

**Rancher**

# Kubeflow with Rancher

Deploying Kubeflow onto an RKE2 cluster with Rancher

Rancher Prime by SUSE  
Rancher Kubernetes Engine 2  
Harvester by SUSE  
SUSE Linux Enterprise Server

Alex Arnoldy, Solutions Architect, Integrated Systems (SUSE)  
Mark Gonnelly, Solutions Architect, GSI/IHV (SUSE)  
Terry Smith, Director, Global Partner Solutions (SUSE)

# Kubeflow with Rancher

## Deploying Kubeflow onto an RKE2 cluster with Rancher

**Date:** 2023-06-15

### Summary

Kubeflow simplifies deployment of machine learning (ML) workflows on Kubernetes clusters. This document provides step-by-step guidance for deploying Kubeflow on an RKE2 cluster with Rancher in a Harvester hyperconverged infrastructure.

### Disclaimer

Documents published as part of the series SUSE Technical Reference Documentation have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

# Contents

- 1 Introduction 4
- 2 Prerequisites 5
- 3 Preparing the Cluster 7
- 4 Installing Kubeflow 14
- 5 Securing Kubeflow with TLS 17
- 6 Summary 25
- 7 Legal notice 27
- 8 GNU Free Documentation License 28

# 1 Introduction

Machine learning (ML) is driving innovation across many domains. The importance of ML is reflected in its growing adoption by enterprises. Businesses use ML to achieve deeper insights, grow capabilities, improve efficiencies, and accelerate results.

ML workflows can be complex, and managing them can be difficult. This is particularly true at production scale. [Kubeflow \(https://www.kubeflow.org/\)](https://www.kubeflow.org/) is an open source project that aims to make deploying and scaling ML models as simple as possible. To achieve this, Kubeflow takes a cloud-native approach, deploying workloads in containers on [Kubernetes \(https://kubernetes.io\)](https://kubernetes.io) clusters.

[Rancher Prime by SUSE \(https://www.suse.com/solutions/enterprise-container-management/#rancher-product\)](https://www.suse.com/solutions/enterprise-container-management/#rancher-product) empowers organizations to unify their Kubernetes landscape with secure, streamlined management. By deploying Kubeflow into a Rancher-managed Kubernetes landscape, you can address the operational and security challenges of managing ML workflows at scale.

The modern, enterprise Kubernetes landscape leverages computing, storage, and networking resources in a variety of environments. [Harvester by SUSE \(https://www.suse.com/products/harvester/\)](https://www.suse.com/products/harvester/) is a next-generation, open source, hyperconverged infrastructure (HCI) solution designed for modern cloud-native environments. Harvester provides operators with a cohesive platform to manage virtual machine and container workloads such as Kubeflow.


## 1.1 Scope



In this guide, you:

1. provision virtual machines for your Kubernetes cluster nodes with [Harvester by SUSE \(https://www.suse.com/products/harvester/\)](https://www.suse.com/products/harvester/).



## Note

Kubeflow can be deployed to any CNCF-certified Kubernetes cluster, regardless of how it is provisioned. Harvester simply provides a convenient mechanism for provisioning and managing IT infrastructure. Additionally Harvester provisions [Longhorn](https://longhorn.io/docs/) (<https://longhorn.io/docs/>)  for persistent storage to support your Kubernetes deployment.

2. use [Rancher Prime by SUSE](https://www.suse.com/solutions/enterprise-container-management/#rancher-product) (<https://www.suse.com/solutions/enterprise-container-management/#rancher-product>)  2.7 to instantiate and configure an [RKE2](https://docs.rke2.io/) (<https://docs.rke2.io/>)  v1.24.9 + rke2r2 cluster.
3. use Rancher to deploy Kubeflow v1.6.1 onto your cluster.
4. secure Kubeflow with TLS.

After completing these steps, you can access the Kubeflow user interface through your Web browser.



## Note

The steps outlined in this guide should require little to no modification for newer versions of the component software.

