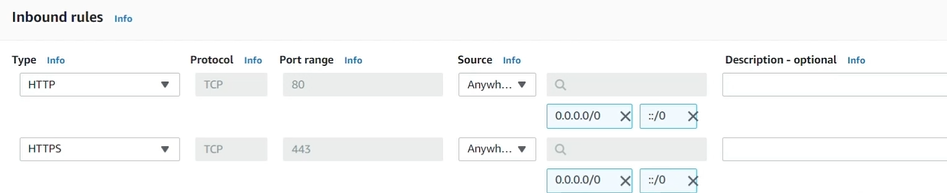
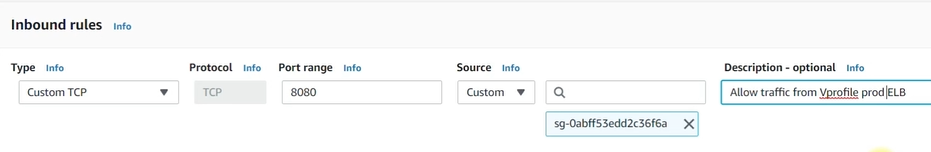


**SG SETUP** (Allow port 22 for SSH as well)

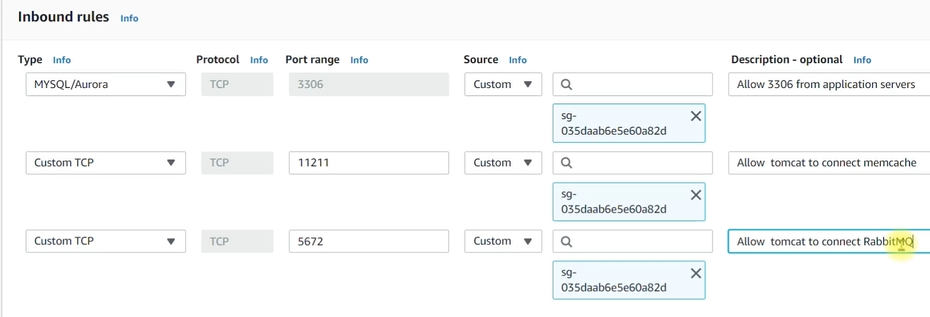
* **Name vprofile-ELB-SG**



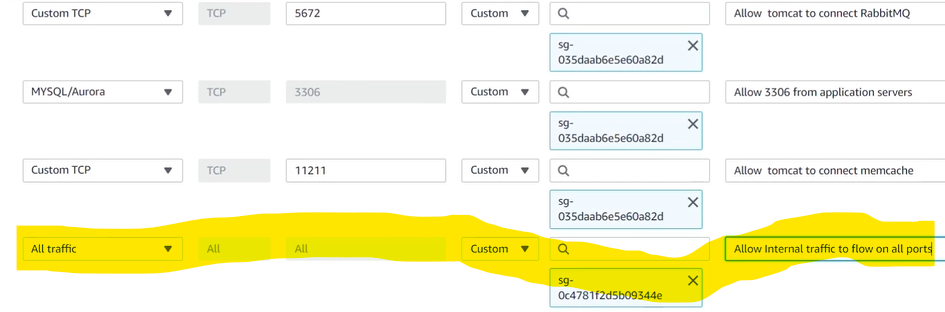
* **Name vprofile-app-SG**



* Name vprofile-backend-SG



All backend server will interact with each other, so add a 3rd rule “All traffic will flow to himself”



**Create a key pair**

**Create EC2 Instances**

Make sure to be in aws-LiftAndShift branch

We will use script in pani-devops-project/userdata directory to create instances (for backend)

Named db01 CentOS7 , t2-micro, mysql.sh, in backend-SG

Named mc01 CentOS7 , t2-micro, memecache.sh, in backend-SG

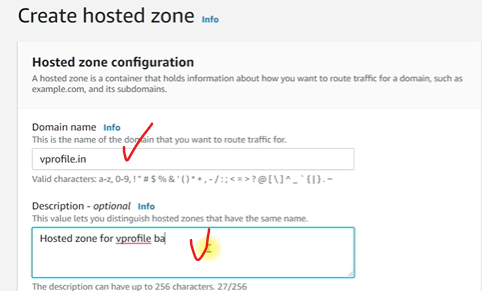
Named rmq01 CentOS7 , t2-micro, rabbitmq.sh, in backend-SG

