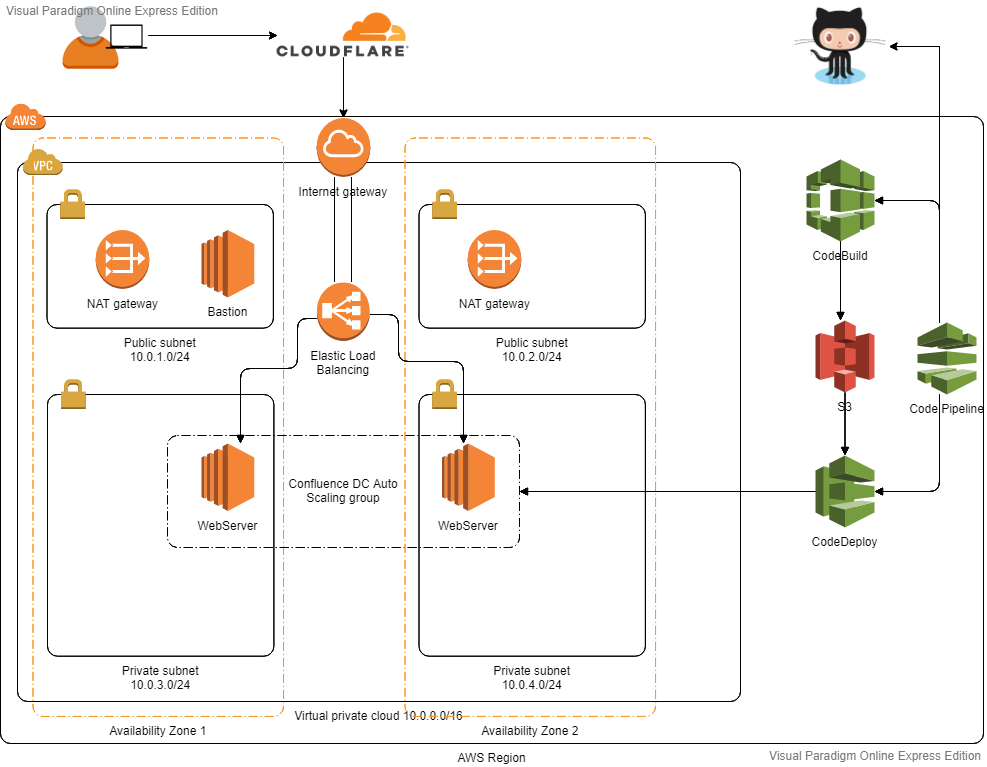
Juan Pablo Camacho Sequeira.

Bellow the answer for the Acceptance Criteria.

* Architecture diagram demonstrating the planned solution



1. At the Front of the application, we have a CDN (Cloudflare). This CDN will in charge of the following points.
   1. Is in charge of the application cache
   2. HTTP to https redirect.
   3. Certificate management
   4. DDOS attacks
2. Then the VPC
   1. Internet GW attached to the VPC.
   2. 2 Public Subnets with the Internet Gateway in the Routing Table. In this Public Subnet, we have the Bastion Server.
   3. 2 Private Subnets with the NAT Getaway Attached in the Routing Table. In the Private Network, we have web servers.
   4. Then we have the Internet Gateway.
3. CI/CD.
   1. Using Code Pipeline
   2. Github
   3. CodeBuild
   4. Code Deploy

* Timeoff-management fork on the local repository (Github)
  + This is the Github Fork <https://github.com/juanpcamacho/application>
* Required Infrastructure running on Cloud of preference, provisioned using some sort of infrastructure as a code solution.
  + All environment described in the image is set up with Terraform. This is the terraform URL. <https://github.com/juanpcamacho/application/blob/master/gorilla.tf>
* The application must be deployed using a fully automated CI solution. Triggered by change source control.
  + Application deployment is performed with Cloud Native CI/CD tools such as Code Pipeline 🡪 GitHub 🡪 Code Build 🡪 Code Deploy
  + The CI/CD is triggered by any change in GitHub.
  + I select Cloud CI/CD because is more native to do the CD and can perform the same tasks as Jenkins.
  + As well you need a Jenkins instance available 24/7 instead the AWS services are auto scalable and you need to pay for usage.
  + As well you can choose the deployment method between Blue/Green and In-Place.
  + Represented in the diagram.
  + In the github can found the files buildspec.yml with all the steps for the build and appsec.yml for the deployment.
  + Deployment of the Pipeline created as well with Terraform – (Pending integrate with the Github webhook because is not possible in personal repos. <https://www.terraform.io/docs/providers/github/r/repository_webhook.html>)
* The application must be secured from external access and application should be serving via standard HTTP and https protocols.
  + All security groups just have the needed ports open
  + The WebServers are located in an Internal Subnet so only through Load Balancer can be accessed Or from the bastion.
  + The CDN is in charge of the port forwarding from HTTP 🡪 https.
  + As well have a certificate installed.

URL: <https://gorilla.camachotec.club/>

Note: If DNS issue is present can be because for cost saving I destroy all at the end of the process and need to upload with the new ELB DNS everytime we start from scratch.

* The application should be Highly Available and Load Balancer.
  + The application has been implemented in 4 different subnets in different Availability Zone.
  + As well we have a Load Balancer for the high availability.
  + I setup autoscaling to increase the number of machines depends on the load.
  + CDN gives us another High Available Cache option for the application.