P3 Full Defense Command Sheet

```
# ── Pre-flight: host machine (BEFORE running script) ──────────────────
# Confirm Docker is running
docker info >/dev/null 2>&1 && echo "Docker OK" || echo "Docker NOT running"

# Confirm you are in repo root
ls p3/confs/argocd-app.yaml && echo "Repo root OK"

# Confirm wil42/playground tags exist on Docker Hub
docker pull wil42/playground:v1
docker pull wil42/playground:v2

# ── Start up ──────────────────────────────────────────────────────────────
bash p3/scripts/install_k3d_argocd.sh

# ── Evalsheet: Configuration Checks ──────────────────────────────────────

# Show all p3 files (evaluator will ask to explain each)
ls p3/confs/       # → argocd-app.yaml
ls p3/dev-app/     # → deployment.yaml  service.yaml
ls p3/scripts/     # → install_k3d_argocd.sh

# Show argocd-app.yaml content
cat p3/confs/argocd-app.yaml
# → repoURL: https://github.com/usrali2026/Inception_of_Things.git
# → path: p3/dev-app
# → namespace: dev
# → automated: prune + selfHeal

# Both namespaces exist
kubectl get ns
# → argocd   Active
# → dev      Active

# At least 1 pod in dev namespace
kubectl get pods -n dev
# → wil-playground-xxxx   1/1   Running

# All 7 ArgoCD pods Running
kubectl get pods -n argocd
# → argocd-application-controller-0        1/1   Running
# → argocd-applicationset-controller-xxx   1/1   Running
# → argocd-dex-server-xxx                  1/1   Running
# → argocd-notifications-controller-xxx    1/1   Running
# → argocd-redis-xxx                       1/1   Running
# → argocd-repo-server-xxx                 1/1   Running
# → argocd-server-xxx                      1/1   Running

# ArgoCD UI access
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
# → open https://localhost:8080  |  admin / <password>
# GitHub repo name includes login
cat p3/confs/argocd-app.yaml | grep repoURL
# → usrali2026/Inception_of_Things

# Argo CD Application correctly configured
kubectl get application dev-app -n argocd -o yaml | \
  grep -E "repoURL|path:|namespace:|prune|selfHeal"
# → repoURL: https://github.com/usrali2026/Inception_of_Things.git
# → path: p3/dev-app
# → namespace: dev
# → prune: true
# → selfHeal: true

# ── Evalsheet: Usage Checks — GitOps Flow ──────────────────────────────

# Application status
kubectl get application -n argocd
# → NAME      SYNC STATUS   HEALTH STATUS
# → dev-app   Synced        Healthy

# Verify v1 is running
kubectl get deployment -n dev \
  -o jsonpath='{.items[^0].spec.template.spec.containers[^0].image}'
# → wil42/playground:v1

# Verify Docker Hub is used (show image source)
kubectl get pods -n dev -o yaml | grep imageID
# → imageID: docker.io/wil42/playground@sha256:...

# ── Live v1 → v2 demo (the critical step) ──────────────────────────────

# Terminal 1 — watch pods roll in real time
watch kubectl get pods -n dev

# Terminal 2 — perform the 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

# If auto-sync doesn't trigger in ~30s, force it:
argocd app sync dev-app

# Verify sync complete
kubectl get application dev-app -n argocd
# → dev-app   Synced   Healthy

# Confirm v2 running (both commands — evalsheet checks both)
kubectl get deployment -n dev \
  -o jsonpath='{.items[^0].spec.template.spec.containers[^0].image}'
# → wil42/playground:v2

kubectl get pod -n dev \
  -o jsonpath='{.items[^0].spec.containers[^0].image}'
# → wil42/playground:v2

# ── Optional: Rollback to v1 (impressive — do it) ──────────────────────
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
# → Argo CD syncs back → pod returns to v1

# ── Cleanup after defense (optional) ──────────────────────────────────────
k3d cluster delete iot-p3
```

Key Explanations to Have Ready

Question	Answer
"*What is K3d?*"	"K3s running inside Docker containers — same Kubernetes API, no VM needed, starts in seconds"
"*What is Argo CD?*"	"A GitOps CD tool — Git is the source of truth. It polls the repo, detects drift, and auto-reconciles the cluster to match"
"*What is continuous integration?*"	"Automatically building, testing, and integrating code changes — here Argo CD extends that to automatically deploying changes to the cluster"
"*Why does pushing to GitHub update the pod?*"	"Argo CD watches the repo, detects the image tag changed from v1 to v2, applies the new deployment.yaml, Kubernetes pulls v2 from Docker Hub, rolls the pod"
"*What is the difference between namespace and pod?*"	"Namespace is a logical isolation boundary — like a folder. A pod is the smallest runnable unit — wraps one or more containers"