**Deploy a Spring Boot Application in AWS using EC2 and S3**

**Step-1** -> Create a New Spring Starter Project – sb--aws-app

Package : com.capgemini.aws Dependency : Spring Web

-> In starter class

Create WelcomeRestController /welcome

-> In pom.xml

<build> <finalName>**sb-aws-app**</finalName></build>

> mvn clean compile install Run As > Maven Install

**Now we need to add this jar to EC2 Instance**

**Amazon S3 Service**

- Secure, durable, and scalable object storage infrastructure.

- Amazon S3 or Amazon **Simple Storage Service** is a service offered by Amazon Web Services that provides object storage through a web service interface. Amazon S3 uses the same scalable storage infrastructure that Amazon.com uses to run its global e-commerce network.

> Place the spring boot jar in S3 bucket and our EC2 instance will fetch it and run it

**Step-2 - Go to AWS Console > Services > Storage > S3**

-- Create a New Bucket (or) Use an Existing Bucket

Create a New Bucket

Bucket Name : spring-aws-bucket

Object Ownership**: ACLs enabled**

**Unselect -** Block all Public Access Acknowledge it > **Create Bucket** --- Bucket Created

Click on the newly Created Bucket

Use Create Folder – if you have different type of applications like Spring, Angular etc

Click on **Upload** > Add files : sb-aws-app.jar > Upload > Close

Now click on the uploaded jar file - - Bucket – Permissions – Object Ownership – ACLs Enables (Access Control List) – select Read , Write for all

**Copy Object URL from "Properties" : https://spring-aws-1.s3.us-east-1.amazonaws.com/microservice/sb-aws.jar**

**Step: 3 To download the jar from S3 bucket to EC2 instance**

Connect to EC2 instance from console

cmd> ssh -i "sunil-ec2.pem" [ubuntu@ec2-54-89-142-235.compute-1.amazonaws.com](mailto:ubuntu@ec2-54-89-142-235.compute-1.amazonaws.com)

install Java

cmd> sudo **apt install openjdk-17-jdk-headless**

**$ wget https://springawssun.s3.amazonaws.com/spring-boot-aws.jar**

--- now downloaded the jar

--- to check jar is present or not **$ ls**

**Step-4** -- run the jar file

**$java -jar sb- aws-app.jar** -- application should be running

On Browser - <http://54.89.142.235:8080/welcome> To get the IP address of EC2 Instance

In AWS Console > EC2 > Instances > Running Instances > Select our instance

-->May not have response

Apply Inbound Rules for 8080 --- EC2 Instance – Security – Security groups – Edit inbound rules – Add rule

Custom Tcp – 8080 – Anywhere --- Save rules

Open Browser - http://54.89.142.235:8080/welcome

**Install Docker on Ubuntu in AWS**

**1. AWS Prerequisites**

-> Set up an EC2 instance in an Amazon VPC

-> An Ubuntu EC2 instance with an internet connection.

**2. Updating package index**

$sudo apt-get update

**3. Installing docker**

$sudo apt-get install docker.io -y

**3A. docker commands**

$ docker -v docker --version

$ sudo docker info

**4. Starting the Docker service**

$sudo systemctl start docker

**5. Verifying the installation**

$ docker -v docker –version

If any error comes like “permission denied”

$ sudo usermod -aG docker $USER

$ sudo reboot

$ sudo docker run hello-world

$sudo docker images

$sudo docker container list $sudo docker ps $sudo docker ps -a

$sudo docker rm <container-id>

$sudo docker rmi <image\_name with tag> sudo docker rmi hello-world:latest

**6. Enabling the Docker service**

To start the Docker service automatically when the instance starts, you can use the following command

$ sudo systemctl enable docker

7. Check the Docker version

$ docker --version

8. Add User to Docker Group

-Add your user to the Docker group to run Docker commands without 'sudo'

$sudo usermod -a -G docker $(whoami)

After executing this command, the user will be added to the docker group and will have the necessary permissions to run Docker commands without sudo

Note that the change to the user’s group membership will not take effect until the next time the user logs in. You can log out and log back in to apply the changes or use the following command to activate the changes without logging out:

$newgrp docker

**---- Install Docker on Linux EC2 Instance -----**

1. connect to EC2 instace

cmd> ssh -i "cap-dev.pem" ec2-user@ec2-98-81-200-231.compute-1.amazonaws.com

-i identity\_file A file from which the identity key (private key) for public key authentication is read.

**2. install docker using the following command**

$sudo yum install -y docker

sudo - substitute user do

yum (Yellow-dog Update Manager) - is used to install, update and remove packages from Red-Hat based Linux systems.

**3. start docker using the following command**

$sudo service docker start

**4. Docker commands**

$sudo docker info

$sudo docker images $sudo docker ps -a

**5. sudo docker run hello-world**