Vadakathi Muhammed Suhaib
Technical Apprentice
Emp ID: X48GRSTML
[*muhammed.suhaib@cprime.com*](mailto:muhammed.suhaib@cprime.com)

Develop Helm Charts for Argo CD, Jenkins, Nexus, SonarQube and Deploy

Jenkins Helm Chart Deployment

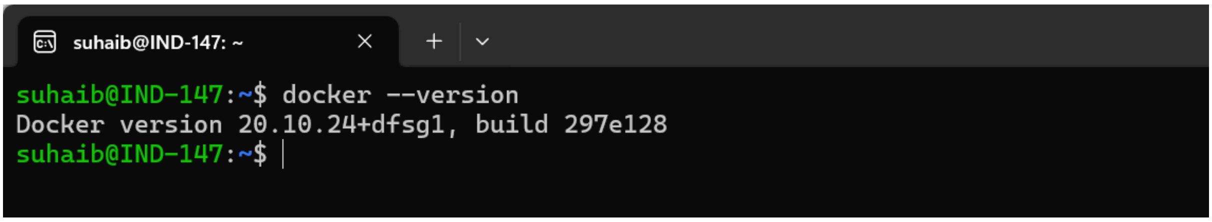
Overview

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

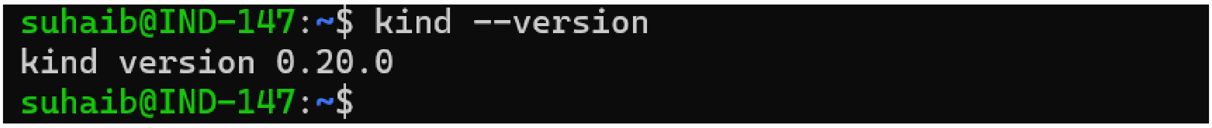
```
# Update system packages
sudo apt update && sudo apt upgrade -y

# Install Docker
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian $(lsb_release -cs) stable" | 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
```



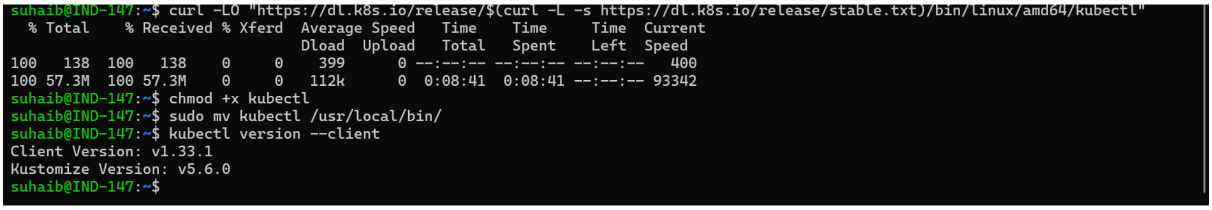
```
suhaib@IND-147:~$ docker --version
Docker version 20.10.24+dfsg1, build 297e128
suhaib@IND-147:~$
```

Step 2: Install Kubernetes (Kind for local development)

```
# Install Kind (Kubernetes in Docker)
curl -Lo ./kind https://github.com/kubernetes-sigs/kind/releases/download/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/stable.txt)/bin/linux/amd64/kubectl"
chmod +x kubectl
sudo mv kubectl /usr/local/bin/

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



```
suhaib@IND-147:~$ kind --version
kind version 0.20.0
suhaib@IND-147:~$
```

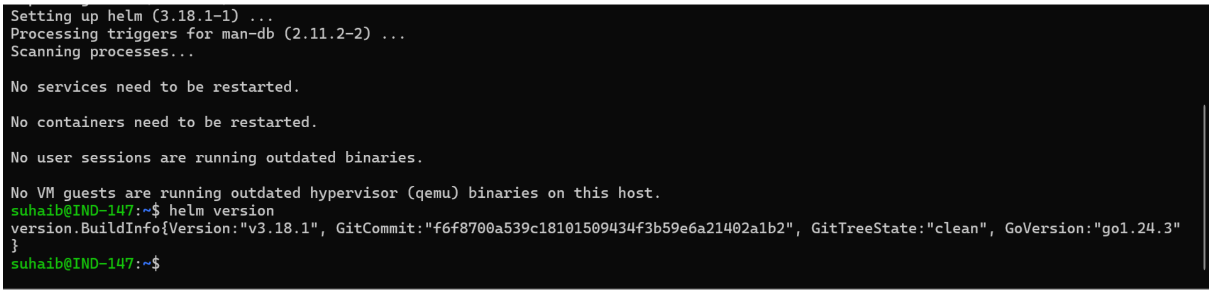


```
suhaib@IND-147:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0    399      0 --:--:-- --:--:-- --:--:--   400
100 57.3M  100 57.3M    0     0   112k      0  0:08:41  0:08:41 --:--:-- 93342
suhaib@IND-147:~$ chmod +x kubectl
suhaib@IND-147:~$ sudo mv kubectl /usr/local/bin/
suhaib@IND-147:~$ kubectl version --client
Client Version: v1.33.1
Kustomize Version: v5.6.0
suhaib@IND-147:~$
```

Step 3: Install Helm

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

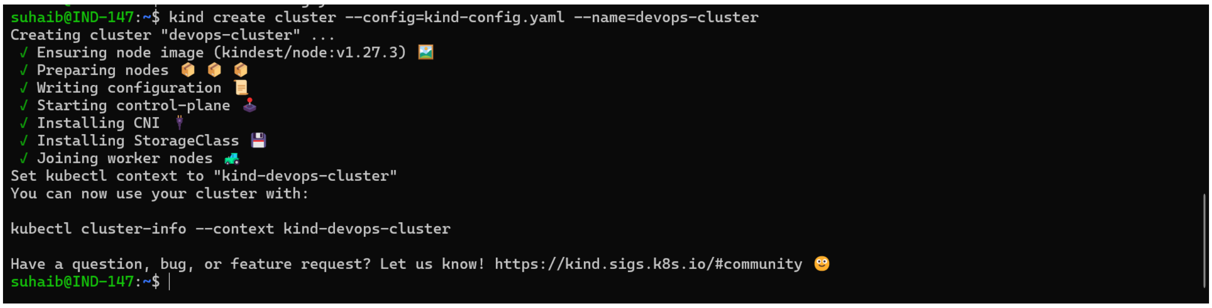
```
r/share/keyrings/helm.gpg > /dev/null
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list
sudo apt update
sudo apt install helm

# Verify Helm installation
helm version
```



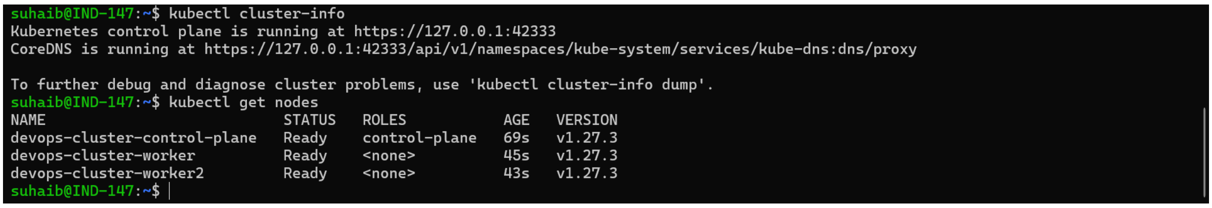
```
Setting up helm (3.18.1-1) ...
Processing triggers for man-db (2.11.2-2) ...
Scanning processes...

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
suhaib@IND-147:~$ helm version
version.BuildInfo{Version:"v3.18.1", GitCommit:"f6f8700a539c18101509434f3b59e6a21402a1b2", GitTreeState:"clean", GoVersion:"go1.24.3"
}
suhaib@IND-147:~$
```

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
```



```
suhaib@IND-147:~$ kind create cluster --config=kind-config.yaml --name=devops-cluster
Creating cluster "devops-cluster" ...
 ✓ Ensuring node image (kindest/node:v1.27.3) 🖼
 ✓ Preparing nodes 📦 📦 📦
 ✓ Writing configuration 📜
 ✓ Starting control-plane 🕹️
 ✓ Installing CNI 🔌
 ✓ Installing StorageClass 💾
 ✓ Joining worker nodes 🚜
Set kubectl context to "kind-devops-cluster"
You can now use your cluster with:

kubectl cluster-info --context kind-devops-cluster

Have a question, bug, or feature request? Let us know! https://kind.sigs.k8s.io/#community 🙂
suhaib@IND-147:~$ 
```

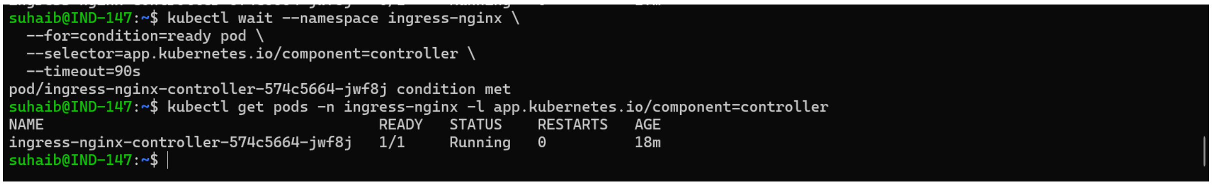


```
suhaib@IND-147:~$ kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:42333
CoreDNS is running at https://127.0.0.1:42333/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
suhaib@IND-147:~$ kubectl get nodes
NAME                           STATUS   ROLES           AGE   VERSION
devops-cluster-control-plane   Ready    control-plane   69s   v1.27.3
devops-cluster-worker          Ready    <none>          45s   v1.27.3
devops-cluster-worker2         Ready    <none>          43s   v1.27.3
suhaib@IND-147:~$ 
```

Step 5: Install NGINX Ingress Controller

```
# Install NGINX Ingress Controller
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/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=controller
```



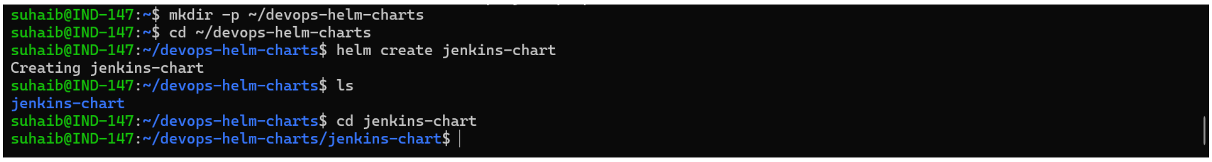
```
suhaib@IND-147:~$ kubectl wait --namespace ingress-nginx \
  --for=condition=ready pod \
  --selector=app.kubernetes.io/component=controller \
  --timeout=90s
pod/ingress-nginx-controller-574c5664-jwf8j condition met
suhaib@IND-147:~$ kubectl get pods -n ingress-nginx -l app.kubernetes.io/component=controller
NAME                                       READY   STATUS    RESTARTS   AGE
ingress-nginx-controller-574c5664-jwf8j   1/1     Running   0          18m
suhaib@IND-147:~$ 
```

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

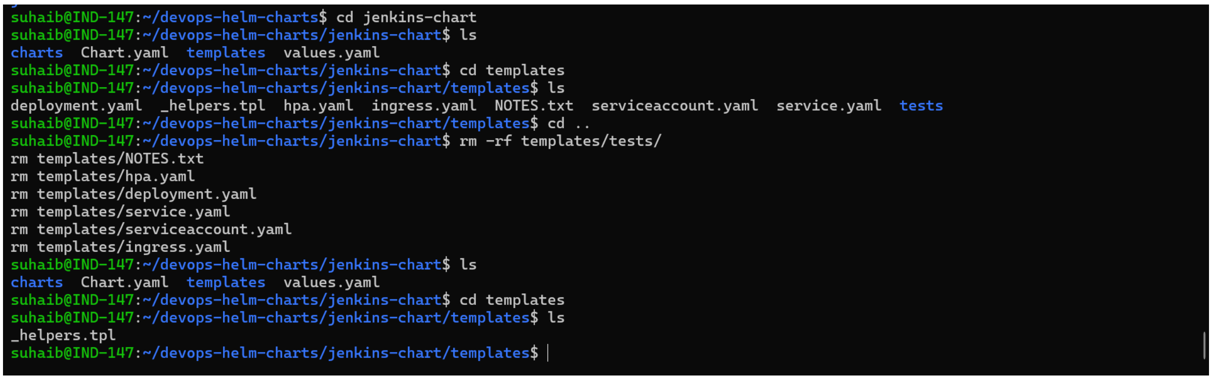
Jenkins:

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
```



```
suhaib@IND-147:~$ mkdir -p ~/devops-helm-charts
suhaib@IND-147:~$ cd ~/devops-helm-charts
suhaib@IND-147:~/devops-helm-charts$ helm create jenkins-chart
Creating jenkins-chart
suhaib@IND-147:~/devops-helm-charts$ ls
jenkins-chart
suhaib@IND-147:~/devops-helm-charts$ cd jenkins-chart
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ 
```



```
suhaib@IND-147:~/devops-helm-charts$ cd jenkins-chart
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ ls
charts  Chart.yaml  templates  values.yaml
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ cd templates
suhaib@IND-147:~/devops-helm-charts/jenkins-chart/templates$ ls
deployment.yaml  _helpers.tpl  hpa.yaml  ingress.yaml  NOTES.txt  serviceaccount.yaml  service.yaml  tests
suhaib@IND-147:~/devops-helm-charts/jenkins-chart/templates$ cd ..
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ 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
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ ls
charts  Chart.yaml  templates  values.yaml
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ cd templates
suhaib@IND-147:~/devops-helm-charts/jenkins-chart/templates$ ls
_helpers.tpl
suhaib@IND-147:~/devops-helm-charts/jenkins-chart/templates$
```

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
```

```yaml
# 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
```

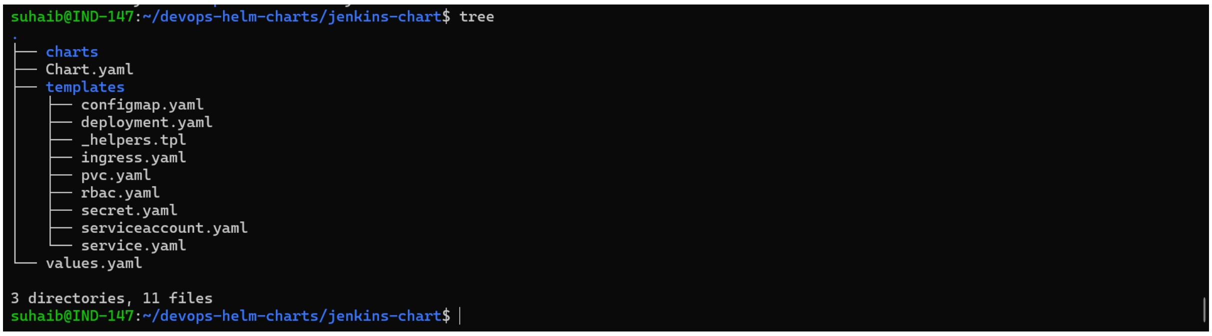
```
    periodSeconds: 10
    timeoutSeconds: 5
    failureThreshold: 3
volumes:
- name: jenkins-home
  {{- if .Values.jenkins.persistence.enabled }}
  persistentVolumeClaim:
    claimName: {{ include "jenkins-chart.fullname" . }}-pvc
  {{- else }}
  emptyDir: {}
  {{- end }}
{{- if .Values.jenkins.jcasc.enabled }}
- 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 }}
```

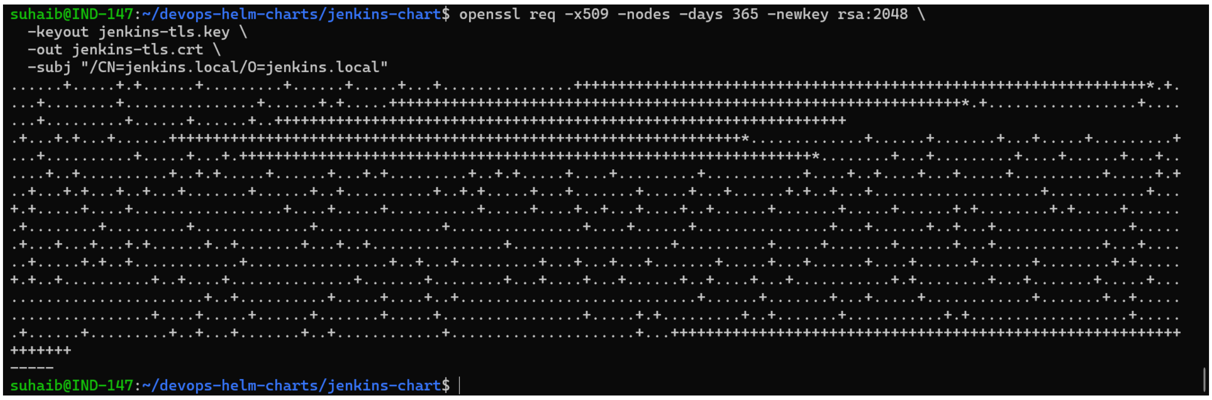


```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ tree
.
├── charts
├── Chart.yaml
├── templates
│   ├── configmap.yaml
│   ├── deployment.yaml
│   ├── _helpers.tpl
│   ├── ingress.yaml
│   ├── pvc.yaml
│   ├── rbac.yaml
│   ├── secret.yaml
│   ├── serviceaccount.yaml
│   └── service.yaml
└── values.yaml

3 directories, 11 files
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$
```