## 1.2 Audience

This document is intended for data scientists, machine learning researchers and developers, operational teams, and others with responsibility for ensuring the success of ML projects.




To be successful with this guide, you should have basic knowledge of Kubernetes operations in a Rancher environment. In addition, you should have basic knowledge of standard Linux command line container management with Kubectl, Helm, and Kustomize.

## 2 Prerequisites

To follow the instructions of this guide, you need the following:


- Management workstation

Your workstation must have access to the cluster environment and have these software tools:

- Kubectl (<https://kubectl.docs.kubernetes.io/installation/kubectl/>) 
- Helm (<https://helm.sh/docs/intro/install/>) 
- Kustomize (<https://kubectl.docs.kubernetes.io/installation/kustomize/>) 



## Note

The operating system of the workstation used in the development of this guide was [SLES](https://www.suse.com/products/server/)  15 SP4.

- Cluster infrastructure

The minimum recommended cluster consists of five nodes as follows:

- Control plane and etcd pool nodes  
Three nodes with vCPUs, 8 GB of RAM, and 40 GB storage volume.
- Workload nodes  
Two nodes with 8 vCPUs, 16 GB of RAM, and 40 GB storage volume.
- Operating system  
All nodes are configured with SLES 15 SP4 minimal, using the QCOW2 image with cloud-init enabled (previously known as the OpenStack image).



## Important

The SSH user for the operating system image is sles.

- Networking  
All nodes are connected to a single VLAN with DHCP, DNS, and routing to the Internet.



## Note

The node sizing used here is for basic testing purposes only. To support a useful Kubeflow workload, you will likely need more vCPU cores, RAM, and storage, particularly for the workload nodes.

# 3 Preparing the Cluster

## 3.1 Deploy an RKE2 cluster with Harvester

Begin by using the Rancher UI to create a project in Harvester named *kubeflow-on-harvester* that contains the *kubeflow-cluster* namespace.

1. Login to the Harvester UI and go to the *Virtual Machines* page.
2. Choose the option to create either multiple VM instances.
3. Set the namespace of your VMs to *kubeflow-cluster*.
4. *VM Name* is a required field.
5. Configure the virtual machines' CPU and memory as specified in the prerequisites.
6. Select SSH keys or upload new keys.
7. Select a custom VM image on the *Volumes* tab.

The default disk will be the root disk. You can add more disks to the VM if you like.

8. To configure networks, go to the *Networks* tab.

The Management Network is added by default, you can remove it if the VLAN network is configured. You can also add additional networks to the VMs using VLAN networks. You may configure the VLAN networks in *Advanced > Networks*.

9. Advanced options such as run strategy, operating system type, and cloud-init data are optional. You may configure these in the *Advanced Options* section if applicable.
10. Select the check box to *Install guest agent*.

The following **User Data cloud-config** (under *Show Advanced*) should be applied to all nodes during RKE2 cluster creation:

```
### cloud-init
#cloud-config
chpasswd:
  list: |
    root:SUSE
    sles:SUSE
  expire: false
ssh_authorized_keys:
  - >-
    <REPLACE WITH SSH PUBLIC KEY OF THE WORKSTATION>
runcmd:
  - SUSEConnect --url <REPLACE WITH RMT SERVER ADDRESS>
  - zypper -n in -t pattern apparmor
  - zypper -n up
  - zypper in --force-resolution --no-confirm --force kernel-default
  - zypper rm kernel-default-base
```

Configure the Kubernetes cluster as follows:

1. On the *Basic* tab:
  - a. Select Kubernetes version v1.24.9 + rke2r2.



### Note

A slightly older version of Kubernetes is used for Harvester Cloud Provider support.

If you are not using Harvester, you can select a newer version.

- b. Enable the Harvester Cloud Provider CSI driver.
    - c. Set *Container Network Interface* to Calico.
    - d. Ensure the *Default Security Pod Policy* is set to Default - RKE2 Embedded.



- e. Leave *Pod Security Admission Configuration Template* set to (None).
  - f. Disable the Nginx Ingress controller under *System Services*.
2. On the *Labels and Annotations* tab, apply a cluster label where the key is app and the value is kubeflow.
  3. Click *Create*.

