6354-b5dc-402b-86a6-a72a196810fe Unavailable

Namespace: fleet-default Age: 2.1 mins

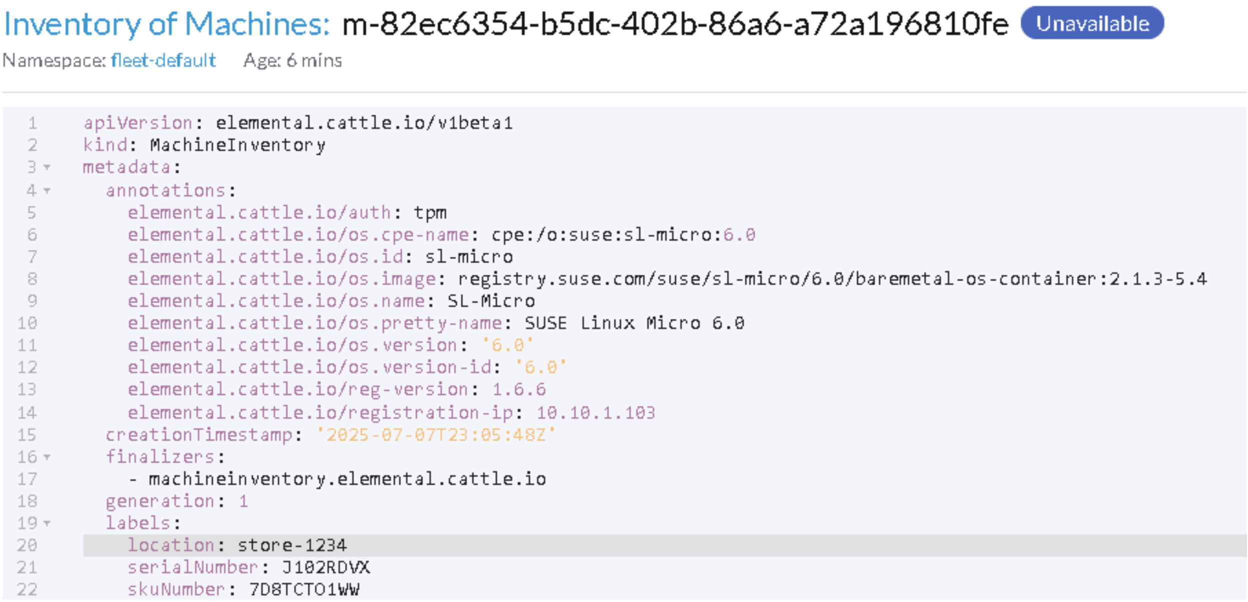
waiting for plan to be applied

```
apiVersion: elemental.cattle.io/v1beta1
kind: MachineInventory
metadata:
  annotations:
    elemental.cattle.io/auth: tpm
    elemental.cattle.io/os.cpe-name: cpe:/o:suse:sl-micro:6.0
    elemental.cattle.io/os.id: sl-micro
    elemental.cattle.io/os.image: registry.suse.com/suse/sl-micro/6.0/baremetal-os-container:2.1.3-5.4
    elemental.cattle.io/os.name: SL-Micro
    elemental.cattle.io/os.pretty-name: SUSE Linux Micro 6.0
    elemental.cattle.io/os.version: '6.0'
    elemental.cattle.io/os.version-id: '6.0'
    elemental.cattle.io/reg-version: 1.6.6
    elemental.cattle.io/registration-ip: 10.10.1.103
  creationTimestamp: '2025-07-07T23:05:48Z'
  finalizers:
    - machineinventory.elemental.cattle.io
  generation: 1
  labels:
    location: in-transit
    serialNumber: J102RDVX
    skuNumber: 7D8TCTO1WW
```

Figure 21: Note that the location has changed to "in-transit"

The system captures detailed metadata information including OS version (SUSE Linux Micro 6.0), TPM-based authentication, registration IP, and device labels like location: in-transit, serialNumber, and skuNumber. This confirms the device is fully registered and ready for further integration.

