

# **Multi-Tier Web Application Stack Setup locally on VMs and on AWS Cloud using EC2**

(Hands on Practice from Udemy)

Date: Feb 15, 2022

## **Scenario**

Consider a running project which has varieties of services that powers our run time. We are required to create set up of Multitier Java Web Application Stack consists of tomcat application server, MySQL db, memcache, rabbitmq, nginx on Virtual machine by using Oracle Virtual box and on AWS EC2.

## **Problems**

- Not comfortable in making changes in real servers.
- Local setup is complex
- Time consuming
- Not repeatable

## **Solution**

- Listed problems are overcome by creation of Local Set up on VM.
- It should be repeatable
- Automated
- Code ( IAAC)
- R&D in your own machine

## **Tools to be used**

- On VM, ORACLE VM Virtual box, Vagrant, Git Bash, VS Code Text editor
- On AWS,

## **Architecture of Project Services**

- NGINX, TOMCAT, Rabbit MQ, Memcached and MYSQL

## **Steps to be followed:**

### **For VM's**

- Download Oracle Virtual box and finish the initial process
- Keep the source code of java web application which runs on tomcat server ready
- Install gitbash on your local machine, install vagrant on that and configure.
- Bring up the vagrant to up virtual machines for below list of each services as per the required OS mentioned
  - 1) MySQL – CentOS
  - 2) Memcached – CentOS
  - 3) Rabbit MQ - CentOS
  - 4) Tomcat – Ubuntu

### 5) Nginx – Ubuntu

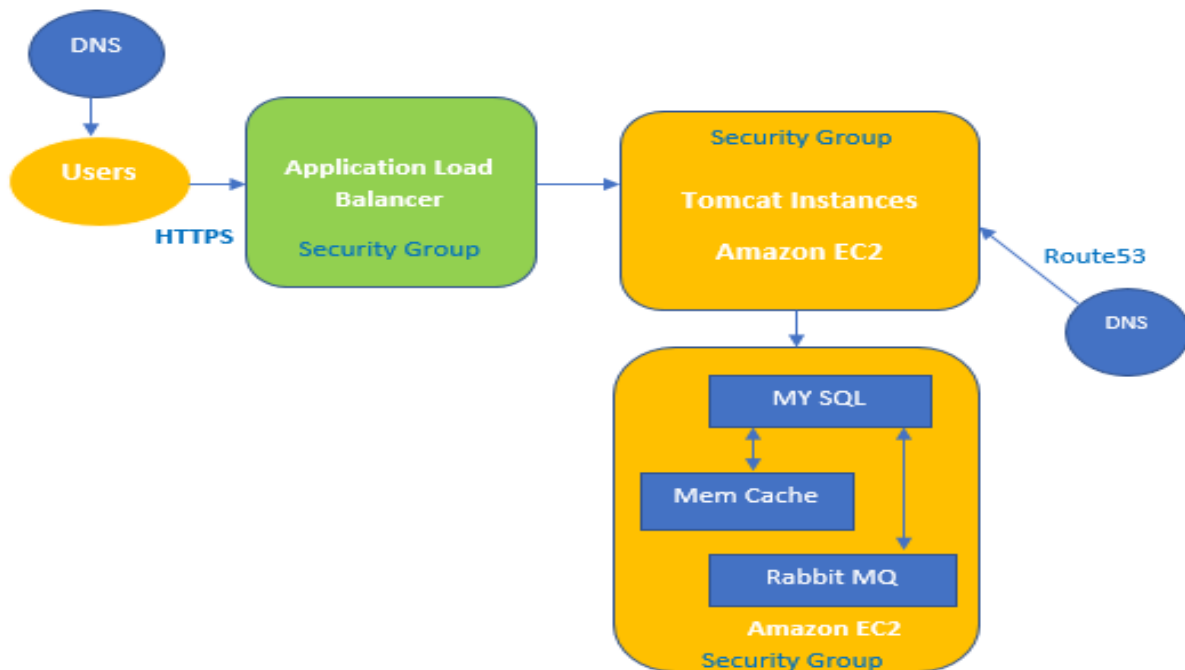
- App build and deploy it on VM.

