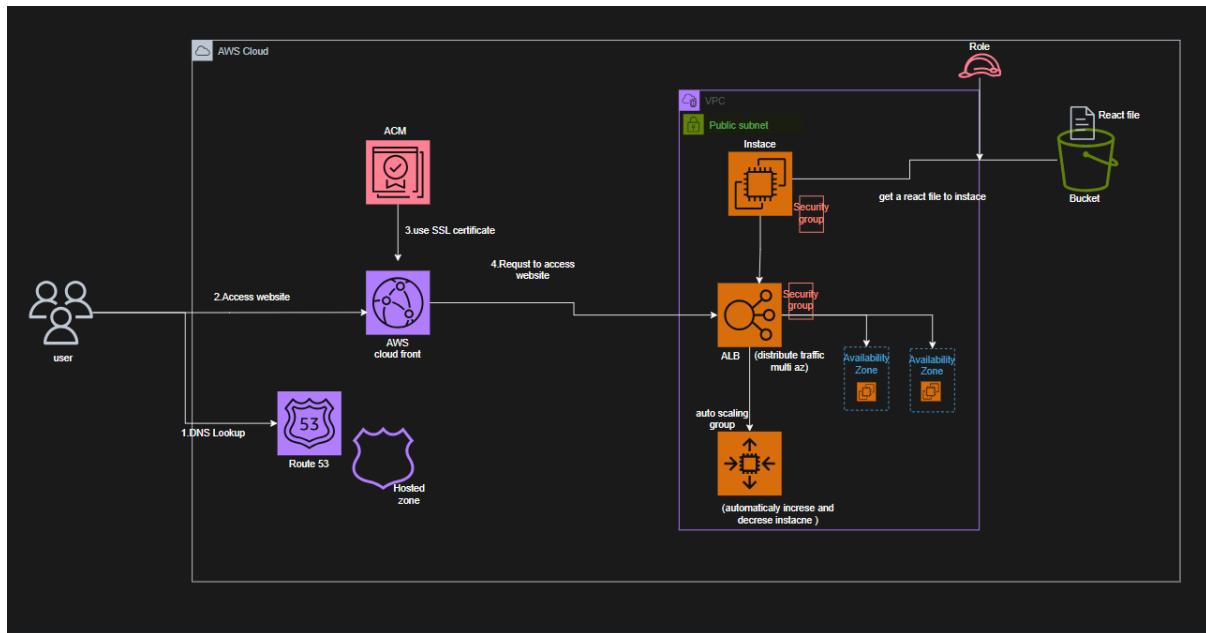


# UI (Frontend) Architecture Design



- Step 1: Go to aws certificate manager and get a SSL certificate [www.xyz.com](http://www.xyz.com).
- Step 2: Go to aws Route53 create a hosted zone get a SSL certificate.
- Step 3: Create s3 bucket disable block all public access name with domain name upload a react file
- Step 4: enable static web site to s3 bucket and give a index file name
- Step 5: Create a cloud front distribution add a domain name  
Enable web application firewall
- Step 6: Go to GODADDY edit a value to get a record name

OR



Step 1: Go to aws certificate manager and get a SSL certificate

[www.xyz.com](http://www.xyz.com).

step 2: Go to aws Route53 create a hosted zone get a SSL certificate.

Step 3:create vpc 2 az and public subnet

Step 4: create a role for access s3 to ec2 instance

Step 5: create ec2 instance with this new vpc ,create security group

(22,443,80) and add a role

Step 6: create Target group for Application load balancer (sg 443)

