**Assignments and Quizzes**

Certainly! Here are the questions with the correct answers:

1. \*\*How can you trigger a GitHub Actions workflow on a pull request event?\*\*

- a. on: workflow

- b. on: push

- c. on: merge\_request

- d. on: pull\_request

- \*\*Answer: d\*\*

2. \*\*What is the purpose of the "jobs" keyword in a GitHub Actions workflow?\*\*

- a. To set up environment variables

- b. To define individual tasks or steps to be executed

- c. To list contributors to the repository

- d. To specify the programming language used

- \*\*Answer: b\*\*

3. \*\*How can you securely store and use sensitive information like API keys in GitHub Actions?\*\*

- a. Hardcode in the workflow file

- b. Share in a public repository

- c. Use the "secrets" context

- d. Include in the README file

- \*\*Answer: c\*\*

4. \*\*What is the purpose of the "cache" action in GitHub Actions?\*\*

- a. To store workflow logs

- b. To speed up workflow execution by caching dependencies

- c. To archive artifacts

- d. To delete old workflow runs

- \*\*Answer: b\*\*

5. \*\*In a GitHub Actions workflow, what does a "job" consist of?\*\*

- a. The repository owner

- b. The programming language of the repository

- c. The version of GitHub Actions to use

- d. The action or container to be used in a job

- \*\*Answer: d\*\*

6. \*\*What is the purpose of "self-hosted runners" in GitHub Actions?\*\*

- a. To automate code reviews

- b. To manage issues

- c. To generate documentation

- d. To execute workflows on your own infrastructure

- \*\*Answer: d\*\*

7. \*\*How can you conditionally run a step based on the outcome of a previous step in GitHub Actions?\*\*

- a. Use the "if" keyword

- b. Set up a cron job

- c. Always run all steps

- d. Split the workflow into separate files

- \*\*Answer: a\*\*

8. \*\*In a GitHub Actions workflow, what does a "job" consist of?\*\*

- a. A branch of code

- b. A specific time zone

- c. A list of contributors

- d. A series of tasks

- \*\*Answer: d\*\*

9. \*\*How can you access the output of a previous job in a GitHub Actions workflow?\*\*

- a. Use the "outputs" keyword

- b. Use environment variables

- c. Manually copy from job to job

- d. It's not possible

- \*\*Answer: a\*\*

10. \*\*How can you visualize the GitHub Actions workflow run history?\*\*

- a. Use the GitHub Actions API

- b. Run a specific command in the terminal

- c. View the "Actions" tab in the GitHub repository

- d. Check the "Releases" section

- \*\*Answer: c\*\*

11. \*\*How can you share a custom action with the GitHub Actions community?\*\*

- a. Fork an existing action

- b. Create a new repository with the action

- c. Submit a pull request to GitHub

- d. Publish it to the GitHub Marketplace

- \*\*Answer: d\*\*

12. \*\*What is the default network used in docker?\*\*

- a. Host

- b. Bridge

- c. Custom

- d. None

- \*\*Answer: b\*\*

13. \*\*Which command/s lists docker images?\*\*

- a. docker -i

- b. docker images ls

- c. docker images

- d. Both b & c

- \*\*Answer: c\*\*

14. \*\*Which command shows the running processes of a container?\*\*

- a. docker top <ContainID>

- b. docker ps < ContainID >

- c. docker stop < ContainID>

- d. docker list < ContainID >

- \*\*Answer: a\*\*

15. \*\*Below can be used to run a command when a container is created.\*\*

- a. ENTRYPOINT

- b. CMD

- c. Both a & b

- d. None of these

- \*\*Answer: c\*\*

16. \*\*How to list only container IDs of running containers?\*\*

- a. docker ps -q

- b. docker ps -l

- c. docker ps -a

- d. docker ps -n 1

- \*\*Answer: a\*\*

17. \*\*How to check exposed ports of a container?\*\*

- a. docker top <ContainID>

- b. docker network <ContainID>.

- c. docker history <ContainID>

- d. docker port <ContainID>

- \*\*Answer: d\*\*

18. \*\*What is the default subnet range for docker containers?\*\*

- a. 172.17. 0.0/16

- b. 192.168.0.0/16

- c. 172.17. 0.0/24

- d. 192.168.0.0/24

- \*\*Answer: a\*\*

19. \*\*Which command shows the CPU and memory percentage used by containers?\*\*

- a. docker stats

- b. docker exec

- c. docker run

- d. docker ps

- \*\*Answer: a\*\*

20. \*\*Which option is used to create a new tag of an image in "docker build"?\*\*

- a. -t

- b. -a

- c. -n

- d. -g

- \*\*Answer: a\*\*

21. \*\*Which instruction in Dockerfile will download a file from the network during image creation?\*\*

- a. EXPOSE

- b. COPY

- c. FROM

- d. ADD

- \*\*Answer: d\*\*

22. \*\*Which command builds a new image from a Dockerfile?\*\*

- a. docker run

- b. docker create

- c. docker exec

- d. docker build

- \*\*Answer: d\*\*

23. \*\*Which orchestration tool handles multiple docker containers across different host machines?\*\*

- a. Docker Swarm

- b. Kubernetes

- c. Both a & b

- d. None of the above

- \*\*Answer: c\*\*

24. \*\*Which command is used to remove docker images?\*\*

- a. docker rm

- b. docker kill

- c. docker rmi

- d. Both a & c

- \*\*Answer: c\*\*

25. \*\*Which option is used to run the container in the background?\*\*

- a. -d

- b. -i

- c. -t

- d. -e

- \*\*Answer: a\*\*

26. \*\*Which command shows the number of images and containers?\*\*

- a. docker network

- b. docker kill

- c. docker info

- d. docker prune

- \*\*Answer: c\*\*

**Quiz 2 Solution**

Sure, let's break down each question:

### Question 1: systeminfo.sh

```bash

#!/bin/bash

# 1. Print kernel information

echo "Kernel Name: $(uname -s)"

echo "Kernel Release: $(uname -r)"

echo "Processor Type: $(uname -p)"

echo "Installed Operating System: $(uname -o)"

# 2. Print editor information

favorite\_editor="nano" # replace with your favorite editor

editor\_location=$(command -v $favorite\_editor)

echo "Favorite Editor: $favorite\_editor"

echo "Editor Location: $editor\_location"

```

\*\*Instructions:\*\*

- Replace "nano" with your favorite editor.

