

## Overview

This tutorial guide intends to explain the deployment of SUSE K3s 3-node compact cluster on the Ampere® Altra® platform. The installation of Rancher for K3s cluster management, Longhorn for block storage, Nginx Ingress Controller for ingress, and Prometheus and Grafana for collecting and presenting metrics are described in the following sections.

K3s is a CNCF sandbox project that delivers a lightweight certified Kubernetes distribution. There is SUSE Rancher for managing and running production workloads across the cluster.

The diagram below shows the network overview with four nodes: K3s 3-node cluster and a bastion node.

For running Video-on-Demand service workload on the cluster, use Nginx Ingress Controller instead of Traefik as the ingress service, and the named (from BIND) is deployed on the bastion node for the ingress endpoints of two pods of the Nginx web server and Nginx VOD container.

## Figure 1: Network Overview of SUSE Rancher K3 3-node Cluster



#### Prerequisite:

A DNS service like BIND (named) runs on the bastion node.

## Setup Instructions

Refer to the following steps to install K3s without Traefik:

1. For the servers provisioned with SUSE Linux Enterprise (SLE) Micro 5.2, ensure the drives for NICs are up to date.

If you want to deploy K3s on other Linux distributions, refer to <u>https://www.suse.com/suse-rancher/support-matrix/all-supported-versions/rancher-v2-6-5/</u> for the Rancher K3s support matrix.



2. Deploy K3s on the first node (node1) and disable "traefik." Based on the support matrix, choose K3s version 1.23.6+k3s1.

```
node1 ~$ export INSTALL_K3S_EXEC="server --no-deploy traefik --cluster-init --write-kubeconfig-
mode=644"
node1 ~$ export K3s_VERSION="v1.23.6+k3s1"
node1 ~$ curl -sfL https://get.k3s.io | \
INSTALL_K3S_VERSION=${K3s_VERSION} \
INSTALL_K3S_EXEC=${INSTALL_K3S_EXEC} \
sh -s -
```

3. For the servers provisioned with SLE Micro 5.1 or later, reboot the system when the K3s installer shell script completes the installation.

\$ sudo reboot

4. Retrieve the access token for the K3s cluster when the system is up and running.

5. On the other two nodes (e.g., node2 and node3), clean up the previous installation of K3s, if any.

```
node2 ~$ sudo /usr/local/bin/k3s-uninstall.sh
node2 ~$ sudo reboot
```

6. Add NODE\_TOKEN and FIRST\_SERVER in the script to include the two nodes into the cluster with the first node. Install K3s on the other node.

7. Repeat step 3 (for the servers provisioned with SLE Micro 5.1 or later, reboot the system when the K3s installer shell script completes the installation). Otherwise, the cluster should be up and running.

node2 ~\$ sudo reboot



8. Execute kubect1 to check the nodes' readiness. If their status is "Ready," a 3-node compact cluster is ready!

nodel ⁄	~\$ kubectl	get nodes		
NAME	STATUS	ROLES	AGE	VERSION
nodel	Ready	control-plane,etcd,master	55m	v1.23.6+k3s1
node2	Ready	control-plane,etcd,master	31m	v1.23.6+k3s1
node3	Ready	control-plane,etcd,master	10m	v1.23.6+k3s1

9. To access the k3s cluster from the bastion node, the k3s.yaml is under **/etc/rancher/k3s**. Replicate it under your home directory as **~/.kube/config.** 

```
node1 ~$ cp /etc/rancher/k3s/k3s.yaml ~/kubeconfig
node1 ~$ scp ~/kubeconfig altrauser@10.76.85.104:./.kube/config
node1 ~$ ssh altrauser@10.76.85.104
[altrauser@mtsnow ~]$ kubectl get node
                                           AGE
NAME
       STATUS ROLES
                                                   VERSION
nodel Ready
               control-plane,etcd,master
                                           6h15m
                                                   v1.23.6+k3s1
node2 Ready
               control-plane,etcd,master
                                           5h39m
                                                   v1.23.6+k3s1
node3 Ready control-plane,etcd,master
                                           5h26m
                                                   v1.23.6+k3s1
```

## Install Nginx Ingress Controller on K3s

 Since there is no Traefik as an ingress controller, choose the right version of the Nginx Ingress Controller for supporting K3s. In this PoC, it is version 1.1.0. It will create an ingress-nginx namespace. Also, you will need a patch for this deployment using a DNS load balancer.

```
$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-
v1.1.0/deploy/static/provider/baremetal/deploy.yaml
$ cat > ingress.yaml <<EOF
spec:
template:
spec:
hostNetwork: true
EOF
$ kubectl patch deployment ingress-nginx-controller -n ingress-nginx --patch "$(cat
ingress.yaml)"
deployment.apps/ingress-nginx-controller patched
```

2. In the DNS load balancer use case, increase the number of ingress-nginx-controller pods to 3 by editing the "replicas" to 3.

```
$ kubectl get deployment -n ingress-nginx
NAME READY UP-TO-DATE AVAILABLE AGE
ingress-nginx-controller 1/1 1 1 19s
$ kubectl edit deployment ingress-nginx-controller -n ingress-nginx
...
#
apiVersion: apps/v1
```



```
kind: Deployment
metadata:
 annotations:
   deployment.kubernetes.io/revision: "2"
 generation: 8
 labels:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/version: 1.1.0
    helm.sh/chart: ingress-nginx-4.0.10
 name: ingress-nginx-controller
 namespace: ingress-nginx
. .
spec:
 progressDeadlineSeconds: 600
 replicas: 3
                 #### <--- here
 revisionHistoryLimit: 10
  selector:
   matchLabels:
      app.kubernetes.io/component: controller
      app.kubernetes.io/instance: ingress-nginx
      app.kubernetes.io/name: ingress-nginx
 strategy:
   rollingUpdate:
     maxSurge: 25%
     maxUnavailable: 25%
   type: RollingUpdate
 template:
    metadata:
      creationTimestamp: null
      labels:
        app.kubernetes.io/component: controller
        app.kubernetes.io/instance: ingress-nginx
        app.kubernetes.io/name: ingress-nginx
```

3. Verify the deployment update is complete and ensure an ingress-nginx-controller is on each node.

\$ kubectl get deployment -	-n ingre	ss-nginx						
NAME	READY	UP-TO-DA	ATE AV	AILABLE	AGE			
ingress-nginx-controller	3/3	3	3		1m			
\$ kubectl get pods -n ingr	ess-ngi	nx						
NAME			READY	STATUS		RESTARTS	AGE	
ingress-nginx-admission-cr	reate-jm	5nd	0/1	Complet	ed	0	1m	
ingress-nginx-admission-pa	atch-c8w	t8	0/1	Complet	ed	1	1m	
ingress-nginx-controller-7	/b8496bf	47-59r5q	1/1	Running	ſ	0	50s	
ingress-nginx-controller-7	b8496bf	47-ltp6r	1/1	Running	ſ	0	50s	



ingress-nginx-contro	oller-7b8496bf47-vm5h	ns 1/1	Running	0	1m	
\$ kubectl get pods -	-n ingress-nginx -o v	vide				
NAME	READY	STATUS	RESTARTS AG	E IP		NODE
NOMINATED NODE REA	ADINESS GATES					
ingress-nginx-admiss	sion-create-jm5nd	0/1	Completed	0	2m	10.42.2.4
node3.k3s.hhii.amp	<none> <r< td=""><td>10ne&gt;</td><td></td><td></td><td></td><td></td></r<></none>	10ne>				
ingress-nginx-admiss	sion-patch-c8wt8	0/1	Completed	1	2m	10.42.2.5
node3.k3s.hhii.amp	<none> <r< td=""><td>10ne&gt;</td><td></td><td></td><td></td><td></td></r<></none>	10ne>				
ingress-nginx-contro	oller-7b8496bf47-59r5	5q 1/1	Running	0	1m	10.76.85.101
node1.k3s.hhii.amp	<none> <r< td=""><td>10ne&gt;</td><td></td><td></td><td></td><td></td></r<></none>	10ne>				
ingress-nginx-contro	oller-7b8496bf47-ltp6	5r 1/1	Running	0	1m	10.76.85.102
node2.k3s.hhii.amp	<none> <r< td=""><td>10ne&gt;</td><td></td><td></td><td></td><td></td></r<></none>	10ne>				
ingress-nginx-contro	oller-7b8496bf47-vm5h	ns 1/1	Running	0	2m	10.76.85.103
node3.k3s.hhii.amp	<none> <r< td=""><td>none&gt;</td><td></td><td></td><td></td><td></td></r<></none>	none>				

**Note:** For other instructions, create a new namespace like nginx-ingress for all namespaces or pre-defined namespaces. In this case, the ingress-nginx namespace is created and used.

## **Install Rancher**

In this instruction, use docker to run the rancher v2.6.4 container image on ports 443 and 80 by referring to the following steps.

Note: A bootstrap password is required to access the Rancher web UI.

For the production, run Rancher on a 3-node K8s cluster for High Availability (HA) instead of a single node or VM with docker.

```
$ sudo docker run -d --restart=unless-stopped -p 80:80 -p 443:443 --name rancher26 --
privileged rancher/rancher:latest
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
fbf6db201863 rancher/rancher:latest "entrypoint.sh" 2 minutes ago Up 2 minutes
0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp rancher26
$ docker logs rancher26 2>&1 | grep "Bootstrap Password"
2022/05/11 03:47:24 [INFO] Bootstrap Password: xxxxxxxxxxxxxxxxx
```



## Import K3s Cluster to Rancher

- 1. Login to Rancher web UI. Click Import Existing.
- Figure 2: Rancher Dashboard of Clusters

= 7	RAN	NCHER							- E 🛛
					Welcome to	Rancher	9	· · ·	
Lear	morea	about the impro	ovements and new	capabilities in this version.				What's	new in 2.6
Cluste	ers	1			Import Existing	Create		Community Support	×
State	0	Name 🗘	Provider 🗘	Kubernetes Version	CPU 0	Memory 0	Pods 🗘	Docs	
Act	ve)	local	k3s	v1.22.7+k3s1	0.1/8 cores	70 MiB/16 GiB	5/110	Forums Slack File an Issue	
								Commercial Support	×
								Learn about commercial sup	oport



### 2. Click Generic.

## Figure 3: Import K3s as a Generic Kubernetes to Rancher

Cluster Management	nt				÷ 🔣
<ul> <li>Cloud Credentials</li> <li>Drivers</li> <li>Pod Security Policies</li> <li>RKE1 Configuration</li> <li>Advanced</li> </ul>	1 Register and V Import any I	Import existing cluster in a host Amazon EKS Kubernetes cluster Generic	ted Kubernetes provider Azure AKS	Google GKE	
			+†+		Cancel



