

Infrastructure as Code, Git as Truth

A Gentle Intro to GitOps

KN Cloud Adoption Sessions

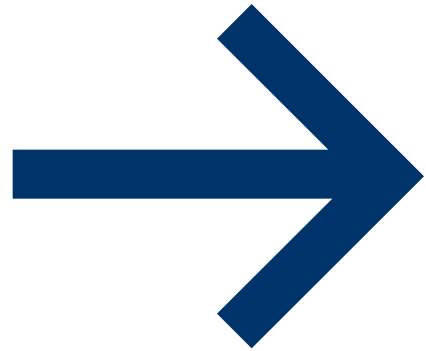
João Pedro Dias (KNITE/MEP)

joao.dias@kuehne-nagel.com

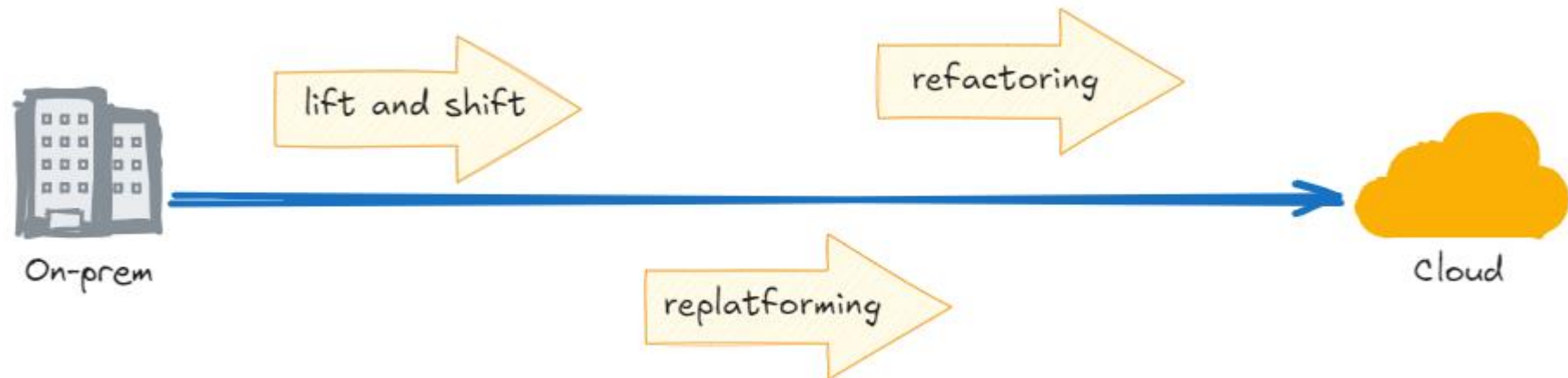
Agenda

- Cloud and AWS
- Infrastructure-as-Code (IaC)
- GitOps
- IaC and GitOps
- Strategies
- Use cases
- Q&A





Cloud Adoption



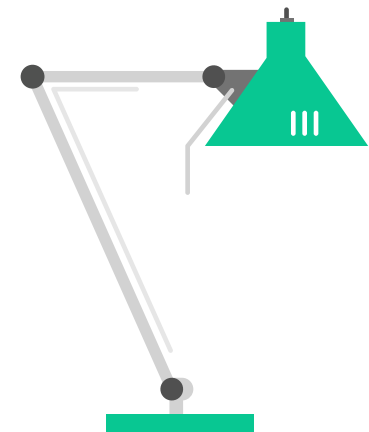
Infrastructure as Code

Infrastructure as Code: Definition

“Infrastructure as Code (IaC) is the managing and provisioning of infrastructure through code instead of manual processes.

With IaC, configuration files are created that contain your infrastructure specifications, which makes it easier to edit and distribute configurations. It also ensures that you provision the same environment every time. By **codifying and documenting your configuration specifications, IaC aids configuration management and helps you avoid undocumented, ad hoc configuration changes.**”

From: What is Infrastructure as Code?, <https://www.redhat.com/en/topics/automation/what-is-infrastructure-as-code-iac>



Infrastructure as Code: Advantages



Infrastructure as Code: Strategies

Imperative

Tell the system **HOW** to do something, detailing every step of the way.



