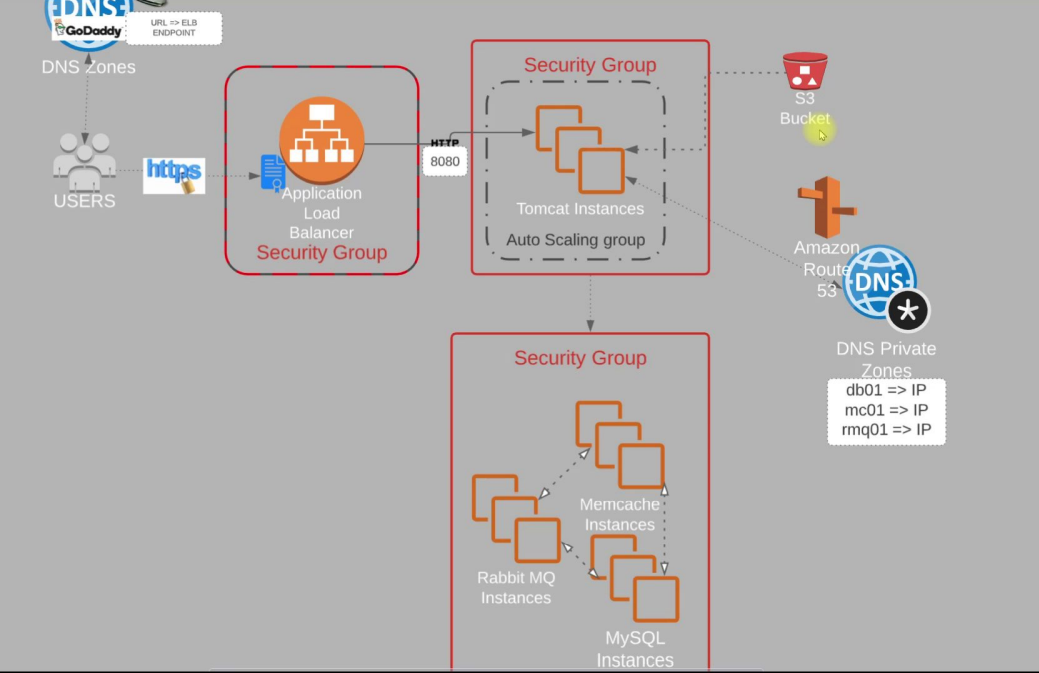
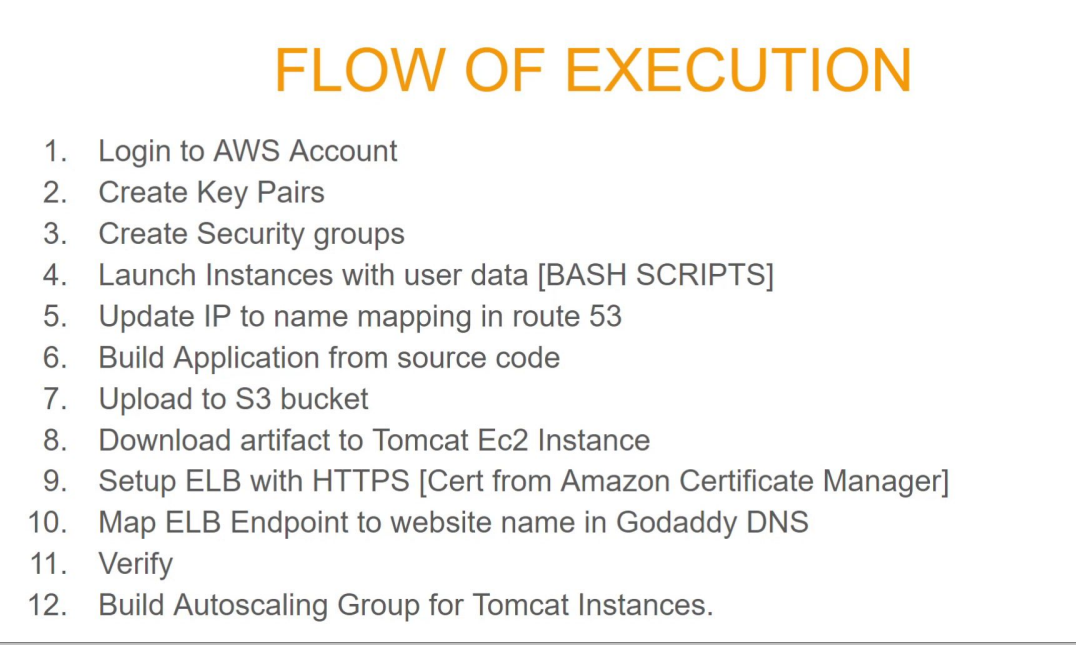






