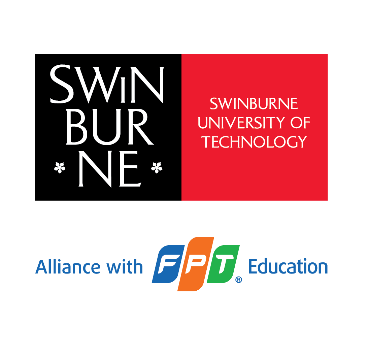
**SWINBURNE VIETNAM**

**HO CHI MINH CAMPUS**



**Class: COS40006**

**Deployment Portfolio Task 2**

**Instructor: Dr. Thomas Hang**

**Student names: Le Quang Hai**

# Task 2.1

## Create an SSH key pair

A screenshot of a computer

Description automatically generated



## Create an EC2 instance

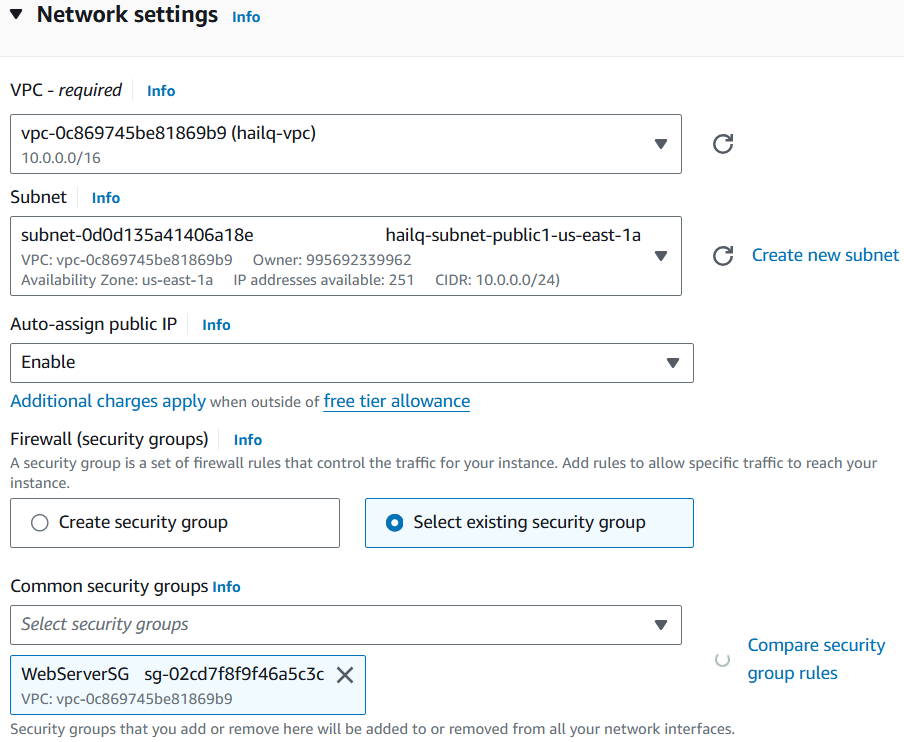
**Initiate instance creation**

A screenshot of a computer

Description automatically generated

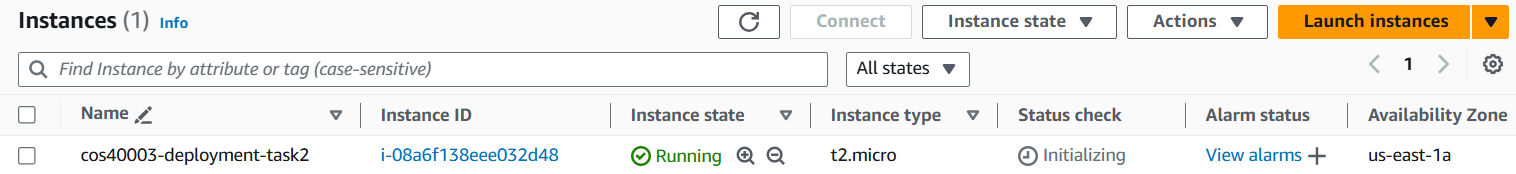


**Select VPC, located subnet, configure security group**

****



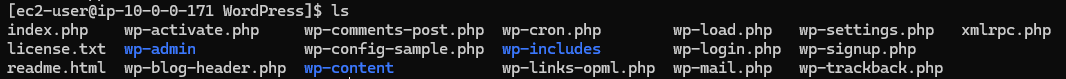
**For other settings, use default setting, then launch instance**