ANSIBLE

```
#!/bin/bash
```



Declarative

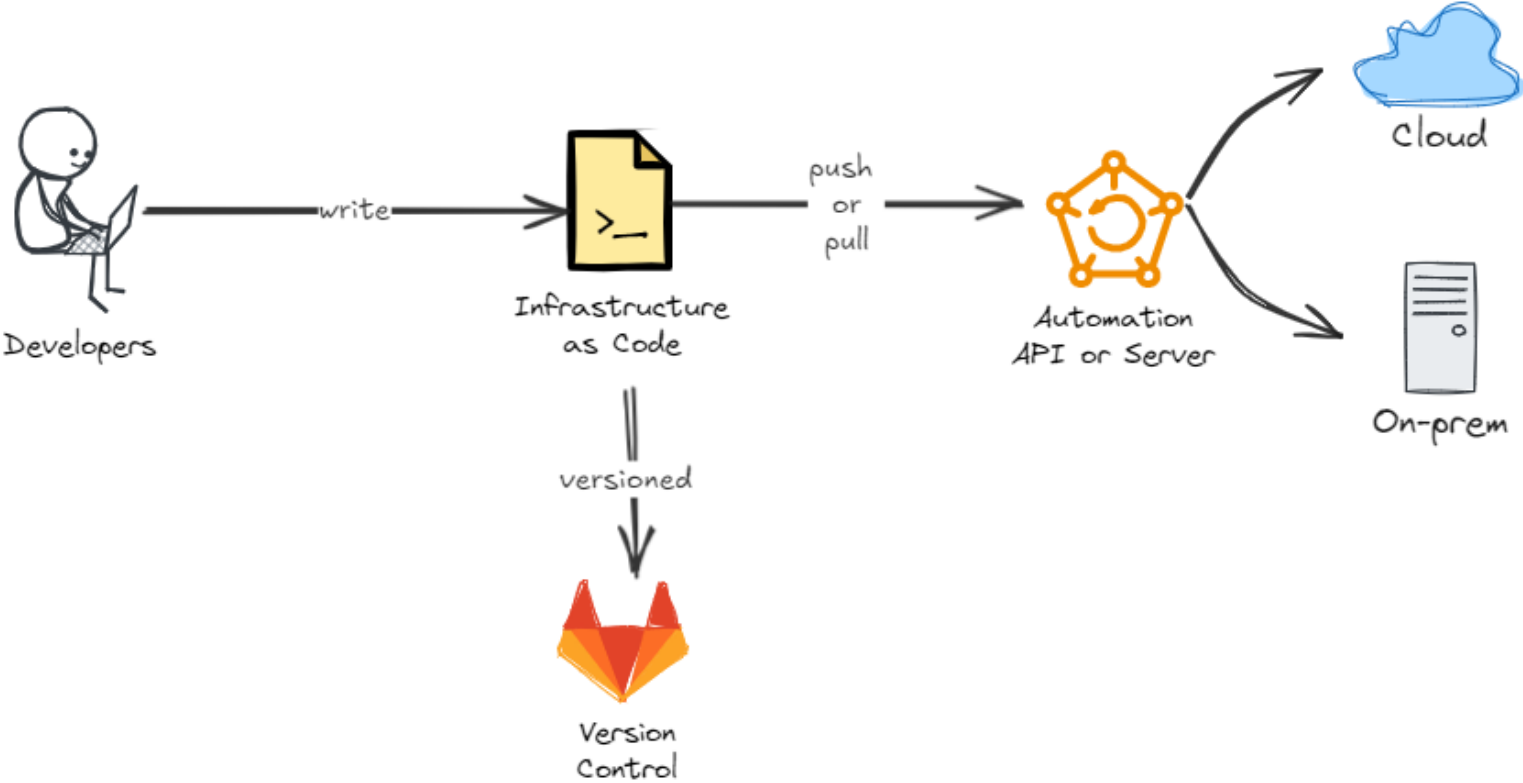
Tell the system **WHAT** you want, and let it figure out HOW to accomplish it.