Node On-site



Inventory of Machines: m-82ec6354-b5dc-402b-86a6-a72a196810fe Unavailable

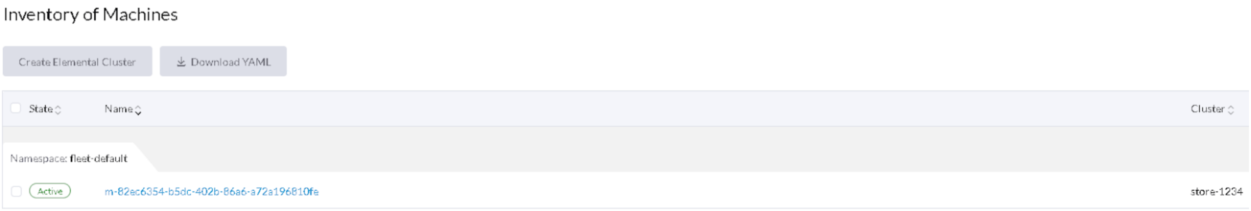
Namespace: fleet-default Age: 6 mins

```
apiVersion: elemental.cattle.io/v1beta1
kind: MachineInventory
metadata:
  annotations:
    elemental.cattle.io/auth: tpm
    elemental.cattle.io/os.cpe-name: cpe:/o:suse:sl-micro:6.0
    elemental.cattle.io/os.id: sl-micro
    elemental.cattle.io/os.image: registry.suse.com/suse/sl-micro/6.0/baremetal-os-container:2.1.3-5.4
    elemental.cattle.io/os.name: SL-Micro
    elemental.cattle.io/os.pretty-name: SUSE Linux Micro 6.0
    elemental.cattle.io/os.version: '6.0'
    elemental.cattle.io/os.version-id: '6.0'
    elemental.cattle.io/reg-version: 1.6.6
    elemental.cattle.io/registration-ip: 10.10.1.103
  creationTimestamp: '2025-07-07T23:05:48Z'
  finalizers:
    - machineinventory.elemental.cattle.io
  generation: 1
  labels:
    location: store-1234
    serialNumber: J102RDVX
    skuNumber: 7D8TCTO1WW
```

Figure 22: A new label is created that has "location: store-1234"

When the node is positioned and cabled at the Edge location, the “location” label is manually updated to identify the specific physical location of the node—for example, "store-1234."

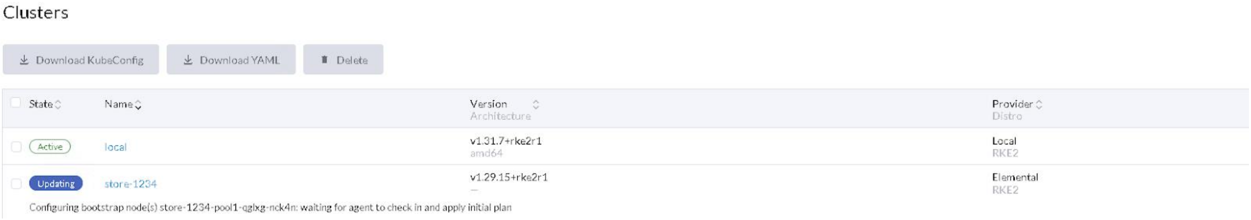
While this labeling process can be automated, it is strongly recommended to include an additional verification step to ensure accuracy. This should involve cross-checking the serial number displayed in Elemental against the physical serial number of the Edge node at the Edge location. This will help to ensure the correct application workloads are deployed to the intended site.



State	Name	Cluster
Namespace: fleet-default		
Active	m-82ec6354-b5dc-402b-86a6-a72a196810fe	store-1234

Figure 23: Status is updated to “Active”

Moving on, upon powering on, the Edge node initiated a check-in process with Elemental, during which its status was updated to “Active.” This designation indicates that the node was online, recognized by the management system, and ready to proceed with the installation of an RKE2 Kubernetes cluster.

