

Overview of the SUSE AI deployment

WHAT?

Basic information about SUSE AI deployment workflow.

WHY?

To better understand the SUSE AI deployment process.

EFFORT

Less than 15 minutes of reading and a basic knowledge of Linux deployment.

Publication Date: 12 Nov 2024

Contents

- 1 Deployment overview 2
- 2 Assigning GPU nodes to applications 4
- 3 Installing Ollama 7
- 4 Installing Open WebUI 19
- 5 Legal Notice 28
- Glossary 28
- A GNU Free Documentation License 29

1 Deployment overview

SUSE AI is a complex product consisting of multiple software layers and components. This topic outlines the complete workflow of deploying and installing all SUSE AI's dependencies as well as installing SUSE AI itself. You can also find references to recommended hardware and software requirements, as well as steps to take after the product installation.



Tip: Hardware and software requirements

For hardware, software and application-specific requirements, refer to [SUSE AI requirements \(https://docserv.suse.de:8085/suse-ai/1.0/html/AI-requirements/index.html\)](https://docserv.suse.de:8085/suse-ai/1.0/html/AI-requirements/index.html).

1.1 Prerequisites for customers who are not already running a Rancher cluster

1. Purchase the Rancher Prime entitlement.
2. Install Rancher Manager (<https://ranchermanager.docs.rancher.com/getting-started/installation-and-upgrade/install-upgrade-on-a-kubernetes-cluster>).
3. Deploy and configure SUSE Security (<https://docserv.suse.de:8085/external-tree/en-us/cloudnative/rancher-manager/v2.8/en/integrations/neuvector/overview.html>).
4. Deploy and configure SUSE Observability (<https://docs.stackstate.com/6.0/get-started/k8s-suse-rancher-prime>).

1.2 Cluster preparation

1. Install and register SUSE Linux Micro 6.0 or later on each RKE2 cluster node. Refer to <https://docserv.suse.de:8085/sle-micro/6.0/> for details.
2. Install the NVIDIA GPU driver on cluster nodes with GPUs. Refer to <https://docserv.suse.de:8085/suse-ai/1.0/html/NVIDIA-GPU-driver-on-SL-Micro/index.html> for details.
3. Install RKE2 Kubernetes distribution on the cluster nodes. Refer to <https://docs.rke2.io/> for details.

4. Install the NVIDIA GPU Operator with the additional option `--set driver.enabled=false`. Refer to <https://docs.nvidia.com/datacenter/cloud-native/gpu-operator/latest/getting-started.html#rancher-kubernetes-engine-2>.
5. Connect the RKE2 cluster to Rancher Manager. Refer to <https://ranchermanager.docs.rancher.com/how-to-guides/new-user-guides/kubernetes-clusters-in-rancher-setup/register-existing-clusters> for details.
6. Configure the GPU enabled nodes so that the SUSE AI containers are assigned to Pods that run on nodes equipped with NVIDIA GPU hardware. Find more details assigning Pods to nodes in *Section 2, "Assigning GPU nodes to applications"*.
7. Configure SUSE Security to scan the nodes used for SUSE AI. Although this step is not required, we strongly encourage it to ensure the security in production environment.
8. Configure SUSE Observability to observe the nodes used for SUSE AI application.

1.3 SUSE AI installation

SUSE AI is being delivered as a set of components that you can combine to meet specific use cases. This provides extraordinary flexibility but means that there is not a single Helm chart that installs the whole stack, for example, for using the Open WebUI chatbot style application. To enable the full integrated stack, you need to deploy multiple applications in sequence. The applications with the fewest dependencies must be installed first, while depending applications after their dependencies have been installed into the cluster.

1. Purchase the SUSE AI entitlement. It is a separate entitlement from Rancher Prime.
2. Access SUSE AI via the Rancher Application Collection at <https://apps.rancher.io/> to perform the check for the SUSE AI entitlement.
3. If the entitlement check is successful, you are given access to the SUSE AI-related Helm charts and container images, and can deploy directly from the Rancher Application Collection.



