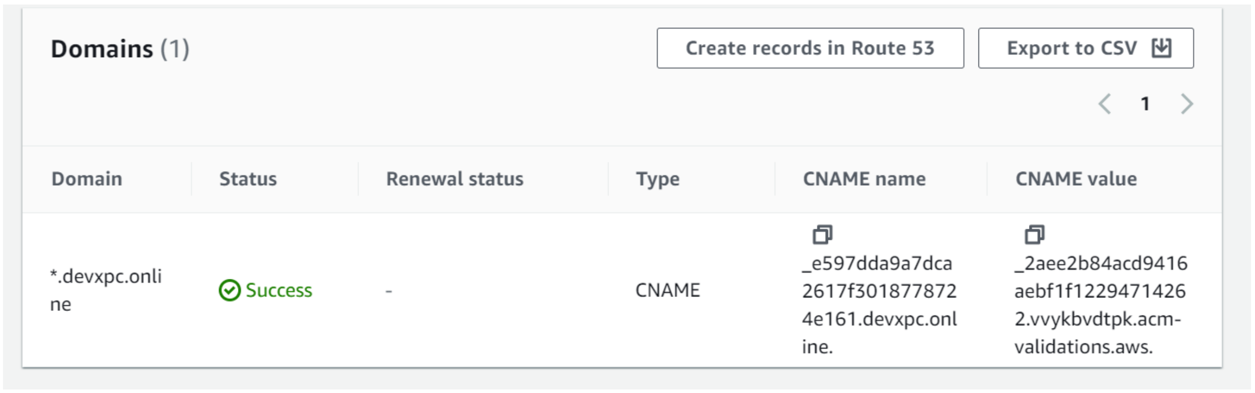
🤓

Class Project IAAS

Certificate Setup

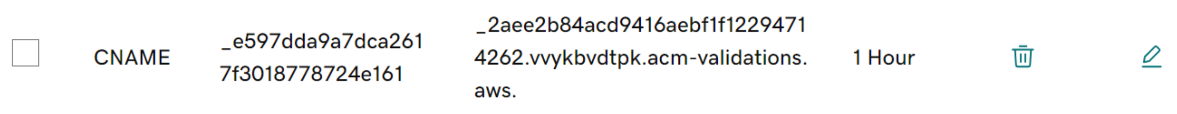
1. Go to ACM
2. It is important to note the region where you are creating the certificate.
3. Request a public certificate
4. Enter your wildcard domain name *.devxpc.online
5. Add a name tag
6. Validate by clicking on the newly created certificate and copy the CNAME name and value and add to your domain registrar
    a. Create a CNAME DNS record on your domain registrar
    b. for the name: remove the the domain name and fullstop at the end. For the value: remove the fullstop at the end
7. ACM will keep checking to see that the domain name resolves to the value

AWS Portal



Domain	Status	Renewal status	Type	CNAME name	CNAME value
*.devxpc.online	Success	-	CNAME	_e597dda9a7dca2617f3018778724e161.devxpc.online.	_2aee2b84acd9416aebf1f12294714262.vvykbvdtpk.acm-validations.aws.

GoDaddy



CNAME	_e597dda9a7dca2617f3018778724e161	_2aee2b84acd9416aebf1f12294714262.vvykbvdtpk.acm-validations.aws.	1 Hour	

FLOW OF EXECUTION

1. Login to AWS Account
2. Create Key Pairs
3. Create Security Groups
4. Lunch Instances with userdata
5. Update IP to name mapping on route 53
6. Build Application From Source Code

7. Upload to S3 bucket
8. Download Artifact to Tomcat Ec2 instance
9. Setup ELB with HTTPS with the ACM Cert
10. Map ELB EndPoint to website name in Godaddy DNS
11. Verify
12. Build AutoScaling Group for TomCat Instances

- Create Security Groups
    - For the load balancer SG allow http & https traffic from anywhere
    - App-SG will allow traffic from loadbalancer on port 8080
    - Backend-SG will allow mysql service on port 3306, 5672 RabbitMQ and 11211 Memecahe
        - After creating the Backend -SG, create an additional rule to allow it receive all traffic from itself
    - Allow SSH on all created security groups
- Create Key Pair to login to the instances

