**Project** [**nginx-argoCD-k8s-demo**](https://github.com/ikanko1989/nginx-argoCD-k8s-demo/tree/main)**:**

**Objectives:**

**Deploying an NGINX web server with custom HTML via a ConfigMap, using Kustomize and Argo CD on the Kubernetes cluster. This includes:**

* **✅ Full GitHub-based GitOps flow**
* **✅ Kustomize folder structure and usage**
* **✅ Argo CD setup and deployment**
* **✅ Testing the app**

**🧱 1. Components**

| **Component** | **Description** |
| --- | --- |
| **NGINX Deployment** | **A simple NGINX pod serving static HTML content** |
| **ConfigMap** | **Contains a custom index.html file** |
| **Kustomize** | **Manages the deployment using modular YAML files** |
| **Argo CD** | **Watches a GitHub repo and syncs the manifests automatically** |
| **GitHub** | **Stores your Kustomize-based Kubernetes manifests** |

**📁2. Directory & File Structure**

**Push structure to GitHub:**

**nginx-argo-k8s-demo/**

**└── nginx-app/**

**├── deployment.yaml**

**├── service.yaml**

**├── configmap.yaml**

**└── kustomization.yaml**

**📄 3. Manifest Files**

**deployment.yaml**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: nginx-deployment**

**spec:**

**replicas: 1**

**selector:**

**matchLabels:**

**app: nginx**

**template:**

**metadata:**

**labels:**

**app: nginx**

**spec:**

**containers:**

**- name: nginx**

**image: nginx:latest**

**ports:**

**- containerPort: 80**

**volumeMounts:**

**- name: html**

**mountPath: /usr/share/nginx/html**

**volumes:**

**- name: html**

**configMap:**

**name: custom-html**

**service.yaml**

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: nginx-service**

**spec:**

**selector:**

**app: nginx**

**ports:**

**- protocol: TCP**

**port: 80**

**targetPort: 80**

**type: NodePort**

**configmap.yaml**

**apiVersion: v1**

**kind: ConfigMap**

**metadata:**

**name: custom-html**

**data:**

**index.html: |**

**<html>**

**<head><title>Welcome to Argo CD + NGINX</title></head>**

**<body>**

**<h1>Hello from ConfigMap!</h1>**

**</body>**

**</html>**

**kustomization.yaml**

**apiVersion: kustomize.config.k8s.io/v1beta1**

**kind: Kustomization**

**resources:**

**- deployment.yaml**

**- service.yaml**

**- configmap.yaml**

**commonLabels:**

**app: nginx**

**🚀 4. Push code to GitHub**

1. **Created a new GitHub repo nginx-argoCD-k8s-demo**
2. **Push code:**

**git init**

**git remote add origin https://github.com/ikanko1989/nginx-argo-k8s-demo.git**

**git add .**

**git commit -m "Initial commit: NGINX with Kustomize"**

**git push -u origin master**

**☸️ 5. Use Kubernetes for ArgoCD deployment**

**a. Create Argo CD namespace:**

**kubectl create namespace argocd**

**c. Install Argo CD:**

**kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml**

**🌐 6. Access Argo CD**

**a. Port-forward Argo CD server:**

**kubectl port-forward svc/argocd-server -n argocd 8080:80**

**b. Get the initial password:**

**kubectl -n argocd get secret argocd-initial-admin-secret \**

**-o jsonpath="{.data.password}" | base64 -d**

**Username: admin  
Password: *(output from above)***

**🔁 7. Create Argo CD Application**

**I used YAML approach instead of UI:**

**nginx-app-argocd.yaml**

**apiVersion: argoproj.io/v1alpha1**

**kind: Application**

**metadata:**

**name: nginx-app**

**namespace: argocd**

**spec:**

**project: default**

**source:**

**repoURL: https://github.com/ikanko1989/nginx-argo-k8s-demo**

**targetRevision: HEAD**

**path: nginx-argo-k8s-demo /nginx-app**

**kustomize: {}**

**destination:**

**server: https://kubernetes.default.svc**

**namespace: default**

**syncPolicy:**

**automated:**

**selfHeal: true**

**prune: true**

**Apply it:**

**kubectl apply -f nginx-app-argocd.yaml**

**✅ 8. Confirm Deployment**

**Check if everything is running:**

**kubectl get pods**

**kubectl get svc**

**Look for nginx-service. Note the NodePort.**

**🌍 9. Access the NGINX App**

**Get the NodePort:**

**kubectl get svc nginx-service**

**Then:**

**curl http://<NODE-IP>:<NODEPORT>**

**Or open in your browser and test**

**Printouts from configuration:**

**student-node ~ ➜ kubectl create ns argocd**

**student-node ~ ➜ kubectl apply -n argocd -f** [**https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml**](https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml)

