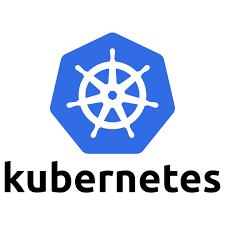
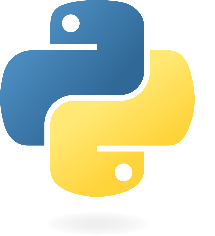


**End to End Python-based microservice application on AWS Elastic Kubernetes Service (EKS).**



1. **Kareem Mahdi Baz**
2. **Ayman Ahmed Elshafie**
3. **Mostafa Ali Mostafa**
4. **Salah Elhalaby**

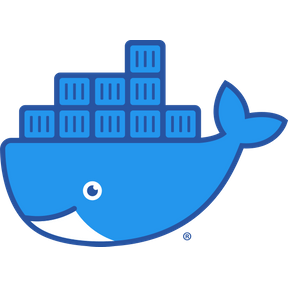


Table of Contents

[Python 3](#_Toc179841484)

[What is the Python language? 3](#_Toc179841485)

[Docker 3](#_Toc179841486)

[What is the Docker? 3](#_Toc179841487)

[Kubernetes 4](#_Toc179841488)

[What is the Kubernetes? 4](#_Toc179841489)

[What is the Cluster Architecture? 4](#_Toc179841490)

[What is the Control plane components? 4](#_Toc179841491)

[kube-apiserver 5](#_Toc179841492)

[etcd 5](#_Toc179841493)

[kube-scheduler 5](#_Toc179841494)

[kube-controller-manager 5](#_Toc179841495)

[cloud-controller-manager 6](#_Toc179841496)

[What Are the Node components? 6](#_Toc179841497)

[Kubelet 6](#_Toc179841498)

[kube-proxy 7](#_Toc179841499)

[Container runtime 7](#_Toc179841500)

[AWS EKS 8](#_Toc179841501)

[What is Amazon Web Services? 8](#_Toc179841502)

[AWS EKS 9](#_Toc179841503)

[Terraform 9](#_Toc179841504)

[What is the terraform? 9](#_Toc179841505)

[Ansible 10](#_Toc179841506)

[What is the Ansible? 10](#_Toc179841507)

[Jenkins 11](#_Toc179841508)

[What is the Jenkins? 11](#_Toc179841509)

[GitHub 12](#_Toc179841510)

[What is the GitHub? 12](#_Toc179841511)

[Project workflow: 12](#_Toc179841512)

[Continuous Delivery/Deployment (CD): 12](#_Toc179841513)

[Docker 13](#_Toc179841514)

[K8s 13](#_Toc179841515)

[Infrastructure as Code (IaC) 15](#_Toc179841516)

[Terraform 15](#_Toc179841517)

[Ansible 15](#_Toc179841518)

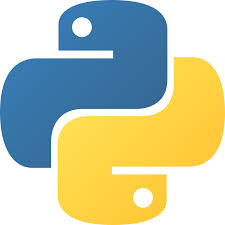
[Continuous Integration (CI) 15](#_Toc179841519)

[AWS 16](#_Toc179841520)

[GitHub 16](#_Toc179841521)

[Conclusion 16](#_Toc179841522)

# Python



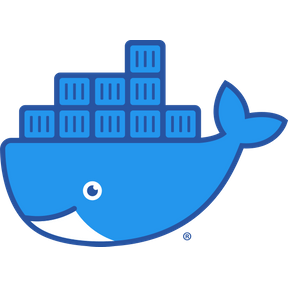
## What is the Python language?

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

# Docker

## What is the Docker?



Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers. The service has both free and premium tiers. The software that hosts the containers is called Docker Engine. It was first released in 2013 and is developed by Docker, Inc.

Docker is a tool that is used to automate the deployment of applications in lightweight containers so that applications can work efficiently in different environments in isolation.