HashiCorp
Terraform



Infrastructure as Code Workflow



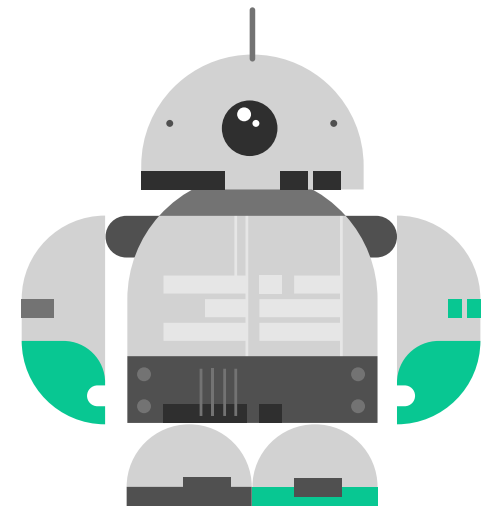
GitOps

GitOps: Definition

“The core idea of GitOps is having a **Git repository that always contains declarative descriptions of the infrastructure** currently desired in the production environment **and an automated process to make the production environment match the described state in the repository.**

If you want to deploy a new application or update an existing one, you only need to update the repository - the automated process handles everything else. **It's like having cruise control for managing your applications in production.**”

From: What is GitOps?, <https://www.gitops.tech/>



GitOps: Advantages

- Easy and fast error recovery (versioning)
- Self-documenting deployments
- Minimal infrastructure access for developers
- Simpler toolchain for CI/CD

Push-based

Whenever the application code is updated, the build pipeline is triggered, updating both application and infrastructure.

