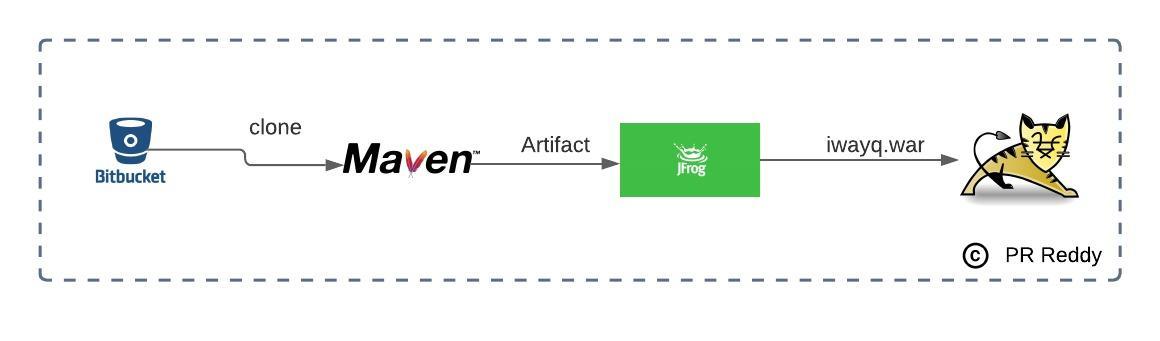
**Deploy Java Spring boot monolithic iwayQ login application**

**Pre-requisites:**

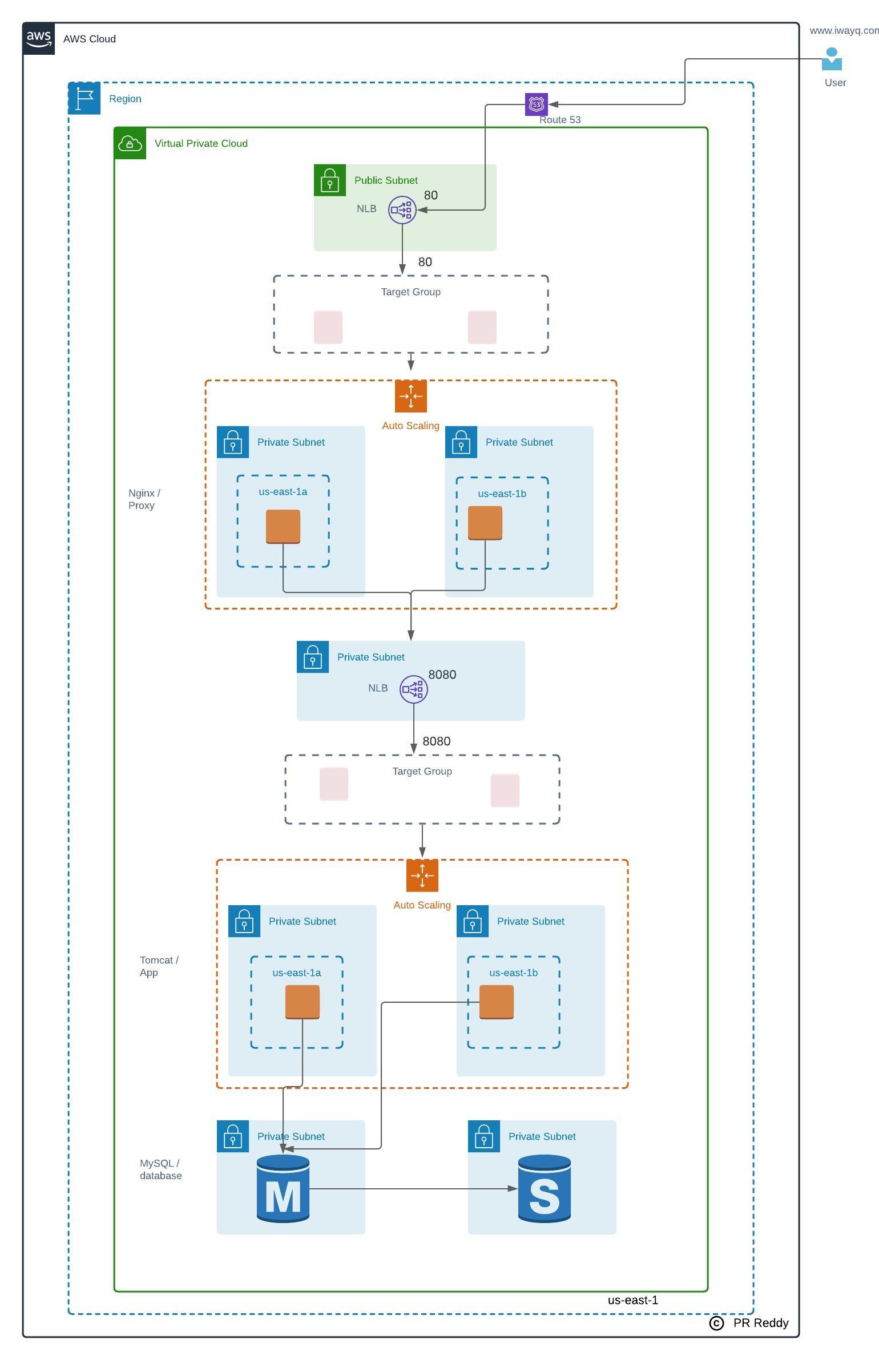
1. Login to AWS management console.
2. Create S3 bucket to store terraform state files
3. Create an IAM user with programmatic access with Administrator Policy attached.
4. Create Account in Jfrog cloud and create Maven repo
5. Create EC2 server manually and configure as Maven Build server
6. Create Golden AMI for Nginx
7. Create Golden AMI for Apache Tomcat
8. Assume dependency resources as Architecture can not include all components.

