STUDENT VERSION (DevOps-Week-2)

TEAMWORK

CLARUSWAY
WAY TO REINVENT YOURSELF

Meeting Agenda

- ▶ Icebreaking
- ▶ Microlearning
- ▶ Questions
- ▶ Interview/Certification Questions
- ▶ Coding Challenge
- ▶ Article of the week
- ▶ Video of the week
- ▶ Retro meeting
- ▶ Case study / project

Teamwork Schedule

Ice-breaking 5m

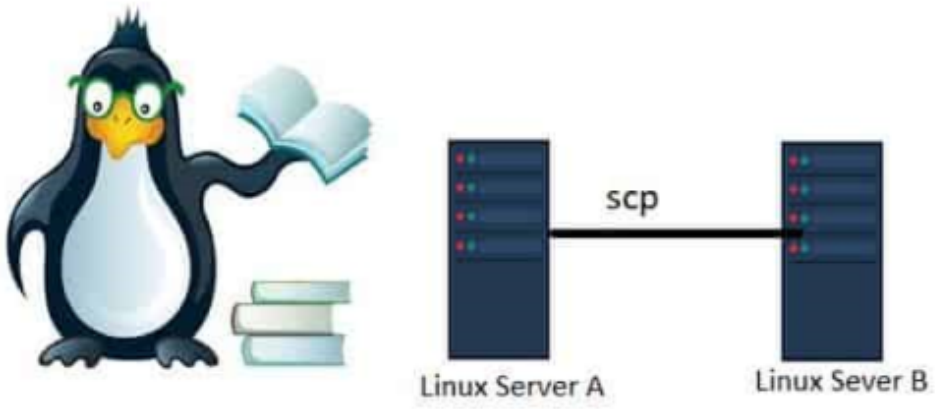
- Personal Questions (Stay at home & Corona, Study Environment, Kids etc.)
- Any challenges (Classes, Coding, AWS, studying, etc.)
- Ask how they're studying, give personal advice.
- Remind that practice makes perfect.

Team work 10m

- Ask what exactly each student does for the team, if they know each other, if they care for each other, if they follow and talk with each other etc.

Microlearning 15m

How to Use SCP Command



SCP (secure copy) is a command-line utility that allows you to securely copy files and directories between two locations. SCP uses by default the port 22, and connect via an encrypted connection or secure shell connection (ssh).

With scp, you can copy a file or directory:

- From your local system to a remote system.

- From a remote system to your local system.
- Between two remote systems from your local system.

When transferring data with scp, both the files and password are encrypted so that anyone snooping on the traffic doesn't get anything sensitive.

SCP Command Syntax:

The scp command syntax take the following form:

```
scp [OPTION] [user@]SRC_HOST:]file1 [user@]DEST_HOST:]file2
```

`OPTION` : scp options such as cipher, ssh configuration, ssh port, limit, recursive copy ...etc.

`[user@]SRC_HOST:]file1` : Source file.

`[user@]DEST_HOST:]file2` : Destination file.

Local files should be specified using an absolute or relative path, while remote file names should include a user and host specification.

scp provides a number of options that control every aspect of its behavior. The most widely used options are:

`-P` : Specifies the remote host ssh port.
`-p` : Preserves files modification and access times.
`-q` : Use this option if you want to suppress the progress meter and non-error messages.
`-C` : This option forces scp to compresses the data as it is sent to the destination machine.
`-r` : This option tells scp to copy directories recursively.

The `colon (:)` is how scp distinguish between local and remote locations.
To be able to copy files, you must have at least read permissions on the source file and write permission on the target system.

1. Copy a Local File to a Remote System with the scp Command:

To copy a file from a local to a remote system run the following command:

```
scp file.txt remote_username@10.10.0.2:/remote/directory
```

`file.txt` is the name of the file we want to copy, `remote_username` is the user on the remote server, `10.10.0.2` is the server IP address. The `/remote/directory` is the path to the directory you want to copy the file to.
If you don't specify a remote directory, the file will be copied to the remote user home directory.

You will be prompted to enter the user password, and the transfer process will start.

Omitting the filename from the destination location copies the file with the original name. If you want to save the file under a different name, you need to specify the new file name:

```
scp file.txt remote_username@10.10.0.2:/remote/directory/newfilename.txt
```

If SSH on the remote host is listening on a port other than the default 22 then you can specify the port using the -P argument:

```
scp -P 2322 file.txt remote_username@10.10.0.2:/remote/directory
```

The command to copy a directory is much like as when copying files. The only difference is that you need to use the -r flag for recursive.

