

CLIENT PROGRESS REVIEW

Computer Science Department
Swinburne Vietnam Alliance Program HCMC

PROJECT

A SECURE CI/CD PIPELINE FOR APPLICATION DEVELOPMENT

GROUP 1

- **Dang Vi Luan - 103802759**
- **Nguyen Linh Dan – 103488557**
- **Nguyen Duy Khang - 104056476**
- **Tran Bao Huy - 103505799**

Table of Contents

Summary

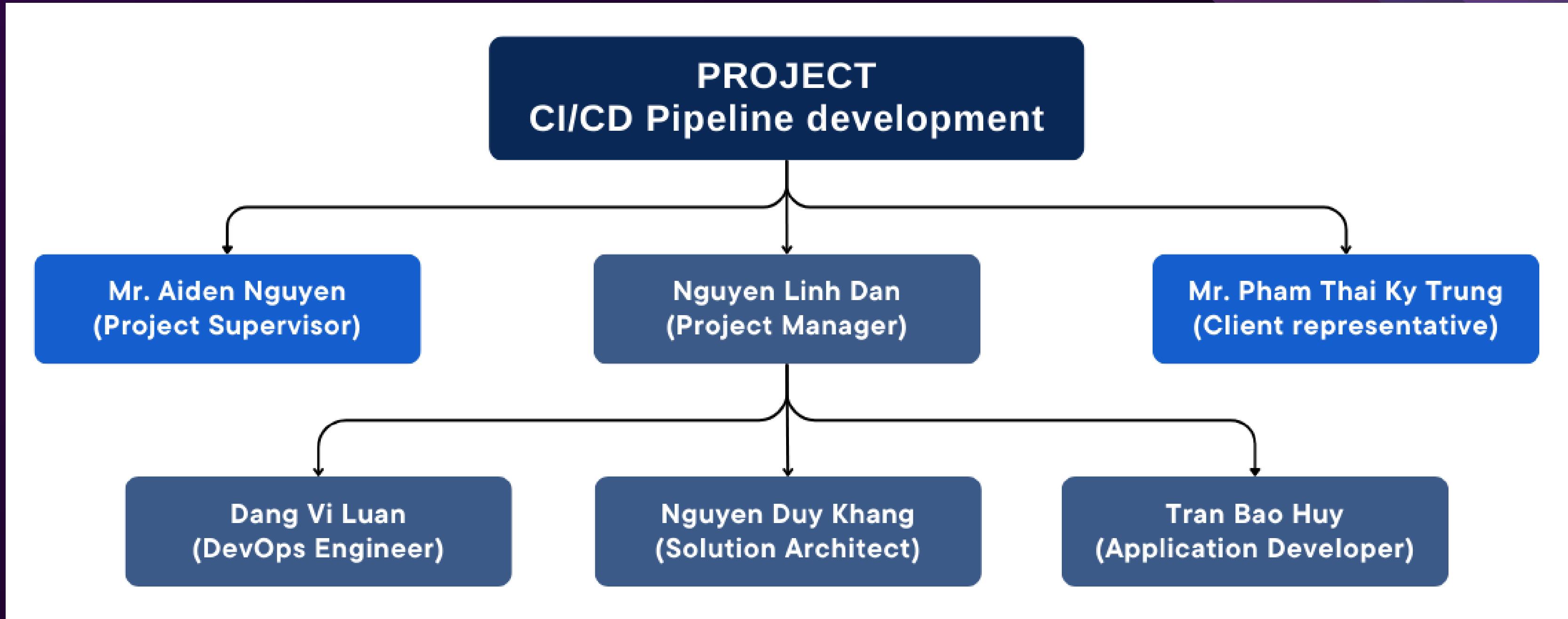
- Project Team
- Project Description / Objective
- Project Scope
- Progress to date - Completed work - Next actions

Up-to-date deliverables

- CI/CD Prototype
- CI/CD Pipeline demo

Questions

OUR PROJECT TEAM



PROJECT DESCRIPTION

Our client

Computer Science department at Swinburne is famous for teaching computer science courses.

They provide students with opportunities to gain real-working experience by developing and deploying application projects.

Problem

The current SDLC is difficult for group collaboration, code merging, change tracking, and lecturer feedback because students are doing everything manually (e.g: gathering all project files manually, testing, and pushing code manually on GitHub)

Solution - Project Deliverables: A SECURE CI/CD PIPELINE

- Automate some stages of the application development cycle (e.g: testing)
- Centralize the project code, especially in big projects
- Make the project assets more secure
- Allow continuous feedback





PROJECT OBJECTIVE

Design the complete CI/CD pipeline

Prepare the infrastructure for the CI/CD pipeline

Implement the CI/CD pipeline

Perform testing after implementation

Prepare User Guide and Training Document

Feedback gathering from client

PROJECT SCOPE

Functional Requirements

DevOps CI/CD Pipeline

- Version Control Integration (Git)
- Continuous Integration - CI (ArgoCD)
- Deployment (Docker)
- Operation (Terraform, Kubernetes)

DevSecOps Integration

- Version Control Integration (AWS Secrets Manager)
- Continuous Integration (SAST, SCA, OWASP ZAP)
- Deployment (Clair to identify vulnerabilities)
- Monitoring and alerting (Prometheus, Grafana)

