

# Files provided for this project

---

- main.tf
- inventory.yml
- caprov-playbook.yml
- config.json
- README.md (pdf)

Create a directory with the above files, main.tf, inventory.yml, caprov-playbook.yml, config.json, README.md (pdf) added. From this point we will call this directory (path/to/your/directory)

- Fork the test repo down below to be able to show the CapRover GitLab app deployed from Gitlab
  - [https://gitlab.com/SaraPetre/u08\\_caprover\\_gitlab](https://gitlab.com/SaraPetre/u08_caprover_gitlab)

## Installation nedded

---

### Terraform

<https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli>

Use the installation guide for your operator system.

Verify the installation

- terraform --version

### Ansible

[https://docs.ansible.com/ansible/latest/installation\\_guide/intro\\_installation.html](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html)

Depending of the operating system you are using, choose the right installation from the above guide.

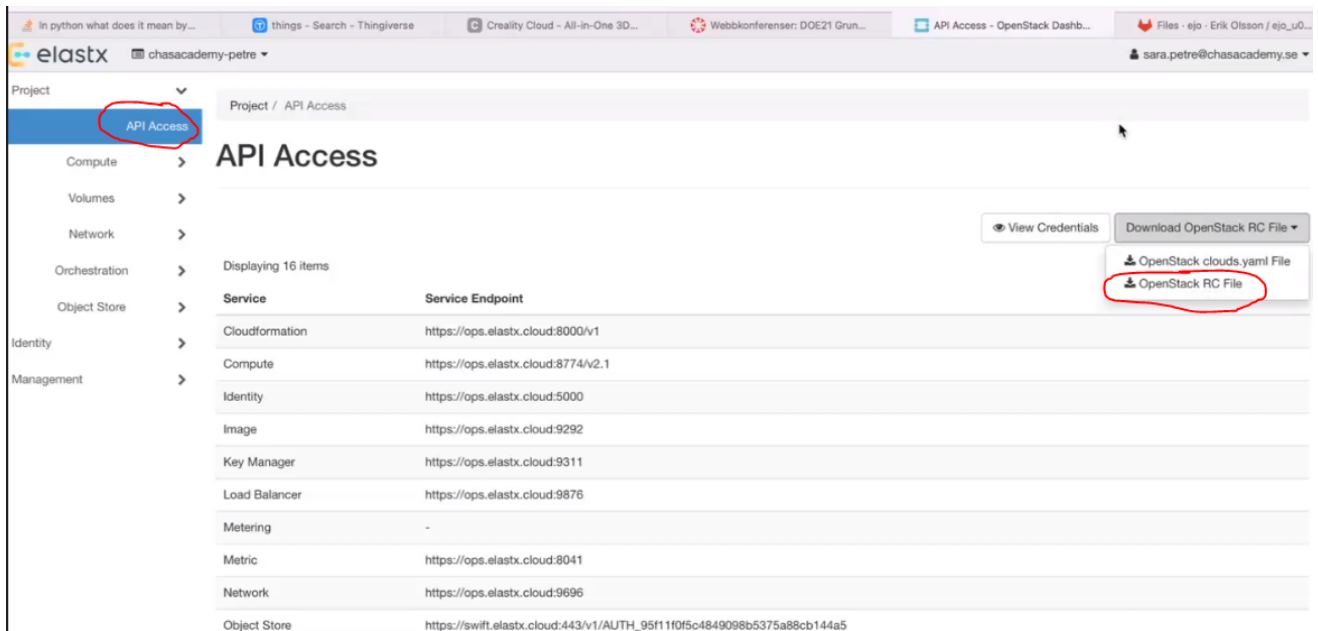
Verify the installation

- ansible --version

## elastx

---

- Log in to your account with your email and provided password
- Go into API Access.
- Go to Download Openstack RC files



- Move your downloaded RC-file into your file-directory(path/to/your/directory)

'chasacademy-petre-openrc.sh' (example of file name)

## VScode

Open VScode in your path/to/your/directory with the above files.