To copy a directory from a local to remote system, use the -r option:

```
scp -r /local/directory remote_username@10.10.0.2:/remote/directory
```

- if you use pem key, you need the add it;

```
scp tyler.pem file.txt remote_username@10.10.0.2:/remote/directory
```

2. Copy a Remote File to a Local System using the scp Command:

To copy a file from a remote to a local system, use the remote location as a source and local location as the destination.

For example to copy a file named **file.txt** from a remote server with IP 10.10.0.2 run the following command:

```
scp remote_username@10.10.0.2:/remote/file.txt /local/directory
```

If you haven't set a passwordless SSH login to the remote machine, you will be asked to enter the user password.

3. Copy a File Between Two Remote Systems using the scp Command:

When using scp you don't have to log in to one of the servers to transfer files from one to another remote machine.

The following command will copy the file /files/file.txt from the remote host host1.com to the directory /files on the remote host host2.com.

```
scp user1@host1.com:/files/file.txt user2@host2.com:/files
```

You will be prompted to enter the passwords for both remote accounts.

To route the traffic through the machine on which the command is issued, use the -3 option:

```
scp -3 user1@host1.com:/files/file.txt user2@host2.com:/files
```

Ask Questions 15m

1. Which command is used to terminate the Terraform-managed infrastructure?

A. terraform terminate
B. terraform erase
C. terraform delete
D. terraform destroy

2. Which command is used to list of the resources in state in Terraform?

A. terraform state --list
B. terraform show list
C. terraform state list
D. terraform ls state

3. A Jira _____is a set os statuses and transitions that an issuw moves through during its lifecycl.

A. epic
B. version
C. workflow
D. report

4. A(n) _____ is a set of jobs that can be divided into manageable user stories.

A. tasks
B. subtasks
C. epic
D. bug

5. Put the steps for starting a sprint in the proper order

(1) Create the Sprint

(2) Start the Sprint

(3) Fill the Sprint Backlog

(4) Fill the Product Backlog

A. 1, 2, 3, 4
B. 4, 1, 2, 3
C. 3, 2, 1, 4
D. 4, 1, 3, 2

Interview/Certification Questions 20m

1. What are the different phases in DevOps?

Plan, code, build, test, release, deploy, operate, monitor and continue

2. Explain the concept behind Infrastructure as Code (IaC).

Kod Olarak Altyapı (IaC), altyapının manuel süreçler yerine kod aracılığıyla yönetilmesi ve sağlanmasıdır.

Infrastructure as Code (IaC) is the managing and provisioning of infrastructure through code instead of through manual processes.

3. How Terraform works?

Terraform allows users to define their entire infrastructure simply by using configuration files and version control. When a command is given to deploy and run a server, database or load balancer, Terraform parses the code and translates it into an application programming interface (API) call to the resource provider.

Terraform, kullanıcıların yalnızca yapılandırma dosyalarını ve sürüm kontrolünü kullanarak tüm altyapılarını tanımlamasına olanak tanır. Bir sunucuyu, veritabanını veya yük dengeleyiciyi dağıtmak ve çalıştırmak için bir komut verildiğinde, Terraform kodu ayrıştırır ve kaynak sağlayıcıya yapılan bir uygulama programlama arabirimi (API) çağrısına çevirir.

4. Why JIRA is used?

Jira Software is part of a family of products designed to help teams of all types manage work. Originally, Jira was designed as a bug and issue tracker.
But today, Jira has evolved into a powerful work management tool for all kinds of use cases, from requirements and test case management to agile software development.

Jira Software, her türden ekibin işi yönetmesine yardımcı olmak için tasarlanmış bir ürün ailesinin parçasıdır. Başlangıçta, Jira bir hata ve sorun izleyici olarak tasarlandı. Ancak bugün Jira, gereksinimler ve test senaryosu yönetiminden çevik yazılım geliştirmeye kadar her türlü kullanım senaryosu için güçlü bir iş yönetimi aracına dönüşmüştür.

5. What is an issue in JIRA?

Issues are the building blocks of any Jira project. An issue could represent a story, a bug, a task, or another issue type in your project.

issues, herhangi bir Jira projesinin yapı taşlarıdır. Bir issue, projenizdeki bir hikayeyi, hatayı, görevi veya başka bir sorun türünü temsil edebilir.

Article of the Week 10m

- How to Use Git/GitHub without asking for authentication always: Passwordless Usage of Private Git Repositories

Video of the Week 10m

- Terraform Explained

Retro Meeting on a personal and team level 10m

Ask the questions below:

- What went well?
- What could be improved?
- What will we commit to do better in the next week?

Coding Challenge 5m

- Coding Challenge: Reverse Input Number

Case study/Project 10m

- Project-202: Phonebook Application (Python Flask) deployed on AWS Application Load Balancer with Auto Scaling and Relational Database Service using Terraform

Closing 5m

-Next week's plan

-QA Session