# Kubernetes

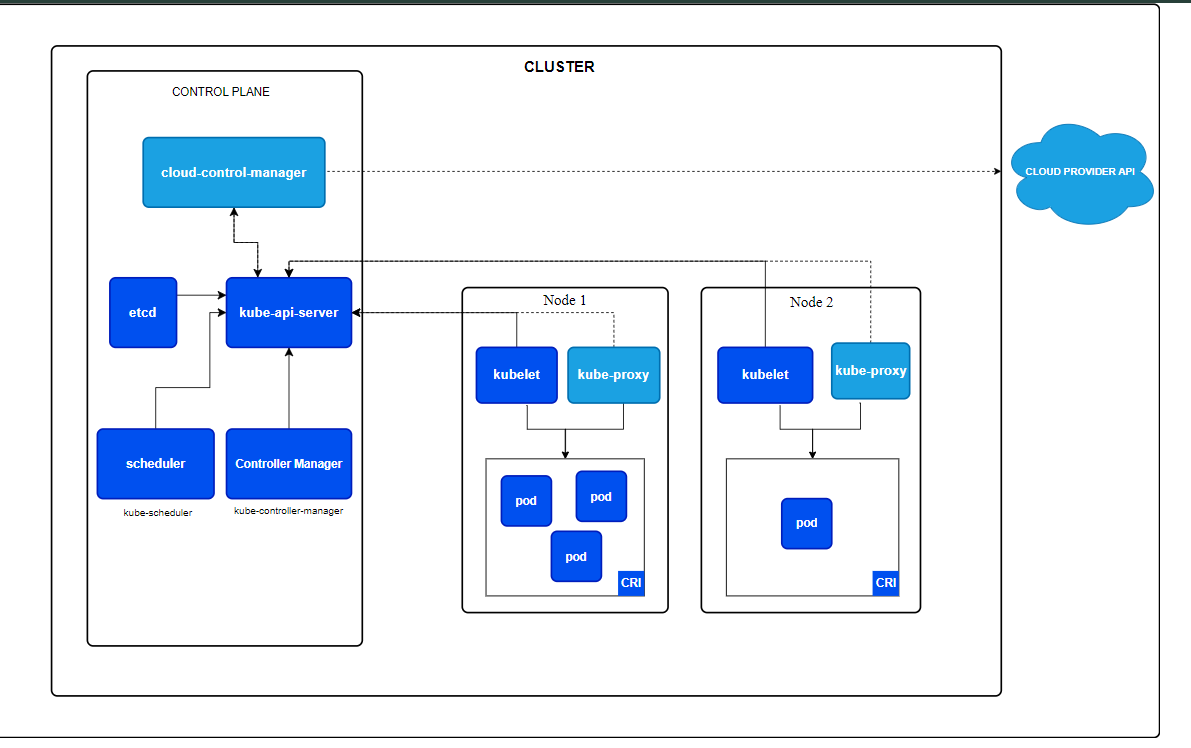
## What is the Kubernetes?

Kubernetes (K8S) is an open-source container orchestration system for automating software deployment, scaling, and management.Originally designed by Google, the project is now maintained by a worldwide community of contributors, and the trademark is held by the Cloud Native Computing Foundation.

## What is the Cluster Architecture?

A Kubernetes cluster consists of a control plane plus a set of worker machines, called nodes, that run containerized applications. Every cluster needs at least one worker node in order to run Pods.

The worker node(s) host the Pods that are the components of the application workload. The control plane manages the worker nodes and the Pods in the cluster. In production environments, the control plane usually runs across multiple computers and a cluster usually runs multiple nodes, providing fault-tolerance and high availability.



## What is the Control plane components?

The control plane's components make global decisions about the cluster (for example, scheduling), as well as detecting and responding to cluster events (for example, starting up a new pod when a Deployment's replicas field is unsatisfied).

