

**DevOps -**  
Containerization,  
CI/CD & Monitoring



# DevOps

Containerization, CI/CD & Monitoring

*February 2022*

**Regular Exam**  
**Exam**

*Stefan Veselinov*

---

## Solution Steps

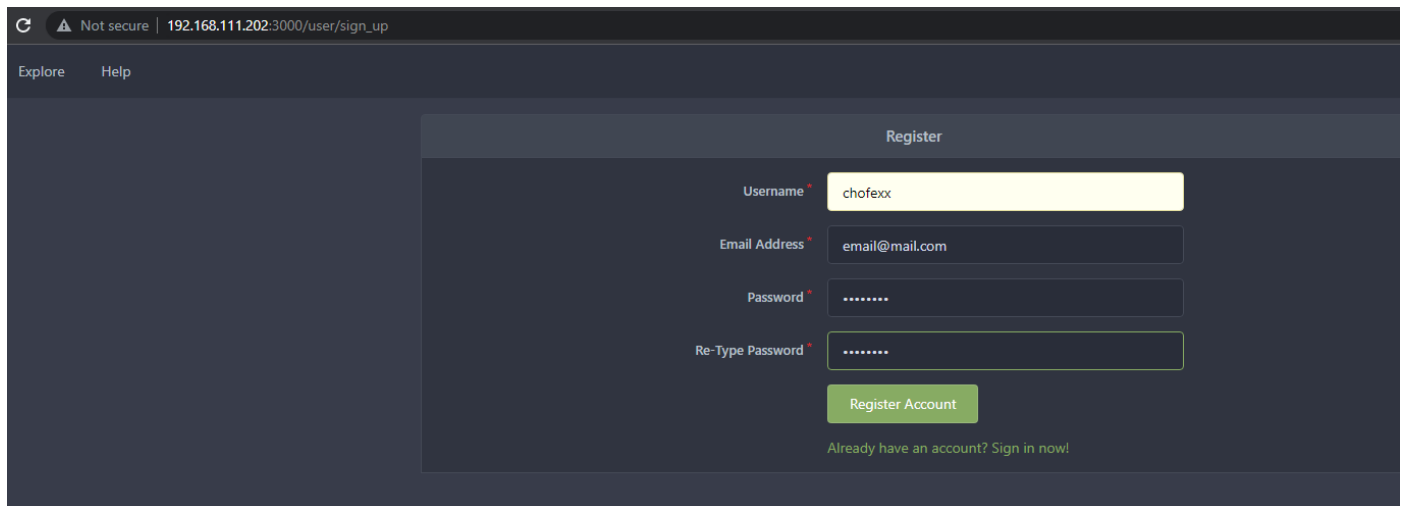
➤ Host Machine - Windows 10 Pro 21H2 19044.1526

- Run **vagrant up** in project folder

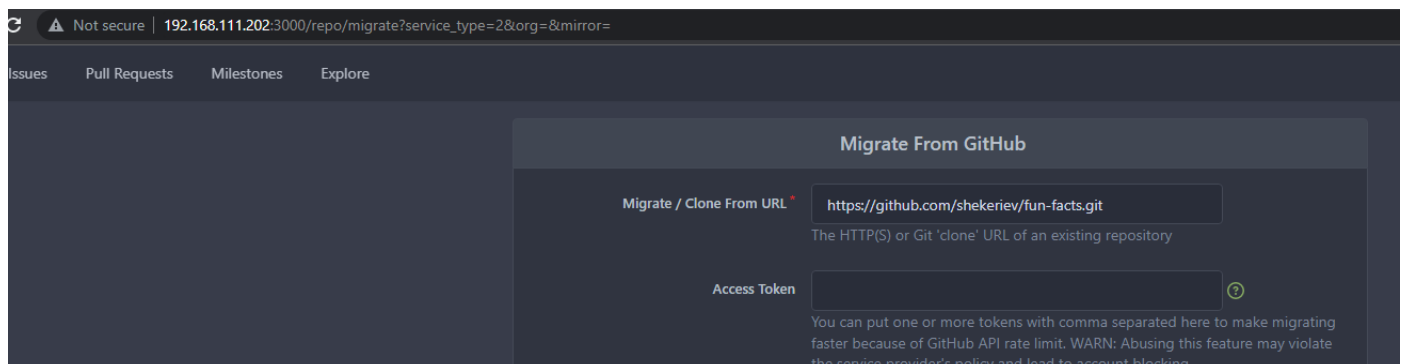
```
PS D:\HUB\SoftUni\DevOps_Containerization_CICD_Monitoring_02_2022\github\DevOps-SoftUni-Feb2022-ExamPrep\exam> vagrant up
Bringing machine 'pipelines' up with 'virtualbox' provider...
Bringing machine 'containers' up with 'virtualbox' provider...
Bringing machine 'monitoring' up with 'virtualbox' provider...
==> pipelines: Importing base box 'veselinovstf/debian-minimal'...
```