**Deployment:**  
Project deployment is divided into two steps.



**Application**

1. Clone bitbucket repository to Build Server  
   1.1 Repo: <https://iwayqtech@bitbucket.org/iwayqtech/javaloginapp.git>
2. Build the source  
   2.1 mvn package
3. Integrate Maven with Jfrog Artifactory Repo
4. deploy artifacts to Apache maven  
   3.1 mvn deploy
5. Verify that the Artifacts are pushed to Jfrog Repo.



**Deploy Network**

1. Deploy VPC
2. Write a terraform module to deploy the VPC architecture shown.
3. Deploy VPC architecture - Choose any IP range of your choice. Below resources need to be deployed.  
   3.1 VPC  
   3.2 Internet Gateway  
   3.3 Public Subnets in 1a & 1b zones  
   3.4 Private Subnets in 1a & 1b zones  
   3.5 Private Route table  
   3.6 Public Route table

**Deploy Route53 hosted zone**

1. Write terraform to deploy Route53 hosted zone
2. If you do not have an authoritative domain then take any xyz.com just to complete the terraform.
3. Below resources need to be deployed  
   3.1 Hosted zone

**Deploy compute resources to host high available monolithic spring boot web application**

1. Write terraform module to deploy compute resources (Proxy, App, database layers )for web application deployment as per the architecture shown into the existing VPC(created in step# 1)  
   Below resources need to be created for Proxy and App tiers  
   3.1 Auto Scaling Group for nginx & tomcat  
   3.2 Application Load Balancer for nginx & tomcat  
   3.3 Target Groups for both Load Balancers  
   3.2 Launch Configuration with userdata for nginx & tomcat  
   3.3 Security Group (allow 80 & 22 ports) for nginx  
   3.4 Security Group (allow 8080 & 22 ) for tomcat  
   3.5 Security Group for RDS.  
   3.5 Multi AZ RDS instance with database and table structure Employee.  
   3.4 Internet facing Network Load Balancer  
   3.5 Target Group for nginx & tomcat  
   3.6 Alias Record in Route 53 hosted zone.  
   3.7 IAM Instance Profile(IAM Role) - attach to both launch configurations  
   3.8 Below policies need to be added to the IAM Instance Profile (IAM Role)  
   i) AmazonEC2forSSM

**Verification**

1. Verify if you able to login to the EC2 instances using session manager
2. Verify EC2 instances are healthy in the target group
3. Verify web application is accessible

**Validation:**

1. Create account in bitbucket.org
2. Login to bitbucket and create a private repository to store the terraform code. Example naming standard for the two repositories.  
   terraform-vpc-base  
   terraform-login-app