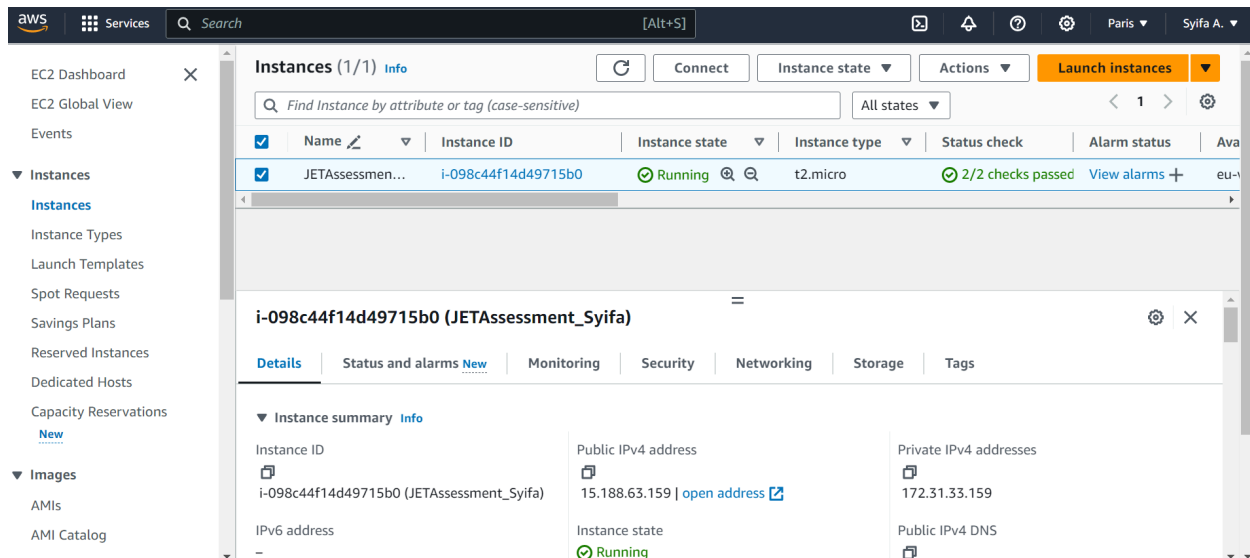# CI/CD Pipeline

Jet Platform Engineering Early Careers Assessment

Syifa Amri

AWS EC2 Instance

I started the assessment by creating an Instance on AWS EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service offered by Amazon Web Services (AWS) that provides resizable compute capacity in the cloud. In simpler terms, it allows users to rent virtual servers, known as instances, on which they can run their applications.

Features:

1. Instances: These are virtual servers that users can rent from AWS. Instances are available in various configurations, such as different CPU, memory, storage, and networking capacities, to suit different workload requirements.
2. Variety of Operating Systems: EC2 supports a wide range of operating systems, including various Linux distributions, Microsoft Windows, and FreeBSD, allowing users to choose the operating system that best fits their needs.
3. Security: EC2 instances run in a virtual private cloud (VPC), providing network isolation and control over inbound and outbound traffic. Users can also implement additional security measures such as firewalls, security groups, and encryption to protect their instances and data.

4. Integration with Other AWS Services: EC2 integrates seamlessly with other AWS services, such as Amazon S3 for storage, Amazon RDS for databases, Amazon VPC for networking, and AWS IAM for access management, allowing users to build complex and scalable architectures.