Tip

Any overrides to the default values in the Helm charts—such as Open WebUI password and URL customizations—occur at this step.

4. (Optional) Install Ollama as described in [Section 3, “Installing Ollama”](#).
5. Install Open WebUI as described in [Section 4, “Installing Open WebUI”](#).

1.4 Steps after the installation is complete

1. Log in to SUSE AI Open WebUI using the default credentials.
2. After you have logged in, update the administrator password for SUSE AI.
3. From the available language models, configure the one you prefer. Optionally, install a custom language model.
4. Configure user management RBAC and SSO (<https://docs.openwebui.com/tutorials/features/sso>)
5. Configure RAG (Retrieval Augmented Generation) to include content relevant to the customer use case in results (<https://docs.openwebui.com/tutorials/features/rag>)

2 Assigning GPU nodes to applications

When deploying a containerized application to Kubernetes, you need to ensure that containers that require GPU resources are run on appropriate worker nodes. For example, Ollama, a core component of SUSE AI, can deeply benefit from the use of GPU acceleration. This topic describes how to satisfy this requirement by explicitly requesting GPU resources and labeling worker nodes for configuring the node selector.

REQUIREMENTS

- Kubernetes cluster—such as RKE2—must be available and configured with more than one worker node in which certain nodes have NVIDIA GPU resources and others do not.
- This document assumes that any kind of deployment to the Kubernetes cluster is done using Helm charts.

2.1 Labeling GPU nodes

To distinguish nodes with the GPU support from non-GPU nodes, Kubernetes uses *labels*. Labels are used for relevant metadata and should not be confused with annotations that provide simple information about a resource. It is possible to manipulate labels with the **kubectl** command, as well as by tweaking configuration files from the nodes. If an IaC tool such as Terraform is used, labels can be inserted in the node resource configuration files.

To label a single node, use the following command:

```
> kubectl label node GPU_NODE_NAME accelerator=nvidia-gpu
```

To achieve the same result by tweaking the `node.yaml` node configuration, add the following content and apply the changes with **kubectl apply -f node.yaml**:

```
apiVersion: v1
kind: Node
metadata:
  name: node-name
  labels:
    accelerator: nvidia-gpu
```



Tip: Labeling multiple nodes

To label multiple nodes, use the following command:

```
> kubectl label node \
  GPU_NODE_NAME1 \
  GPU_NODE_NAME2 ... \
  accelerator=nvidia-gpu
```



Tip

If Terraform is being used as an IaC tool, you can add labels to a group of nodes by editing the `.tf` files and adding the following values to a resource:

```
resource "node_group" "example" {
  labels = {
    "accelerator" = "nvidia-gpu"
  }
}
```

To check if the labels are correctly applied, use the following command:

```
> kubectl get nodes --show-labels
```

2.2 Assigning GPU nodes

The matching between a container and a node is configured by the explicit resource allocation and the use of labels and node selectors. The use cases described below focus on NVIDIA GPUs.

2.2.1 Enable GPU passthrough

Containers are isolated from the host environment by default. For the containers that rely on the allocation of GPU resources, their Helm charts must enable GPU passthrough so that the container can access and use the GPU resource. Without enabling the GPU passthrough, the container may still run, but it can only use the main CPU for all computations. Refer to [Ollama Helm chart \(https://docserv.suse.de:8085/AI-deployment-intro/html/AI-deployment-intro/\)](https://docserv.suse.de:8085/AI-deployment-intro/html/AI-deployment-intro/) [↗](#) for an example of the configuration required for GPU acceleration.