**customresourcedefinition.apiextensions.k8s.io/applications.argoproj.io created**

**customresourcedefinition.apiextensions.k8s.io/applicationsets.argoproj.io created**

**customresourcedefinition.apiextensions.k8s.io/appprojects.argoproj.io created**

**serviceaccount/argocd-application-controller created**

**serviceaccount/argocd-applicationset-controller created**

**serviceaccount/argocd-dex-server created**

**serviceaccount/argocd-notifications-controller created**

**serviceaccount/argocd-redis created**

**serviceaccount/argocd-repo-server created**

**serviceaccount/argocd-server created**

**role.rbac.authorization.k8s.io/argocd-application-controller created**

**role.rbac.authorization.k8s.io/argocd-applicationset-controller created**

**role.rbac.authorization.k8s.io/argocd-dex-server created**

**role.rbac.authorization.k8s.io/argocd-notifications-controller created**

**role.rbac.authorization.k8s.io/argocd-redis created**

**role.rbac.authorization.k8s.io/argocd-server created**

**clusterrole.rbac.authorization.k8s.io/argocd-application-controller created**

**clusterrole.rbac.authorization.k8s.io/argocd-applicationset-controller created**

**clusterrole.rbac.authorization.k8s.io/argocd-server created**

**rolebinding.rbac.authorization.k8s.io/argocd-application-controller created**

**rolebinding.rbac.authorization.k8s.io/argocd-applicationset-controller created**

**rolebinding.rbac.authorization.k8s.io/argocd-dex-server created**

**rolebinding.rbac.authorization.k8s.io/argocd-notifications-controller created**

**rolebinding.rbac.authorization.k8s.io/argocd-redis created**

**rolebinding.rbac.authorization.k8s.io/argocd-server created**

**clusterrolebinding.rbac.authorization.k8s.io/argocd-application-controller created**

**clusterrolebinding.rbac.authorization.k8s.io/argocd-applicationset-controller created**

**clusterrolebinding.rbac.authorization.k8s.io/argocd-server created**

**configmap/argocd-cm created**

**configmap/argocd-cmd-params-cm created**

**configmap/argocd-gpg-keys-cm created**

**configmap/argocd-notifications-cm created**

**configmap/argocd-rbac-cm created**

**configmap/argocd-ssh-known-hosts-cm created**

**configmap/argocd-tls-certs-cm created**

**secret/argocd-notifications-secret created**

**secret/argocd-secret created**

**service/argocd-applicationset-controller created**

**service/argocd-dex-server created**

**service/argocd-metrics created**

**service/argocd-notifications-controller-metrics created**

**service/argocd-redis created**

**service/argocd-repo-server created**

**service/argocd-server created**

**service/argocd-server-metrics created**

**deployment.apps/argocd-applicationset-controller created**

**deployment.apps/argocd-dex-server created**

**deployment.apps/argocd-notifications-controller created**

**deployment.apps/argocd-redis created**

**deployment.apps/argocd-repo-server created**

**deployment.apps/argocd-server created**

**statefulset.apps/argocd-application-controller created**