This instance has an IAM user that I have set up with its own credentials
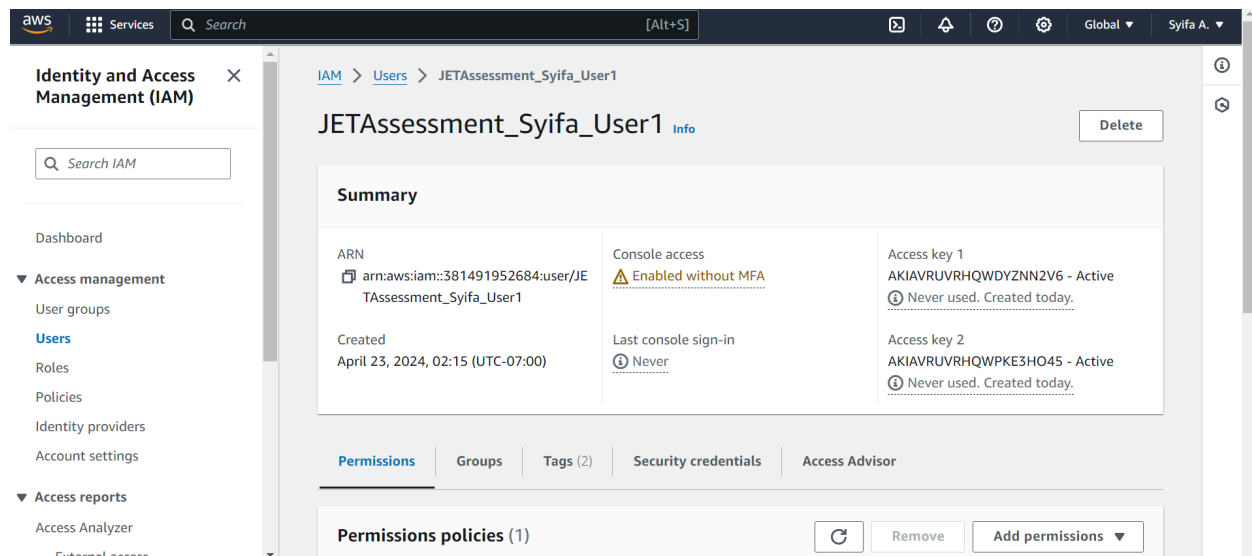
The secret keys have been stored as Secrets in my repository, allowing for the CI/CD workflow to automate its process securely.

Following that, I created a GitHub workflow yaml file for the CI/CD automation in the workflow directory.

Definition of Version Control:
Version control is a system that records changes to files over time, allowing you to recall specific versions later. It enables multiple developers to work on the same codebase simultaneously while keeping track of changes made by each contributor. Version control helps maintain code integrity, facilitates collaboration, and provides a history of changes.

Overview of Git:

Git is a distributed version control system widely used in software development. It allows developers to track changes to source code, collaborate with others, and manage codebase efficiently. Git provides features such as branching, merging, tagging, and distributed repositories, making it powerful and flexible for managing code versions.

Introduction to CI/CD, Automation, and Basic Cloud Deployment:

Continuous Integration (CI) and Continuous Deployment (CD) are practices in software development aimed at automating the integration, testing, and deployment of code changes. CI focuses on integrating code changes frequently, running automated tests, and providing fast feedback to developers. CD extends CI by automating the deployment process, allowing code changes to be released to production automatically. Automation plays a crucial role in CI/CD pipelines by reducing manual effort, ensuring consistency, and enabling faster delivery of software. Basic cloud deployment involves deploying applications on cloud infrastructure, leveraging advantages such as scalability, flexibility, and cost-effectiveness.

2. Version Control

Explanation of Version Control Systems (VCS):
Version control systems (VCS) are software tools that manage changes to files over time. They maintain a history of revisions, enabling users to track changes, revert to previous versions, and collaborate with others. VCS helps manage codebase complexity, prevent conflicts, and ensure code integrity in collaborative environments.

Benefits of Version Control:

Version control offers several benefits, including:

- Collaboration: Multiple developers can work on the same codebase simultaneously without interfering with each other's changes.
- Version History: Maintains a record of every change made to the code, allowing developers to track progress and revert to previous versions if needed.
- Code Integrity: Ensures that changes are tracked, documented, and validated, maintaining the stability and reliability of the codebase.