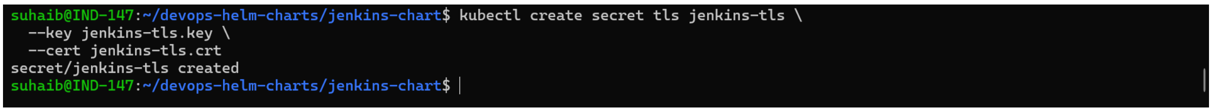
Deployment and Configuration

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
```



```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout jenkins-tls.key \
  -out jenkins-tls.crt \
  -subj "/CN=jenkins.local/O=jenkins.local"
......+.....+.+......+.........+......+.....+...+..............+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+.
...+........+...............+......+.+.....+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+................+....
...+.........+......+......+..+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
.+...+.+...+......+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.............+......+.......+...+.....+.........+
...+..........+.....+...+.+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*........+...+..........+....+......+...+..
....+..+..........+..+.+.....+......+...+.+.........+..+.+.....+.....+..+....+.........+..........+....+..+....+...+.....+..........+.....+.+
..+...+.+...+..+...+.......+......+..+..........+..+.+.....+...+.......+.......+.....+...+......+.+..+...+...................+...........+...
+.+.....+....+.................+....+.....+..........+.....+....+..+....+..+....+...+....+..+......+.......+.....+......+.+........+.+.....+......
.+........+..........+.............+..............+...............+....+..+....+......+..............+...+......+..+...+...............+.....
.+...+...+...+.+......+..+.......+...+..+...............+...............+..........+.....+......+......+...+............+...+....
..+.....+.+..+...........+................+..+...+.........+..+...+.......+...+....+...+......+..+....+...+..........+.+........+...+........+.+....
+.+..+.........+..+...+..............+.......+........+...+...+...+.......+...+......+..+....+...+..........+.+........+...+.......+.....+...
....................+..+..........+.....+....+..+..........................+......+.......+...+.....+............+.......+..+.....
................+....+.....+......+.......+.....+...................+.....+.+.........+..+.......+...........+.+...................+.....
.+......+.........+..+...+.......+..+.............+.....................+...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++
-----
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ 
```

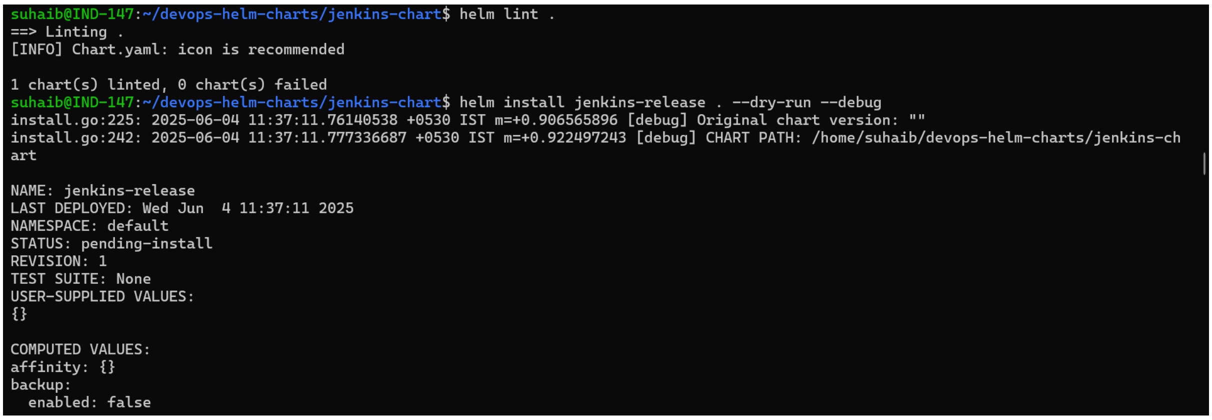


```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ kubectl create secret tls jenkins-tls \
  --key jenkins-tls.key \
  --cert jenkins-tls.crt
secret/jenkins-tls created
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ 
```

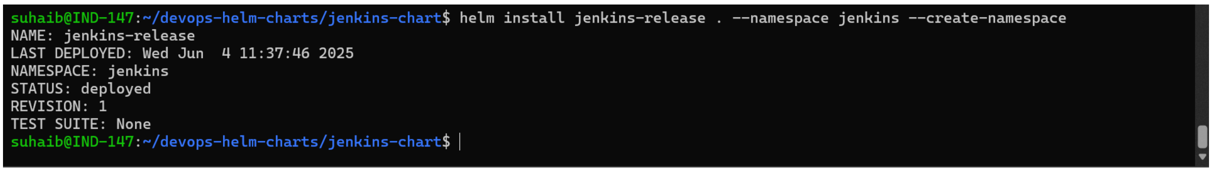
Step 11: Deploy Jenkins Using Helm

```
# 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
```



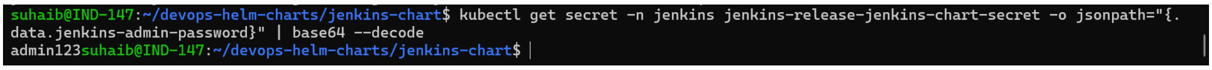
```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ helm lint .
==> Linting .
[INFO] Chart.yaml: icon is recommended

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
```



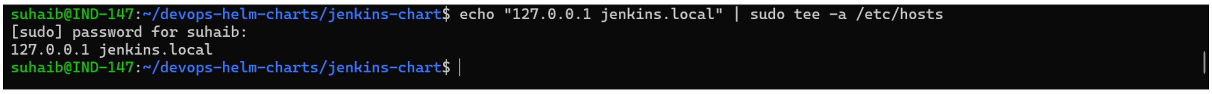
```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ helm install jenkins-release . --namespace jenkins --create-namespace
NAME: jenkins-release
LAST DEPLOYED: Wed Jun  4 11:37:46 2025
NAMESPACE: jenkins
STATUS: deployed
REVISION: 1
TEST SUITE: None
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$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

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
```



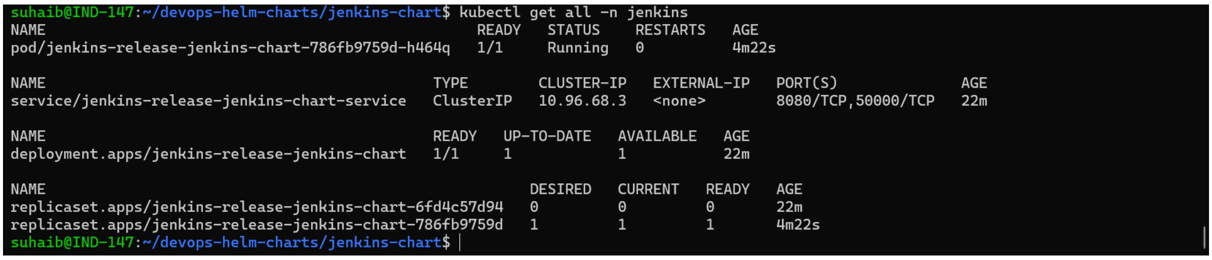
```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ kubectl get secret -n jenkins jenkins-release-jenkins-chart-secret -o jsonpath="{.
data.jenkins-admin-password}" | base64 --decode
admin123suhaib@IND-147:~/devops-helm-charts/jenkins-chart$
```



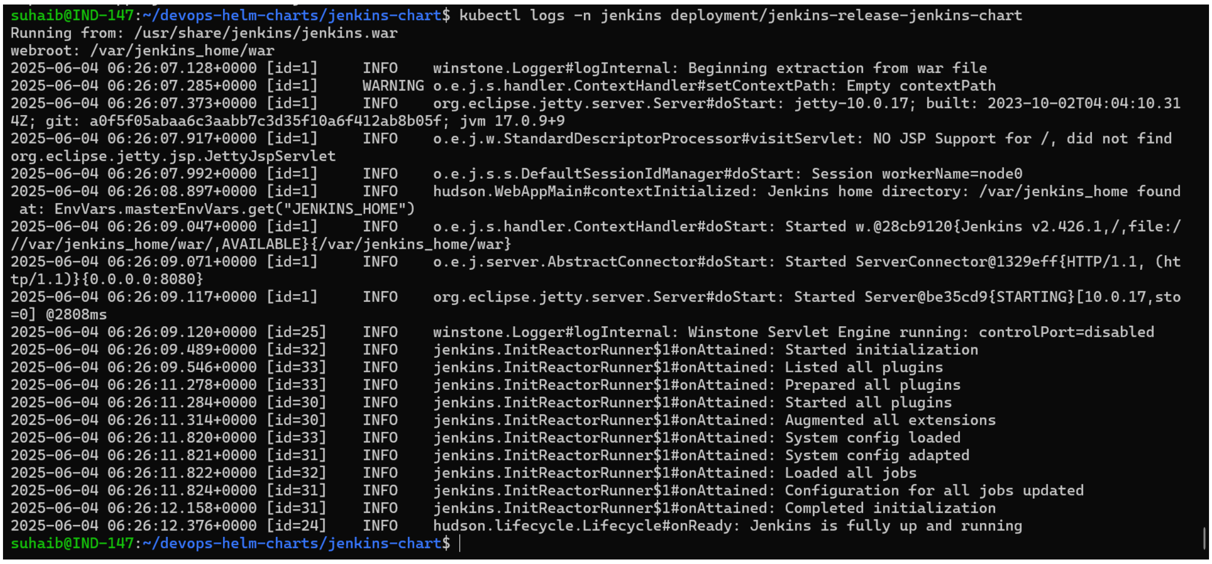
```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ echo "127.0.0.1 jenkins.local" | sudo tee -a /etc/hosts
[sudo] password for suhaib:
127.0.0.1 jenkins.local
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$
```

Step 13: Verify Installation

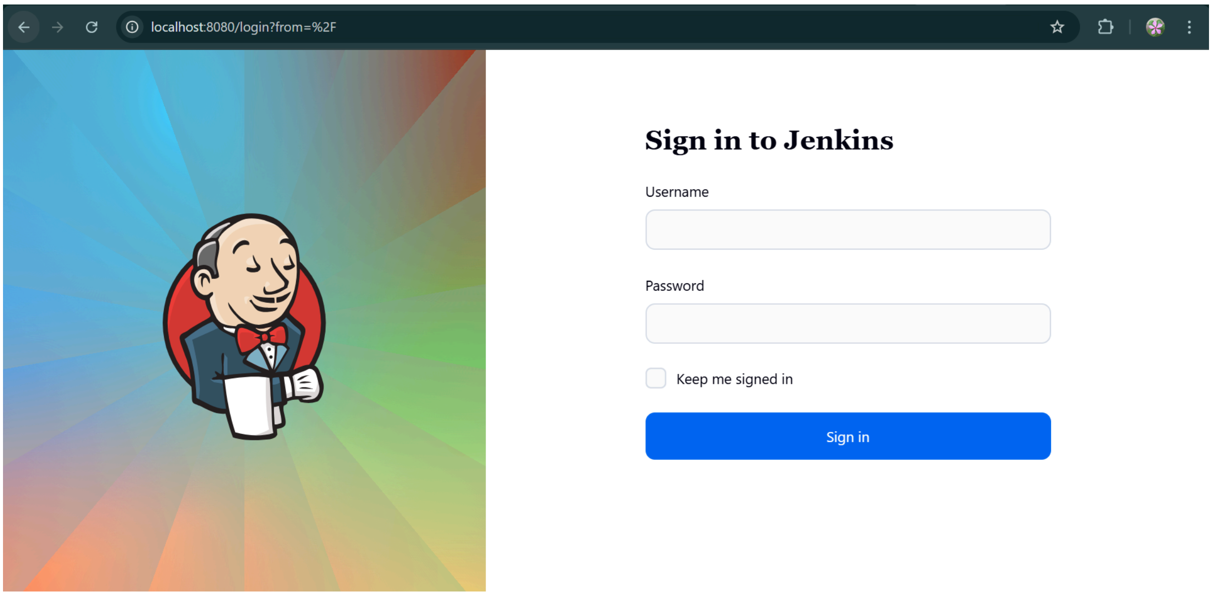
```
# Check all resources
kubectl get all -n jenkins

# 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$1#onAttained: Completed initialization
2025-06-04 06:26:12.376+0000 [id=24]    INFO    hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ 
```



Sign in to Jenkins

Username

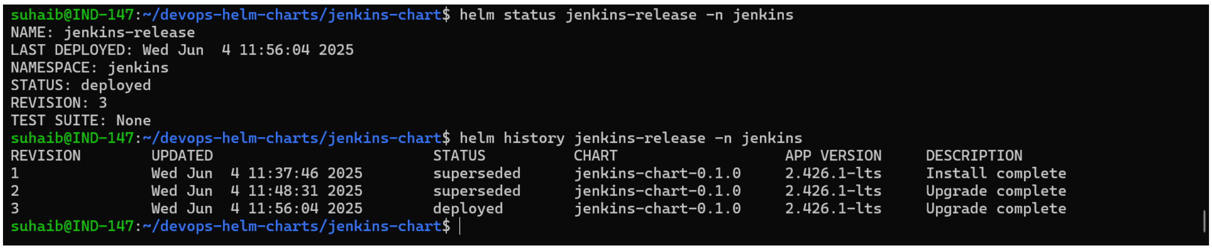
Password

□ Keep me signed in

Management and Maintenance

Step 14: Upgrade Jenkins

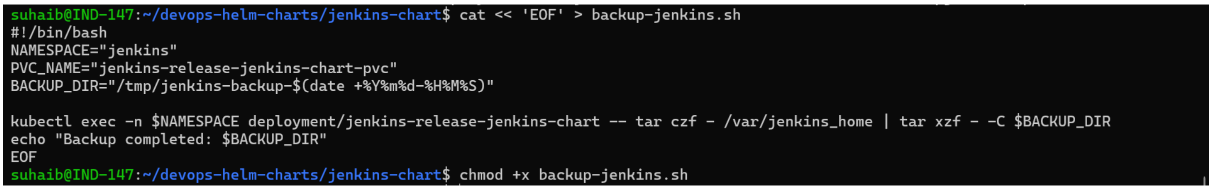
```
# Update values.yaml with new configuration
# Then upgrade
helm upgrade jenkins-release . --namespace jenkins

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



```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ helm status jenkins-release -n jenkins
NAME: jenkins-release
LAST DEPLOYED: Wed Jun  4 11:56:04 2025
NAMESPACE: jenkins
STATUS: deployed
REVISION: 3
TEST SUITE: None
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ helm history jenkins-release -n jenkins
REVISION        UPDATED                         STATUS          CHART                   APP VERSION     DESCRIPTION
1               Wed Jun  4 11:37:46 2025        superseded      jenkins-chart-0.1.0     2.426.1-lts     Install complete
2               Wed Jun  4 11:48:31 2025        superseded      jenkins-chart-0.1.0     2.426.1-lts     Upgrade complete
3               Wed Jun  4 11:56:04 2025        deployed        jenkins-chart-0.1.0     2.426.1-lts     Upgrade complete
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ 
```

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
```



```
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ 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: $BACKUP_DIR"
EOF
suhaib@IND-147:~/devops-helm-charts/jenkins-chart$ 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 -- /bin/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

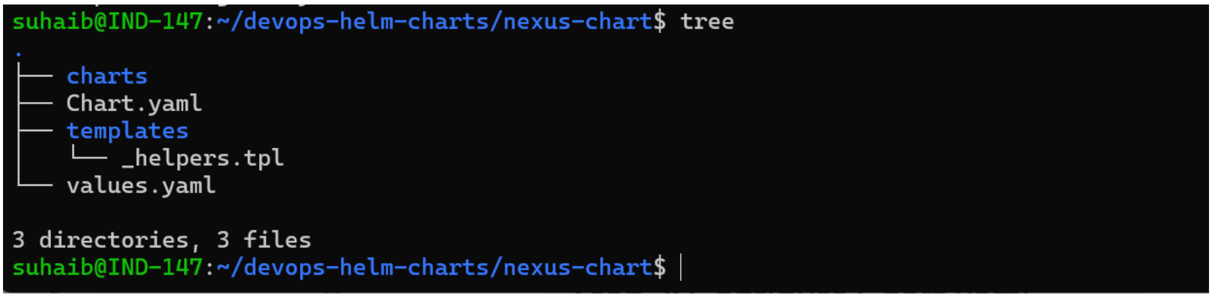
To remove Jenkins deployment:

```
# 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
```