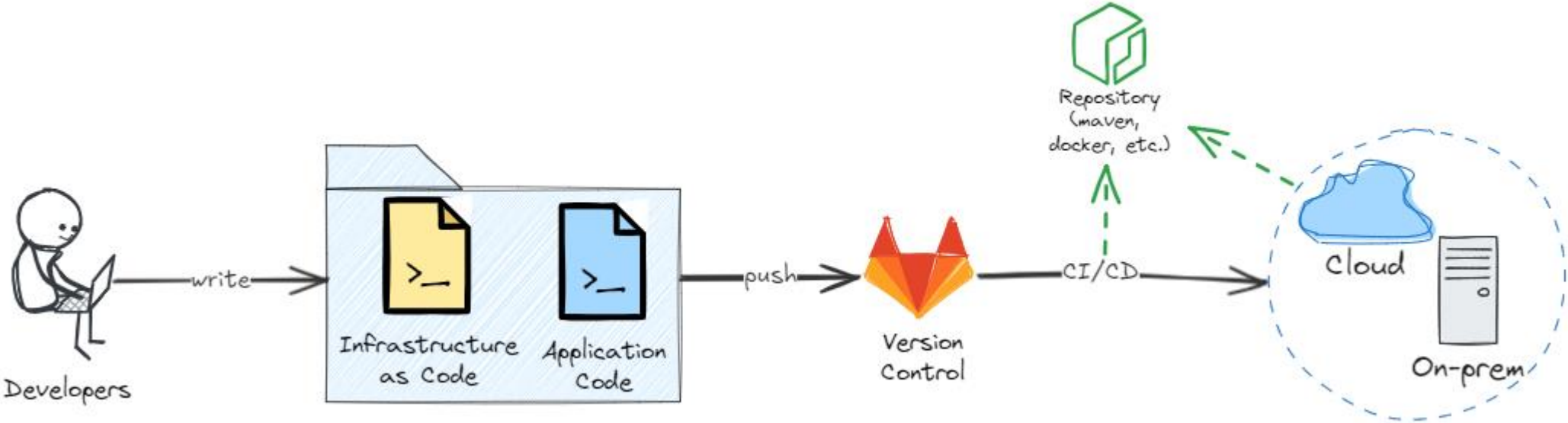


Pull-based

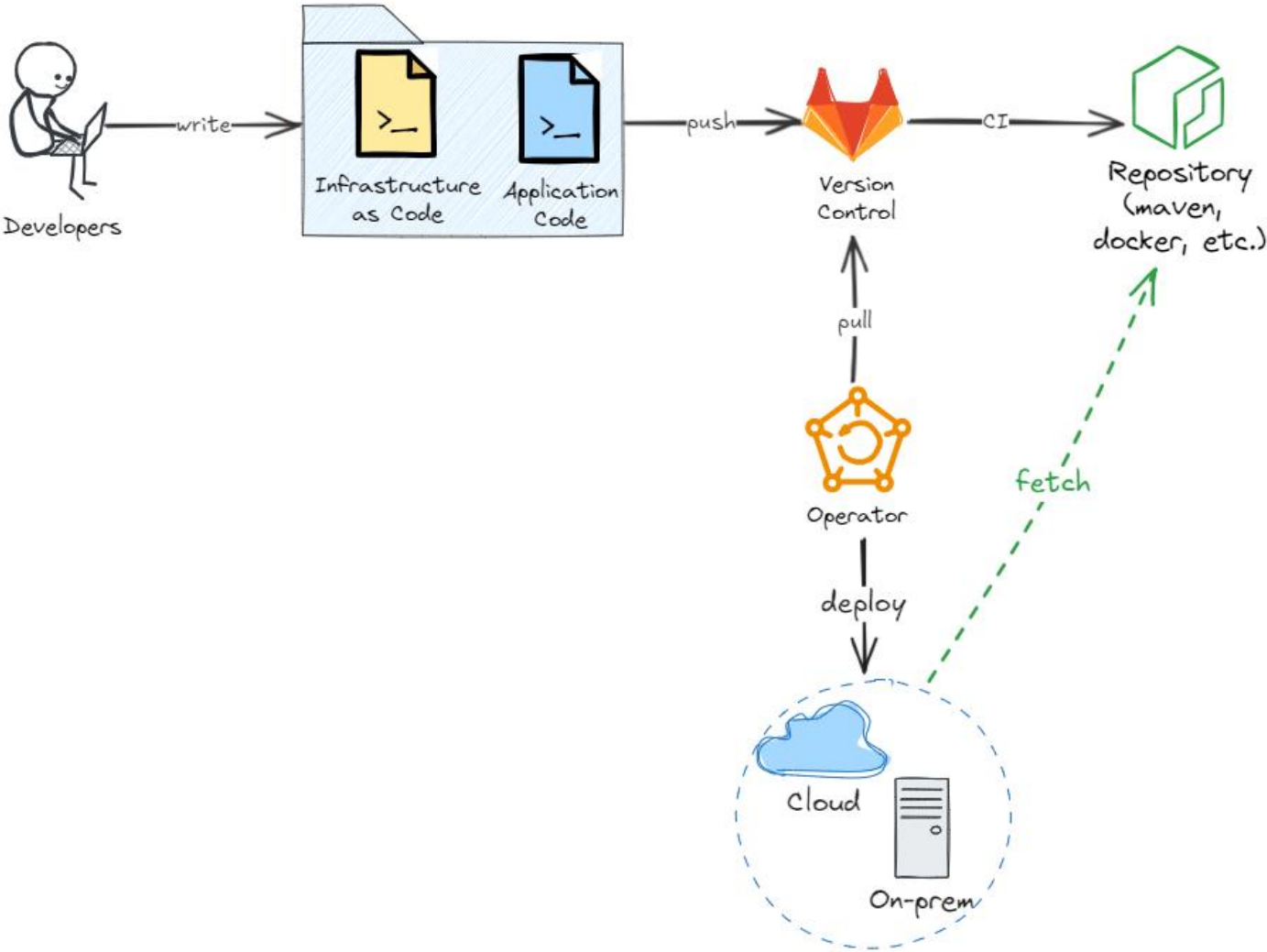
With the pull-based deployment approach, the *operator* is introduced. It takes over the role of the pipeline by continuously comparing the desired state in the environment repository with the actual state in the deployed infrastructure. Whenever differences are noticed, the operator updates the infrastructure to match the environment repository.