2.2.2 Assignment by resource request

After the NVIDIA GPU Operator is configured on a node, you can instantiate applications requesting the resource `nvidia.com/gpu` provided by the operator. Add the following content to your `values.yaml` file. Specify the number of GPUs according to your setup.

```
resources:
  requests:
    nvidia.com/gpu: 1
  limits:
    nvidia.com/gpu: 1
```

2.2.3 Assignment by labels and node selectors

If affected cluster nodes are labeled with a label such as `accelerator=nvidia-gpu`, you can configure the node selector to check for the label. In this case, use the following values in your `values.yaml` file.

```
nodeSelector:  
  accelerator: nvidia-gpu
```

2.3 Verify Ollama GPU assignment

If GPU is correctly detected, the Ollama container logs such event:

```
| [...] source=routes.go:1172 msg="Listening on :11434 (version 0.0.0)"  
|  
| [...] source=payload.go:30 msg="extracting embedded files" dir=/tmp/ollama2502346830/  
runners  
|  
| [...] source=payload.go:44 msg="Dynamic LLM libraries [cuda_v12 cpu cpu_avx cpu_avx2]"  
|  
| [...] source=gpu.go:204 msg="looking for compatible GPUs"  
|  
| [...] source=types.go:105 msg="inference compute" id=GPU-c9ad37d0-d304-5d2a-c2e6-  
d3788cd733a7 library=cuda compute |
```

3 Installing Ollama

Ollama is a tool for running and managing language models locally on your computer. It offers a simple interface to download, run and interact with models without relying on cloud resources.



Tip

When installing SUSE AI, Ollama is installed by the Open WebUI installation by default. If you decide to install Ollama separately, disable its installation during the installation of Open WebUI.

The following procedure describes how to install Ollama as a separate application.

1. Visit <https://apps.rancher.io/applications/ollama> with your Web browser.
2. Run the indicated `helm pull` command.

3. Tip

To override the default installation values, create a custom `values.yaml` file and specify it during the following `helm install` command with the `-f values.yaml` option. For a list of all installation options with examples, refer to [Section 3.1, “Values for the Ollama Helm chart”](#).

Install the Ollama chart. Assume that the release name is `ollama`, for example.

```
> helm install ollama \
  --set 'global.imagePullSecrets[0].name'=my-pull-secrets \
  oci://dp.apps.rancher.io/charts/ollama
```

3.1 Values for the Ollama Helm chart

To override the default values during Ollama installation using the Helm chart, you can create a `values.yaml` file and specify your custom values. Then you can apply the values by specifying the path to the `values.yaml` file during the `helm install` command.

EXAMPLE 1: BASIC `values.yaml` WITH GPU AND TWO MODELS PULLED AT STARTUP

```
ollama:
  gpu:
    # -- Enable GPU integration
    enabled: true

    # -- GPU type: 'nvidia' or 'amd'
    type: 'nvidia'

    # -- Specify the number of GPU to 1
    number: 1

  # -- List of models to pull at container startup
  models:
    - mistral
    - llama2
```

EXAMPLE 2: BASIC `values.yaml` WITH INGRESS

```
ollama:
  models:
```



```

- llama2

ingress:
  enabled: true
  hosts:
    - host: ollama.domain.lan
    paths:
      - path: /
        pathType: Prefix

```

Ollama's API is reachable at `ollama.domain.lan` in this example.

TABLE 1: `values.yaml` OPTIONS FOR THE OLLAMA HELM CHART

Key	Type	Default	Description
<code>affinity</code>	object	<code>{}</code>	Affinity for pod assignment
<code>autoscaling.enabled</code>	bool	<code>false</code>	Enable autoscaling
<code>autoscaling.maxReplicas</code>	int	<code>100</code>	Number of maximum replicas
<code>autoscaling.minReplicas</code>	int	<code>1</code>	Number of minimum replicas
<code>autoscaling.targetCPUUtilizationPercentage</code>	int	<code>80</code>	CPU usage to target replica
<code>extraArgs</code>	list	<code>[]</code>	Additional arguments on the output Deployment definition.
<code>extraEnv</code>	list	<code>[]</code>	Additional environments variables on the output Deployment definition.
<code>fullnameOverride</code>	string	<code>""</code>	String to fully override template