#### 3. Enter the Cluster Name and Description.

#### Figure 4: Import Generic Information

Clusters	1						
Cloud Credentials		Cluster: Import Ge	neric				
Drivers							
Pod Security Policies	÷	Import Harvester Clusters via	Virtualization Management				×
Advanced	~						
		Cluster Name * k3s-poc1			Cluster Description 3-node compact K3s Cluster		
							_
		Member Roles					
		Agent Environment Vars	User		Role		
		Labels & Annotations	Default Admin (admin) Local		Cluster Owner		
			Add	+			
				10		_	

4. Refer to the following command to import k3s cluster to the Rancher:

<pre>\$ curlinsecure -sfL https://10.76.85.165/v3/import/</pre>	
xxxxx7wq6rrrpjft4gvzblksc9xzjchvl22xxxx_c-m-9xrzlsg4.yaml   kubectl apply -f -	
clusterrole.rbac.authorization.k8s.io/proxy-clusterrole-kubeapiserver created	ļ
clusterrolebinding.rbac.authorization.k8s.io/proxy-role-binding-kubernetes-master created	
namespace/cattle-system created	
serviceaccount/cattle created	
clusterrolebinding.rbac.authorization.k8s.io/cattle-admin-binding created	
secret/cattle-credentials-b36fb99 created	
clusterrole.rbac.authorization.k8s.io/cattle-admin created	
Warning:	
$\verb spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSetup=0.5$	1
ectorTerms[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. After a few minutes, the cluster appears on the dashboard.

#### Figure 5: List of Clusters on the Dashboard

Clusters	2			Import Exi	sting Create Filter	
State 🔾	Name 🗘	Provider ≎ ← +	Kubernetes Version	CPU 🗘	Memory 0	Pods 🗘
Active	k3s-poc1	k3s	v1.23.6+k3s1	0.2/480 cores	140 MiB/373 GiB	7/330
Active	local	k3s	v1.22.7+k3s1	0.1/8 cores	70 MiB/16 GiB	5/110

6. In the cluster view, there are 3 nodes with the Kubernetes version 1.23.6+k3s1.

#### Figure 6: Nodes Page

≡ 📙 k3s-poc1							Only User Na	nespaces		~	Î	٤١	6	Q	:	w,
Starred Cluster	*	The base Kuberr	netes Node res	ource represents a virtual o	r physical machine w	which hosts deployments. To	manage the mad	hine lifecyle,	, if available	t, go to C	luster Ma	anagemen	<u>L</u>			×
Projects/Namespaces     Nodes	3	Nodes *														
© Cluster Members Workload	~	* Download	YAML	Delete									Filter			
Apps & Marketplace	ř	State 🔿	Name 🗘	Roles 🖯	Version 0	External/Internal IP 🔅	os o	CPU 0		RAM		Pods 🔿		A	ge 🔾	
Storage	~	Active	node1	Control Plane, Etcd	v1.23.6+k3s1	None / 10.76.85.101 👸	Linux		0.4%		2.49	6 (	3.6%	3.3 h	ours	1
More Resources	~	Active	node2	Control Plane, Etcd	v1.23.6+k3s1	None / 10.76.85.102 👸	Linux		0.25%	<u> </u>	2.59	(	1.8%	2.7 h	ours	1
		Active	node3	Control Plane, Etcd	v1.23.6+k3s1	None / 10.76.85.103 👩	Linux	-	0.07%	<u> </u>	2.19		0%	2.5 h	ours	1

## Install Longhorn for Block Storage

- 1. For a clean deployment, clean up the target storage(s) for the Longhorn storage cluster using the following script with root permission.
- **Note:** You can add more storage devices in the loop if needed.

```
$ sudo su -
$ for DISK in "/dev/nvmeln1" "/dev/nvme2n1" "/dev/nvme3n1";
do echo $DISK && \
sgdisk --zap-all $DISK && \
dd if=/dev/zero of="$DISK" bs=1M count=100 oflag=direct,dsync && \
blkdiscard $DISK
done
```



2. Make partitions on target storage(s) and format the partitions in xfs.

```
$ sudo fdisk /dev/nvmeln1
$ sudo fdisk /dev/nvme2n1
$ sudo mkfs.xfs /dev/nvmeln1p1 -f
$ sudo mkfs.xfs /dev/nvme2n1p1 -f
```

3. Create mount points and edit **/etc/fstab** for the mount points.

```
$ sudo mkdir -p /var/lib/longhorn
$ sudo mkdir -p /var/lib/longhorn2
$ sudo vi /etc/fstab
##add the line below
/dev/nvmeln1p1 /var/lib/longhorn xfs defaults 0 1
/dev/nvme2n1p1 /var/lib/longhorn2 xfs defaults 0 1
## save the file and quit vi editor
```

4. Mount the storage devices to the mount points.

\$ sudo mount /var/lib/longhorn/
\$ sudo mount /var/lib/longhorn2/



5. Click Longhorn from Apps & Marketplace > Charts to install Longhorn charts for block storage.

#### Figure 7: Charts Tab



- 6. For servers provisioned with SLE Micro,
  - Ensure open-iscsi has been installed.
  - iscsid daemon is running on all nodes of the Kubernetes cluster.

If not, execute the following command to install:

```
$ sudo transactional-update pkg install open-iscsi
## after installation complete
$ sudo reboot
```



7. Click Install to deploy the Longhorn charts.

#### Figure 8: Longhorn Charts Deployment



8. After choosing the preferred customization on **Step 2** page, click **Install** to complete the deployment. There will be a chart console showing up the deployment process on Rancher.

