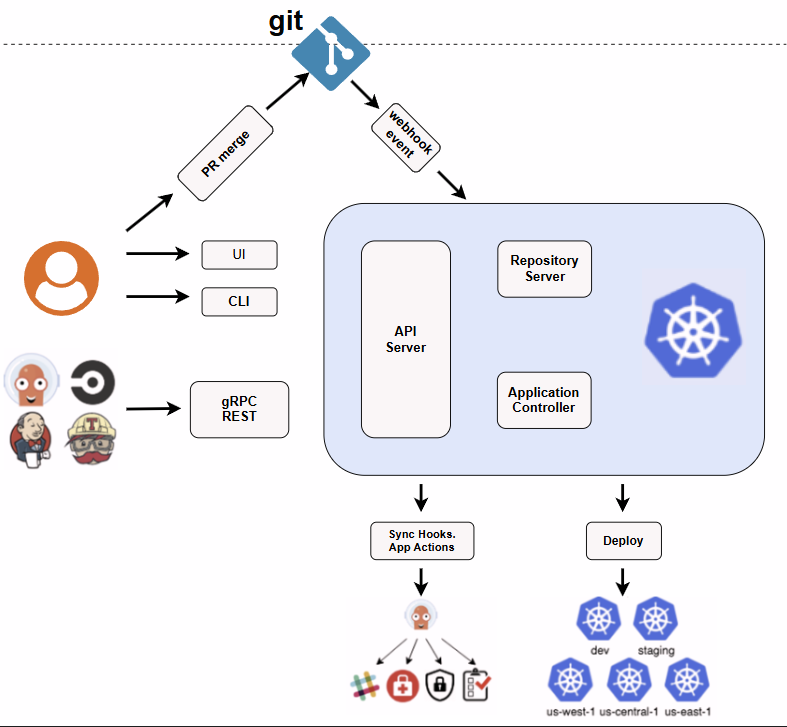
**Argo CD**

**Argo CD:-**

https://argo-cd.readthedocs.io/en/stable/getting\_started/  
Argo CD (Argo Continuous Delivery) is a **declarative, GitOps-based continuous delivery tool for Kubernetes**.  
It follows the GitOps principle: **Git is the source of truth for application definitions**.  
Argo CD ensures that the **running state of applications in Kubernetes matches the desired state stored in Git**.  
  
**Architecture:-**  


**Key Features:-**

* Declarative GitOps approach (Git is the single source of truth).
* Automated deployment to Kubernetes clusters.
* Continuous synchronization (keeps cluster in sync with Git).
* Web UI, CLI, and API support.
* Role-based access control (RBAC) and SSO integration.

**ArgoCD Workflow:-**

* Developer pushes code to Git → CI pipeline builds artifact (Docker image).
* CI pipeline updates manifest in Git (e.g., image tag).
* ArgoCD detects Git changes.
* ArgoCD syncs manifests → deploys to Kubernetes cluster.
* Cluster state always matches Git.

**Getting Started:-**

**Prerequisites:-**

Before you begin, make sure you have:

* **kubectl** installed and configured to connect to your Kubernetes cluster.
* A valid **kubeconfig** file (typically located at ~/.kube/config).
* **CoreDNS** enabled if you're using microk8s (run microk8s enable dns && microk8s stop && microk8s start)

**1. Install Argo CD:-**

kubectl create namespace argocd

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

* Argo CD runs in its own namespace (argocd).
* Default installation uses **self-signed TLS certificates**.
* For testing, you can use --insecure flag in the CLI.
* You can simplify future commands by setting the namespace as default:

kubectl config set-context --current --namespace=argocd

**2. Download the Argo CD CLI:-**You’ll need the argocd command-line tool to interact with your Argo CD instance.  
The **CLI (Command-Line Interface)** is a client tool (argocd) used to interact with the Argo CD API server (login, manage apps, sync, etc.).  
  
Download the latest Argo CD version from <https://github.com/argoproj/argo-cd/releases/latest>. More detailed installation instructions can be found via the [CLI installation documentation](https://argo-cd.readthedocs.io/en/stable/cli_installation/).  
  
Also available in Mac, Linux and WSL Homebrew:  
brew install argocd

**3. Access the Argo CD API Server:-**

The API server is the **entry point** for all Argo CD operations. It exposes endpoints for CLI, UI, and API requests.

**Methods of Access:-**

1. **LoadBalancer (Cloud environments)**

kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "LoadBalancer"}}'

Then retrieve external IP:

kubectl get svc argocd-server -n argocd

**2. Ingress (Production setup)**

Configure with your ingress controller (NGINX, Traefik, etc.).

**3. Port Forwarding (Local testing)**

kubectl port-forward svc/argocd-server -n argocd 8080:443

Access via → https://localhost:8080

**Notes:-**

* For **local development**, port forwarding is easiest.
* For **production**, prefer LoadBalancer or Ingress with TLS.

**4. Login with the CLI:-**

Authentication is required before using Argo CD. The default admin user is created during installation.

**Procedure:-**

1. **Get the initial admin password:**

argocd admin initial-password -n argocd

(Reads from secret: argocd-initial-admin-secret)

1. **Login:**

argocd login <ARGOCD\_SERVER>

Example (with port forwarding):

argocd login localhost:8080

1. **Change the password:**

argocd account update-password

1. **(Recommended) Delete the initial password secret:**

kubectl delete secret argocd-initial-admin-secret -n argocd

**Notes:-**

* With self-signed certs, add --insecure during login.
* Always change the default password for security.