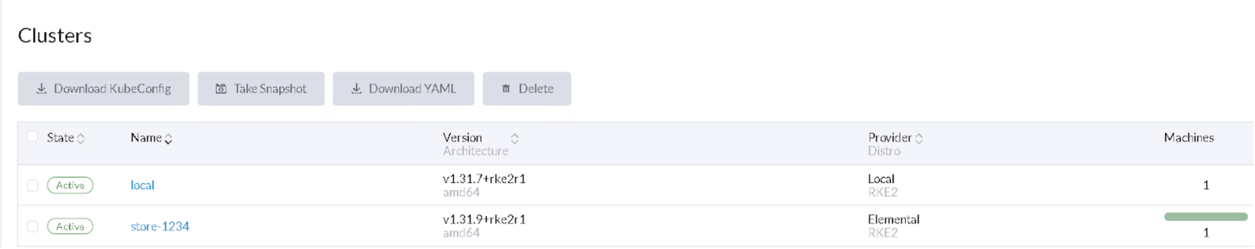


State	Name	Version Architecture	Provider Distro
Active	local	v1.31.7+rke2r1 amd64	Local RKE2
Updating	store-1234	v1.29.15+rke2r1 —	Elemental RKE2

Configuring bootstrap node(s) store-1234-pool1-qglxg-nck4n: waiting for agent to check in and apply initial plan

Figure 24: On site cluster being created

Consequently, when the “location” label was set to “store-1234”, Rancher Prime automatically initiated the deployment of the RKE2 Kubernetes cluster onto it. This label-driven mechanism ensured that the correct infrastructure was provisioned at the designated Edge location without requiring manual intervention, streamlining the setup process and maintaining consistency across deployments.

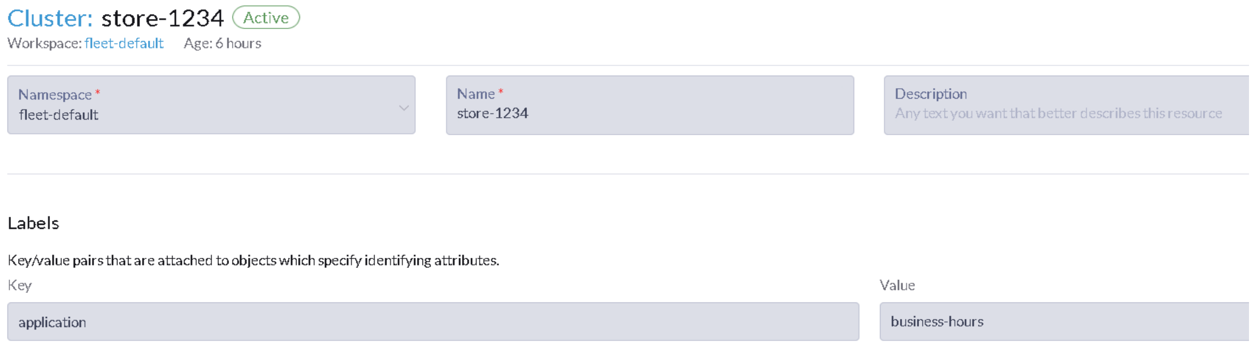


State	Name	Version Architecture	Provider Distro	Machines
Active	local	v1.31.7+rke2r1 amd64	Local RKE2	1
Active	store-1234	v1.31.9+rke2r1 amd64	Elemental RKE2	1

Figure 25: Store 1234 cluster active

The Edge node at Store 1234 now has an installed and operational RKE2 Kubernetes cluster. This deployment process is consistent for both single and multi-node Kubernetes clusters, ensuring uniformity in setup and management.

