DevOps Project 1

SSL Certificate Setup in AWS cloud for our domain

1. Login to AWS account
2. Search for ACM – Certificate Manager
3. Remember in which region you are creating the certificate (N. Virginia)
4. Click on “Request Certificate”
5. Enter your domain name as **\*.techgeeks.tech**
6. Add tag like name of the certificate – Name, techgeeks.com
7. Select DNS validation option
8. Click on Request
9. You can see “Pending validation” status. You need to verify it. To do verify click on certificate ID
10. Create CNAME record in your domain by copying the CNAME name and CNAME value
11. Come back to AWS and refresh. It will take some to verify and issue.

Security Group

1. Login to AWS account and go to EC2
2. Go to Security group
3. Create security group, give name “project-1-ELB-SG”
4. Add inbound rule for http & https, allowed from anywhere IPv4 & IPv6
5. Create one more security group as, “project-1-app-sg”, description “security group for tomcat instances”
6. Add inbound rule for port 8080 & allowed only from load balancer security group which we just added
7. One more security group for backend services. You can give name as “project-1-backend-sg”
8. Add inbound rule for mysql 3306 port, allowed only from “application security group”

Add inbound rule for memcache 11211 port, allowed only from ‘application security group”

Add inbound rule for rabbitMq 5672 port, allowed only from “application security group”

Add inbound rule to allow port 22, allowed from my ip

Save it.

1. Edit the “project-1-backend-sg”. Add one inbound rule for each backend servers interacting with each other.

Create rule for “all traffic” from itself “project-1-backend-sg”

Key Pair

1. Go to Key Pairs
2. Create key pair
3. Project-prod-key
4. Use .pem format

EC2 Instances

MYSQL Instance

1. Launch instance “project-1-db01”
2. We need “centos 7” image for that click on “Browse more AMI’s”
3. Go to AWS marketplace
4. Search for “centos 7” & select it
5. Instance type t2.micro is fine
6. Select the keypair which we created
7. Select security group “project-1-backend-sg”
8. Click on Advance details
9. Copy paste the user data script of mysql.sh
10. Click on Launch Instance

# Use below command to connect to this machine

Ssh -i Downloads/project-prod.pem centos@public\_IP

Login as root user:

Sudo -I

To see the userdata from this machine use below command

$curl <http://public_IP/latest/user-data>

$ systemctl status mariadb

Ps -ef

# mysql -u root -p

# show databases;

# use accounts;

# show tables;

MEMCACHE Instance

1. Launch instance “project-1-mc01”
2. We need “centos 7” image for that click on “Browse more AMI’s”
3. Go to AWS marketplace
4. Search for “centos 7” & select it
5. Instance type t2.micro is fine
6. Select the keypair which we created
7. Select security group “project-1-backend-sg”
8. Click on Advance details
9. Copy paste the user data script of memcache.sh
10. Click on Launch Instance

Ssh -i Downloads/project-prod.pem centos@public\_IP

Login as root user:

Sudo -I

To see the userdata from this machine use below command

$curl <http://public_IP/latest/user-data>

$ systemctl status Memcached

$ ss -tunpl | grep 11211

RabbitMQ Instance

1. Launch instance “project-1-rmq01”
2. We need “centos 7” image for that click on “Browse more AMI’s”
3. Go to AWS marketplace
4. Search for “centos 7” & select it
5. Instance type t2.micro is fine
6. Select the keypair which we created
7. Select security group “project-1-backend-sg”
8. Click on Advance details
9. Copy paste the user data script of rabbitmq.sh
10. Click on Launch Instance

Ssh -i Downloads/project-prod.pem centos@public\_IP

Login as root user:

Sudo -I

To see the userdata from this machine use below command

$curl <http://public_IP/latest/user-data>

$ systemctl status rabbitmq-server

Update the private IP addresses of backend servers in Route 53 private DNS zone

1. Go to Route 53 service
2. Click on create hosted zone
3. Select “Private hosted zone”
4. Give any domain name. Like sscademy.in
5. Select the region “N.Virginia(US-East1)
6. Create record
7. Select simple routing, click Next
8. Define simple record
9. db01.sscademy.in, point to “IP Address or another value depending on the record type”
10. Give the private IP address
11. Same create for all three backend services

