☰   🔍 (https://profile.intra.42.fr/searches)   **ishaimou**

(https://profile.intra.42.fr)

Remember that the quality of the defenses, hence the quality of the of the school on the labor market depends on you. The remote defences during the Covid crisis allows more flexibility so you can progress into your curriculum, but also brings more risks of cheat, injustice, laziness, that will harm everyone's skills development. We do count on your maturity and wisdom during these remote defenses for the benefits of the entire community.

SCALE FOR PROJECT INCEPTION-OF-THINGS (/PROJECTS/INCEPTION-OF-THINGS)

You should evaluate 2 students in this team

★

Git repository

`git@vogsphere-v2.1337.ma:vogsphere/intra-uuid-12f246ba-62cd`   📋

Introduction

Please comply with the following rules:

- Remain polite, courteous, respectful and constructive throughout the evaluation process. The well-being of the community depends on it.

- Identify with the student or group whose work is evaluated the possible dysfunctions in their project. Take the time to discuss and debate the problems that may have been identified.

- You must consider that there might be some differences in how your peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade them as honestly as possible. The pedagogy is useful only and only if the peer-evaluation is done seriously.

Guidelines

- Only grade the work that was turned in the Git repository of the evaluated student or group.

- Double-check that the Git repository belongs to the student(s). Ensure that the project is the one expected. Also, check that 'git clone' is used in an empty folder.

- Check carefully that no malicious aliases was used to fool you and make you evaluate something that is not the content of the official repository.

- To avoid any surprises and if applicable, review together any scripts used to facilitate the grading (scripts for testing or automation).

- If you have not completed the assignment you are going to evaluate, you have to read the entire subject prior to starting the evaluation process.

- Use the available flags to report an empty repository, a non-functioning program, a Norm error, cheating, and so forth.
In these cases, the evaluation process ends and the final grade is 0, or -42 in case of cheating. However, except for cheating, student are strongly encouraged to review together the work that was turned in, in order to identify any mistakes that shouldn't be repeated in the future.

Attachments

subject.pdf (https://cdn.intra.42.fr/pdf/pdf/31724/en.subject.pdf)

Preliminaries

If cheating is suspected, the evaluation stops here. Use the "Cheat" flag to report it. Take this decision calmly, wisely, and please, use this button with caution.

Preliminary tests

- Defense can only happen if the evaluated group is present. This way everybody learns by sharing knowledge with each other.
- If no work has been submitted (or wrong files, or wrong directory, or wrong filenames), the grade is 0, and the evaluation process ends.
- For this project, you have to clone the Git repository on the group's machine.
- For this project, you must use the virtual machine of 42.

⊘ Yes   ✕ No

General instructions

General instructions

- During the defense, whenever you need help in order to verify a requirement of the subject, the evaluated group must help you.
- Ensure that all the files required for the three different parts of

the project are in the folders p1, p2 and p3 respectively. There may
be an additional bonus folder.

◌ Yes   ✕ No

Mandatory part

The project consists of setting up several infrastructures with different services that use K3s, Vagrant and K3d. Make sure that all of the following requirements are met.

Global configuration and explanation

- Those being evaluated should explain to you in a simple way:
- The basic operation of K3s.
- The basic operation of Vagrant.
- The basic operation of K3d.
- What is a continuous integration and Argo CD.

◌ Yes   ✕ No

Part 1 - Configuration

- Check that a Vagrantfile is present in the p1 folder. Once done, check
its content. Thanks to the help of the evaluated persons, you should
basically understand this file. It must be similar to the example
given in the subject.
- Check that there are two virtual machines in the Vagrantfile.
- In the Vagrantfile, check that the latest stable version of CentOS is
used for both virtual machines.
- Check that there is an eth1 interface with the IP addresses required
by the subject.
- The names chosen for the two virtual machines must include a login of
a member of the group. For the first machine, it must be followed by
S (like Server), and for the second one, by SW (like ServerWorker).
If something does not work as expected, the evaluation stops here.

◌ Yes   ✕ No

Part 1 - Usage

- Use Vagrant to SSH into both virtual machine with the help of the
evaluated group.
- Ensure there is an eth1 interface with the IP addresses required by
the subject by using the command:
"ifconfig eth1"
- Ensure both machines have the hostname required by the subject.

- Then, check that both virtual machines use K3s. The evaluated group
should be able to help you.
- Finally, verify that the Server machine and the Agent machine are in
the same cluster by running this command on the Server machine:
"kubectl get nodes -o wide"
An output similar to the one given as an example in the subject is
expected. The evaluated group must explain to you the output.
If something does not work as expected, the evaluation stops here.