*

Key	Type	Default	Description
global.imagePullSecrets	list	[]	Global override for container image registry pull secrets
global.imageRegistry	string	""	Global override for container image registry
hostIPC	bool	false	Use the host's IPC namespace.
hostNetwork	bool	false	Use the host's network namespace.
hostPID	bool	false	Use the host's PID namespace.
image.pullPolicy	string	"IfNotPresent"	Image pull policy to use for the Ollama container
image.registry	string	"dp.apps.rancher.io"	Image registry to use for the Ollama container
image.repository	string	"containers/ollama"	Image repository to use for the Ollama container
image.tag	string	"0.3.6"	Image tag to use for the Ollama container
imagePullSecrets	list	[]	Docker registry secret names as an array

*

Key	Type	Default	Description
ingress.annotations	object	{}	Additional annotations for the Ingress resource.
ingress.className	string	""	IngressClass that is used to implement the Ingress (Kubernetes 1.18+)
ingress.enabled	bool	false	Enable Ingress controller resource
ingress.hosts[0].host	string	"ollama.local"	
ingress.hosts[0].paths[0].path	string	"/"	
ingress.hosts[0].paths[0].pathType	string	"Prefix"	
ingress.tls	list	[]	The TLS configuration for host names to be covered with this Ingress record.
initContainers	list	[]	Init containers to add to the pod
knative.container-Concurrency	int	0	Knative service container concurrency
knative.enabled	bool	false	Enable Knative integration
knative.idleTimeoutSeconds	int	300	Knative service idle timeout seconds

*

Key	Type	Default	Description
knative.responseStartTimeoutSeconds	int	300	Knative service response start timeout seconds
knative.timeoutSeconds	int	300	Knative service timeout seconds
livenessProbe.enabled	bool	true	Enable livenessProbe
livenessProbe.failureThreshold	int	6	Failure threshold for livenessProbe
livenessProbe.initialDelaySeconds	int	60	Initial delay seconds for livenessProbe
livenessProbe.path	string	"/"	Request path for livenessProbe
livenessProbe.periodSeconds	int	10	Period seconds for livenessProbe
livenessProbe.successThreshold	int	1	Success threshold for livenessProbe
livenessProbe.timeoutSeconds	int	5	Timeout seconds for livenessProbe
nameOverride	string	""	String to partially override template (maintains the release name)
nodeSelector	object	{}	Node labels for pod assignment.

Key	Type	Default	Description
ollama.gpu.enabled	bool	false	Enable GPU integration
ollama.gpu.number	int	1	Specify the number of GPUs
ollama.gpu.nvidiaResource	string	"nvidia.com/gpu"	Only for NVIDIA cards; change to <u>nvidia.com/mig-1g.10gb</u> to use MIG slice
ollama.gpu.type	string	"nvidia"	GPU type: "nvidia" or "amd". If "ollama.gpu.enabled" is enabled, the default value is "nvidia". If set to "amd", this adds the "rocm" suffix to the image tag if "image.tag" is not override. This is because AMD and CPU/CUDA are different images
ollama.insecure	bool	false	Add insecure flag for pulling at container startup
ollama.models	list	[]	List of models to pull at container startup The more you add, the longer the container takes to start

*

Key	Type	Default	Description
			if models are not present models: - llama2 - mistral
ollama.mountPath	string	""	Override ollama-data volume mount path, default: "/root/.ollama"
persistentVolume.accessModes	list	["ReadWriteOnce"]	Ollama server data Persistent Volume access modes. Must match those of existing PV or dynamic provisioner, see http://kubernetes.io/docs/user-guide/persistent-volumes/ ↗
persistentVolume.annotations	object	{}	Ollama server data Persistent Volume annotations
persistentVolume.enabled	bool	false	Enable persistence using PVC
persistentVolume.existingClaim	string	""	If you want to bring your own PVC for persisting Ollama state, pass the name of the created + ready PVC here. If set, this Chart does not create the default

*

Key	Type	Default	Description
			PVC. Requires <code>server.persistentVolume.enabled: true</code>
<code>persistentVolume.size</code>	string	"30Gi"	Ollama server data Persistent Volume size
<code>persistentVolume.storageClass</code>	string	""	Ollama server data Persistent Volume Storage Class. If defined, storageClassName: if set to "-", storageClassName: "", which disables dynamic provisioning If undefined (the default) or set to null, no storageClassName spec is set, choosing the default provisioner. (gp2 on AWS, standard on GKE, AWS & OpenStack)
<code>persistentVolume.subPath</code>	string	""	Subdirectory of Ollama server data Persistent Volume to mount. Useful if the volume's root directory is not empty.
<code>persistentVolume.volumeMode</code>	string	""	Ollama server data Persistent Volume Binding Mode.