NOTE: Beware of network errors while vagrant is bringing up the vm's!

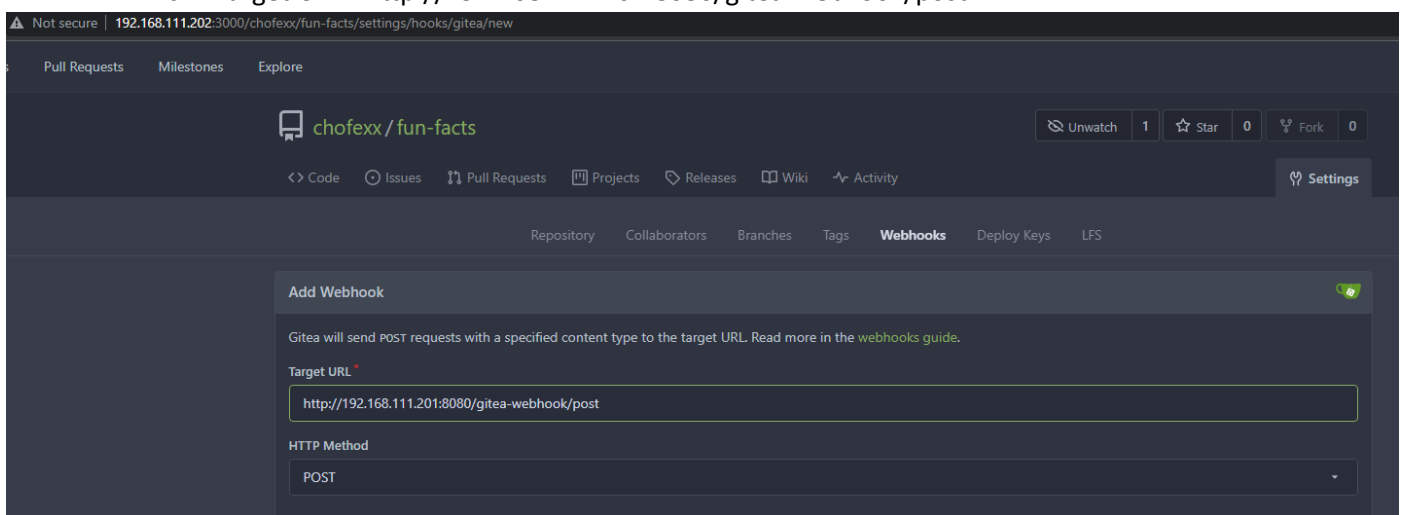
- Create Gitea User: chofexx with password: password - [http://192.168.111.202:3000/user/sign\\_up](http://192.168.111.202:3000/user/sign_up)



- Create New Migration From GitHub repository - <https://github.com/shekeriev/fun-facts.git>



- Create Webhook for repository –
  - <http://192.168.111.202:3000/chofexx/fun-facts/settings/hooks>
  - Target URL: <http://192.168.111.201:8080/gitea-webhook/post>