Deploying VMS

Use tc2.micro

- rmq - centos9
- mc - centos9
- db -centos9
- app -ubuntu 22.04

Run the required user data scripts. they can be found in userdata folder

Db server script

```
#!/bin/bash
DATABASE_PASS='admin123'
sudo yum update -y
sudo yum install epel-release -y
sudo yum install git zip unzip -y
sudo yum install mariadb-server -y


# starting & enabling mariadb-server
sudo systemctl start mariadb
sudo systemctl enable mariadb
cd /tmp/
git clone -b main https://github.com/hkhcoder/vprofile-project.git
#restore the dump file for the application
sudo mysqladmin -u root password "$DATABASE_PASS"
sudo mysql -u root -p"$DATABASE_PASS" -e "UPDATE mysql.user SET Password=PASSWORD('$DATABASE_PASS') WHERE User='root'"
sudo mysql -u root -p"$DATABASE_PASS" -e "DELETE FROM mysql.user WHERE User='root' AND Host NOT IN ('localhost', '127.0.0.1', '::1')"
sudo mysql -u root -p"$DATABASE_PASS" -e "DELETE FROM mysql.user WHERE User=''"
sudo mysql -u root -p"$DATABASE_PASS" -e "DELETE FROM mysql.db WHERE Db='test' OR Db='test\_%'"
sudo mysql -u root -p"$DATABASE_PASS" -e "FLUSH PRIVILEGES"
sudo mysql -u root -p"$DATABASE_PASS" -e "create database accounts"
sudo mysql -u root -p"$DATABASE_PASS" -e "grant all privileges on accounts.* TO 'admin'@'localhost' identified by 'admin123'"
sudo mysql -u root -p"$DATABASE_PASS" -e "grant all privileges on accounts.* TO 'admin'@'%' identified by 'admin123'"
sudo mysql -u root -p"$DATABASE_PASS" accounts < /tmp/vprofile-project/src/main/resources/db_backup.sql
sudo mysql -u root -p"$DATABASE_PASS" -e "FLUSH PRIVILEGES"

# Restart mariadb-server
sudo systemctl restart mariadb


#starting the firewall and allowing the mariadb to access from port no. 3306
```

```
sudo systemctl start firewalld
sudo systemctl enable firewalld
sudo firewall-cmd --get-active-zones
sudo firewall-cmd --zone=public --add-port=3306/tcp --permanent
sudo firewall-cmd --reload
sudo systemctl restart mariadb
```

MC user script

```
#!/bin/bash
sudo dnf install epel-release -y
sudo dnf install memcached -y
sudo systemctl start memcached
sudo systemctl enable memcached
sudo systemctl status memcached
sed -i 's/127.0.0.1/0.0.0.0/g' /etc/sysconfig/memcached
sudo systemctl restart memcached
firewall-cmd --add-port=11211/tcp
firewall-cmd --runtime-to-permanent
firewall-cmd --add-port=11111/udp
firewall-cmd --runtime-to-permanent
sudo memcached -p 11211 -U 11111 -u memcached -d
```

RMQ user script

```
#!/bin/bash
sudo yum install epel-release -y
sudo yum update -y
sudo yum install wget -y
cd /tmp/
dnf -y install centos-release-rabbitmq-38
 dnf --enablerepo=centos-rabbitmq-38 -y install rabbitmq-server
 systemctl enable --now rabbitmq-server
 firewall-cmd --add-port=5672/tcp
 firewall-cmd --runtime-to-permanent
sudo systemctl start rabbitmq-server
sudo systemctl enable rabbitmq-server
sudo systemctl status rabbitmq-server
sudo sh -c 'echo "[{rabbit, [{loopback_users, []}]}]." > /etc/rabbitmq/rabbitmq.config'
sudo rabbitmqctl add_user test test
sudo rabbitmqctl set_user_tags test administrator
sudo systemctl restart rabbitmq-server
```

Tomcat user script

```
#!/bin/bash
sudo apt update
sudo apt upgrade -y
sudo apt install openjdk-11-jdk -y
sudo apt install tomcat9 tomcat9-admin tomcat9-docs tomcat9-common git -y
```

CHECKS

To retrieve user data information, run the following curl the path http://169.254.169.254/latest/user-data

Login to each server and check that the deployed services are up and running

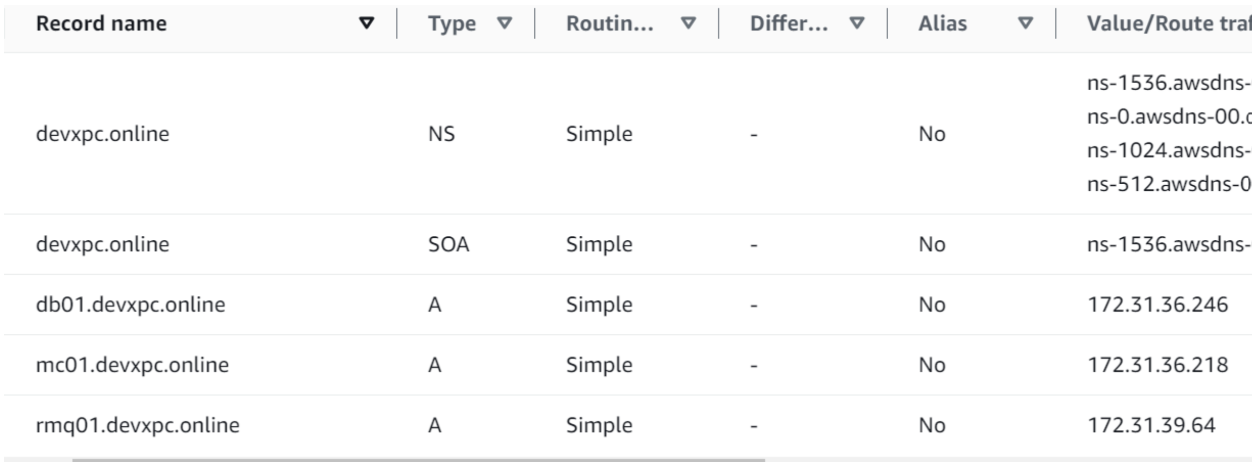
- mdb
  - systemctl status mariadb
  - mysql -u admin -padmin123 accounts
    - show tables;
- mc