Figure 9: Longhorn Charts Deployment: Step 2

≡ 📙 k3s-poc1				All Namespaces	~	Ţ	٤		Ø	Q	:	2
uster forkload ops & Marketplace rvice Discovery orace		All charts have at least one we Longhorn 100.1.2+up1.2.4	rsion that is installable on clusters with Linux and Install: Step 2 Change how the App works	Windows nodes unless otherwise indicated. O Metadata							Valu	×
ore Resources	*   	Configure Values used by He Edit Options Edit YAML C Longhorn Images Values Private Registry Settings Longhorn Default Settings Settings Services and Load Balancing Other Settings	Im that help define the App. ampare Changes	Use default Longh	nnimage	5				Vie	w Char	t info
Cluster Tools     6.4		Cancel						Prev	ious	0	Startin	ng



9. Once the deployment is completed, the Longhorn page appears on the left-side navigation bar. Click Longhorn.

#### Figure 10: Longhorn Overview Page

Starred	~	Overview	
Cluster	~	Overview	
Workload	~	Powered By: Longhorn	
Apps & Marketplace	~	Longhorn	ø
Service Discovery	~	Manage storage system via UI	
Storage	~		
Monitoring	~		
Longhorn			
More Resources	~		

10. The Longhorn storage console opens on another tab. Click Node for adding the drives to the Longhorn cluster.

## Figure 11: Dashboard Tab of the Longhorn Console





11. For node 1 from the **Operation** column drop-down, click **Edit node and disks** to add the second drive on each to the Longhorn cluster.

Figure 12: Node Tab of the Longhorn Console

Expand All	Delete Edit Node					N	ame 🗸	
•	Status 🗘	Readiness	Name 💠	Replicas 🌻	Allocated 🗘	Used 🗘	Size ≑	Tags Operat
+ M	Schedulable	Ready	node1.k3s.hhii.amp 10.42.0.7	0	0 / 1303.48 Gi	6.64 / 931.06 Gi	652 Gi +279 Gi Reserved	Edit node and disks
•	Schedulable	Ready	node2.k3s.hhii.amp 10.42.1.5	0	0 / 1303.48 Gi	6.647931.06 Gi	652 GI +279 Gi Reserved	Remove Node
•	Schedulable	Ready	node3.k3s.hhii.amp 10.42.2.8	0	0 / 1303.48 Gi	6.647 931.06 Gi	652 Gi +279 Gi Reserved	≡ ∨



- 12. A dialog window appears for the node.
  - a. Click Add Disk.
  - b. Enter the name of the disk. For example, node1-disk1.
  - c. Enter the path (/var/lib/longhorn2/) for the new drive mounted.
  - d. Select Enable under Scheduling to make the new drive functional in the cluster.
  - e. Click Save.

### Figure 13: Dialog Window for the Node

torage Available		Storage Scheduled	Storage Max	(
24.41Gi		0Gi	931.06Gi	
Name				
default-disk-10307000	00000			
Path				
/var/lib/longhorn/				
torage Reserved		Scheduling	Eviction Requested	
279.32	Gi	Enable      Disable	🔿 True 💿 False	
+ New Disk Tag torage Available IGI Name node1-disk1		Storage Scheduled 0Gi	Storage Max <b>0Gi</b>	t
Path				
/var/lib/longhorn2/				
torage Reserved		Scheduling	Eviction Requested	
D	Gi	🔿 Enable 💿 Disable	🔿 True 💿 False	Û
				Add Disk



13. Once it saves the new drive information, the Node tab displays the updated node's capacity.

Figure 14: Node Tab of the Longhorn Console with the Updated Node's Capacity

4	L	ONG	GHORN	le Da	ashboard	🛛 Node 🗐	Volume	C Recurring Job	🗍 Backup	Setting	g∨				
⊒ Noc	de														
E	xpan	id All	Delete Ed	lit Node								Name			Go
		•	Status	¢	Readiness	N	lame ≑	Replicas	\$ Alloc	ated 🗘	Used 🗢	Siz	e ¢	Tags	Operation
e	Ð	<b>v</b>	Schedu	able	Ready	node1.	.k3s.hhii.am	op 0	0/46	27.69 Gi	19.73 / 2793.16 G	<b>2</b> . +479 G	26 Ti i Reserved		≡ ×
•			Schedu	able	Ready	node2.	.k3s.hhii.am	np 👌	0/13	103.48 Gi	6.64 / 931.06 Gi	65 +279 G	i <b>2 Gi</b> i Reserved		≡ ∨
•	•		Schedu	able	Ready	node3.	.k3s.hhii.am	0 qu	0/13	03.48 Gi	6.64 / 931.06 Gi	65 +279 G	2 Gi i Reserved		$\equiv$ $\vee$
											¢]	4			



14. Repeat steps 11-13 for the other nodes.

## Figure 15: Dialog Window for Other Node

٦	7	LONG	HORN 🛛 🖬 🖻	ashboard	Storage Available	Storage Scheduled	Storage Max	
	_				924.41Gi	0Gi	931.06Gi	
	lode				Name			
					default-disk-103050000	00000		
	Expa	and All	Delete Edit Node		Path			
					/var/lib/longhorn/			
		•	Status 🖨	Readiness	Storage Reserved	Scheduling	Eviction Requested	
	+		Schedulable	Ready	279.32	Gi   Enable   Disable	🔿 True 💿 False	Ť.
	•		Schedulable	Ready	+ New Disk Tag			
			[a		Storage Available	Storage Scheduled	Storage Max	
			Schedulable	Ready	0Gi	0Gi	0Gi	
					Name			
					disk-2			
					Path			
					Path mounted by the	e disk, e.g. /mnt/disk1		
					Storage Reserved	Scheduling	Eviction Requested	
					0	Gi C Enable   Disable	○ True ● False	Û
								Add Disk
<b>(</b> ) 1	124 [			lle File an Issue				Cancel Save



