WideBot Task

What is the DevOps ?

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization’s ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

Description the Problem :

Need to provide the application and service to the user and making sure that it is available at all times and working effectively on it, and securing the Data Base.

Objectives of the task :

Provisioning Infrastructure to the Web Applications , Databases (MongoDB, SQL Server) , Redis for caching, with Domain Name, Certificate, and Load Balancer . And I've done that in two solutions .

Solutions Delivery:

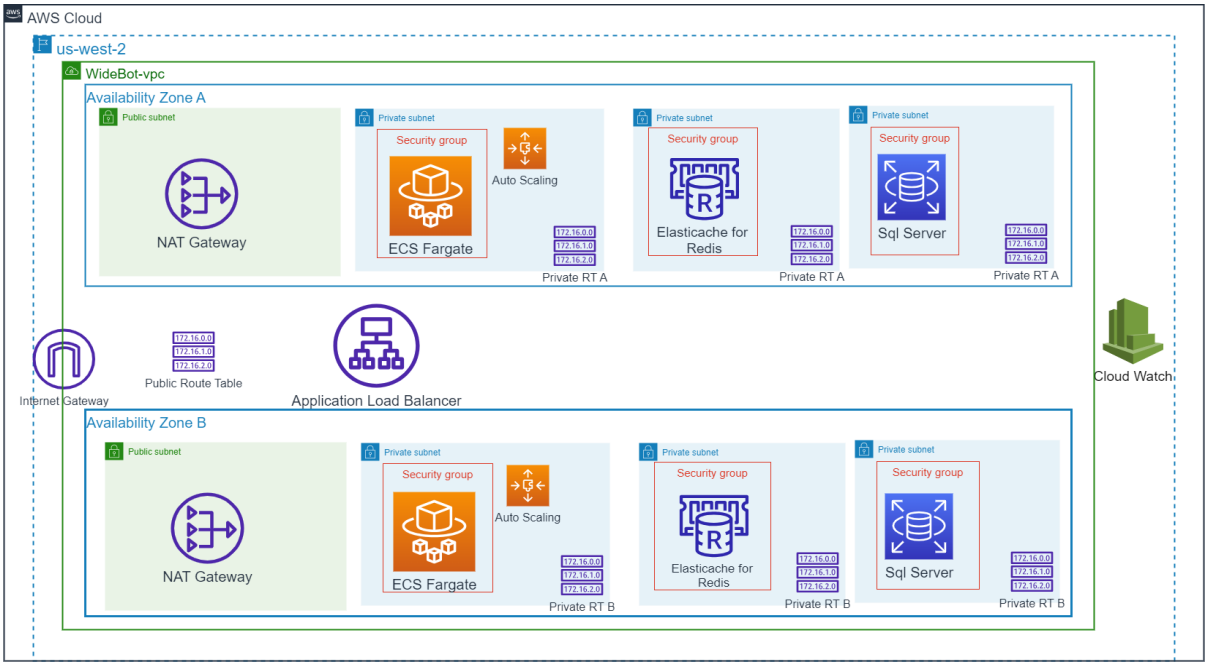
https://github.com/alimostafa095/witebot/tree/master

First solution

- **Description :**

  write a terraform template to provision the infrastructure for a web application and its associated databases (Sql server), and Redis for caching.
  configure Load Balancer to availability and auto scale to ensure scalability .
  Use AWS EKS to let up a replication controller to run pods that are accessed as services on Docker.

- **Design :**



- **Resources is used :-**

1. 2 Availability Zone to make sure the environment is high available.
2. 8 subnets (2 public and 6 private) the public subnet to create NAT-GW and private subnet for each service .
3. ECS cluster in 2 private subnets in different AZ , the ecs is high performance container management service that supports docker containers and allows you to easily run applications on a managed cluster of Amazon Elastic Compute Cloud (Amazon EC2) instances .
4. aws elastic cache for redis cluster with characteristics (node type = cache.t2.micro , engine = memcached)
5. Sql server over RDS DB with characteristics (engine = sqlserver , instance_class = db.t3.medium).
6. Application load balancer to distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones. This increases the availability of your application.
7. Internet Gateway to enable communication between your public subnets and the internet .
8. CloudWatch enables you to monitor your complete stack (applications, infrastructure, network, and services) Create security groups to control incoming and outgoing traffic .
9. Create Public and Private route tables and associate it to the subnets .