Automatic Application Deployment



Cluster: store-1234 Active
Workspace: fleet-default Age: 6 hours

Namespace *: fleet-default
Name *: store-1234
Description: Any text you want that better describes this resource

Labels

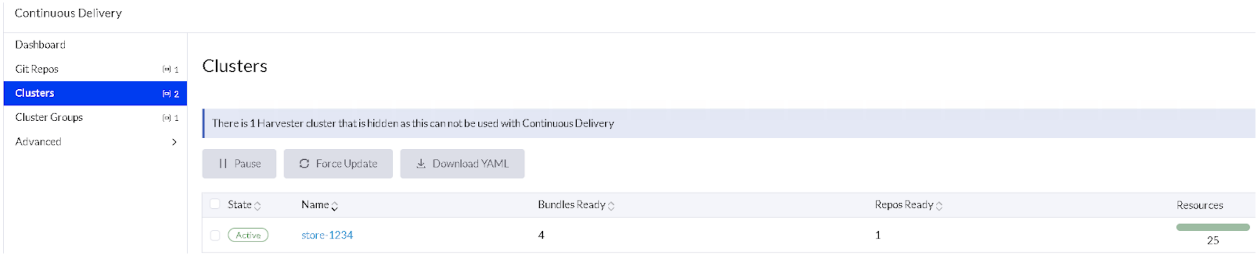
Key/value pairs that are attached to objects which specify identifying attributes.

Key	Value
application	business-hours

Figure 26: Cluster "store-1234" has the label "application: business-hours"

Administrators can verify which label(s) have been applied to the Store 1234 RKE2 cluster; in this case the label "application" is set to "business-hours" to signal that that application should be deployed.

In the Rancher Prime Continuous Delivery Engine, the initial application to be deployed to the RKE2 cluster is determined by the label previously applied during the cluster setup process. This label acts as a selector, allowing the Continuous Delivery Engine to match the cluster with the appropriate application deployment configuration and deploy the designated application automatically.



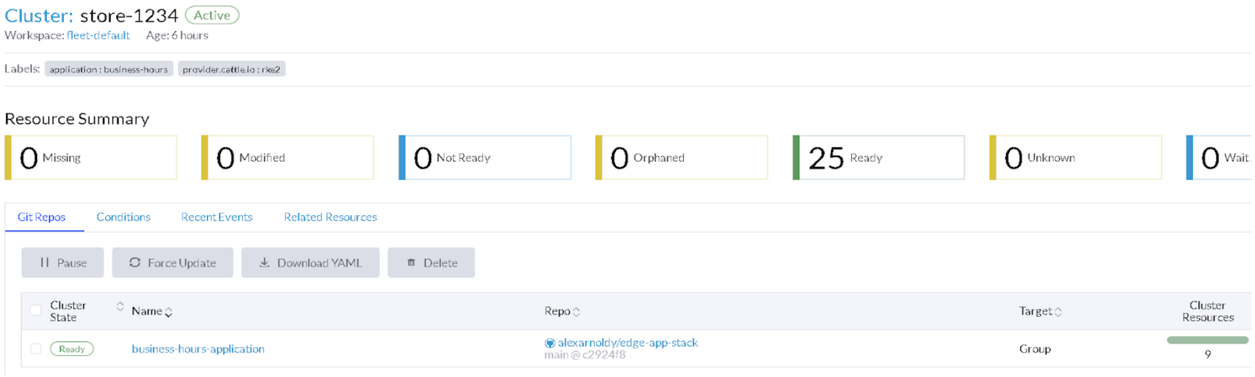
Clusters

There is 1 Harvester cluster that is hidden as this can not be used with Continuous Delivery

State	Name	Bundles Ready	Repos Ready	Resources
Active	store-1234	4	1	25

Figure 27: CD cluster business hours application is "Active"

The RKE2 cluster deployed at Store 1234 now has its designated applications automatically installed, triggered immediately after the cluster became active and was labeled appropriately. This automated deployment ensures that the cluster is production-ready without requiring manual intervention, streamlining the rollout of essential workloads to the Edge environment.