- Save the script as "systeminfo.sh".

- Make it executable: `chmod +x systeminfo.sh`.

- Run the script: `./systeminfo.sh`.

### Question 2: usergroup.sh

```bash

#!/bin/bash

# 1. Create user

sudo useradd devopsuser

# 2. Create group

sudo groupadd devopsgroup

# 3. Add user to group

sudo usermod -aG devopsgroup devopsuser

```

\*\*Instructions:\*\*

- Save the script as "usergroup.sh".

- Make it executable: `chmod +x usergroup.sh`.

- Run the script with sudo: `sudo ./usergroup.sh`.

### Question 3: sshconfig.sh

```bash

#!/bin/bash

# 1. Generate SSH key pair

ssh-keygen -t rsa -b 2048 -f ~/.ssh/id\_rsa -N ""

# 2. Disable password authentication

sudo sed -i 's/PasswordAuthentication yes/PasswordAuthentication no/' /etc/ssh/sshd\_config

sudo systemctl restart ssh

# 3. Set up SSH for passwordless login

echo -e "Host \*\n UseKeychain yes\n AddKeysToAgent yes\n IdentityFile ~/.ssh/id\_rsa" > ~/.ssh/config

# 4. SSH into the system non-interactively

ssh localhost 'mkdir ~/test && echo "I am doing the task2." > ~/test/filecreatedinnoninteractivemode.txt'

```

\*\*Instructions:\*\*

- Save the script as "sshconfig.sh".

- Make it executable: `chmod +x sshconfig.sh`.

- Run the script: `./sshconfig.sh`.

### Question 4: filedir.sh

```bash

#!/bin/bash

# 1. Create directory and file

sudo mkdir /devopsdir

sudo sh -c 'echo "AoA, Hello DevOps! " > /devopsdir/devopsfile.txt'

# 2. Set permissions

sudo chmod 644 /devopsdir/devopsfile.txt

sudo chmod 711 /devopsdir

```

\*\*Instructions:\*\*

- Save the script as "filedir.sh".

- Make it executable: `chmod +x filedir.sh`.

- Run the script with sudo: `sudo ./filedir.sh`.

### Question 5: backup.sh

```bash

#!/bin/bash

# 1. Backup script

backup\_dir="/home/devopsuser/backup"

timestamp=$(date +"%Y%m%d%H%M%S")

tar -zcvf $backup\_dir/backup\_$timestamp.tar.gz /devopsdir

# Bonus: Setup cron job

(crontab -l ; echo "\*/10 \* \* \* \* $PWD/backup.sh") | crontab -

```

\*\*Instructions:\*\*

- Save the script as "backup.sh".

- Make it executable: `chmod +x backup.sh`.

- Run the script: `./backup.sh`.

- Bonus: The script sets up a cron job to run the backup every 10 minutes.

### Question 6: transfer.sh

```bash

#!/bin/bash

# 1. Create tarball

tarball\_name=$(echo "$USER.tar.gz")

tar -zcvf $tarball\_name \*.sh

# 2. Copy to remote server

scp $tarball\_name your\_reg\_number@server\_ip:/home/your\_reg\_number/

# Replace 'your\_reg\_number' and 'server\_ip' with actual values

```

\*\*Instructions:\*\*

- Save the script as "transfer.sh".

- Make it executable: `chmod +x transfer.sh`.

- Run the script: `./transfer.sh`.

\*\*Note:\*\* Ensure that you replace placeholders like 'your\_reg\_number' and 'server\_ip' with your actual values.

Certainly! Here are the MCQs in the requested format:

### Question 1: systeminfo.sh

1. What does the following command print in a Bash script named "systeminfo.sh"?

```bash

echo "Kernel Name: $(uname -s)"

```

a. Display the processor type

b. Print the installed operating system

c. Show the kernel name

d. Print the kernel release

\*\*Answer: c\*\*

2. In a Bash script, what does the variable `favorite\_editor` hold in "systeminfo.sh"?

a. The editor location

b. The editor name

c. The kernel release

d. The processor type

\*\*Answer: b\*\*

### Question 2: usergroup.sh

3. What does the following command do in a Bash script named "usergroup.sh"?

```bash

sudo useradd devopsuser

```

a. Create a user named "devopsuser"

b. Add a user to a group

c. Remove a user

d. Change user password

\*\*Answer: a\*\*

4. In "usergroup.sh," what command creates a group named "devopsgroup"?

a. sudo groupadd devopsgroup

b. sudo addgroup devopsgroup

c. sudo creategroup devopsgroup

d. sudo newgroup devopsgroup

\*\*Answer: a\*\*

### Question 3: sshconfig.sh

5. What is the purpose of the following command in "sshconfig.sh"?

```bash

ssh-keygen -t rsa -b 2048 -f ~/.ssh/id\_rsa -N ""

```

a. Connect to a remote server

b. Generate an SSH key pair

c. Set up password authentication

d. Create a new user

\*\*Answer: b\*\*

6. How can you configure SSH to disable password authentication in "sshconfig.sh"?

a. sudo systemctl enable ssh

b. sudo sed -i 's/PasswordAuthentication yes/PasswordAuthentication no/' /etc/ssh/sshd\_config

c. ssh -N ""

d. ssh-keygen -p

\*\*Answer: b\*\*

### Question 4: filedir.sh

7. What does the following command achieve in "filedir.sh"?

```bash

sudo chmod 711 /devopsdir

```

a. Remove execute permissions

b. Give read and write permissions

c. Give read, write, and execute permissions

d. Restrict permissions to the owner

\*\*Answer: c\*\*

8. How do you create a file named "devopsfile.txt" in "/devopsdir" in "filedir.sh"?

a. touch /devopsdir/devopsfile.txt

b. echo "AoA, Hello DevOps! " > /devopsdir/devopsfile.txt

c. sudo createfile /devopsdir/devopsfile.txt

d. sudo cat > /devopsdir/devopsfile.txt

\*\*Answer: b\*\*

### Question 5: backup.sh

9. What does the following command do in "backup.sh"?

```bash

tar -zcvf $backup\_dir/backup\_$timestamp.tar.gz /devopsdir