After the cluster has been created, verify and reboot the RKE2 nodes:

1. SSH to each node as the user sles.
2. Verify that the kernel-default kernel has been installed and kernel-default-base kernel has been removed.

```
sudo zypper se kernel-default
```



### Tip

After kernel-default has been installed, you can remove the kernel-default-base kernel.

```
sudo zypper rm kernel-default-base
```

3. Verify that all operating system software has been patched to the latest update.

```
sudo zypper up
```

4. Reboot each node to enable the kernel-default kernel.

After the RKE2 cluster has been created, download the kubeconfig file from the Rancher Management server to your workstation. See [Accessing Clusters with kubectl from Your Workstation](https://ranchermanager.docs.rancher.com/how-to-guides/new-user-guides/manage-clusters/access-clusters/use-kubectl-and-kubeconfig#accessing-clusters-with-kubectl-from-your-workstation) (<https://ranchermanager.docs.rancher.com/how-to-guides/new-user-guides/manage-clusters/access-clusters/use-kubectl-and-kubeconfig#accessing-clusters-with-kubectl-from-your-workstation>)<sup>7</sup>.

## 3.2 Configure the MetalLB load balancer

**MetalLB** is a network load balancer implementation for Kubernetes clusters on bare metal. This allows you to create Kubernetes services of the type `LoadBalancer` to provide two important features:

- *address allocation*: MetalLB will take care of assigning individual IP addresses from configured address pools as services are launched and reclaiming those addresses when the services end.
- *external announcement*: MetalLB uses standard network or routing protocols to announce that the IP address is in use by a service.

### 3.2.1 Deploy MetalLB

1. Open a terminal on your workstation.
2. Pull and apply the MetalLB manifests.

```
kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.12.1/manifests/namespace.yaml
kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.12.1/manifests/metallb.yaml
# On first install only
kubectl create secret generic -n metallb-system memberlist --from-literal=secretkey="$(openssl rand -base64 128)"
```

3. Specify IP address ranges for default and reserved pools.  
MetalLB assigns IP addresses to services from these pools.

- a. Define the default IP pool.

MetalLB will automatically assign IP addresses from the default pool to services. Specify the starting and ending IP address for this default pool by replacing the addresses shown in the following commands with the appropriate addresses for your environment.

```
export DEFAULT_IP_RANGE_START=10.0.0.10
export DEFAULT_IP_RANGE_END=10.0.0.40
```

- b. Define the reserved IP pool.

Reserved IP addresses are not autoassigned by MetalLB.

```
export RESERVED_IP_RANGE_START=aa.bb.cc.dd
export RESERVED_IP_RANGE_END=ee.ff.gg.hh
```

#### 4. Create the MetalLB configuration file for layer 2 routing.

For other routing options, see [MetalLB: Layer 2 Routing \(https://metallb.universe.tf/configuration/#layer-2-configuration\)](https://metallb.universe.tf/configuration/#layer-2-configuration) and [MetalLB: Example Configuration \(https://raw.githubusercontent.com/google/metallb/v0.9.3/manifests/example-config.yaml\)](https://raw.githubusercontent.com/google/metallb/v0.9.3/manifests/example-config.yaml).

```
cat <<EOF> metallb-config.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: metallb-system
  name: config
data:
  config: |
    address-pools:
    - name: default
      protocol: layer2
      addresses:
      - ${DEFAULT_IP_RANGE_START}-${DEFAULT_IP_RANGE_END}
    - name: rsvd
      protocol: layer2
      auto-assign: false
      addresses:
      - ${RESERVED_IP_RANGE_START}-${RESERVED_IP_RANGE_END}
EOF
```

#### 5. Create the ConfigMap.

```
kubectl apply -f metallb-config.yaml
```

#### 6. Verify the configuration was applied correctly.

```
kubectl get configmap config -n metallb-system -o yaml
```



### Important

Be sure to verify the IP address pools.

