What is Argo CD?

Argo CD (short for Argo Continuous Delivery) is a declarative, GitOps-based continuous delivery tool for Kubernetes. It helps manage application deployment and lifecycle using Git repositories as the source of truth.

Argo CD is a Kubernetes-native tool that:

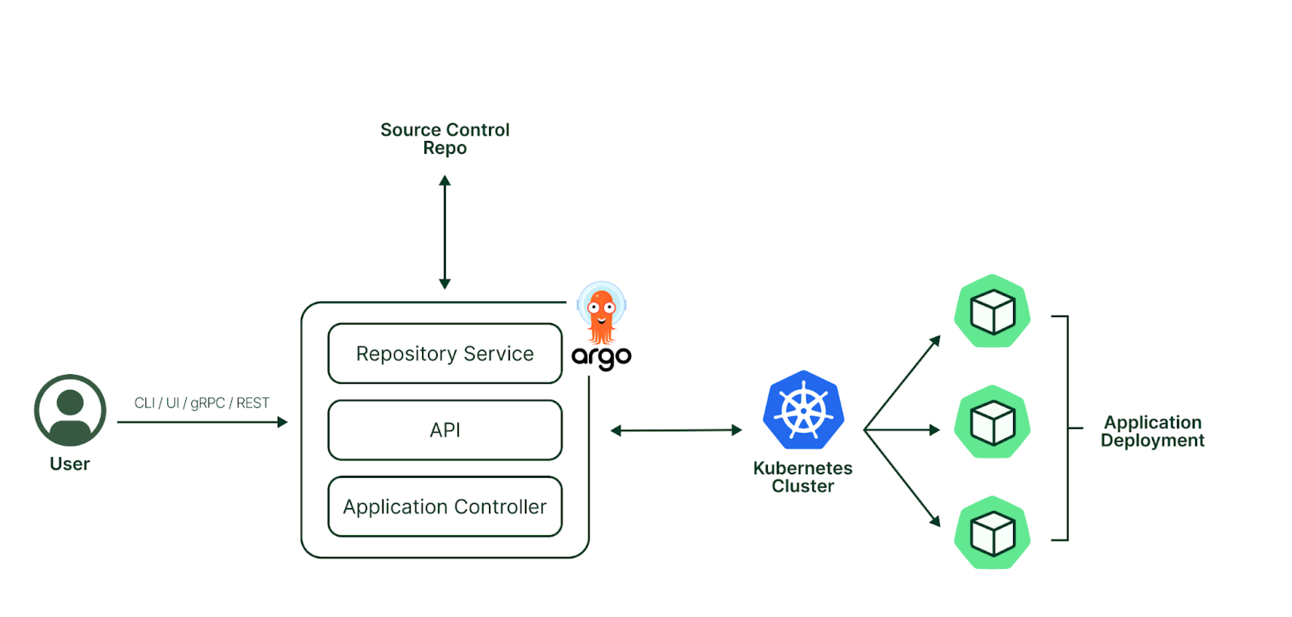
* Automates the deployment of applications to Kubernetes.
* Ensures the live state of your cluster matches the declared state in Git.
* Supports multi-cluster management.
* Has a web UI, CLI, and API for visibility and control.

Why use Agro CD?

1. GitOps Workflow
2. Visibility and Auditing
3. Safe, Automated Deployments
4. Multi-Cluster Support

How Does Argo CD Work?

1. Set Up a Git Repository
2. Register the Git Repository in Argo CD
3. Define Applications
4. Sync Mechanism
5. UI and CLI



#### 1. ****User****

* Interacts with Argo CD via:
  + **CLI**
  + **Web UI**
  + **gRPC or REST API**

#### 2. ****Source Control Repo (Git)****

* Stores the **desired state** of your applications (Kubernetes manifests, Helm charts, Kustomize files, etc.).
* Argo CD continuously monitors this repo for changes.

#### 3. ****Argo CD Core Components****

* **Repository Service**:
  + Connects to Git.
  + Pulls application manifests from the repo.
* **API**:
  + Provides communication endpoints for CLI, UI, and integrations.
* **Application Controller**:
  + Compares the desired state (in Git) with the live state (in Kubernetes).
  + Syncs the cluster if they differ.

#### 4. ****Kubernetes Cluster****

* The environment where your applications are deployed.
* Argo CD ensures this environment matches what’s declared in Git.

#### 5. ****Application Deployment****

* Actual pods, services, etc., deployed in Kubernetes as per the Git definition.
* If something changes manually in the cluster, Argo CD detects the drift and can auto-correct it.

How to set up Argo CD

# Firstly install kubectl:

* Install kubectl binary with curl on Linux

curl -LO [https://dl.k8s.io/release/**$(**curl -L -s https://dl.k8s.io/release/stable.txt**)**/bin/linux/amd64/kubectl](https://dl.k8s.io/release/$(curl%20-L%20-s%20https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl)

* Validate the binary (optional)

Download the kubectl checksum file:

curl -LO [https://dl.k8s.io/release/**$(**curl -L -s https://dl.k8s.io/release/stable.txt**)**/bin/linux/amd64/kubectl.sha256](https://dl.k8s.io/release/$(curl%20-L%20-s%20https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256)

* Validate the kubectl binary against the checksum file:

echo "**$(**cat kubectl.sha256**)**kubectl" | sha256sum –check

* Install kubectl

sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

* Test to ensure the version you installed is up-to-date:

kubectl version --client

* If kubectl version is not visible execute following commands (its optional)

chmod +x kubectl

mkdir -p ~/.local/bin

mv ./kubectl ~/.local/bin/kubectl

kubectl version --client

# Install Argo CD

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

# Download Argo CD CLI

* curl -sSL -o argocd-linux-amd64 <https://github.com/argoproj/argo-cd/releases/latest/download/argocd-linux-amd64>
* sudo install -m 555 argocd-linux-amd64 /usr/local/bin/argocd
* rm argocd-linux-amd64

# Access The Argo CD API Server

* kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "LoadBalancer"}}'
* kubectl get svc argocd-server -n argocd -o=jsonpath='{.status.loadBalancer.ingress[0].ip}'

# Login Using The CLI

* argocd admin initial-password -n argocd

Using the username admin and the password by using above command, login to Argo CD's IP or hostname

* argocd login <ARGOCD\_SERVER>

Change the password using the command:

* argocd account update-password