Jenkins Helm Chart Deployment

Overview

This document provides detailed step-by-step processes and deliverables for deploying core DevOps tools (Argo CD, Jenkins, Nexus, SonarQube) using Helm charts in a Kubernetes environment.

Environment Setup on Debian WSL

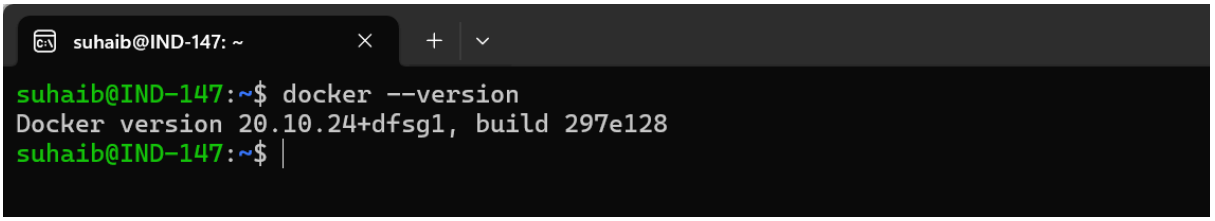
Step 1: Update System and Install Prerequisites

```
# Update system packages
sudo apt update && sudo apt upgrade -y

# Install Docker
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dea
rmor -o /usr/share/keyrings/docker-archive-keyring.gpg
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-ke
yring.gpg] https://download.docker.com/linux/debian $(lsb_release -cs) sta
ble" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io

# Add user to docker group
sudo usermod -aG docker $USER
newgrp docker

# Verify Docker installation
docker --version
```



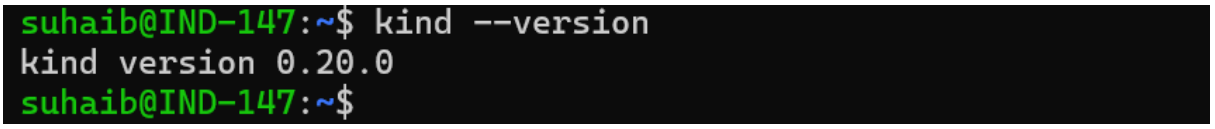
Step 2: Install Kubernetes (Kind for local development)

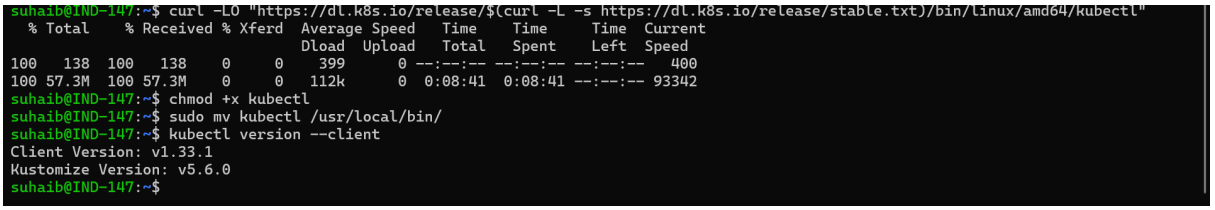
```
# Install Kind (Kubernetes in Docker)
curl -Lo ./kind https://github.com/kubernetes-sigs/kind/releases/downloa
d/v0.20.0/kind-linux-amd64
chmod +x ./kind
sudo mv ./kind /usr/local/bin/kind

# Verify Kind installation
kind --version

# Install kubectl
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stabl
e.txt)/bin/linux/amd64/kubectl"
chmod +x kubectl
sudo mv kubectl /usr/local/bin/

# Verify kubectl installation
kubectl version --client
```





Step 3: Install Helm

```
# Install Helm
curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee /us
```

```
r/share/keyrings/helm.gpg > /dev/null
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrin
gs/helm.gpg] https://baltocdn.com/helm/stable/debian/ all main" | sudo tee
/etc/apt/sources.list.d/helm-stable-debian.list
sudo apt update
sudo apt install helm

# Verify Helm installation
helm version
```

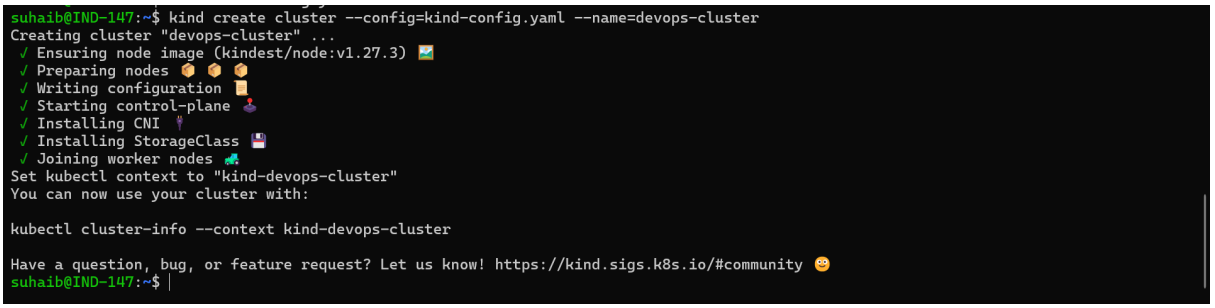
Step 4: Create Kubernetes Cluster

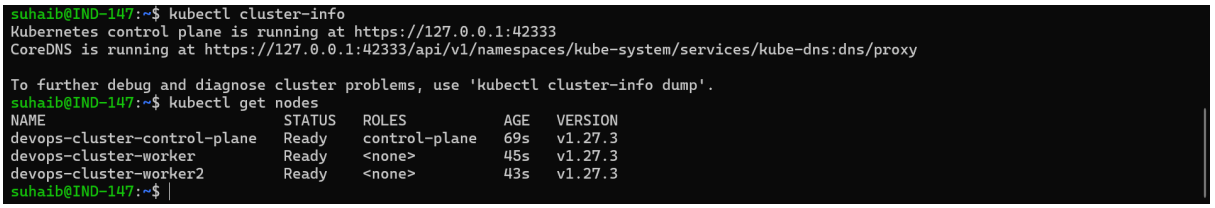
```
# Create Kind cluster configuration
cat << EOF > kind-config.yaml
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
nodes:
- role: control-plane
  kubeadmConfigPatches:
  - |
    kind: InitConfiguration
    nodeRegistration:
      kubeletExtraArgs:
        node-labels: "ingress-ready=true"
  extraPortMappings:
  - containerPort: 80
    hostPort: 8081
    protocol: TCP
  - containerPort: 443
    hostPort: 443
    protocol: TCP
```

```
  - containerPort: 8080
    hostPort: 8082
    protocol: TCP
- role: worker
- role: worker
EOF

# Create the cluster
kind create cluster --config=kind-config.yaml --name=devops-cluster

# Verify cluster
kubectl cluster-info
kubectl get nodes
```





Step 5: Install NGINX Ingress Controller

```
# Install NGINX Ingress Controller
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-ngi
nx/main/deploy/static/provider/kind/deploy.yaml

# Wait for ingress controller to be ready
kubectl wait --namespace ingress-nginx \
  --for=condition=ready pod \
  --selector=app.kubernetes.io/component=controller \
  --timeout=90s
```

```
kubectl get pods -n ingress-nginx -l app.kubernetes.io/component=control
ler
```

Task 4: Develop Helm Charts for Argo CD, Jenkins, Nexus, SonarQube

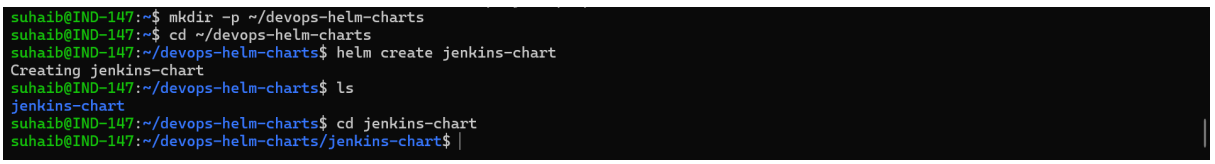
Jenkins:

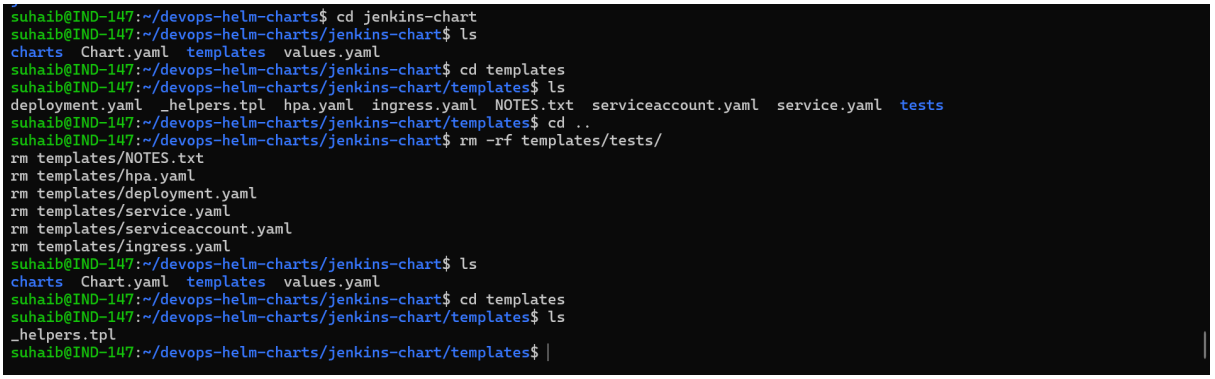
Step 6: Create Jenkins Helm Chart Structure

```
# Create project directory
mkdir -p ~/devops-helm-charts
cd ~/devops-helm-charts

# Create Jenkins Helm chart
helm create jenkins-chart
cd jenkins-chart

# Clean up default files
rm -rf templates/tests/
rm templates/NOTES.txt
rm templates/hpa.yaml
rm templates/deployment.yaml
rm templates/service.yaml
rm templates/serviceaccount.yaml
rm templates/ingress.yaml
```





Step 7: Configure Chart.yaml

```
# Chart.yaml
apiVersion: v2
name: jenkins-chart
description: A Helm chart for Jenkins CI/CD with custom configuration
type: application
version: 0.1.0
appVersion: "2.426.1-lts"
keywords:
  - jenkins
  - ci
  - cd
  - devops
home: https://jenkins.io/
sources:
  - https://github.com/jenkinsci/jenkins
maintainers:
  - name: DevOps Team
    email: devops@cprime.com
```

Step 8: Configure values.yaml

```
# values.yaml
jenkins:
  image:
    repository: jenkins/jenkins
    tag: "2.426.1-lts"
    pullPolicy: IfNotPresent
```

```
# Resource configuration
resources:
  requests:
    memory: "512Mi"
    cpu: "500m"
  limits:
    memory: "2Gi"
    cpu: "2000m"

# Service configuration
service:
  type: ClusterIP
  port: 8080
  targetPort: 8080
  name: jenkins-service

# Persistence configuration
persistence:
  enabled: true
  storageClass: "standard"
  size: "10Gi"
  accessMode: ReadWriteOnce

# Security context
securityContext:
  runAsUser: 1000
  runAsGroup: 1000
  fsGroup: 1000

# Jenkins admin configuration
admin:
  username: "admin"
  password: "admin123"

# Reduced plugin list for faster startup
installPlugins:
  - kubernetes:latest
```

```
    - workflow-aggregator:latest
    - git:latest
    - configuration-as-code:latest
    - blueocean:latest
    - pipeline-stage-view:latest

  # JCasC configuration
  jcasc:
    enabled: true
    configScripts:
      welcome-message: |
        jenkins:
          systemMessage: "Welcome to Jenkins - Deployed via Helm!"
      security-realm: |
        jenkins:
          securityRealm:
            local:
              allowsSignup: false
              users:
                - id: "${JENKINS_ADMIN_USERNAME}"
                  password: "${JENKINS_ADMIN_PASSWORD}"
      authorization-strategy: |
        jenkins:
          authorizationStrategy:
            loggedInUsersCanDoAnything:
              allowAnonymousRead: false

# Ingress configuration - UPDATED PORT
ingress:
  enabled: true
  className: "nginx"
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
    nginx.ingress.kubernetes.io/ssl-redirect: "false"
    nginx.ingress.kubernetes.io/backend-protocol: "HTTP"
  hosts:
    - host: jenkins.local
      paths:
```

```
        - path: /
          pathType: Prefix
  tls:
    - secretName: jenkins-tls
      hosts:
        - jenkins.local

# Service Account
serviceAccount:
  create: true
  name: "jenkins-sa"
  annotations: {}

# RBAC
rbac:
  create: true
  rules:
    - apiGroups: [""]
      resources: ["pods", "pods/exec", "pods/log"]
      verbs: ["create", "delete", "get", "list", "patch", "update", "watch"]
    - apiGroups: [""]
      resources: ["secrets", "configmaps"]
      verbs: ["get", "list", "watch"]

# Monitoring
monitoring:
  enabled: true
  serviceMonitor:
    enabled: false
    interval: "30s"
    path: "/prometheus"

# Node selector and tolerations
nodeSelector: {}
tolerations: []
affinity: {}
```

Step 9: Create Template Files

Create the following template files:

templates/configmap.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ include "jenkins-chart.fullname" . }}-config
  labels:
    {{- include "jenkins-chart.labels" . | nindent 4 }}
data:
  plugins.txt: |
    {{- range .Values.jenkins.installPlugins }}
    {{ . }}
    {{- end }}
  {{- if .Values.jenkins.jcasc.enabled }}
  jenkins.yaml: |
    {{- range $key, $val := .Values.jenkins.jcasc.configScripts }}
    {{ $val | nindent 4 }}
    {{- end }}
  {{- end }}
```

templates/secret.yaml

```
apiVersion: v1
kind: Secret
metadata:
  name: {{ include "jenkins-chart.fullname" . }}-secret
  labels:
    {{- include "jenkins-chart.labels" . | nindent 4 }}
type: Opaque
data:
  jenkins-admin-user: {{ .Values.jenkins.admin.username | b64enc | quote
}}
  jenkins-admin-password: {{ .Values.jenkins.admin.password | b64enc | q
uote }}
  {{- if .Values.database.external.enabled }}
  database-username: {{ .Values.database.external.username | b64enc | qu
```

```
ote }}
  database-password: {{ .Values.database.external.password | b64enc | qu
ote }}
  {{- end }}
```

templates/pvc.yaml

```
{{- if .Values.jenkins.persistence.enabled }}
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: {{ include "jenkins-chart.fullname" . }}-pvc
  labels:
    {{- include "jenkins-chart.labels" . | nindent 4 }}
spec:
  accessModes:
    - {{ .Values.jenkins.persistence.accessMode }}
  resources:
    requests:
      storage: {{ .Values.jenkins.persistence.size }}
  {{- if .Values.jenkins.persistence.storageClass }}
  storageClassName: {{ .Values.jenkins.persistence.storageClass }}
  {{- end }}
{{- end }}
```

templates/serviceaccount.yaml

```
{{- if .Values.serviceAccount.create }}
apiVersion: v1
kind: ServiceAccount
metadata:
  name: {{ include "jenkins-chart.fullname" . }}-sa
  labels:
    {{- include "jenkins-chart.labels" . | nindent 4 }}
  {{- with .Values.serviceAccount.annotations }}
  annotations:
    {{- toYaml . | nindent 4 }}
```

```
  {{- end }}
automountServiceAccountToken: true
{{- end }}
```

templates/rbac.yaml

```
{{- if .Values.rbac.create }}
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: {{ include "jenkins-chart.fullname" . }}-role
  labels:
    {{- include "jenkins-chart.labels" . | nindent 4 }}
rules:
{{- with .Values.rbac.rules }}
  {{- toYaml . | nindent 2 }}
{{- end }}
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: {{ include "jenkins-chart.fullname" . }}-rolebinding
  labels:
    {{- include "jenkins-chart.labels" . | nindent 4 }}
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: {{ include "jenkins-chart.fullname" . }}-role
subjects:
- kind: ServiceAccount
  name: {{ include "jenkins-chart.fullname" . }}-sa
  namespace: {{ .Release.Namespace }}
{{- end }}
```

templates/deployment.yaml

```
# templates/deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ include "jenkins-chart.fullname" . }}
  labels:
    {{- include "jenkins-chart.labels" . | nindent 4 }}
spec:
  replicas: 1
  selector:
    matchLabels:
      {{- include "jenkins-chart.selectorLabels" . | nindent 6 }}
  template:
    metadata:
      labels:
        {{- include "jenkins-chart.selectorLabels" . | nindent 8 }}
    spec:
      serviceAccountName: {{ include "jenkins-chart.fullname" . }}-sa
      securityContext:
        {{- toYaml .Values.jenkins.securityContext | nindent 8 }}
      containers:
      - name: jenkins
        image: "{{ .Values.jenkins.image.repository }}:{{ .Values.jenkins.image.
tag }}"
        imagePullPolicy: {{ .Values.jenkins.image.pullPolicy }}
        ports:
        - name: http
          containerPort: 8080
          protocol: TCP
        - name: jnlp
          containerPort: 50000
          protocol: TCP
        env:
        - name: JENKINS_ADMIN_USERNAME
          valueFrom:
            secretKeyRef:
              name: {{ include "jenkins-chart.fullname" . }}-secret
              key: jenkins-admin-user
```

```
      - name: JENKINS_ADMIN_PASSWORD
        valueFrom:
          secretKeyRef:
            name: {{ include "jenkins-chart.fullname" . }}-secret
            key: jenkins-admin-password
      - name: CASC_JENKINS_CONFIG
        value: "/var/jenkins_home/casc_configs/jenkins.yaml"
      - name: JAVA_OPTS
        value: >
          -Djenkins.install.runSetupWizard=false
          -Djava.awt.headless=true
          -Dhudson.security.csrf.DefaultCrumbIssuer.EXCLUDE_SESSION_ID
=true
      volumeMounts:
      - name: jenkins-home
        mountPath: /var/jenkins_home
      {{- if .Values.jenkins.jcasc.enabled }}
      - name: jenkins-config
        mountPath: /var/jenkins_home/casc_configs
      {{- end }}
      - name: jenkins-plugins
        mountPath: /usr/share/jenkins/ref/plugins.txt
        subPath: plugins.txt
      resources:
        {{- toYaml .Values.jenkins.resources | nindent 10 }}
      livenessProbe:
        httpGet:
          path: /login
          port: 8080    # Fixed to match container port
        initialDelaySeconds: 180   # Increased delay
        periodSeconds: 30
        timeoutSeconds: 10
        failureThreshold: 5
      readinessProbe:
        httpGet:
          path: /login
          port: 8080    # Fixed to match container port
        initialDelaySeconds: 120   # Increased delay
```

```
    periodSeconds: 10
    timeoutSeconds: 5
    failureThreshold: 3
volumes:
- name: jenkins-home
  {{- if .Values.jenkins.persistence.enabled }}
  persistentVolumeClaim:
    claimName: {{ include "jenkins-chart.fullname" . }}-pvc
  {{- else }}
  emptyDir: {}
  {{- end }}
{{- if .Values.jenkins.jcasc.enabled }}
- name: jenkins-config
  configMap:
    name: {{ include "jenkins-chart.fullname" . }}-config
    items:
    - key: jenkins.yaml
      path: jenkins.yaml
{{- end }}
- name: jenkins-plugins
  configMap:
    name: {{ include "jenkins-chart.fullname" . }}-config
    items:
    - key: plugins.txt
      path: plugins.txt
{{- with .Values.nodeSelector }}
nodeSelector:
  {{- toYaml . | nindent 8 }}
{{- end }}
{{- with .Values.affinity }}
affinity:
  {{- toYaml . | nindent 8 }}
{{- end }}
{{- with .Values.tolerations }}
tolerations:
  {{- toYaml . | nindent 8 }}
{{- end }}
```

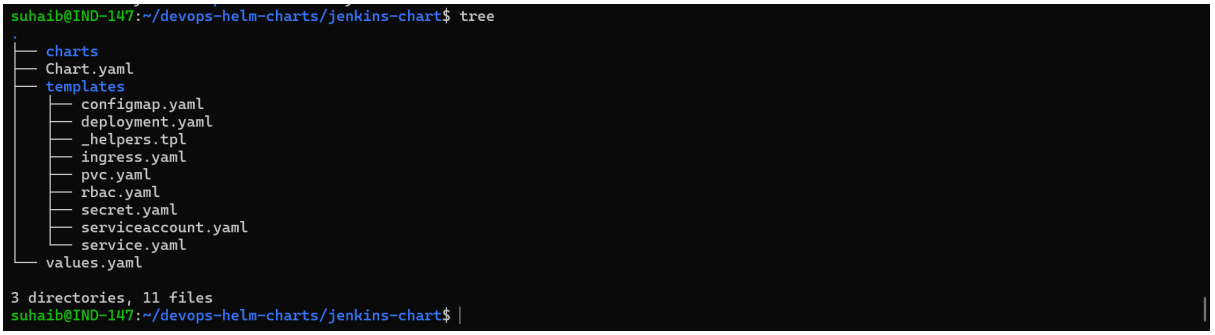
templates/service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: {{ include "jenkins-chart.fullname" . }}-service
  labels:
    {{- include "jenkins-chart.labels" . | nindent 4 }}
spec:
  type: {{ .Values.jenkins.service.type }}
  ports:
  - port: {{ .Values.jenkins.service.port }}
    targetPort: {{ .Values.jenkins.service.targetPort }}
    protocol: TCP
    name: http
  - port: 50000
    targetPort: 50000
    protocol: TCP
    name: jnlp
  selector:
    {{- include "jenkins-chart.selectorLabels" . | nindent 4 }}
```

templates/ingress.yaml

```
{{- if .Values.ingress.enabled -}}
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: {{ include "jenkins-chart.fullname" . }}-ingress
  labels:
    {{- include "jenkins-chart.labels" . | nindent 4 }}
  {{- with .Values.ingress.annotations }}
  annotations:
    {{- toYaml . | nindent 4 }}
  {{- end }}
spec:
  {{- if .Values.ingress.className }}
  ingressClassName: {{ .Values.ingress.className }}
```

```
  {{- end }}
  {{- if .Values.ingress.tls }}
  tls:
    {{- range .Values.ingress.tls }}
    - hosts:
        {{- range .hosts }}
        - {{ . | quote }}
        {{- end }}
      secretName: {{ .secretName }}
    {{- end }}
  {{- end }}
  rules:
    {{- range .Values.ingress.hosts }}
    - host: {{ .host | quote }}
      http:
        paths:
          {{- range .paths }}
          - path: {{ .path }}
            pathType: {{ .pathType }}
            backend:
              service:
                name: {{ include "jenkins-chart.fullname" $ }}-service
                port:
                  number: {{ $.Values.jenkins.service.port }}
          {{- end }}
    {{- end }}
{{- end }}
```

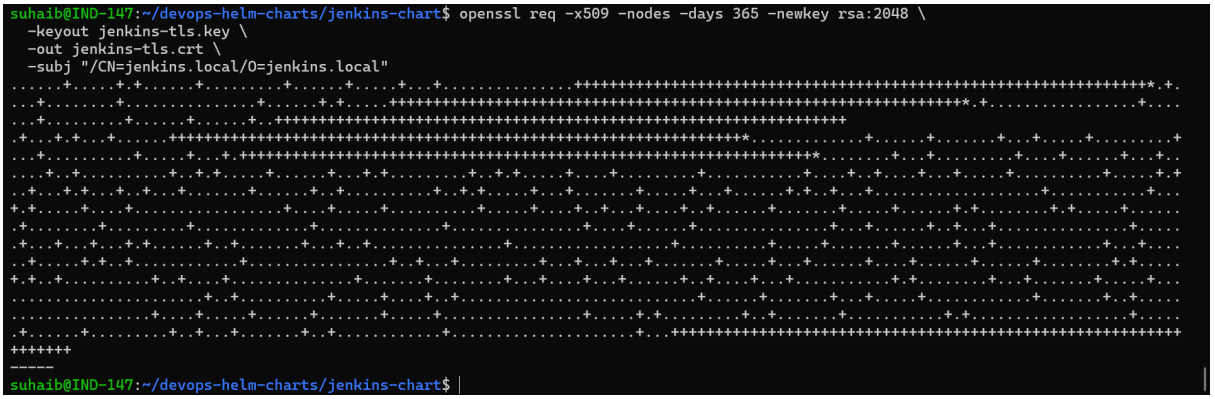


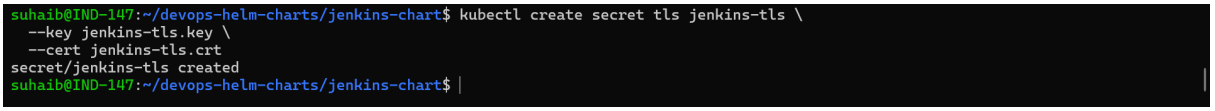
Deployment and Configuration

Step 10: Create TLS Certificate for Ingress

```
# Create self-signed certificate for local development
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout jenkins-tls.key \
  -out jenkins-tls.crt \
  -subj "/CN=jenkins.local/O=jenkins.local"

# Create TLS secret in Kubernetes
kubectl create secret tls jenkins-tls \
  --key jenkins-tls.key \
  --cert jenkins-tls.crt
```





Step 11: Deploy Jenkins Using Helm

```
# Validate the Helm chart
helm lint .

# Dry run to check templates
helm install jenkins-release . --dry-run --debug

# Install Jenkins
helm install jenkins-release . --namespace jenkins --create-namespace

# Check deployment status
```

```
kubectl get pods -n jenkins
kubectl get services -n jenkins
kubectl get ingress -n jenkins
```

```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ helm lint .
==> Linting .
[INFO] Chart.yaml: icon is recommended

1 chart(s) linted, 0 chart(s) failed
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ helm install jenkins-release . --dry-run --debug
install.go:225: 2025-06-04 11:37:11.76140538 +0530 IST m=+0.906565896 [debug] Original chart version: ""
install.go:242: 2025-06-04 11:37:11.777336687 +0530 IST m=+0.922497243 [debug] CHART PATH: /home/suhaib/devops-helm-charts/jenkins-ch
art

NAME: jenkins-release
LAST DEPLOYED: Wed Jun  4 11:37:11 2025
NAMESPACE: default
STATUS: pending-install
REVISION: 1
TEST SUITE: None
USER-SUPPLIED VALUES:
{}

COMPUTED VALUES:
affinity: {}
backup:
  enabled: false
```

```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ helm install jenkins-release . --namespace jenkins --create-namespace
NAME: jenkins-release
LAST DEPLOYED: Wed Jun  4 11:37:46 2025
NAMESPACE: jenkins
STATUS: deployed
REVISION: 1
TEST SUITE: None
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$
```

Step 12: Access Jenkins

```
# Add jenkins.local to your hosts file (in WSL)
echo "127.0.0.1 jenkins.local" | sudo tee -a /etc/hosts

# Port forward if ingress is not working
kubectl port-forward -n jenkins svc/jenkins-release-jenkins-chart-service 8080:8080

# Get Jenkins admin password (if different from values.yaml)
kubectl get secret -n jenkins jenkins-release-jenkins-chart-secret -o jsonpath="{.data.jenkins-admin-password}" | base64 --decode
```

```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ kubectl get secret -n jenkins jenkins-release-jenkins-chart-secret -o jsonpath="{.
data.jenkins-admin-password}" | base64 --decode
admin123suhaib@IND-147:~/devops-helm-charts/jenkins-chart$
```

```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ echo "127.0.0.1 jenkins.local" | sudo tee -a /etc/hosts
[sudo] password for suhaib:
127.0.0.1 jenkins.local
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$
```

Step 13: Verify Installation

```
# Check all resources
kubectl get all -n jenkins

# Check logs
kubectl logs -n jenkins deployment/jenkins-release-jenkins-chart
```