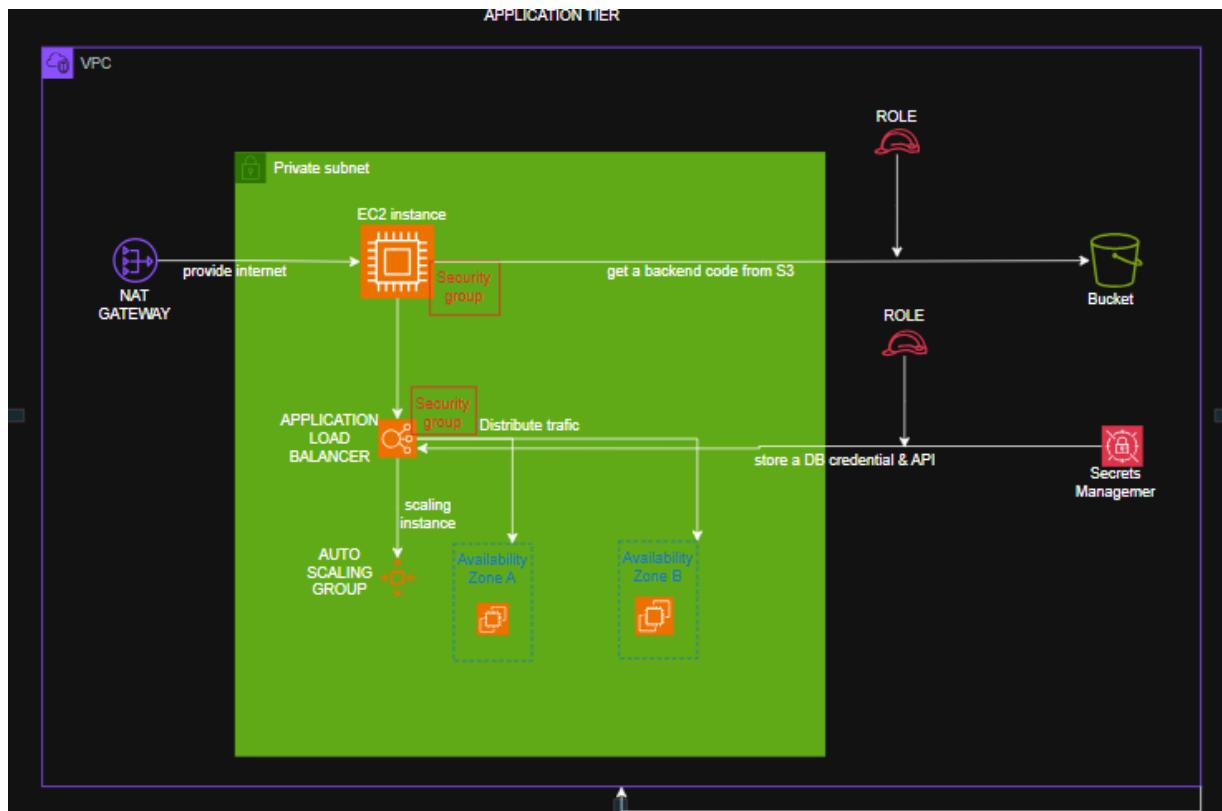
Step 7: get a ami of this image to launch template -> create auto scaling and set a throusold

Step 8:create cloud front distribution and add a origin type load balancer to our ALB and enable WAF.

Step 9: Go to GODADDY edit a value to get a record name.

## 2.API (Backend) Architecture Design

## 2.API (Backend) Architecture Design



Step 1: create VPC with 2 Az and 2 private subnet give a static IP and NAT gateway for internet.

Step 2: create iam role access s3 bucket to ec2 instance.  
get a backend code from S3 bucket.

Step 3: Create ec2 instance with that private subnet and SG(22,allow web tire)

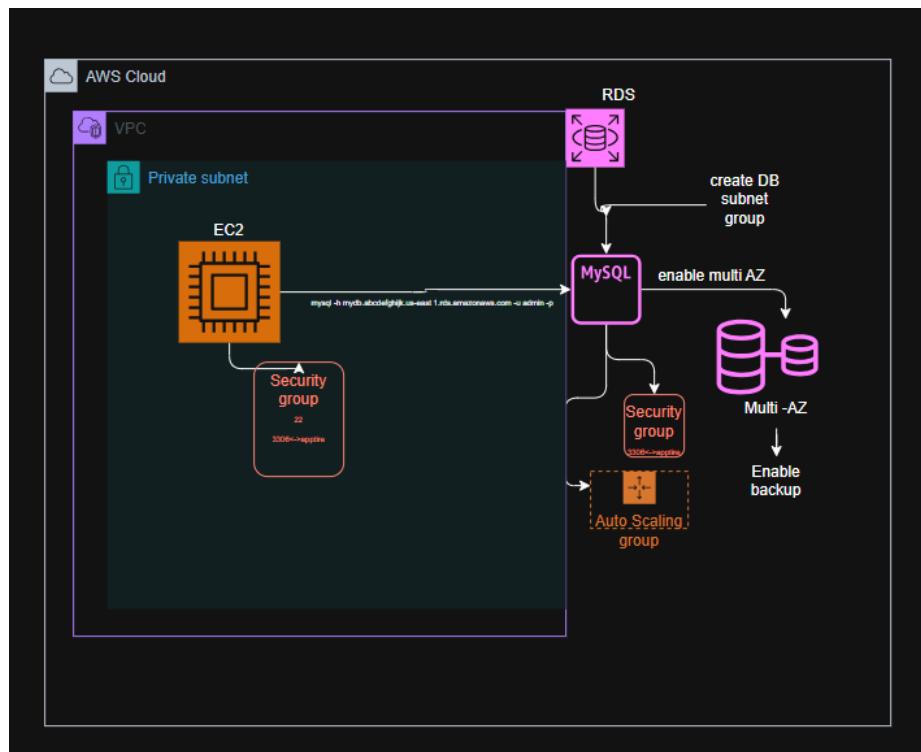
Step 4: create Target group for Application load balancer (SG allow web tire)

Step 5: get a AMI of this image to launch template -> create auto scaling group and set a threshold

Step 6:Create AWS secret manager -> to store a DB credentials, API keys . need iam role (`secretsmanager:GetSecretValue ssm:GetParameter`)

Step 7:enter this to instance to connect my sql (`mysql -h mydb.abcdefgijk.us-east-1.rds.amazonaws.com -u admin -p`)

### 3.Database Architecture Design



**Step 1:** Create ec2 instance to 1pvt subnet with

SG (22,3306->app tire).