GitOps: Workflow (push-based)

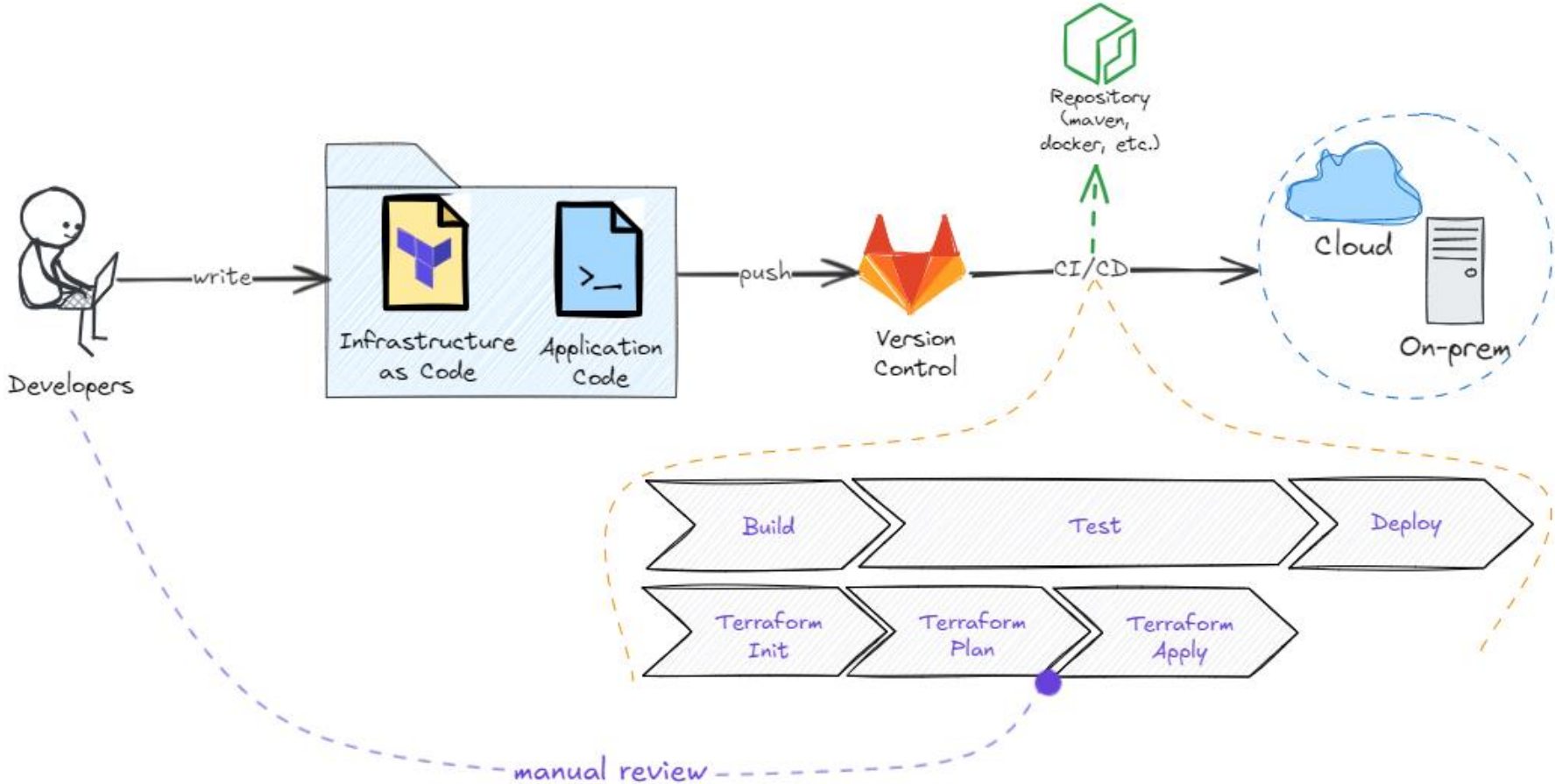


GitOps: Workflow (pull-based)