- Connect to containers vm - **vagrant ssh containers** – NOTE: don't forget to commit **.env** file

```
git clone http://192.168.111.202:3000/chofexx/fun-facts.git
cd fun-facts/
cp /vagrant/src/app/docker-compose.Production.yml docker-compose.Production.yml
cp /vagrant/src/app/docker-compose.Development.yml docker-compose.Development.yml
cp /vagrant/src/app/Jenkinsfile Jenkinsfile
git commit -m "Adding CI/CD"
git push
    Username for 'http://192.168.150.102:3000': chofexx
    Password for 'http://chofexx@192.168.150.102:3000': password
cp /vagrant/src/app/.env .env
git add .env
git commit -m "Adding .env"
git push
    Username for 'http://192.168.150.102:3000': chofexx
    Password for 'http://chofexx@192.168.150.102:3000': password
```

```
PS D:\HUB\SoftUni\DevOps_Containerization_CICD_Monitoring_02_2022\github\DevOps-SoftUni-Feb2022-ExamPrep\exam> vagrant ssh containers
Linux containers 5.10.0-11-amd64 #1 SMP Debian 5.10.92-1 (2022-01-18) x86_64
```

The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/\*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.

Last login: Thu Feb 10 13:13:01 2022 from 10.0.2.2

```
vagrant@containers:~$ git clone http://192.168.111.202:3000/chofexx/fun-facts.git
```

```
cd fun-facts/
```

```
cp /vagrant/src/app/docker-compose.Production.yml docker-compose.Production.yml
```

```
cp /vagrant/src/app/docker-compose.Development.yml docker-compose.Development.yml
```

```
cp /vagrant/src/app/Jenkinsfile Jenkinsfile
```

```
git add *
```

```
git commit -m "Adding CI/CD"
```

```
git push
```

```
Cloning into 'fun-facts'...
```

```
remote: Enumerating objects: 26, done.
```

```
remote: Counting objects: 100% (26/26), done.
```

```
remote: Compressing objects: 100% (22/22), done.
```

```
remote: Total 26 (delta 0), reused 26 (delta 0), pack-reused 0
```

```
Receiving objects: 100% (26/26), 108.25 KiB | 12.03 MiB/s, done.
```

```
[main 1526d98] Adding CI/CD
```

```
Committer: vagrant <vagrant@containers.dol.exam>
```

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the  
following command and follow the instructions in your editor to edit  
your configuration file:

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
3 files changed, 158 insertions(+)
```