Control plane components can be run on any machine in the cluster. However, for simplicity, setup scripts typically start all control plane components on the same machine, and do not run user containers on this machine. See Creating Highly Available clusters with kubeadm for an example control plane setup that runs across multiple machines.

### kube-apiserver

The API server is a component of the Kubernetes control plane that exposes the Kubernetes API. The API server is the front end for the Kubernetes control plane.

The main implementation of a Kubernetes API server is kube-apiserver. kube-apiserver is designed to scale horizontally—that is, it scales by deploying more instances. You can run several instances of kube-apiserver and balance traffic between those instances.

### etcd

Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data.

If your Kubernetes cluster uses etcd as its backing store, make sure you have a back up plan for the data.

You can find in-depth information about etcd in the official documentation.

### kube-scheduler

Control plane component that watches for newly created Pods with no assigned node, and selects a node for them to run on.

Factors taken into account for scheduling decisions include: individual and collective resource requirements, hardware/software/policy constraints, affinity and anti-affinity specifications, data locality, inter-workload interference, and deadlines.

### kube-controller-manager

Control plane component that runs controller processes.

Logically, each controller is a separate process, but to reduce complexity, they are all compiled into a single binary and run in a single process.

There are many different types of controllers. Some examples of them are:

Node controller: Responsible for noticing and responding when nodes go down.

Job controller: Watches for Job objects that represent one-off tasks, then creates Pods to run those tasks to completion.

EndpointSlice controller: Populates EndpointSlice objects (to provide a link between Services and Pods).

ServiceAccount controller: Create default ServiceAccounts for new namespaces.

The above is not an exhaustive list.

### cloud-controller-manager

A Kubernetes control plane component that embeds cloud-specific control logic. The cloud controller manager lets you link your cluster into your cloud provider's API, and separates out the components that interact with that cloud platform from components that only interact with your cluster.

The cloud-controller-manager only runs controllers that are specific to your cloud provider. If you are running Kubernetes on your own premises, or in a learning environment inside your own PC, the cluster does not have a cloud controller manager.

As with the kube-controller-manager, the cloud-controller-manager combines several logically independent control loops into a single binary that you run as a single process. You can scale horizontally (run more than one copy) to improve performance or to help tolerate failures.

The following controllers can have cloud provider dependencies:

* Node controller: For checking the cloud provider to determine if a node has been deleted in the cloud after it stops responding
* Route controller: For setting up routes in the underlying cloud infrastructure
* Service controller: For creating, updating and deleting cloud provider load balancers

## What Are the Node components?

Node components run on every node, maintaining running pods and providing the Kubernetes runtime environment.

### Kubelet

An agent that runs on each node in the cluster. It makes sure that containers are running in a Pod.

The kubelet takes a set of PodSpecs that are provided through various mechanisms and ensures that the containers described in those PodSpecs are running and healthy. The kubelet doesn't manage containers which were not created by Kubernetes.

### kube-proxy

kube-proxy is a network proxy that runs on each node in your cluster, implementing part of the Kubernetes Service concept.

kube-proxy maintains network rules on nodes. These network rules allow network communication to your Pods from network sessions inside or outside of your cluster.

kube-proxy uses the operating system packet filtering layer if there is one and it's available. Otherwise, kube-proxy forwards the traffic itself.

If you use a network plugin that implements packet forwarding for Services by itself, and providing equivalent behavior to kube-proxy, then you do not need to run kube-proxy on the nodes in your cluster.

### Container runtime

A fundamental component that empowers Kubernetes to run containers effectively. It is responsible for managing the execution and lifecycle of containers within the Kubernetes environment.