**THE PROJECT ARCHITECTURE**



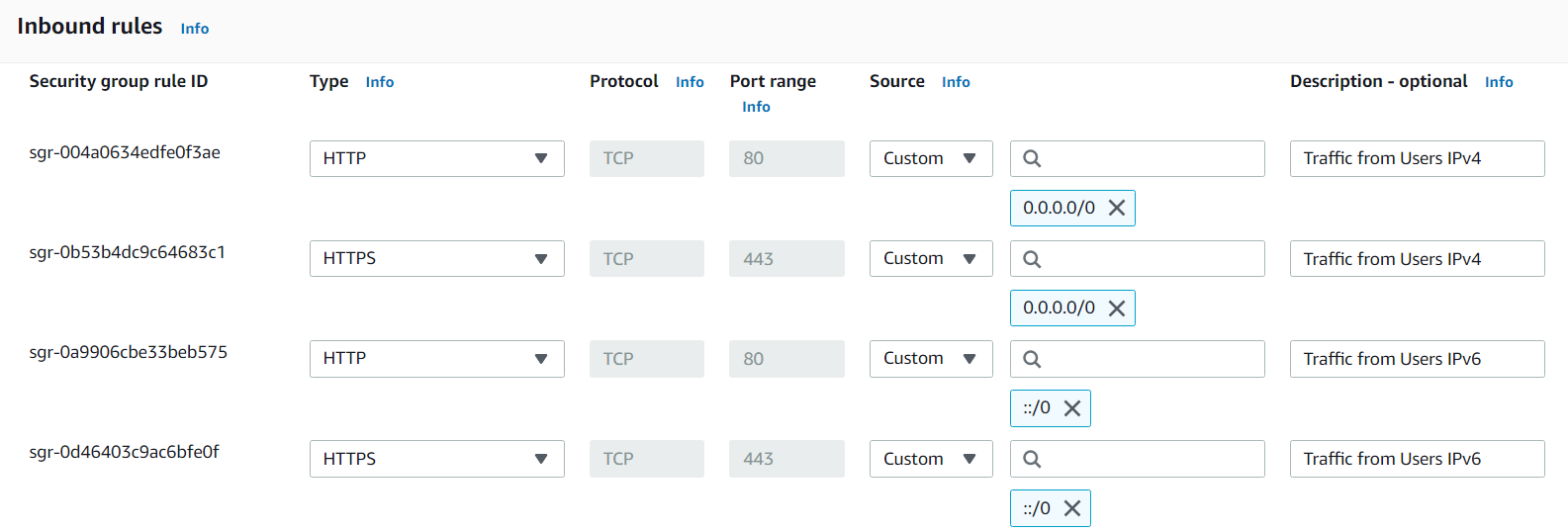


1. Login to AWS Account
2. Create Key Pairs

For the instances

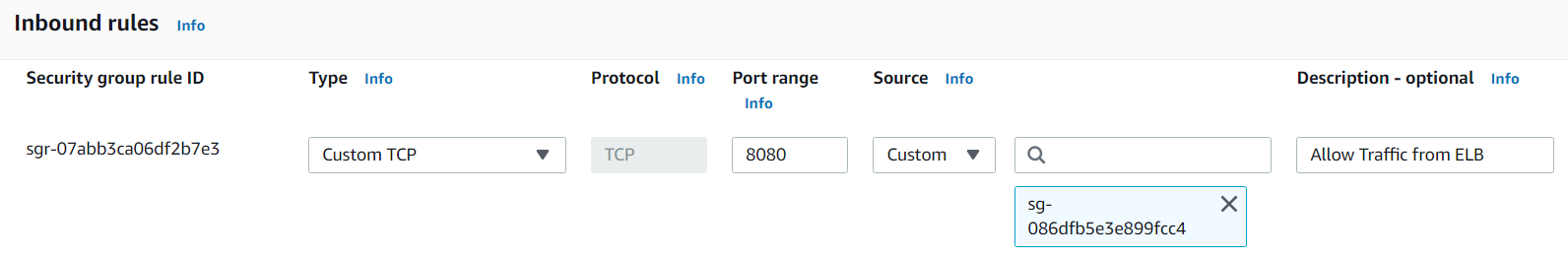