⟋ Yes   ✕ No

Part 2 - Configuration

- To avoid space/performance issues, you can of course shut down every
other running virtual machines with the help of the evaluated group.
- Check that a Vagrantfile is present in the p2 folder. Once done, check
its content. Thanks to the help of the evaluated persons, you should
basically understand this file. It must be similar to the example
given in the part 1 of the subject.
- Check that there is only one virtual machine in the Vagrantfile.
- In the Vagrantfile, check that the latest stable version of CentOS is
used for the virtual machine.
- Check that there is an eth1 interface with the IP address required by
the subject.
- The name chosen for the virtual machine must include a login of a
member of the group followed by the capital letter S.
- If extra files are present in the p2 folder, verify them and ask for
explanations.
If something does not work as expected, the evaluation stops here.

⟋ Yes   ✕ No

Part 2 - Usage

- Use Vagrant to SSH into the virtual machine with the help of the
evaluated group.
- Ensure there is an eth1 interface with the IP address required by the
subject by using the command:
"ifconfig eth1"
- Ensure the machine has the hostname required by the subject.
- Then, check that the virtual machine uses K3s. The evaluated group
should be able to help you.
- Verify that the virtual machine meets the subject's requirements. To
do so, use the following commands:
"kubectl get nodes -o wide"
It should display the name of the controller and the internal IP address.
"kubetctl get all -n kube-system"
It should display 3 applications. The second one must have 3 replicas.

The evaluated group must explain to you each output.
Next, they must show you how their Ingress works. The command is
deliberately not given here.
- Now, check that the 3 applications can be accessed depending on the
HOST header that is used (have a look at the subject). To do so, you
can use curl with the help of evaluated group, or just a browser (for
a web application). You will have to change the HOST in order to see
some differences.

If something does not work as expected, the evaluation stops here.

⊘ Yes   ✕ No

Part 3 - Configuration

- Thanks to the evaluted group, start up the infrastructure.
- Check that the configuration files are present in the p3 folder. Once
done, check their content. Don't hesitate to ask for more precise
explanations. This part is essential to understand what's next.
- Make sure there is at least 2 namespaces in K3d: "argocd" and "dev".
Use the command:
"kubectl get ns".
- Verify that there is at least 1 pod in the "dev" namespace. Use the
command:
"kubectl get pods -n dev"
- Ensure the group members understand the differences between a namespace
and a pod.
- Check that all the required services are running with the help of the
evaluated group.
- Check that Argo CD is installed and configured. You can access it in
your web browser. You will need a login and a password. The evaluated
group will give them to you.
- Check that the login of someone of the group was put in the name of
the Github repository (e.g., if a login was "wil", the name could be
"wil_config" or "wil-ception"). Read the subject carefully to understand
this part.
- Check that a Docker image is used in the Github repository. The image
can be Wil's or a custom one. In the second case, verify that the login
of a member of the group was put in the name of the Dockerhub
repository. Also, ensure that there are the two required tags in the
Dockerhub repository.
- If there are extra files in the p3 folder, ask for explanations.
If something does not work as expected, the evaluation stops here.

⊘ Yes   ✕ No

Partie 3 - Usage

- Now that you can use Argo CD, try to understand how it basically
works. With the help of the evaluated group, navigate through the
application. Do not hesitate to ask questions here. If you have any
doubt (maybe their explanations are confused or they can't explain
something they should know), the evaluation stops now. It is important.
- Check that the v1 application can be accessed from this machine. You
can use curl (there is an example usage in the subject).
- Verify that Dockerhub is used. This part is important. In case of any
doubt, the evaluation stops now.
- Since you can see the v1 application, you must be able to update it
with the help of the evaluated group. Use the configuration file on
Github that Argo CD relays on. You must commit and push a modification.
It will automatically trigger the update of your application. You must
be able to understand how this whole process works. Do not hesitate
to ask for explanations.
- Now that you have pushed the v2 application on Github, if
synchronizing didn't happen, do it manually in Argo CD (if it did
happen, skip this step). The evaluated people must help you.
- Ensure that the application was successfully synchronized using
operation given as an example in the subject. The evaluated people
must help you.

If something does not work as expected, the evaluation stops now.

⊘ Yes   ✕ No

Bonus

Evaluate the bonus part if, and only if, the mandatory part has been entirely and perfectly done, and the error management handles unexpected or wrong usage. In case all the mandatory points were not passed during the defense, bonus points must be totally ignored.

Bonus

- Check if there are configuration files in the bonus folder. Ask for
explanations about each of them.
- Test Gitlab functions correctly and was properly implemented. To do
so, create a new repository with the help of the evaluated group.
Then, try to add some code in it. Check the operation was successful
on Gitlab.
- The last step is quite simple. Make sure that the operations of the
part 3 of the subject still function correctly. Ensure that the
repository used in Argo CD is a local repository on Gitlab. The
evaluated group should guide you in this process so you can verify the
operations works as expected with the two versions of the chosen

application.
- If the synchronization and the version change of the application
are completed with no errors, validate this part.

☑ Yes   ✕ No

Ratings

Don't forget to check the flag corresponding to the defense

✔ Ok   ★ Outstanding project

▮ Empty work   ▮ Incomplete work   ▤ Cheat   ♟ Crash   ♟ Incomplete group

⚠ Concerning situation   ⊘ Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation