< AWS CLOUD FOR WEB APP SETUP >

IMPLEMENTATION PLAN

VERSION 1.0 | 20/05/2022 – 22/05/2022

Purpose

In this section, describe the purpose to setup multi-tier web application using AWS Cloud.

About the Project

- *Multi-Tier web application Stack*
- *Host and Run on AWS Cloud for Production*
- *Baseline for upcoming projects.*
- *Lift and Shift Strategy*

Problem Statement

- *Complex Management of local Setup*
- *Scale Up/Down complexity.*
- *Upfront CapEx and regular OpEx*
- *Manual process, Difficult to automate & Time Consuming*

Solution

- *Cloud Setup*
- *Automation*
- *Code (IAAS)*
- *PayAsUgo*
- *Ease of infra management using Cloud*

AWS Services

- *EC2 Instances – VMs for (TOMCAT, RABBITMQ, MEMCACHE, MYSQL)*
- *ELB Load Balancer – NGINX LB replacement*
- *Autoscaling – Automation for VM Scaling*
- *S3/EFS Storage – Shared Storage*
- *Route 53 – Private DNS Service*

Objective

- *Flexible Infra*
- *No Upfront Cost*
- *Modernize Effectively*
- *IAAC*

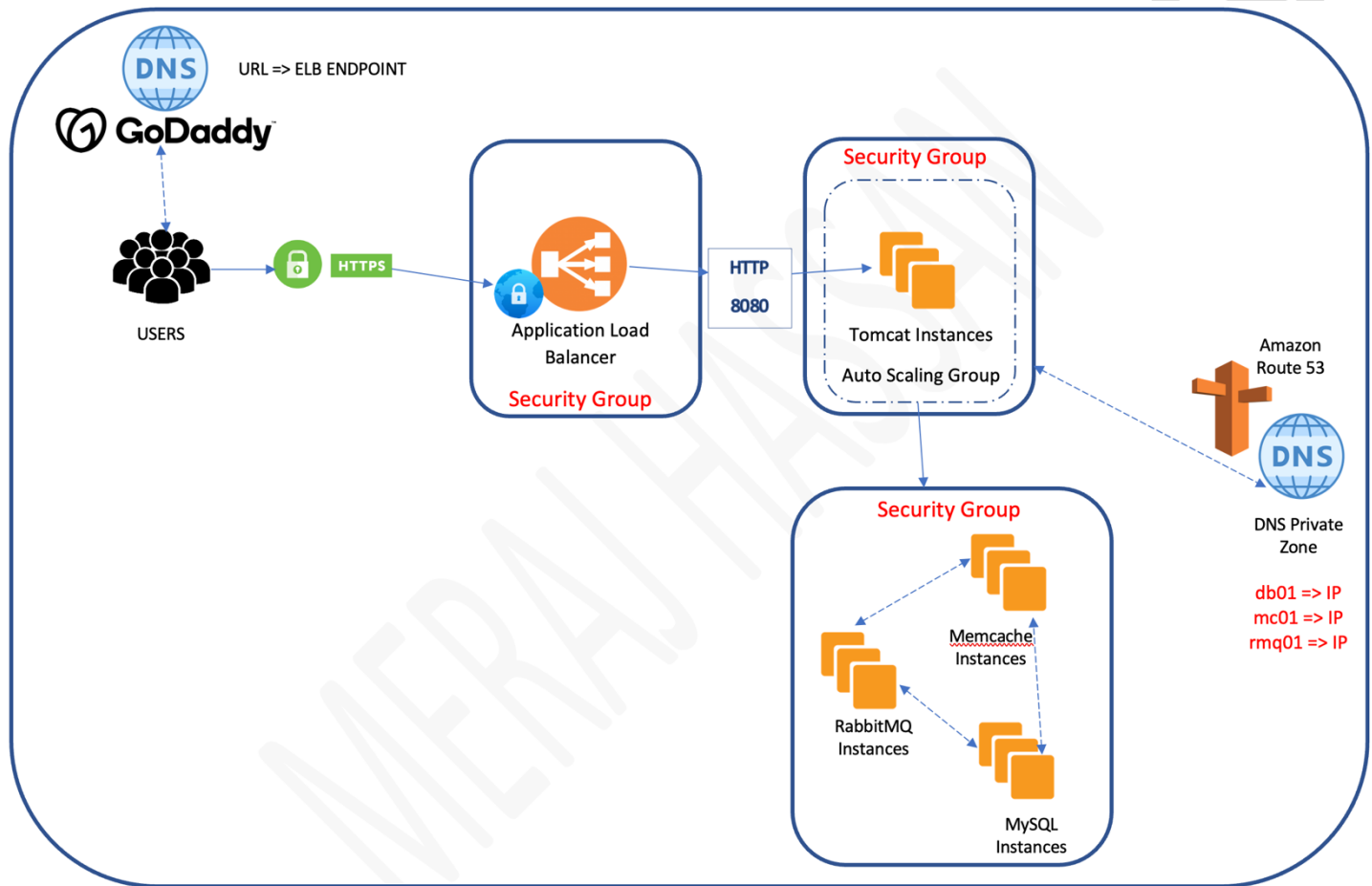**Architecture of AWS Services**

- *EC2 Instances*
- *ELB*
- *Autoscaling*
- *EFS/S3 for Shared Storage*
- *Amazon Certificate Manage*

- *Route 53*

Architecture of Automated Setup

- *VAGRANT*
- *VIRTUALBOX*
- *GITBASH*

Project Services Overview

Flow of Execution

- *Login to AWS Account*
- *Create Key Pairs*
- *Create Security Group*
- *Launch Instances with user data [BASH SCRIPT]*
- *Update IP to name mapping in Route 53*
- *Build Application from source code*
- *Upload to S3 Bucket*
- *Download artifact to TOMCAT EC2 instance*
- *Setup ELB with HTTPS [Cert from Amazon Certificate Manager]*

- *Map ELB Endpoint to Website name Godaday DNS*
- *Verify*
- *Build Autoscaling Group for Tomcat Instances*

Prerequisite

1. *AWS Account*