Nexus Helm Chart Deployment

Step 1: Create Nexus Helm Chart Structure

```
# Navigate to your charts directory
cd ~/devops-helm-charts

# Create Nexus Helm chart
helm create nexus-chart
cd nexus-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
```



```
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ tree
.
├── charts
├── Chart.yaml
├── templates
│   └── _helpers.tpl
└── values.yaml

3 directories, 3 files
suhaib@IND-147:~/devops-helm-charts/nexus-chart$
```

Step 2: Configure Chart.yaml

```
# Chart.yaml - Updated
apiVersion: v2
name: nexus-chart
description: A Helm chart for Nexus Repository Manager with custom configuration
type: application
version: 0.2.0  # Increment version
```

```
appVersion: "3.68.1"  # Updated to secure version
keywords:
  - nexus
  - repository
  - artifacts
  - devops
home: https://www.sonatype.com/nexus/repository-oss
sources:
  - https://github.com/sonatype/nexus-public
maintainers:
  - name: DevOps Team
    email: devops@cprime.com
```

Step 3: Configure values.yaml

```
# values.yaml - Updated configuration
nexus:
  image:
    repository: sonatype/nexus3
    tag: "3.68.1"  # Updated to secure version
    pullPolicy: IfNotPresent

  # Add context path configuration
  contextPath: "/"

  # Resource configuration
  resources:
    requests:
      memory: "2Gi"
      cpu: "1000m"
    limits:
      memory: "4Gi"
      cpu: "2000m"

  # Service configuration
  service:
    type: ClusterIP
    port: 8081
    targetPort: 8081
    name: nexus-service

  # Persistence configuration
  persistence:
    enabled: true
    storageClass: "standard"
    size: "20Gi"
    accessMode: ReadWriteOnce

  # Security context
  securityContext:
    runAsUser: 997
    runAsGroup: 997
    fsGroup: 997

  # Nexus admin configuration - FIXED
  admin:
    username: "admin"
    randomPassword: true  # Changed to true to generate admin.password file

  # Environment variables - FIXED
  env:
    - name: NEXUS_SECURITY_RANDOMPASSWORD
      value: "true"  # Changed to true
    - name: NEXUS_JVM_HEAP_MIN
      value: "1024m"
    - name: NEXUS_JVM_HEAP_MAX
      value: "2048m"
    - name: NEXUS_CONTEXT_PATH
      value: "/"

# Database configuration (optional external DB)
database:
  external:
    enabled: false
    host: ""
    port: 5432
    name: "nexus"
    username: "nexus"
    password: ""

# Ingress configuration
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"
    nginx.ingress.kubernetes.io/proxy-body-size: "0"
    nginx.ingress.kubernetes.io/proxy-read-timeout: "300"
    nginx.ingress.kubernetes.io/proxy-send-timeout: "300"
  hosts:
    - host: nexus.local
      paths:
        - path: /
          pathType: Prefix
  tls:
    - secretName: nexus-tls
      hosts:
        - nexus.local

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

# RBAC
rbac:
  create: true
  rules:
    - apiGroups: [""]
```

```
      resources: ["secrets", "configmaps"]
      verbs: ["get", "list", "watch"]
    - apiGroups: [""]
      resources: ["persistentvolumeclaims"]
      verbs: ["get", "list", "watch", "create", "update", "patch"]

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

Step 4: Create Template Files

templates/configmap.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ include "nexus-chart.fullname" . }}-config
  labels:
    {{- include "nexus-chart.labels" . | nindent 4 }}
data:
  nexus.properties: |
    # Nexus configuration
    {{- $randomPassword := "false" }}
    {{- range .Values.nexus.env }}
    {{- if eq .name "NEXUS_SECURITY_RANDOMPASSWORD" }}
    {{- $randomPassword = .value }}
    {{- end }}
    {{- end }}
    nexus.security.randompassword={{ $randomPassword }}
    nexus.cleanup.retainDays=30
    nexus.scripts.allowCreation=true

    # Context path configuration - CRITICAL FIX
    nexus-context-path={{ .Values.nexus.contextPath | default "/" }}

    # Application port
    application-port={{ .Values.nexus.service.targetPort | default 8081 }}

    # Essential nexus-args configuration
    nexus-args=${jetty.etc}/jetty.xml,${jetty.etc}/jetty-http.xml,${jetty.etc}/jetty-requestlog.xml

  {{- if .Values.database.external.enabled }}
  database.properties: |
    nexus.datastore.enabled=true
    nexus.datastore.nexus.type=jdbc
    nexus.datastore.nexus.jdbcUrl=jdbc:postgresql://{{ .Values.database.external.host }}:{{ .Values.database.external.port }}/{{ .Values.database.external.name }}
    nexus.datastore.nexus.username={{ .Values.database.external.username }}
    nexus.datastore.nexus.password={{ .Values.database.external.password }}
  {{- end }}
```

templates/secret.yaml

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

templates/pvc.yaml

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

templates/serviceaccount.yaml

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

templates/deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
```

```
metadata:
  name: {{ include "nexus-chart.fullname" . }}
  labels:
    {{- include "nexus-chart.labels" . | nindent 4 }}
spec:
  replicas: 1
  selector:
    matchLabels:
      {{- include "nexus-chart.selectorLabels" . | nindent 6 }}
  template:
    metadata:
      labels:
        {{- include "nexus-chart.selectorLabels" . | nindent 8 }}
    spec:
      {{- if .Values.serviceAccount.create }}
      serviceAccountName: {{ include "nexus-chart.fullname" . }}-sa
      {{- end }}
      securityContext:
        runAsUser: {{ .Values.nexus.securityContext.runAsUser }}
        runAsGroup: {{ .Values.nexus.securityContext.runAsGroup }}
        fsGroup: {{ .Values.nexus.securityContext.fsGroup }}
      containers:
      - name: nexus
        image: "{{ .Values.nexus.image.repository }}:{{ .Values.nexus.image.tag }}"
        imagePullPolicy: {{ .Values.nexus.image.pullPolicy }}
        # Fixed startup command with proper nexus-args
        command: ["/opt/sonatype/nexus/bin/nexus"]
        args: ["run"]
        env:
        # Environment variables from values.yaml
        {{- range .Values.nexus.env }}
        - name: {{ .name }}
          value: {{ .value | quote }}
        {{- end }}
        # CRITICAL: Set context path environment variable
        - name: NEXUS_CONTEXT_PATH
          value: {{ .Values.nexus.contextPath | default "/" | quote }}
        # Critical: Set NEXUS_ARGS environment variable
        - name: NEXUS_ARGS
          value: "${jetty.etc}/jetty.xml,${jetty.etc}/jetty-http.xml,${jetty.etc}/jetty-requestlog.xml"
        - name: NEXUS_DATA
          value: "/nexus-data"
        - name: NEXUS_HOME
          value: "/opt/sonatype/nexus"
        ports:
        - containerPort: {{ .Values.nexus.service.targetPort }}
          name: http
        volumeMounts:
        - name: nexus-data
          mountPath: /nexus-data
        - name: nexus-config
          mountPath: /opt/sonatype/nexus/etc/nexus.properties
          subPath: nexus.properties
        {{- if .Values.database.external.enabled }}
        - name: nexus-config
          mountPath: /opt/sonatype/nexus/etc/fabric/nexus-store.properties
          subPath: database.properties
        {{- end }}
        livenessProbe:
          httpGet:
            path: {{ if ne (.Values.nexus.contextPath | default "/") "/" }}{{ .Values.nexus.contextPath }}{{ end }}/service/rest/v1/status
            port: http
          initialDelaySeconds: 300
          periodSeconds: 30
          timeoutSeconds: 10
          failureThreshold: 6
        readinessProbe:
          httpGet:
            path: {{ if ne (.Values.nexus.contextPath | default "/") "/" }}{{ .Values.nexus.contextPath }}{{ end }}/service/rest/v1/status
            port: http
          initialDelaySeconds: 180
          periodSeconds: 30
          timeoutSeconds: 10
          failureThreshold: 3
        resources:
          {{- toYaml .Values.nexus.resources | nindent 10 }}
      volumes:
      - name: nexus-config
        configMap:
          name: {{ include "nexus-chart.fullname" . }}-config
          items:
          - key: nexus.properties
            path: nexus.properties
          {{- if .Values.database.external.enabled }}
          - key: database.properties
            path: database.properties
          {{- end }}
      {{- if .Values.nexus.persistence.enabled }}
      - name: nexus-data
        persistentVolumeClaim:
          claimName: {{ include "nexus-chart.fullname" . }}-pvc
      {{- else }}
      - name: nexus-data
        emptyDir: {}
      {{- end }}
      {{- with .Values.nodeSelector }}
      nodeSelector:
        {{- toYaml . | nindent 8 }}
      {{- end }}
      {{- with .Values.affinity }}
      affinity:
        {{- toYaml . | nindent 8 }}
      {{- end }}
      {{- with .Values.tolerations }}
      tolerations:
        {{- toYaml . | nindent 8 }}
      {{- end }}
```

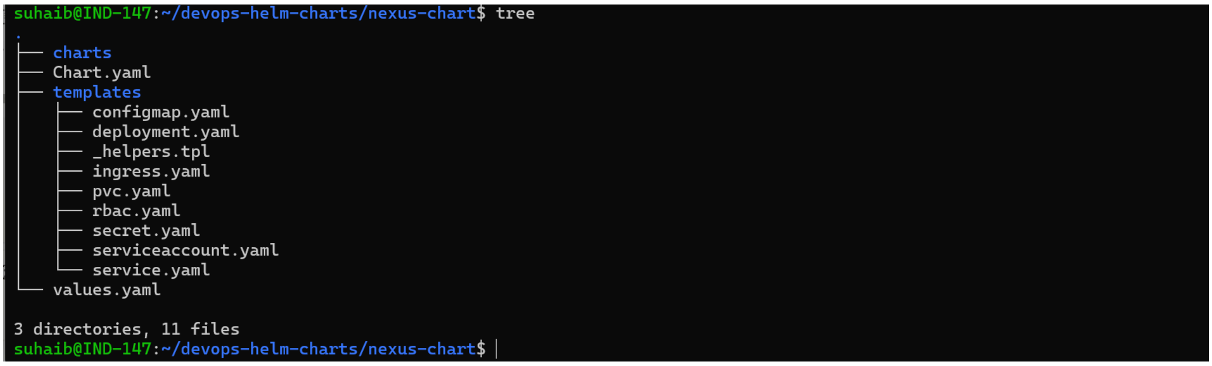
templates/service.yaml

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

templates/ingress.yaml

```
{{- if .Values.ingress.enabled -}}
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: {{ include "nexus-chart.fullname" . }}-ingress
  labels:
    {{- include "nexus-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 "nexus-chart.fullname" $ }}-service
        port:
         number: {{ $.Values.nexus.service.port }}
     {{- end }}
  {{- end }}
{{- end }}
```

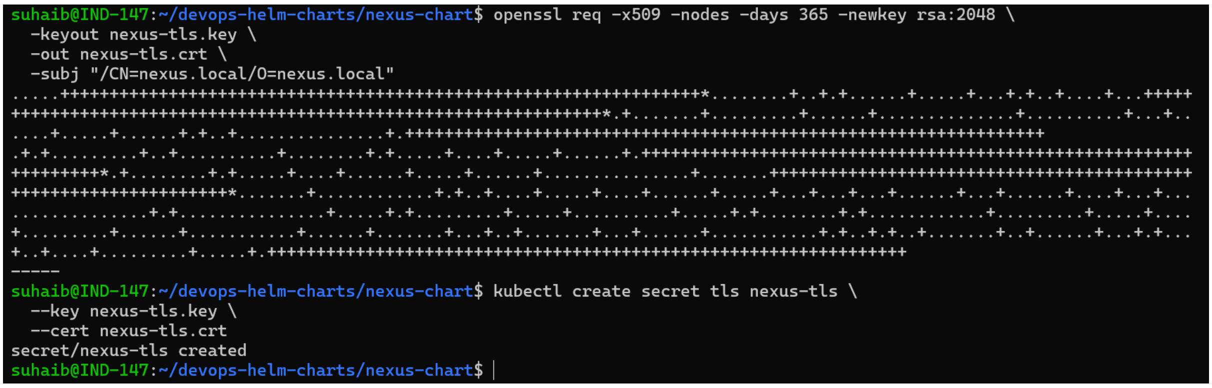


```
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ tree
.
├── charts
├── Chart.yaml
├── templates
│   ├── configmap.yaml
│   ├── deployment.yaml
│   ├── _helpers.tpl
│   ├── ingress.yaml
│   ├── pvc.yaml
│   ├── rbac.yaml
│   ├── secret.yaml
│   ├── serviceaccount.yaml
│   └── service.yaml
└── values.yaml

3 directories, 11 files
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ |
```

Step 5: Deployment Steps

Create TLS Certificate for Ingress

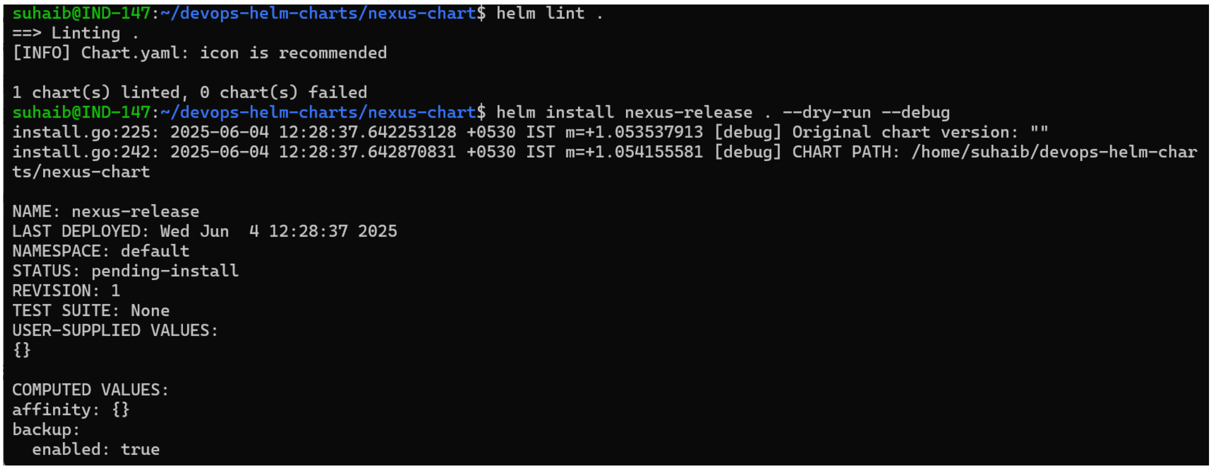
```
# Create self-signed certificate for local development
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout nexus-tls.key \
  -out nexus-tls.crt \
  -subj "/CN=nexus.local/O=nexus.local"

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



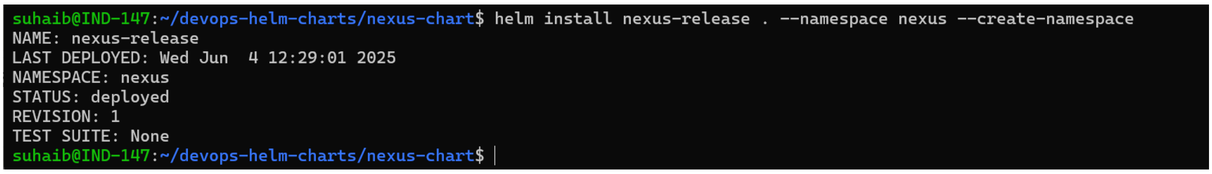
```
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout nexus-tls.key \
  -out nexus-tls.crt \
  -subj "/CN=nexus.local/O=nexus.local"
.....++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.......+..+.+......+.....+...+.+..+....+...+++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+.......+.........+......+..............+..........+...+..
....+.....+......+.+..+..............+.+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
.+.+.........+..+..........+........+.+.....+....+.....+......+.++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++*.+........+.+.....+....+......+.....+......+..............+.......+++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++*.......+...........+.+..+....+.....+....+......+.....+...+...+...+......+...+......+....+...+...
..............+.+..............+.....+.+.........+.....+..........+.....+.+........+.+...........+.........+.....+....
+.........+.....+...........+......+.......+...+..+.......+...+.......+..........+.+..+.+..+.......+..+......+...+.+...
+..+....+.........+.....+.+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-----
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ kubectl create secret tls nexus-tls \
  --key nexus-tls.key \
  --cert nexus-tls.crt
secret/nexus-tls created
suhaib@IND-147:~/devops-helm-charts/nexus-chart$
```