Kubernetes supports container runtimes such as containerd, CRI-O, and any other implementation of the Kubernetes CRI (Container Runtime Interface)

# AWS EKS

## What is Amazon Web Services?



Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered, pay-as-you-go basis. Clients will often use this in combination with autoscaling (a process that allows a client to use more computing in times of high application usage, and then scale down to reduce costs when there is less traffic). These cloud computing web services provide various services related to networking, compute, storage, middleware, IoT and other processing capacity, as well as software tools via AWS server farms. This frees clients from managing, scaling, and patching hardware and operating systems. One of the foundational services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, with extremely high availability, which can be interacted with over the internet via REST APIs, a CLI or the AWS console. AWS's virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard-disk (HDD)/SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM).

AWS services are delivered to customers via a network of AWS server farms located throughout the world. Fees are based on a combination of usage (known as a "Pay-as-you-go" model), hardware, operating system, software, and networking features chosen by the subscriber requiring various degrees of availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either. Amazon provides select portions of security for subscribers (e.g. physical security of the data centers) while other aspects of security are the responsibility of the subscriber (e.g. account management, vulnerability scanning, patching). AWS operates from many global geographical regions including seven in North America

## AWS EKS

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed service that eliminates the need to install, operate, and maintain your own Kubernetes control plane on Amazon Web Services (AWS).

# Terraform

## What is the terraform?

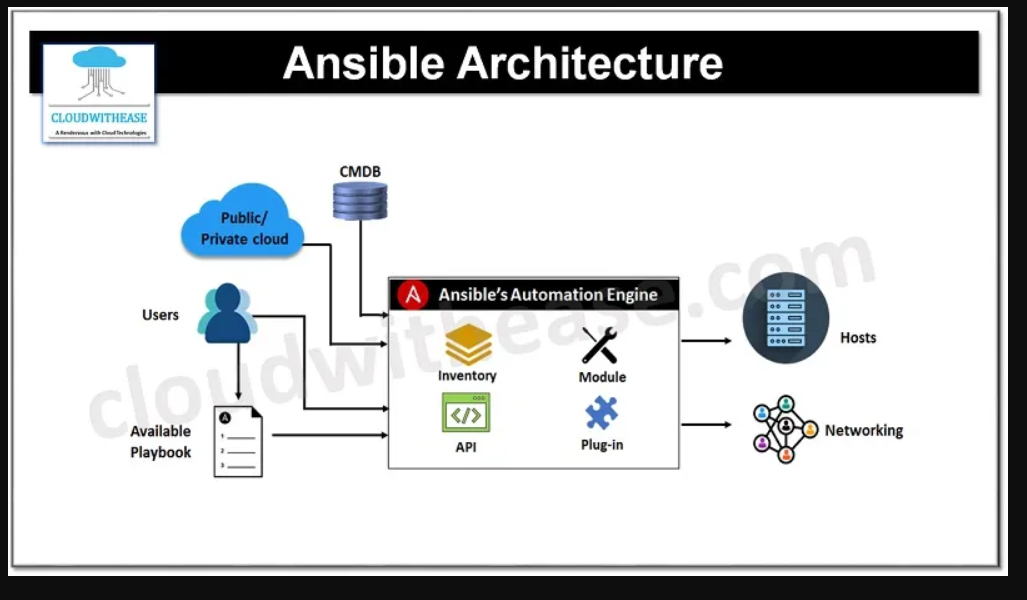
Terraform manages external resources (such as public cloud infrastructure, private cloud infrastructure, network appliances, software as a service, and platform as a service) with "providers". HashiCorp maintains an extensive list of official providers, and can also integrate with community-developed providers. Users can interact with Terraform providers by declaring resources or by calling data sources. Rather than using imperative commands to provision resources, Terraform uses declarative configuration to describe the desired final state. Once a user invokes Terraform on a given resource, Terraform will perform CRUD actions on the user's behalf to accomplish the desired state. The infrastructure as code can be written as modules, promoting reusability and maintainability.

Terraform supports a number of cloud infrastructure providers such as Amazon Web Services, Cloudflare, Microsoft Azure, IBM Cloud, Serverspace, Selectel Google Cloud Platform, DigitalOcean, Oracle Cloud Infrastructure, Yandex.Cloud, VMware vSphere, and OpenStack.

HashiCorp maintains a Terraform Module Registry, launched in 2017. In 2019, Terraform introduced the paid version called Terraform Enterprise for larger organizations

# Ansible

## What is the Ansible?



Ansible is a suite of software tools that enables infrastructure as code. It is open-source and the suite includes software provisioning, configuration management, and application deployment functionality.

Originally written by Michael DeHaan in 2012, and acquired by Red Hat in 2015, Ansible is designed to configure both Unix-like systems and Microsoft Windows. Ansible is agentless, relying on temporary remote connections via SSH or Windows Remote Management which allows PowerShell execution. The Ansible control node runs on most Unix-like systems that are able to run Python, including Windows with Windows Subsystem for Linux installed. System configuration is defined in part by using its own declarative language.

# Jenkins

## What is the Jenkins?

Jenkins is an open-source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration, and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat. It supports version control tools, including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, ClearCase, and RTC, and can execute Apache Ant, Apache Maven, and sbt based projects as well as arbitrary shell scripts and Windows batch commands.

# GitHub

## What is the GitHub?

GitHub is a developer platform that allows developers to create, store, manage and share their code. It uses Git software, providing the distributed version control of access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project.Headquartered in California, it has been a subsidiary of Microsoft since 2018.

It is commonly used to host open source software development projects. As of January 2023, GitHub reported having over 100 million developers and more than 420 million repositories, including at least 28 million public repositories. It is the world's largest source code host as of June 2023.

# Project workflow:

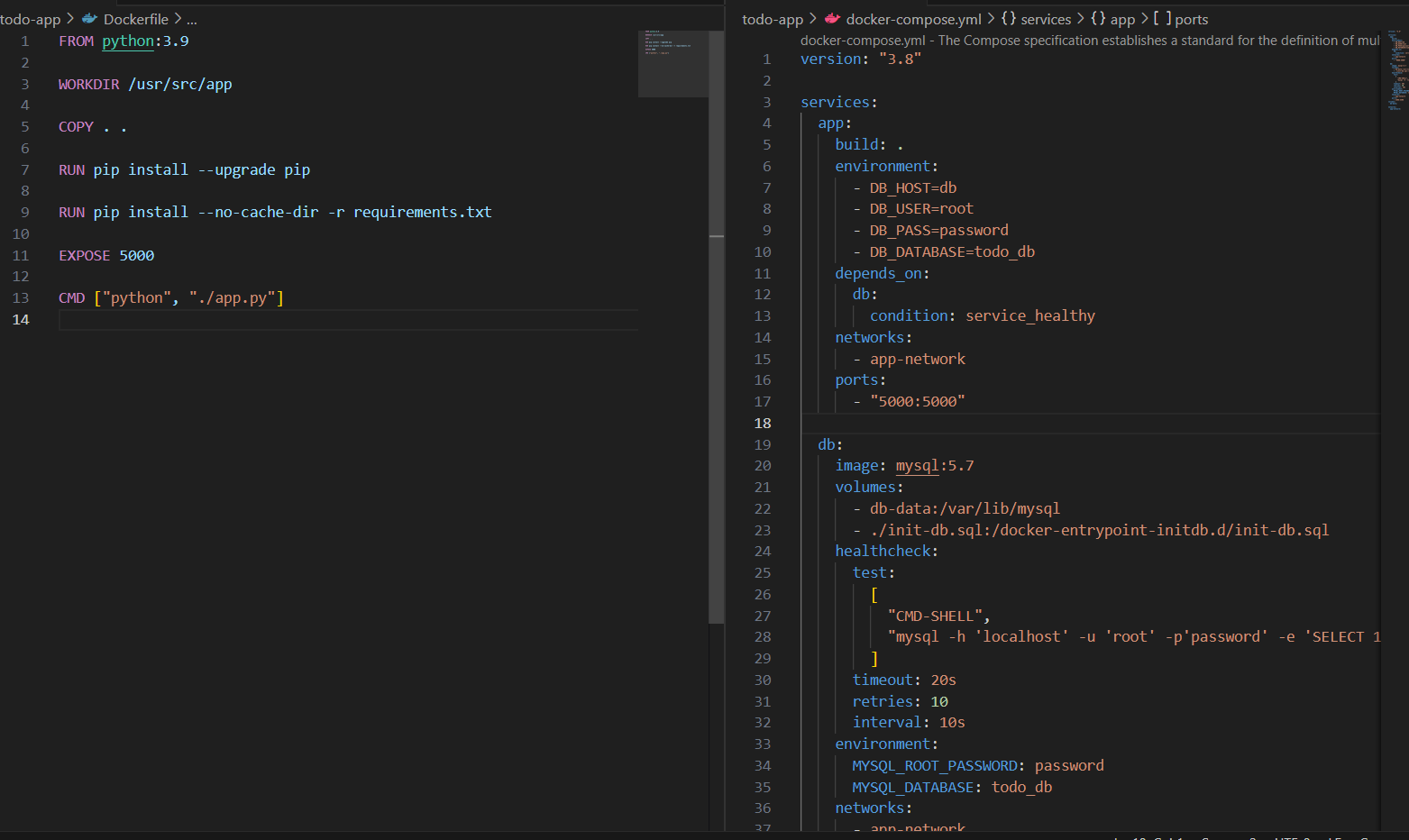
This project aims to streamline the CI/CD pipeline for a mid-sized e-commerce platform by integrating automated testing, continuous deployment, and monitoring solutions. The goal is to reduce downtime, ensure faster release cycles, and improve overall system stability.