*

Key	Type	Default	Description
			If empty (the default) or set to null, no volumeBindingMode specification is set, choosing the default mode.
persistentVolume.volumeName	string	""	Ollama server Persistent Volume name. It can be used to force-attach the created PVC to a specific PV.
podAnnotations	object	{}	Map of annotations to add to the pods.
podLabels	object	{}	Map of labels to add to the pods.
podSecurityContext	object	{}	Pod Security Context
readinessProbe.enabled	bool	true	Enable readinessProbe
readinessProbe.failureThreshold	int	6	Failure threshold for readinessProbe.
readinessProbe.initialDelaySeconds	int	30	Initial delay seconds for readinessProbe.
readinessProbe.path	string	"/"	Request path for readinessProbe.
readinessProbe.periodSeconds	int	5	Period seconds for readinessProbe.

Key	Type	Default	Description
readinessProbe.successThreshold	int	1	Success threshold for readinessProbe.
readinessProbe.timeoutSeconds	int	3	Timeout seconds for readinessProbe.
replicaCount	int	1	Number of replicas.
resources.limits	object	{}	Pod limit
resources.requests	object	{}	Pod requests.
runtimeClassName	string	""	Specify runtime class.
securityContext	object	{}	Container Security Context.
service.annotations	object	{}	Annotations to add to the service.
service.nodePort	int	31434	Service node port when service type is "NodePort".
service.port	int	11434	Service port.
service.type	string	"ClusterIP"	Service type.
serviceAccount.annotations	object	{}	Annotations to add to the service account.
serviceAccount.autoMount	bool	true	Whether automatically mount a ServiceAccount's API credentials.

Key	Type	Default	Description
serviceAccount.create	bool	true	Whether a service account should be created.
serviceAccount.name	string	""	The name of the service account to use. If not set and create is "true", a name is generated using the full name template.
tolerations	list	[]	Tolerations for pod assignment.
topologySpreadConstraints	object	{}	Topology Spread Constraints for pod assignment.
updateStrategy	object	{"type":""}	How to replace existing pods.
updateStrategy.type	string	""	Can be "Recreate" or "RollingUpdate". Default is "RollingUpdate".
volumeMounts	list	[]	Additional volumeMounts on the output Deployment definition.
volumes	list	[]	Additional volumes on the output Deployment definition.

*

4 Installing Open WebUI

Open WebUI is a Web-based user interface designed for interacting with AI models.

TABLE 2: OPEN WEBUI INSTALLATION REQUIREMENTS

Repository	Name	Version	Note
oci://dp.apps.rancher.io/charts ↗	cert-manager	> = 1.16.1	By default, Open WebUI is deployed with TLS enabled. Default TLS source is “self-signed”.
oci://dp.apps.rancher.io/charts ↗	ollama	> = 0.54.0	By default, Ollama is enabled.
https://helm.openwebui.com/ ↗	pipelines	> = 0.0.1	Pipelines are disabled by default and SUSE does not support enabling pipelines for this release. Please refer to the upstream pipelines chart https://github.com/open-webui/helm-charts/tree/main/charts/pipelines ↗ for deployment of pipelines outside of the scope of SUSE support.

1. Visit <https://apps.rancher.io/applications/open-webui> ↗ with your Web browser.
2. Run the indicated `helm pull` command.

3. Tip

To override the default installation values, create a custom `values.yaml` file and specify it during the following `helm install` command with the `-f values.yaml` option. For a list of all installation options with examples, refer to [Section 4.1, “Values for the Open WebUI Helm chart”](#).

Install the Ollama chart. Assume that the release name is `open-webui`.