1. Create Security groups:

* Load Balancer



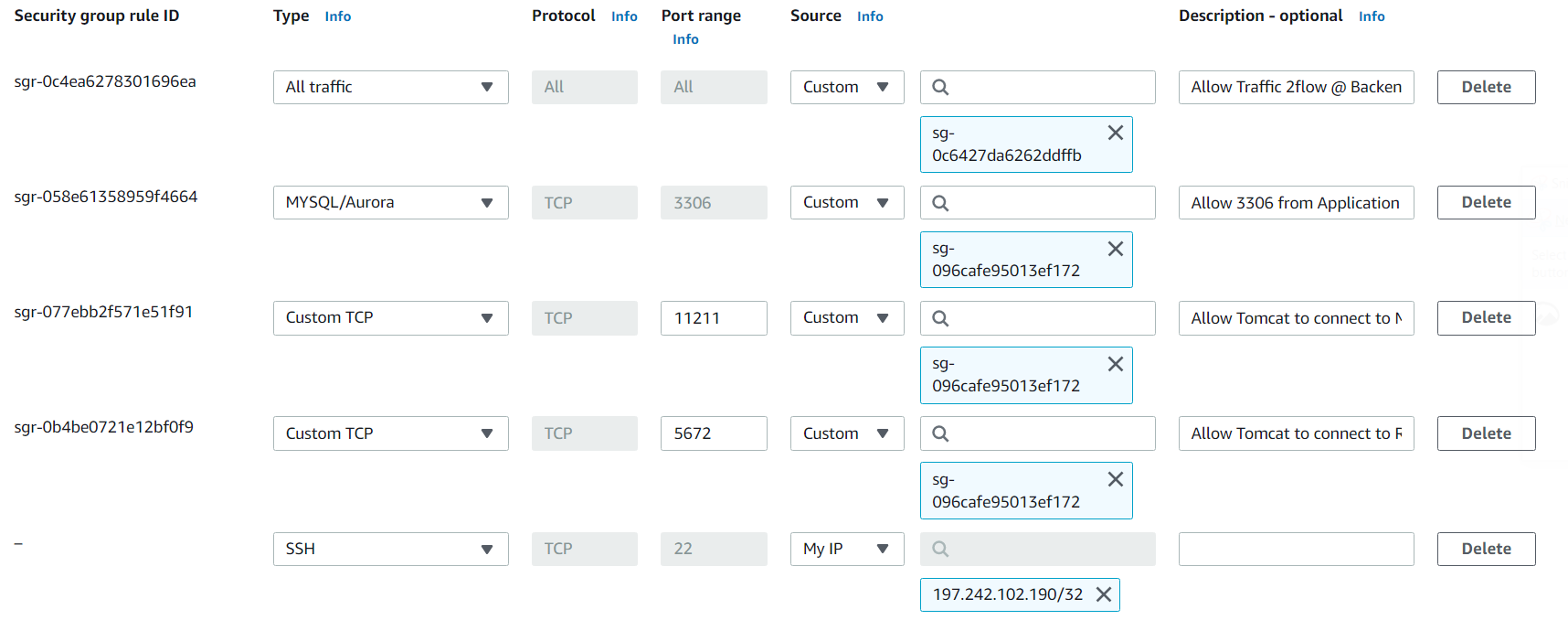
* Tomcat: It will be used for all the Autoscaling instances

