# A. Instructions of deployment the solution infrastructure.

For the solution purposes I’ve built a HA (High Availability) cluster, using docker-swarm orchestrator. Cluster is consisting of three nodes, one Leader and two workers. The host system is Intel Core i9-12950HX, 32 Gb RAM, Windows 11 Pro. The virtualization is done using Oracle Virtualbox and orchestrator for creating the underlying infrastructure is Hashicorp Vagrant. Vagrant file is presented in the repository should you decide to recreate a cluster from scratch. Take a note, that I gave 4 virtual cores for every node and 8GB of RAM, and should you need to give less, make a corresponding changes at the bottom of the Vagrantfile (vb.cpus = "4" and vb.memory = "8192").

Now, assuming, that you have cluster up and running already, which have at least 3 nodes, and these nodes are named node1, node2 and node3 (docker-swarm file utilizes nodes’ names) here are the instructions of deployment.

1. Log in to your leader node (in my case it’s node1) and clone the following git repository:

https://github.com/Dealmi/ci\_cd\_in\_cluster

Type “git clone <https://github.com/Dealmi/ci_cd_in_cluster.git>” and you should get almost all files you need. For Jenkins backup kindly download this file https://drive.google.com/file/d/1efeFCH1T5c\_-uB-xAf8fB318q9tDO\_r8/view?usp=sharing

For that you can use wget.

2. Next step is to make start.sh executable, for that type ”chmod +x start.sh”. Then run it in order to start our infrastructure. Please, wait a minute or so to insure, that all services are up and running.

3. Now we need to restore a Jenkins backup, which is needed to do our CI\CD part.

Jenkins container is running on node3, so copy file jenkins\_backup.tgz to that node using your preferred method of transfering files. Execute “docker container ls” to find the name of Jenkins’ container then copy an archive to the root directory of it by using ”docker cp jenkins\_backup.tgz

skanestas\_jenkins.1.xxxxxxxxx:/” where xxxx is a serial number of the container, usually it’s fine to just press the Tab button and system should fill the rest. Afterwards you need to unpack the backup, so get inside the skanesta\_jenkins container by doing ”docker exec -it skanestas\_jenkins.1.xxxx bash” . Once you are in type “tar -xzf jenkins\_backup.tgz” and it should unpack in /var/tmp/FULL<timestamp> directory. You will need the initial password for starting using Jenkins, so while you are still inside, type “cat /var/jenkins\_home/secrets/initialAdminPassword” and take a note of the password.

4. After you have done unpacking the backup, open in browser of your choice, outside of the cluster, and type the following address: [http://192.168.100.10:8080](http://192.168.100.10:8080/) (ip is the address of the node1, defined in Vagrantfile). You will be presented with “Getting started page” where you will type the password obtained in a previos step into the “Administrator password” form.

5. On the next screen choose “Select plugins to install” then click None at the top of the screen, then “Install”. No worries, you will install all you need with the backup restoration. Afterwards fill in the form, click “Save and continue” then on next screen click “save and finish”, and finally click “Start using Jenkins” on follow-up screen and proceed to the main Jenkins’ skreen.

6. You need to install backup plugin. For that go to “Manage Jenkins” > “Manage Plugins” > “Available “ and in the search field type “thinbackup”. In the result tick the square off and click “Install without restart”. On the next screen scroll to the bottom and tick off the “Restart Jenkins ..” option. After Jenkins finished installing the plugin, It will restart.

7. Login to it and proceed to backup screen. Go “ Manage Jenkins” > “ThinBackup” > “Settings”.

In the backup directory type “/var/tmp”, scroll to the bottom and click “save”. Then click “Restore” button and choose the only backup from the drop-down menu. Tick off “Restore plugins” option and click “Restore”. To the left in “Build queue” section you will see “Jenkins is going to shut down. No further builds will be performed” It means that Jenkins restoring the backup. Go back to “Manage Jenkins” section, click “System Log” > “All Jenkins logs” and scroll the window all the way down. You should see “Restore finished” message. If it’s not there, just wait a couple of minutes and refresh the page until you do.

8. Now we need to restart Jenkins. Best way is to kill the container, since all data is persistant per configuration. Get back to node3, and there type “docker container kill skanestas.jenkinsxxxxxx” (use tab button to autofill) and execute it. Wait a few second and a new container should go online.

9. Open [http://192.168.100.10:8080](http://192.168.100.10:8080/) in a browser again and log in. You should see “upload to database” job in the main screen. It will do all the CI\CD process of continuously delivering data to database as soon as new portion of data is introduced.

10. Finally, you will need to restore grafana dashboard to see the statistics.

Open <http://192.168.100.10:3000/>and login into it, using default admin\admin credential.

In the home screen choose “Add your first data source”. Scroll to “PostgreSQL” section and click it. In the form, fill in the following fields:

Host: skanestas\_db

Database: skanestas

User: postgres, Password: post123gre

TSS:disable

Version: 12+

Min time interval : 1m

Then press “Save & test”. It should succeed. You should see green flag and message “Database connection OK”.

Now let’s restore the dashboard. Click square icon (Dashboards) and choose “+ import”. Click “Upload JSON file”. Navigate to “skanestas trade-grafana.json file (it should be in the cloned repository) and open it. You will see “Options” window with Name: skanestas trade and Folder: General. Choose “PostgreSQL” from the “PostgrSQL” drop-down menu and click “Import”. You will be greeted with the dashboard featuring graphs of ask\_01,bid\_01 then their sum and the cases, where ask\_01+bid\_01<105.

# B. Explanations:

1. I’ve modified the task code in order for it to output logs into json file.

2. The CI\CD process works as following: container wiht the logs generator generates file. Jemkins finds it, and sends signal to the service container through web request. Service reads the file and sends it’s content into PostgreSQL database. Thus the process of continuous integration and continous delivery is established.

3. Grafana collects statistics about ask\_01 and bid\_01 fields from database and represents it to us in a form of graphs.

4. From the repository we really only need jenkins\_backup.tgs, skanestas trade-grafana.json, skanestas.yml and start.sh for our infrastructure. The rest of files is there to demonstrate my work.

---

**I hope you had as much fun looking at results of my job as me doing it! Have a beautiful day!**