```
create mode 100755 Jenkinsfile
```

```
create mode 100755 docker-compose.Development.yml
```

```
create mode 100755 docker-compose.Production.yml
```

```
Username for 'http://192.168.111.202:3000': chofexx
```

```
Password for 'http://chofexx@192.168.111.202:3000':
```

```
Enumerating objects: 6, done.
```

```
Counting objects: 100% (6/6), done.
```

```
Compressing objects: 100% (5/5), done.
```

```
Writing objects: 100% (5/5), 1.45 KiB | 494.00 KiB/s, done.
```

```
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
```

```
remote: . Processing 1 references
```

```
remote: Processed 1 references in total
```

```
To http://192.168.111.202:3000/chofexx/fun-facts.git
```

```
4f58c69..1526d98 main -> main
```

- Go to <http://192.168.111.201:8080/> and login – admin/admin
  - – Build Now –initial build – every next build is going to be trier by repository change

The screenshot shows the Jenkins web interface for a pipeline named 'ExamPipeline-Pipeline'. The left sidebar contains navigation links: Back to Dashboard, Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Rename, Pipeline Syntax, GitHub Hook Log, and Polling Log. The main area displays the 'Pipeline ExamPipeline-Pipeline' with a 'Recent Changes' section showing a change on Apr 03 at 10:25. Below this is the 'Stage View' showing three stages: 'Declarative: Checkout SCM' (20s), 'Prepare - Clean Up Any Leftover Docker Containers' (2s), and 'Deploy - Run Docker Compose' (39s). The 'Build History' section at the bottom shows a single build #1 from Apr 3, 2022, 3:25 AM.

