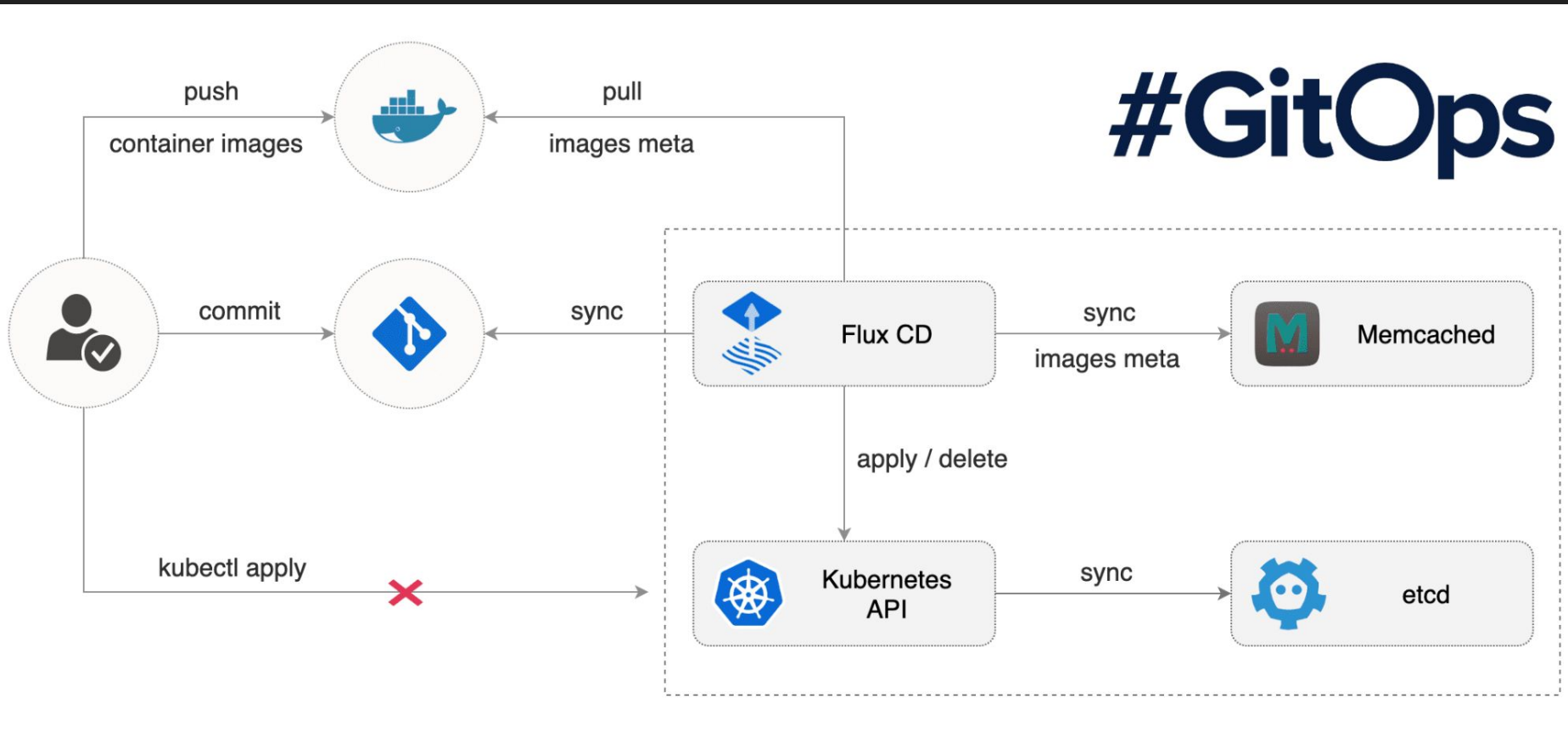
GitOps

What is it and why do we want it?

Definition

GitOps upholds the principle that Git is the one and only source of truth. GitOps requires the desired state of the system to be stored in version control such that anyone can view the entire audit trail of changes. All changes to the desired state are fully traceable commits associated with committer information, commit IDs and time stamps. This means that both the application and the infrastructure are now versioned artifacts and can be audited using the gold standards of software development and delivery

Cloudbees, 2020


#GitOps
push
container images
pull
images meta
commit
sync
Flux CD
sync
images meta
Memcached
apply / delete
kubectl apply
Kubernetes API
sync
etcd

Advantages

- Declarative definition of desired state for every environment (like terraform infrastructure as code)
- All deployment definition is versioned and traceable in Git
- No manual changes in the cluster
- Pull requests for changes with people responsible for approving or denying
- Complete history of changes is auditable
- Alert if the desired state diverges from the observed state, also success messages are possible
- Helps to achieve more and faster deployments

Advantages part 2

- In case of a problem the commit that was responsible is easy to find (compare to manual changes on servers...)
- Rollback by reverting commits
- Diff can communicate changes to anyone interested (e.g. what changed between now and a month before)
- The real state is completely documented in Git