****



# Deploy a WordPress instance

**Load the source code to the EC2 instance**

****

**Create scripts to run the application**

**A screenshot of a computer program

Description automatically generated**



**Create an S3 bucket for storing source code**

**A screenshot of a computer

Description automatically generated**



**Set public policy for S3 bucket**

**A screenshot of a computer

Description automatically generated**



**Bundle the application's files into a single archive file and push the archive file**



**A screen shot of a computer

Description automatically generated**



**A screenshot of a computer

Description automatically generated**

**Create a deployment group**

**A screenshot of a computer

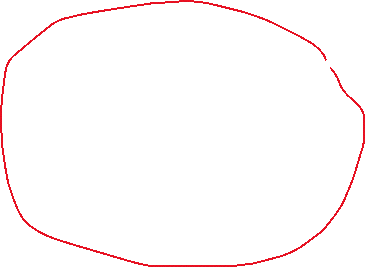
Description automatically generated**



**Successfully deploy the app**

A screenshot of a computer

Description automatically generated



**Access the application via http**

A screenshot of a computer

Description automatically generated



**Notes: the given script on Amazon website is compatible with Amazon Linux 2 EC2 instance; since we use the newest version Amazon Linux 2023, some script must be re-written.**

* For installing PHP use: sudo dnf install php php-cli php-common php-mysqlnd php-json php-xml -y
* For installing MariaDB use: sudo dnf install mariadb105-server -y