#### 7. Verify MetalLB is running.

```
kubectl get all -n metallb-system
```

### 3.2.2 Validate MetalLB and the Harvester/Longhorn CSI

1. Open a command terminal on your workstation.
2. Set the `NAMESPACE` variable to the target namespace.

```
export NAMESPACE="kubeflow"
```

3. Create the namespace.

```
kubectl create namespace ${NAMESPACE}
```

4. Create a manifest for the nginx pod, PVC, and load balancer service.

```
cat <<EOF> nginx-metallb-test.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  namespace: ${NAMESPACE}
spec:
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1
          ports:
            - name: http
              containerPort: 80
          volumeMounts:
            - mountPath: /mnt/test-vol
              name: test-vol
      volumes:
        - name: test-vol
          persistentVolumeClaim:
            claimName: nginx-pvc
---
kind: PersistentVolumeClaim
apiVersion: v1
```

```

metadata:
  name: nginx-pvc
  namespace: ${NAMESPACE}
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi

---
apiVersion: v1
kind: Service
metadata:
  name: nginx
  namespace: ${NAMESPACE}
spec:
  ports:
    - name: http
      port: 8080
      protocol: TCP
      targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
EOF

```

5. Apply the manifest to create the nginx pod, PVC, and load balancer service.

```
kubectl apply -f nginx-metallb-test.yaml
```

6. Verify the pod is "Running", that the harvester StorageClass is the (default), that the persistentvolumeclaim is Bound, and that the service has an external IP address.

```
kubectl get pod,sc,pvc,svc -n ${NAMESPACE}
```

7. Verify that the service is reachable through the load balancer IP address from outside the cluster.

```

IPAddr=$(kubectl get svc -n ${NAMESPACE} | grep -w nginx | awk '{print$4":"$5}' |
awk -F: '{print$1":"$2}')
curl http://${IPAddr} 2>/dev/null | grep "Thank you for using nginx"

```

An HTML encoded output should display the phrase "Thank you for using nginx."

8. Verify that the volume is mounted in the test pod.

```
TEST_POD=$(kubectl get pods -n ${NAMESPACE} | awk '/nginx/ {print$1}')
kubectl exec -it ${TEST_POD} -n ${NAMESPACE} -- mount | grep test-vol
```

The output should show that the volume is mounted at the location /mnt/test-vol.

9. When finished with testing, delete the pod and service.

```
kubectl delete -f nginx-metallb-test.yaml
```

## 4 Installing Kubeflow

The Kubeflow Manifests Working Groups provides two options for installing Kubeflow:

- A single command that installs all components and targets ease of deployment for users. This is the recommended method and is the method illustrated in this guide.
- A multi-command approach that allows individual components to be included or excluded.

### 4.1 Download the Kubeflow manifests

Obtain the manifests for Kubeflow v1.6.1 from the project's GitHub repository.

1. Open a terminal on your workstation.
2. Create and change to a directory into which you will clone the Kubeflow manifests.

```
mkdir $HOME/kubeflow
cd $HOME/kubeflow
```

3. Clone the [Kubeflow manifests \(https://github.com/kubeflow//manifests\)](https://github.com/kubeflow//manifests) repository for the desired release.

```
git clone git@github.com:kubeflow/manifests.git --branch v1.6.1
```



#### Note

This guide is verified with Kubeflow v1.6.1. If you want to try a different version, select it from the [available releases \(https://github.com/kubeflow//manifests/releases\)](https://github.com/kubeflow//manifests/releases) and update the above command with the appropriate release tag.

4. Change to the new manifests directory on your workstation.

```
cd manifests
```

5. Keep the terminal open and proceed to the next section.

## 4.2 Run the installer



### Important

Kubeflow is in active development. Consult the latest [Kubeflow Installation \(https://github.com/kubeflow//manifests#install-individual-components\)](https://github.com/kubeflow//manifests#install-individual-components) [↗](#) documentation prior to proceeding.