- **Solution source Code :-**

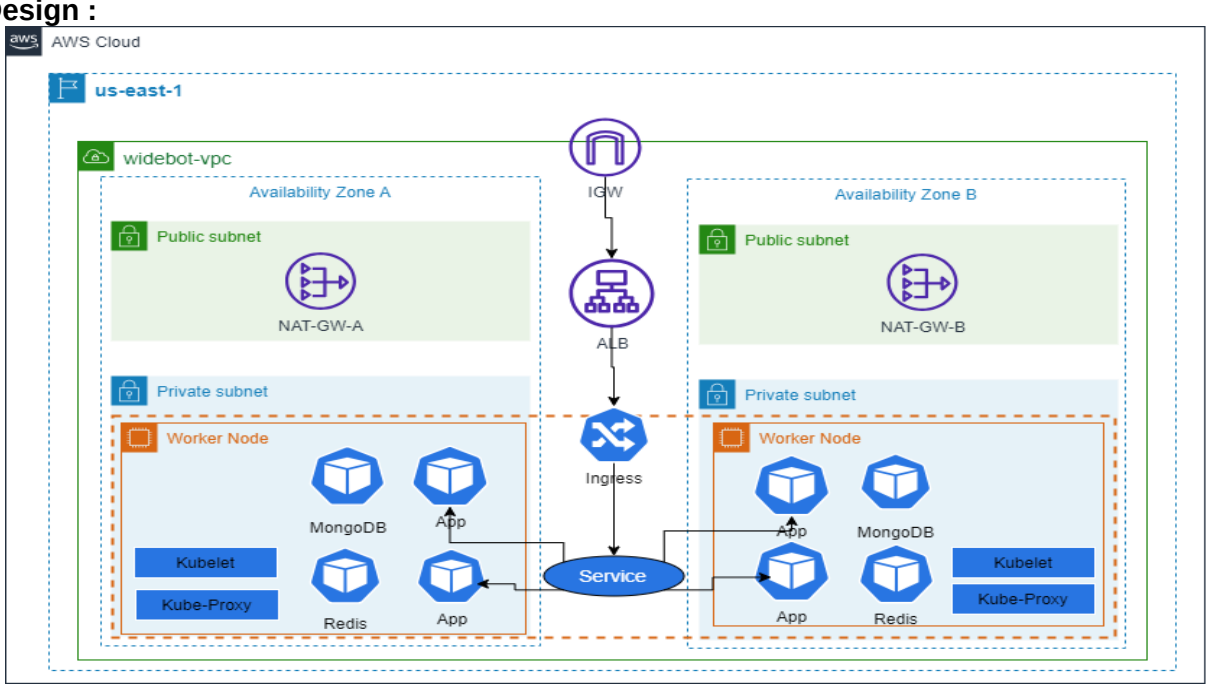
  https://github.com/alimostafa095/witebot/tree/master/first%20solution

Second Solution :-

- **Description :**

  write a terraform template to provision the infrastructure for a web application and its associated databases (MongoDB), and Redis for caching. configure Load Balancer to availability and auto scale to ensure scalability .
  Use AWS EKS to let up a replication controller to run pods that are accessed as services on Docker.

- **Design :**



- **Resources is used :**

1. 2 Availability Zone to make sure the environment is high available.
2. 4 subnets (2 public and 2 private) the public subnet to create a NAT-GW and private subnet for the EKS cluster .
3. Build Amazon Elastic Kubernetes Service (Amazon EKS) is a managed Kubernetes service that makes it easy for you to run Kubernetes on AWS and on-premises. Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.
4. Application load balancer to distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones, and communicate with Ingress service ,This increases the availability of your application.
5. Internet Gateway to enable communication between your public subnets and the internet
6. Deploy the application in Kubernetes using eks deployment (is a managed service that you can use to run Kubernetes on AWS without needing to install, operate, and maintain your own Kubernetes control plane or nodes) .
7. Deploy Mongo data base in Kubernetes. Using MongoDB inside your Kubernetes opens a new world where you can run a tiny NoSQL database and get all the power from the MongoDB database hosted inside your Kubernetes cluster.
8. Deploy elastic cache for redis in kubernetes .
9. Deploy Ingress service (the ingress controller works inside the AWS EKS by “grouping” the ingress resources under a single name, making them accessible and routable from a single AWS Application Load Balancer).

- **Solution source code :**

  https://github.com/alimostafa095/witebot/tree/master/second%20solution