DevOps Introduction

Definition, Concepts, Demo

Mehdi Elwafi

2024

About ME

El Mehdi El Wafi
ENSIAS, Msr Cloud & HPC

Blog

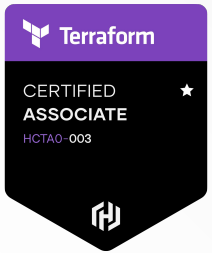
mehdij4.com

LinkedIn

@elwafi-elmehdi

Role

Cloud & DevOps Engineer at YouCan



DevOps Introduction

#1 Definition

- DevOps is a set of practices
- Shorten the development life cycle
- Combine two spaces (Dev + Ops)
- Unified perspective on software development and delivery.

#2 Core Concepts

- Collaboration and Communication
- Automation
- Continuous Integration/Continuous Delivery (CI/CD)
- Infrastructure as Code (IaC)

#3 Why use DevOps?

- Faster Delivery and Shorter Development Cycles.
- Automation
- Enhanced Security and Compliance
- Continuous Improvement and Learning
- Benefits from both worlds (Ops, Dev)

DevOps Stages & Tools

#1 Version Control

- Where the code resides
- Pair reviewing Pull requests
- Mono Repo
- Tools:
  - Git
  - Github
  - Azure DevOps
  - GitLab

#2 Build

- Process compiles the code, resolves dependencies, and packages it into (Jar,War,Docker).
- Tools:
  - Maven
  - Gradle
  - JS-bundlers : parcel,webpack
  - Make
  - Docker

#3 Testing

- Verifying that the code behaves as expected.
- Automated tests such as unit tests, integration tests...
- Tools:
  - JUnit
  - Mockito
  - Jest

#4 CI/CD

- CI (Continuous Integration) is practice of frequently merging code changes into a shared repository.
- CD (Continuous Delivery) extends CI by ensuring that code changes are automatically prepared for release to a production environment.
- Automated Releasing.
- GitOps workflow.
- Tools:
    - Jenkins
    - Travis CI
    - Ansible

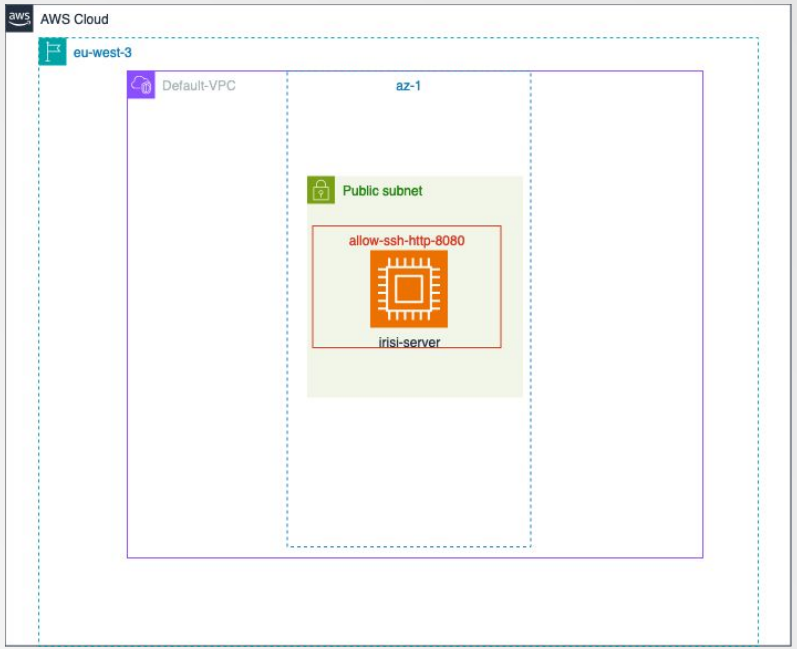
#5 Deployment

- The process of running the application in full production.
- Configuring network subnets,routes,firewalls.
- Monitoring system and app metrics.
- Collecting App logs.
- Troubleshooting performance and bugs.
- Tools:
  - Terraform
  - Ansible
  - Grafana,Prometheus, ELK stack
  - Chef
  - Bash scripts

Demo Time!

Objectives

- Setup AWS credentials
- Deploy EC2 instance using IaC
- Deploy the application using automation
- Verifying the application is actually running in port 8080.
- Tools: Terraform & Ansible



Source Code

Q/A Time!

Takeaways

- Choose one discipline and get really good at it.
- Learn Git and (GitLab or github).
- Learn to write at least unit tests.
- Get efficient with CLI (Linux, or MacOS).
- Try to keep in mind the cost of the code you are developing.
- Automate repetitive tasks.

Thank you