SALIM HABIB UNIVERSITY

Department of Computer Science

Assignment # 2

Class/Section:	**BSCS-VIII**	Category	**Bash Script/AWS CLI**
Course Instructor:	**Engr. Gulraeez Gulshan**	Total Marks:	**[10]**
Date Due	**24-June-2022**	Time Due	**11:59 PM**
Course Title	**Cloud Computing**	Course Code	**CSC 412**

Pre-requisite Instruction:

1. Each task of the assignment will be carried out in one region i.e **us-east-1**.
2. It is required that you delete some of the mentioned resources that were already created/provisioned in **us-east-1** before starting off the assignment to get the desired script output.
   - Terminate all EC2 Instances if running or stopped.
   - Delete all keypairs.
   - Delete all security groups except the default one.
3. Create an EC2 instance in the **us-east-1** region with the following configuration:

• Name	:	**bash-server**
• Type	:	t2.micro
• AMI	:	Ubuntu 20.04
• Key-Pair	:	**assignment-02-kp**
• Availability-Zone	:	us-east-1e
• Security Group	:	**assignment-02-sg** [Allow Port 22, Port 80]

4. Connect to above EC2 instance using puTTy and perform all task inside this server.
5. Once connected, create a directory named **assignment-02** inside **/ubuntu/home** directory.
6. Switch inside the newly-created directory i.e **assignment-02** and perform all tasks.
7. Don’t delete any resources unless told by instructor.
8. You will need to share key-pair of EC2 to Instructor so that he can check your scripts.

[Task 01]

Install the NGINX Service on an ubuntu machine.
Script Name: **task_01.sh**

Objectives:

1. The script is only run when you run it as a **sudo** user.
2. The script installs the latest NGINX service.
3. If the NGINX is already installed, the script upgrades the NGINX but does not install it again.

[Task 02]

Verify that the NGINX service is active or inactive.
Script Name: **task_02.sh**

Objectives:

1. The script is only run when you run it as a **sudo** user.
2. If the NGINX service is active, the screen should echo in green colour "**NGINX is Running**"
3. If the NGINX service is inactive, the screen should prompt in red colour "**NGINX is Dead. Do you want to run NGINX [y/n]?**"
4. If the user press "**y**", it activates the NGINX, otherwise it remains dead.
5. In case of any error the screen should echo in red colour **"Something went wrong, NGINX cannot be activated"**

[Task 03]

Install latest **aws-cli** on ubuntu machine based on Linux architecture (x86 or ARM).
Script Name: **task_03.sh**

Objectives:

1. The script is only run when you run it as a **sudo** user.
2. The script installs **aws-cli** from the instructions provided in the AWS documentation.
   https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html
   (Don't use **apt-get install awscli**)
3. The script installs **aws-cli** based on the current architecture of the ubuntu machine i.e. x86 or ARM.
4. If the **aws-cli** is already installed, the script does not install it again, however it echoes on the screen "**aws-cli 2.7.9 is already installed in your machine**"

[Task 04]

Write a script to configure AWS credentials in the ubuntu machine
Script Name: **task_04.sh**

Objectives:

1. The script is only run when you run it as a **sudo** user.
2. The script installs **aws-cli** first if it is not installed.
3. The script configures the credentials of your admin account in the ubuntu machine.
4. **AWS_ACCESS_KEY_ID** and **AWS_SECRET_ACCESS_KEY** must be fetched from AWS SSM Service.

[Task 05]

Write a script to launch the two EC2 instances of the following configuration.
Script Name: **task_05.sh**

	Instance # 1	**Instance # 2**
Region	us-east-1	us-east-1
Instance Type	t2.micro	t2.micro
Key Name	assignment-02-kp	assignment-02-kp
Availability Zone	us-east-1a	us-east-1b
Instance Name	**ubuntu-server-01**	**ubuntu-server-02**
AMI	Ubuntu 20.04	Ubuntu 20.04
Security Group	assignment-02-sg	assignment-02-sg

Objectives:

