Workshop Handout Lecture „Config Managament & Continuous Delivery"

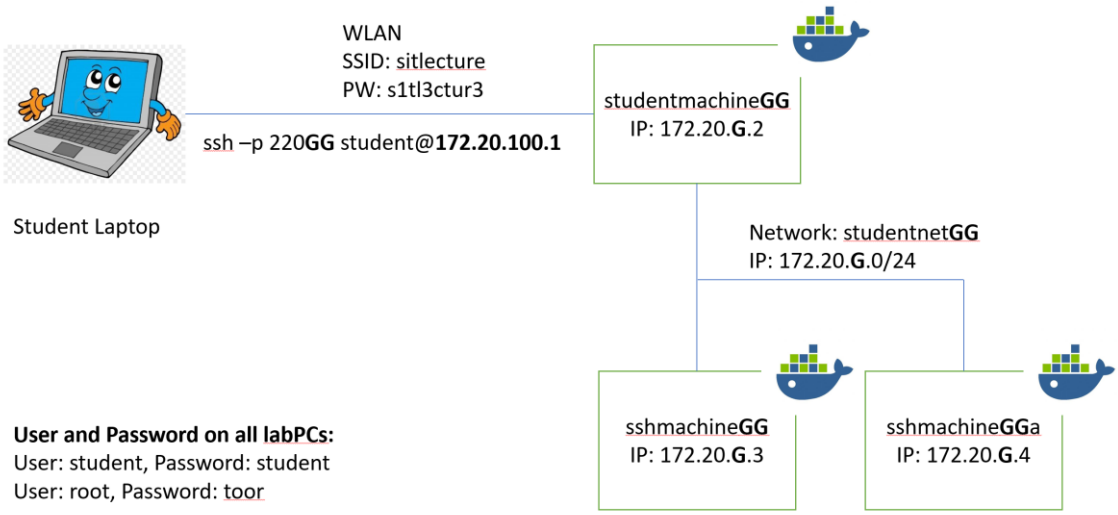
G__

Requirements:

What you need: Laptop with SSH Client, e.g. Putty or MobaXTerm
You can get them free here:
Putty: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
MobaXTerm: https://mobaxterm.mobatek.net/download-home-edition.html

Lecture Lab Network Layout

Part 1: Ansible



Task 1: Connect to your group studentmachine via ssh and try to ping your "sshmachine"

SSH command to access your studentmachine:
`ssh -p 220GG student@172.20.100.1`

Example: Group **01**:
Studentmachine1 IP: 172.20.**1**.2
SSH on Studentmachine is listening on port: 220**01**

You should see something similar to this:

```
*******************************
 Hello to your studentmachine  *
*******************************
student@8a52ea0aed32:~$
```

T1.1: Ping your corresponding sshmachines (ping sshmachine**GG**, sshmachine**GG**a).

Task 2: Ansible ping

Enter the directory "/home/student/ansible".
Here you can find a basic ansible directory structure. Look a bit around.

T2.1: Update the inventory file "`studentlab`", with your your "sshmachine**"** hosts in the inventory group **"studentlab"**

Try `ansible -m ping <hostname>` to validate that Ansible can reach and access each of your sshmachine.

T2.2: Try to use `ansible -m ping -I <inventory> <inventory_group>` to reach your whole group "studentlab" from your inventory.

Task 3: Deploy “message of the day”

T3.1: Ssh into your sshmachine**GG** and look at the prompt how you are greeted. You should see something like this:

```
******************************
 Hello to your sshmachine  *
******************************
student@5cbc954c1b05:~$
```

After you saw it, log out to be in your studentmachine again.

> Hint: you can use the linux cmd:
> `figlet <some text>`
> To create some ASCII Arts for your motd file ;-)

T3.2: Read what a Linux “Message of the Day (MOTD) is:
https://linuxconfig.org/how-to-set-a-custom-message-of-the-day-on-linux

T3.3: Now you want to create your own motd file and deploy it to your sshmachineGG. Create a folder for a role “`message-of-the-day`” with subfolders “`files`” and “`tasks`”. Create a file with name `motd` under `/roles/message-of-the-day/files/motd` with the following content:

```
****************************************
* sshmachine: group <YOUR GROUP Number> *
****************************************
```

T3.4: Now create a file `/roles/message-of-the-day/tasks/main.yml` , where you use the ansible module “copy” to deploy your motd file to your sshmachines to the folder /etc/motd.

Ansible copy module: https://docs.ansible.com/ansible/latest/modules/copy_module.html

T3.5: Write a simple Ansible playbook “`lecture-playbook.yaml`”, which executes your role “`message-of-the-day`” to your ssh machine. Use the already prepared example file under ansible/playbook as a help.