(All should be allowed from the Load Balancer Security Group)



* Backend services: RabbitMQ, Memcache & MySQL

(All should be allowed from the Tomcat Application Security Group)



4. Launch Instances with User Data (Bash Scripts)

Check the folder scripts for the configuration scripts which would be used to provision the EC2 instances for the different services on the project Architecture.

* Start with EC2 instance for Mariadb
* Use Backend security group and same keypair
* ssh into the instance as root
* Use these code to check each provisioning for the Instances:

curl http://169.254.169.254/latest/user-data

* systemctl status mariadb
* ps –ef (To check the running processes)
* Don’t try to login with browser it is a db service
* mysql -u root –p(password)
* show databases; ……. to check databases created
* use accounts;
* show tables;
* Setup EC2 for Memcache
* Use scripts to provision
* Use Backend security group and same keypair
* use centos 7
* Use these code to check each provisioning for the Instances:

curl http://169.254.169.254/latest/user-data

* systemctl status memcached
* ss –tunpl | grep 11211
* Setup Ec2 for RabbitMQ
* Use Backend security group and same keypair
* Use scripts to provision
* use centos 7
* Use these code to check each provisioning for the Instances:

curl http://169.254.169.254/latest/user-data

* systemctl status rabbitmq-server

5. Updating the private IP (mapping of private IP’s) of the backend services in route53

* Get the private IP’s of the 3 Instances in a notepad

Example:

db01…… 18.191.11.61

mc01 ………. 18.217.126.52

rmq01 ….. 18.117.110.57

* Open Route53 service

click on hosted zones …… click create hosted zones ….. well done create records

6. Build Application from source code (Tomcat)

* Create and provision an Instance (Ubuntu 18 for Tomcat 8) while (Ubuntu 21 for Tomcat 9)

sudo su

apt-get update && apt-get upgrade

apt-get install tomcat8

apt-get install tomcat8-docs tomcat8-examples tomcat8-admin

sudo systemctl start tomcat8

sudo systemctl stop tomcat8

sudo systemctl restart tomcat8

Do these in gitbash:

/var/lib/tomcat8/conf/tomcat-users.xml

<role rolename="manager-gui"/>

<user username="tomcat" password="tomcat" roles="manager-gui"/>

sudo systemctl restart tomcat8

* Use same keypair used in backend services and Security group created for Tomcat
* Building Application from some source code

make sure you have jdk8 on your system. Open powershell as administrator (choco install jdk8 –y …… choco install maven -y)

* Check if some functions are on your system(powershell); mvn –version ….. java –version (both versions must be same for java) 🡺 choco uninstall maven -y

choco uninstall jdk8 –y

* Building Artifact:
* Go to git bash and make sure you are in root …… cd to where your dir where the artifact is(/c/gitrepos/cloudcodesandsecurity/DevOps/projects/Lift-Shift-Application -Workload/resources-artifact)
* vim application.properties