In your terminal, issue the following command to download and install Kubeflow:

```
while ! kustomize build example | awk '!/well-defined/' | kubectl apply -f -; do echo "Retrying to apply resources"; sleep 10; done
```



### Note

The `kubectl apply` command may fail on the first try. Hence a while loop is used to retry until success is achieved.

## 4.3 Validating the Kubeflow installation

It can take some time for all the Kubeflow components to deploy their pods. Check that all Kubeflow-related pods are Ready with the following commands:

```
kubectl get pods -n cert-manager
kubectl get pods -n istio-system
kubectl get pods -n auth
kubectl get pods -n knative-eventing
kubectl get pods -n knative-serving
kubectl get pods -n kubeflow
kubectl get pods -n kubeflow-user-example-com
```

To further validate:

1. Enable port-forwarding with kubectl in a terminal.

```
kubectrl port-forward svc/istio-ingressgateway -n istio-system 8080:80
```

2. Open a Web browser on your workstation to <http://127.0.0.1:8080>




## Note

Use the HTTP connection protocol, not HTTPS.

3. Log in to the Kubeflow user interface (UI) with the credentials:
  - Email address: [user@example.com](mailto:user@example.com)
  - Password: [12341234](#)
4. After verifying that you can log in to the Kubeflow UI, you can simply close your browser.
5. Close the `kubectrl` port-forward session by issuing `Ctrl+C` in the terminal.

## 4.4 Troubleshooting the Kubeflow installation

Some things that could prevent connecting or logging in to the Kubeflow UI include:

- Your local copy of the <https://github.com/kubeflow/manifests>  is out-of-date.  
Update your local copy by cloning a newer release of the Kubeflow manifests.
- The Kubeflow installation has not completed or failed to complete.  
Ensure all containers and pods are in a 'Running' state (see [Section 4.3, "Validating the Kubeflow installation"](#)).  
A high number of container restarts can indicate other issues preventing the installation from completing successfully.
- Cluster resources are saturated.  
Right-sizing a cluster can be challenging. The Harvester UI can help you identify if the physical nodes are overburdened or experiencing failures. You can also check running processes with the Linux `top` command on each of the worker nodes to determine if node CPU and memory resources are overburdened.



## 5 Securing Kubeflow with TLS

Up to this point, you have been able to access the Kubeflow UI over HTTP from your workstation configured to connect to the cluster. In most cases, you will want to enable access to Kubeflow from additional systems. This should be done securely through [HTTPS](https://en.wikipedia.org/wiki/HTTPS) (<https://en.wikipedia.org/wiki/HTTPS>) with a [TLS](https://en.wikipedia.org/wiki/Transport_Layer_Security) ([https://en.wikipedia.org/wiki/Transport\\_Layer\\_Security](https://en.wikipedia.org/wiki/Transport_Layer_Security)) certificate. Moreover, many Kubeflow components leverage this cryptographic protocol for secure cookies and may not function without it.

To overcome this, you need to configure [Istio](https://istio.io/) (<https://istio.io/>) (which was installed with Kubeflow) to use MetalLB with the HTTPS protocol on the [Istio Gateway](https://istio.io/latest/docs/reference/config/networking/gateway/) (<https://istio.io/latest/docs/reference/config/networking/gateway/>). You will also need a signed TLS certificate.

If you do not have a signed TLS certificate, you can get one for free through [Let's Encrypt](https://letsencrypt.org/) (<https://letsencrypt.org/>). **Let's Encrypt** uses a [challenge process](https://letsencrypt.org/docs/challenge-types/) (<https://letsencrypt.org/docs/challenge-types/>) to make sure you actually control the domain for which you are requesting a certificate. One of these challenge methods uses [DNS-01](https://letsencrypt.org/docs/challenge-types/#dns-01-challenge) (<https://letsencrypt.org/docs/challenge-types/#dns-01-challenge>). You can satisfy this challenge using AWS Route 53 and other DNS providers (<https://community.letsencrypt.org/t/dns-providers-who-easily-integrate-with-lets-encrypt-dns-validation/86438>).