I talk about the most important steps and topics in the project:

## Continuous Delivery/Deployment (CD):

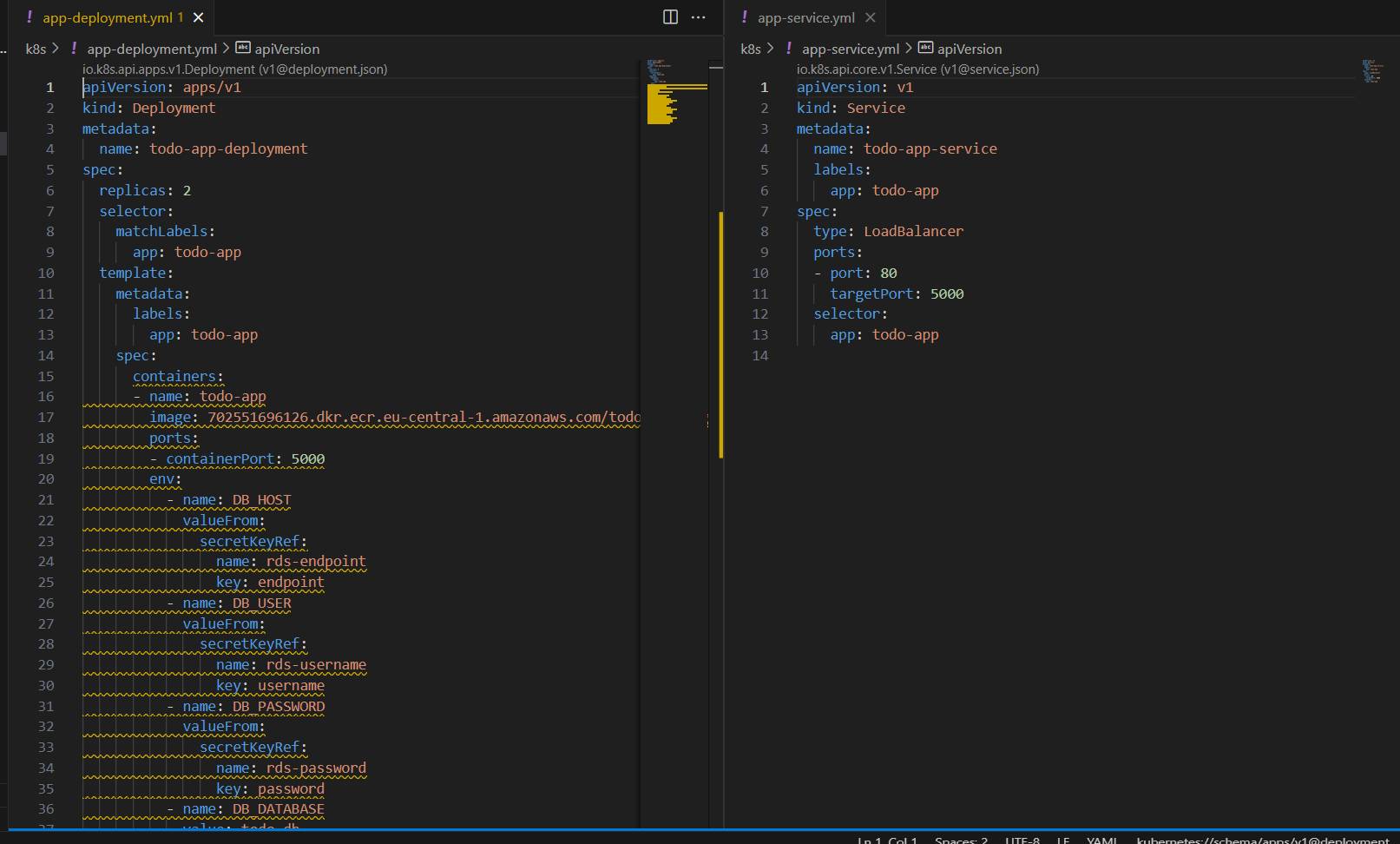
Here the webside project was written for the Python language and databases using MYSQL.