**Pipeline ExamPipeline-Pipeline**

**Recent Changes**

**Stage View**

Stage	Average stage times:
Declarative: Checkout SCM	20s
Prepare - Clean Up Any Leftover Docker Containers	2s
Deploy - Run Docker Compose	39s

**Permalinks**

- make change in project repo - push
- check if Jenkins is building the job
- Check Prometheus targets - <http://192.168.111.203:9090/targets>

←

→

Not secure

192.168.111.203:9090/targets

Prometheus Alerts Graph Status ▾ Help Classic UI

## Targets

All Unhealthy Collapse All

Q

Filter by endpoint or labels

containers (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.111.202:9100/metrics	UP	instance="192.168.111.202:9100" job="containers"	11.155s ago	27.274ms	

docker (1/1 up) [show less](#)

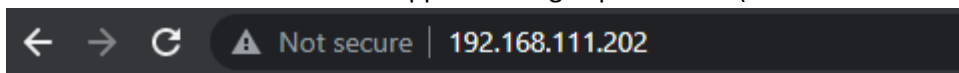
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.111.202:7070/metrics	UP	instance="192.168.111.202:7070" job="docker"	11.331s ago	15.214ms	

monitoring (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.111.203:9090/metrics	UP	instance="192.168.111.203:9090" job="monitoring"	14.552s ago	4.375ms	

pipelines (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.111.201:9100/metrics	UP	instance="192.168.111.201:9100" job="pipelines"	6.407s ago	25.871ms	



Recently discovered fun facts. Refresh in a minute or two to see more.

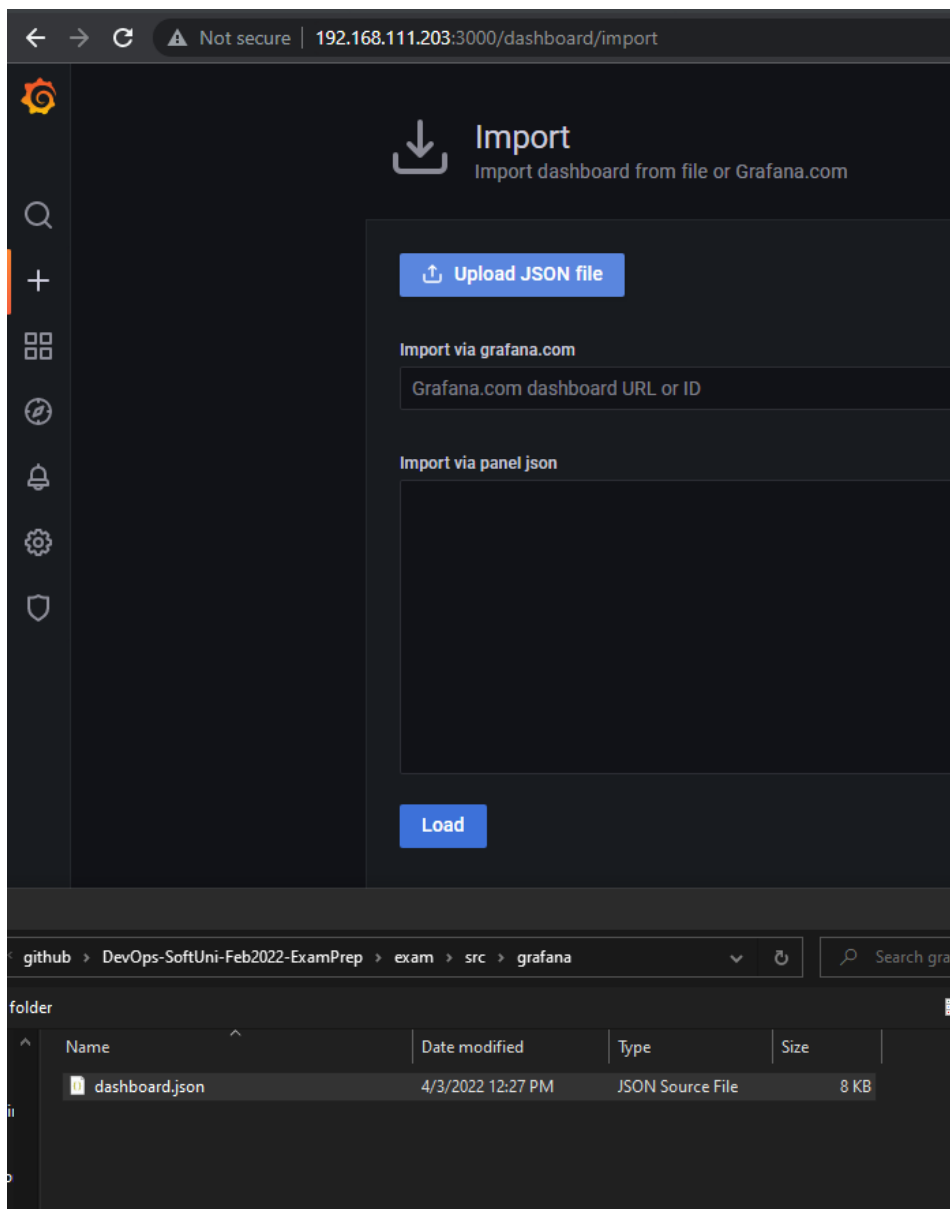