The following command uses the default Open WebUI vector DB:

```
> helm install open-webui \
  --set 'global.imagePullSecrets[0].name'=my-pull-secrets \
  --set 'persistence.storageClass'=my-storage-class \
  --set 'ingress.host'=my-host \
  oci://dp.apps.rancher.io/charts/open-webui
```

To use Milvus as the vector DB, the installation command has the following syntax:

```
> helm install open-webui \
  --set 'global.imagePullSecrets[0].name'=my-pull-secrets \
  --set 'persistence.storageClass'=my-storage-class \
  --set 'ingress.host'=my-host \
  --set 'extraEnvVars[0].name=VECTOR_DB' --set 'extraEnvVars[0].value=milvus' \
  --set 'extraEnvVars[1].name=MILVUS_URI' \
  --set-string 'extraEnvVars[1].value=http://my-milvusuri' \
  oci://dp.apps.rancher.io/charts/open-webui
```

4.1 Values for the Open WebUI Helm chart

To override the default values during Open WebUI installation using the Helm chart, you can create a `values.yaml` file and specify your custom values. Then you can apply the values by specifying the path to the `values.yaml` file during the `helm install` command.


TABLE 3: `values.yaml` OPTIONS FOR THE OPEN WEBUI HELM CHART

Key	Type	Default	Description
affinity	object	{}	Affinity for pod assignment.

Key	Type	Default	Description
annotations	object	{}	
cert-manager.enabled	bool	true	
clusterDomain	string	"cluster.local"	Value of cluster domain.
containerSecurityContext	object	{}	Configure container security context, see https://kubernetes.io/docs/tasks/configure-pod-container/security-context/#set-the-security-context-for-a-container
extraEnvVars	list	[{"name":"OPENAI_API_KEY", "value":"0p3n-w3bu!"}]	Environment variables added to the Open WebUI deployment. Most up-to-date environment variables can be found in https://docs.openwebui.com/getting-started/env-configuration/ .
extraEnvVars[0]	object	{"name":"OPENAI_API_KEY", "value":"0p3n-w3bu!"}	Default API key value for Pipelines. It should be updated in a production deployment, changed to the required API key if not using Pipelines.

*

Key	Type	Default	Description
global.imagePullSecrets	list	[]	Global override for container image registry pull secrets.
global.imageRegistry	string	""	Global override for container image registry.
global.tls.additionalTrustedCAs	bool	false	
global.tls.issuerName	string	"suse-private-ai"	
global.tls.letsEncrypt.email	string	"none@example.com"	
global.tls.letsEncrypt.environment	string	"staging"	
global.tls.letsEncrypt.ingress.class	string	""	
global.tls.source	string	"suse-private-ai"	The source of Open WebUI TLS keys, see Section 4.1.1, "TLS sources" .
image.pullPolicy	string	"IfNotPresent"	Image pull policy to use for the Open WebUI container.
image.registry	string	"dp.apps.rancher.io"	Image registry to use for the Open WebUI container.

Key	Type	Default	Description
image.repository	string	"containers/open-we-bui"	Image repository to use for the Open WebUI container.
image.tag	string	"0.3.32"	Image tag to use for the Open WebUI container.
imagePullSecrets	list	[]	Configure imagePullSecrets to use private registry, see https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry 
ingress.annotations	object	{"nginx.ingress.kubernetes.io/ssl-redirect":"true"}	Use appropriate annotations for your Ingress controller, such as nginx.ingress.kubernetes.io/rewrite-target: / for NGINX.
ingress.class	string	""	
ingress.enabled	bool	true	
ingress.existingSecret	string	""	
ingress.host	string	""	
ingress.tls	bool	true	

*

Key	Type	Default	Description
nameOverride	string	""	
nodeSelector	object	{}	Node labels for pod assignment.
ollama.enabled	bool	true	Automatically install Ollama Helm chart from https://otwld.github.io/ollama-helm/ . Configure the following Helm values (https://github.com/otwld/ollama-helm/#helm-values).
ollama.fullnameOverride	string	"open-webui-ollama"	If enabling embedded Ollama, update fullnameOverride to your desired Ollama name value, or else it will use the default ollama.name value from the Ollama chart.
ollamaUrls	list	[]	A list of Ollama API endpoints. These can be added in lieu of automatically installing the Ollama Helm chart, or in addition to it.