Tomcat EC2 Instance

1. Launch instance “project-1-app01”
2. We need “ubuntu server 18.04 LTS” image
3. Instance type t2.micro is fine
4. Select the keypair which we created
5. Select security group “project-1-backend-sg”
6. Click on Advance details
7. Copy paste the user data script of tomcat\_ubuntu.sh

Click on Launch Instance

Build Artifact

1. Install JDK & Maven in our Windows 10 laptop
2. Open powershell as administrator
3. Run below command to install chocolaty in Windows

Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))

1. Install JDK8 using below command

Choco install jdk8

1. Go to location /src/main/resources
2. Edit the “application.properties” file
3. Edit the URLs as we added to private Route 53 hosted zone

jdbc.url=jdbc:mysql://db01.sscademy.in:3306

memcached.active.host=mc01.sscademy.in

rabbitmq.address=rmq01.sscademy.in

1. Save the file
2. Go back to the root directory location where we have the pom.xml file
3. Run command “mvn install”
4. Now it will create ‘Target” folder and inside we will have our aritifact name as

sscadmy-v2.war

Upload Artifact to S3 bucket

1. Install awscli using “choco install awscli” in Powershell as administrator
2. Create an IAM user who will be the admin for S3
3. Give fullaccess to S3
4. Run command “aws configure”
5. Enter the AWS Access Key ID
6. AWS Secret Access Key
7. Enter Region Name as “us-east-1”
8. Enter default output format as “json”
9. Now we will create S3 bucket using command “**aws s3 mb s3://sscademy-artifact-storage”**
10. Go to the generated artifact location inside target folder

D:/sscademy/aws\_project\_1/target

1. Copy the artifact using command “**aws s3 cp sscademy-v2.war s3://sscademy-artifact-storage/sscademy-v2.war”**
2. **Aws s3 ls s3://sscademy-artifact-storage**

Deploy Artifact to TOMCAT Server

1. Create an IAM role for EC2 instance to download the artifact
2. Go to IAM in AWS console
3. Click on ‘Roles”
4. Create role, select option “common use cases” – “EC2”
5. Click Next, search “s3fullaccess”, select it.
6. Give name as “sscademy-artifact-storage-role”
7. Click create role
8. Lets attach the role to TOMCAT EC2 instance
9. Go to EC2 instance from AWS console
10. Select Tomcat server – Actions – Security – Modify IAM role – select the newly created role and click on update IAM role
11. Login to TOMCAT instance from your machine
12. Switch to root account using sudo su –
13. Systemctl status tomcat8
14. Go to /var/lib/tomcat8/
15. Cd webapps/
16. Systemctl stop tomcat8
17. Rm -rf ROOT
18. Apt install awscli -y
19. Check the artifact stored on s3 bucket using command “**aws s3 ls s3://sscademy-artifact-storage”**
20. Download the artifact using command “**aws s3 cp s3://sscademy-artifact-storage/ sscademy-v2.war /tmp/sscademy-v2.war”**
21. **Copy artifact to cp /tmp/sscademy-v2.war /var/lib/tomcat8/wedapps/ROOT.war”**
22. **Lets start the tomcat again “systemctl start tomcat8”**
23. **Cd /var/lib/tomcat8/webapps/ROOT/WEB-INF/classes**
24. **Check application.properties file**
25. **Apt install telnet -y**
26. **telnet db01.sscademy.in 3306**

LOAD BALANCER and DNS

1. Setup load balancer using following steps
2. Go to EC2 from AWS console
3. Click on “Target Group” in Load Balancing service
4. Create ‘Target group”
5. Select “Instances” - Type the Target group name “sscademy-app-tg”
6. Select Protocol – HTTP & Port - 8080
7. Edit health check to - **/login**
8. **Click on Advance health check settings**
9. **Change the port to – Override & give port number as 8080**
10. Change healthy threshold to 3
11. Click Next
12. Select application server - tomcat
13. Click “Include as pending below”
14. Create target group
15. Now lets create **Load Balancer**
16. **Select Application Load Balancer**
17. **Give load balancer name – sscademy-prod-lb**
18. **Select Internet facing**
19. **Select all the zones**
20. **Select load balancer security group which we have created earlier**
21. **Select Listner for HTTP 80 & HTTPS 443 & select target group**
22. Select the certificate which we created in ACM
23. Click on create load balancer
24. Copy the end point of DNS of load balancer
25. Go to Hostinger and add CNAME record for this DNS name for load balancer
26. Give **“host”** as **“sscademyapp”** and value copy paste from load balancer DNS