### 5.1 Update Istio to use the MetalLB

1. Verify the current `istio-ingressgateway` service type.

It is likely `ClusterIP`.

```
kubectl -n istio-system get svc istio-ingressgateway -o jsonpath='{.spec.type}' ;  
echo ""
```

2. Patch the service to change the type to `LoadBalancer`.

```
kubectl -n istio-system patch svc istio-ingressgateway -p '{"spec": {"type":  
  "LoadBalancer"}}'
```

3. Verify the service is now type of `LoadBalancer` and take note of the IP address.

```
kubectl -n istio-system get svc istio-ingressgateway
```

## 5.2 Enable HTTPS on the KubeFlow Istio Gateway

1. Edit the kubeflow-gateway resource to add HTTPS routing.

```
kubectl edit -n kubeflow gateways.networking.istio.io kubeflow-gateway
```

2. Add the following to the bottom of the `spec:` section.

```
  tls:
    httpsRedirect: false
- hosts:
  - "*"
  port:
    name: https
    number: 443
    protocol: HTTPS
  tls:
    mode: SIMPLE
    credentialName: kubeflow-certificate-secret
```

When complete, the entire `spec:` section look like:

```
spec:
  selector:
    istio: ingressgateway
  servers:
  - hosts:
    - '*'
    port:
      name: http
      number: 80
      protocol: HTTP
    tls:
      httpsRedirect: false
  - hosts:
    - "*"
    port:
      name: https
      number: 443
      protocol: HTTPS
    tls:
      mode: SIMPLE
      credentialName: kubeflow-certificate-secret
```

3. Update AWS Route53 (or the DNS provider you use) with the KubeFlow IP address and the desired fully qualified domain name (FQDN) for the KubeFlow UI.

4. Use your browser to connect to the Kubeflow UI (over the HTTP protocol) at the specified FQDN.

Your browser should redirect to a login prompt.



### Note

For on-premises deployments, Kubeflow uses [Dex \(https://dexidp.io/\)](https://dexidp.io/) as a federated OpenID connection provider that can be integrated with a wide variety of identity providers, such as LDAP, SAML, and OAuth.

5. Log in to the Kubeflow UI with the credentials:

- Email address: user@example.com
- Password: 12341234



### Important

Proceed to the next section only after you have verified that you can connect to and log in to the Kubeflow UI.

## 5.3 Configure cert-manager with a Let's Encrypt staging certificate

**cert-manager** (<https://cert-manager.io/>) is a powerful X.509 certificate controller for Kubernetes workloads and can manage certificates from many public certificate issuers. You can configure cert-manager for [Automated Certificate Management Environment \(ACME\)](https://cert-manager.io/docs/configuration/acme/) (<https://cert-manager.io/docs/configuration/acme/>) issuers, like Let's Encrypt. Use the steps outlined below to configure cert-manager to use Let's Encrypt certificates validated through [DNS-01](https://cert-manager.io/docs/configuration/acme/dns01/route53/) and [AWS Route53](https://cert-manager.io/docs/configuration/acme/dns01/route53/) (<https://cert-manager.io/docs/configuration/acme/dns01/route53/>).



### Note

An AWS user with appropriate IAM policies and API access keys is needed for cert-manager to access the Route53 DNS records.

1. Set some variables to simplify later commands.

```
# aws_access_key_id and aws_secret_access_key for the configured AWS user:
```

```

export AWS_ACCESS_KEY_ID=""           # valid AWS access key ID
export AWS_SECRET_ACCESS_KEY=""       # valid AWS secret access key
export AWS_REGION=""                  # such as "us-west-2"
export DNSZONE=""                     # such as "mycompany.com"
export FQDN=""                        # such as "kubeflow.mycompany.com"
export EMAIL_ADDR=""                  # valid email address

```

To avoid potential issues with rate limiting on issuing Let's Encrypt production TLS certificates, when initially creating the cert-manager Issuer, ensure the `server: https://acme-staging-v02` line is uncommented and the `server: https://acme-v02` line is commented out. After verifying that the certificate can be issued correctly, we will reverse this to obtain the valid production certificate.