    - ss -tunlp | grep 11211
    - systemctl status memcached
- rmq
    - systemctl status rabbitmq-server
    - ss -tulnp | grep 64292 #check if the port is open by checking with the pid or service name [Tell Us Local Network Port]
- App
    - systemctl status tomcat9
    - ls /var/lib/tomcat9
    - curl -vL localhost:8080

Route53

A zone needs to be created with your domain name. this zone should be private which means it will not be resolved from the internet.

- click create hosted zone [select n.vgn region and private hosted zone]
- After creation click on create record (select - use wizard at to right Create Simple Record) using the private IP addresses
    - db01 172.31.36.264



Record name	Type	Routin...	Differ...	Alias	Value/Route traf
devxpc.online	NS	Simple	-	No	ns-1536.awsdns- ns-0.awsdns-00.c ns-1024.awsdns- ns-512.awsdns-0
devxpc.online	SOA	Simple	-	No	ns-1536.awsdns-
db01.devxpc.online	A	Simple	-	No	172.31.36.246
mc01.devxpc.online	A	Simple	-	No	172.31.36.218
rmq01.devxpc.online	A	Simple	-	No	172.31.39.64

Deploying Artifact

VSCODE Setup

- ctr+shft+p
- search for “select default profile” click and select git bash

1. **change the entries to the fqdn created in the previous step at application.properties file in src/main/resources**
2. ensure dependencies are installed
    a. install chocolate package manager (this is used to install the packages you will need to run the app on your computer) -run in powershell

```
Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointMana
```

   b. choco install corretto11jdk -y
   c. choco install maven -y
3. **create the artifact by running “mvn install” on the command line.**
4. Create an s3admin user under IAM Add users
   a. Select attach policies directly and search for s3 and grant AmazonS3FullAccess
   b. create access key for cmd
   c. On your terminal enter “aws configure”
5. Create S3 Bucket. #The Bucket name must be globally unique

```
aws s3 mb s3://ducket-repos-artifact

ouutput:make_bucket: ducket-repos-artifact
```

6. Copy files from local machine to Bucket

```
aws s3 cp target/vprofile-v2.war s3://ducket-repos-artifact/
```

7. Create IAM role for app01 instance
   a. IAM Roles Create Roles EC2
   b. choose a permission policy “s3FullAccess” and give the role a name.
   c. Select the Instance Action Security modify IAM role select role and save
8. Login to app01

```
apt update
apt install awscli -y
aws s3 ls
aws s3 cp s3://<bucket-name>/vprofile-v2.war /tmp/
systemctl stop tomcat9
rm -rf /var/lib/tomcat9/webapps/ROOT
cp /tmp/vprofile-v2.war /var/lib/tomcat9/webapps/ROOT.war
systemctl start tomcat9

--Verify
ls /var/lib/tomcat9/webapps/
ROOT  ROOT.war
cat /var/lib/tomcat9/webapps/ROOT/WEB-INF/classes/application.properties
#ensure the dns changes made persist
```

Load Balancer

Go to load balancer in the EC2 dashboard and:

1. Create a Target Group
   - protocol: HTTP, Port: 8080
   - Health check: /login
     - Advance Health check: Port: Override: 8080
     - Health Threshhold: 3
   - Select App01 from the next plane and include as pending, create target group.
2. Create an Application load balancer:

    a. It should be Ipv4 internet facing.
    b. Select all the AZ
    c. add both http and https to the Listener
    d. Add the newly created ACM Certificate: if it does not appear, ensure that you are in the right region

3. Create a CNAME Entry on godaddy:
    a. Create New Record
    b. enter a name you want to serve as the prefix before your domain
    c. enter the loadbalancer DNS name as the Value e.g prod-elb-920676313.us-east-1.elb.amazonaws.com
4. Visit the site e.g hagitex.devxpc.online
5. login with:
    - username: admin_vp
    - password: admin_vp

AutoScaling Group

1. Slect instance ▯ Actions ▯ Image and Templates ▯ Create Image
2. Create Autoscaling group and lunch template in one go

- For the lunch template;
    - Select the AMI you just created and t2micro instance type
    - Enure you assign your IAM role to your instance
    - Select the App security group
- For Auto-Scaling group:
    - Select Lunch template and select all the subnet.
    - enable load balance and select the target group, also allow health check on ELB
    - select desired min and max
    - Add notification to your sns topic