Edit the line with yellow with details from route53 : (jdbc.url=jdbc:mysql://db01.vprofile.com:3306/accounts?useUnicode=true&characterEncoding=UTF-8&zeroDateTimeBehavior=convertToNull)

memcached.active.host=mc01.vprofile.com

rabbitmq.address=rmq01.vprofile.com

* Move levels back to the dir where you see a POM.xml file……. then initiate the command

mvn install

* A dir named TARGET will be created containing our artifact

7.) Upload the Artifact into an S3 Bucket

* We would upload the artifact named (vprofile-v2.war) into an S3 Bucket
* We need AWSCLI (Open powershell as admin….. choco install awscli –y)
* Create an IAM user which would be used for S3 with AWSCLI. Make sure you grant AmazonS3 full access to it (Download the .csv file)
* cat Downloads/new\_user\_credentials.csv ……. to view the passwords (Be secretive with it)
* aws configure …. insert the access details from the .csv
* create an S3 Bucket: aws s3 mb s3://vprofile-artifact-storage-prep
* cd into the dir named target where the artifact is stored
* copy artifact into S3 bucket created: aws s3 cp vprofile-v2.war s3://vprofile-artifact-storage-prep/ vprofile-v2.war
* aws s3 ls s3://vprofile-artifact-storage-prep/
* create a role in IAM users and grant the role AmazonS3fullAccess
* Go to instances and select Tomcat-App01 …… click on Actions and click on security …….. click on modify IAM role …….. select the IAM role created and save
* Go to git bash and ssh to Tomcat App01
* systemctl status tomcat8
* cd /var/lib/tomcat8/ ……. ls
* cd webapps/ …. ls
* systemctl stop tomcat8 ……. ls
* rm -rf ROOT …… to remove the default app

8.) Download the Artifact into Tomcat EC2 instance

* apt install awscli -y
* aws s3 ls s3://vprofile-artifact-storage-prep
* aws s3 cp s3://vprofile-artifact-storage-prep/vprofile-v2.war /tmp/vprofile-v2.war
* cd /tmp/ …… ls
* cp vprofile-v2.war /var/lib/tomcat8/webapps/ROOT.war
* systemctl start tomcat8
* ls /var/lib/tomcat8/webapps/ …………… you will have a new extracted root dir
* cd /var/lib/tomcat8/webapps/ROOT/ …… ls
* cd WEB-INF/ ……ls
* cd classes/
* cat application.properties
* In case you spot errors: vim application.properties
* apt install telnet –y
* telnet db01.vprofile.com 3306 ……. to test your connection

9.) Setup ELB with HTTPS [Cert. from Amazon Certificate Manager]

* Go to Amazon certificate manager and ensure you have an existing issued certificate
* Go to Load balancers on the left pane…. click on target groups
* target type: Instances, port: 8080, Healthcheck: /login, Advanced Health check: port=8080, healthy threshold=3
* Select only Tomcat and include as Pending below
* now head on to load balancers …… add two listeners http: 80 & Https: 443
* Insert your Amazon certificate and click create ELB
* copy your ELB DNS name

10.) Map ELB Endpoint to website name in Godaddy DNS

* Go to your Doname provider and click on Domain Setting and click on DNS Management, Under records to add a CNAME host=vprofileapp, points to=ELB DNS link copied

11.) Verify

* Go to your browser and validate: vprofileapp.cloudcodesandsecurity.com
* login with details: admin\_vp (both username & pass)

12.) Building Autoscaling Group for Tomcat Instances

* We need to create an AMI for Tomcat app
* Go to Instances ….. select Tomcat app and click on Actions …… click Image& Templates and create image
* Go to Images on the left pane to check
* Go to Launch configuration on the left pane, click create launch config (Security group=app sg, keypair=use same keypair)
* Create Autoscaling group check on the left pane, click create,
* click on switch to launch configuration and select from drop down …. for subnets=select all …. click next …. enable load balancing ….. choose your target group…… do health checks on load balancer …..
* click on next … Capacity=1,2,4, target scaling policy, Target value=50, enable instance scale protection, add notification(Default), Add tag(Name=vprofile-app, Project=vprofile, Owner=Ola-Gabriel)
* verify and click create
* Go to your browser and validate: vprofileapp.cloudcodesandsecurity.com