



Class Project PAAS

1. Create a key for Beanstalk
2. Create RDS but first Create an RDS Subnet Group and Parameter Group (Add description)



An RDS Subnet Group is **a collection of subnets that you can use to designate for your RDS database instance in a VPC**. Your VPC must have at least two subnets. These subnets must be in two different Availability Zones in the AWS Region where you want to deploy your DB instance.



Parameter groups **contain configuration parameters and their settings**. They govern the behavior of your RDS database.

- Select Databases on the left plane and create a new DB.
 - Select mysql
 - For the engine version use: 8.0.32
 - While creating the DB, in additional configurations, give the db a name (accounts)
- 3. Create Elastic Cache
 - a. Create Subnet and parameter group first (memecache is what is required for the project)
 - b. create a memecachecluster
 - i. node type; t2micro

Ensure all services are placed in the Backend Security Group

4. Create Amazon MQ (RabbitMQ)
 - a. under advanced, network and security should be private access and select the backend-sg
- Copy the database endpoint and note that because the database is only accessible through a private network. you need to create a virtual machine to initialize it.
5. Create a linux vm and add it to a security group that has ssh access
6. Edit the backend security group to allow access from the linux-vm security group on **port 3306**.

ssh to the vm and run the following commands - for RHEL/amazonlinux install mariadb

```
sudo apt update && sudo apt install mysql-client -y
mysql -h <endpoint> -u admin -p<PASSWORD> <database>
mysql -h database-1.ckopomjiovva.us-east-1.rds.amazonaws.com -u admin -p7NMiMm6A2hzbXknz68uI accounts
>show tables;
#create a db_backup.sql file and copy the script to it
mysql -h database-1.ckopomjiovva.us-east-1.rds.amazonaws.com -u admin -p7NMiMm6A2hzbXknz68uI accounts < db_backup.sql
>show tables;
```

db_backup.sql → v-profile/src/main/resources/db_backup.sql

```
-- MySQL dump 10.13 Distrib 5.7.18, for Linux (x86_64)
--
-- Host: localhost    Database: accounts
```

```

-----
-- Server version 5.7.18-0ubuntu0.16.10.1

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `role`
--

DROP TABLE IF EXISTS `role`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `role` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `role`
--

LOCK TABLES `role` WRITE;
/*!40000 ALTER TABLE `role` DISABLE KEYS */;
INSERT INTO `role` VALUES (1, 'ROLE_USER');
/*!40000 ALTER TABLE `role` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `user`
--

DROP TABLE IF EXISTS `user`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) DEFAULT NULL,
  `userEmail` varchar(255) DEFAULT NULL,
  `profileImg` varchar(255) DEFAULT NULL,
  `profileImgPath` varchar(255) DEFAULT NULL,
  `dateOfBirth` varchar(255) DEFAULT NULL,
  `fatherName` varchar(255) DEFAULT NULL,
  `motherName` varchar(255) DEFAULT NULL,
  `gender` varchar(255) DEFAULT NULL,
  `maritalStatus` varchar(255) DEFAULT NULL,
  `permanentAddress` varchar(255) DEFAULT NULL,
  `tempAddress` varchar(255) DEFAULT NULL,
  `primaryOccupation` varchar(255) DEFAULT NULL,
  `secondaryOccupation` varchar(255) DEFAULT NULL,
  `skills` varchar(255) DEFAULT NULL,
  `phoneNumber` varchar(255) DEFAULT NULL,
  `secondaryPhoneNumber` varchar(255) DEFAULT NULL,
  `nationality` varchar(255) DEFAULT NULL,
  `language` varchar(255) DEFAULT NULL,
  `workingExperience` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `user`
--

LOCK TABLES `user` WRITE;
/*!40000 ALTER TABLE `user` DISABLE KEYS */;
INSERT INTO `user` VALUES (7, 'admin_vp', 'admin@visualpathit.com', NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL);

```

```

/*!40000 ALTER TABLE `user` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `user_role`
--

DROP TABLE IF EXISTS `user_role`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `user_role` (
  `user_id` int(11) NOT NULL,
  `role_id` int(11) NOT NULL,
  PRIMARY KEY (`user_id`,`role_id`),
  KEY `fk_user_role_roleid_idx` (`role_id`),
  CONSTRAINT `fk_user_role_roleid` FOREIGN KEY (`role_id`) REFERENCES `role` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_user_role_userid` FOREIGN KEY (`user_id`) REFERENCES `user` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `user_role`
--

LOCK TABLES `user_role` WRITE;
/*!40000 ALTER TABLE `user_role` DISABLE KEYS */;
INSERT INTO `user_role` VALUES (4,1),(5,1),(6,1),(7,1),(8,1),(9,1),(10,1),(11,1),(12,1),(13,1);
/*!40000 ALTER TABLE `user_role` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2017-12-07 16:32:31

```

- Terminate Instance.

Amazon MQ Elastic Cache

- Copy the end point of the MQ & Elastic cache and remove the port number

BeanStalk Setup

- Set IAM role for Bean-stalk
 - Go to IAM → Roles → Create Role → AWS Service
 - Service or use case: EC2 → Next
 - Under Permission, Search for "bean" an select
 - AWSElasticBeanstalkWebTier
 - AdministratorAccess-AWSElasticBeanstalk
 - AWSElasticBeanstalkRoleSNS
 - AWSElasticBeanstalkCustomPlatformforEC2Role
- Delete any role with "aws-elasticbeanstalk-service-role"
- BeanStalk; Create Application
 - webserver environment
 - application name: lab-app

- select a loadbalancer name and select platform details

Platform [Info](#)

Platform type

- ☒ **Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)
- ☐ **Custom platform**
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Tomcat

Platform branch

Tomcat 8.5 with Corretto 11 running on 64bit Amazon Linux 2

Platform version

4.3.11 (Recommended)

corretto → java

- Select Custom configuration
 - Select the role you created from IAM and allow beanstalk to create service role and select keypair
 - NEXT: Select default VPC and Activate public IP Address and select all the subnets except for us-east-1e
 - **NEXT: Skip security group and allow it create its own.** Change Capacity from single instance to load balanced. Select: min;2 max;2
 - For instance type remove t3small and leave t3 micro
 - Ensure Load Balance **Visibility: Public**

S3 Bucket

- go to the elastic beanstalk bucket created by beanstalk under s3
- go to permissions → Object Ownership → enable ACL (Bucket Owner)

Beanstalk

- under environment, got to configuration → Instance traffic and scaling → Edit
- scroll down to processes and edit. change path from / to /login
- click on the sessions dropdown and enable stickiness
- Add Listener for port 443

come down and click on apply

EC2 SG ↔ Backend Security Group Setup

- go to ec2, select an instance → security → click on the security group → and copy the security group ID
- Allow ALL TRAFFIC From that security group to the BACKEND. go to the backend security and add inbound rule

applications.properties file

go to src → main → resources → applications.properties

```
mvn -version
mvn install
```

Deploy the .war file to the ElasticBean stalk

when you use certificate for a different domain it shows unsecure.

- Update domain

New Records

[CNAME records](#) are a type of subdomain, or alias, that points to another domain name.



Type *	Name *	Value *	TTL
CNAME ▾	grace	devprofile.us-east-1.elasticbeanstalk.com	1/2 Hour ▾

Add More Records

Save

Cancel

login with admin_vp

CloudFront [CDN]

Allowed HTTP methods: select all

ADD alternative CNAME

Select Cert and use tlsV1