```

a. Create a compressed archive of "/devopsdir"

b. Extract the contents of "/devopsdir"

c. Delete "/devopsdir"

d. Copy "/devopsdir" to another location

\*\*Answer: a\*\*

10. How can you set up a cron job for the backup script in "backup.sh"?

a. Add a line to /etc/crontab

b. Run the script with `cron setup`

c. Manually edit the crontab file

d. Add a line to /etc/cron.d/backup

\*\*Answer: c\*\*

### Question 6: transfer.sh

11. What does the following command do in "transfer.sh"?

```bash

tar -zcvf $tarball\_name \*.sh

```

a. Create a new script

b. Compress all Bash scripts into a tarball

c. Extract files from a tarball

d. Delete all Bash scripts

\*\*Answer: b\*\*

12. How do you copy the tarball to a remote server in "transfer.sh"?

a. scp $tarball\_name remote\_server:/home/username/

b. rsync $tarball\_name remote\_server:/home/username/

c. ssh-copy-id $tarball\_name remote\_server:/home/username/

d. sftp $tarball\_name remote\_server:/home/username/

\*\*Answer: a\*\*

**Assignment 1**

Certainly! Here are the Linux commands for the mentioned tasks:

### Task 1: Setting Up a Linux Environment on AWS

1. \*\*Create IAM User:\*\*

```bash

aws iam create-user --user-name cs423-devops

aws iam create-login-profile --user-name cs423-devops --password <your\_password> --password-reset-required

aws iam attach-user-policy --user-name cs423-devops --policy-arn arn:aws:iam::aws:policy/AdministratorAccess

aws iam create-access-key --user-name cs423-devops

```

2. \*\*Create VPC:\*\*

```bash

aws ec2 create-vpc --cidr-block <your\_cidr\_block>

aws ec2 create-subnet --vpc-id <your\_vpc\_id> --cidr-block <your\_subnet\_cidr\_block\_1>

aws ec2 create-subnet --vpc-id <your\_vpc\_id> --cidr-block <your\_subnet\_cidr\_block\_2>

aws ec2 create-route-table --vpc-id <your\_vpc\_id>

aws ec2 associate-route-table --subnet-id <your\_subnet\_id\_1> --route-table-id <your\_route\_table\_id>

aws ec2 associate-route-table --subnet-id <your\_subnet\_id\_2> --route-table-id <your\_route\_table\_id>

aws ec2 create-internet-gateway

aws ec2 attach-internet-gateway --vpc-id <your\_vpc\_id>

aws ec2 run-instances --image-id <your\_ami\_id> --subnet-id <your\_subnet\_id\_1> --instance-type t2.micro --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=Assignment1-EC2-1}]'

aws ec2 run-instances --image-id <your\_ami\_id> --subnet-id <your\_subnet\_id\_2> --instance-type t2.micro --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=Assignment1-EC2-2}]'

```

### Task 2: User and Group Management

1. \*\*Create Users:\*\*

```bash

sudo useradd cs423\_devops\_user\_1

sudo useradd cs423\_devops\_user\_2

# ... Repeat for cs423\_devops\_user\_3 to cs423\_devops\_user\_5

sudo passwd cs423\_devops\_user\_1 # Set password for the first user

```

2. \*\*Create Groups:\*\*

```bash

sudo addgroup fcse

sudo addgroup fee

# ... Repeat for fes, fme, fcve

sudo usermod -aG fcse cs423\_devops\_user\_1

sudo usermod -aG fee cs423\_devops\_user\_2

# ... Repeat for other groups and users

```

3. \*\*Delegate Permissions:\*\*

```bash

sudo visudo

# Add the following line for the second user to reboot

cs423\_devops\_user\_2 ALL=(ALL) /sbin/reboot

sudo usermod -aG sudo cs423\_devops\_user\_1

```

4. \*\*Generate SSH Keys:\*\*

```bash

sudo su cs423\_devops\_user\_1

ssh-keygen -t rsa -b 2048 -f ~/.ssh/cs423\_devops\_user\_1

# ... Repeat for other users

```

5. \*\*Configure SSH Connection:\*\*

```bash

nano ~/.ssh/config

```

Add the following for each user:

```

Host cs423\_devops\_user\_1

HostName <user\_1\_server\_ip>

User cs423\_devops\_user\_1

IdentityFile ~/.ssh/cs423\_devops\_user\_1

# ... Repeat for other users

```

6. \*\*Attach My Key:\*\*

```bash

cat <your\_keyfile.pub> >> ~/.ssh/authorized\_keys

```

7. \*\*SSH Connection Ports:\*\*

- Update `/etc/ssh/sshd\_config` to listen on ports 22 and 25.

### Task 3: Managing Files and Directories

1. \*\*Create Directories:\*\*

```bash

sudo mkdir /fcse /fee /fes /fme /fcve

```

2. \*\*Set Permissions:\*\*

```bash

sudo chown root:fcse /fcse

sudo chmod 770 /fcse

# ... Repeat for other groups and directories

sudo setfacl -m g:fcse:rx /fee /fes /fme /fcve

# ... Repeat for other groups

```

3. \*\*Use ACL Permissions:\*\*

- Use `setfacl` commands as needed.

4. \*\*File Creation and Permissions:\*\*

- Use `touch` and `nano` to create files in respective directories.

5. \*\*Use of Find Command:\*\*

```bash

find / -name "testfile.txt" # As root

sudo -u fcse find / -name "testfile.txt"

sudo -u some\_other\_user find / -name "testfile.txt"

```

6. \*\*File Manipulation:\*\*

- Use the provided Bash script and `tar`, `gzip`, `find`, and `delete` commands as instructed.

### Task 4: Updating and Installing Packages

1. \*\*Update Package Inventory:\*\*

```bash

sudo apt update > task4/task4-1.txt

```

2. \*\*Show Nano Details:\*\*

```bash

apt show nano > task4/task4-2.txt

```

3. \*\*Upgrade Nano Editor:\*\*

```bash

sudo apt upgrade nano

```

4. \*\*Install WordPress:\*\*

- Install Apache, PHP, MySQL/MariaDB, and set up WordPress as instructed.

### Task 5: Monitoring Resource Performance and Status

1. \*\*Hardware Information:\*\*

- Execute provided commands and redirect outputs to text files.