**Step 2:** Go to RDS create DB subnet group.

**Step 3:** select my sql ,select version, select multi-AZ deployment.

**Step 4:** Create DB name , create master name and password.

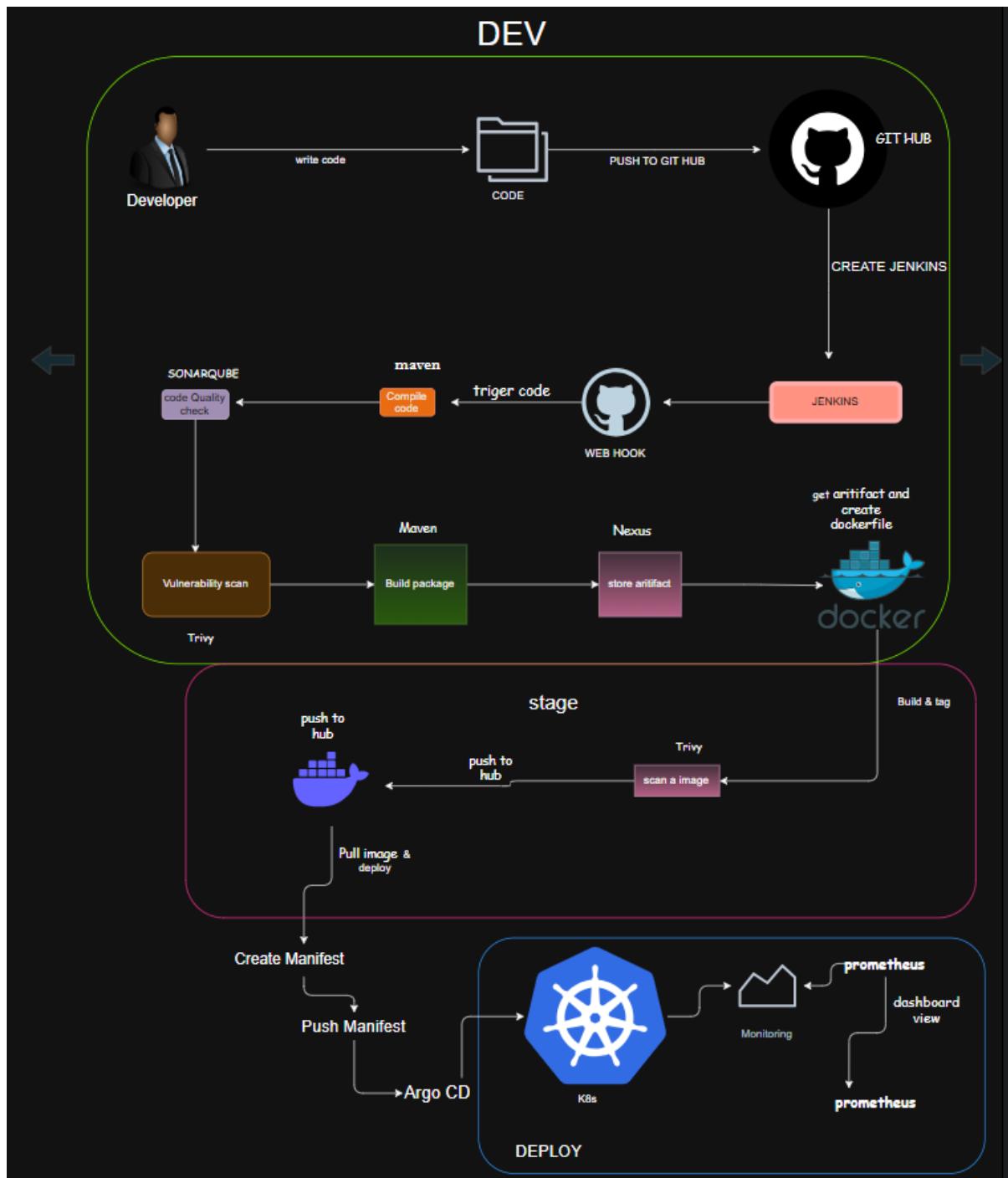
**Step 5:** select storage and check auto scaling is enable.

**Step 6:** select Connect to an EC2 compute resource and select instance.

**Step 7:** add subnet name and SG(3306->APP tire), select AZ

**Step 8:** enable monitoring

## CI/CD PIPELINE



- step 1: pull a code form GIT HUB
- step 2: set a web hook to GIT
- step 3: to bulid the code -> maven
- step 4: check the quality of this code -> sonarqube
- step 5: scan the code -> trivy

step 6: build a package

step 7: store the aritifat -> nexus

step 8: create a dockerfile and buld :tag

step 9:scan the image -> trivy

step 10:push the image to hub -> Docker hub

step 11:pull the image and create Manifest repo then Push Manifest

Step12:ARGO CD Detect manifest change if it detect a chage to  
deploy the chage to k8s

step 13: monitor the deploy machine also monitor the Jenkins  
machine