Non-Functional Requirements

- Performance
- Reliability
- Integrability
- Observability

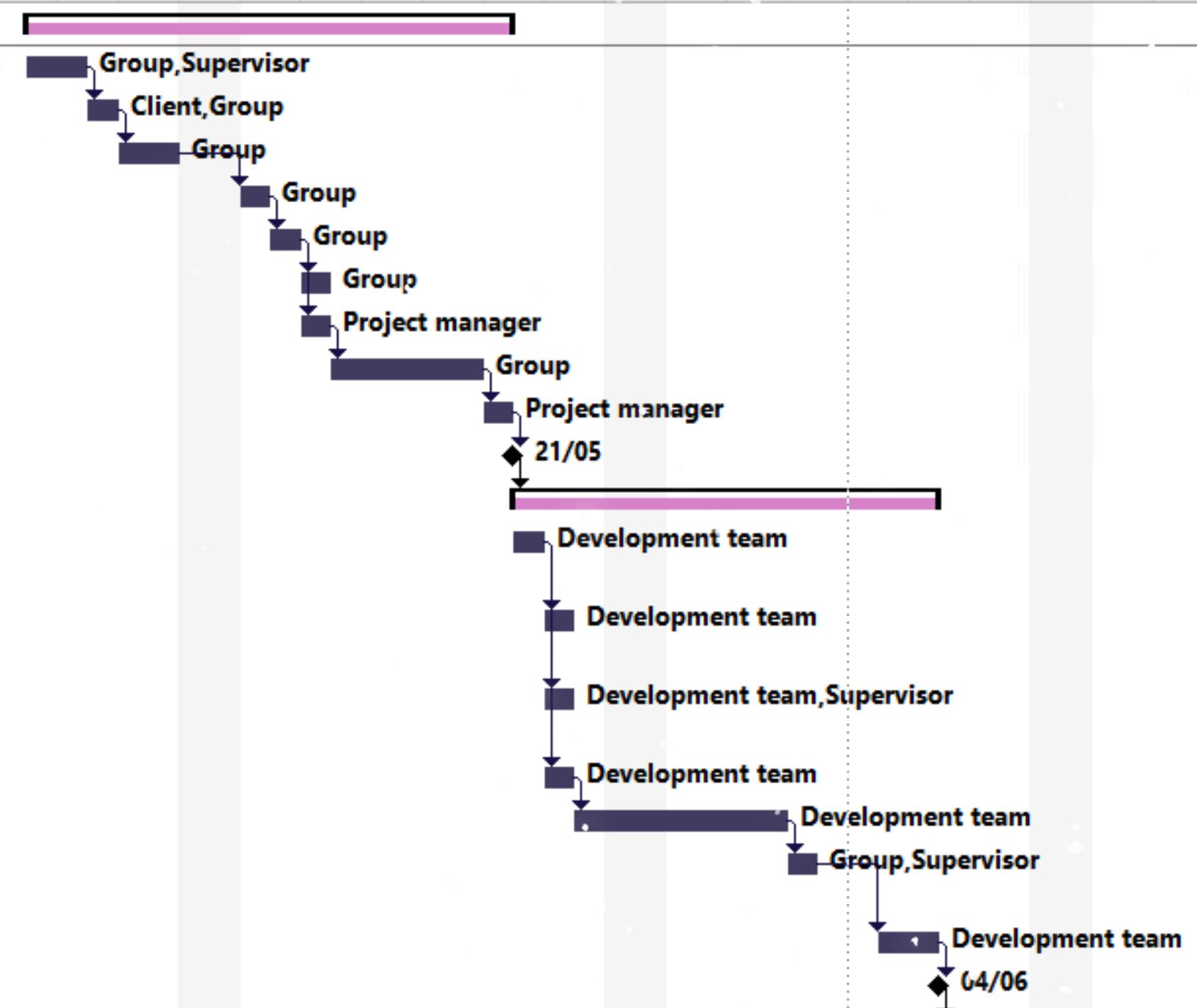
Out of scope

- Training for Cloud Services:
- Marketing Plan
- Third-Party Tools and Services guidelines
- Bug Fixing
- Additional Functionality

UPDATED PROGRESS

We have done Phase 1 and Phase 2

Phase 1: Project Preparation	12 days	Mon 06/05/24	Tue 21/05/24		
Prepare project brief for first client visit	2 days	Mon 06/05/24	Tue 07/05/24		Group, Supervisor
Ask for client signoff in MOU	1 day	Wed 08/05/24	Wed 08/05/24	2	Client, Group
Prepare agenda for first client visit	2 days	Thu 09/05/24	Fri 10/05/24	3	Group
Conduct stakeholder interview	1 day	Mon 13/05/24	Mon 13/05/24	4	Group
Collect and document requirements	1 day	Tue 14/05/24	Tue 14/05/24	5	Group
Prioritize the requirements	1 day	Wed 15/05/24	Wed 15/05/24	6	Group
Define project scope and objectives	1 day	Wed 15/05/24	Wed 15/05/24	6	Project manager
Finalize the initial project proposal	3 days	Thu 16/05/24	Mon 20/05/24	8	Group
Establish communication channels	1 day	Tue 21/05/24	Tue 21/05/24	9	Project manager
Milestone 1: Project proposal completed	0 days	Tue 21/05/24	Tue 21/05/24	10	
Phase 2: Secure CI/CD Pipeline Design	10 days	Wed 22/05/24	Tue 04/06/24	11	
Analyze current development and deployment conditions at Swinburne	1 day	Wed 22/05/24	Wed 22/05/24		Development team
Identify security requirements of pipeline	1 day	Thu 23/05/24	Thu 23/05/24	13	Development team
Determine third-party services to integrate	1 day	Thu 23/05/24	Thu 23/05/24	13	Development team, Supervisor
Determine integration points	1 day	Thu 23/05/24	Thu 23/05/24	13	Development team
Design the CI/CD pipeline architecture	5 days	Fri 24/05/24	Thu 30/05/24	16	Development team
Determine tools and technologies used to develop	1 day	Fri 31/05/24	Fri 31/05/24	17	Group, Supervisor
Develop the pipeline workflow	2 days	Mon 03/06/24	Tue 04/06/24	18	Development team
Milestone 2: CI/CD Pipeline design	0 days	Tue 04/06/24	Tue 04/06/24	19	