Most used GitOps tools for Kubernetes

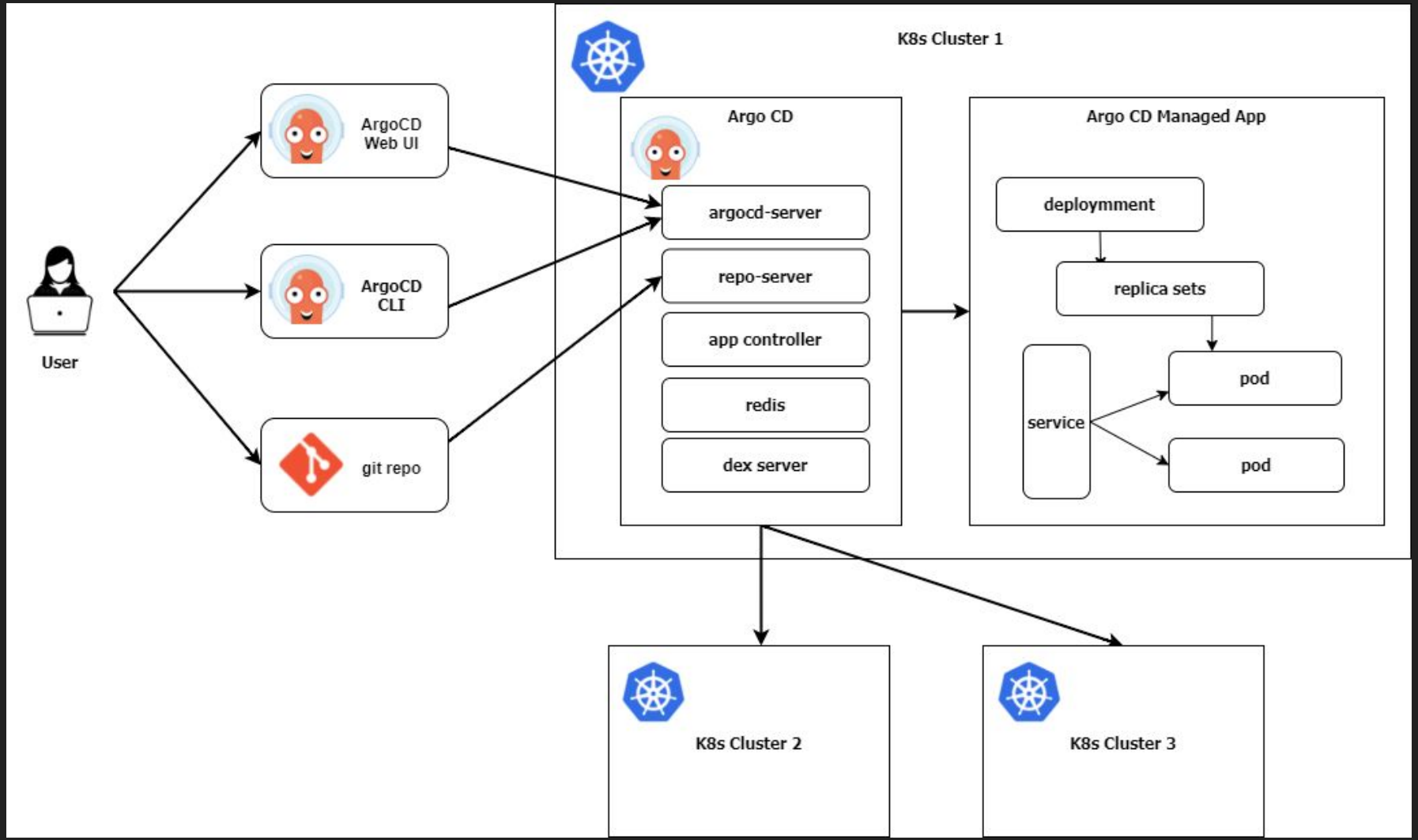
- ArgoCD
- JenkinsX
- Flux

Why ArgoCD

- Better to show due to UI

- I know it better

- Both tools are great and have their advantages and disadvantages


K8s Cluster 1
Argo CD
argocd-server
repo-server
app controller
redis
dex server
Argo CD Managed App
deploymment
replica sets
service
pod
pod
ArgoCD Web UI
ArgoCD CLI
git repo
User
K8s Cluster 2
K8s Cluster 3

Where does Git and ArgoCD run?

ArgoCD:

- Central ArgoCD for all deployments
- One instance per target cluster running in the cluster
- Even one instance per namespace possible

Git repositories:

- Can be spread as necessary
- You can link as many repos (including private repos) as you want

Supported technologies

- Plain resources
- kustomize
- helm
- Ksonnet
- Jsonnet
- Bring your own

ArgoCD CRDs (Custom resources)

- Project
- Application
- (Application Set)

Demo time!

Notes 1

- Gcp gke cluster
- Ingress-nginx
- hostNetwork to save money
- ArgoCD installed with ingress

Steps

- https://github.com/sadoMasupilami/argo-test.git ← plain resources
- Self heal

- https://charts.bitnami.com/bitnami
- Changes from versions