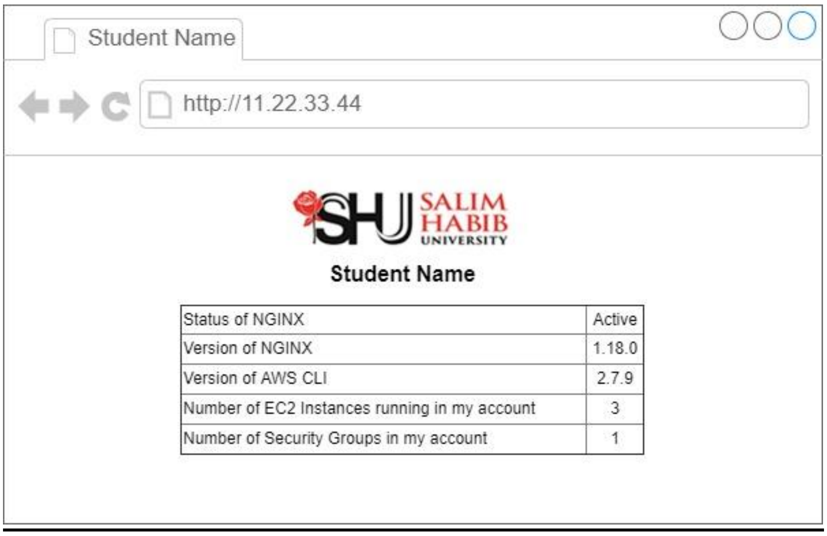
1. The script is only run when you run it as a **sudo** user.
2. The script installs **aws-cli** first if it is not installed.
3. The script configures the AWS credentials first, if not configured.

[Task 06]

Write a script to deploy a static webpage on EC2 Instance.
Script Name: **task_06.sh**

Objectives:

1. Make sure the static webpage is deployed to **bash-server** and accessible to its public IP.
2. The name of the webpage is **index.html.**
3. NGINX service will be used as web hosting server.
4. The webpage should display as following after dynamically getting values.



Student Name

Status of NGINX	Active
Version of NGINX	1.18.0
Version of AWS CLI	2.7.9
Number of EC2 Instances running in my account	3
Number of Security Groups in my account	1

[Task 07]

Write a script to download and deploy a static website on EC2 Instance.
Script Name: **task_07.sh**

Objectives:

1. The script creates an EC2 Instance of the following configuration.

	Instance
Region	us-east-1
Instance Type	t2.micro
Key Name	assignment-02-kp
Availability Zone	us-east-1c
Instance Name	**website-server**
AMI	Ubuntu 20.04
Security Group	assignment-02-sg

2. NGINX service will be used as a web hosting server.
3. The script downloads any free CSS template from **https://www.free-css.com/** on website-server**.**
4. The script must be dynamic so that it can download any template and deploy.
5. The script replaces the title of the **index.html** page with student name.

[Task 08]

Write a script to echo top cryptocurrencies listed on CoinGecko.
Script Name: **task_08.sh**

Objectives:

1. The following curl command can be used to download a list of top cryptocurrencies in JSON:

```
curl -o coins.json -X 'GET' \
        'https://api.coingecko.com/api/v3/coins/markets?vs_currency=pkr&order=market_cap_desc&per_page
=10&page=1&sparkline=false' \
        -H 'accept: application/json' &> /dev/null
```

2. The user can provide two arguments to script 1) **vs_currency**, 2) **per_page,** so the JSON output can be changed accordingly. (Note the yellow highlighted below)

```
curl -o coins.json -X 'GET' \
        'https://api.coingecko.com/api/v3/coins/markets?vs_currency=pkr&order=market_cap_desc&per_page
=10&page=1&sparkline=false' \
        -H 'accept: application/json' &> /dev/null
```

For example, if the arguments are passed as [`bash script.sh usd 5`], it will list the top 5 currencies with all values in USD currency.

3. Once you have downloaded the desired JSON file of top cryptocurrencies, now you can use "**jq**" tool to work with JSON in bash.
4. The output of the script when run as [**`bash script.sh usd 4`**], should be echoed like below:
(It is recommended to use **printf** for better formatting)

```
Rank     Name             Symbol           Current Price (USD)
 01      Bitcoin          btc              19147.22
 02      Ethereum         eth              996.98
 03      Tether           usdt             1.001
 04      USD Coin         usdc             1.003
```