In terminal:

- source 'chasacademy-petre-openrc.sh' (your downloaded file)
- Write your elastx password, when asked for it, in the terminal

## Terraform

In terminal:

- terraform init
- terraform apply
  - yes (to approve the plan and go ahead with apply)
- chmod 600 caprov\_keypair\_rsa

## Ansible

- Go back into your elastx account to verify that your setup is completed. Do that by checking under 'Compute Overview'. All your setups should now be seen.
- Under instances copy your ipaddresses from your server and worker, see picture below, and replace the ipaddresses in the inventory.yml-file **inventory.yml**, see picture down below.

Project / Compute / Instances

# Instances

Instance ID =

Displaying 2 items

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status
caprover-worker	ubuntu-22.04-server-20220607	192.168.1.102, 91.197.41.188	v1-c4-m8-d120	aras_u08_repo_keypair	Active
caprover-server	ubuntu-22.04-server-20220607	192.168.1.101, 91.197.41.163	v1-c4-m8-d120	aras_u08_repo_keypair	Active

```
! inventory.yml × ! caprov-playbook.yml $ setup.sh {} config.json
! inventory.yml
1 all:
2   children:
3     caprov_server:
4       hosts:
5         91.197.41.163:
6           ansible_user: ubuntu
7           ansible_ssh_private_key_file: aras_u08_repo_keypair
8
9     caprov_workers:
10      hosts:
11        91.197.41.188:
12          ansible_user: ubuntu
13          ansible_ssh_private_key_file: aras_u08_repo_keypair
```

Make sure you can ssh into server and worker. In terminal SSH into server:

- `ssh -i caprov_keypair_rsa ubuntu@xxx.xxx.xxx.xxx` (use server ip)
- `exit`

SSH into worker:

- `ssh -i caprov_keypair_rsa ubuntu@xxx.xxx.xxx.xxx` (use runner ip)
- `exit`

Set up your DNS:

- <https://developers.cloudflare.com/dns/manage-dns-records/reference/wildcard-dns-records/>
- <https://www.namecheap.com/support/knowledgebase/article.aspx/597/2237/how-can-i-set-up-a-catchall-wildcard-subdomain/>
- To set up the wildcard you need to use your public server IP-adress
- open your config.json-file and add your personal Domain, new password, email and caproverName(here we have "cap-server").

```

{} config.json > caproverName
1  {}
2      "caproverIP": "192.168.1.101",
3      "caproverPassword": "captain42",
4      "caproverRootDomain": "[your domain]",
5      "newPassword": "[your password]",
6      "certificateEmail": "[add email for cerificate]",
7      "caproverName": "cap-server"
8  }

```

Run ansible:

- `ansible-playbook -i inventory.yml caprov-playbook.yml`

## CapRover cluster

- In your browser. Navigate to CapRover with your domain name. The domain name can be found in your config.json file "caproverRootDomain" (in my case <https://captain.aras.ejo.one/>) and log in.
- Go to Cluster and scroll down to "Alternative Method":
- Follow the instructions and run the commands in your terminal in VScode.

The screenshot shows the CapRover web interface. On the left is a dark sidebar with navigation links: Dashboard, Apps, Monitoring, Cluster (highlighted), and Settings. The main content area has a light background. At the top, there are two input fields: 'New node IP Address' with the value '123.123.123.123' and 'CapRover IP Address' with the value '91.197.41.163'. Below these is a section titled 'SSH Private Key for root' with a text area containing a base64-encoded key. To the right of this text area are two buttons: 'Join as worker node' and 'Join as manager node'. Below these buttons are two more input fields: 'SSH Port' with the value '22' and 'SSH User' with the value 'root'. At the bottom, there's a section titled 'Alternative Method' with text explaining that CapRover uses SSH to connect to nodes. It provides instructions on how to manually join a worker node using the 'docker swarm join' command, showing the command and its output. It also mentions that depending on network configurations, you may need to append the command with '--advertise-addr WORKER\_EXTERNAL\_IP:2377'.

In terminal.

SSH:a into server: Run:

- `sudo docker swarm join-token worker`

Take the output: In my case:

- `docker swarm join --token SWMTKN-1-4mp0om0q2vxzvjq4zlaitcqm19hh6vwf4uxm1oxlklisuucxkj4-1eepwoybp0iih0kknle9fjugu 91.197.41.163:2377`

Logout from server:

- exit

SSH:a into worker:

Run the above output command:

- `sudo docker swarm join --token SWMTKN-1-4mp0om0q2vxzvjq4zlaitcqm19hh6vwf4uxm1oxlksuucxkj4-1eepwoybp0iih0kknle9fjugu91.197.41.163:2377`

Output

- T- his node joined a swarm as a worker.