## Figure 16: Node Tab of the Longhorn Console with the Updated Node's Capacity

			-							
Expar	nd All	Delete Edit Nod	e				Ν	ame V		Go
	۰	Status ≑	Readiness	Name 🗘	Replicas 🗘	Allocated 🗘	Used 🗘	Size 🗘	Tags	Operation
+		Schedulable	Ready	node1.k3s.hhii.amp 10.42.0.7	0	0 / 4627.69 Gi	19.73 / 2793.16 Gi	<b>2.26 Ti</b> +479 Gi Reserved		$\equiv$ $\vee$
P.		Schedulable	Ready	node2.k3s.hhii.amp 10.42.1.5	0	0 / 4627.69 Gi	19.73 / 2793.16 Gi	<b>2.26 Ti</b> +479 Gi Reserved		≡ •
	Dis	ks								
		Schedulable	default-disk- 1030500000000	Path: /var/lib/lo	nghorn/	0	0 / 1303.48 Gi	6.64/931.06 Gi	652 Gi +279 Gi Reserved	
		Schedulable	node2-disk1 ←	Path: /var/lib/loo	nghorn2/	0	0/3324.21 Gi	13.08 / 1862.11 Gi	1.62 Ti +200 Gi Reserved	
+		Schedulable	Ready	node3.k3s.hhii.amp 10.42.2.8	0	0 / 1303.48 Gi	6.647931.06 Gi	652 Gi +279 Gi Reserved		$\equiv$ $\vee$



15. Once the new drives are added to the cluster, the **Dashboard** displays the total storage capacity to be scheduled in the Longhorn cluster.

Figure 17: Dashboard Tab of the Longhorn Console with the Total Storage Capacity

LONGHORN	Dashbaard 🖬 Node 🗐 Volu	ume 🕓 Recurring Job 🗍 Backup	o । ◎ Setting∨		
Dashboard					
No Volume		6.72 Storage Scher	Ti Sulable	3 Nodes	
Healthy	0	Schedulable	6.72 Ti	Schedulable	3
😑 Degraded	0	Reserved	1.4 Ti	😑 Unschedulable	0
In Progress	0	Used	59.2 Gi	Oown	0
Fault	0	Disabled	0 Bi	Disabled	0
Detached	0	Total	8.18 Ti	Total	3
2.4 Documentation Generate Support Bui	dle File an Issue Slack				ABRABA

16. If you add new drives for each node later, mount them on **/var/lib/longhorn3** and repeat steps 11- 13 to extend the storage capacity.



## **Install Prometheus and Grafana with PVC**

### Prerequisite:

Ensure Longhorn service is up and running to enable persistent volumes for Prometheus and Grafana.

1. Click **Monitoring** from **Apps & Marketplace > Charts** to install Monitoring charts.

## Figure 18: Charts Tab to Install Monitoring

≡ 📙 k3s-poc1				All Namespaces
Cluster Workload Apps & Marketplace	* *	Charts		
⊕ Charts		All charts have at least one version that is installable on clu	usters with Linux and Windows nodes unless otherwise indicated	L
<ul> <li>Installed Apps</li> <li>Repositories</li> <li>Recent Operations</li> </ul>	3 2 1	All	✓ All Categories	← Filter
Service Discovery Storage Longhorn More Resources	> > > >	Alerting Drivers The manager for third-party webhook receivers used in Prometheus Alertmanager	CIS Benchmark The cis-operator enables running CIS benchmark security scans on a kubernetes cluster	External IP Webhook Deploy the external-ip-webhook to mitigate k8s CVE-2020-8554 Linux only
		Harvester CSI Driver A Helm chart for Harvester CSI driver	Istio A basic Istio setup that installs with the isticctl. Refer to https://istio.io/latest/ for details. Linux only	Collects and filter logs using highly configurable CRDs. Powered by Banzai Cloud Logging Operator.
		Monitoring Collects several diated telm charts, Grafana dashbourds, and Prometheus rules combined with Doc mentation Deploys on Windows	OPA Gatekeeper Modifies Open Policy Agent's upstream gatekeeper chart that provides policy- based control for cloud native Linux only	rancher-wins-upgrader Manages upgrading the wins server version and configuration across all of your Windows nodes
		Windows GMSA	Ambassador Edge Stack	artifactory-ha