#### For AWS EC2

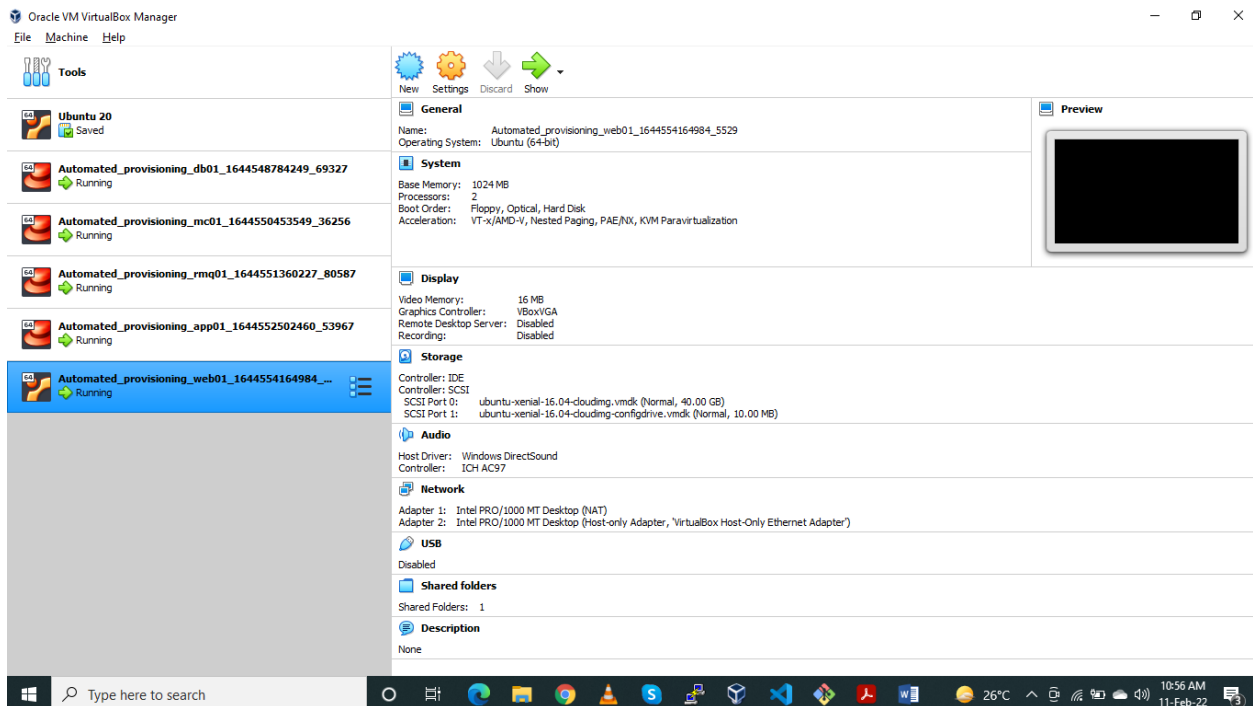
- Login to AWS account
- Create Key pairs
- Create security groups for EC2 machines which is to be configured with different services such as Tomcat, MYSQL, Rabbit MQ and Memcached
- Launch EC2 instances from console with user data (Bash script) for respective services.
- Update IP to name mapping to Route53
- Build application from Source code and get the artifacts in WAR format
- Upload the artifacts in S3 bucket
- Download artifact to tomcat Ec2 instance
- Setup Elastic load balancer with HTTPS from ACM
- Map ELB endpoint to website name in Go daddy DNS
- Verify it once done, by placing the endpoint on web browser and application should be in running status.