UPDATED PROGRESS

PHASE 3

Done

- Setup CI
- Integrate Code versioning
- Prepare deployment scripts
- Implement automated build

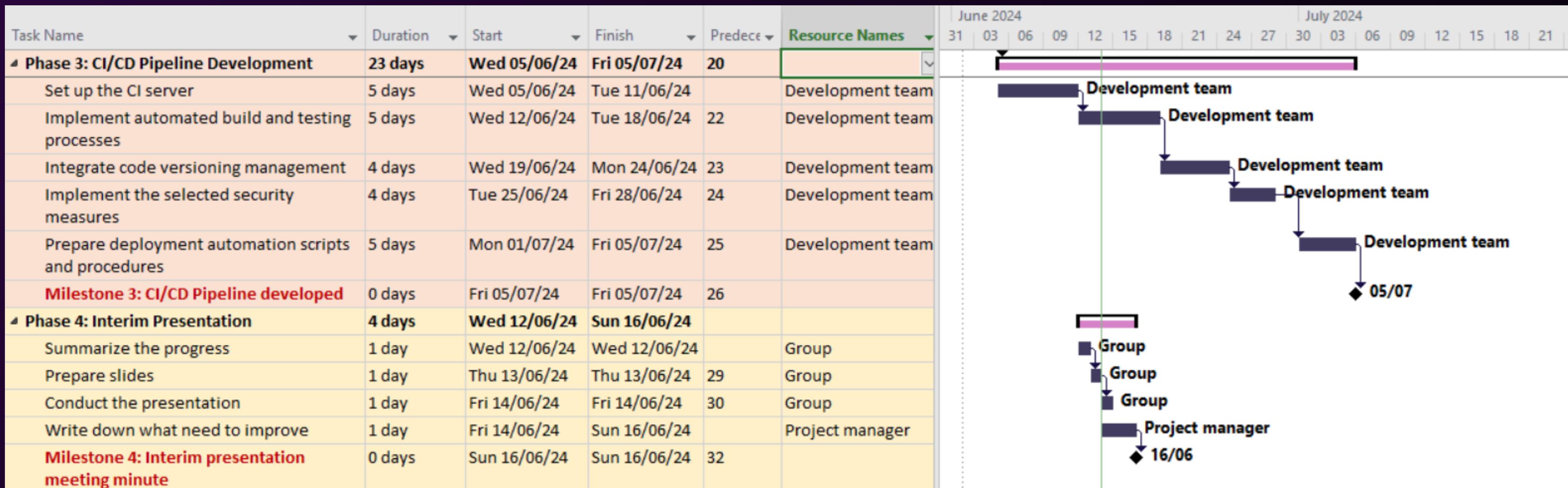
Not Done

- Implement automated testing
- Implement security

Note

We modified the Gantt Chart and let Phase 3 overlap Phase 4 instead of working on them sequentially.

CI/CD Development > Interim presentation > CI/CD Development



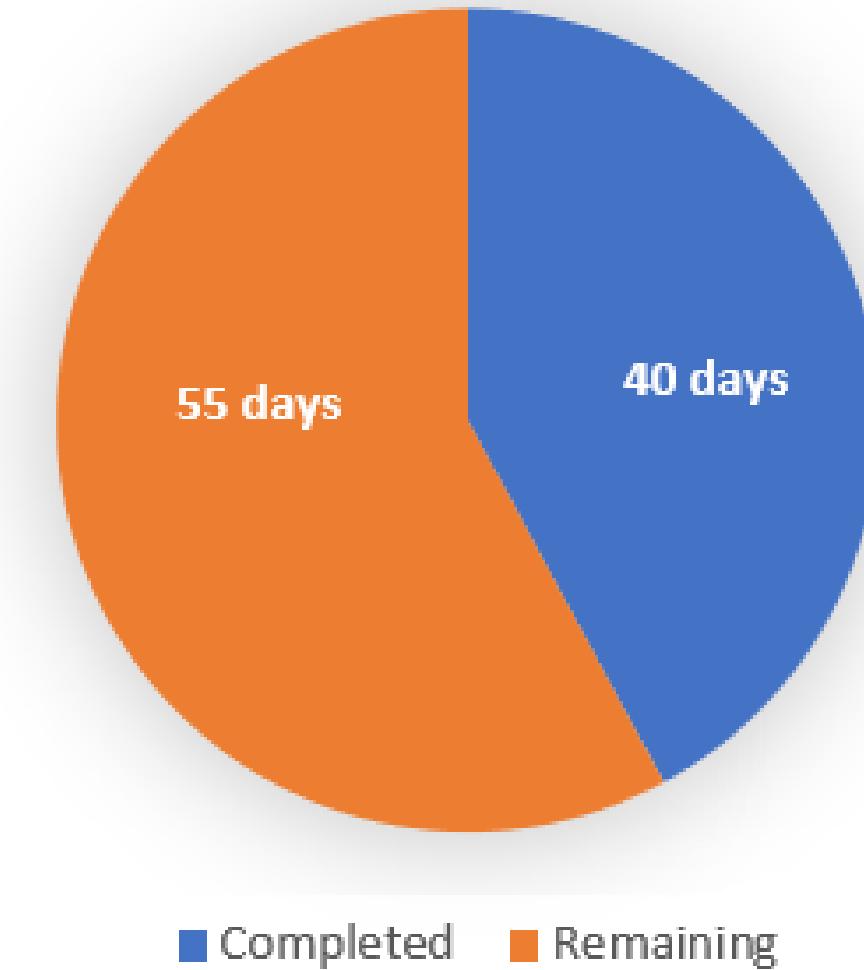
COMPLETED WORK - WEEK 6 (10/6 - 14/6)