**networkpolicy.networking.k8s.io/argocd-application-controller-network-policy created**

**networkpolicy.networking.k8s.io/argocd-applicationset-controller-network-policy created**

**networkpolicy.networking.k8s.io/argocd-dex-server-network-policy created**

**networkpolicy.networking.k8s.io/argocd-notifications-controller-network-policy created**

**networkpolicy.networking.k8s.io/argocd-redis-network-policy created**

**networkpolicy.networking.k8s.io/argocd-repo-server-network-policy created**

**networkpolicy.networking.k8s.io/argocd-server-network-policy created**

**student-node ~ ➜ kubectl port-forward svc/argocd-server -n argocd 8080:80**

**Forwarding from [::1]:8080 -> 8080**

**student-node ~ ➜ curl -k http://localhost:8080**

**<!doctype html><html lang="en"><head><meta charset="UTF-8"><title>Argo CD</title><base href="/"><meta name="viewport" content="width=device-width,initial-scale=1"><link rel="icon" type="image/png" href="assets/favicon/favicon-32x32.png" sizes="32x32"/><link rel="icon" type="image/png" href="assets/favicon/favicon-16x16.png" sizes="16x16"/><link href="assets/fonts.css" rel="stylesheet"><script defer="defer" src="main.5baca1d37bdaf8256580.js"></script></head><body><noscript><p>Your browser does not support JavaScript. Please enable JavaScript to view the site. Alternatively, Argo CD can be used with the <a href="https://argoproj.github.io/argo-cd/cli\_installation/">Argo CD CLI</a>.</p></noscript><div id="app"></div></body><script defer="defer" src="extensions.js"></script></html>**

**student-node ~ ✖ kubectl -n argocd get secret argocd-initial-admin-secret \**

**-o jsonpath="{.data.password}" | base64 -d**

**BJxI5r4EAhazT--l**

**student-node ~ ➜ vim nginx-app-argocd.yaml**

A screenshot of a computer program

AI-generated content may be incorrect.

**student-node ~ ➜ cat nginx-app-argocd.yaml**

**apiVersion: argoproj.io/v1alpha1**

**kind: Application**

**metadata:**

**name: nginx-app**

**namespace: argocd**

**spec:**

**project: default**

**source:**

**repoURL: https://github.com/ikanko1989/nginx-argo-k8s-demo**

**targetRevision: HEAD**

**path: nginx-argo-k8s-demo/nginx-app**

**kustomize: {}**

**destination:**

**server: https://kubernetes.default.svc**

**namespace: default**

**syncPolicy:**

**automated:**

**selfHeal: true**

**prune: true**

**student-node ~ ➜ kubectl apply -f nginx-app-argocd.yaml**

**application.argoproj.io/nginx-app created**

A screen shot of a computer

AI-generated content may be incorrect.

**student-node ~ ✖ kubectl get nodes -o wide**

**NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION CONTAINER-RUNTIME**

**cluster1-node01 Ready <none> 32m v1.30.0+k3s1 192.168.102.149 <none> Alpine Linux v3.16 5.15.0-1083-gcp containerd://1.7.15-k3s1**

**cluster1-controlplane Ready control-plane,master 33m v1.30.0+k3s1 192.168.144.169 <none> Alpine Linux v3.16 5.15.0-1083-gcp containerd://1.7.15-k3s1**

**cluster1-node02 Ready <none> 32m v1.30.0+k3s1 192.168.144.218 <none> Alpine Linux v3.16 5.15.0-1083-gcp containerd://1.7.15-k3s1**