#### Workflow

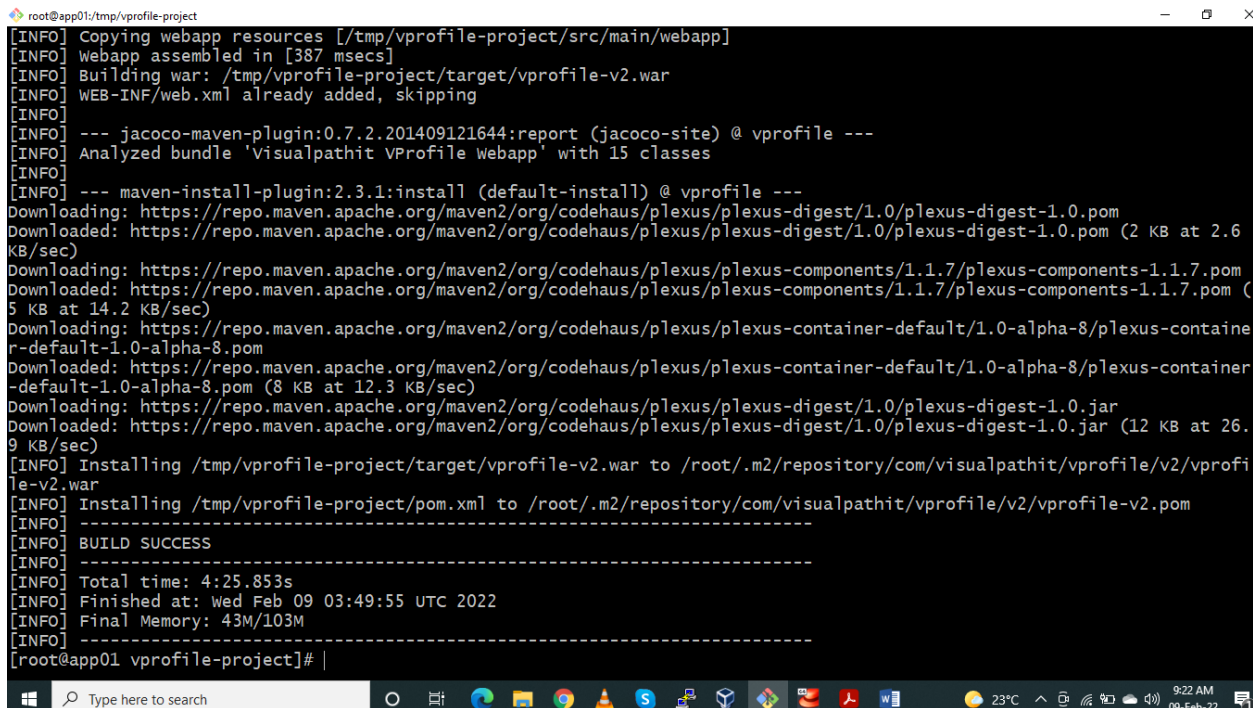
##### For AWS EC2



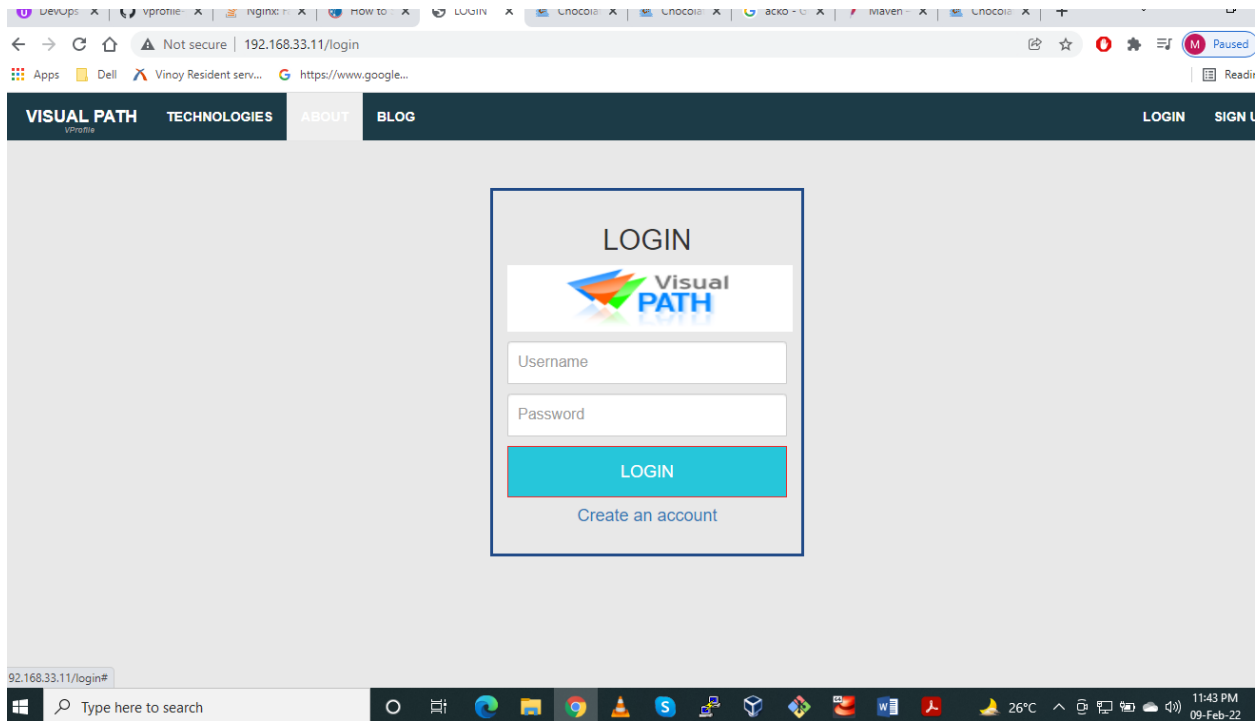
## Screenshot of vagrant up, and Virtual box initiates to launches the Virtual machine for mentioned services above