2. Click Install to deploy the Monitoring charts.

#### Figure 19: Monitoring Charts Deployment

≡ 📙 k3s-poc1			All Namespaces	1	٤		G	Q	:	ж.
Cluster	~									
Workload	~	All charts have at least one version that is installable on clusters with Linux and Windows nodes unless otherwise indicated.							×	
Apps & Marketplace	~	<b>A</b>							1 constant of the	
Service Discovery	~	Charts: Monitoring (100.1.2+up19.0.3)							Install	
Storage	~	Collects several related Helm charts, Grafana dashboards, and Prometheus rules combined with documentation and scripts to provide easy to op	erate end-to-end Kubernetes cluster monit	oring wit	h Prometh	ieus using	the			
Longhorn		rionaunos operatore								
More Resources	~	This chart is based on the upstream kube-prometheus-stack chart. The chart deploys Prometheus Operator and its CRDs along additional charts / Kubernetes manifests to gather metrics. It allows users to monitor their Kubernetes clusters, view metrics in notifications. For more information on how to use the feature, refer to our docs. The chart installs the following components:	t with Grafana, Prometheus Adapter at n Grafana dashboards, and set up alert:	nd s and	Ch 100 100	art Ver: 1.2+up1	sions 9.0.3 9.0.3 6.6.0	Wed, N Thu, I Thu, I	4ar 30 20 Feb 24 20 Feb 24 20	022 022 022
		<ul> <li>Prometheus Operator - The operator provides easy monitoring definitions for Kubernetes services, manages Prometheus and AlertManage Kubarnetes services, manages Prometheus and AlertManage</li> </ul>	er instances, and adds default scrape target	s for son	e Ap	plicatio	n Versi	ion		
		<ul> <li>kube-promethus - A collection of community-curated Kubernetes manifests, Grafana Dashboards, and PrometheusRules that deploy a de Grafana - Grafana allows a user to create / view dashboards based on the cluster metrics collected by Prometheus.</li> </ul>	efault end-to-end cluster monitoring configu	ration.	0.50	0.0				
		<ul> <li>node-exporter / kube-state-metrics / rancher-pushprox - These charts monitor various Kubernetes components across different Kubernet</li> <li>Prometheus Adapter - The adapter allows a user to expose custom metrics, resource metrics, and external metrics on the default Promethe</li> </ul>	tes cluster types. eus instance to the Kubernetes API Server.		Ho	me				
		For more information, review the Helm README of this chart.			http kub	e-promet	.com/pro heus	ometheu	s-operate	or/
		Upgrading from 100.0.0+up16.6.0 to 100.1.0+up19.0.3			Ma	intaine	rs			
1			t T T		veli	niniaev				

3. Under Edit Options tab enter the preferred customization on Step 2 page. Click Install to complete the deployment.

Figure 20: Monitoring Charts Deployment: Step 2

≡ 📙 k3s-poc1				All Namespaces 🗸	<u>۲</u> ک	a p	: 30
Cluster	All charts have at least one vers	ion that is installable on clusters with Linux and Windows no	ides unless otherwise indicated.				×
Apps & Marketplace Service Discovery	Monitoring I 100.1.2+up19.0.3	Install: Step 2 Change how the App works	O Metadata	Values			O Helm Options
Storage	Configure Values used by Helm	that help define the App.					
	Edit Options Edit YAML Com	npare Changes				V	iew Chart Info
	General Prometheus Alerting	General Cluster Type K3s					~
	Grafana	Create Default Monitoring Cluster Roles O	0	Aggregate to Default Kubernetes Roles O			
© Cluster Tools	Cancel					Previous	Next



- 4. To configure Prometheus option,
  - a. Check Persistent Storage for Prometheus.
  - b. Choose longhorn from the Storage Class Name drop-down.

## Figure 21: Prometheus Configuration

k3s-poc1					All Namespaces	Υ <u>τ</u> ε	
ster rkload	Ĵ,	All charts have at least one w	ersion that is installable on clusters with Linux and Windows	nodes unless otherwise indicated.			×
os & Marketplace vice Discovery	ĩ (	Monitoring 100.1.2+up19.0.3	Install: Step 2 Change how the App works	O Metadata		Values	O Helm Option
age ghorn e Resources	Ĵ		Retention 10d		Retention Size 50GiB		
			Resource Limits				
			Requested CPU 750m		Requested Memory 1750Mi		
			CPU Limit 1000m		Memory Limit 3000Mi		
			Persistent Storage for Prometheus				
			Size 50Gi		Storage Class Name		^
			Access Mode ReadWriteOnce ×	~	local-path longhorn		
			Selector				
			If you are using a dynamic provisioner (e.g. Longhorn)	no Selectors should be specified since a PVC with	a non-empty selector can't ha	ve a PV dynamically provisioned for it	
			Add Rule				
Cluster Tools		Canad					During
4		Cancel					Previous



- 5. To configure Grafana option,
  - a. Check Enable with PVC Template.
  - b. Enter **50Gi** as the **Size** field.
  - c. Choose longhorn from the Storage Class Name drop-down.
  - d. Choose ReadWriteOnce" from the Access Mode drop-down.
  - e. Click Next to Step 3.

## Figure 22: Grafana Configuration

≡ 📙 k3s-poc1					All Namespaces 🗸	± ≥ ∎	Ø Ø	2 <b>(*</b>	
Cluster Workload	> >	All charts have at least one ve	rsion that is installable on clusters with Linux and Windows nodes unless otherwise indi	icated.				×	
Apps & Marketplace Service Discovery	* *	Monitoring 100.1.2+up19.0.3	Install: Step 2 Change how the App works	O Metadata	• Values			O Helm Options	
Longhorn	+++	Configure Values used by He	in that help define the App.				_		
More Resources	- 0	Edit Options Edit YAML Co General Prometheus Alerting Grafana	mpare Changes Grafana Configure Grafana Grafana Storage Disabled Enable with PVC Template Enable with StatefulSet Template Size		Storage Class Name		v	ew Chart Info	
			Access Mode ReadWriteOnce ReadOnlyMany ReadWriteMany PVC Finalizers Add	^	angjon II				
© Cluster Tools		Cancel					Previous	Next	1



