**Eval kubernetes:**

**General instructionsGeneral instructions**  
- Check if all the necessary configuration files of your application are in the folder srcs.  
- Check if the setup.sh file is at the root of the repository.  
- Execute “setup.sh”

**Mandatory part**  
*This project consist of clusturing an application and deploying it with Kubernetes. Check if all the next points are respected. At the first error, you stop the correction and put a zero.*

**Services environment**  
- Verify if the application is deployed with all containers only with “setup.sh” script.  
- Check using the dashboard if the evaluated has all the different containers.   
They are nginx, ftps, wordpress, mysql, grafana and influxDB.   
These containers must have the same name. If this is not the case or if a container is missing, the correction stops.  
- Check if all containers are build with Alpine Linux. If not, the correction stops here.  
- Also, check that each container has its Dockerfile, which was built using setup.sh.  
The evaluated have to build himself the images that he will use. It is forbidden to take already build images  
or use services like DockerHub.

If one is missing, or if one does not start with “FROM: alpine” or any other local image, the correction stops here.

**Expose it !**  
- Check that redirections to your services are done with a Load Balancer.  
To do that use “kubectl get services”:  
Grafana, ftps, Wordpress, phpmyadmin & nginx must be “LoadBalancer” type and EXTERNAL-IP field cannot be 127.0.0.1 or end with .0 or .255.  
Influxdb & mysql must be “ClusterIP” type.  
Other entry can appear, but none must be “NodePort” type.  
Also, check that “setup.sh” file don’t use any “kubectl port-forward” command.

**Nginx**  
- Try to access http (port 80) and verify that you are automatically redirected to https (port 443).  
Then, execute the command “curl -I (i) [http://IP](http://ip/)” and check if return code is a  
“301 Moved Permanently” and the “Location” line is the same IP but in https.  
Page displayed does not matter, as long it’s not a web error (404, 503 etc).  
SSL certificate are not necessarily recognized, certificate error on https is normal.  
Check that nginx redirects on a /wordpress route with a “307 redirect” and on a /phpmyadmin route with a “reverse proxy”.  
If one point listed previously differ, the correction stops.

**FTP(s) server**  
- Check if the FTPS server is listening on port 21. Make sure it is a FTPS server (s for secure)   
and not a basic FTP server.  
If the FTPS server does not work as it should, the correction stops here.  
- Check if you can upload and download files without any errors.

kubectl exec -it pod/ftps-647ddf58f8-ggdlq – sh

ls -la

cd var

cd ftp

**Hello Word(Press), MySQL and PhpMyAdmin**  
- Check if the WordPress website works under port 5050. Connect to it using the administrator account,  
check if several users is present, after leave a comment under post. Make sure your comment is added to the database entries.  
For that, you can access PhpMyAdmin interface, must be under port 5000 and check the database (“wp\_comments” table).  
Wordpress and PhpMyAdmin needs its own nginx server.  
The database must persist in case failure.   
To test this, you can remove the MySQL container with the Kubernetes dashboard.  
It must recreate itself automatically, after check if your comment is still in the database.  
If something does not work as expected, the correction stops here.

**Grafana and InfluxDB**  
- Check if grafana is running under port 3000. Connect to the interface.  
You have to check if grafana is monitoring all containers. To do that search for the dashboard list,  
click on the dashboards one by one.  
Like MySQL database, after deleting the InfluxDB container, check that the data prior to deletion is still present in Grafana.  
If something does not work as expected, the correction stops here.

* Klopt het dat grafana zelf geen data heeft?

**Persistence!**  
- Check that in the event of a crash/shutdown of one of the services, the associated container relaunches correctly.  
To do this, use “kubectl exec deploy/SERVICE -- pkill APP” (replacing SERVICE and APP of course).  
Stop web containers (usually nginx or php-fpm APP) and stop grafana APP for this same container.  
Stop the ftps server (usually vsftpd APP).  
Stop mysql and influxd APP for corresponding databases (recheck if the data persists).  
Stop the ssh server on the Nginx container (sshd APP).  
If a container hasn’t relaunched after several minutes, if the MySQL or InfluxDB data is lost  
or if a service has restarted badly (in particular ftps), the correction stops here.

kubectl exec deploy/nginx -- pkill nginx  
kubectl exec deploy/nginx -- pkill sshd  
kubectl exec deploy/ftps -- pkill vsftpd  
kubectl exec deploy/grafana -- pkill grafana  
kubectl exec deploy/mysql -- pkill mysqld  
kubectl exec deploy/influxdb -- pkill influxd  
kubectl exec deploy/phpmyadmin -- pkill php-fpm  
kubectl exec deploy/telegraf -- pkill telegraf  
kubectl exec deploy/wordpress -- pkill nginx