Deploy Nexus Using Helm

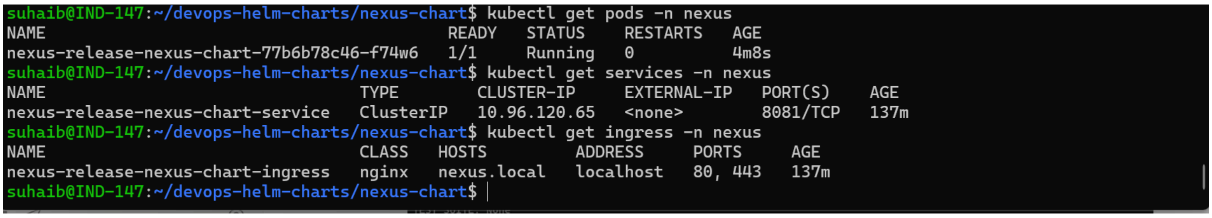
```
# Validate the Helm chart
helm lint .

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

# Install Nexus
helm install nexus-release . --namespace nexus --create-namespace

# Check deployment status
kubectl get pods -n nexus
kubectl get services -n nexus
kubectl get ingress -n nexus
```



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

1 chart(s) linted, 0 chart(s) failed
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ helm install nexus-release . --dry-run --debug
install.go:225: 2025-06-04 12:28:37.642253128 +0530 IST m=+1.053537913 [debug] Original chart version: ""
install.go:242: 2025-06-04 12:28:37.642870831 +0530 IST m=+1.054155581 [debug] CHART PATH: /home/suhaib/devops-helm-char
ts/nexus-chart

NAME: nexus-release
LAST DEPLOYED: Wed Jun  4 12:28:37 2025
NAMESPACE: default
STATUS: pending-install
REVISION: 1
TEST SUITE: None
USER-SUPPLIED VALUES:
{}

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



```
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ helm install nexus-release . --namespace nexus --create-namespace
NAME: nexus-release
LAST DEPLOYED: Wed Jun  4 12:29:01 2025
NAMESPACE: nexus
STATUS: deployed
REVISION: 1
TEST SUITE: None
suhaib@IND-147:~/devops-helm-charts/nexus-chart$
```



```
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ kubectl get pods -n nexus
NAME                                         READY   STATUS    RESTARTS   AGE
nexus-release-nexus-chart-77b6b78c46-f74w6   1/1     Running   0          4m8s
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ kubectl get services -n nexus
NAME                              TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
nexus-release-nexus-chart-service   ClusterIP   10.96.120.65   <none>        8081/TCP   137m
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ kubectl get ingress -n nexus
NAME                              CLASS   HOSTS         ADDRESS     PORTS     AGE
nexus-release-nexus-chart-ingress   nginx   nexus.local   localhost   80, 443   137m
suhaib@IND-147:~/devops-helm-charts/nexus-chart$
```

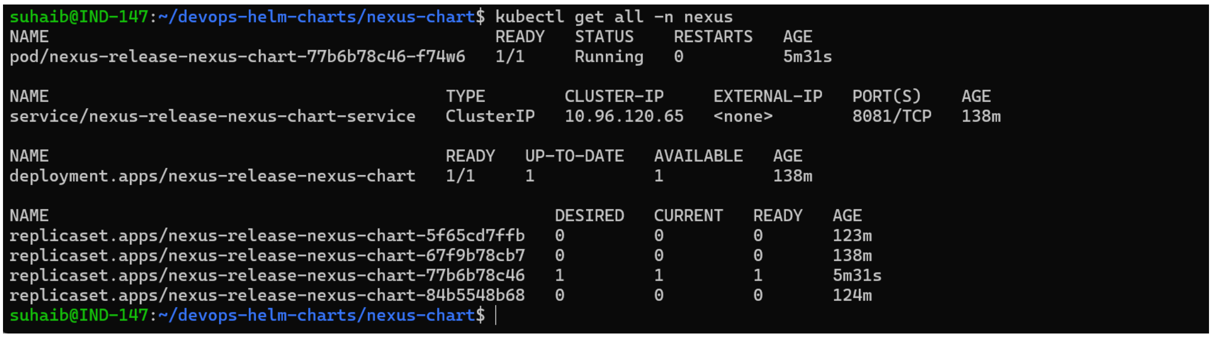
Access Nexus

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

# Port forward if ingress is not working
kubectl port-forward -n nexus svc/nexus-release-nexus-chart-service 8081:8081

# Get Nexus admin password (auto-generated on first run)
kubectl exec -n nexus deployment/nexus-release-nexus-chart -- cat /nexus-data/admin.password
```

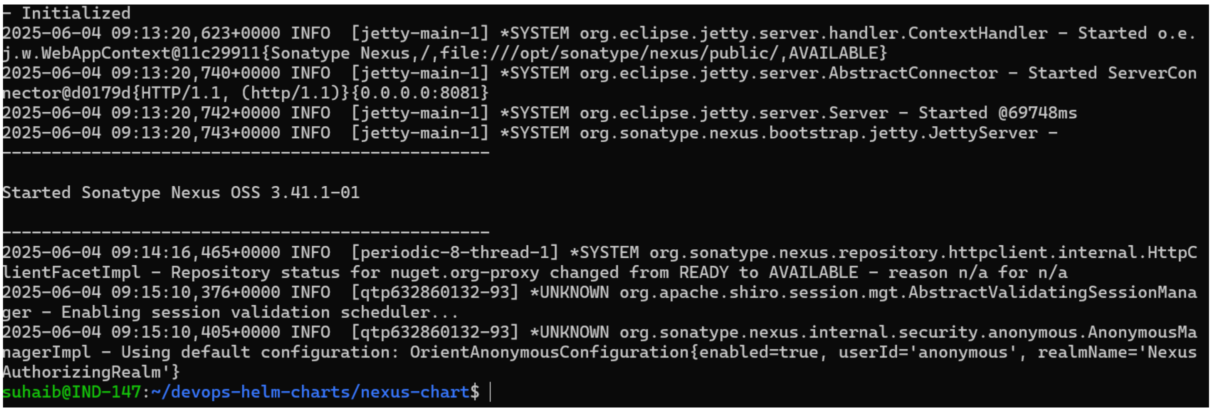
Verify Installation

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

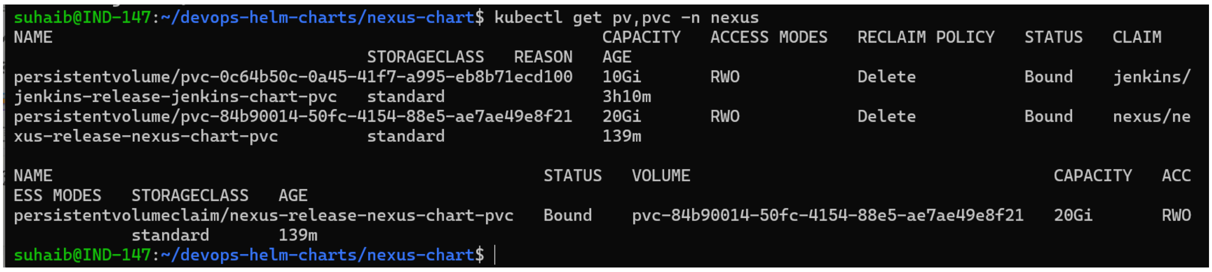
```
# Check logs
kubectl logs -n nexus deployment/nexus-release-nexus-chart

# Check persistent volume
kubectl get pv,pvc -n nexus
```



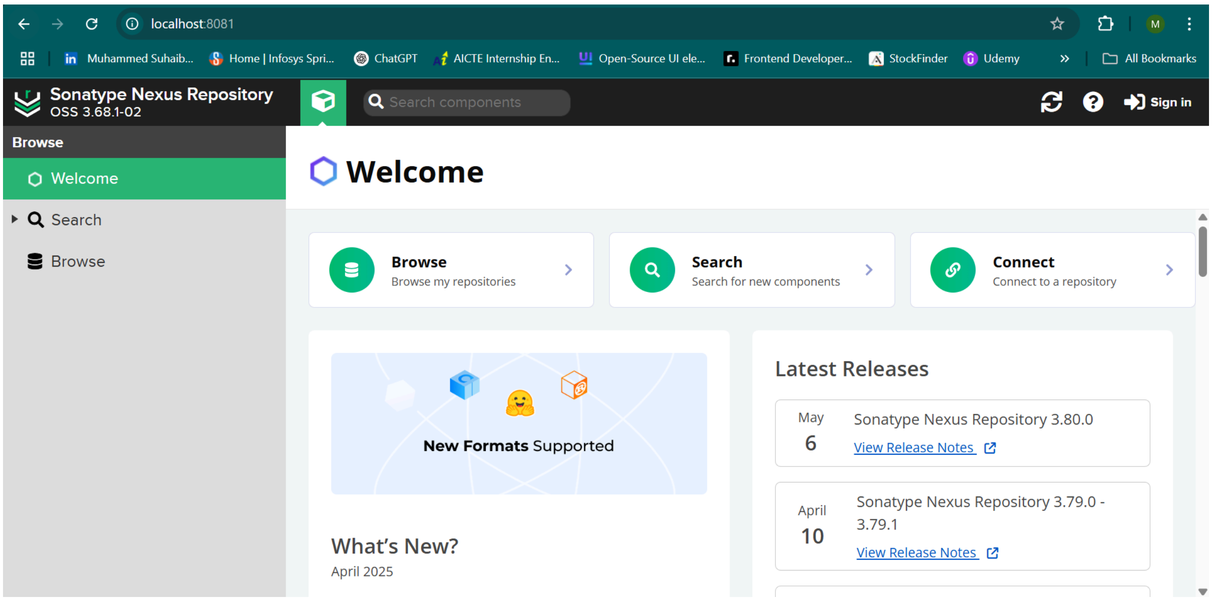
```
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ kubectl get all -n nexus
NAME                                             READY   STATUS    RESTARTS   AGE
pod/nexus-release-nexus-chart-77b6b78c46-f74w6   1/1     Running   0          5m31s

NAME                                        TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
service/nexus-release-nexus-chart-service   ClusterIP   10.96.120.65   <none>        8081/TCP   138m

NAME                                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nexus-release-nexus-chart   1/1     1            1           138m

NAME                                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/nexus-release-nexus-chart-5f65cd7ffb   0         0         0       123m
replicaset.apps/nexus-release-nexus-chart-67f9b78cb7   0         0         0       138m
replicaset.apps/nexus-release-nexus-chart-77b6b78c46   1         1         1       5m31s
replicaset.apps/nexus-release-nexus-chart-84b5548b68   0         0         0       124m
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ |
```



```
- Initialized
2025-06-04 09:13:20,623+0000 INFO  [jetty-main-1] *SYSTEM org.eclipse.jetty.server.handler.ContextHandler - Started o.e.
j.w.WebAppContext@11c29911{Sonatype Nexus,/,file:///opt/sonatype/nexus/public/,AVAILABLE}
2025-06-04 09:13:20,740+0000 INFO  [jetty-main-1] *SYSTEM org.eclipse.jetty.server.AbstractConnector - Started ServerCon
nector@d0179d{HTTP/1.1, (http/1.1)}{0.0.0.0:8081}
2025-06-04 09:13:20,742+0000 INFO  [jetty-main-1] *SYSTEM org.eclipse.jetty.server.Server - Started @69748ms
2025-06-04 09:13:20,743+0000 INFO  [jetty-main-1] *SYSTEM org.sonatype.nexus.bootstrap.jetty.JettyServer -
-------------------------------------------------

Started Sonatype Nexus OSS 3.41.1-01

-------------------------------------------------
2025-06-04 09:14:16,465+0000 INFO  [periodic-8-thread-1] *SYSTEM org.sonatype.nexus.repository.httpclient.internal.HttpC
lientFacetImpl - Repository status for nuget.org-proxy changed from READY to AVAILABLE - reason n/a for n/a
2025-06-04 09:15:10,376+0000 INFO  [qtp632860132-93] *UNKNOWN org.apache.shiro.session.mgt.AbstractValidatingSessionMana
ger - Enabling session validation scheduler...
2025-06-04 09:15:10,405+0000 INFO  [qtp632860132-93] *UNKNOWN org.sonatype.nexus.internal.security.anonymous.AnonymousMa
nagerImpl - Using default configuration: OrientAnonymousConfiguration{enabled=true, userId='anonymous', realmName='Nexus
AuthorizingRealm'}
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ |
```



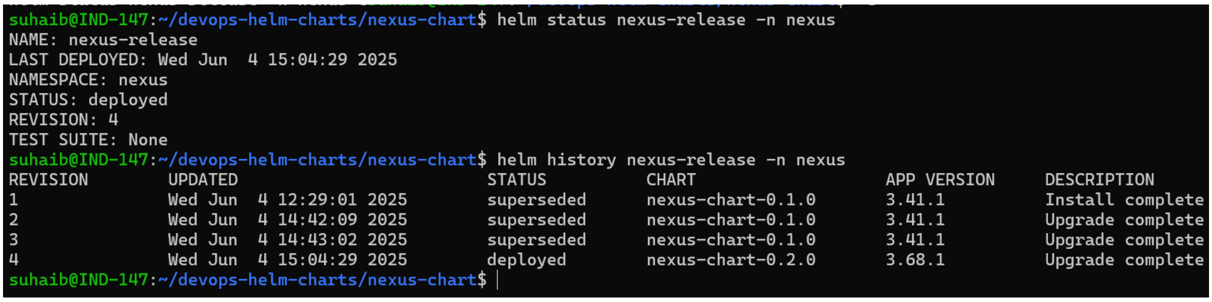
```
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ kubectl get pv,pvc -n nexus
NAME                                                        CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM
                                    STORAGECLASS   REASON   AGE
persistentvolume/pvc-0c64b50c-0a45-41f7-a995-eb8b71ecd100   10Gi       RWO            Delete           Bound    jenkins/
jenkins-release-jenkins-chart-pvc   standard                3h10m
persistentvolume/pvc-84b90014-50fc-4154-88e5-ae7ae49e8f21   20Gi       RWO            Delete           Bound    nexus/ne
xus-release-nexus-chart-pvc         standard                139m

NAME                                                  STATUS   VOLUME                                     CAPACITY   ACC
ESS MODES   STORAGECLASS   AGE
persistentvolumeclaim/nexus-release-nexus-chart-pvc   Bound    pvc-84b90014-50fc-4154-88e5-ae7ae49e8f21   20Gi       RWO
            standard       139m
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ |
```



Step 6: Management and Maintenance

Upgrade Nexus

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

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

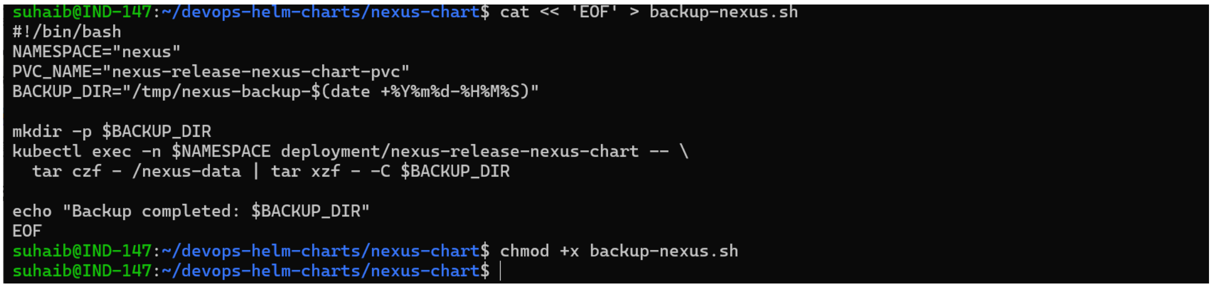


```
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ helm status nexus-release -n nexus
NAME: nexus-release
LAST DEPLOYED: Wed Jun  4 15:04:29 2025
NAMESPACE: nexus
STATUS: deployed
REVISION: 4
TEST SUITE: None
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ helm history nexus-release -n nexus
REVISION        UPDATED                         STATUS          CHART                   APP VERSION     DESCRIPTION
1               Wed Jun  4 12:29:01 2025        superseded      nexus-chart-0.1.0       3.41.1          Install complete
2               Wed Jun  4 14:42:09 2025        superseded      nexus-chart-0.1.0       3.41.1          Upgrade complete
3               Wed Jun  4 14:43:02 2025        superseded      nexus-chart-0.1.0       3.41.1          Upgrade complete
4               Wed Jun  4 15:04:29 2025        deployed        nexus-chart-0.2.0       3.68.1          Upgrade complete
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ |
```

Backup and Recovery

