

STUDENT VERSION (DevOps-Week-1)



CLARUSWAY
WAY TO REINVENT YOURSELF

Meeting Agenda

- ▶ Icebreaking
- ▶ Microlearning
- ▶ Questions
- ▶ Interview/Certification Questions
- ▶ Coding Challenge
- ▶ 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

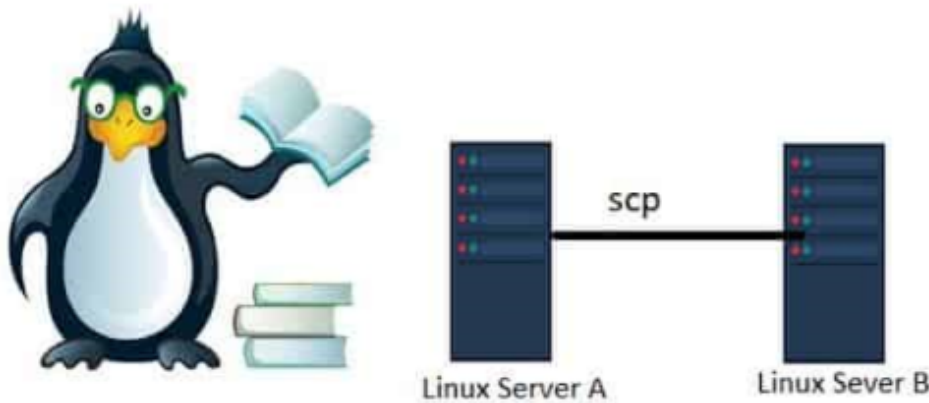
10m

- 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 to 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. Containers include the application and all of its dependencies, but share the kernel with other containers. They run as an isolated process in userspace on the host operating system. They're also not tied to any specific infrastructure – Docker containers run on any computer, on any infrastructure, and in any cloud.

- A. **True**
- B. False

4. What command should you run to see all running container in Docker?

- A. docker run
- B. **docker ps**
- C. docker --help
- D. docker build
- E. docker pull

5. Which of the following is not a state of Docker container?

- A. Running
- B. Freezed
- C. Paused
- D. Restarting
- E. Exited

Interview/Certification Questions

20m

1. What are the different phases in DevOps?
2. Explain the concept behind Infrastructure as Code (IaC).
3. How Terraform works?
4. You have an EC2 instance in the AWS us-east-1 region. The application in the instance needs to access a DynamoDB table that is located in the AWS us-east-2 region. The connection must be private without leaving the Amazon network and the instance should not use any public IP for communication. How would you configure this?
 - A. Configure an inter-region VPC endpoint for the DynamoDB service.
 - B. Configure inter-region VPC peering and create a VPC endpoint for DynamoDB in us-east-2.
 - C. Create an inter-region VPC peering connection between us-east-1 and us-east2.
 - D. There is no way to setup the private inter-region connections.
5. You own a MySQL RDS instance in AWS Region us-east-1. The instance has a Multi-AZ instance in another availability zone for high availability. As business grows, there are more and more clients coming from Europe (eu-west-2) and most of the database workload is read-only. What is the proper way to reduce the load on the source RDS instance?
 - A. Create a snapshot of the instance and launch a new instance in eu-west-2.
 - B. Promote the Multi-AZ instance to be a Read Replica and move the instance to eu-west-2 region.
 - C. Configure a read-only Multi-AZ instance in eu-west-2 as Read Replicas cannot span across regions.
 - D. Create a Read Replica in the AWS Region eu-west-2.

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-201: Create Apache Servers with Terraform

Closing

5m

-Next week's plan

-QA Session

created	A container that has been created (e.g. with <u>docker create</u>) but not started
restarting	A container that is in the process of being restarted
running	A currently running container
paused	A container whose processes have been paused
exited	A container that ran and completed ("stopped" in other contexts, although a created container is technically also "stopped")
dead	A container that the daemon tried and failed to stop (usually due to a busy device or resource used by the container)