- Create the cert-manager issuer file.



## Note

You may encounter issues when setting up cert-manager for the first time. To avoid potential rate limiting, you can use the Let's Encrypt staging server for testing and switch to the Let's Encrypt production server once you are successful.

```

cat <<EOF> letsencrypt-issuer.yaml
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: letsencrypt-issuer
  namespace: istio-system
spec:
  acme:
    email: ${EMAIL_ADDR}
    server: https://acme-staging-v02.api.letsencrypt.org/directory # Use this line
    to avoid Let's Encrypt production rate limits
    # server: https://acme-v02.api.letsencrypt.org/directory # Use this line after
    the certificate issues correctly
    privateKeySecretRef:
      name: letsencrypt-issuer-priv-key # K8s secret that will contain the private
    key for this, specific issuer
    solvers:
      - selector:
          dnsZones:
            - "${DNSZONE}"
      dns01:
        route53:

```

```

    region: ${AWS_REGION}
    accessKeyID: ${AWS_ACCESS_KEY_ID}
    secretAccessKeySecretRef:
      name: route53-credentials-secret
      key: secret-access-key
EOF

```



## Important

Review the [letsencrypt-issuer.yaml](#) file for accuracy before continuing.

1. Create the [letsencrypt-issuer](#) resource.

```
kubectl apply -f letsencrypt-issuer.yaml
```

2. Create the Kubernetes secret containing the `aws_secret_access_key` for the AWS user.

```
kubectl create -n istio-system secret generic route53-credentials-secret --
from-literal=secret-access-key=${AWS_SECRET_ACCESS_KEY}
```

3. Verify the contents of the secret.

```
kubectl get -n istio-system secret route53-credentials-secret -o
jsonpath={.data.secret-access-key} | base64 -d; echo ""
```

4. Verify that the host name for the certificate resolves correctly.

```
getent hosts ${FQDN}
```

5. Create the cert-manager certificate resource file.

```

cat <<EOF> kubeflow-certificate.yaml
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: kubeflow-certificate
  namespace: istio-system
spec:
  secretName: kubeflow-certificate-secret # Kubernetes secret that will contain
  the tls.key and tls.crt of the new certificate
  commonName: ${FQDN}
  dnsNames:
    - ${FQDN}
  issuerRef:

```

```
name: letsencrypt-issuer
kind: Issuer
EOF
```



## Important

Verify the contents of the certificate resource file before proceeding.

6. Create the certificate resource.

```
kubectl apply -f kubeflow-certificate.yaml
```

7. Check the status of the certificate.

```
kubectl get -w -n istio-system certificate
```



## Tip

The `-w` flag causes `kubectl` to watch for and display changes. Exit this loop with `Ctrl+C`.

8. If needed, check the progress of the certificate.



## Important

The certificate commonly takes 100 seconds to be issued but can take up to three minutes. Do not proceed to the next step until the cert-manager READY status is True. This indicates that the certificate has been issued.

```
kubectl describe -n istio-system certificate kubeflow-certificate
```

If the certificate seems to be taking a long time to be issued, review the cert-manager logs for clues. Common errors are related to DNS resolution, credentials, and IAM policies. Keep checking back for the status of the certificate, since it will likely keep working in the background.

You can review the cert-manager logs with:

```
kubectl logs -n cert-manager -l app=cert-manager
```

9. Use a browser to connect to the Kubeflow UI. Because you are using the certificate issued by the Let's Encrypt staging server, your browser will not trust it.
10. Use your browser's UI to view the connection certificate and verify that it was issued by the "Let's Encrypt (Staging)" server.

## 5.4 Switch to a Let's Encrypt production certificate

After you have successfully configured cert-manager with the Let's Encrypt staging certificate, you can proceed to switch to the production certificate.