```
# Create backup script
cat << 'EOF' > backup-nexus.sh
#!/bin/bash
```

```
NAMESPACE="nexus"
PVC_NAME="nexus-release-nexus-chart-pvc"
BACKUP_DIR="/tmp/nexus-backup-$(date +%Y%m%d-%H%M%S)"

mkdir -p $BACKUP_DIR
kubectl exec -n $NAMESPACE deployment/nexus-release-nexus-chart -- \
  tar czf - /nexus-data | tar xzf - -C $BACKUP_DIR

echo "Backup completed: $BACKUP_DIR"
EOF

chmod +x backup-nexus.sh
```



```
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ cat << 'EOF' > backup-nexus.sh
#!/bin/bash
NAMESPACE="nexus"
PVC_NAME="nexus-release-nexus-chart-pvc"
BACKUP_DIR="/tmp/nexus-backup-$(date +%Y%m%d-%H%M%S)"

mkdir -p $BACKUP_DIR
kubectl exec -n $NAMESPACE deployment/nexus-release-nexus-chart -- \
  tar czf - /nexus-data | tar xzf - -C $BACKUP_DIR

echo "Backup completed: $BACKUP_DIR"
EOF
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ chmod +x backup-nexus.sh
suhaib@IND-147:~/devops-helm-charts/nexus-chart$ 
```

Monitoring and Troubleshooting

```
# Monitor Nexus resources
kubectl top pods -n nexus
kubectl describe pod -n nexus nexus-release-nexus-chart-xxx

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

# Debug common issues
kubectl exec -it -n nexus deployment/nexus-release-nexus-chart -- /bin/bash
```

Step 7: Configuration Guide

Repository Configuration

After deployment, access Nexus at `http://nexus.local` and configure repositories:

1. **Maven Central Proxy Repository**
2. **Docker Registry**
3. **NPM Registry**
4. **Raw Repository for artifacts**

Security Configuration

1. Change default admin password
2. Create service accounts for CI/CD
3. Configure LDAP/Active Directory integration
4. Set up repository permissions

Integration with Jenkins

Update Jenkins to use Nexus for artifact storage:

```
# In Jenkins values.yaml
jenkins:
  jcasc:
    configScripts:
      nexus-config: |
        unclassified:
          globalNexusConfiguration:
            nxrmConfigs:
              - nxrmUrl: "http://nexus-release-nexus-chart-service.nexus.svc.cluster.local:8081"
                credentialsId: "nexus-credentials"
```

Step 8: Cleanup

```
# Uninstall Nexus
helm uninstall nexus-release -n nexus

# Delete namespace
kubectl delete namespace nexus
```

```
# Delete TLS secret (if needed)
kubectl delete secret nexus-tls
```

Production Considerations

Resource Scaling

```
nexus:
  resources:
    requests:
      memory: "4Gi"
      cpu: "2000m"
    limits:
      memory: "8Gi"
      cpu: "4000m"
```

External Database

For production, use external PostgreSQL:

```
database:
  external:
    enabled: true
    host: "postgres.example.com"
    port: 5432
    name: "nexus"
    username: "nexus"
    password: "your-secure-password"
```

High Availability

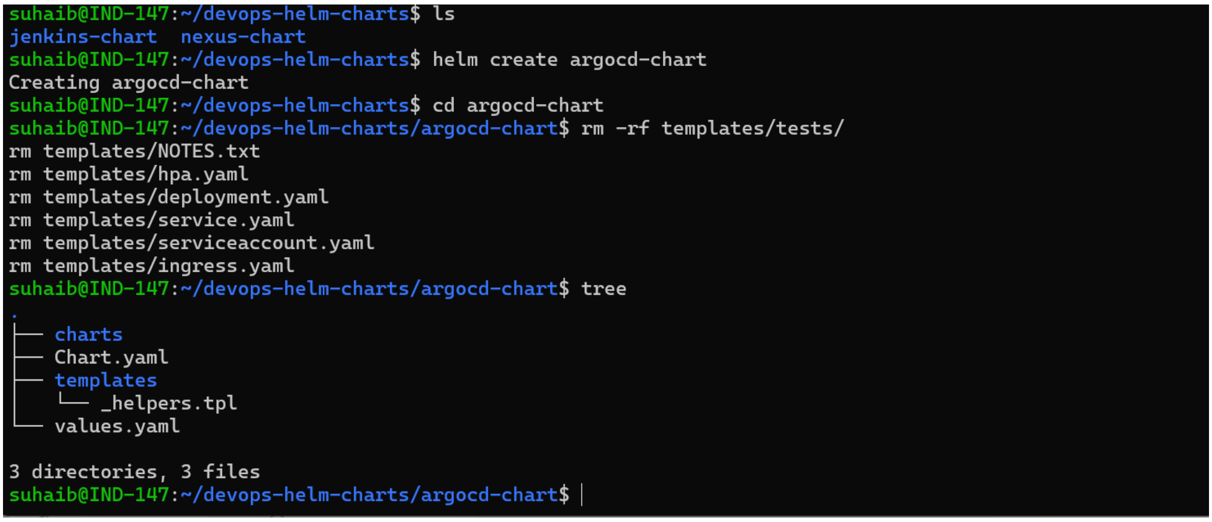
For HA setup, consider:

- External blob storage (S3, Azure Blob)
- Database clustering
- Load balancer configuration
- Multiple replicas with shared storage

Argo CD Helm Chart Deployment

Step 1: Create Argo CD Helm Chart Structure

```
# Navigate to your project directory
cd ~/devops-helm-charts

# Create Argo CD Helm chart
helm create argocd-chart
cd argocd-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
```



```
suhaib@IND-147:~/devops-helm-charts$ ls
jenkins-chart  nexus-chart
suhaib@IND-147:~/devops-helm-charts$ helm create argocd-chart
Creating argocd-chart
suhaib@IND-147:~/devops-helm-charts$ cd argocd-chart
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ 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
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ tree
.
├── charts
├── Chart.yaml
├── templates
│   └── _helpers.tpl
└── values.yaml

3 directories, 3 files
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ |
```

Step 2: Configure Chart.yaml

```
apiVersion: v2
name: argocd
description: A Helm chart for deploying Argo CD in Kubernetes
version: 0.1.0
appVersion: "2.12.4"
```

Step 3: Configure values.yaml

```
argocd:
  namespace: argocd
  image:
    repository: quay.io/argoproj/argocd
    tag: v2.12.4
    pullPolicy: IfNotPresent
  resources:
    requests:
      memory: "512Mi"
      cpu: "500m"
    limits:
      memory: "2Gi"
      cpu: "2000m"
  server:
    replicas: 1
    extraArgs: []
  ingress:
    enabled: true
    hostname: argocd.yourdomain.com
    tls:
      enabled: true
      secretName: argocd-tls
  persistence:
    enabled: true
    storageClass: standard
    size: 8Gi
  admin:
    password: "admin123"  # Change in production
  rbac:
    enabled: true
  config:
    applicationNamespaces: "*"
    url: "https://argocd.yourdomain.com"
```

Step 4: Create Template Files

templates/configmap.yaml

```
# templates/configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: argocd-cm
  namespace: {{ .Values.argocd.namespace }}
  labels:
    app.kubernetes.io/name: argocd-cm
    app.kubernetes.io/part-of: argocd
data:
  application.instanceLabelKey: argocd.argoproj.io/instance
  url: {{ .Values.argocd.config.url | quote }}
  application.namespaces: {{ .Values.argocd.config.applicationNamespaces | quote }}
```

templates/secret.yaml

```
# templates/secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: argocd-secret
  namespace: {{ .Values.argocd.namespace }}
type: Opaque
data:
  admin-password: {{ .Values.argocd.admin.password | b64enc }}
```

templates/pvc.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: {{ .Release.Name }}-argocd-pvc
  namespace: {{ .Values.argocd.namespace }}
spec:
  accessModes:
  - ReadWriteOnce
  storageClassName: {{ .Values.argocd.persistence.storageClass }}
  resources:
    requests:
      storage: {{ .Values.argocd.persistence.size }}
```

templates/rbac.yaml

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: {{ .Release.Name }}-argocd-role
rules:
- apiGroups: [""]
  resources: ["pods", "services", "configmaps", "secrets"]
  verbs: ["get", "list", "watch", "create", "update", "delete"]
- apiGroups: ["apps"]
  resources: ["deployments", "statefulsets"]
  verbs: ["get", "list", "watch", "create", "update", "delete"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: {{ .Release.Name }}-argocd-binding
subjects:
- kind: ServiceAccount
  name: {{ .Release.Name }}-argocd-sa
  namespace: {{ .Values.argocd.namespace }}
roleRef:
  kind: ClusterRole
  name: {{ .Release.Name }}-argocd-role
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: {{ .Release.Name }}-argocd-sa
  namespace: {{ .Values.argocd.namespace }}
```

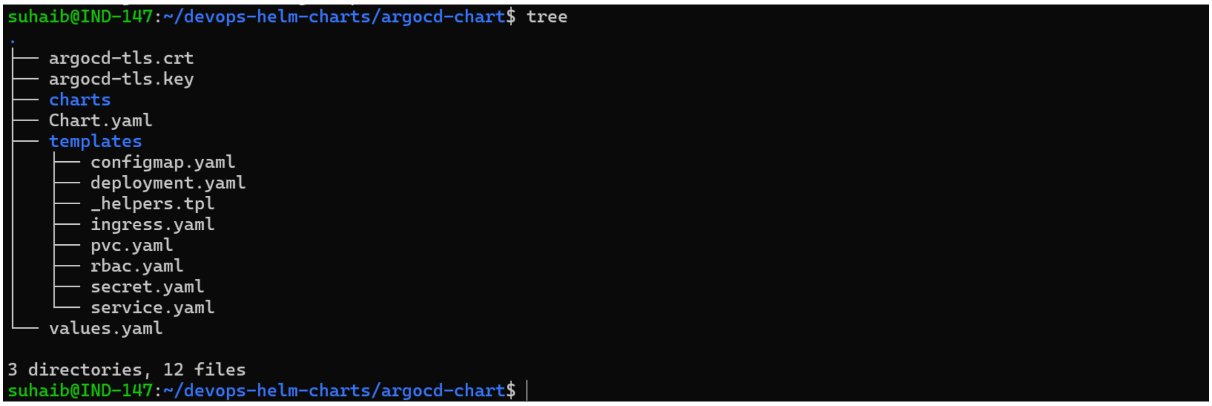
templates/service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-argocd-server
  namespace: {{ .Values.argocd.namespace }}
spec:
  selector:
    app: argocd-server
  ports:
  - port: 80
    targetPort: 8080
    protocol: TCP
    name: http
  type: ClusterIP
```

templates/ingress.yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: {{ .Release.Name }}-argocd-ingress
  namespace: {{ .Values.argocd.namespace }}
  annotations:
    nginx.ingress.kubernetes.io/ssl-redirect: "true"
spec:
  ingressClassName: nginx
```

```
rules:
- host: {{ .Values.argocd.ingress.hostname }}
  http:
    paths:
    - path: /
      pathType: Prefix
      backend:
        service:
          name: {{ .Release.Name }}-argocd-server
          port:
            number: 80
{{- if .Values.argocd.ingress.tls.enabled }}
tls:
- hosts:
  - {{ .Values.argocd.ingress.hostname }}
  secretName: {{ .Values.argocd.ingress.tls.secretName }}
{{- end }}
```



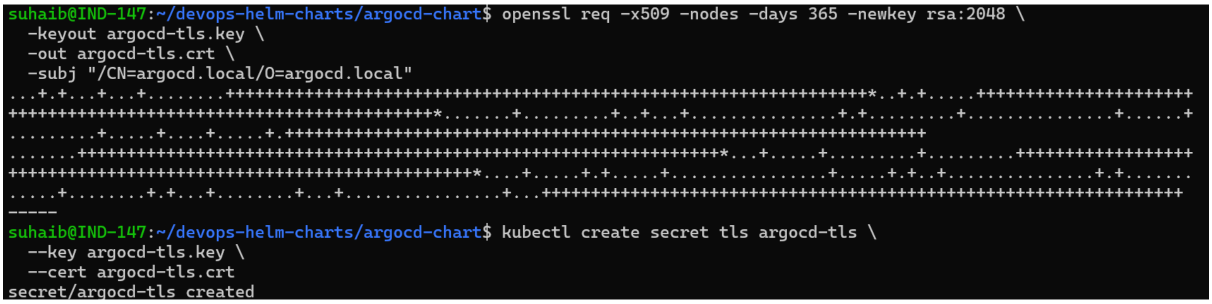
```
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ tree
.
├── argocd-tls.crt
├── argocd-tls.key
├── charts
├── Chart.yaml
├── templates
│   ├── configmap.yaml
│   ├── deployment.yaml
│   ├── _helpers.tpl
│   ├── ingress.yaml
│   ├── pvc.yaml
│   ├── rbac.yaml
│   ├── secret.yaml
│   └── service.yaml
└── values.yaml

3 directories, 12 files
suhaib@IND-147:~/devops-helm-charts/argocd-chart$
```

Step 5: Deployment and Configuration

Create TLS Certificate for Ingress

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

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

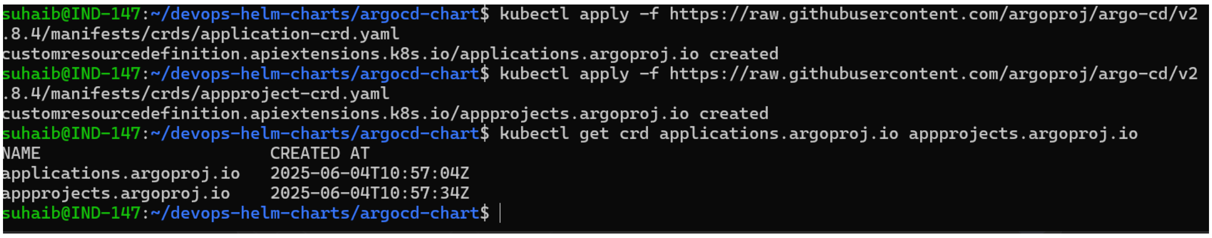


```
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout argocd-tls.key \
  -out argocd-tls.crt \
  -subj "/CN=argocd.local/O=argocd.local"
...+.+...+...+........+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*..+.+.....++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++*.......+.........+..+...+...............+.+.........+...............+......+
.........+.....+....+.....+.+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
.......+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*...+.....+.........+.........++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++*....+.....+.+.....+................+.....+.+..+...............+.+.......
.....+........+.+...+........+...+................+...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-----
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ kubectl create secret tls argocd-tls \
  --key argocd-tls.key \
  --cert argocd-tls.crt
secret/argocd-tls created
```

Install ArgoCD CRDs

Before installing your custom Helm chart, you need to install the ArgoCD CRDs:

```
# Install ArgoCD CRDs
kubectl apply -f https://raw.githubusercontent.com/argoproj/argo-cd/v2.8.4/manifests/crds/application-crd.yaml
kubectl apply -f https://raw.githubusercontent.com/argoproj/argo-cd/v2.8.4/manifests/crds/appproject-crd.yaml

kubectl get crd applications.argoproj.io appprojects.argoproj.io
```



```
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ kubectl apply -f https://raw.githubusercontent.com/argoproj/argo-cd/v2
.8.4/manifests/crds/application-crd.yaml
customresourcedefinition.apiextensions.k8s.io/applications.argoproj.io created
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ kubectl apply -f https://raw.githubusercontent.com/argoproj/argo-cd/v2
.8.4/manifests/crds/appproject-crd.yaml
customresourcedefinition.apiextensions.k8s.io/appprojects.argoproj.io created
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ kubectl get crd applications.argoproj.io appprojects.argoproj.io
NAME                          CREATED AT
applications.argoproj.io      2025-06-04T10:57:04Z
appprojects.argoproj.io       2025-06-04T10:57:34Z
suhaib@IND-147:~/devops-helm-charts/argocd-chart$
```

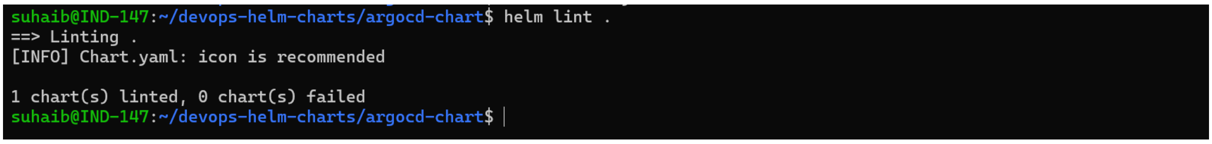
Give Necessary Permissions