*

Key	Type	Default	Description
openaiBaseApiUrl	string	""	OpenAI base API URL to use. Defaults to the Pipelines service endpoint when Pipelines are enabled, or to https://api.openai.com/v1 if Pipelines are not enabled and this value is blank.
persistence.accessModes	list	["ReadWriteOnce"]	If using multiple replicas, you must update accessModes to ReadWriteMany.
persistence.annotations	object	{}	
persistence.enabled	bool	true	
persistence.existingClaim	string	""	Use existingClaim to re-use an existing Open WebUI PVC instead of creating a new one.
persistence.selector	object	{}	
persistence.size	string	"2Gi"	
persistence.storageClass	string	""	

*

Key	Type	Default	Description
pipelines.enabled	bool	false	Automatically install Pipelines chart to extend Open WebUI functionality using Pipelines, see https://github.com/open-webui/pipelines .
pipelines.extraEnvVars	list	[]	This section can be used to pass required environment variables to your pipelines (such as the Langfuse host name).
podAnnotations	object	{}	
podSecurityContext	object	{}	Configure pod security context, see https://kubernetes.io/docs/tasks/configure-pod-container/security-context/#set-the-security-context-for-a-container .
replicaCount	int	1	
resources	object	{}	
service	object	{"annotations": {}, "containerPort": 8080, "labels": {}, "loadBal-	Service values to expose Open WebUI pods to cluster

*

Key	Type	Default	Description
		ancerClass":""," "node-Port":"","port":80,"type":"ClusterIP"}	
tolerations	list	[]	Tolerations for pod assignment.
topologySpreadConstraints	list	[]	Topology Spread Constraints for pod assignment.

4.1.1 TLS sources

There are three recommended options where Open WebUI can obtain TLS certificates for secure communication.

Self-Signed TLS certificate

This is the default method. You need to install `cert-manager` on the cluster to issue and maintain the certificates. This method generates a CA and signs the Open WebUI certificate using the CA. `cert-manager` then manages the signed certificate.

For this method, use the following Helm chart option:

```
global.tls.source=suse-private-ai
```

Let's Encrypt

This method also uses `cert-manager` but it is combined with a special issuer for Let's Encrypt that performs all actions—including request and validation—to get the Let's Encrypt certificate issued. This configuration uses HTTP validation (HTTP-01) and therefore the load balancer must have a public DNS record and be accessible from the internet.

For this method, use the following Helm chart option:

```
global.tls.source=letsEncrypt
```

Provide your own certificate

This method allows you to bring your own signed certificate to secure the HTTPS traffic. In this case, you must upload this certificate and associated key as PEM-encoded files named `tls.crt` and `tls.key`.

For this method, use the following Helm chart option:

```
global.tls.source=secret
```

5 Legal Notice

Copyright© 2006–2024 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”.

For SUSE trademarks, see <https://www.suse.com/company/legal/>. All other third-party trademarks are the property of their respective owners. Trademark symbols (®, ™ etc.) denote trademarks of SUSE and its affiliates. Asterisks (*) denote third-party trademarks.

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither SUSE LLC, its affiliates, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

Glossary

AI

Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are designed to learn and solve problems like humans. AI enables computers to understand language, making decisions, and improving from experience.

GenAI

Generative AI (GenAI) is a type of artificial intelligence that can create new content such as text, images or music.

NLG

Natural Language Generation (NLG) is a process of automatically generating human-like text from structured data or other forms of input. NLG systems are designed to convert raw data into coherent and meaningful language easily understood by humans.

NLU

Natural Language Understanding (NLU) is a process the AI uses to analyze and understand the meaning of the input query.

A GNU Free Documentation License

Copyright (C) 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 non-commercially. 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 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 non-commercially, 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 <https://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.