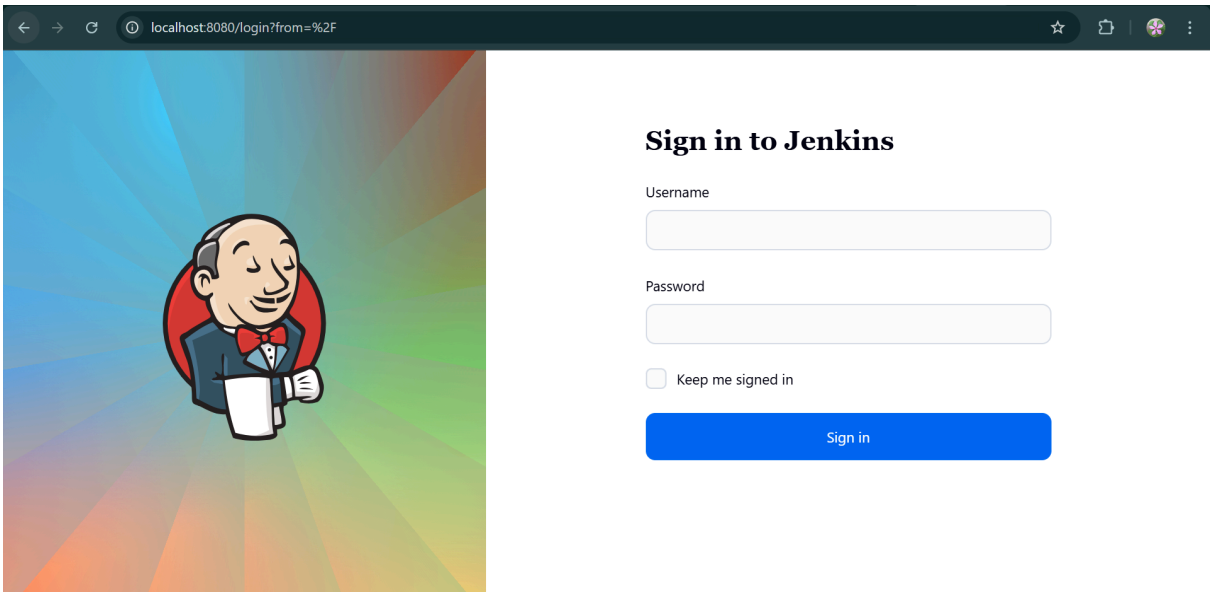
```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ kubectl get all -n jenkins
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/jenkins-release-jenkins-chart-786fb9759d-h464q   1/1     Running   0          4m22s

NAME                                                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)              AGE
service/jenkins-release-jenkins-chart-service        ClusterIP   10.96.68.3   <none>        8080/TCP,50000/TCP   22m

NAME                                                 READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/jenkins-release-jenkins-chart        1/1     1            1           22m

NAME                                                            DESIRED   CURRENT   READY   AGE
replicaset.apps/jenkins-release-jenkins-chart-6fd4c57d94        0         0         0       22m
replicaset.apps/jenkins-release-jenkins-chart-786fb9759d        1         1         1       4m22s
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$
```

```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ kubectl logs -n jenkins deployment/jenkins-release-jenkins-chart
Running from: /usr/share/jenkins/jenkins.war
webroot: /var/jenkins_home/war
2025-06-04 06:26:07.128+0000 [id=1]     INFO    winstone.Logger#logInternal: Beginning extraction from war file
2025-06-04 06:26:07.285+0000 [id=1]     WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2025-06-04 06:26:07.373+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: jetty-10.0.17; built: 2023-10-02T04:04:10.31
4Z; git: a0f5f05abaa6c3aabb7c3d35f10a6f412ab8b05f; jvm 17.0.9+9
2025-06-04 06:26:07.917+0000 [id=1]     INFO    o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find
org.eclipse.jetty.jsp.JettyJspServlet
2025-06-04 06:26:07.992+0000 [id=1]     INFO    o.e.j.s.s.DefaultSessionIdManager#doStart: Session workerName=node0
2025-06-04 06:26:08.897+0000 [id=1]     INFO    hudson.WebAppMain#contextInitialized: Jenkins home directory: /var/jenkins_home found
 at: EnvVars.masterEnvVars.get("JENKINS_HOME")
2025-06-04 06:26:09.047+0000 [id=1]     INFO    o.e.j.s.handler.ContextHandler#doStart: Started w.@28cb9120{Jenkins v2.426.1,/,file:/
//var/jenkins_home/war/,AVAILABLE}{/var/jenkins_home/war}
2025-06-04 06:26:09.071+0000 [id=1]     INFO    o.e.j.server.AbstractConnector#doStart: Started ServerConnector@1329eff{HTTP/1.1, (ht
tp/1.1)}{0.0.0.0:8080}
2025-06-04 06:26:09.117+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: Started Server@be35cd9{STARTING}[10.0.17,sto
=0] @2808ms
2025-06-04 06:26:09.120+0000 [id=25]    INFO    winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2025-06-04 06:26:09.489+0000 [id=32]    INFO    jenkins.InitReactorRunner$1#onAttained: Started initialization
2025-06-04 06:26:09.546+0000 [id=33]    INFO    jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2025-06-04 06:26:11.278+0000 [id=33]    INFO    jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2025-06-04 06:26:11.284+0000 [id=30]    INFO    jenkins.InitReactorRunner$1#onAttained: Started all plugins
2025-06-04 06:26:11.314+0000 [id=30]    INFO    jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2025-06-04 06:26:11.820+0000 [id=33]    INFO    jenkins.InitReactorRunner$1#onAttained: System config loaded
2025-06-04 06:26:11.821+0000 [id=31]    INFO    jenkins.InitReactorRunner$1#onAttained: System config adapted
2025-06-04 06:26:11.822+0000 [id=32]    INFO    jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2025-06-04 06:26:11.824+0000 [id=31]    INFO    jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2025-06-04 06:26:12.158+0000 [id=31]    INFO    jenkins.InitReactorRunner$1#onAttained: Completed initialization
2025-06-04 06:26:12.376+0000 [id=24]    INFO    hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$
```

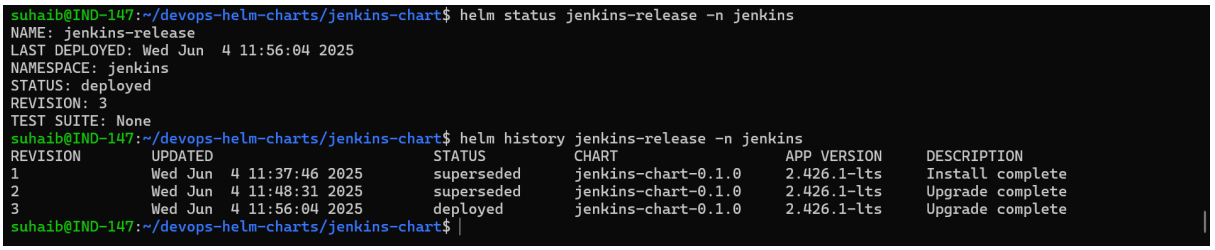


Management and Maintenance

Step 14: Upgrade Jenkins

```
# Update values.yaml with new configuration
# Then upgrade
helm upgrade jenkins-release . --namespace jenkins

# Check upgrade status
helm status jenkins-release -n jenkins
helm history jenkins-release -n jenkins
```

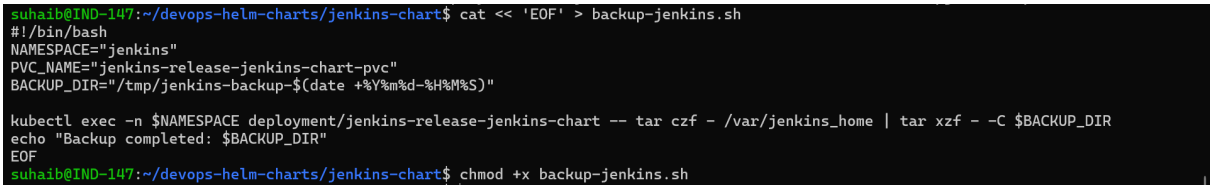


Step 15: Backup and Recovery

```
# Create backup script
cat << 'EOF' > backup-jenkins.sh
#!/bin/bash
NAMESPACE="jenkins"
PVC_NAME="jenkins-release-jenkins-chart-pvc"
BACKUP_DIR="/tmp/jenkins-backup-$(date +%Y%m%d-%H%M%S)"

kubectl exec -n $NAMESPACE deployment/jenkins-release-jenkins-chart -
- tar czf - /var/jenkins_home | tar xzf - -C $BACKUP_DIR
echo "Backup completed: $BACKUP_DIR"
EOF

chmod +x backup-jenkins.sh
```



Step 16: Monitoring and Troubleshooting

```
# Monitor Jenkins resources
kubectl top pods -n jenkins
kubectl describe pod -n jenkins jenkins-release-jenkins-chart-xxx

# Check events
kubectl get events -n jenkins --sort-by=.metadata.creationTimestamp

# Debug common issues
kubectl exec -it -n jenkins deployment/jenkins-release-jenkins-chart -- /bi
n/bash
```

Configuration Guide

Custom Plugin Installation

To add more plugins, update the `values.yaml` file:

```
jenkins:
  installPlugins:
    - your-plugin-name:version
```

External Database Configuration

For production use with external PostgreSQL:

```
database:
  external:
    enabled: true
    host: "postgres.example.com"
    port: 5432
    name: "jenkins"
```

```
    username: "jenkins"
    password: "your-password"
```

Resource Scaling

Adjust resources based on your needs:

```
jenkins:
  resources:
    requests:
      memory: "1Gi"
      cpu: "1000m"
    limits:
      memory: "4Gi"
      cpu: "4000m"
```

Security Configuration

Enable HTTPS and update security settings:

```
ingress:
  tls:
    - secretName: jenkins-tls-prod
      hosts:
        - jenkins.yourdomain.com
```

Cleanup

To remove Jenkins deployment:

```
# Uninstall Jenkins
helm uninstall jenkins-release -n jenkins

# Delete namespace
kubectl delete namespace jenkins

# Delete Kind cluster (if needed)
kind delete cluster --name=devops-cluster
```