```
# Apply the missing permissions directly
kubectl apply -f - <<EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: argocd-release-argocd-role-patch
```

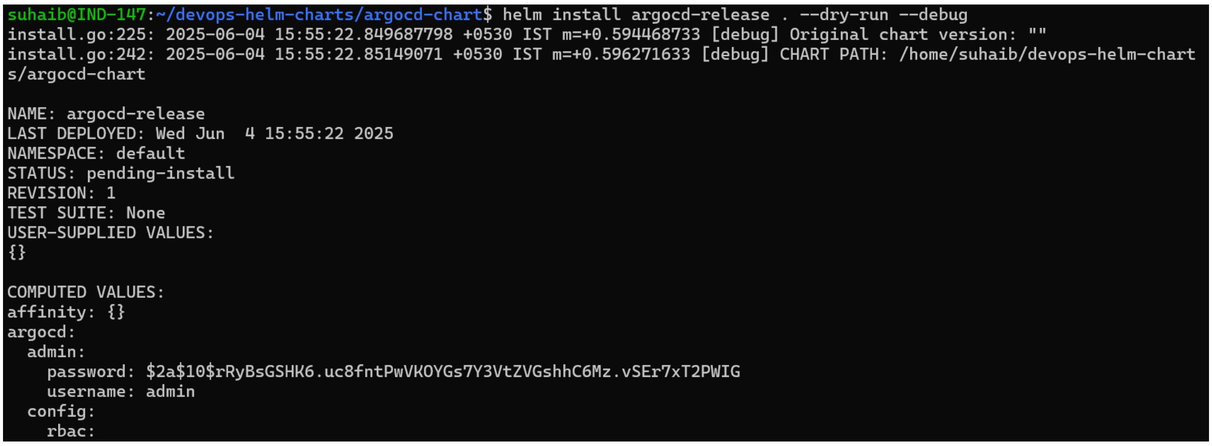
```
rules:
- apiGroups: ["argoproj.io"]
  resources: ["applications", "appprojects", "applicationsets"]
  verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]
- apiGroups: [""]
  resources: ["events", "namespaces"]
  verbs: ["create", "list", "get", "watch"]
EOF

# Bind it to your service account
kubectl apply -f - <<EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: argocd-release-argocd-binding-patch
subjects:
- kind: ServiceAccount
  name: argocd-release-argocd-sa
  namespace: argocd
roleRef:
  kind: ClusterRole
  name: argocd-release-argocd-role-patch
  apiGroup: rbac.authorization.k8s.io
EOF
```

Deploy Argo CD Using Helm

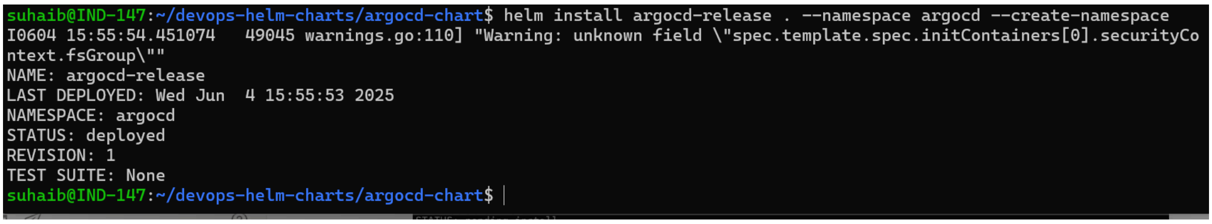
```
# Validate the Helm chart
helm lint .

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

# Install Argo CD
helm install argocd-release . --namespace argocd --create-namespace

# Check deployment status
kubectl get pods -n argocd
kubectl get services -n argocd
kubectl get ingress -n argocd
```



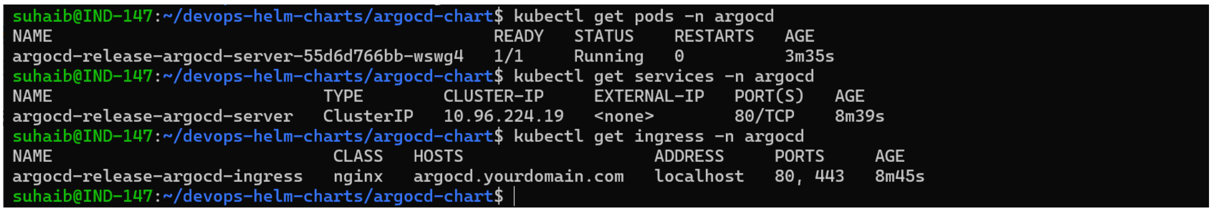
```
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ helm lint .
==> Linting .
[INFO] Chart.yaml: icon is recommended

1 chart(s) linted, 0 chart(s) failed
suhaib@IND-147:~/devops-helm-charts/argocd-chart$
```



```
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ helm install argocd-release . --dry-run --debug
install.go:225: 2025-06-04 15:55:22.849687798 +0530 IST m=+0.594468733 [debug] Original chart version: ""
install.go:242: 2025-06-04 15:55:22.85149071 +0530 IST m=+0.596271633 [debug] CHART PATH: /home/suhaib/devops-helm-chart
s/argocd-chart

NAME: argocd-release
LAST DEPLOYED: Wed Jun  4 15:55:22 2025
NAMESPACE: default
STATUS: pending-install
REVISION: 1
TEST SUITE: None
USER-SUPPLIED VALUES:
{}

COMPUTED VALUES:
affinity: {}
argocd:
  admin:
    password: $2a$10$rRyBsGSHK6.uc8fntPwVKOYGs7Y3VtZVGshhC6Mz.vSEr7xT2PWIG
    username: admin
  config:
    rbac:
```



```
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ helm install argocd-release . --namespace argocd --create-namespace
I0604 15:55:54.451074   49045 warnings.go:110] "Warning: unknown field \"spec.template.spec.initContainers[0].securityCo
ntext.fsGroup\""
NAME: argocd-release
LAST DEPLOYED: Wed Jun  4 15:55:53 2025
NAMESPACE: argocd
STATUS: deployed
REVISION: 1
TEST SUITE: None
suhaib@IND-147:~/devops-helm-charts/argocd-chart$
```

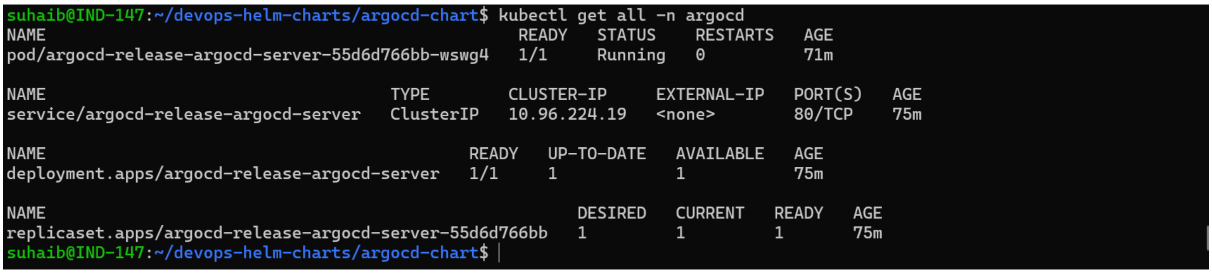


```
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ kubectl get pods -n argocd
NAME                                         READY   STATUS    RESTARTS   AGE
argocd-release-argocd-server-55d6d766bb-wswg4   1/1     Running   0          3m35s
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ kubectl get services -n argocd
NAME                           TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
argocd-release-argocd-server   ClusterIP   10.96.224.19   <none>        80/TCP    8m39s
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ kubectl get ingress -n argocd
NAME                            CLASS   HOSTS                  ADDRESS     PORTS     AGE
argocd-release-argocd-ingress   nginx   argocd.yourdomain.com   localhost   80, 443   8m45s
suhaib@IND-147:~/devops-helm-charts/argocd-chart$
```

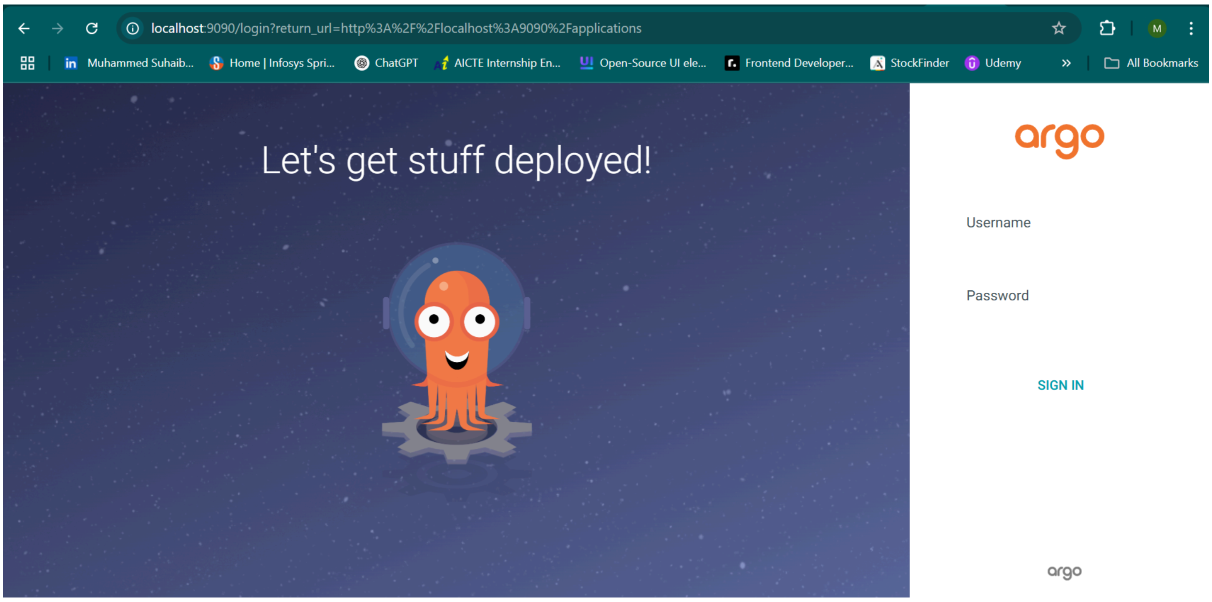
Verify Installation

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

# Check logs
kubectl logs -n argocd deployment/argocd-release-argocd-server

#Port forwarding
```

```
kubectl port-forward svc/argocd-release-argocd-server -n argocd 9090:80
```



```
suhaib@IND-147:~/devops-helm-charts/argocd-chart$ kubectl get all -n argocd
NAME                                          READY   STATUS    RESTARTS   AGE
pod/argocd-release-argocd-server-55d6d766bb-wswg4   1/1     Running   0          71m

NAME                                   TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
service/argocd-release-argocd-server   ClusterIP   10.96.224.19   <none>        80/TCP    75m

NAME                                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/argocd-release-argocd-server   1/1     1            1           75m

NAME                                                      DESIRED   CURRENT   READY   AGE
replicaset.apps/argocd-release-argocd-server-55d6d766bb   1         1         1       75m
suhaib@IND-147:~/devops-helm-charts/argocd-chart$
```



Step 6: Management and Maintenance

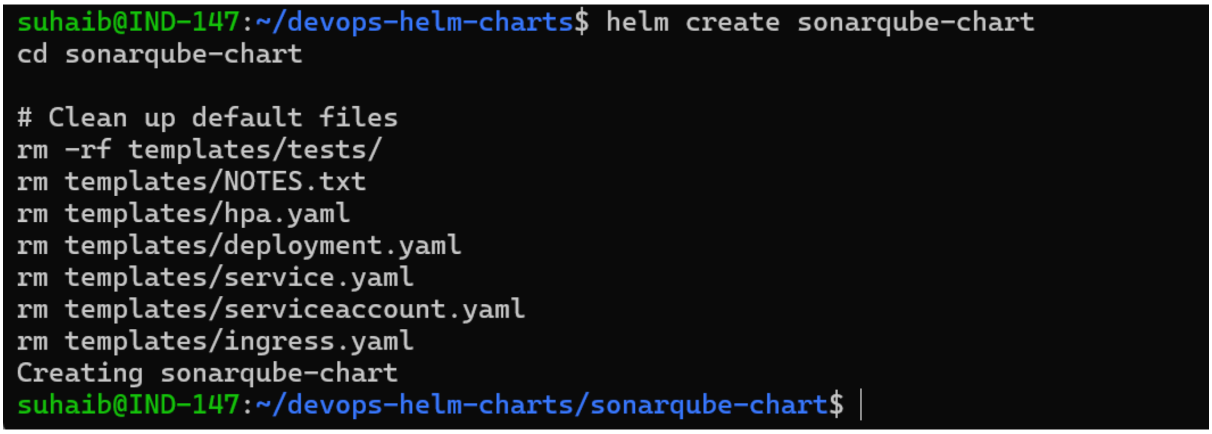
Upgrade Argo CD

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

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

SonarQube Helm Chart Deployment

Step 1: Create SonarQube Helm Chart Structure

```
# Navigate to your project directory
cd ~/devops-helm-charts

# Create SonarQube Helm chart
helm create sonarqube-chart
cd sonarqube-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
```



```
suhaib@IND-147:~/devops-helm-charts$ helm create sonarqube-chart
cd sonarqube-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
Creating sonarqube-chart
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$
```

Step 2: Configure Chart.yaml

```
apiVersion: v2
name: sonarqube
```

```
description: A Helm chart for deploying SonarQube in Kubernetes
version: 0.1.0
appVersion: "10.3.0-community"
```

Step 3: Configure values.yaml

```
sonarqube:
  namespace: sonarqube
  image:
    repository: sonarqube
    tag: "10.3.0-community"
    pullPolicy: IfNotPresent

  resources:
    requests:
      memory: "2Gi"
      cpu: "500m"
    limits:
      memory: "4Gi"
      cpu: "2000m"

  server:
    replicas: 1
    port: 9000
    jvmOpts: "-Xmx2g -Xms512m"

  ingress:
    enabled: true
    hostname: sonarqube.yourdomain.com
    tls:
      enabled: true
      secretName: sonarqube-tls

  persistence:
    enabled: true
    storageClass: standard
    size: 20Gi
    dataSize: 10Gi
    logsSize: 5Gi
    extensionsSize: 5Gi

  admin:
    username: "admin"
    password: "admin123"  # Change in production

  database:
    # Set to true to use external database
    external: false
    # External DB settings (when external: true)
    host: ""
    port: 5432
    name: "sonarqube"
    username: "sonarqube"
    password: "sonarqube123"
    # Internal PostgreSQL settings (when external: false)
    postgresql:
      enabled: true
      image:
        repository: postgres
        tag: "13"
      storage: 10Gi
      username: "sonarqube"
      password: "sonarqube123"
      database: "sonarqube"

  config:
    sonarWebContext: "/"
    sonarJdbcMaxActive: 60
    sonarJdbcMaxIdle: 5
    sonarJdbcMinIdle: 2
    sonarJdbcMaxWait: 5000

  rbac:
    enabled: true
```