## Docker

The first stage begins, which is Docker, the code is Tested to the website, and after that, Docker is used to write the special Dockerfile for the web side, and from here Image is done , The next step is that we write Docker Configure, so that more than one integration can be written together, so that the databases and the frontend are connected to each other, and the communication between them is sequential.

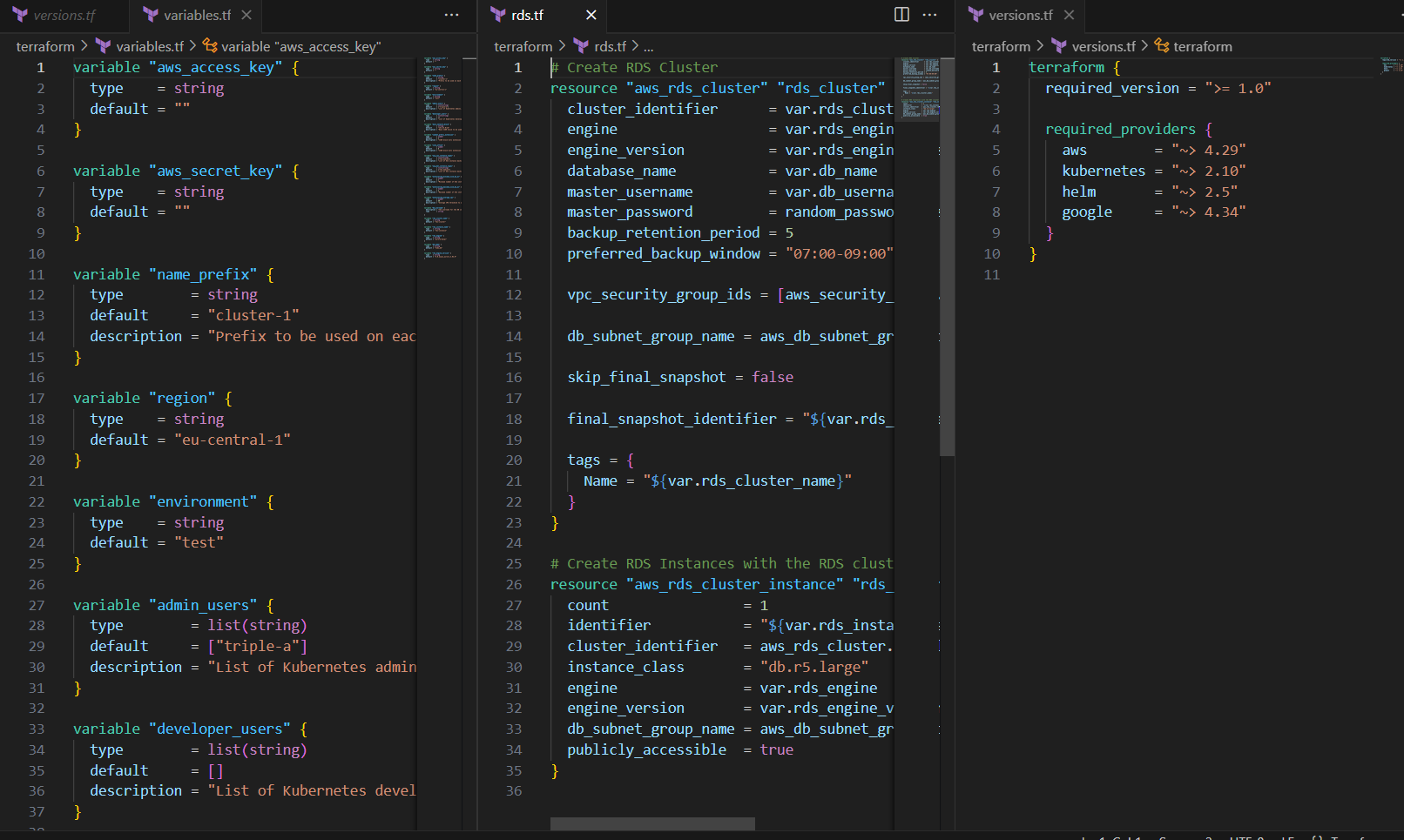
## K8s

Here comes the K8s stage:-

Here is the responsibility if there is a problem with the container or it occurs due to pressure on it, or a technical malfunction is solved in the code, and here the container cannot perform the tasks it is responsible for. K8s can solve this problem in more than one way, which is to do remove to container and build again.It can make sure that the system does not fall through deployment so that he does not lose his company. It also makes sure that the service is running and that each pod is connected to the pod in another namespace.

## Infrastructure as Code (IaC)

### Terraform

It means that I can control my project remotely without accessing the AWS website, and I can follow the project, troubleshoot any problem that occurs through it, and create and build the project.

## Ansible

It means that I can control my project remotely without accessing the AWS website, and I can follow the project, trablshoot any problem that occurs through it, and create and build the project.

The difference between Ansible and Terraform is that Terraform is more distinguished in the work of creating, building, controlling the project, and following up on all the problems and errors that occur in it.

Ansible :It has more control over installing and downloading any package on the container

## Continuous Integration (CI)

Jenkins

It consists of two parts:

1. CI: It is responsible for identifying any update to the image and then build it, push the container on the ECR
2. CD: automates the release process to ensure that code changes are delivered to production or a staging environment in a reliable and low-risk manner.

## AWS

AWS is a cloud platform that provides a wide range of services, including computing power, storage, and deployment tools, all of which are crucial for a modern DevOps workflow. In this project, AWS was utilized to host applications, manage infrastructure, and automate the CI/CD pipeline.

## GitHub

GitHub is a cloud-based platform that facilitates version control using Git. It allows multiple developers to collaborate on code

We upload the project files to it and control and modify the files. You can create more than one version of the project.

# Conclusion

This Project’s aim is to cover all what we learned during the initiative as the period was very fruitful as we learned many things.