Introduction to GitOps

- Introduction
  -
- GitOps
  -
  -
- Demonstrations
  -
  -
  -
  -
  -
  -
  -
  -
- Required knowledge
  -
  -
  -
  -

- Core concepts: A quick introduction
  -
  -
  -
  -
  -
- What GitOps is not
  -
  -
- The use case: Deploying a highly available microservice
  -
- Implementing GitOps for Kubernetes in AWS
  -
- Logging in to the Amazon Web Services Console
  -
  -
  -
  -
  -

Part 2: Setting up the Tools

- Kubernetes
  -

- Preparation: Setting up kubectl
  -

- Preparation: Access a cluster through kubectl/k9s.
  -
  -

- Repository

- Preparation: Infrastructure repository
  -

- Application repository
  -
  -

- ArgoCD

- Why ArgoCD?
  -

- Exercise: Add ArgoCD to the cluster.
  -
  -
  -

Part 3: Deploying a Microservice

- Exercise: Prepare a simple microservice to be deployed in k8s.
  - ○

  - ○

- Exercise: Create a k8s deployment.
  - ○

  - ○

  - ○

  - ○

  - ○

  - ○

  - ○

- Automated synchronization
  - ○

  - ○

- Exercise: Create a k8s service. (A deployment alone doesn't expose the microservice, so let's build on that.)
  - ○

- Wrap-up: This covers the workflow of deploying an application and then performing updates and changes on it.
  - ○

  - ○

- Pull request
  - ○

  - ○

  - ○

- Exercise: Create a load balancer. (You still can't access service from the outside.)
  - ○

  - ○

  - ○

- Exercise: Update the application.
  - ○

  - ○

  - ○

  - ○

  - ○

- Exercise: Update the infrastructure. (Why do this? So you can demonstrate that changing the application and the infrastructure results in blurry boundaries.)
  - ○

  - ○

  - ○

Part 4: Promoting Changes Through Different Environments

- In advance: Prepare a second cluster.
  - ○

- Preparation: Register the cluster in ArgoCD to allow deployments to it.
- From development to production
  - ○
  - ○

  - ○
  - ○
  - ○

- Exercise: Promotion of a version
  - ○
  - ○
  - ○
  - ○

- Exercise: Sealed secrets
  - ○

  - ○

  - ○
  - ○
  - ○
  - ○
  - ○

- Canary release
  - ○

- Exercise: Error handling (This exercise shows that failure in infra deployment is expected and is handled through code changes — not panicked actions!)
  - ○

  - ○
  - ○
  - ○
  - ○
  - ○

- Accessing resources
  - ○

- Secrets
  - ○

- Vault
  - ○

- Exercise: Blue/Green
  - ○
  - ○
  - ○
  - ○

Part 7: Recap

- Core concepts: Infra as code, Git as the source of truth, pull model, converging changes
  -

- Next steps
  -
  -

- Preparation: Access a cluster through kubectl/k9s.

  -

  -