6. On Step 3, review the check boxes for additional deployment options and the timeout option. Click Install.

#### Figure 23: Additional Deployment Options

≡ 📙 k3s-poc1				All Namespaces	ŕ ±	٤ 🖬	G	<u>م</u>	125
Cluster Workload	* *	All charts have at least one version that is installable on clusters	with Linux and Windows nodes unless otherwise indicated.						×
Apps & Marketplace Service Discovery Storage	* * *	Monitoring Install: Step 3 100.12+up19.0.3 Change how the app is deployed	O Metadata	O Value	5			Heim	• Options
Longhorn	v	Supply additional deployment options							
n e mandel 623 — «		Apply custom resource definitions     Execute chart hooks     Validate OpenAPI schema     Wait							
		Timeout seconds							
		Description e.g. Purpose of helm command							
					•				
O Cluster Tools		Carvel					Previo		Install
v2.6.4									



7. During installation, the Installed Apps page appears with the installation status.

## Figure 24: Installed Apps Page with Installation Status

≡ 📙 k3s-poc1		All Namespaces 🗸	1 L B G D : 🎽
Cluster Workload	An installed application is a Helm 3 chart that was installed either via our charts or through the Helm CLI.		×
Apps & Marketplace	Installed Apps 🔅		
Installed Apps	± Download YAML ■ Delete		Filter
© Repositories			
Recent Operations	State O Name Q Chart O	Upgradable 🖯	Resources Age 🖯
Service Discovery			
Storage	Namespace: cattle-fleet-system		
Cluster Tools	Deployed fleet-agent-k3s-poc1 fleet-agent-k3s-poc1:0.0.0+s-d33f3ac	Managed	6 5 mins
V2.6.4			
Man, Jul 25 2022 9:07:0 Man, Jul 25 2022 9:07:0	<pre>version=100.1.2+up19.0.3wait=true rancher-monitoring-crd /home/shell/helm/rancher-mon Release "rancher-monitoring-crd" does not exist. Installing it now. am W0725 16:07:08.911773 39 warnings.go:70] policy/vibetal PodSecurityPolicy is deprecated creating is resource(s) am W0725 16:07:09.361733 39 warnings.go:70] policy/vibetal PodSecurityPolicy is deprecated ab Eginning wait for 5 resources with timeout of 10m05 an Starting delete for "rancher-monitoring-crd-create" Job jobs.batch "rancher-monitoring-crd-create" not found a creating 1 resource(s) am Watching for changes to Job rancher-monitoring-crd-create with timeout of 10m05 am fadd/Modify event for rancher-monitoring-crd-create: MDDED am rancher-monitoring-crd-create: Jobs active: 0, jobs succeeded: 0 am Add/Modify event for rancher-monitoring-crd-create: MDDIFIED am rancher-monitoring-crd-create: Jobs active: 1, jobs failed: 0, jobs succeeded: 0</pre>	hitoring-crd-1001.2-up19.03.tgz in v1.21+, unavailable in v1.25+ in v1.21+, unavailable in v1.25+	
Follow Clear I	ownload	Use Previous Contain	er Connected

a. Check Longhorn Dashboard, you will see 2 volumes In Progress.

#### Figure 25: Longhorn Dashboard







8. When the installation is complete, a SUCCESS message appears on the Installed Apps page.

### Figure 26: Installed Apps Page with the Success Message

≡ 📙 k3s-poc1							All Namespaces	~	Ť	٤	6	<u>ب</u> م	25
Cluster Workload Apps & Marketplace	*	installed Apps ☆								Filter			
© Charts		State 🔿	Name 🗘		Chart 🔿			Upgradable 🔾		Reso	ources	Age 🗧	
Installed Apps     Repositories	2												
Recent Operations	2	Deployed	fleet-agent-k3s-poc1		fleet-agent-k3s-poc1:0.0.0+s	i-d33f3ac		Managed		-	6	7 mins	1
Storage Monitoring	* *	Namespace: cattle	-monitoring-system										
Longhorn		Deployed	rancher-monitoring		rancher-monitoring:100.1.2+	up19.0.3				1	63	1.2 mins	1
More Resources	*	Deployed	rancher-monitoring-crd		rancher-monitoring-crd:100.	1.2+up19.0.3		-		_	5	1.5 mins	÷
		Namespace: longhe	orn-system										
		Deployed	longhorn		longhorn:100.1.2+up1.2.4					-	18	4.9 mins	1
Cluster Tools		Deployed	longhorn-crd		longhorn-crd:100.1.2+up1.2.	4		-		_	15	4.9 mins	1
v2.6.4 Install cattle-monitoring-syst	temirano	cher-monitoring 🗵	0										
Non, Jul 25 2022 9:08:38 am REVISION: 1 Non, Jul 25 2022 9:08:38 am Tots SUITE: None Non, Jul 25 2022 9:08:38 am NOTEs: Non, Jul 25 2022 9:08:38 am Kubect — namespace cattle-monitoring-system get pods - L "release=rancher-monitoring" Non, Jul 25 2022 9:08:38 am Non, Jul 25 2022 9:08:38 am													
Follow Clear	Downlo	pad						Use Previous Conta	ainer 💿	Filter		Disco	nnected

9. The Monitoring page appears on the left-side navigation bar. Click Monitoring to access the Dashboard.

#### Figure 27: Dashboard of Monitoring

≡ 📙 k3s-poc1					Only User Namespaces	~	Ţ	٤	ľ	Ō	Q	:	×
Cluster Workload Apps & Marketplace	* * *	Dashboard Powered By: Prometheus											
Service Discovery Storage	* *	Alertmanager	Ø	<b>(</b>	Grafana			G	8				
Monitoring	^	Active Alerts		Metrics Da	ashboards								
Monitors Monitors Koutes and Receivers Advanced Longhorn More Resources	v	Prometheus Graph PromQL Graph	Ø	Configured	Prometheus Rules d Rules			G	7				
PINE RESOLICES		Configured Targets	Ø										



