**REPORT**

1.2:

The diagrams are created with using the AWS provided logos.

1.3

The steps in the tutorial were followed in order to create the EC2 instances and the associated LAMP server & the WordPress Blog.

1.4

The steps for creating an image was followed. The image wp-22 was created using which EC2 instance wp22 was also created. The answer to the discussion question is present in the slides.

1.5

A separate EC2 instance was created in which only MySQL was installed. The previously created EC2 instance WordPress configuration was modified in order to connect to this newly created EC2 instance hosting the database. The load balancer created in the previous task is now valid in this context.

The steps in the tutorial to create the Auto Scaling Group was followed. wp-asg was the group created. The previously created WordPress image wp-22 was used for the Launch Configuration of the Auto Scaling Group. Then, the Auto Scaling Group was connected to the Load Balancer.

**Challenges Faced:**

* Initially the default MySQL port 3306 was not exposed on the EC2 instance hosting the MySQL database. Hence the WordPress instances was not able to connect to the database. This issue was resolved by modifying the security group of the EC2 instance hosting the database. The privileges of the wordpress user in the database were modified to allow access from external IP addresses.

1.6

The Bitnami Image for running the WordPress on EC2 instance was deployed. Simple and straightforward deployment.

1.7

The concept of container virtualization that is implemented by Docker was familiarized by following tutorials provided in the official Docker website.

Steps to deploy WordPress instance on EC2 from a container image using Docker are provided in the Slides.

**Challenges Faced:**

* The familiarization and deployment of Docker was performed on a Windows machine which lead to the following issue: Upon successful deployment of the EC2 instance from a container image using Docker, we were unable to access the WordPress URL because the WordPress port wasn’t exposed on the Virtualized Linux machine. This issue was resolved by adding --expose & --net = host parameters in the docker run command.

1.8

We resorted to shell scripting for Configuration Management to deploy WordPress instance. The deployment steps are present in the video & ReadMe.

**Challenges Faced:**

* We faced issue in running Puppet Modules: Hunner Wordpress (<https://forge.puppet.com/hunner/wordpress>). The error was: Puppet could not find the declared class wordpress.