2. *Website from Godady*

Security Group and Key Pairs

$ git clone https://github.com/merajafnan/DevOps_Projects.git
$ git checkout aws-LiftShift
'userdate folder' contains bash script to create EC2 instances

$ chmod 400 eereeda-prod-key.pem
$ ssh -i eereeda-prod-key.pem centos@3.110.123.174
$ curl http://169.254.169.254/latest/user-data
$ systemctl status memcached
$ ssh -i eereeda-prod-key.pem centos@35.154.148.117
$ systemctl status rabbitmq-server
$ ssh -i eereeda-prod-key.pem centos@43.204.145.221
$ systemctl status mariadb

Private IPs:

Db01 172.31.15.198

Rbmq01 172.31.13.115

Mc01 172.31.0.151

Route 53

	Record name ▽	Type ▽	Routin... ▽	Differ... ▽	Value/Route traffic to ▽
☐	eereeda.in	NS	Simple	-	ns-1536.awsdns-00.co.uk. ns-0.awsdns-00.com. ns-1024.awsdns-00.org. ns-512.awsdns-00.net.
☐	eereeda.in	SOA	Simple	-	ns-1536.awsdns-00.co.uk. awsdns-hostmaster.am
☐	db01.eereeda.in	A	Simple	-	172.31.15.198
☐	mc01.eereeda.in	A	Simple	-	172.31.0.151
☐	rmq01.eereeda.in	A	Simple	-	172.31.13.115

Install Tomcat on Ubuntu Server

ch for services, features, blogs, docs, and more	[Option+S]		▣ ⬫ ⑦ Mumbai ▼	meraj @ 1628-5739-0402 ▼

Instances (1/4) **Info**   ⟳   Connect   Instance state ▼   Actions ▼   **Launch instances** ▼

🔍 Search   ‹ 1 › ⚙

☑	Name ▽	Instance ID	Instance state ▽	Instance type ▽	Status check	Alarm status	Availability Zone ▽
☐	ereeda-db01	i-0ef2ddfb944a44fd4	⊘ Running ⊕⊖	t2.micro	⊘ 2/2 checks passed	No alarms ＋	ap-south-1b
☐	eereeda-mc01	i-0fffe2584c66cc407	⊘ Running ⊕⊖	t2.micro	⊘ 2/2 checks passed	No alarms ＋	ap-south-1b
☐	eereeda-rmq01	i-0235dfc5642f1a465	⊘ Running ⊕⊖	t2.micro	⊘ 2/2 checks passed	No alarms ＋	ap-south-1b
☑	eereeda-app01	i-0f31a8a48d4a1ad08	⊘ Running ⊕⊖	t2.micro	⊘ 2/2 checks passed	No alarms ＋	ap-south-1b