Run your ansible playbook to deploy your motd file to your sshmachines:
`ansible -playbooks -l studentlab playbooks/lecture-playbook`

Watch the output, it should look like this:

```
student@d845828634cc:~/ansible$ ansible-playbook -l studentlab playbooks/lecture-playbook.yaml

PLAY [studentlab] ***************************************************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************************************
Friday 10 January 2020  14:51:38 +0000 (0:00:00.024)       0:00:00.024 ********
ok: [sshmachine01a]
ok: [sshmachine01]

TASK [message-of-the-day : Deploy own motd file] ********************************************************************************************************
Friday 10 January 2020  14:51:39 +0000 (0:00:01.132)       0:00:01.157 ********
changed: [sshmachine01]
changed: [sshmachine01a]

PLAY RECAP **********************************************************************************************************************************************
sshmachine01               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
sshmachine01a              : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

Friday 10 January 2020  14:51:40 +0000 (0:00:00.868)       0:00:02.025 ********
===============================================================================
Gathering Facts ---------------------------------------------------------------------------------------------------------------------------------- 1.13s
message-of-the-day : Deploy own motd file -------------------------------------------------------------------------------------------------------- 0.87s
Playbook run took 0 days, 0 hours, 0 minutes, 2 seconds
```

SSH into your sshmachines and check if your deployment worked.

T3.3. Run your playbook again and look at the ansible output. Notice, that it should not change the /etc/motd file, since it is still correct.
Congrats, you reached a reproducible target state!

Task 4: Using Ansible Variables
In Task 3, we deployed a static configuration file of our message of the day.
We want to make this role more general, using variables.

T4.1: Look at the folder `group_vars`
Edit the file `all` and add a variable with the name: "`group`" and assign it the value: "`group`**`GG`**" where GG is your group number.
Save and exit the file.

Now modify your motd role in a way to use the Ansible module "`template`" and the Variable "`group`" so the text in your motd file shows the group id now based on this variable.

> Hint: Variable in Ansible tasks are accessed by:
> "`{{ variable_name }}`"

Create folder `roles/message-of-the-day/templates`.
Copy your `roles/message-of-the-day/files/motd` file to `roles/message-of-the-day/templates/motd.j2`
Edit it and put in your Ansible variable into it.

T4.3: Edit your `roles/message-of-the-day/tasks/main.yml` and replace the Ansible `copy` module with the Ansible `template` module.

Test your rollout by using your `lecture-playbook` and check the results as described in Task 3.5.

Helpful Documentation:
https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html

Your output should look like this:

Ansible playbook run:

```
student@8639857cc793:~/ansible$ ansible-playbook -l studentlab playbooks/lecture-playbook.yaml

PLAY [all] *********************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************
Saturday 11 January 2020  18:14:38 +0000 (0:00:00.023)       0:00:00.023 ******
ok: [sshmachine01a]
ok: [sshmachine01]

TASK [message-of-the-day : Deploy own motd file using variable group = "01"] ***************************************************************************
Saturday 11 January 2020  18:14:40 +0000 (0:00:02.242)       0:00:02.266 ******
changed: [sshmachine01a]
changed: [sshmachine01]

PLAY RECAP *********************************************************************************************************************************************
sshmachine01               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
sshmachine01a              : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

Saturday 11 January 2020  18:14:41 +0000 (0:00:00.906)       0:00:03.173 ******
===============================================================================
Gathering Facts ----------------------------------------------------------------------------------------------------------------------------- 2.24s
message-of-the-day : Deploy own motd file using variable group = "01" ----------------------------------------------------------------------- 0.91s
Playbook run took 0 days, 0 hours, 0 minutes, 3 seconds
```

SSH check:

```
student@8639857cc793:~/ansible$ ssh sshmachine01
Last login: Sat Jan 11 18:02:30 2020 from 172.20.1.2
-bash: warning: setlocale: LC_ALL: cannot change locale (en_US.utf8)
 _                 _
 _   _| |__   _   _ _ __ | |_ _   _
| | | | '_ \| | | | '_ \| __| | | |
| |_| | |_) | |_| | | | | |_| |_| |
 \__,_|_.__/ \__,_|_| |_|\__|\__,_|

We are in group: "01" :-)
```

Task 5: Checkout, build and run applications

Next, we want to get familiar with the "Continuous Delivery" part of the workshop. Therefore, we first want to get some example source code and manually build and run the applications

```
# checkout git repository
git clone $REPO_URL
# build it
source /opt/ros/melodic/setup.bash
catkin_make talker
catkin_make listener
# run it
source devel/setup.bash
rosrun talker talker
```

Task 5.1: Automate the build

Now we want start automating above steps using Docker. There are two Dockerfiles that we need to adapt for that (add build commands)

1. Add build command for talker application in src/talker/Dockerfile
2. Add build command for talker application in src/listener/Dockerfile
3. cd src/talker && docker build –t talker_G:latest .
4. cd src/talker && docker build –t listener_G:lates .

Task 6: Automate deployment

To prepare deployment of your applications, we want to implement a docker-compose.yaml file, that describes how our docker images from Task 5 are run.

1. Rename the services described in docker-compose.yaml to make them unique
2. Use your build docker images (e.g. talker_G:latest) in docker-compose.yaml
3. Check if applications start properly
    a. docker-compose up

Task 7: Implement CI

1. Create a feature branch for your team, add, commit and push changes made in the previous task
    a. git checkout –b feature/team-G
    b. git add *
    c. git commit –m "my commit message"
    d. git push feature/team-G origin/feature/team-G
2. Add docker build commands from Task 5 to .gitlab-ci.yaml file
3. Commit and push change of .gitlab-ci.yaml file

Task 8: Implement CD

1. Add docker-compose command from Task 6 to .gitlab-ci.yaml file
2. Commit and push change of .gitlab-ci.yaml file