Pre-flight Checks

```
docker info >/dev/null 2>&1 && echo "Docker OK" || echo "Docker NOT running"
ls p3/confs/argocd-app.yaml && echo "Repo root OK"
docker pull wil42/playground:v1
docker pull wil42/playground:v2
```

Start Up

```
bash p3/scripts/install_k3d_argocd.sh
```

Configuration Checks (Evalsheet)

```
ls p3/confs/        # → argocd-app.yaml
ls p3/dev-app/      # → deployment.yaml  service.yaml
ls p3/scripts/      # → install_k3d_argocd.sh

cat p3/confs/argocd-app.yaml
kubectl get ns      # → argocd Active | dev Active
kubectl get pods -n dev     # → wil-playground-xxxx  1/1  Running
kubectl get pods -n argocd  # → 7 pods Running

# ArgoCD UI
kubectl -n argocd port-forward svc/argocd-server 8080:443 &
kubectl -n argocd get secret argocd-initial-admin-secret \
  -o jsonpath='{.data.password}' | base64 -d

# Verify app config
kubectl get application dev-app -n argocd -o yaml | \
  grep -E "repoURL|path:|namespace:|prune|selfHeal"
```

GitOps Flow Demo (Critical)

```
# Terminal 1 — watch live
watch kubectl get pods -n dev

# Terminal 2 — trigger update
sed -i 's/wil42\/playground:v1/wil42\/playground:v2/' p3/dev-app/deployment.yaml
git add p3/dev-app/deployment.yaml
git commit -m "upgrade to v2"
git push

# Force sync if needed after ~30s
argocd app sync dev-app

# Verify
kubectl get application dev-app -n argocd
kubectl get deployment -n dev -o jsonpath='{.items[^0].spec.template.spec.containers[^0].image}'
# → wil42/playground:v2
```

Rollback (Impressive bonus)

```
sed -i 's/wil42\/playground:v2/wil42\/playground:v1/' p3/dev-app/deployment.yaml
git add p3/dev-app/deployment.yaml
git commit -m "rollback to v1"
git push
```

Key Q&A for Evaluator

Question	Answer
What is K3d?	K3s inside Docker containers — same Kubernetes API, no VM, starts in seconds
What is Argo CD?	GitOps CD tool — Git is source of truth, polls repo, auto-reconciles cluster
Why does a git push update the pod?	ArgoCD detects image tag change, applies new deployment.yaml, Kubernetes pulls v2 from Docker Hub
Namespace vs Pod?	Namespace = logical isolation boundary (folder); Pod = smallest runnable unit wrapping containers