- Optimized / Refactored Code Repository
- Completed workflow for Infrastructure deployment
- Application containerization and storage on DockerHub
- Application infrastructure hosting using Kubernetes
- Synchronization and deployment using ArgoCD

NEXT ACTIONS - WEEK 7 (17/6 - 21/6)

- Deploy current workflow on AWS
- Develop test project for testing purposes
- Automate Docker Image pulling using Kubernetes
- Integrate Monitor and Visualization

Updated project timeline



Up-to-date deliverables

- CI/CD PROTOTYPE (WITH DIAGRAMS)
- CI FLOW
- CD FLOW
- CI/CD PIPELINE DEMO

Application code - code

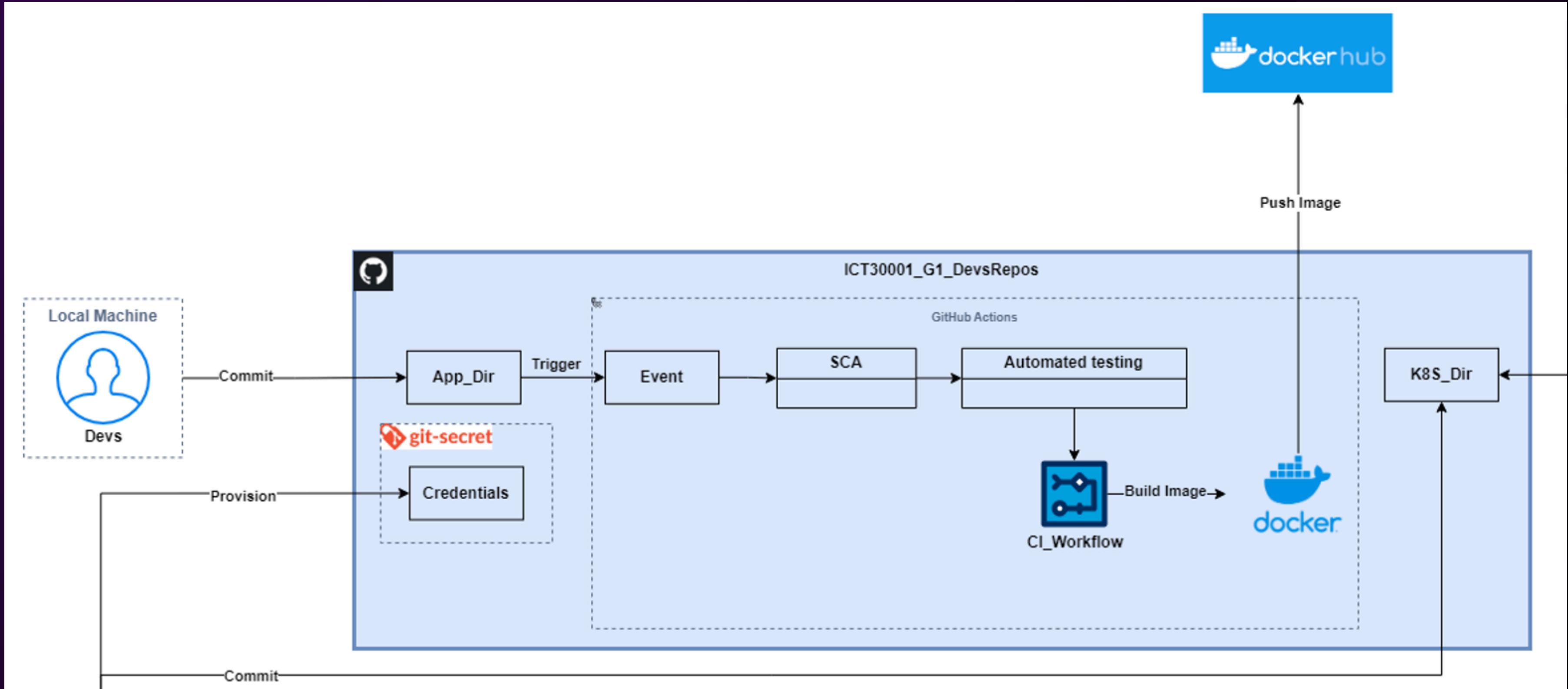


Figure 1: Application code workflow

Infrastructure as Code - IaC

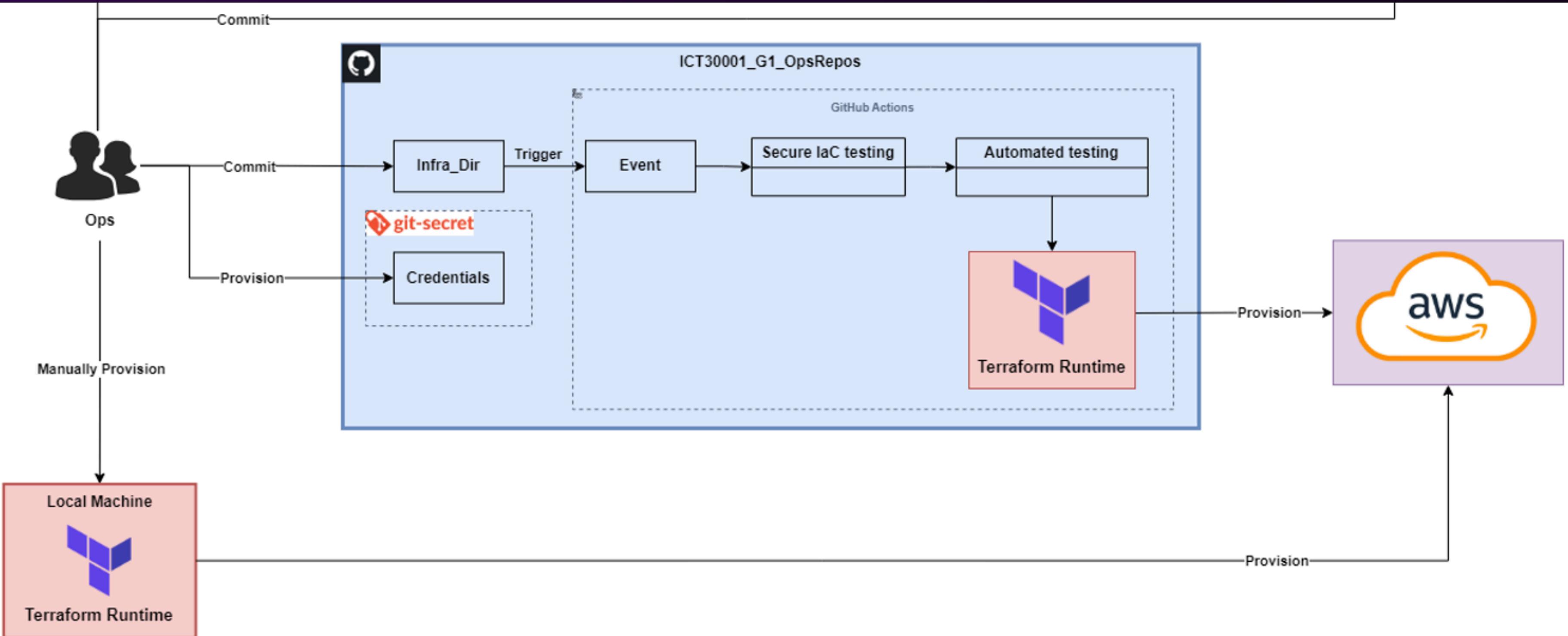


Figure 2: Infrastructure as Code Workflow

