



GitOps with Argo CD



GitOps คือการสร้างระบบ Continuous Delivery (CD) ด้วย tools ที่เราใช้กัน เป็นอยู่แล้ว ซึ่งก็คือ Git นั่นเอง โดยเราจะทำการเก็บสิ่งต่างๆ ที่ใช้ในการสร้าง infrastructure ให้ application ของเรา ให้อยู่ใน Git repository และเมื่อข้อมูลในนั้น มีการเปลี่ยนแปลง (มี commit ใหม่) ระบบก็จะจัดการสร้างสิ่งต่างๆ ที่เราทำการ

ระบุไว้ใน Git ให้โดยอัตโนมัติ

Argo CD ทำงานในรูปแบบ pull deployment ซึ่ง Argo CD ที่เราติดตั้งลงไป ใน Kubernetes Cluster ของเรา จะเฝ้าดู Git Repository และทำให้ states ใน Cluster ของเราตรงกับที่ config ที่อยู่ใน Git และเมื่อมี commit ใหม่ใน Git repository มันก็จะทำการจัดการ สร้าง ลบ อัปเดต resources ต่างๆ ใน cluster ของเรา ให้ตรงกับใน Git นั่นเอง

วัตถุประสงค์:

- เรียนรู้การสร้างระบบ Continuous Delivery (CD)
- เรียนรู้ขั้นตอนการนำ GitOps เข้ามาใช้งานในโปรเจกต์
- เข้าใจแนวทางการแก้ไขปัญหาที่เกิดขึ้นจากการใช้งาน
- เข้าใจการเก็บสิ่งต่างๆ ที่จำเป็นต่อการสร้าง Infrastructure
- เข้าใจการเก็บสิ่งต่างๆ ที่จำเป็นต่อการสร้าง Environment
- เข้าใจการเก็บสิ่งต่างๆ ที่จำเป็นต่อการสร้าง Automation process
- เรียนรู้การทำงานร่วมกับ GitLab CI
- เรียนรู้การทำงานร่วมกับ Argo CD Vault Plugin



กลุ่มเป้าหมาย:

- นักเรียนนักศึกษา
- ครู อาจารย์ วิทยากรที่สนใจ
- นักวิชาการ นักไอที หรือผู้ดูแลระบบ
- ตลอดจนผู้สนใจทั่วไป

ความรู้พื้นฐาน:

- ใช้ระบบ Windows หรือ OSX ได้
- พื้นฐานการใช้งาน GitHub
- พื้นฐานการค้นหาข้อมูลจากอินเทอร์เน็ต

ระยะเวลาในการอบรม:

- 18 ชั่วโมง (3 วัน)

ราคาคอร์สอบรม:

- 9,500 บาท / คน (ราคานี้ยังไม่ได้รวมภาษีมูลค่าเพิ่ม)

วิทยากรผู้สอน:

- อาจารย์สามิตร ไทยม

คอร์สที่ควรอบรมก่อนหน้า:

- Git GitHub and Bitbucket for Beginner (พื้นฐานการใช้งาน Git)



เนื้อหาการอบรม:

Section 1: Introduction

- Knowledge Prerequisites
- Tool setup and config
- What is GitOps

Section 2: Core Concepts

- Intro to ArgoCD
- Core Concepts
- ArgoCD Architecture Overview

Section 3: Setting up ArgoCD

- Installation options
- Non-HA Setup
- Getting Initial Admin Password
- Accessing ArgoCD Server
- Access ArgoCD Server Using Port-Forward
- Install ArgoCD CLI

Section 4: Applications

- Defining Applications
- Creating an Application Declaratively using Yaml
- Creating an Application Using Web UI
- Creating an Application Using CLI
- Tools Detection
- Helm Options
- Directory of Files Options
- Kustomize Options



Section 5: Projects

- Why Projects
- Creating Projects
- Creating Basic Project
- Creating a Project with Allowing Specific Destinations
- Project Roles

Section 6: Repositories

- Private Git Repos
- K8s Secret for Argo CD Repos
- Private Repos using Https
- Private Repos using SSH
- Private Helm Repos
- Credential Templates

Section 7: Sync Policies and Options

- Automated Sync
- Automated Pruning
- Automated Self-Healing
- Sync Options
- Fail On Shared Resources
- Replace Resources

Section 8: Tracking Strategies

- Tracking Strategies
- Tracking Git Tag
- Tracking Git Commit SHA
- Tracking HEAD
- Tracking Helm Chart Range of Versions
- Tracking Helm Chart Latest Version
- Tracking Strategies for Helm Charts



Section 9: Diffing Customization

- Diffing Customization
- Diffing Customization Demo
- Diffing Customization, Istio Case

Section 10: Sync Phases and Waves

- Sync Phases and Hooks
- Sync Waves

Section 11: Remote Kubernetes Clusters

- Defining K8s Clusters
- Demo: Remote Clusters
- Practice (Interactive) - Remote Clusters

Section 12: Automation b CI Pipelines

- CICD Flow
- Demo: Basic CI Pipeline
- Practice (Interactive) - Basic CI/CD