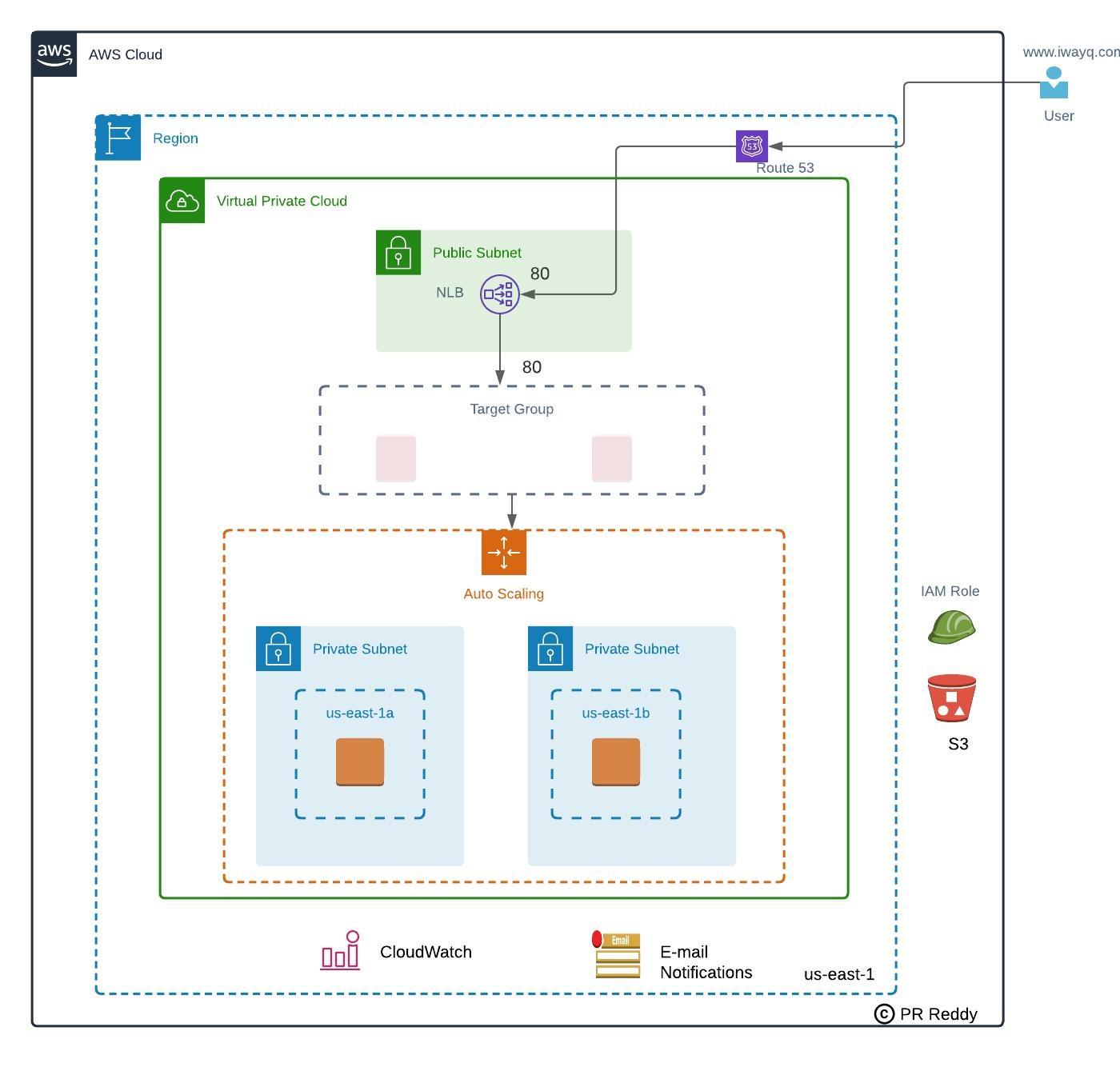
**Deploy Scalable, High Available, and Secured Web App in AWS cloud**



**Goal:** Goal of this POC is to prove that webservices can be deployed into private subnets and serve the traffic to public users via Internet facing Load Balancer.

**Resources Used:**  
VPC  
Subnets  
Route Tables  
Internet gateway  
NAT Gateway  
Security Groups  
Network ACL  
Elastic IP  
S3 Bucket  
SNS Topic  
Cloud Watch Alarms  
EC2 Instances  
Auto Scaling Group  
Network Load Balancer  
Route53 Hosted Zone  
IAM Policy  
IAM Role  
Session Manager

**Configuration:**  
You can choose any EC2 configuration (OS, Instance Type, Storage Size, Key Pair etc...) as the goal to deploy a webapp.

**Deployment:**

1. Deploy Scalable, High Available, and Secured Web App in AWS cloud as per the architecture provided above.
2. Create Golden AMI contains global configuration.  
   Install Java8  
   Install httpd web server  
   Install epel repo
3. Deploy Auto Scaling group with the Golden AMI created in the previous step.
4. Configure notification on each Auto Scaling Group event change.
5. Configure Scaling Policy to scale out when CPU utilization breaches the threshold 80% utilization.
6. Configure Scaling Policy to scale In when CPU utilization below the threshold 80% utilization.
7. Keep the index.html file in S3 bucket and configure Launch Template with user data to GET the index.html to /var/www/html location during EC2 launch.
8. Assume Security Group and Network ACL are integrated in this architecture to allow ports 22 and 80.

**Verification:**  
Login to EC2 instances using Session Manager and run stress command to verify the Scale Out activity  
Browse the web service using a browser from the public internet .

**Destroy:**  
Once POC is completed, destroy the resources created in this project.