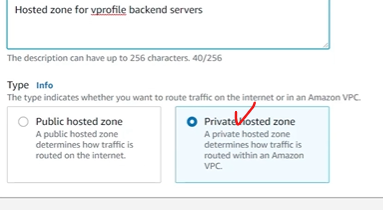
Then VAILDATE each instances

UPDATE Private IPs of the 3 instances into Route 53 Private DNS zone

Route 53 --🡪 Create hosted zone -🡪

Domain name can be any think

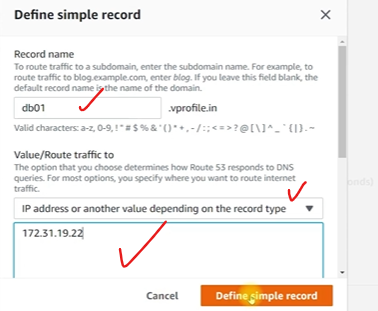


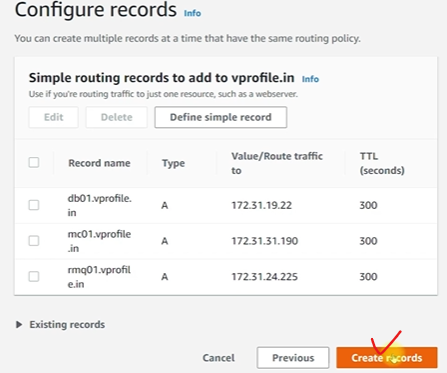


**Create record**

--🡪 Simple routing

---🡪 Define simple record ( For all 3 knowing the private IPs)





This record allow us to use the record name instead of IP or hostame so that even if we recreate the backend server we won’t need to modifier our app as long as we use same hostname.

**Create Tomcat server**

Named app01 Ubuntu.18 , t2-micro, tomcat\_ubuntu.sh, in app-SG

**BUILD AND DEPLOY ARTIFACTS**

Make sure to have java and maven installed

Update in pani-devops-project/src/main/resources/application.properties file the hostname with the record name

Go to top level directory where we have pom.xml

# mvn install (will take some time to run)

In the target directory, you will see the .war file that generated, .war will be store in S3 for later use

Create an AWS user with S3 access and setup your machine to run AWS cmd or awscli

# aws s3 mb s3://pani-artifact-storage

# cd target ; aws s3 cp vprofile-v2.war s3://pani-artifact-storage/vprofile-v2.war

In oder to download artifact to Tomcat, we will create a role with S3FullAcess and update to APP01

Login to APP01

Username is ubuntu

# sudo -i

# systemctl status tomcat8 (to make sure it’s running)

# /var/lib/tomcat8

# cd webapps (Will delete ROOT dir)

# systemctl stop tomcat8

# rm -rf ROOT

Download .war from S3

# apt install awscli -y

# aws s3 cp s3://pani-artifact-storage/ vprofile-v2.war /tmp

# cd /tmp

# cp vprofile-v2.war /var/lib/tomcat8/webapps/ROOT.war

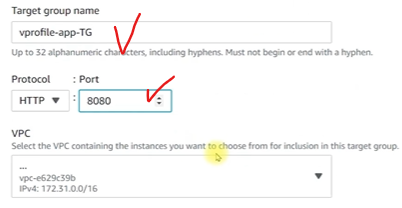
# systemctl start tomcat8

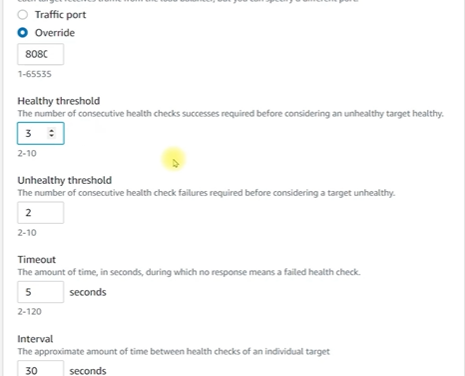
# ls /var/lib/tomcat8/webapps

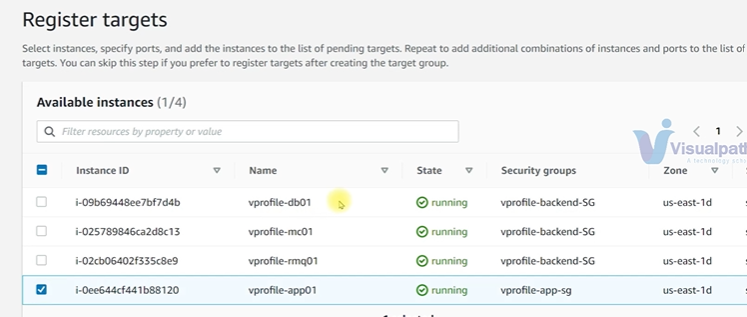
You can check the WEB-INF in ROOT to see component of your app