2. \*\*Process Management:\*\*

```bash

ps aux

sleep 10 &

jobs

top

```

3. \*\*File System and Memory Information:\*\*

```bash

df -h

```

```bash

df -h > <your\_reg\_number>\_mfsu.txt

```

```bash

free -h

```

### Submission

1. \*\*Bundle and Upload:\*\*

```bash

tar -czvf assignment1\_<your\_reg\_number>.zip -C ~ .

```

- Upload the tarball on Teams.

Certainly! Here are the Linux commands for the mentioned tasks:

### Task 1: Setting Up a Linux Environment on AWS

1. \*\*Create IAM User:\*\*

```bash

aws iam create-user --user-name cs423-devops

aws iam create-login-profile --user-name cs423-devops --password <your\_password> --password-reset-required

aws iam attach-user-policy --user-name cs423-devops --policy-arn arn:aws:iam::aws:policy/AdministratorAccess

aws iam create-access-key --user-name cs423-devops

```

2. \*\*Create VPC:\*\*

```bash

aws ec2 create-vpc --cidr-block <your\_cidr\_block>

aws ec2 create-subnet --vpc-id <your\_vpc\_id> --cidr-block <your\_subnet\_cidr\_block\_1>

aws ec2 create-subnet --vpc-id <your\_vpc\_id> --cidr-block <your\_subnet\_cidr\_block\_2>

aws ec2 create-route-table --vpc-id <your\_vpc\_id>

aws ec2 associate-route-table --subnet-id <your\_subnet\_id\_1> --route-table-id <your\_route\_table\_id>

aws ec2 associate-route-table --subnet-id <your\_subnet\_id\_2> --route-table-id <your\_route\_table\_id>

aws ec2 create-internet-gateway

aws ec2 attach-internet-gateway --vpc-id <your\_vpc\_id>

aws ec2 run-instances --image-id <your\_ami\_id> --subnet-id <your\_subnet\_id\_1> --instance-type t2.micro --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=Assignment1-EC2-1}]'

aws ec2 run-instances --image-id <your\_ami\_id> --subnet-id <your\_subnet\_id\_2> --instance-type t2.micro --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=Assignment1-EC2-2}]'

```

### Task 2: User and Group Management

1. \*\*Create Users:\*\*

```bash

sudo useradd cs423\_devops\_user\_1

sudo useradd cs423\_devops\_user\_2

# ... Repeat for cs423\_devops\_user\_3 to cs423\_devops\_user\_5

sudo passwd cs423\_devops\_user\_1 # Set password for the first user

```

2. \*\*Create Groups:\*\*

```bash

sudo addgroup fcse

sudo addgroup fee

# ... Repeat for fes, fme, fcve

sudo usermod -aG fcse cs423\_devops\_user\_1

sudo usermod -aG fee cs423\_devops\_user\_2

# ... Repeat for other groups and users

```

3. \*\*Delegate Permissions:\*\*

```bash

sudo visudo

# Add the following line for the second user to reboot

cs423\_devops\_user\_2 ALL=(ALL) /sbin/reboot

sudo usermod -aG sudo cs423\_devops\_user\_1

```

4. \*\*Generate SSH Keys:\*\*

```bash

sudo su cs423\_devops\_user\_1

ssh-keygen -t rsa -b 2048 -f ~/.ssh/cs423\_devops\_user\_1

# ... Repeat for other users

```

5. \*\*Configure SSH Connection:\*\*

```bash

nano ~/.ssh/config

```

Add the following for each user:

```

Host cs423\_devops\_user\_1

HostName <user\_1\_server\_ip>

User cs423\_devops\_user\_1

IdentityFile ~/.ssh/cs423\_devops\_user\_1

# ... Repeat for other users

```

6. \*\*Attach My Key:\*\*

```bash

cat <your\_keyfile.pub> >> ~/.ssh/authorized\_keys

```

7. \*\*SSH Connection Ports:\*\*

- Update `/etc/ssh/sshd\_config` to listen on ports 22 and 25.

### Task 3: Managing Files and Directories

1. \*\*Create Directories:\*\*

```bash

sudo mkdir /fcse /fee /fes /fme /fcve

```

2. \*\*Set Permissions:\*\*

```bash

sudo chown root:fcse /fcse

sudo chmod 770 /fcse

# ... Repeat for other groups and directories

sudo setfacl -m g:fcse:rx /fee /fes /fme /fcve

# ... Repeat for other groups

```

3. \*\*Use ACL Permissions:\*\*

- Use `setfacl` commands as needed.

4. \*\*File Creation and Permissions:\*\*

- Use `touch` and `nano` to create files in

respective directories.

5. \*\*Use of Find Command:\*\*

```bash

find / -name "testfile.txt" # As root

sudo -u fcse find / -name "testfile.txt"

sudo -u some\_other\_user find / -name "testfile.txt"

```

6. \*\*File Manipulation:\*\*

- Use the provided Bash script and `tar`, `gzip`, `find`, and `delete` commands as instructed.

### Task 4: Updating and Installing Packages

1. \*\*Update Package Inventory:\*\*

```bash

sudo apt update > task4/task4-1.txt

```

2. \*\*Show Nano Details:\*\*

```bash

apt show nano > task4/task4-2.txt

```

3. \*\*Upgrade Nano Editor:\*\*

```bash

sudo apt upgrade nano

```

4. \*\*Install WordPress:\*\*

- Install Apache, PHP, MySQL/MariaDB, and set up WordPress as instructed.

### Task 5: Monitoring Resource Performance and Status

1. \*\*Hardware Information:\*\*

- Execute provided commands and redirect outputs to text files.

2. \*\*Process Management:\*\*

```bash

ps aux

sleep 10 &

jobs

top

```

3. \*\*File System and Memory Information:\*\*

```bash

df -h

```

```bash

df -h > <your\_reg\_number>\_mfsu.txt

```

```bash

free -h

```

### Submission

1. \*\*Bundle and Upload:\*\*

```bash

tar -czvf assignment1\_<your\_reg\_number>.zip -C ~ .

```

- Upload the tarball on Teams.