## Update the WordPress application

**Set up the site**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

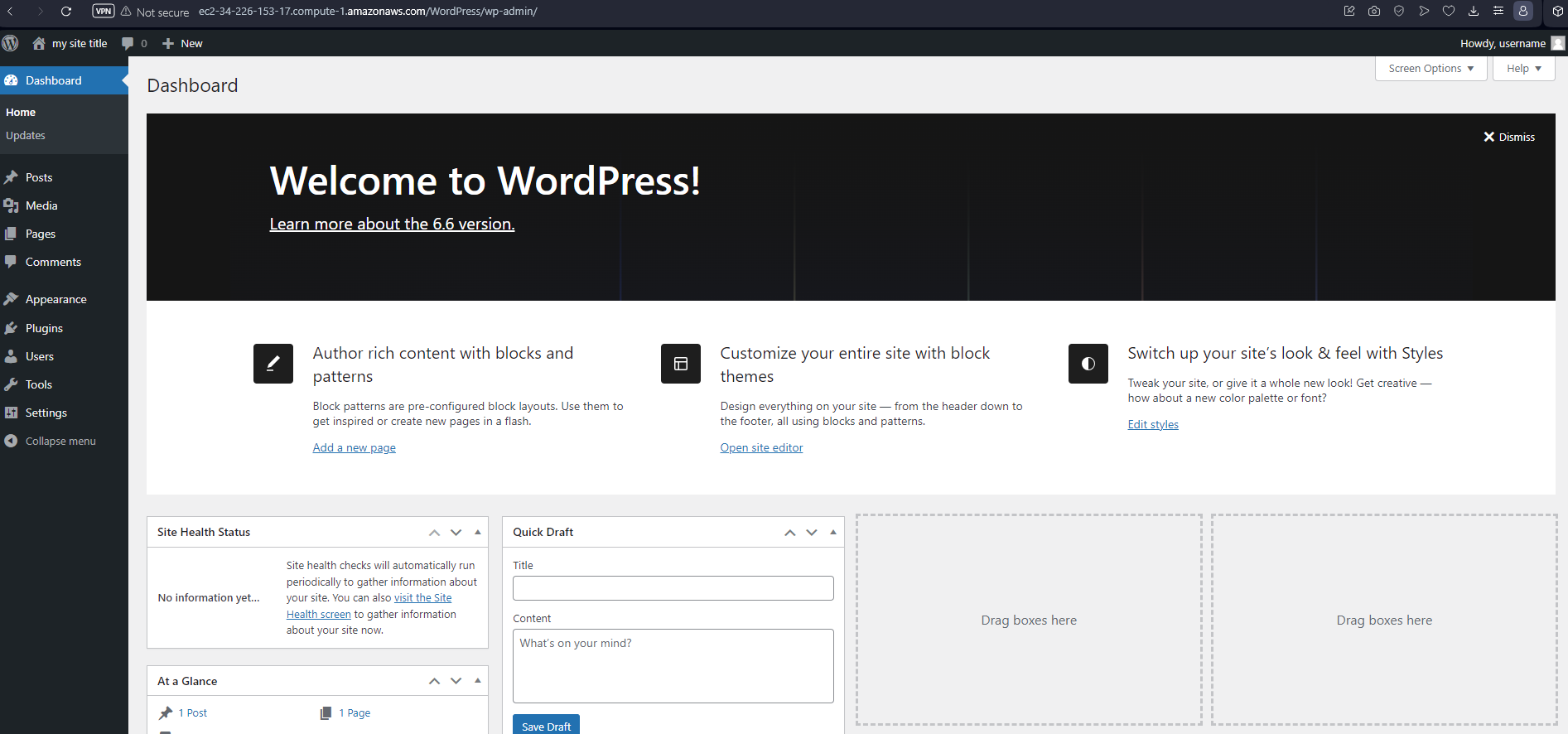


**Install WordPress**

**A screenshot of a computer

Description automatically generated**

**Successfully deploy**

****

**A screenshot of a computer

Description automatically generated**

# Task 2.2

## Create an ELB

**Create a target group, prepare for ELB creation**

**A screenshot of a computer

Description automatically generated**



**Create security group for ELB**

**A screenshot of a computer

Description automatically generated**



**Create an application ELB with the prepared resources**

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated



## Re-install WordPress using an external database

**Create an AWS RDS database instance**

**A screenshot of a computer

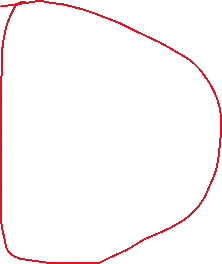
Description automatically generated**



**Modify the settings in the wp-config.php file corresponding to the RDS database instance**

**A computer screen with text on it

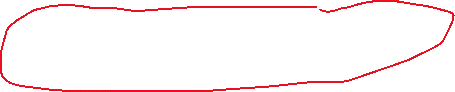
Description automatically generated**



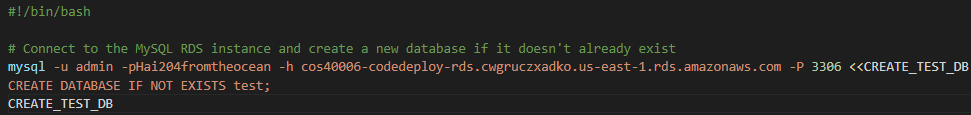
**Modify the scripts/install\_dependencies.sh file to install mysql-client instead of mariadb-server**

**A computer screen shot of a program

Description automatically generated**



**Modify the scripts/create\_test\_db.sh correspondingly to the RDS instance**

****

**Modify the scripts/start\_server.sh to no longer start a local database service**

**A screen shot of a computer

Description automatically generated**



**Deploy similarly to Task 2.1**

**A screenshot of a computer

Description automatically generated**



**-----------------------------------------------------------------------------------------------------------------------------**

**A screenshot of a website

Description automatically generated**

## Backup instance to S3

**To back up the instance, AWS provides the EBS snapshot creation. The snapshot is then stored in S3.**

**A screenshot of a computer

Description automatically generated**



**We create a schedule, specifically:**

* **A snapshot is taken every day at 12 am**
* **The snapshot remains in S3 for 24 hours**

**A screenshot of a computer

Description automatically generated**



**Now we ensure the backup is always the most up-to-date. For testing purposes, we create a snapshot manually**

**A screenshot of a computer

Description automatically generated**

## Create an instance from S3 backup

**From the snapshot taken above, create an AWS AMI out of it**

**A screenshot of a computer

Description automatically generated**



**From the AMI create we can create an EC2 instance with an exact copy of the files in the CodeDeployDemo instance**

**A screenshot of a computer

Description automatically generated**



**In order to listen on http request, the newly created instance must start httpd and php-frm service, add these commands to the data executed on boot up**

**A computer screen with text

Description automatically generated**

**Now the WordPress site should be available for http request**

**A screenshot of a computer

Description automatically generated**

# Task 2.3

## **Create a Launch Template (formerly Launch Configuration\* now deprecated)**

A screenshot of a computer

Description automatically generated



## Create an ASG

**Create an elastic application load balancer at layer 7 to prepare for ASG creation**

**A screenshot of a computer

Description automatically generated**



## Configure ASG to launch instance

A screenshot of a launch page

Description automatically generated



A screenshot of a computer

Description automatically generated



A screenshot of a computer screen

Description automatically generated



A screenshot of a computer screen

Description automatically generated



**After creating ASG, 2 instances is initialized as specified in settings**

**A screenshot of a computer

Description automatically generated**



## Test scaling

**On the instances, install “stress” with**

**A group of white text on a black background

Description automatically generated**

**Then overload CPU usage of the instances by**

****

**Now a new instance is automatically created**

**A screenshot of a computer

Description automatically generated**

# Task 2.4

## **Connecting to instances via SSH**

A screenshot of a computer screen

Description automatically generated

## **All the tasks above can be automated completely via AWS CLI**

Before starting, to avoid error, delete the existing resources (perhaps from last deployment, etc.)

A black background with white text

Description automatically generated

Create a AWS CodeDeploy Application via CLI



Push the current directory to S3 and attach it with the Application created via CLI



Create a deployment group for the application

A screenshot of a computer program

Description automatically generated

Deploy the application from the using that deployment group

A screen shot of a computer

Description automatically generated

Create a launch template via CLI

A black background with a line

Description automatically generated

Launch an EC2 instance from the created launch template



Create an application elastic load balancer

A computer screen with numbers and letters

Description automatically generated

Finally create an auto scaling group via CLI

A screen shot of a computer

Description automatically generated