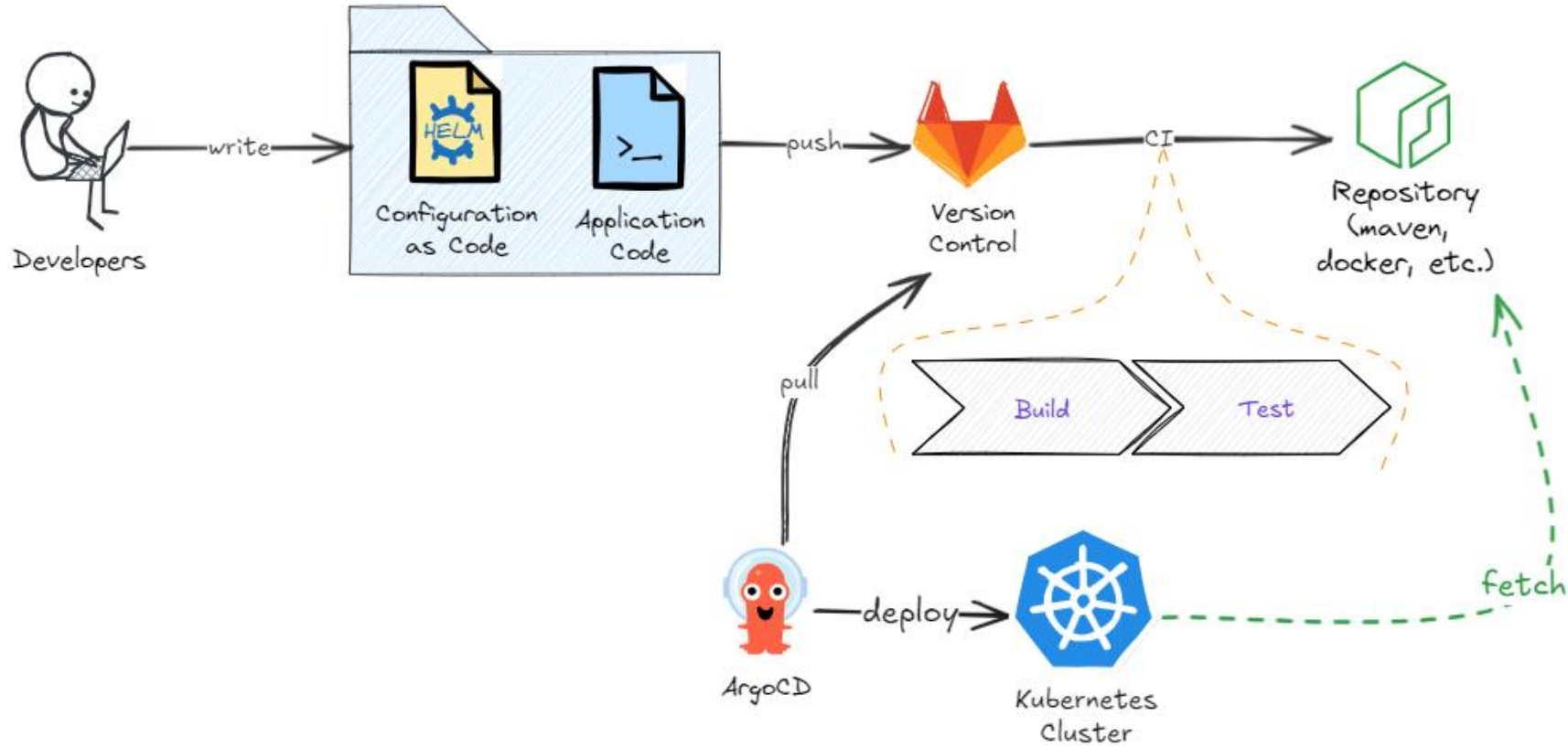


laC + GitOps

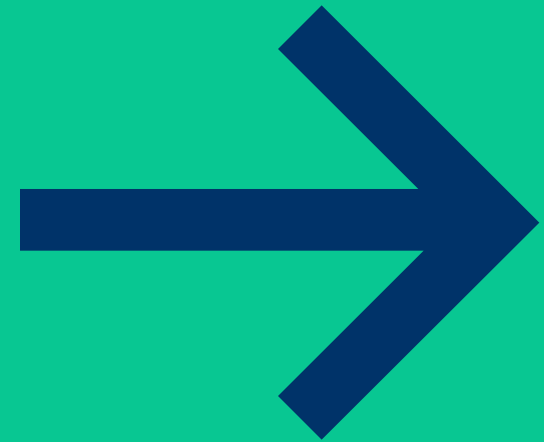
Terraform + Gitlab CI



Kubernetes (Helm, ArgoCD)



Q & A



Inspire. Empower. Deliver.