1. Update the `letsencrypt-issuer.yaml` file.

Comment out the `server: https://acme-staging-v02.api.letsencrypt.org/directory` line and uncomment the `server: https://acme-v02.api.letsencrypt.org/directory` line. That is, the file should look like:

```
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: letsencrypt-issuer
  namespace: istio-system
spec:
  acme:
    email: ${EMAIL_ADDR}
    # server: https://acme-staging-v02.api.letsencrypt.org/directory # Use this line
    # to avoid Let's Encrypt production rate limits
    server: https://acme-v02.api.letsencrypt.org/directory # Use this line after the
    certificate issues correctly
    privateKeySecretRef:
      name: letsencrypt-issuer-priv-key # K8s secret that will contain the private
    key for this, specific issuer
    solvers:
      - selector:
          dnsZones:
            - "${DNSZONE}"
        dns01:
          route53:
            region: ${AWS_REGION}
            accessKeyID: ${AWS_ACCESS_KEY_ID}
            secretAccessKeySecretRef:
              name: route53-credentials-secret
              key: secret-access-key
```

2. Update the `letsencrypt-issuer` resource.

```
kubectl apply -f letsencrypt-issuer.yaml
```

3. Remove the staging certificate and its associated secret.

```
kubectl -n istio-system delete secret kubeflow-certificate-secret  
kubectl -n istio-system delete certificate kubeflow-certificate
```

4. Recreate the certificate resource.

You do not need to change the `kubeflow-certificate.yaml` file.

```
kubectl apply -f kubeflow-certificate.yaml
```

5. Check the status of the certificate.

```
kubectl get -w -n istio-system certificate
```



## Note

The `READY` status will become `True` when the certificate has been issued. Recall that this can take up to three minutes.

6. Refresh the istio-gateway deployment to use the new certificate.

```
kubectl rollout restart deployment -n istio-system istio-ingressgateway
```

## 5.5 Validate secure access

With a valid TLS certificate deployed and enabled, validate that you have secure access.

1. Open a browser to the Kubeflow UI using the HTTPS URL.





## Note

You may need to clear your browser's cache.

2. Use your browser interface to verify that you are using the production, signed certificate.
3. Log in to the Kubeflow UI with the credentials:
  - Email address: user@example.com
  - Password: 12341234

## 5.6 Automatically redirect to secure access

Optionally, you can set the gateway to automatically redirect from HTTP to HTTPS.

1. Edit the kubeflow-gateway resource.

```
kubectrl edit -n kubeflow gateways.networking.istio.io kubeflow-gateway
```

2. Change httpsRedirect: false to httpsRedirect: true
3. Confirm the change by attempting to access the Kubeflow UI by the HTTP URL and seeing that you are redirected to the HTTPS URL.

## 6 Summary

Kubeflow is an important tool for managing your ML workloads in Kubernetes. Combined with Rancher Prime by SUSE, you can leverage an enterprise solution for deploying and managing your entire Kubernetes landscape, securely and at scale.

In this guide, you used Harvester by SUSE to provision nodes to host an RKE2 Kubernetes cluster. Then you installed, configured, and secured Kubeflow so it could be managed by Rancher Prime by SUSE.

Continue your learning journey with these resources:


- [Kubeflow Examples \(https://www.kubeflow.org/docs/started/kubeflow-examples/\)](https://www.kubeflow.org/docs/started/kubeflow-examples/) ↗
- [SUSE Technical Reference Documentation \(https://documentation.suse.com/trd/\)](https://documentation.suse.com/trd/) ↗



## 7 Legal notice

Copyright © 2006–2025 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled "GNU Free Documentation License".

SUSE, the SUSE logo and YaST are registered trademarks of SUSE LLC in the United States and other countries. For SUSE trademarks, see <https://www.suse.com/company/legal/> .

Linux is a registered trademark of Linus Torvalds. All other names or trademarks mentioned in this document may be trademarks or registered trademarks of their respective owners.

Documents published as part of the series SUSE Technical Reference Documentation have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

## 8 GNU Free Documentation License

Copyright © 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

### 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition. The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.

- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.



If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all

Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## ADDENDUM: How to use this License for your documents

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2

```
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “ with... Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.