src>main>resource>application.properties

make changes as per Route 53 DNS

```
#JDBC Configutation for Database Connection
jdbc.driverClassName=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://db01.eereeda.in:3306/accounts?useUnicode=true&characterEncoding=UTF
jdbc.username=admin
jdbc.password=admin123

#Memcached Configuration For Active and StandBy Host
#For Active Host
memcached.active.host=mc01.eereeda.in
memcached.active.port=11211
#For StandBy Host
memcached.standBy.host=127.0.0.2
memcached.standBy.port=11211

#RabbitMq Configuration
rabbitmq.address=rmq01.eereeda.in
rabbitmq.port=5672
rabbitmq.username=test
rabbitmq.password=test

#Elasticesearch Configuration
elasticsearch.host =192.168.1.85
elasticsearch.port =9300
elasticsearch.cluster=vprofile
elasticsearch.node=vprofilenode
```

Build Artifact:

$ brew tap AdoptOpenJDK/openjdk

$ brew install adoptopenjdk8 --cask

$ brew install maven

$ mvn install

```
[merajhassan@MERAJs-MacBook-Air ~ % ls Desktop/DevOps_Projects/target
classes                   maven-archiver            test-classes
generated-sources         maven-status              vprofile-v2
generated-test-sources    site                      vprofile-v2.war
jacoco.exec               surefire-reports
```

Upload artifact to AWS S3 Bucket:

$ brew install awscli

Create IAM user for AWS cli authentication – S3FullAccess

	User	Access key ID	Secret access key
▶ ✅	eereeda-s3-admin	AKIASL2YOSFBKK2MUQ54 📋	********* Show

```
merajhassan@MERAJs-MacBook-Air Downloads % cat new_user_credentials.csv
User name,Password,Access key ID,Secret access key,Console login link
eereeda-s3-admin,,AKIASL2YOSFBKK2MUQ54,aNPfhHyLj19W3t9pjwKynPR̶█████████r,https://162857390402.signin.aws.amazon.com/console
merajhassan@MERAJs-MacBook-Air Downloads % aws configure
AWS Access Key ID [None]: AKIASL2YOSFBKK2MUQ54
AWS Secret Access Key [None]: aNPfhHyLj19W3t9pjwKynPRz0/V+2█████
Default region name [None]: ap-south-1
Default output format [None]: json
```

$ cat new_user_credentials.csv

$ aws configure

$ aws s3 mb s3://eereeda-artifact-storage

$ aws s3 cp vprofile-v2.war s3://eereeda-artifact-storage/vprofile-v2.war

$ aws s3 ls s3://eereeda-artifact-storage

Create role - S3FullAccess

EC2 – Action – Security – Modify IAM Role – S3FullAccess

$ ssh -i eereeda-prod-key.pem centos@35.154.102.80

$ systemctl status tomcat8

$ cd /var/lib/tomcat8/webapps/

$ systemctl stop tomcat8

$ rm -rf ROOT/

$ apt install awscli

$ aws s3 ls s3://eereeda-artifact-storage

$ aws s3 cp s3://eereeda-artifact-storage/vprofile-v2.war /tmp/vprofile-v2.war

$ cd /tmp/

$ cp vprofile-v2.war /var/lib/tomcat8/webapps/ROOT.war

$ cd /var/lib/tomcat8/webapps/ROOT/WEB-INF/

$ systemctl start tomcat8

$ cd classes/

$ cat application.properties

$ telnet db01.eereeda.in 3306

Load Balancer and DNS:

1. **Target Group**

EC2 > Target groups

Target groups (1) Info

Name	ARN	Port	Protocol	Target type	Load balancer
eereeda-app-TB	arn:aws:elasticloadbalancin...	8080	HTTP	Instance	ⓘ None associate

2. Load Balancer

Copy DNS name and add CNAME record in Godady

11. References

The following table summarizes the documents referenced in this plan.

DOCUMENT NAME	INSTRUCTOR	LOCATION
DevOps Beginners to Advanced	Imran Teli	https://www.udemy.com/course/decodingdevops/learn/lecture/28273912?start=0#overview