**student-node ~ ➜ curl http://192.168.102.149:32387**

**<html>**

**<head><title>Welcome to Argo CD + NGINX</title></head>**

**<body>**

**<h1>Hello from ConfigMap!</h1>**

**</body>**

**</html>**

**student-node ~ ➜ kubectl get all -A**

**NAMESPACE NAME READY STATUS RESTARTS AGE**

**kube-system pod/coredns-576bfc4dc7-pksp7 1/1 Running 0 35m**

**kube-system pod/local-path-provisioner-75bb9ff978-dfkht 1/1 Running 0 35m**

**kube-system pod/helm-install-traefik-crd-7ddq6 0/1 Completed 0 35m**

**kube-system pod/svclb-traefik-767d3aee-qrf58 2/2 Running 0 35m**

**kube-system pod/helm-install-traefik-mz29g 0/1 Completed 1 35m**

**kube-system pod/traefik-5fb479b77-mx69w 1/1 Running 0 35m**

**kube-system pod/metrics-server-557ff575fb-tcz66 1/1 Running 0 35m**

**kube-system pod/svclb-traefik-767d3aee-m4cdk 2/2 Running 0 35m**

**kube-system pod/svclb-traefik-767d3aee-7d9rd 2/2 Running 0 35m**

**argocd pod/argocd-notifications-controller-56f87565c7-lt2kg 1/1 Running 0 25m**

**argocd pod/argocd-applicationset-controller-8679b4c4b4-6dvt2 1/1 Running 0 25m**

**argocd pod/argocd-redis-86c7487f97-d44bb 1/1 Running 0 25m**

**argocd pod/argocd-dex-server-546bcf4b75-9mmwt 1/1 Running 0 25m**

**argocd pod/argocd-application-controller-0 1/1 Running 0 25m**

**argocd pod/argocd-repo-server-7dc9987cdb-28clr 1/1 Running 0 25m**

**argocd pod/argocd-server-78676c7456-bspgt 1/1 Running 0 25m**

**default pod/nginx-deployment-5f9f5cfd55-llbtp 1/1 Running 0 6m23s**

**NAMESPACE NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE**

**default service/kubernetes ClusterIP 10.43.0.1 <none> 443/TCP 35m**

**kube-system service/kube-dns ClusterIP 10.43.0.10 <none> 53/UDP,53/TCP,9153/TCP 35m**

**kube-system service/metrics-server ClusterIP 10.43.20.175 <none> 443/TCP 35m**

**kube-system service/traefik LoadBalancer 10.43.91.51 192.168.102.149,192.168.144.169,192.168.144.218 80:32682/TCP,443:32026/TCP 35m**