### 10. Click Grafana.

11. The Grafana web application opens on a new tab of the browser.

## Figure 28: Grafana Web Console

ĝ	88 General / Home										9	
Q B	Welcome to Grafana		þ			Need help? Documentation Tutorials Community Public Stack						
	CPU Utilization		Memory Utilization				Disk Utilization					
	CPU Used O cores	CPU Total O cores	Memory Use	1	Memor (	ry Total D	Disk Used O B			Disk T	otal 3	
	CPU Usage			Memo	femory Usage Disk Usage							
	50% No data		0.500% 0%	No	o data		0.500 0. 0. 0.					
	-100% OB:10 OB:20 OB:30 OB:40	08.50 09.00	-1% 08:10 08:20	08:30	08:40 08:50	09:00	08:10 08:20 — Cluster	08:30	08:40	08:50	09:00	
	Recently viewed dashboards	Dashboards			About Rancher Monitoring Rancher Monitoring is a Heim chard developed by Rancher that is powered by Prometheus Operator. It is based on the upstream						n the upstream	
	Kubernetes / Compute Resources / Workload		kude prometheus stack Herm Chart maintained by the Prometheus Community.     By default, the chart deploys Crafana alongside a set of Grafana dashboards curated by the kube prometheus project.									
Ð	Kubernetes / Compute Resources / Node (Pods)	(Pods)		For more information rancher-monitoring of	n on how Rancher M chart located in the	fonitoring differs from kube-pro rancher/charts repository.	metheus-stack, pl	¢. ₹	the CHANGE	LOG.md of the		
0	Rancher / etcd			¢	For more information	For more information about how to configure Rancher Monitoring, please view the Rancher docs.						



12. After a few minutes, the charts display the history of telemetries and utilizations.

#### Figure 29: Grafana Charts with Telemetries and Utilizations Data



#### Notes:

• If you want to re-install K3s on any node, clean up the node by uninstalling K3s and the drives for Longhorn storage service:

```
$ sudo su -
$ /usr/local/bin/k3s-uninstall.sh
$ umount /dev/nvmeln1 /dev/nvme2n1 /dev/nvme3n1
$ for DISK in "/dev/nvmeln1" "/dev/nvme2n1" "/dev/nvme3n1";
do echo $DISK && \
sgdisk --zap-all $DISK && \
dd if=/dev/zero of="$DISK" bs=1M count=100 oflag=direct,dsync && \
blkdiscard $DISK
done
```

• If you install K3s on SLE Micro v5.1 or later, reboot the system to complete the transaction update.

\$ reboot

- The cluster is gracefully shutdown by the following steps:
  - a. Login to the nodes through SSH or cockpit web console.



b. For root privileges, execute the sudo command.

\$ sudo su -#

c. Shutdown K3s on each node.

# k3s-killall.sh

- d. Wait for the script to complete its jobs.
- e. Shutdown the nodes by entering:

#	poweroff
or	
#	shutdown -h now

The rancher console displays the state of the cluster is **Unavailable**.

## Figure 30: Rancher Console

=	TRANCHER					: 🗯
				Welcome to Rancher	•	
	Learn more about the improvements	and new capabilities in this ver	rsion.			What's new in 2.6
	Clusters 2				Import Existin	g Create Filter
	State 🗘 Name 🗘	Provider 0	Kubernetes Version	CPU 0	Memory 0	Pods 🗘
	Unavailable k3s-poc1 Cluster agent is not connected	k3s	v1.23.6+k3s1	97.31/384 cores	2.71 GiB/752 GiB	83/330
	(Active local	k3s	v1.22.7+k3s1	0.1/8 cores	70 MiB/31 GiB	5/110



# Document Revision History

ISSUE	DATE	DESCRIPTION
1.00	November 7, 2022	Initial release.



November 7, 2022

Ampere Computing reserves the right to change or discontinue this product without notice.

While the information contained herein is believed to be accurate, such information is preliminary, and should not be relied upon for accuracy or completeness, and no representations or warranties of accuracy or completeness are made.

The information contained in this document is subject to change or withdrawal at any time without notice and is being provided on an "AS IS" basis without warranty or indemnity of any kind, whether express or implied, including without limitation, the implied warranties of non-infringement, merchantability, or fitness for a particular purpose.

Any products, services, or programs discussed in this document are sold or licensed under Ampere Computing's standard terms and conditions, copies of which may be obtained from your local Ampere Computing representative. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Ampere Computing or third parties.

Without limiting the generality of the foregoing, any performance data contained in this document was determined in a specific or controlled environment and not submitted to any formal Ampere Computing test. Therefore, the results obtained in other operating environments may vary significantly. Under no circumstances will Ampere Computing be liable for any damages whatsoever arising out of or resulting from any use of the document or the information contained herein.



Ampere Computing 4655 Great America Parkway, Santa Clara, CA 95054 Phone: (669) 770-3700 https://www.amperecomputing.com

Ampere Computing reserves the right to make changes to its products, its datasheets, or related documentation, without notice and warrants its products solely pursuant to its terms and conditions of sale, only to substantially comply with the latest available datasheet.

Ampere, Ampere Computing, the Ampere Computing and 'A' logos, Altra, and eMAG are registered trademarks of Ampere Computing.

Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All other trademarks are the property of their respective holders.

Copyright © 2022 Ampere Computing. All rights reserved.