Configuration Management

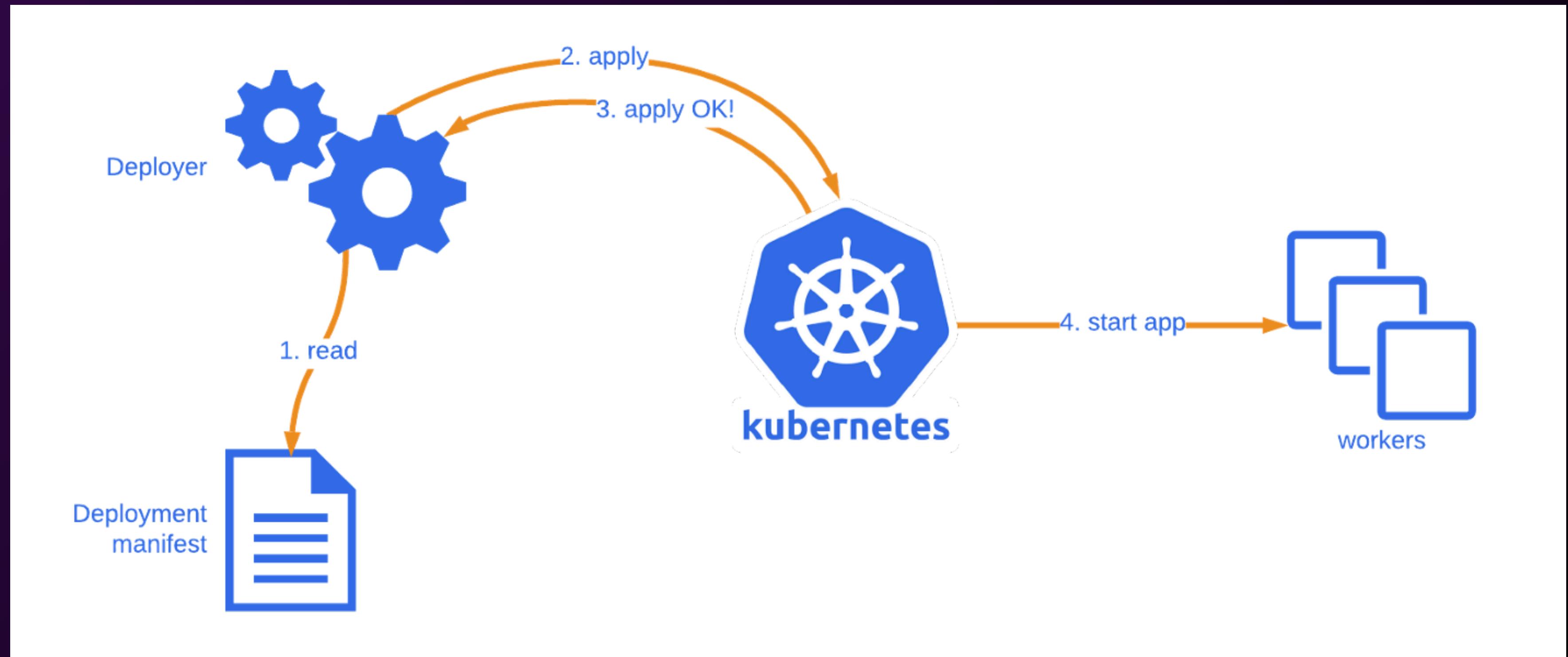


Figure 3: Kubernetes in a nutshell

GitOps-in-a-nutshell

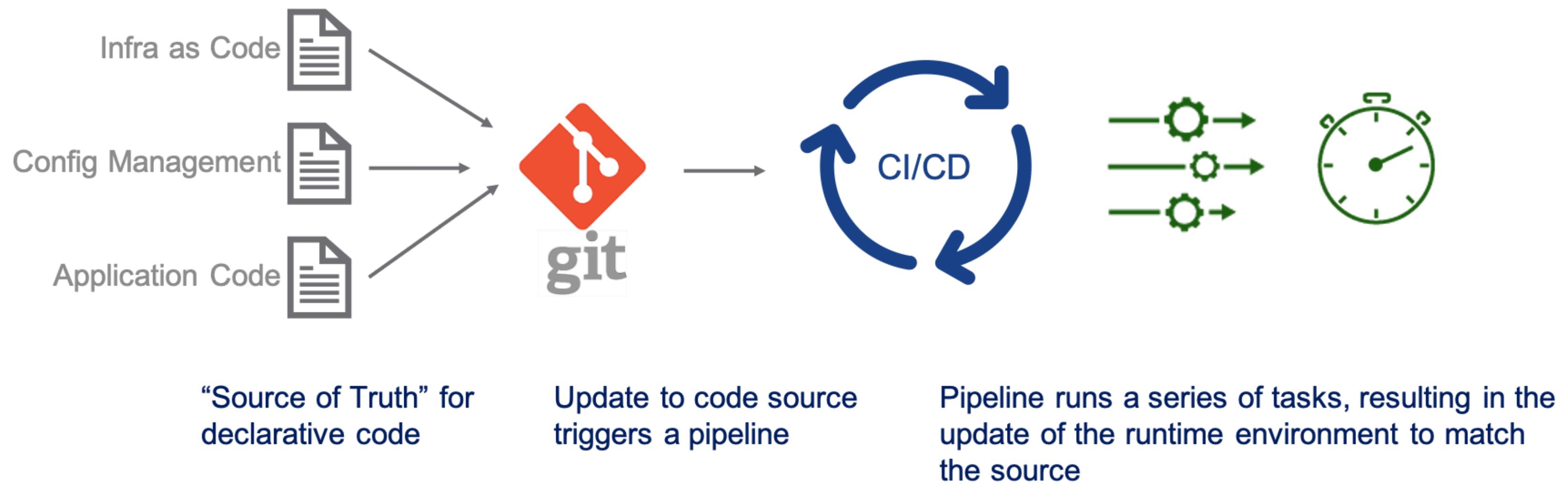


Figure 4: GitOps in a nutshell

GitOps is great because:

- Single Source of Truth for everything using Git
- A clear change history
- Immutable and reproducible deployments
- Reflect changes and support rollback when needed