**argocd service/argocd-applicationset-controller ClusterIP 10.43.58.92 <none> 7000/TCP,8080/TCP 25m**

**argocd service/argocd-dex-server ClusterIP 10.43.32.77 <none> 5556/TCP,5557/TCP,5558/TCP 25m**

**argocd service/argocd-metrics ClusterIP 10.43.65.26 <none> 8082/TCP 25m**

**argocd service/argocd-notifications-controller-metrics ClusterIP 10.43.76.130 <none> 9001/TCP 25m**

**argocd service/argocd-redis ClusterIP 10.43.124.115 <none> 6379/TCP 25m**

**argocd service/argocd-repo-server ClusterIP 10.43.132.122 <none> 8081/TCP,8084/TCP 25m**

**argocd service/argocd-server ClusterIP 10.43.206.212 <none> 80/TCP,443/TCP 25m**

**argocd service/argocd-server-metrics ClusterIP 10.43.72.128 <none> 8083/TCP 25m**

**default service/nginx-service NodePort 10.43.106.141 <none> 80:32387/TCP 6m23s**

**NAMESPACE NAME DESIRED CURRENT READY UP-TO-DATE AVAILABLE NODE SELECTOR AGE**

**kube-system daemonset.apps/svclb-traefik-767d3aee 3 3 3 3 3 <none> 35m**

**NAMESPACE NAME READY UP-TO-DATE AVAILABLE AGE**

**kube-system deployment.apps/coredns 1/1 1 1 35m**

**kube-system deployment.apps/local-path-provisioner 1/1 1 1 35m**

**kube-system deployment.apps/traefik 1/1 1 1 35m**

**kube-system deployment.apps/metrics-server 1/1 1 1 35m**

**argocd deployment.apps/argocd-notifications-controller 1/1 1 1 25m**

**argocd deployment.apps/argocd-applicationset-controller 1/1 1 1 25m**

**argocd deployment.apps/argocd-redis 1/1 1 1 25m**

**argocd deployment.apps/argocd-dex-server 1/1 1 1 25m**

**argocd deployment.apps/argocd-repo-server 1/1 1 1 25m**

**argocd deployment.apps/argocd-server 1/1 1 1 25m**

**default deployment.apps/nginx-deployment 1/1 1 1 6m23s**

**NAMESPACE NAME DESIRED CURRENT READY AGE**

**kube-system replicaset.apps/coredns-576bfc4dc7 1 1 1 35m**

**kube-system replicaset.apps/local-path-provisioner-75bb9ff978 1 1 1 35m**

**kube-system replicaset.apps/traefik-5fb479b77 1 1 1 35m**

**kube-system replicaset.apps/metrics-server-557ff575fb 1 1 1 35m**

**argocd replicaset.apps/argocd-notifications-controller-56f87565c7 1 1 1 25m**

**argocd replicaset.apps/argocd-applicationset-controller-8679b4c4b4 1 1 1 25m**

**argocd replicaset.apps/argocd-redis-86c7487f97 1 1 1 25m**

**argocd replicaset.apps/argocd-dex-server-546bcf4b75 1 1 1 25m**

**argocd replicaset.apps/argocd-repo-server-7dc9987cdb 1 1 1 25m**

**argocd replicaset.apps/argocd-server-78676c7456 1 1 1 25m**

**default replicaset.apps/nginx-deployment-5f9f5cfd55 1 1 1 6m23s**

**NAMESPACE NAME READY AGE**

**argocd statefulset.apps/argocd-application-controller 1/1 25m**

**NAMESPACE NAME STATUS COMPLETIONS DURATION AGE**

**kube-system job.batch/helm-install-traefik-crd Complete 1/1 16s 35m**

**kube-system job.batch/helm-install-traefik Complete 1/1 20s 35m**

**student-node ~ ➜ kubectl get cm**

**NAME DATA AGE**

**kube-root-ca.crt 1 39m**

**custom-html 1 10m**

**student-node ~ ➜ kubectl describe cm custom-html**

**Name: custom-html**

**Namespace: default**

**Labels: app=nginx**

**Annotations: argocd.argoproj.io/tracking-id: nginx-app:/ConfigMap:default/custom-html**

**Data**

**====**

**index.html:**

**----**

**<html>**

**<head><title>Welcome to Argo CD + NGINX</title></head>**

**<body>**

**<h1>Hello from ConfigMap!</h1>**

**</body>**

**</html>**

**BinaryData**

**====**

**Events: <none>**

A screenshot of a computer program

AI-generated content may be incorrect.

