Data Engineering and MLOps in Business

CI/CD & Yaml & SSH

Primoz Konda

AAUBS

March 11, 2025

pk@business.aau.dk

Outline

1 Intro

2 CI/CD

3 GitHub Actions

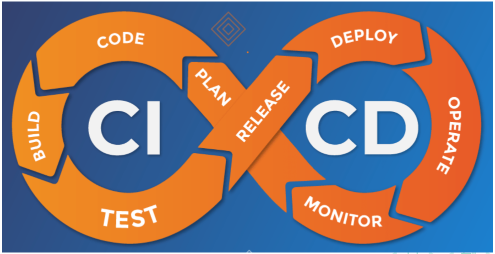
4 Secure SH (SSH)

Where did we end yesterday?

- ?
- Questions?

What is CI/CD?

- **Continuous Integration (CI):** The practice of automatically integrating code changes into a shared repository multiple times a day.
- **Continuous Deployment (CD):** The process of automatically deploying integrated changes to production or a staging environment.



Why CI/CD Matters in MLOps

- Automates model(app) training, testing, and deployment.
- Ensures reproducibility and consistency in ML workflows.
- Helps detect and mitigate model drift over time.

Popular CI/CD Tools

- **GitHub Actions** (integrated with GitHub, declarative YAML syntax)
- **GitLab CI/CD** (pipeline-driven, strong Kubernetes support)
- **Jenkins** (open-source, highly customizable, strong community support)
- **CircleCI** (cloud-native, efficient parallelism, strong caching mechanisms)

Introduction to GitHub Actions

- A CI/CD automation tool built into GitHub.
- Uses YAML-based workflow files stored in `.github/workflows`.
- Can be triggered by events like code push, pull request, or scheduled execution.
- Documentation: Yaml Documentation

GitHub Actions Workflow: NEWS Example

IDEA: Make an app that scrapes news from specific News Website, use LLM to overview it, and finally gives us a morning recap.

YAML

- YAML (Yet Another Markup Language) is used for configuration.
- Uses indentation-based syntax.
- Commonly used in CI/CD pipelines, Kubernetes, and configuration files.

What is SSH?

- SSH (Secure Shell) is a cryptographic network protocol for securely operating network services over an unsecured network.
- Commonly used for remote login and command execution on servers.
- Provides secure authentication, encryption, and integrity.

Basic SSH Commands

Connecting to a Remote Server

```
ssh user@remote_host
```

Using a Specific Port

```
ssh -p 2222 user@remote_host
```

Running a Single Command

```
ssh user@remote_host "ls -l /var/log"
```

Key-Based Authentication

```
ssh-keygen -t rsa -b 4096
ssh-copy-id user@remote_host
```

- Generates an SSH key pair.
- Copies the public key to the remote server for password-less login.

File Transfers with SCP

Copy a File from Local to Remote

```
scp file.txt user@remote_host:/remote/directory/
```

Copy a File from Remote to Local

```
scp user@remote_host:/remote/file.txt /local/directory/
```

Copy a Directory Recursively

```
scp -r local_dir user@remote_host:/remote/directory/
```

SSH Tunneling (Port Forwarding)

Local Port Forwarding (-L)

```
ssh -L 8080:localhost:80 user@remote_host
```

Maps local port 8080 to remote port 80.

Remote Port Forwarding (-R)

```
ssh -R 9090:localhost:3000 user@remote_host
```

Allows remote access to local port 3000 through port 9090.

Best Practices

- Use SSH keys instead of passwords.
- Use `fail2ban` or other security measures to prevent brute-force attacks.
- Use SSH agent (`ssh-agent`) for managing keys securely.