Certainly! Here are the correct answers for the provided multiple-choice questions:

### Task 1: Setting Up a Linux Environment on AWS

1. \*\*How do you grant console access to an IAM user in AWS?\*\*

Answer: a. Attach a policy

2. \*\*What is the purpose of a VPC in AWS?\*\*

Answer: a. Virtualized Private Cloud

3. \*\*Which service provides internet access to resources in a VPC?\*\*

Answer: c. Internet Gateway

4. \*\*When launching EC2 instances, what is an AMI?\*\*

Answer: a. Amazon Machine Image

### Task 2: User and Group Management

5. \*\*How do you create a new user in Linux using the command line?\*\*

Answer: c. `useradd`

6. \*\*Which command is used to assign a user to a specific group?\*\*

Answer: b. `usermod`

7. \*\*How do you grant administrative privileges to a user using sudo?\*\*

Answer: c. `sudo usermod -aG sudo`

8. \*\*What is the purpose of an SSH config file?\*\*

Answer: d. Simplify SSH connections

### Task 3: Managing Files and Directories

9. \*\*What command is used to set up Linux permissions on a file or directory?\*\*

Answer: a. `chmod`

10. \*\*How can you use ACL permissions to grant access to a specific user?\*\*

Answer: c. `setfacl -m u:user:read`

### Task 4: Updating and Installing Packages

11. \*\*How do you upgrade a package in Ubuntu using the command line?\*\*

Answer: c. `apt upgrade`

12. \*\*Which command is used to install a package in Ubuntu?\*\*

Answer: c. `apt-get install`

### Task 5: Monitoring Resource Performance and Status

13. \*\*What command provides detailed information about CPU in Linux?\*\*

Answer: a. `cat /proc/cpuinfo`

14. \*\*How do you list all running processes in Linux?\*\*

Answer: c. `ps aux`

15. \*\*Which command shows information about mounted file systems?\*\*

Answer: a. `df -h`

16. \*\*What is the purpose of the `free` command in Linux?\*\*

Answer: b. Display free memory

### Bonus Question

17. \*\*How can you schedule a cron job to run every day at 3:30 AM?\*\*

Answer: a. `30 3 \* \* \*`

**Assignment 2**

Certainly! Here's a step-by-step guide with commands and explanations for each task in the assignment:

### Task 1: Selection and deployment of a simple open-source application on localhost

1. \*\*Clone the repository:\*\*

```bash

git clone https://github.com/yourusername/ReactNodeTesting.git

```

2. \*\*Follow instructions in Readme:\*\*

- Navigate to the cloned directory and follow the instructions in the Readme file to set up and run the application locally.

3. \*\*Initialize and push to new repository:\*\*

```bash

git init

git add .

git commit -m "Initial commit"

git remote add origin https://github.com/yourusername/your-new-repo.git

git push -u origin main

```

4. \*\*Add a collaborator:\*\*

- Go to your GitHub repository > Settings > Collaborators > Add collaborator (add your team member).

5. \*\*Make repository public and add collaborators:\*\*

- Go to your GitHub repository > Settings > General > Change repository visibility to "Public" and add your instructor as a collaborator.

6. \*\*Update repository settings:\*\*

- Go to your GitHub repository > Settings > Branches > Set "main" branch as the default branch.

- Go to your GitHub repository > Settings > Branches > Add a branch protection rule for "main" to prevent direct commits. Require pull request reviews before merging and include administrators.

### Task 2: Create two EC2 instances on AWS for Testing and Staging environment

1. \*\*Create EC2 instances:\*\*

- Use the AWS Management Console to create two EC2 instances with appropriate names, selecting Ubuntu Server 22.04 LTS (HVM) as the AMI.

2. \*\*Configure security group:\*\*

- Create a new security group or use an existing one and configure rules for both instances.

### Task 3: Automating application from development to deployment

1. \*\*Create a feature or fix branch:\*\*

```bash

git checkout -b feature-branch

```

2. \*\*Push changes and create a pull request:\*\*

```bash

git add .

git commit -m "Your changes"

git push origin feature-branch

```

3. \*\*GitHub Actions Workflow:\*\*

- Create a `.github/workflows/main.yml` file in your repository with the workflow configuration. Example:

```yaml

name: CI/CD Workflow

on:

pull\_request:

branches:

- main

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Checkout repository

uses: actions/checkout@v2

- name: Build project

run: |

# Your build commands here

- name: Unit testing

run: |

# Your unit testing commands here

- name: Code analysis/linting

run: |

# Your linting commands here

- name: Notification

run: |

# Your notification commands here

```

4. \*\*Trigger workflow manually:\*\*

- In your GitHub repository, go to the "Actions" tab, select your workflow, and click "Run workflow" to trigger the workflow manually.

5. \*\*QA Testing:\*\*

- QA tests the deployed project on the AWS testing server using the provided link.

6. \*\*Merge changes to main:\*\*

- Once QA confirms and merges changes into the main branch, it triggers the staging deployment workflow.

7. \*\*Staging Deployment Workflow:\*\*

- Similar to the testing deployment workflow but for the staging environment.

8. \*\*Access deployed project:\*\*

- All users can access the deployed project on the AWS staging server using the provided link.

### What to Hand In

1. \*\*Submit the zip file:\*\*

- Compress your selected project along with all created workflows and submit the zip file.

2. \*\*Submit the Word document:\*\*

- Include the following in your Word document:

- A short introduction about the project and your assignment objective.

- List down any challenges faced while deploying.

- A complete flow diagram of the assignment.

- Screenshots for each significant action performed in each task and subtask with appropriate captions.

Ensure that the document is organized, well-structured, and visually appealing.

1. How do you make a GitHub repository public?

a. Settings > General > Change repository visibility to "Public"

b. Settings > Collaborators > Make repository public

c. Repository > Make public

d. Repository > Settings > Set visibility to "Public"

