

## Here When You Need Us

# How to shutdown a Kubernetes cluster (Rancher Kubernetes Engine (RKE) CLI provisioned or Rancher v2.x Custom clusters)

This document **(000020031)** is provided subject to the [disclaimer](#) at the end of this document.

## Situation

### Task

This article provides instructions for safely shutting down a Kubernetes cluster provisioned via the Rancher Kubernetes Engine (RKE) CLI or a Rancher v2.x provisioned Custom Cluster.

### Requirements

- A Kubernetes cluster launched with the RKE CLI or from Rancher 2.x as a Custom Cluster

### Background

If you have a need to shut down the infrastructure running a Kubernetes

### How We Use Cookies



We use cookies to give you a better experience, improve performance and analyze traffic. By using our website you agree to our use of cookies.

**N.B.** If you have nodes that share worker, control plane, or etcd roles, postpone the `docker stop` and shutdown operations until worker or control plane containers have been stopped.

## Draining nodes.

For all nodes, prior to stopping the containers, run:

```
kubectl get nodes
```

To identify the desired node, then run:

```
kubectl drain <node name>
```

This will safely evict any pods, and you can proceed with the following steps to a shutdown.

## Shutting down the workers nodes

For each worker node:

1. ssh into the worker node
2. stop kubelet and kube-proxy by running `sudo docker stop kubelet kube-proxy`
3. stop docker by running `sudo service docker stop` or `sudo systemctl stop docker`
4. shutdown the system `sudo shutdown now`

## Shutting down the control plane nodes

For each control plane node:

1. ssh into the control plane node
2. stop kubelet and kube-proxy by running `sudo docker stop kubelet kube-proxy`
3. stop kube-scheduler and kube-controller-manager by running `sudo docker stop kube-scheduler kube-controller-manager`
4. stop kube-apiserver by running `sudo docker stop kube-apiserver`
5. stop docker by running `sudo service docker stop` or `sudo systemctl stop docker`
6. shutdown the system `sudo shutdown now`

## Shutting down the etcd nodes

For each etcd node:

1. ssh into the etcd node

2. stop kubelet and kube-proxy by running `sudo docker stop kubelet kube-proxy`
3. stop etcd by running `sudo docker stop etcd`
4. stop docker by running `sudo service docker stop` or `sudo systemctl stop docker`
5. shutdown the system `sudo shutdown now`

### Shutting down storage

Shut down any persistent storage devices that you might have in your datacenter (such as NAS storage devices) if applicable. It is important that you do this after shutting everything else down to prevent data loss/corruption for containers requiring persistency.

**N.B.** If you are running a cluster that was not deployed through RKE then the order of the process is still the same, however the commands may vary. For instance, some distributions run kubelet and other control plane items as a service on the node rather than in docker. Check documentation for the specific Kubernetes distribution for information as to how to stop these services.

### Starting a Kubernetes cluster up after shutdown

Kubernetes is good about recovering from a cluster shutdown and requires little intervention, though there is a specific order in which things should be powered back on to minimize errors.

1. Power on any storage devices if applicable.

Check with your storage vendor on how to properly power on your storage devices and verify that they are ready.

2. For each etcd node:

1. Power on the system/start the instance.
2. Log into the system via ssh.
3. Ensure docker has started `sudo service docker status` or `sudo systemctl status docker`
4. Ensure etcd and kubelet's status shows Up in Docker `sudo docker ps`

3. For each control plane node:

1. Power on the system/start the instance.
2. Log into the system via ssh.
3. Ensure docker has started `sudo service docker status` or `sudo systemctl`

`status docker`

4. Ensure kube-apiserver, kube-scheduler, kube-controller-manager, and kubelet's status shows Up in Docker `sudo docker ps`
4. For each worker node:
  1. Power on the system/start the instance.
  2. Log into the system via ssh.
  3. Ensure docker has started `sudo service docker status` or `sudo systemctl status docker`
  4. Ensure kubelet's status shows Up in Docker `sudo docker ps`
5. Log into the Rancher UI (or use kubectl) and check your various projects to ensure workloads have started as expected. This may take a few minutes depending on the number of workloads and your server capacity.

## Disclaimer

---

This Support Knowledgebase provides a valuable tool for SUSE customers and parties interested in our products and solutions to acquire information, ideas and learn from one another. Materials are provided for informational, personal or non-commercial use within your organization and are presented "AS IS" WITHOUT WARRANTY OF ANY KIND.

Document ID:

**000020031**

---

Creation Date:

**27-Sep-2022**

---

Modified Date:

**27-Sep-2022**

---

**SUSE Rancher**

---

[< Back to Support Search](#)

For questions or concerns with the SUSE Knowledgebase please contact:  
tidfeedback[at]suse.com



# SUSE Support Forums

Get your questions answered by experienced Sys Ops or interact with other SUSE community experts.

[Join Our Community](#)

## Support Resources

Learn how to get the most from the technical support you receive with your SUSE Subscription, Premium Support, Academic Program, or Partner Program.

[SUSE Customer Support Quick Reference Guide](#)

[SUSE Technical Support Handbook](#)

[Update Advisories](#)

[Support FAQ](#)

## Open an Incident

Open an incident with SUSE Technical Support, manage your subscriptions, download patches, or manage user access.

[Go to Customer Center](#)

**[Report a Software Vulnerability](#)**

**[Submit Tips, Tricks, and Tools](#)**



Support: [Open a Support Case](#)