How to execute GitOps - ArgoCD

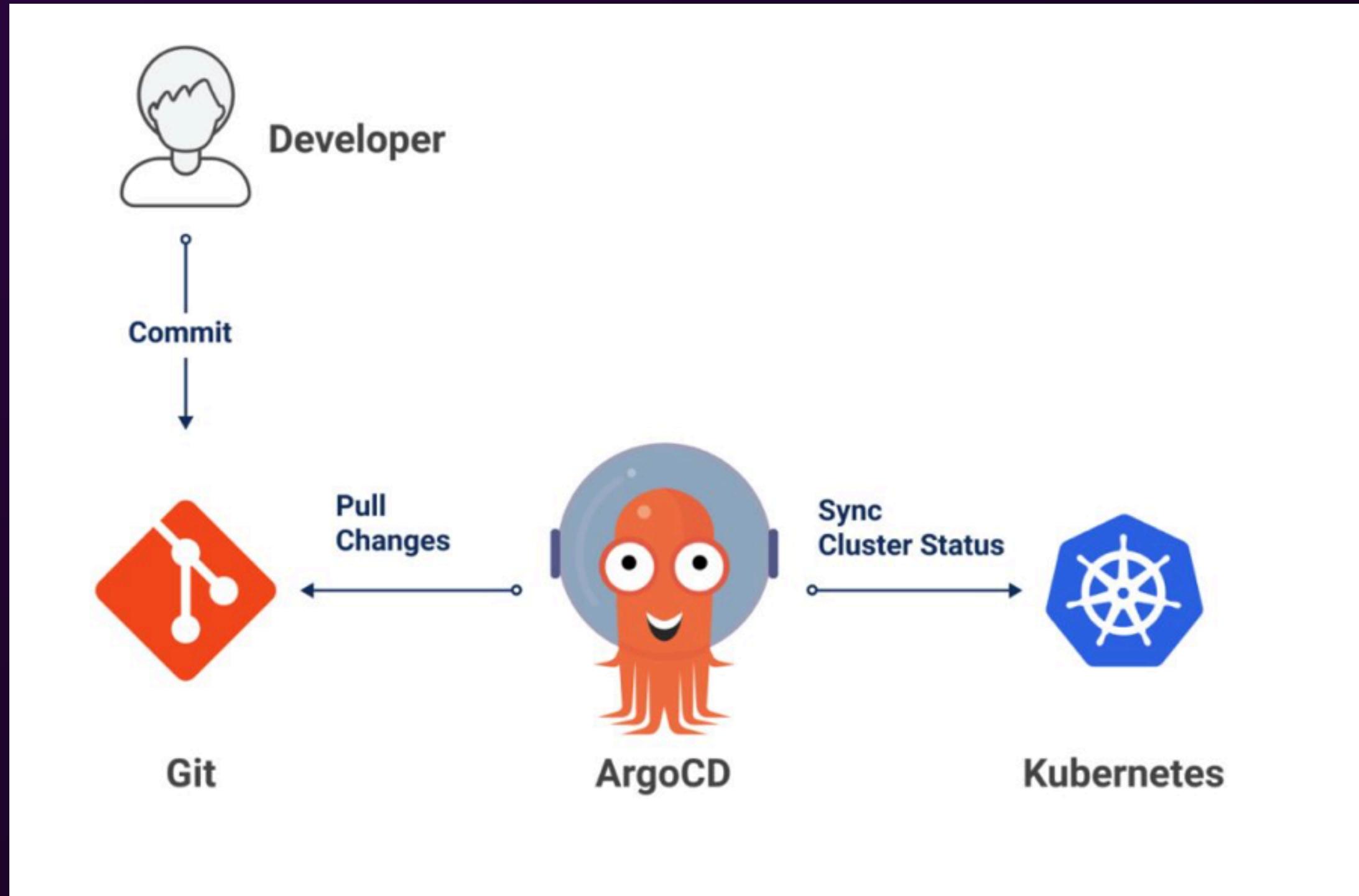


Figure 5: How ArgoCD works

How to execute GitOps - K8S

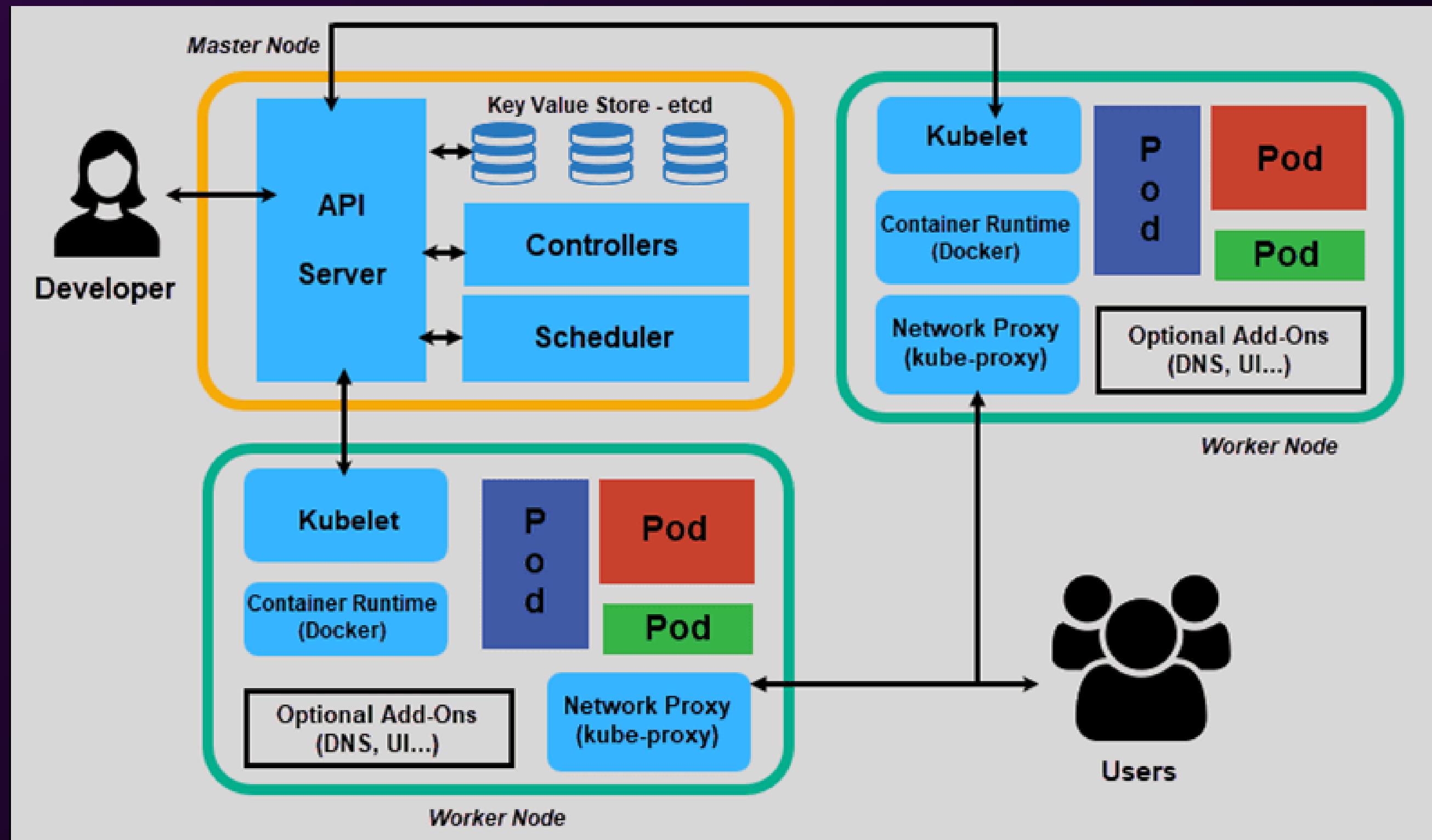
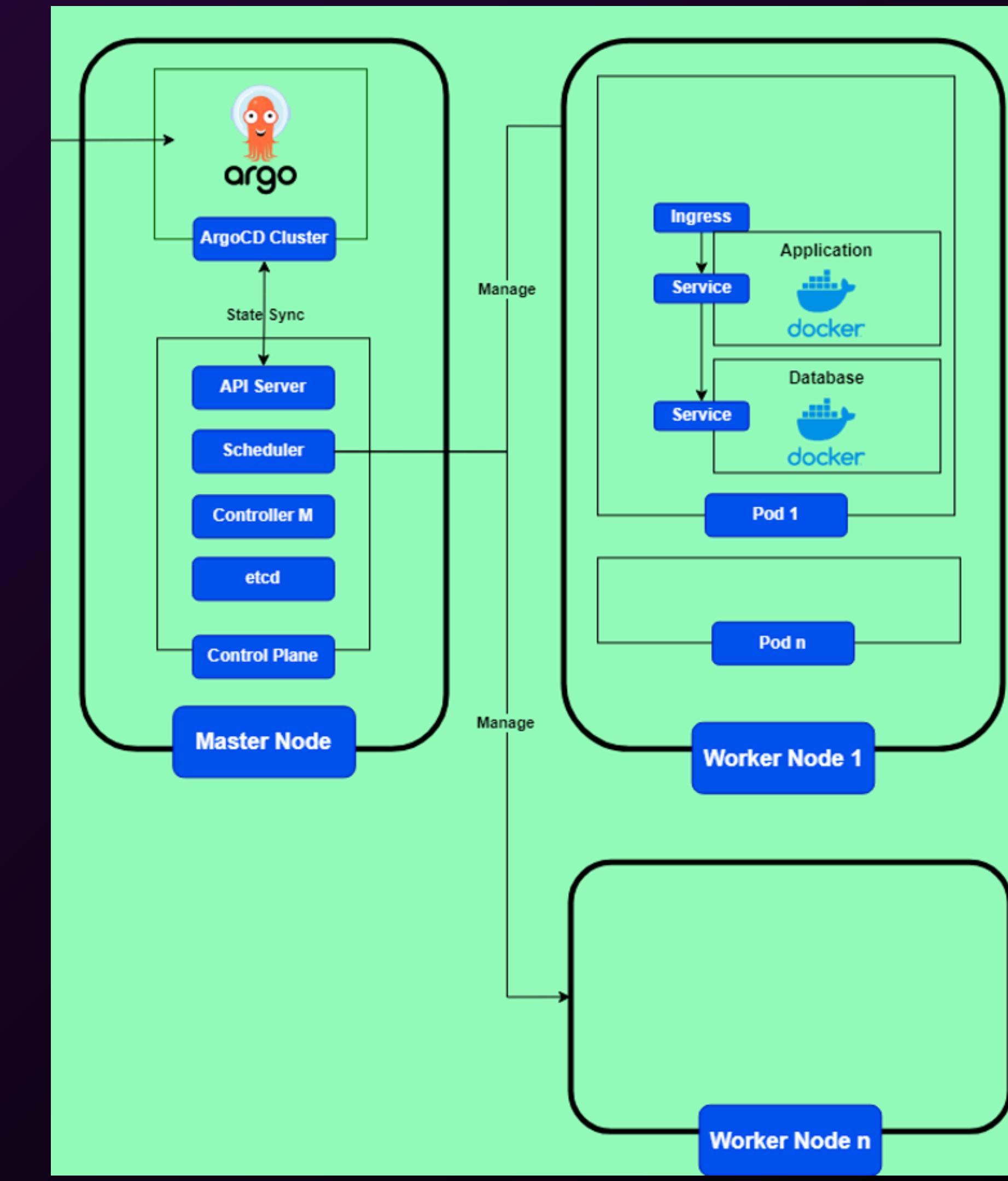


Figure 6: Kubernetes Architecture

Combine them all

Figure 7: Integration between ArgoCD and K8S



DEMO

Demo 1 - CI Workflow

Demo 2 - CD workflow

Demo 3 - Infrastructure syncing

DISCUSSION SESSION

QUESTIONS

PROJECT TEAM 1

THANK YOU