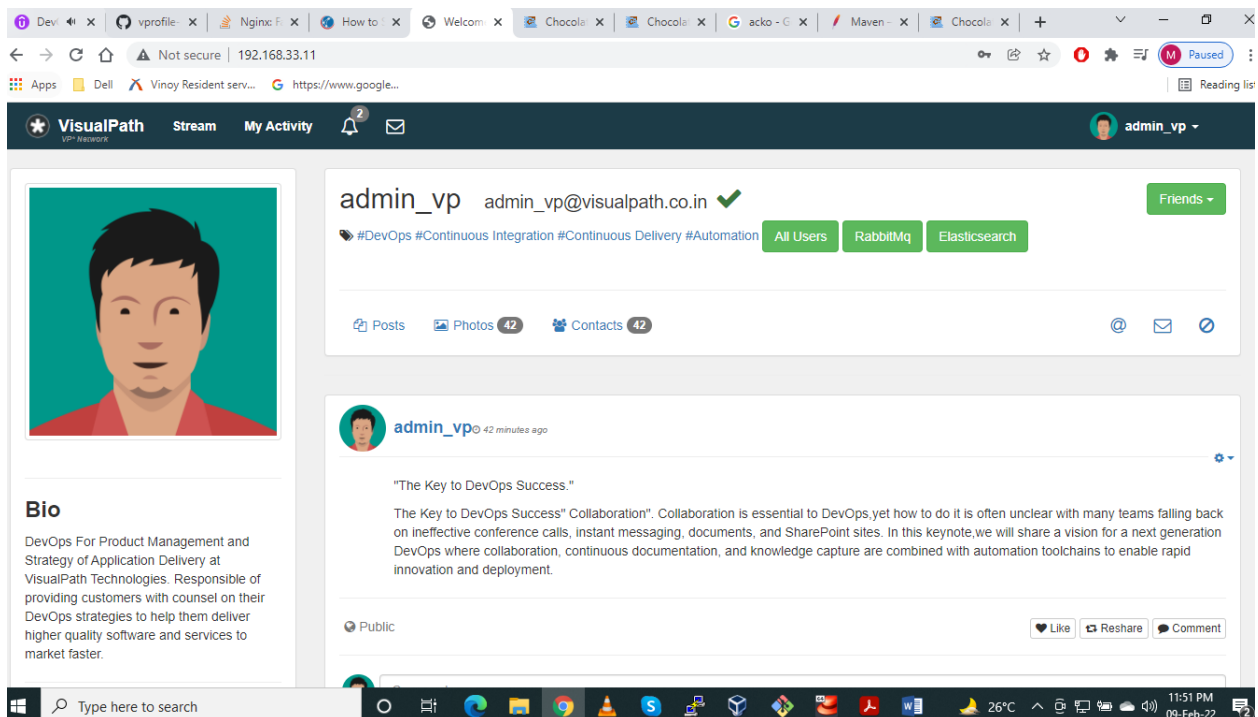
## Build output of screen shot from Maven for VM set up on Virtual box



*Screenshot of web application through browser with the local host id.*



*Screenshot of web application once logged in from home page, data pulled from mysql.*



## Screenshot of Memcache

The screenshot shows a web browser window with the URL `192.168.33.11/users/10`. The page has a dark blue header with navigation links: **VISUAL PATH**, **TECHNOLOGIES**, **ABOUT**, **CONTACT**, **BLOG**, **LOGIN**, and **SIGN UP**. The main content area is white and displays a message **[Data is From Cache]** with a **Back** button. Below this, the **User Primary Details** are shown in a table:

Id	Name	Father's Name	Mother's Name	Email	Phone Number
10	WahidKhan2	M Khan	R Khan	wahid.khan741@gmail.com	777777777

Below the primary details, the **User Extra Details** are shown in a table:

Date Of Birth	Gender	Marital Status	Permanent Address	Temporary Address	Primary Occupation	Secondary Occupation	Skills	Secondary PhoneNumber	Nationality	Language
28/03/1994	male	unMarried	Ameerpet,Hyderabad	Ameerpet,Hyderabad	Software Engineer	Software Engineer	Java HTML CSS	777777777	India	english

The browser's taskbar at the bottom shows the Windows search bar, taskbar icons, and system tray information: 27°C, 10:35 AM, 10-Feb-22.

## Screenshot of instance created for different services as Tomcat, MySQL database, rabbit MQ and memcached.

The screenshot shows the AWS Management Console for the `ap-south-1` region. The **Instances (4)** page is active, displaying a list of four running EC2 instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
vprofileapplication-mc01	i-077d84b539faceadb	Running	t2.micro	2/2 checks passed	No alarms
vprofileapplication-db01	i-0575b995f5cacf8e	Running	t2.micro	2/2 checks passed	No alarms
vprofileapplication-rmq01	i-0763922474d8ae0d6	Running	t2.micro	2/2 checks passed	No alarms
vprofileapplication-app01	i-0791b9247b409a9de	Running	t2.micro	2/2 checks passed	No alarms

The left sidebar shows the navigation menu with **Instances** selected. The bottom of the console shows the footer with copyright information: © 2022, Amazon Internet Services Private Ltd. or its affiliates. The Windows taskbar at the bottom shows the system tray information: 26°C, 11:36 AM, 15-Feb-22.

*Further screenshots of load balancer setup, maven build, web application running from web browser through load balancer endpoint and logged in page.*

The screenshot shows the AWS Management Console interface. On the left, the navigation menu is visible with categories like Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The 'Load Balancing' section is expanded, showing 'Load Balancers' and 'Target Groups'. The main content area displays the details for the 'vprofile-prod-elb' load balancer. A table lists the load balancer's Name, DNS name, State (Active), VPC ID, and Availability Zone. Below the table, a detailed view shows the Name, ARN, and DNS name (A Record) for the load balancer.

Name	DNS name	State	VPC ID	Avail
vprofile-prod-elb	vprofile-prod-elb-347856072.ap-south-1.elb.amazonaws.com	Active	vpc-07e5306c	ap-south-1

Details for vprofile-prod-elb:

- Name: vprofile-prod-elb
- ARN: arn:aws:elasticloadbalancing:ap-south-1:516828972978:loadbalancer/app/vprofile-prod-elb/4e971c94abdfa0c8
- DNS name: vprofile-prod-elb-347856072.ap-south-1.elb.amazonaws.com (A Record)

The screenshot shows a terminal window with the output of a Maven build command. The output displays the download progress of various dependencies from the central repository, including plexus-pom, plexus-digest, plexus-components, plexus-container-default, and plexus-utils. The build process is successful, and the final output shows the installation of the vprofile-v2.war file to the local repository.

```
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-pom/3.1/plexus-pom-3.1.pom (19 kB at 40 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.pom (1.1 kB at 2.1 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1.1.7/plexus-components-1.1.7.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1.1.7/plexus-components-1.1.7.pom (5.0 kB at 11 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-8/plexus-container-default-1.0-alpha-8.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-8/plexus-container-default-1.0-alpha-8.pom (7.3 kB at 15 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.jar (230 kB at 461 kB/s)
[INFO] Installing F:\vprofile-project\target\vprofile-v2.war to C:\Users\mailt\.m2\repository\com\visualpathit\vprofile\v2\vprofile-v2.war
[INFO] Installing F:\vprofile-project\pom.xml to C:\Users\mailt\.m2\repository\com\visualpathit\vprofile\v2\vprofile-v2.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 04:49 min
[INFO] Finished at: 2022-02-14T16:55:30+05:30
[INFO] -----
mailt@LAPTOP-C7PUA8R0 MINGW64 /f/vprofile-project (aws-LiftAndShift)
$ |
```