\*\*Answer: a\*\*

2. What is the purpose of a security group in AWS EC2?

a. To manage IAM users

b. To control access to instances

c. To configure VPCs

d. To define IAM policies

\*\*Answer: b\*\*

3. In GitHub Actions, where do you define the workflow configuration?

a. `.github/main.yml`

b. `.github/config.yml`

c. `.github/workflows/main.yml`

d. `.workflow/main.yml`

\*\*Answer: c\*\*

4. How can you manually trigger a GitHub Actions workflow?

a. Run command in terminal: `github-actions run`

b. In Actions tab, select workflow, and click "Run workflow"

c. Add a new event in the workflow file

d. Push a new commit to the main branch

\*\*Answer: b\*\*

5. What is the primary purpose of a feature or fix branch in Git?

a. To store large binary files

b. To isolate changes for testing

c. To create a backup of the repository

d. To merge changes directly into main

\*\*Answer: b\*\*

6. Which Ubuntu version is specified for the EC2 instances in Task 2?

a. Ubuntu Server 20.04 LTS

b. Ubuntu Server 18.04 LTS

c. Ubuntu Server 22.04 LTS

d. Ubuntu Server 16.04 LTS

\*\*Answer: c\*\*

7. What is the role of the `.gitignore` file in a Git repository?

a. Specify collaborators

b. Ignore changes in the project

c. Define workflow configurations

d. Set repository visibility

\*\*Answer: b\*\*

8. How do you create a new branch in Git and switch to it?

a. `git branch new-branch`

b. `git switch -b new-branch`

c. `git checkout -b new-branch`

d. `git create new-branch`

\*\*Answer: c\*\*

9. What is the purpose of the `rsync` command in the context of copying files?

a. Remove files

b. Synchronize files between directories

c. Rename files

d. Archive files

\*\*Answer: b\*\*

10. Where do you configure branch protection rules in GitHub?

a. Repository > Settings > Branches

b. Repository > Collaborators > Rules

c. Workflow > Rules > Branch protection

d. Collaborators > Branch protection

\*\*Answer: a\*\*

11. How can you access the deployed project for testing on the AWS testing server?

a. http://<Instance\_IP>/testing

b. http://<Instance\_IP>/qa

c. http://<Instance\_IP>/dev

d. http://<Instance\_IP>

\*\*Answer: d\*\*

12. How do you copy the tarball to a remote server in "transfer.sh"?

a. `scp $tarball\_name remote\_server:/home/username/`

b. `rsync $tarball\_name remote\_server:/home/username/`

c. `ssh-copy-id $tarball\_name remote\_server:/home/username/`

d. `sftp $tarball\_name remote\_server:/home/username/`

\*\*Answer: a\*\*

13. In Task 2, which environment should have the same security group as the other?

a. Testing

b. Staging

c. Production

d. Development

\*\*Answer: a\*\*

14. What does the GitHub Actions workflow do when triggered by a pull request in Task 3?

a. Deploy changes to the production server

b. Trigger unit tests only

c. Deploy changes to the testing server

d. Send an email to the team

\*\*Answer: c\*\*

15. How can you ensure that no collaborator can commit directly to the main branch in GitHub?

a. Set branch protection rules

b. Change collaborators' permissions

c. Use a pre-commit hook

d. Disable direct commits in repository settings

\*\*Answer: a\*\*

16. What is the purpose of the `unit testing` step in the GitHub Actions workflow?

a. Deploy changes to the server

b. Verify individual components or functions

c. Perform static code analysis

d. Send email notifications

\*\*Answer: b\*\*

17. How can you add a collaborator to your GitHub project in Task 1?

a. Send an invitation link

b. Add their GitHub username in a text file

c. Collaborators > Add collaborator

d. Settings > Manage collaborators

\*\*Answer: c\*\*

18. Why is the `.github` directory used in the GitHub Actions workflow configuration?

a. To store screenshots

b. To organize workflow files

c. To define IAM policies

d. To configure VPCs

\*\*Answer: b\*\*

19. What is the significance of the `.gitignore` file in Task 1?

a. Preventing AWS key exposure

b. Ignoring changes in the GitHub Actions workflow

c. Ignoring specific files or directories

d. Configuring branch protection

\*\*Answer: c\*\*

20. How do you specify dependencies for a NodeJS project in Task 2?

a. Update the `.gitignore` file

b. Configure the security group

c. Edit the `.env` file

d. Update the `package.json` file

\*\*Answer: d\*\*

21. What should the link for QA testing on the AWS testing server look like?

a. http://<Instance\_IP>/qa

b. http://<Instance\_IP>/testing

c. http://<Instance\_IP>/dev

d. http://<Instance\_IP>

\*\*Answer: b\*\*

22. In Task 3, what triggers the workflow for deploying changes to the AWS staging server?

a. Pull request creation

b. Push to the main branch

c. A button in the workflow file

d. Sending an email to QA

\*\*Answer: b\*\*

**Assignment 3**

Certainly! Below is a guide with Linux commands for each step of the assignment:

### Task 1:

#### 1. Initialize Cloned Repository and Push to GitHub:

```bash

# Clone the existing repository

git clone <repository\_url>

# Navigate to the cloned repository

cd <repository\_name>

# Initialize a new Git repository

git init

# Add a new remote repository on GitHub

git remote add origin <new\_repository\_url>

# Add, commit, and push the changes

git add .

git commit -m "Initial commit"

git push -u origin main

```

#### 2. Add Collaborator and Make Repository Public:

```bash

# Add a collaborator to the GitHub project

# Navigate to the repository on GitHub, go to Settings > Collaborators, and add the collaborator's username or email.

# Make the repository public

# Navigate to the repository on GitHub, go to Settings > General > Danger Zone > Change repository visibility.

```

#### 3. Deploy Frontend and Backend Services Locally with Docker:

Assuming Docker is installed:

```bash

# Build Docker images for frontend and backend

docker build -t frontend-image ./frontend

docker build -t backend-image ./backend

# Run containers for frontend and backend

docker run -d -p 80:80 --name frontend-container frontend-image

docker run -d -p 3000:3000 --name backend-container backend-image

# Optionally, create a separate container for the database if needed

docker build -t db-image ./database

docker run -d -p 5432:5432 --name db-container db-image

```

### Task 2:

Refer to the instructions from Task 2 of the previous assignment for creating EC2 instances on AWS.

### Task 3:

#### 1. Dockerize the Application:

Assuming Dockerfiles are created for frontend, backend, and database:

```bash

# Build Docker images

docker build -t frontend-image ./frontend

docker build -t backend-image ./backend

docker build -t db-image ./database

```

#### 2. Use Docker Compose for Multi-Container Deployment:

Assuming a `docker-compose.yml` file is created:

```bash

# Use Docker Compose to deploy multi-container application

docker-compose up -d

```

#### 3. GitHub Actions Workflow:

Assuming a GitHub Actions workflow file (e.g., `.github/workflows/deploy.yml`) is created:

```yaml

# Workflow steps for deploying to AWS testing server

on:

pull\_request:

branches:

- main

jobs:

deploy\_to\_testing:

runs-on: ubuntu-latest

steps:

- name: Checkout Repository

uses: actions/checkout@v2

- name: Build Project

run: |

# Add commands to compile and build the project

- name: Unit Testing

run: |

# Add commands for running unit tests

- name: Code Analysis/Linting

run: |

# Add commands for code analysis/linting

- name: Notify on Success or Failure

run: |

# Add commands to send email notifications on success or failure

```

### Submission:

Follow the instructions to submit the zip file and Word document with the necessary details, flow diagrams, and screenshots.

Certainly! Here are some multiple-choice questions (MCQs) based on the assignment:

### Task 1:

1. \*\*How do you initialize a new Git repository for the cloned project?\*\*

a. git init

b. git clone

c. git start

d. git create

\*\*Answer: a\*\*

2. \*\*What command is used to add a new remote repository on GitHub?\*\*

a. git push

b. git remote add

c. git origin

d. git add origin

\*\*Answer: b\*\*

3. \*\*How can you make a GitHub repository public?\*\*

a. Settings > General > Danger Zone > Change repository visibility

b. Settings > Collaborators > Make Public

c. Repository > Make Public

d. Repository > Change Visibility

\*\*Answer: a\*\*

4. \*\*What command is used to build a Docker image for the frontend service?\*\*

a. docker build -t frontend-image ./frontend

b. docker create -t frontend-image ./frontend

c. docker compose -t frontend-image ./frontend

d. docker init -t frontend-image ./frontend

\*\*Answer: a\*\*

### Task 2:

5. \*\*In AWS, what is the purpose of naming EC2 instances appropriately?\*\*

a. Aesthetic purposes

b. No specific purpose

c. To improve security

d. To represent the environment

\*\*Answer: d\*\*

6. \*\*Which Amazon Machine Image (AMI) should be used for the EC2 instances?\*\*

a. Ubuntu Server 18.04 LTS

b. Ubuntu Server 22.04 LTS

c. Amazon Linux

d. CentOS 7

\*\*Answer: b\*\*

### Task 3:

7. \*\*What is the purpose of a Dockerfile in containerization?\*\*

a. Execute Docker commands

b. Define Docker volumes

c. Build Docker images

d. Manage Docker networks

\*\*Answer: c\*\*

8. \*\*Which command is used to run containers defined in a Docker Compose file?\*\*

a. docker start

b. docker up

c. docker run

d. docker compose

\*\*Answer: b\*\*

9. \*\*In a GitHub Actions workflow, what event triggers the workflow for deploying to the testing server?\*\*

a. pull\_request

b. push

c. deploy

d. merge

\*\*Answer: a\*\*

10. \*\*What is the purpose of the `runs-on` field in a GitHub Actions job?\*\*

a. Specify Docker image

b. Set up environment variables

c. Define the operating system for the job

d. Run a specific command

\*\*Answer: c\*\*

These questions cover various aspects of the assignment, including Git, Docker, AWS, and GitHub Actions.

Certainly! Here are some additional multiple-choice questions (MCQs) based on the extended assignment:

### Task 1:

11. \*\*When adding a collaborator to a GitHub repository, what permissions can you assign to them?\*\*

a. Read-only

b. Write-only

c. Admin-only

d. Read/Write

\*\*Answer: d\*\*

12. \*\*How can you deploy both frontend and backend services as separate containers using Docker Compose?\*\*

a. Separate Dockerfiles

b. Single Dockerfile

c. Docker Swarm

d. Docker Hub

\*\*Answer: a\*\*

### Task 2:

13. \*\*What is the role of the common security group for EC2 instances in different environments?\*\*

a. Restricting access between environments

b. Enforcing encryption standards

c. Managing container dependencies

d. Providing access to common resources

\*\*Answer: a\*\*

14. \*\*Why might you consider creating AWS instances in different accounts for a group project?\*\*

a. Cost management

b. Improved security

c. Simplified deployment

d. Resource optimization

\*\*Answer: a\*\*

### Task 3:

15. \*\*In Docker Compose, what is the purpose of defining networks and volumes for the application?\*\*

a. Isolate containers

b. Improve security

c. Simplify deployment

d. Manage dependencies

\*\*Answer: a\*\*

16. \*\*Which GitHub Actions event is triggered when changes are pushed to the main branch for deployment to the staging server?\*\*

a. pull\_request

b. push

c. deploy

d. merge

\*\*Answer: b\*\*

17. \*\*What does linting check for in the context of code analysis during a GitHub Actions workflow?\*\*

a. Code compilation errors

b. Code quality and style

c. Unit test failures

d. Network vulnerabilities

\*\*Answer: b\*\*

18. \*\*How does the GitHub Actions workflow notify stakeholders when it successfully deploys changes to the testing server?\*\*

a. Sends a text message

b. Posts on Slack

c. Sends an email

d. Opens a GitHub issue

\*\*Answer: c\*\*

19. \*\*What is the purpose of the `jobs` section in a GitHub Actions workflow file?\*\*

a. Define workflow triggers

b. Specify job dependencies

c. Set up environment variables

d. Define the individual tasks to be executed

\*\*Answer: d\*\*

20. \*\*Why is creating a visual flow diagram beneficial in the context of this assignment?\*\*

a. Aesthetic presentation

b. Documenting infrastructure

c. Required for submission

d. Identifying security vulnerabilities

\*\*Answer: b\*\*

These questions cover various aspects of Git, Docker, AWS, GitHub Actions, and the overall deployment process.

**Assignment 4**

**Task 1 – IAM User:**

1. **Create IAM User 'terraform-cs423-devops':**

bash

 aws iam create-user --user-name terraform-cs423-devops

 **Assign Administrator Policy:**

bash

1. aws iam attach-user-policy --user-name terraform-cs423-devops --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
2. **Generate User Credentials:**
   * This step is typically done through the AWS Management Console, but you need to ensure the generated credentials are securely stored.

**Task 2 – Networking:**

1. **Create VPC:**
   * Refer to your Terraform configuration files.
2. **Create Subnets, Route Tables, and Internet Gateway:**
   * Refer to your Terraform configuration files.

**Task 3 – Security Groups:**

1. **Create Security Group:**
   * Refer to your Terraform configuration files.