Overview of Git Features:

Git provides various features that enhance version control, such as:

- Branching: Allows developers to create separate branches to work on specific features or fixes independently.
- Merging: Integrates changes from one branch into another, allowing developers to combine code changes seamlessly.
- Tagging: Marks specific points in the version history, such as releases or milestones, for easy reference.

3. CI/CD Concepts

Definition of Continuous Integration (CI):
Continuous Integration (CI) is a development practice where developers integrate code changes into a shared repository frequently, typically several times a day. Each integration triggers automated tests to ensure code quality and detect issues early in the development process.

Explanation of Continuous Deployment (CD):

Continuous Deployment (CD) is an extension of CI that automates the deployment process, allowing code changes to be released to production automatically. CD ensures that validated code changes are deployed to production environments quickly and reliably.

Benefits of CI/CD:

CI/CD offers several benefits, including:

- Faster Development Cycles: Automates repetitive tasks, such as building, testing, and deploying code changes, resulting in shorter development cycles and faster time-to-market.
- Improved Code Quality: Ensures that code changes are validated through automated tests, code reviews, and continuous integration, leading to higher code quality and reliability.
- Reduced Risk: Detects and addresses issues early in the development process, reducing the risk of introducing bugs or regressions in production environments.

4. Automation in Software Development

Importance of Automation:
Automation plays a critical role in software development by:

- Streamlining Development Processes: Automates repetitive tasks, such as code compilation, testing, and deployment, freeing up developers' time for more valuable work.
- Ensuring Consistency: Reduces human error and ensures that development, testing, and deployment processes are consistent and repeatable across environments.
- Improving Scalability: Enables organizations to scale their development and deployment processes efficiently, accommodating changes in workload and demand.

Role of Automation in CI/CD Pipelines:

CI/CD tools automate various stages of the software development lifecycle, including:

- Build: Compiles source code, packages application artifacts, and performs static code analysis.
- Test: Executes automated tests, such as unit tests, integration tests, and end-to-end tests, to validate code changes.

- Deploy: Deploys validated code changes to production, staging, or testing environments automatically, ensuring reliable and consistent deployments.

Overview of CI/CD Tools:

CI/CD tools, such as Jenkins, GitLab CI/CD, and GitHub Actions, provide features for automating build, test, and deployment workflows. They offer integration with version control systems, flexible configuration options, and scalability to support complex software projects.

5. Basic Cloud Deployment

Introduction to Cloud Computing:
Cloud computing involves the delivery of computing services, such as servers, storage, databases, networking, and software, over the internet. It offers advantages such as scalability, flexibility, and cost-effectiveness compared to traditional on-premises infrastructure.

Explanation of Basic Cloud Deployment Concepts:

Basic cloud deployment involves:

- Infrastructure as Code (IaC): Treating infrastructure configuration as code, allowing infrastructure to be managed, provisioned, and updated programmatically using scripts or configuration files.
- Server Provisioning: Creating and managing virtual servers, known as instances, on cloud platforms to host applications and services.

Overview of Cloud Service Providers:

Cloud service providers, such as AWS (Amazon Web Services), Azure (Microsoft Azure), and Google Cloud Platform, offer a wide range of services for deploying applications, managing infrastructure, and accessing cloud resources. They provide infrastructure services, platform services, and software services to support various use cases and workloads.

6. Benefits of CI/CD

Streamlining Development Processes:
CI/CD automates repetitive tasks, such as code integration, testing, and deployment, streamlining development processes and enabling faster delivery of software updates.

Enabling Faster Feedback Loops:

CI/CD facilitates rapid iteration cycles by providing fast feedback on code changes through automated tests, code reviews, and continuous integration, leading to shorter time-to-market and improved agility.

Improving Code Quality:

CI/CD ensures that code changes are thoroughly tested, validated, and reviewed before being deployed to production, resulting in higher code quality, reliability, and customer satisfaction.

Enhancing Collaboration and Visibility:

CI/CD promotes collaboration among development teams by providing shared workflows, automated processes, and real-time visibility into the status of code changes and deployments. It fosters a culture of transparency, accountability, and continuous improvement within organizations.