Step 4: Create Template Files

templates/configmap.yaml

```
# templates/configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ .Release.Name }}-sonarqube-config
  namespace: {{ .Values.sonarqube.namespace }}
  labels:
    app.kubernetes.io/name: sonarqube
    app.kubernetes.io/instance: {{ .Release.Name }}
    app.kubernetes.io/component: configmap
data:
  sonar.properties: |
    sonar.web.context={{ .Values.sonarqube.config.sonarWebContext }}
    sonar.jdbc.maxActive={{ .Values.sonarqube.config.sonarJdbcMaxActive }}
    sonar.jdbc.maxIdle={{ .Values.sonarqube.config.sonarJdbcMaxIdle }}
    sonar.jdbc.minIdle={{ .Values.sonarqube.config.sonarJdbcMinIdle }}
    sonar.jdbc.maxWait={{ .Values.sonarqube.config.sonarJdbcMaxWait }}
    {{- if .Values.sonarqube.database.external }}
    sonar.jdbc.url=jdbc:postgresql://{{ .Values.sonarqube.database.host }}:{{ .Values.sonarqube.database.port }}/{{ .Values.sonarqube.database.name }}
    {{- else }}
    sonar.jdbc.url=jdbc:postgresql://{{ .Release.Name }}-postgresql:5432/{{ .Values.sonarqube.database.postgresql.database }}
    {{- end }}
    sonar.path.data=/opt/sonarqube/data
    sonar.path.logs=/opt/sonarqube/logs
    sonar.path.temp=/opt/sonarqube/temp
```

templates/secret.yaml

```
# templates/secret.yaml
apiVersion: v1
```

```
kind: Secret
metadata:
  name: {{ .Release.Name }}-sonarqube-secret
  namespace: {{ .Values.sonarqube.namespace }}
  labels:
    app.kubernetes.io/name: sonarqube
    app.kubernetes.io/instance: {{ .Release.Name }}
type: Opaque
data:
  admin-username: {{ .Values.sonarqube.admin.username | b64enc }}
  admin-password: {{ .Values.sonarqube.admin.password | b64enc }}
  {{- if .Values.sonarqube.database.external }}
  db-username: {{ .Values.sonarqube.database.username | b64enc }}
  db-password: {{ .Values.sonarqube.database.password | b64enc }}
  {{- else }}
  db-username: {{ .Values.sonarqube.database.postgresql.username | b64enc }}
  db-password: {{ .Values.sonarqube.database.postgresql.password | b64enc }}
  {{- end }}
```

templates/pvc.yaml

```
# templates/pvc.yaml
{{- if .Values.sonarqube.persistence.enabled }}
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: {{ .Release.Name }}-sonarqube-data-pvc
  namespace: {{ .Values.sonarqube.namespace }}
  labels:
    app.kubernetes.io/name: sonarqube
    app.kubernetes.io/instance: {{ .Release.Name }}
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: {{ .Values.sonarqube.persistence.storageClass }}
  resources:
    requests:
      storage: {{ .Values.sonarqube.persistence.dataSize }}
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: {{ .Release.Name }}-sonarqube-logs-pvc
  namespace: {{ .Values.sonarqube.namespace }}
  labels:
    app.kubernetes.io/name: sonarqube
    app.kubernetes.io/instance: {{ .Release.Name }}
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: {{ .Values.sonarqube.persistence.storageClass }}
  resources:
    requests:
      storage: {{ .Values.sonarqube.persistence.logsSize }}
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: {{ .Release.Name }}-sonarqube-extensions-pvc
  namespace: {{ .Values.sonarqube.namespace }}
  labels:
    app.kubernetes.io/name: sonarqube
    app.kubernetes.io/instance: {{ .Release.Name }}
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: {{ .Values.sonarqube.persistence.storageClass }}
  resources:
    requests:
      storage: {{ .Values.sonarqube.persistence.extensionsSize }}
{{- end }}
```

templates/postgresql.yaml

```
# templates/postgresql.yaml
{{- if and (not .Values.sonarqube.database.external) .Values.sonarqube.database.postgresql.enabled }}
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}-postgresql
  namespace: {{ .Values.sonarqube.namespace }}
  labels:
    app.kubernetes.io/name: postgresql
    app.kubernetes.io/instance: {{ .Release.Name }}
spec:
  replicas: 1
  selector:
    matchLabels:
      app: postgresql
      release: {{ .Release.Name }}
  template:
    metadata:
      labels:
        app: postgresql
        release: {{ .Release.Name }}
    spec:
      containers:
      - name: postgresql
        image: "{{ .Values.sonarqube.database.postgresql.image.repository }}:{{ .Values.sonarqube.database.postgresql.image.tag }}"
        ports:
        - containerPort: 5432
        env:
        - name: POSTGRES_DB
          value: {{ .Values.sonarqube.database.postgresql.database }}
        - name: POSTGRES_USER
          value: {{ .Values.sonarqube.database.postgresql.username }}
        - name: POSTGRES_PASSWORD
          value: {{ .Values.sonarqube.database.postgresql.password }}
        volumeMounts:
        - name: postgresql-data
```

```yaml
          mountPath: /var/lib/postgresql/data
        resources:
          requests:
            memory: "256Mi"
            cpu: "250m"
          limits:
            memory: "512Mi"
            cpu: "500m"
      volumes:
      - name: postgresql-data
        persistentVolumeClaim:
          claimName: {{ .Release.Name }}-postgresql-pvc
---
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-postgresql
  namespace: {{ .Values.sonarqube.namespace }}
  labels:
    app.kubernetes.io/name: postgresql
    app.kubernetes.io/instance: {{ .Release.Name }}
spec:
  selector:
    app: postgresql
    release: {{ .Release.Name }}
  ports:
  - port: 5432
    targetPort: 5432
    protocol: TCP
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: {{ .Release.Name }}-postgresql-pvc
  namespace: {{ .Values.sonarqube.namespace }}
  labels:
    app.kubernetes.io/name: postgresql
    app.kubernetes.io/instance: {{ .Release.Name }}
```

```
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: {{ .Values.sonarqube.persistence.storageClass }}
  resources:
    requests:
      storage: {{ .Values.sonarqube.database.postgresql.storage }}
{{- end }}
```

templates/rbac.yaml

```
# templates/rbac.yaml
{{- if .Values.sonarqube.rbac.enabled }}
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: {{ .Release.Name }}-sonarqube-role
  labels:
    app.kubernetes.io/name: sonarqube
    app.kubernetes.io/instance: {{ .Release.Name }}
rules:
- apiGroups: [""]
  resources: ["pods", "services", "configmaps", "secrets", "persistentvolumeclaims"]
  verbs: ["get", "list", "watch", "create", "update", "delete"]
- apiGroups: ["apps"]
  resources: ["deployments", "statefulsets"]
  verbs: ["get", "list", "watch", "create", "update", "delete"]
- apiGroups: ["extensions", "networking.k8s.io"]
  resources: ["ingresses"]
  verbs: ["get", "list", "watch", "create", "update", "delete"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: {{ .Release.Name }}-sonarqube-binding
  labels:
    app.kubernetes.io/name: sonarqube
    app.kubernetes.io/instance: {{ .Release.Name }}
subjects:
- kind: ServiceAccount
  name: {{ .Release.Name }}-sonarqube-sa
  namespace: {{ .Values.sonarqube.namespace }}
roleRef:
  kind: ClusterRole
  name: {{ .Release.Name }}-sonarqube-role
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: {{ .Release.Name }}-sonarqube-sa
  namespace: {{ .Values.sonarqube.namespace }}
  labels:
    app.kubernetes.io/name: sonarqube
    app.kubernetes.io/instance: {{ .Release.Name }}
{{- end }}
```

templates/deployment.yaml

```
# templates/deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}-sonarqube
  namespace: {{ .Values.sonarqube.namespace }}
  labels:
    app.kubernetes.io/name: sonarqube
    app.kubernetes.io/instance: {{ .Release.Name }}
spec:
  replicas: {{ .Values.sonarqube.server.replicas }}
  selector:
    matchLabels:
      app: sonarqube
      release: {{ .Release.Name }}
  template:
```

```
  metadata:
   labels:
    app: sonarqube
    release: {{ .Release.Name }}
  spec:
   {{- if .Values.sonarqube.rbac.enabled }}
   serviceAccountName: {{ .Release.Name }}-sonarqube-sa
   {{- end }}
   initContainers:
   - name: init-sysctl
    image: busybox:1.35
    command:
    - sh
    - -c
    - |
     sysctl -w vm.max_map_count=524288
     sysctl -w fs.file-max=131072
    securityContext:
     privileged: true
   containers:
   - name: sonarqube
    image: "{{ .Values.sonarqube.image.repository }}:{{ .Values.sonarqube.image.tag }}"
    imagePullPolicy: {{ .Values.sonarqube.image.pullPolicy }}
    ports:
    - containerPort: {{ .Values.sonarqube.server.port }}
     name: http
    env:
    - name: SONAR_JDBC_USERNAME
     valueFrom:
      secretKeyRef:
       name: {{ .Release.Name }}-sonarqube-secret
       key: db-username
    - name: SONAR_JDBC_PASSWORD
     valueFrom:
      secretKeyRef:
       name: {{ .Release.Name }}-sonarqube-secret
       key: db-password
```

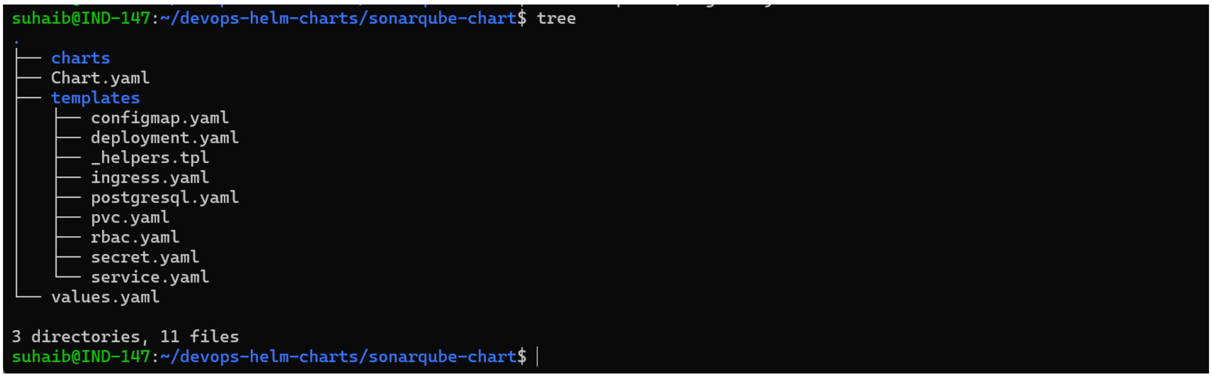
```
        - name: SONAR_WEB_JAVAADDITIONALOPTS
          value: "{{ .Values.sonarqube.server.jvmOpts }}"
        volumeMounts:
        - name: sonarqube-config
          mountPath: /opt/sonarqube/conf/sonar.properties
          subPath: sonar.properties
        {{- if .Values.sonarqube.persistence.enabled }}
        - name: sonarqube-data
          mountPath: /opt/sonarqube/data
        - name: sonarqube-logs
          mountPath: /opt/sonarqube/logs
        - name: sonarqube-extensions
          mountPath: /opt/sonarqube/extensions
        {{- end }}
        resources:
          {{- toYaml .Values.sonarqube.resources | nindent 10 }}
        livenessProbe:
          httpGet:
            path: /api/system/status
            port: {{ .Values.sonarqube.server.port }}
          initialDelaySeconds: 120
          periodSeconds: 30
          timeoutSeconds: 10
        readinessProbe:
          httpGet:
            path: /api/system/status
            port: {{ .Values.sonarqube.server.port }}
          initialDelaySeconds: 30
          periodSeconds: 10
          timeoutSeconds: 5
      volumes:
      - name: sonarqube-config
        configMap:
          name: {{ .Release.Name }}-sonarqube-config
      {{- if .Values.sonarqube.persistence.enabled }}
      - name: sonarqube-data
        persistentVolumeClaim:
          claimName: {{ .Release.Name }}-sonarqube-data-pvc
      - name: sonarqube-logs
        persistentVolumeClaim:
          claimName: {{ .Release.Name }}-sonarqube-logs-pvc
      - name: sonarqube-extensions
        persistentVolumeClaim:
          claimName: {{ .Release.Name }}-sonarqube-extensions-pvc
      {{- end }}
```

templates/service.yaml

```
# templates/service.yaml
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-sonarqube-service
  namespace: {{ .Values.sonarqube.namespace }}
  labels:
    app.kubernetes.io/name: sonarqube
    app.kubernetes.io/instance: {{ .Release.Name }}
spec:
  selector:
    app: sonarqube
    release: {{ .Release.Name }}
  ports:
  - port: 80
    targetPort: {{ .Values.sonarqube.server.port }}
    protocol: TCP
    name: http
  type: ClusterIP
```

templates/ingress.yaml

```
# templates/ingress.yaml
{{- if .Values.sonarqube.ingress.enabled }}
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: {{ .Release.Name }}-sonarqube-ingress
  namespace: {{ .Values.sonarqube.namespace }}
  labels:
    app.kubernetes.io/name: sonarqube
    app.kubernetes.io/instance: {{ .Release.Name }}
  annotations:
    nginx.ingress.kubernetes.io/ssl-redirect: "true"
    nginx.ingress.kubernetes.io/proxy-body-size: "50m"
    nginx.ingress.kubernetes.io/proxy-read-timeout: "300"
    nginx.ingress.kubernetes.io/proxy-send-timeout: "300"
spec:
  ingressClassName: nginx
  rules:
  - host: {{ .Values.sonarqube.ingress.hostname }}
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: {{ .Release.Name }}-sonarqube-service
            port:
              number: 80
  {{- if .Values.sonarqube.ingress.tls.enabled }}
  tls:
  - hosts:
    - {{ .Values.sonarqube.ingress.hostname }}
    secretName: {{ .Values.sonarqube.ingress.tls.secretName }}
  {{- end }}
{{- end }}
```

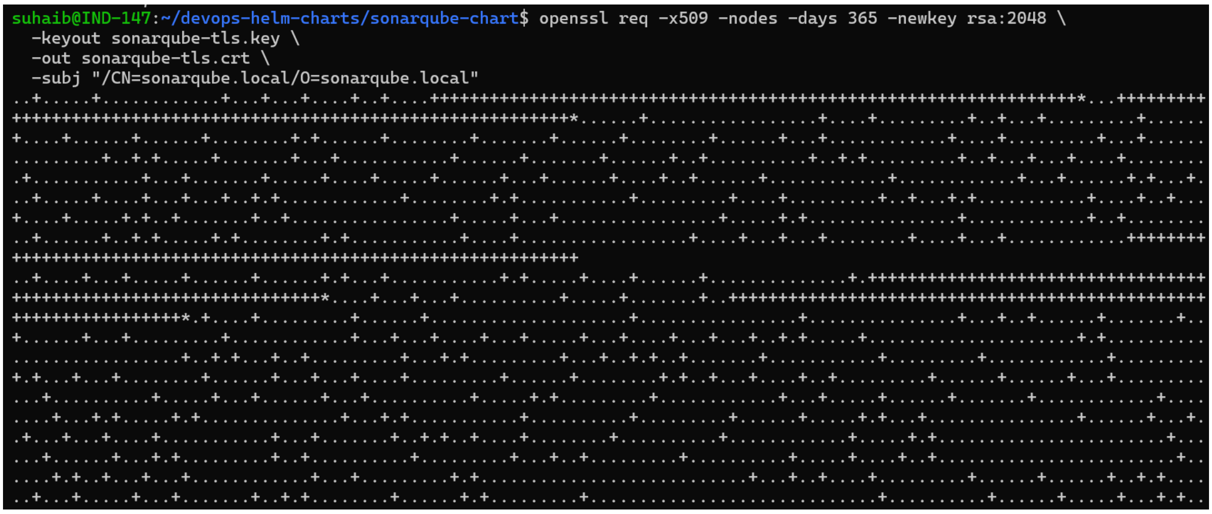


```
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$ tree
.
├── charts
├── Chart.yaml
├── templates
│   ├── configmap.yaml
│   ├── deployment.yaml
│   ├── _helpers.tpl
│   ├── ingress.yaml
│   ├── postgresql.yaml
│   ├── pvc.yaml
│   ├── rbac.yaml
│   ├── secret.yaml
│   └── service.yaml
└── values.yaml

3 directories, 11 files
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$ 
```

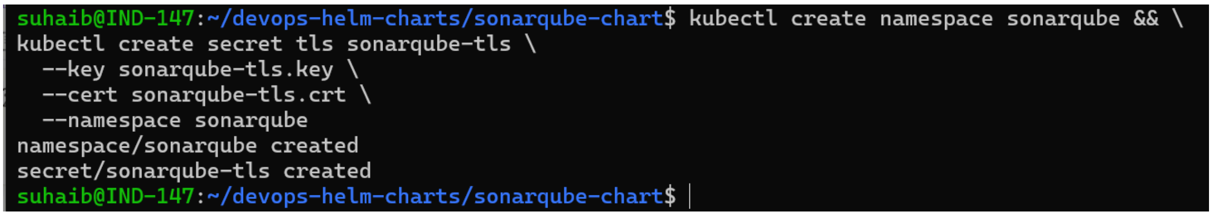
Step 5: Deployment and Configuration Commands