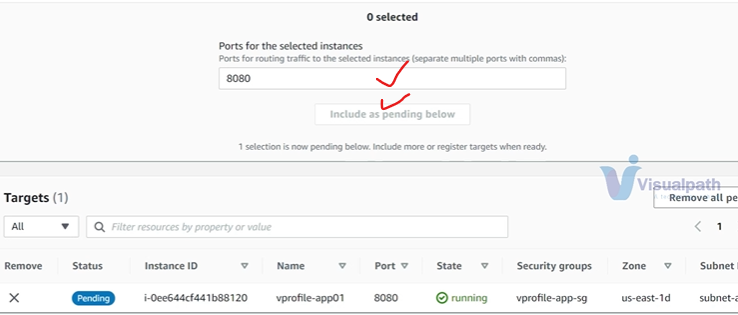
**CREATE APP ELB**

* Create Target group first

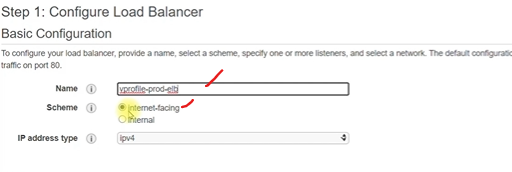




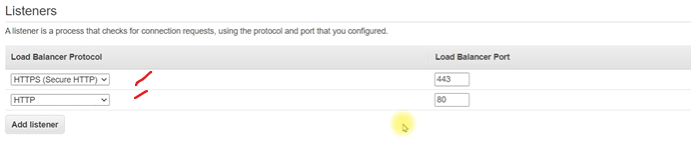




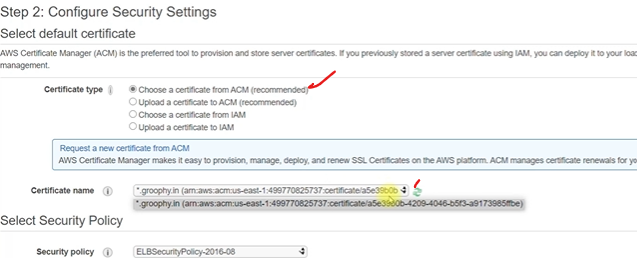
* **Create ELB**



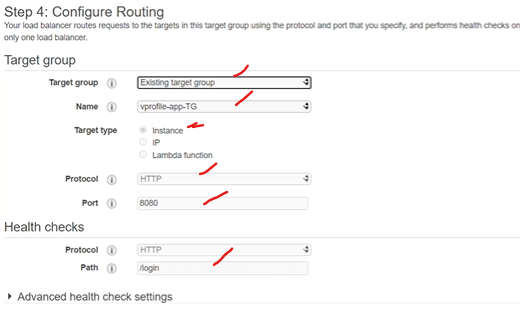
You will add port 80 and remove later



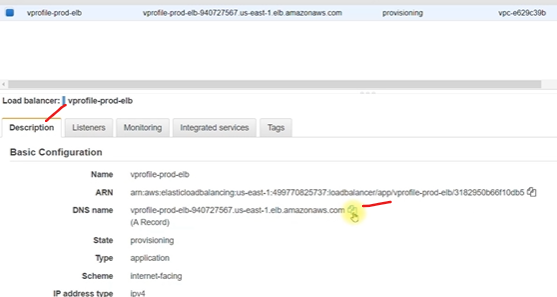
For AZs, select all zones



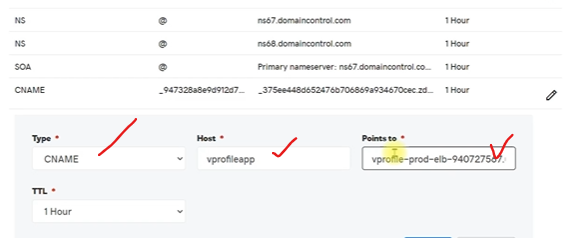
Choose ELB-SG



Copy your End point (DNS name) and update to your godaddy



Godaddy ---🡪 my domain --🡪 domain settings



Then our url will be <https://vprofileapp.itresolveit.com>

Where host = vprofileapp and your domain from godaddy = itresolveit.com

Validate your app (<https://vprofileapp.itresolveit.com>)

**CREATE ASG**

* Create image of the app01
* Create Launch config (add IAM role so it can get artifact from S3, add APP-SG and key pair)
* Create ASG

