

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.uc8fntPwVK0YGs7Y3VtZVGshhC6Mz.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

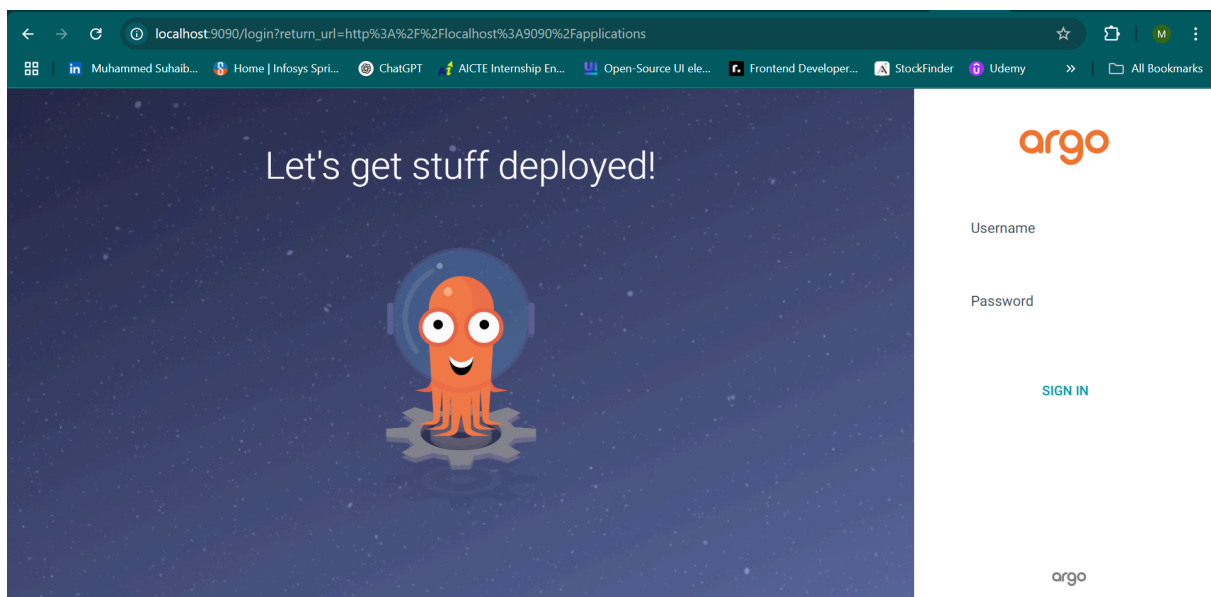
```
kubectl port-forward svc/argocd-release-argocd-server -n argocd 9090:80
```

```
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
```