**student-node ~ ➜ kubectl get pods -o wide**

**NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES**

**nginx-deployment-5f9f5cfd55-llbtp 1/1 Running 0 11m 10.42.1.6 cluster1-node01 <none> <none>**

**student-node ~ ➜ kubectl describe pod nginx-deployment-5f9f5cfd55-llbtp**

**Name: nginx-deployment-5f9f5cfd55-llbtp**

**Namespace: default**

**Priority: 0**

**Service Account: default**

**Node: cluster1-node01/192.168.102.149**

**Start Time: Sun, 17 Aug 2025 10:53:00 +0000**

**Labels: app=nginx**

**pod-template-hash=5f9f5cfd55**

**Annotations: <none>**

**Status: Running**

**IP: 10.42.1.6**

**IPs:**

**IP: 10.42.1.6**

**Controlled By: ReplicaSet/nginx-deployment-5f9f5cfd55**

**Containers:**

**nginx:**

**Container ID: containerd://45738b9ed1f498b1b47411632ae15042623f3c26d2989dc6021d4075a63e0e21**

**Image: nginx:latest**

**Image ID: docker.io/library/nginx@sha256:33e0bbc7ca9ecf108140af6288c7c9d1ecc77548cbfd3952fd8466a75edefe57**

**Port: 80/TCP**

**Host Port: 0/TCP**

**State: Running**

**Started: Sun, 17 Aug 2025 10:53:06 +0000**

**Ready: True**

**Restart Count: 0**

**Environment: <none>**

**Mounts:**

**/usr/share/nginx/html from html (rw)**

**/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-l4lfl (ro)**

**Conditions:**

**Type Status**

**PodReadyToStartContainers True**

**Initialized True**

**Ready True**

**ContainersReady True**

**PodScheduled True**

**Volumes:**

**html:**

**Type: ConfigMap (a volume populated by a ConfigMap)**

**Name: custom-html**

**Optional: false**

**kube-api-access-l4lfl:**

**Type: Projected (a volume that contains injected data from multiple sources)**

**TokenExpirationSeconds: 3607**

**ConfigMapName: kube-root-ca.crt**

**ConfigMapOptional: <nil>**

**DownwardAPI: true**

**QoS Class: BestEffort**

**Node-Selectors: <none>**

**Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s**

**node.kubernetes.io/unreachable:NoExecute op=Exists for 300s**

**Events:**

**Type Reason Age From Message**

**---- ------ ---- ---- -------**

**Normal Scheduled 11m default-scheduler Successfully assigned default/nginx-deployment-5f9f5cfd55-llbtp to cluster1-node01**

**Normal Pulling 11m kubelet Pulling image "nginx:latest"**

**Normal Pulled 11m kubelet Successfully pulled image "nginx:latest" in 5.491s (5.491s including waiting). Image size: 72324501 bytes.**

**Normal Created 11m kubelet Created container nginx**

**Normal Started 11m kubelet Started container nginx**

**student-node ~ ➜ kubectl describe pod nginx-deployment-5f9f5cfd55-llbtp**

**Name: nginx-deployment-5f9f5cfd55-llbtp**

**Namespace: default**

**Priority: 0**

**Service Account: default**

**Node: cluster1-node01/192.168.102.149**

**Start Time: Sun, 17 Aug 2025 10:53:00 +0000**

**Labels: app=nginx**

**pod-template-hash=5f9f5cfd55**

**Annotations: <none>**

**Status: Running**

**IP: 10.42.1.6**

**IPs:**

**IP: 10.42.1.6**

**Controlled By: ReplicaSet/nginx-deployment-5f9f5cfd55**

**Containers:**

**nginx:**

**Container ID: containerd://45738b9ed1f498b1b47411632ae15042623f3c26d2989dc6021d4075a63e0e21**

**Image: nginx:latest**

**Image ID: docker.io/library/nginx@sha256:33e0bbc7ca9ecf108140af6288c7c9d1ecc77548cbfd3952fd8466a75edefe57**

**Port: 80/TCP**

**Host Port: 0/TCP**

**State: Running**

**Started: Sun, 17 Aug 2025 10:53:06 +0000**

**Ready: True**

**Restart Count: 0**

**Environment: <none>**

**Mounts:**

**/usr/share/nginx/html from html (rw)**

**/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-l4lfl (ro)**

**Conditions:**

**Type Status**

**PodReadyToStartContainers True**

**Initialized True**

**Ready True**

**ContainersReady True**

**PodScheduled True**

**Volumes:**

**html:**

**Type: ConfigMap (a volume populated by a ConfigMap)**

**Name: custom-html**

**Optional: false**

**kube-api-access-l4lfl:**

**Type: Projected (a volume that contains injected data from multiple sources)**

**TokenExpirationSeconds: 3607**

**ConfigMapName: kube-root-ca.crt**

**ConfigMapOptional: <nil>**

**DownwardAPI: true**

**QoS Class: BestEffort**

**Node-Selectors: <none>**

**Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s**

**node.kubernetes.io/unreachable:NoExecute op=Exists for 300s**

**Events:**

**Type Reason Age From Message**

**---- ------ ---- ---- -------**

**Normal Scheduled 11m default-scheduler Successfully assigned default/nginx-deployment-5f9f5cfd55-llbtp to cluster1-node01**

**Normal Pulling 11m kubelet Pulling image "nginx:latest"**

**Normal Pulled 11m kubelet Successfully pulled image "nginx:latest" in 5.491s (5.491s including waiting). Image size: 72324501 bytes.**

**Normal Created 11m kubelet Created container nginx**

**Normal Started 11m kubelet Started container nginx**