Create TLS Certificate for Ingress

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

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



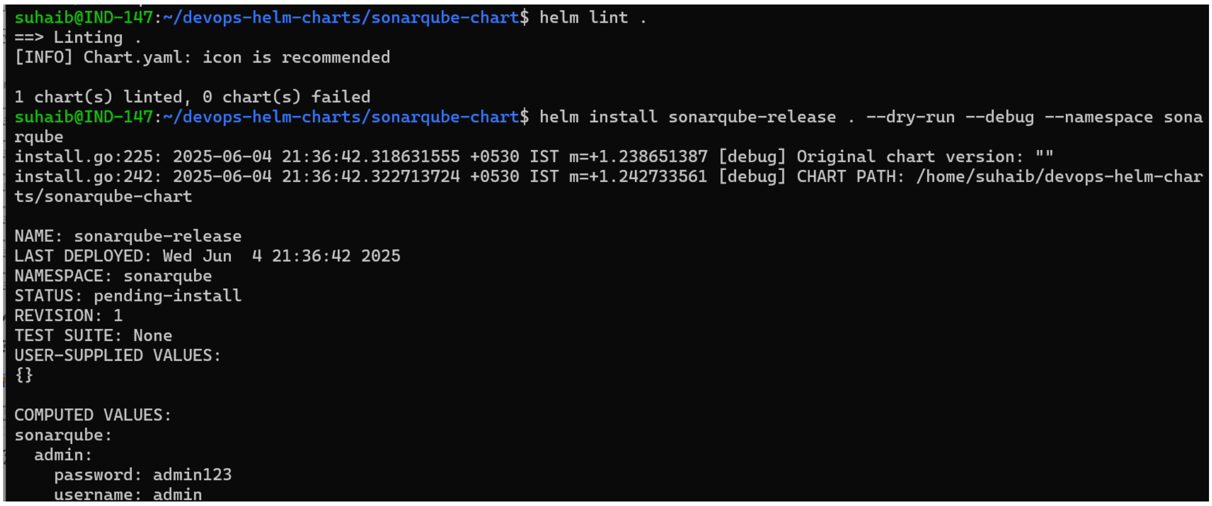
```
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout sonarqube-tls.key \
  -out sonarqube-tls.crt \
  -subj "/CN=sonarqube.local/O=sonarqube.local"
..+.....+...........+...+...+....+..+...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*...+++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++*......+.................+....+.........+..+...+.........+......
+....+......+.......+.........+.+......+........+.......+.......+......+........+......+...+............+....+.........+...+......
.........+..+.+.....+.......+...+...........+......+.......+.......+.......+..+.........+..+.+.........+..+...+...+....+.........
.+..........+...+.......+.....+....+.....+......+...+.......+....+..+......+............+.............+...+.......+.+...+.
..+.....+....+...+...+..+.+............+........+.+...........+.........+....+.........+..+...+.+............+....+..+...
+....+.....+.+..+.......+..+................+.....+...+.................+.....+.+..............+.............+..+.........
..+.......+..+.+.....+.+.........+.+............+....+.............+....+...+...+.........+....+...+.............++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
..+....+...+.....+.......+.......+.+...+............+.+.....+....+.+....+.......+..............+.+++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++*....+...+...+..........+.....+......+........+..++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++*.+....+.........+......+.....................+.......+................+...+..+.......+........+..
+......+...+..........+.............+...+...+....+...+.....+...+..+...+....+...+..+.+.....+......................+.+..........
..................+..+.+...+..+..........+...+.+..........+...+..+..+.+..+........+............+..........+..............+.........
+.+...+...+.........+.......+...+...+....+..........+......+.........+.+..+...+....+..+..........+......+.......+...+..........
...+...........+......+...+.......+...+...........+.....+.+.........+.............+...+.....+.......+.......+..............+...
....+...+.+......+.+...............+...+.+............+...........+..........+.......+.....+.+...+......................+.....+.
.+...+...+....+............+...+........+..+.+..+....+..........+...+...........+.............+.....+...+..+.....................+..
...+.......+...+.+..........+..+...........+............+...+...............+...+..+.........................................+...
....+.+..+...+...+..............+...+..........+.+.....................+...+..+....+...........+.......+.......+..+.+....
..+...+......+...+........+..+.+.........+.......+.+..........+.....+...........................+...........+.......+....+...+.+..
```



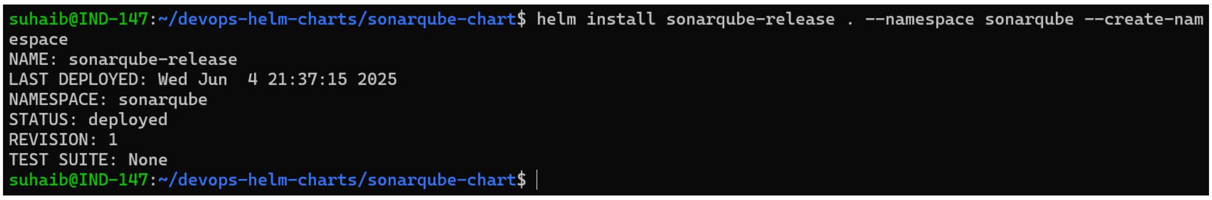
```
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$ kubectl create namespace sonarqube && \
kubectl create secret tls sonarqube-tls \
  --key sonarqube-tls.key \
  --cert sonarqube-tls.crt \
  --namespace sonarqube
namespace/sonarqube created
secret/sonarqube-tls created
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$ 
```

Deploy SonarQube Using Helm

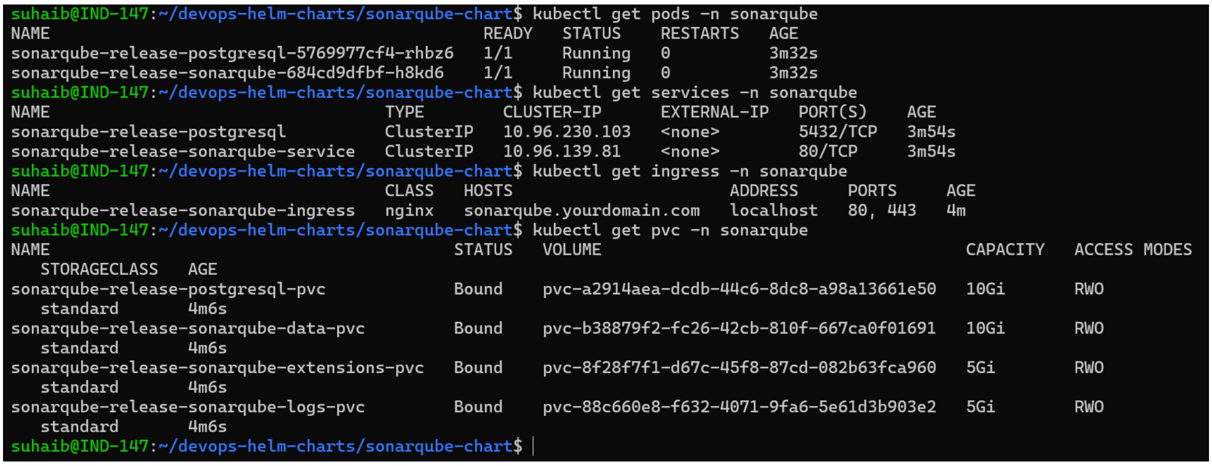
```
# Validate the Helm chart
helm lint .

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

# Install SonarQube
helm install sonarqube-release . --namespace sonarqube --create-namespace

# Check deployment status
kubectl get pods -n sonarqube
kubectl get services -n sonarqube
kubectl get ingress -n sonarqube
kubectl get pvc -n sonarqube
```



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

1 chart(s) linted, 0 chart(s) failed
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$ helm install sonarqube-release . --dry-run --debug --namespace sona
rqube
install.go:225: 2025-06-04 21:36:42.318631555 +0530 IST m=+1.238651387 [debug] Original chart version: ""
install.go:242: 2025-06-04 21:36:42.322713724 +0530 IST m=+1.242733561 [debug] CHART PATH: /home/suhaib/devops-helm-char
ts/sonarqube-chart

NAME: sonarqube-release
LAST DEPLOYED: Wed Jun  4 21:36:42 2025
NAMESPACE: sonarqube
STATUS: pending-install
REVISION: 1
TEST SUITE: None
USER-SUPPLIED VALUES:
{}

COMPUTED VALUES:
sonarqube:
  admin:
    password: admin123
    username: admin
```



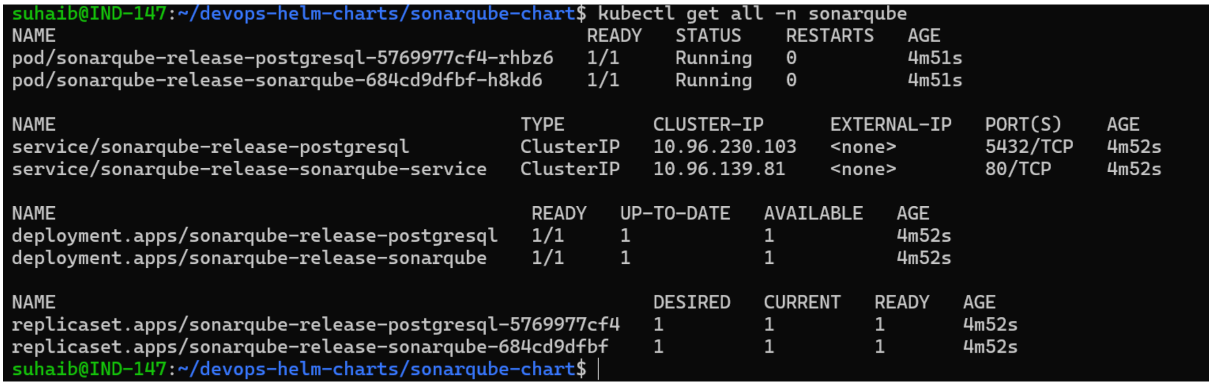
```
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$ helm install sonarqube-release . --namespace sonarqube --create-namespace
NAME: sonarqube-release
LAST DEPLOYED: Wed Jun  4 21:37:15 2025
NAMESPACE: sonarqube
STATUS: deployed
REVISION: 1
TEST SUITE: None
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$
```



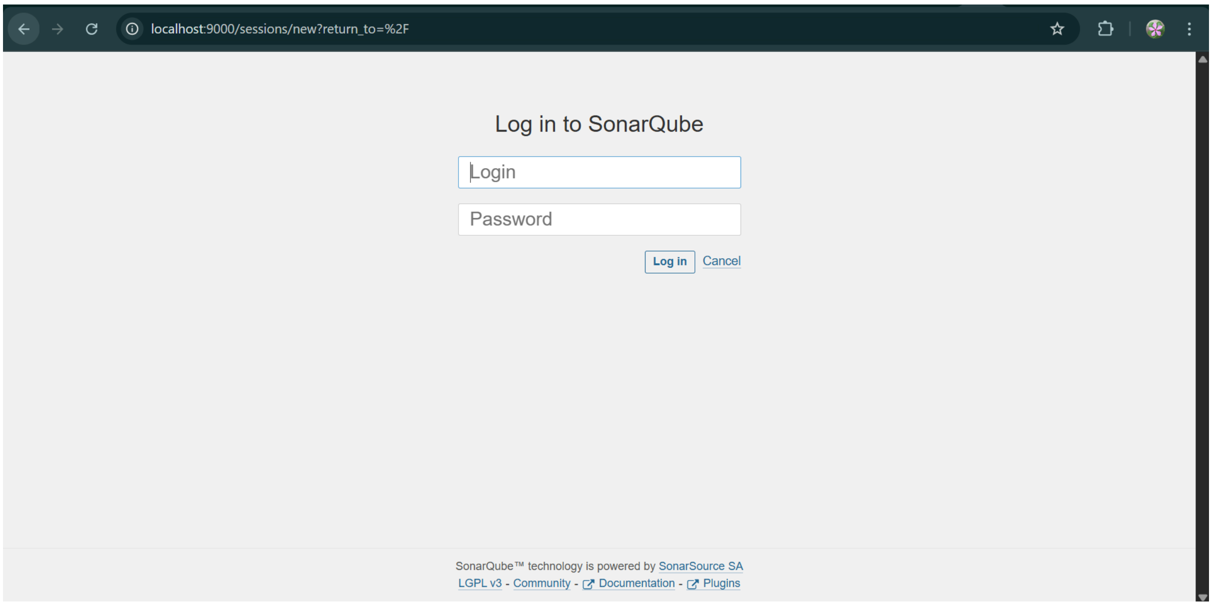
```
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$ kubectl get pods -n sonarqube
NAME                                              READY   STATUS    RESTARTS   AGE
sonarqube-release-postgresql-5769977cf4-rhbz6     1/1     Running   0          3m32s
sonarqube-release-sonarqube-684cd9dfbf-h8kd6      1/1     Running   0          3m32s
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$ kubectl get services -n sonarqube
NAME                                    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
sonarqube-release-postgresql            ClusterIP   10.96.230.103   <none>        5432/TCP   3m54s
sonarqube-release-sonarqube-service     ClusterIP   10.96.139.81    <none>        80/TCP     3m54s
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$ kubectl get ingress -n sonarqube
NAME                                    CLASS   HOSTS                      ADDRESS     PORTS     AGE
sonarqube-release-sonarqube-ingress     nginx   sonarqube.yourdomain.com   localhost   80, 443   4m
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$ kubectl get pvc -n sonarqube
NAME                                         STATUS   VOLUME                                     CAPACITY   ACCESS MODES
   STORAGECLASS   AGE
sonarqube-release-postgresql-pvc             Bound    pvc-a2914aea-dcdb-44c6-8dc8-a98a13661e50   10Gi       RWO
   standard       4m6s
sonarqube-release-sonarqube-data-pvc         Bound    pvc-b38879f2-fc26-42cb-810f-667ca0f01691   10Gi       RWO
   standard       4m6s
sonarqube-release-sonarqube-extensions-pvc   Bound    pvc-8f28f7f1-d67c-45f8-87cd-082b63fca960   5Gi        RWO
   standard       4m6s
sonarqube-release-sonarqube-logs-pvc         Bound    pvc-88c660e8-f632-4071-9fa6-5e61d3b903e2   5Gi        RWO
   standard       4m6s
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$
```

Verify Installation

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

# Check logs
kubectl logs -n sonarqube deployment/sonarqube-release-sonarqube -f

# Check SonarQube status
kubectl port-forward -n sonarqube service/sonarqube-release-sonarqube-service 9000:80
# Then access http://localhost:9000
```



```
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$ kubectl get all -n sonarqube
NAME                                                READY   STATUS    RESTARTS   AGE
pod/sonarqube-release-postgresql-5769977cf4-rhbz6   1/1     Running   0          4m51s
pod/sonarqube-release-sonarqube-684cd9dfbf-h8kd6    1/1     Running   0          4m51s

NAME                                          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
service/sonarqube-release-postgresql          ClusterIP   10.96.230.103   <none>        5432/TCP   4m52s
service/sonarqube-release-sonarqube-service   ClusterIP   10.96.139.81    <none>        80/TCP     4m52s

NAME                                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/sonarqube-release-postgresql   1/1     1            1           4m52s
deployment.apps/sonarqube-release-sonarqube    1/1     1            1           4m52s

NAME                                                      DESIRED   CURRENT   READY   AGE
replicaset.apps/sonarqube-release-postgresql-5769977cf4   1         1         1       4m52s
replicaset.apps/sonarqube-release-sonarqube-684cd9dfbf    1         1         1       4m52s
suhaib@IND-147:~/devops-helm-charts/sonarqube-chart$
```



localhost:9000/sessions/new?return_to=%2F

Log in to SonarQube

Login

Password

SonarQube™ technology is powered by SonarSource SA
LGPL v3 - Community - Documentation - Plugins

Step 6: Management and Maintenance

Upgrade SonarQube

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

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

Rollback if needed

```
# Rollback to previous version
helm rollback sonarqube-release 1 -n sonarqube
```

Uninstall

```
# Uninstall SonarQube
helm uninstall sonarqube-release -n sonarqube
```

```
# Delete namespace (optional)
kubectl delete namespace sonarqube
```

Configuration Options

External Database Configuration

To use an external PostgreSQL database, update `values.yaml`:

```
sonarqube:
  database:
    external: true
    host: "your-postgres-host"
    port: 5432
    name: "sonarqube"
    username: "sonarqube"
    password: "your-secure-password"
```

Resource Customization

Adjust resources based on your requirements:

```
sonarqube:
  resources:
    requests:
      memory: "4Gi"
      cpu: "1000m"
    limits:
      memory: "8Gi"
      cpu: "4000m"
```

JVM Options

Customize JVM settings:

```
sonarqube:
  server:
    jvmOpts: "-Xmx4g -Xms1g -XX:+HeapDumpOnOutOfMemoryError"
```

Troubleshooting

Common Issues and Solutions

1. **Init container fails**: Ensure your cluster allows privileged containers
2. **Database connection issues**: Verify database credentials and connectivity
3. **Memory issues**: Increase resource limits and JVM heap size
4. **Storage issues**: Ensure sufficient disk space and proper StorageClass

Check System Requirements

```
# Verify system requirements are met
kubectl exec -n sonarqube deployment/sonarqube-release-sonarqube -- \
  sh -c "cat /proc/sys/vm/max_map_count && cat /proc/sys/fs/file-max"
```