ARGO CD
ANTI-PATTERNS

GitOps with a UI... until everything breaks in sync

Argo CD makes GitOps easy to adopt, and surprisingly easy to misuse.

Here are some anti-patterns I've seen in production Argo CD setups, and how to avoid them.

#1:
Mixing CI with CD

Running tests and builds in Argo?

That's CI logic in your CD tool.

Let CI build and push.

Argo should only sync from Git.

#2:
Sync Waves for Service Control

Using sync waves to control service startup? You're encoding app logic into deploy order.

Use readiness probes, not sync order. Decouple apps from orchestration.

#3:
GitOps as a Gatekeeper

Blocking changes until they pass "the GitOps process"? That's policy theater, not platform strategy.

Make Argo a guardrail, not a bottleneck. Collaborate with dev teams early.

#4:
Manual Overrides via UI

Clicking "Sync" on broken apps?

That's not GitOps... that's firefighting.

Fix configs in repository.

Keep drift and sync failures visible

and traceable.

#5:
Huge Application Sets

One AppSet deploying 200+ services? That's not scale, that's a single point of failure.

Split into domains. Use folders or clusters to group by ownership.

#6:
No Drift Alerts

Drift detected, but no one sees it? Broken deploys can go unnoticed.

Integrate Argo with alerts and dashboards. Treat sync status as a production signal.

#7:
Overloaded Helm Values

Hundreds of values injected into Helm charts? You've built a config mess inside a deploy tool.

Standardize required values.
Avoid snowflake chart overrides.

#8:
Using Argo CD for Secrets

Storing secrets (even encrypted) in repositories? Argo isn't a vault.

Use External Secrets, SOPS, or sealed-secrets. Keep them separate from manifests.

#9:
Ignoring Health Checks

Just because it shows "Synced"
doesn't mean it's "healthy".

Define proper health checks. Monitor
both sync and app status.

#10:
Confusing Argo CD with GitOps

Using Argo CD ≠ practicing GitOps well.

GitOps is a workflow and mindset.
Argo CD is just one tool. Design your platform intentionally.

Argo CD can bring powerful automation, or fragile complexity. Which anti-patterns have you run into?

If this helped, save & share with your team!