**Task 4 – Key Pair:**

1. **Create Key Pair:**

bash

1. aws ec2 create-key-pair --key-name cs423-assignment4-key --query 'KeyMaterial' --output text > cs423-assignment4-key.pem
2. chmod 400 cs423-assignment4-key.pem

**Task 5 – EC2 Instances:**

1. **Launch EC2 Instances:**
   * Refer to your Terraform configuration files.
2. **SSH Connection to EC2 Instance:**

bash

1. ssh -i cs423-assignment4-key.pem ubuntu@<instance-ip>
2. **Ensure Web Server is Running:**
   * After connecting to the instance, check the web server status.

**Task 6 – Outputs:**

1. **Define Outputs in Terraform Configuration:**
   * Refer to your Terraform configuration files.

**Submission:**

1. **Add Collaborator and Submit:**
   * Add the instructor as a collaborator to your GitHub repository.
2. **Zip Terraform Files:**

bash

 zip cs423\_assignment\_4.zip \*.tf

 **Submit on Teams:**

* Submit the zip file and the Word document with project details and architecture diagrams on Teams.

Certainly! Here are some possible Multiple-Choice Questions (MCQs) that could be relevant to the given Terraform assignment:

1. \*\*What is the purpose of the IAM user 'terraform-cs423-devops' in this assignment?\*\*

a. To deploy EC2 instances

b. To manage VPCs

c. To create security groups

d. All of the above

\*\*Answer: d\*\*

2. \*\*Which policy is assigned to the IAM user 'terraform-cs423-devops'?\*\*

a. EC2FullAccess

b. AdministratorAccess

c. VPCFullAccess

d. S3FullAccess

\*\*Answer: b\*\*

3. \*\*How do you generate the credentials for the IAM user 'terraform-cs423-devops'?\*\*

a. Automatically generated by AWS

b. Use `aws generate-credentials` command

c. Manually input during user creation

d. Credentials are not needed

\*\*Answer: a\*\*

4. \*\*In Task 2, what is the purpose of the VPC named 'devops-assignment-4'?\*\*

a. To store IAM users

b. To host a simple web server

c. To manage security groups

d. To create EC2 instances

\*\*Answer: d\*\*

5. \*\*What does a private route table ensure in Task 2?\*\*

a. Allows internet access to resources

b. Denies internet access to resources

c. Manages IAM policies

d. Provides VPN access

\*\*Answer: b\*\*

6. \*\*In Task 3, what is the primary purpose of the Security Group created?\*\*

a. Control inbound and outbound traffic for EC2 instances

b. Manage IAM user permissions

c. Define VPC configurations

d. Monitor CloudWatch metrics

\*\*Answer: a\*\*

7. \*\*What is the significance of the Key Pair named 'cs423-assignment4-key'?\*\*

a. For EC2 instance monitoring

b. For IAM user authentication

c. For secure SSH connection to EC2 instances

d. For VPC encryption

\*\*Answer: c\*\*

8. \*\*Which EC2 instance is configured for hosting a simple web server?\*\*

a. Assignment4-EC2-1

b. Assignment4-EC2-2

c. Both

d. None

\*\*Answer: a\*\*

9. \*\*How do you SSH into the EC2 instance using the created Key Pair?\*\*

a. `ssh -i cs423-assignment4-key.pem ubuntu@<instance-ip>`

b. `ssh -u cs423-assignment4-key.pem ubuntu@<instance-ip>`

c. `ssh -p 80 cs423-assignment4-key.pem ubuntu@<instance-ip>`

d. `ssh -key cs423-assignment4-key.pem ubuntu@<instance-ip>`

\*\*Answer: a\*\*

10. \*\*What information is included in the Outputs in Task 5?\*\*

a. IAM user details

b. Public and private IP addresses of EC2 instances

c. Both a and b

d. None of the above

\*\*Answer: c\*\*

Remember to review the actual Terraform configurations and adjust questions based on specific details in your assignment.

Certainly! Here are 10 more Multiple-Choice Questions (MCQs) based on the given Terraform assignment:

11. \*\*What is the CIDR block for the VPC named 'devops-assignment-4' in Task 2?\*\*

a. 10.0.0.0/8

b. 192.168.1.0/24

c. 172.16.0.0/16

d. <address of your choice>/16

\*\*Answer: d\*\*

12. \*\*What is the purpose of the private subnet in Task 2?\*\*

a. Hosts the web server

b. Allows internet access

c. Configures IAM roles

d. No access to the internet

\*\*Answer: d\*\*

13. \*\*In Task 3, which principle is followed when configuring the security group for EC2 instances?\*\*

a. Least Privilege

b. Full Access

c. Open Access

d. Default Access

\*\*Answer: a\*\*

14. \*\*What does the user\_data.sh script do in Task 4 for configuring EC2 instances?\*\*

a. Installs Docker

b. Sets up IAM roles

c. Configures routing tables

d. None of the above

\*\*Answer: a\*\*

15. \*\*What is the primary purpose of the internet gateway created in Task 2?\*\*

a. Allows communication between subnets

b. Provides a secure connection to the internet

c. Enables access to resources in VPC

d. Manages IAM policies

\*\*Answer: b\*\*

16. \*\*How do you ensure that only necessary ports are open in the security group for the web server in Task 3?\*\*

a. Define rules for all ports

b. Use the default security group

c. Follow the principle of least privilege

d. Open all ports and restrict at the OS level

\*\*Answer: c\*\*

17. \*\*What is the role of the IAM user in Task 1 after its creation?\*\*

a. Hosts the web server

b. Manages the VPC

c. Authenticates SSH connections

d. Manages AWS resources through the console

\*\*Answer: d\*\*

18. \*\*Which file should contain the Terraform configurations for creating EC2 instances?\*\*

a. main.tf

b. ec2.tf

c. instances.tf

d. aws.tf

\*\*Answer: a\*\*

19. \*\*What is the significance of creating multiple subnets in different Availability Zones in Task 2?\*\*

a. Load balancing

b. High availability and fault tolerance

c. IAM user management

d. Redundant key pairs

\*\*Answer: b\*\*

20. \*\*How do you verify if the simple web server is running on the EC2 instance in Task 4?\*\*

a. Check the CloudWatch metrics

b. Use the AWS CLI

c. Visit http://<instance-ip>:80 in a web browser

d. Connect through SSH and run a command

\*\*Answer: c\*\*

Feel free to adjust or customize these questions based on the specific details and requirements of your assignment.