Build: 2022.04.03.10:15:05

- Check Jenkins

## Stage View

	Declarative: Checkout SCM	Prepare - Clean Up Any Leftover Docker Containers	Deploy - Run Docker Compose	Testing Development	Stopping Application - Development	Login to Docker Hub	Publishing - Client - to Docker Hub	Publishing - Generator - to Docker Hub	Publishing - Storage - to Docker Hub	Deploy - Run Docker Production
Average stage times: (Average full run time: ~1min 53s)	1s	1s	5s	1min 9s	7s	1s	2s	2s	2s	2s
#5 Apr 03 11:53 1 commit	420ms	1s	5s	1min 9s	13s	2s	4s	4s	4s	4s

- Go to Grafana - From menu – Dashboard/Browse and click Import Dashboard. Click Upload and select \src\grafana\ dashboard.json. Change name or UID if error accrues



NAME

Exam Board

**⚠ A dashboard or a folder with the same name already exists**

Folder

General

**Unique identifier (UID)**

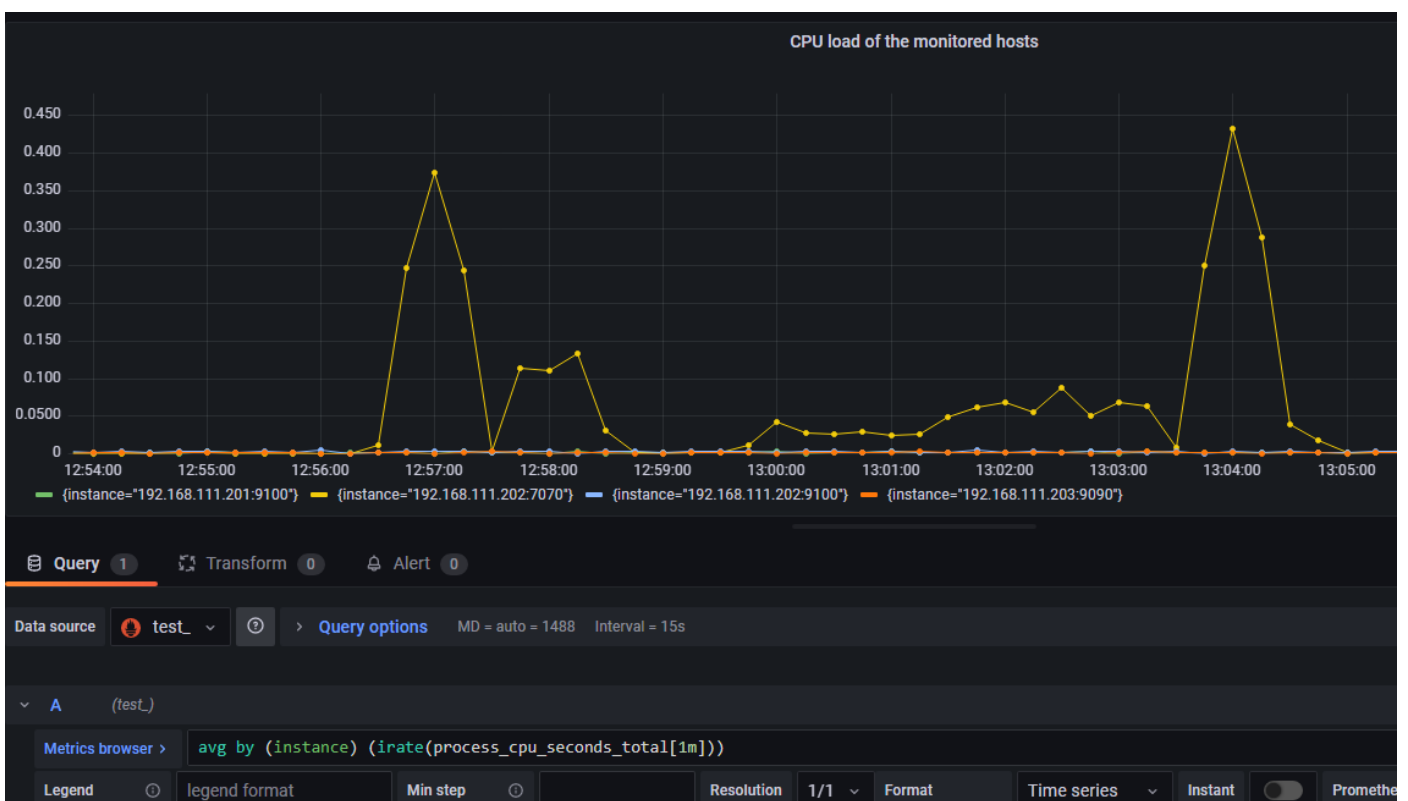
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent UI for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

nkbSW\_ynz

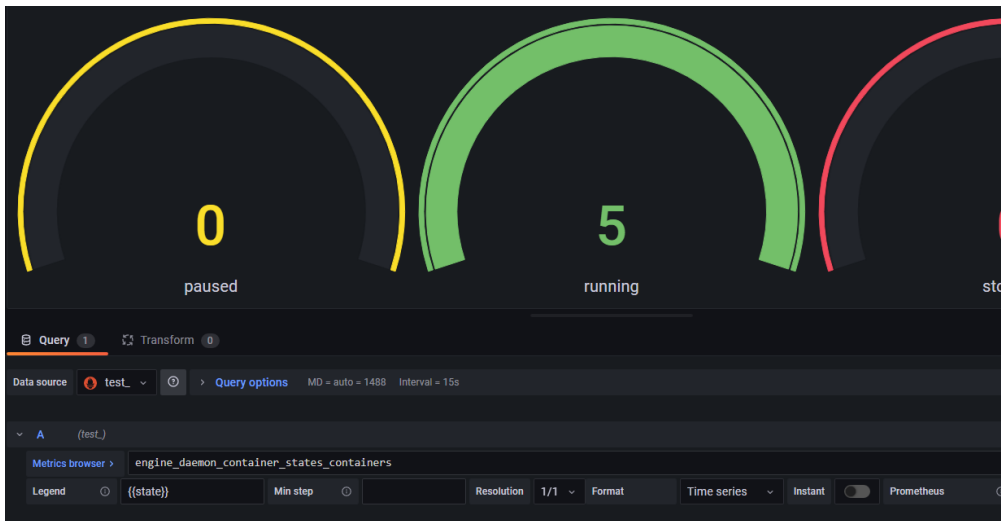
**⚠ Dashboard named 'Exam Board' in folder 'General' has the same UID**

**Import (Overwrite)** **Cancel**

- Check the dashboard – Dashboard/Browse/Exam Board
  - Check – Edit on each Panel
  - CPU load of the monitored hosts - avg by (instance) (irate(process\_cpu\_seconds\_total[1m]))



- Containers - engine\_daemon\_container\_states\_containers



- RAM utilization of the monitored hosts - node\_memory\_Active\_bytes