Cluster: store-1234 Active
Workspace: fleet-default Age: 6 hours

Labels: application : business-hours provider.cattle.io : rke2

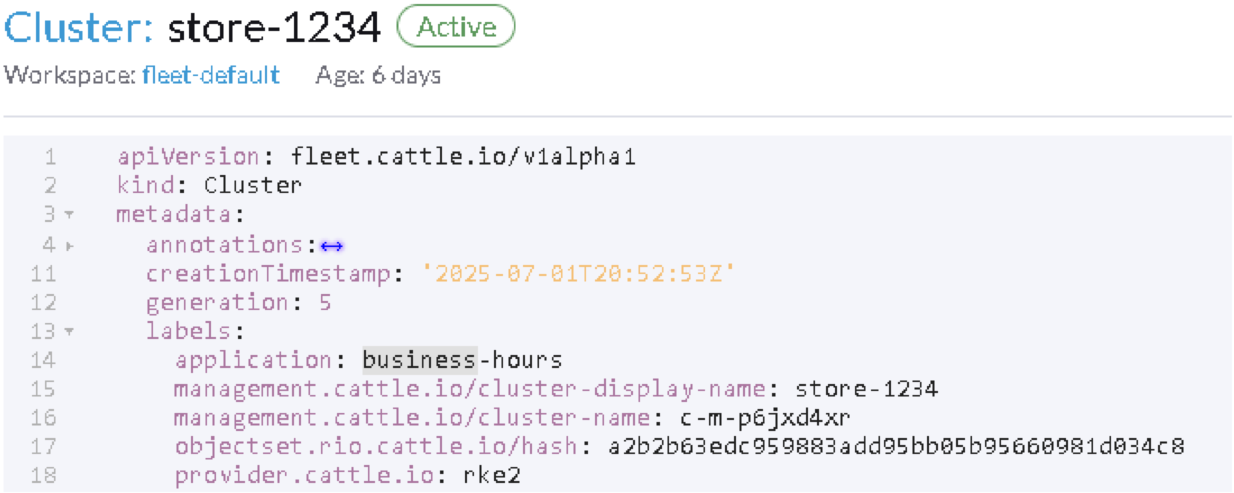
Resource Summary

0 Missing | 0 Modified | 0 Not Ready | 0 Orphaned | 25 Ready | 0 Unknown | 0 Wait

Cluster State	Name	Repo	Target	Cluster Resources
Ready	business-hours-application	alexarnoldy/edge-app-stack main @ c2924f8	Group	9

Figure 28: The business-hours-application was deployed to the cluster "store-1234"

Upon further inspection, it is confirmed that the business-hours application was automatically deployed to the RKE2 cluster at Store 1234. This deployment was triggered by the label applied during cluster setup, ensuring that the correct application was matched and delivered to the appropriate Edge location without manual intervention.



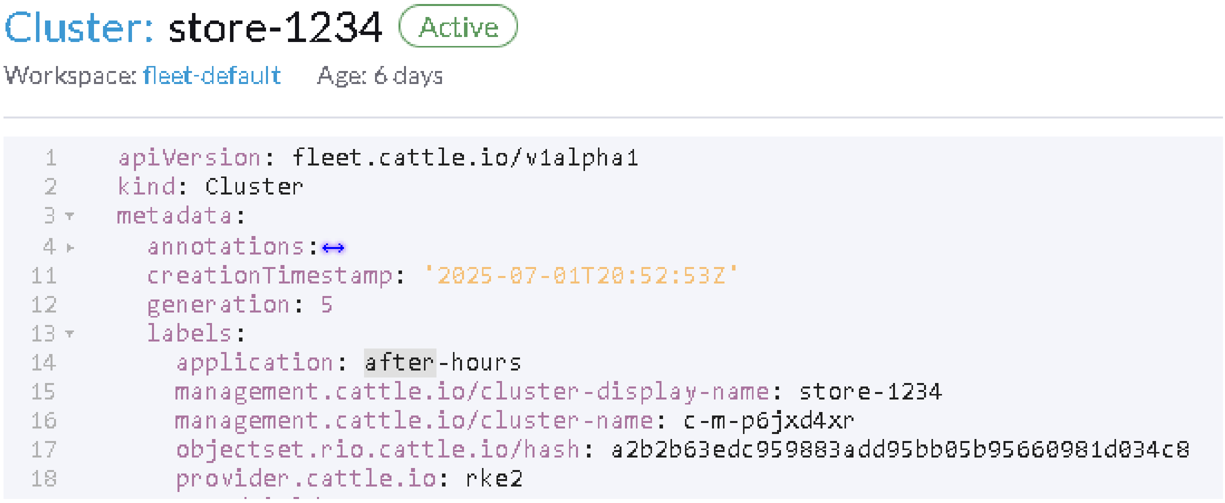
Cluster: store-1234 Active

Workspace: fleet-default Age: 6 days

```
apiVersion: fleet.cattle.io/v1alpha1
kind: Cluster
metadata:
  annotations:↔
  creationTimestamp: '2025-07-01T20:52:53Z'
  generation: 5
  labels:
    application: business-hours
    management.cattle.io/cluster-display-name: store-1234
    management.cattle.io/cluster-name: c-m-p6jxd4xr
    objectset.rio.cattle.io/hash: a2b2b63edc959883add95bb05b95660981d034c8
    provider.cattle.io: rke2
```

Figure 29: Cd update cluster label step 1 (CD using the label "application" to define the deployment policy)

As long as the RKE2 cluster at Store 1234 retains the label "application: business-hours," it will continue to run the business-hours application as defined by the deployment policy in Rancher Prime Continuous Delivery. Additional labels could be applied to this cluster to automatically deploy other applications or configurations, enabling flexible and scalable workload management across the entire Edge infrastructure.



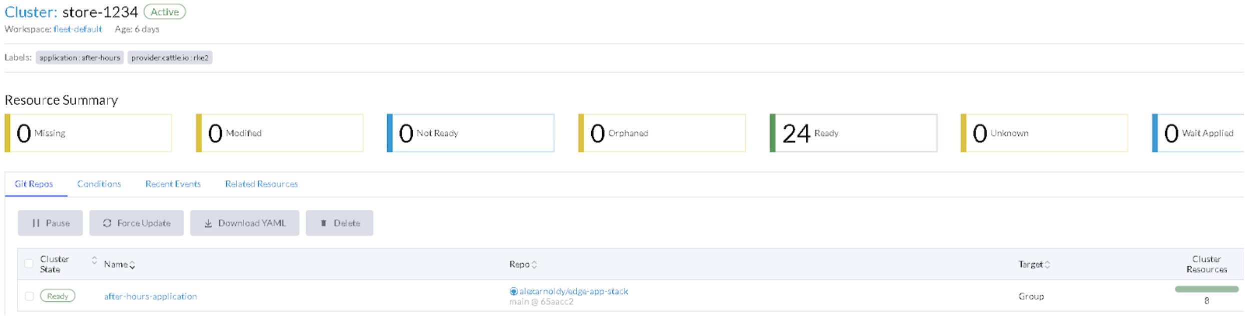
Cluster: store-1234 Active

Workspace: fleet-default Age: 6 days

```
apiVersion: fleet.cattle.io/v1alpha1
kind: Cluster
metadata:
  annotations:↔
  creationTimestamp: '2025-07-01T20:52:53Z'
  generation: 5
  labels:
    application: after-hours
    management.cattle.io/cluster-display-name: store-1234
    management.cattle.io/cluster-name: c-m-p6jxd4xr
    objectset.rio.cattle.io/hash: a2b2b63edc959883add95bb05b95660981d034c8
    provider.cattle.io: rke2
```

Figure 30: Continuous Delivery removes the business-hours, and deploys the after-hours application when the label changes from "application: business-hours" to "application: after-hours"

Changing the label on the Store 1234 RKE2 cluster from “application: business-hours” to “application: after-hours” triggers Rancher Prime Continuous Delivery to automatically remove the business-hours application and deploy the after-hours application in its place. This transition is handled entirely by the Continuous Delivery system, ensuring application workloads and configurations are applied based on label driven policies without requiring manual intervention.



Cluster: store-1234 Active

Workspace: fleet-default Age: 6 days

Labels: application : after-hours provider.cattle.io : rke2

Resource Summary

0 Missing | 0 Modified | 0 Not Ready | 0 Orphaned | 24 Ready | 0 Unknown | 0 Wait Applied

Cluster State	Name	Repo	Target	Cluster Resources
Ready	after-hours-application	alexarnoldy/edge-app-stack main @ 65aacc2	Group	8

Figure 31: The label on the Store-1234 has changed to “after-hours-application”

With the label change applied to the Store 1234 RKE2 cluster, the expected automatic transition of applications occurred, removing the business-hours application and deploying the after-hours application, demonstrating the effectiveness of Rancher Prime Continuous Delivery in managing workload changes through label-driven automation.

7 Conclusion:

This Lenovo Validated Design (LVD) for SUSE Edge 3.3 Kubernetes Deployment, integrates SUSE Edge Elemental and Lenovo ThinkEdge servers to demonstrate the ease of Edge compute deployment, application orchestration and lifecycle management of Edge workloads.

Key Capabilities

- **SUSE Elemental OS managemen**t: Ease of provisioning and friction-free logistics.
- **Flexibility:** Fully decoupled deployment processes easily adapt to changing schedules and workflows**.**
- **Automatic OS installation:** Highly scalable with low technical debt.
- **Scaling**: Elemental OS Seed Image can be used to deploy and whip up multiple nodes.
- **Security**: Robust security features of ThinkEdge servers ensure the safety of the Edge data and operations.
- **Automated Application Deployment:** Once the Git Repositories and the Cloud Fields are configured, the correct applications will be deployed.

Business Value

- **Faster scaling:** Low/no technical debt at Edge locations ensures scaling Edge projects without artificial limitations.
- **Data Protection:** Lenovo technology provides security-in-depth for individual servers and full clusters.
- **Lowered Labor Costs:** Technical debt has been lowered across the board, from design, deployment, central management to Edge location.
- **Faster Time to Value:** Trust a solution that is designed to adapt to changes and challenges, from Day 0 and beyond.

Thus, this document offers a replicable methodology to deploy edge nodes at scale.

8 Appendix: Lenovo ThinkEdge Security Features

Support for a Kensington lock and cable	ThinkEdge server has a slot on the rear side of the server which a customer-supplied Kensington lock and cable can be attached to, to help prevent theft of the server
Front bezel with lock	Optional component that mounts on the front of the server that restricts access to networking connectors on the front of the server. (not available in ThinkEdge SE360 V2)
Chassis Intrusion detection	ThinkEdge servers are equipped with an intrusion switch to detect unexpected removal of server’s cover. If Chassis intrusion detection is enabled in server’s BMC (XClarity Controller), the server will be locked down if the server's top cover is removed
Motion detection	ThinkEdge servers are also equipped with security sensors to detect unexpected movement of the server. When motion detection is enabled in server’s BMC (XClarity Controller), ThinkEdge server will be locked down if server is moved beyond predefined ranges.
Intrusion are/ disarm	The security keylock can be used as an electronic switch to disarm the intrusion switch detection, so that authorized servicing of the hardware can be performed without triggering the security actions.
Integrated password protection	Administrator password and power-on password stored in UEFI ensure that the server will not boot unless the password is entered correctly
Onboard Trusted Platform Module (TPM)	Supports TPM 2.0 and enables advanced cryptographic functionality in the operating system and applications. For users in China, the server has an internal TCM port that supports a Nationz TPM 2.0 module
Support for secure boot	To ensure only immutable and signed software are loaded during the boot time. The use of Secure Boot helps prevent malicious code from being loaded and helps prevent attacks

Table 1: Security Features of a Think Edge Server

9 References

Reference Architecture for Secure Edge Infrastructure with Lenovo ThinkEdge Servers and SUSE Linux Enterprise Micro.

Trademarks and special notices

References in this document to Lenovo products or services do not imply that Lenovo intends to make them available in every country.

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®
ThinkShield®
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.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used Lenovo products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-Lenovo products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by Lenovo. Sources for non-Lenovo list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. Lenovo has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-Lenovo products. Questions on the capability of non-Lenovo products should be addressed to the supplier of those products.

All statements regarding Lenovo future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local Lenovo office or Lenovo authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in Lenovo product announcements. The information is presented here to communicate Lenovo's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard Lenovo benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

Any references in this information to non-Lenovo websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this Lenovo product and use of those websites is at your own risk.