Working =)

Logout from worker:

- exit

Navigate back to your Caprover: In my case: <https://captain.aras.ejo.one/#/login>

- Log in with your password

Navigate to cluster. You can now see that you are clustered. Se down below picture:

---

Current Cluster Nodes	
<div><div>Node ID: g6uhmnp21v1pj1ojmnk9gdf1g</div><div><div><div>Type: Leader (Main Node)</div><div>State: ready</div><div>RAM: 7.76 GB</div><div>CPU: 4 cores</div><div>Hostname: caprover-server</div></div><div><div>IP: 91.197.41.163</div><div>Status: active</div><div>OS: linux</div><div>Architecture: x86_64</div><div>Docker Version: 20.10.12</div></div></div></div>	
<div><div>Node ID: v0uvteeyaqf56tyerlnkxklco</div><div><div><div>Type: worker</div><div>State: ready</div><div>RAM: 7.76 GB</div><div>CPU: 4 cores</div><div>Hostname: caprover-worker</div></div><div><div>IP: 91.197.41.188</div><div>Status: active</div><div>OS: linux</div><div>Architecture: x86_64</div><div>Docker Version: 20.10.12</div></div></div></div>	

---

## Gitlab deployed on CapRover




<https://caprover.com/docs/ci-cd-integration/deploy-from-gitlab.html>




\*\*!!! In this project a test repo is added to use with the files needed, see link down below. The repo should be forked! You can therefore jump to part 3. !!!

[https://gitlab.com/SaraPetre/u08\\_caprover\\_gitlab](https://gitlab.com/SaraPetre/u08_caprover_gitlab)\*\*






## 1. Create GitLab Repository




SaraPetre > u08\_caprover\_gitlab







**u08\_caprover\_gitlab**   
Project ID: 40801374 


  Star 0  Fork 0







10 Commits 1 Branch 0 Tags 597 KB Project Storage


master u08\_caprover\_gitlab /   Find file  Web IDE   Clone

 **Update README.md**  
SaraPetre authored 1 minute ago  81d5764e 

 README  CI/CD configuration  Add LICENSE  Add CHANGELOG  Add CONTRIBUTING  Add Kubernetes cluster

 Configure Integrations

Name	Last commit	Last update
 images	added readme.md and images	24 minutes ago
 .gitignore	added files for test	1 day ago
 .gitlab-ci.yml	gitlab-ci.yml file added	4 days ago
 Dockerfile	removed part of Dockerfile	1 day ago
 README.md	Update README.md	1 minute ago
 index.php	removed all not used files	43 minutes ago

 README.md

## 2. Add sample Source code-file, Dockerfile and a .gitlab-ci.yml-file. The content ara copied from the documnet:

<https://caprover.com/docs/ci-cd-integration/deploy-from-gitlab.html>

The files and content can be seen in this repo.

## 3. Create CI/CD Variables

Go to your forked project page on GitLab.

Navigate to Settings > CI/CD.

In Variables add the following variables:

- Key : CAPROVER\_URL , Value : <https://captain.root.domain.com> [replace it with your domain]

- Key : CAPROVER\_PASSWORD , Value : mYpAsSwOrD [replace it with your password specified in config.json-file, "newPassword"]
- Key : CAPROVER\_APP , Value : my-test-gitlab-deploy [replace it with the app name you want to create]

## 4. Create an Access Token for CapRover

Navigate to [https://gitlab.com/-/profile/personal\\_access\\_tokens](https://gitlab.com/-/profile/personal_access_tokens) and create a token.

Make sure to assign read\_registry and write\_registry permissions for this token.

## 5. Add Token to CapRover

Login to your CapRover web dashboard, under Cluster click on Add Remote Registry. Then enter these fields:

- Username: your gitlab username
- Password: your gitlab Token [From the previous step]
- Domain: registry.gitlab.com
- Image Prefix: Leave this blank

## 6. Create a CapRover App

On CapRover go to "Apps" and create a new app:

- You need to name the app with the name that you set up in part 3. CAPROVER\_APP value.

## 7. Push to your repo

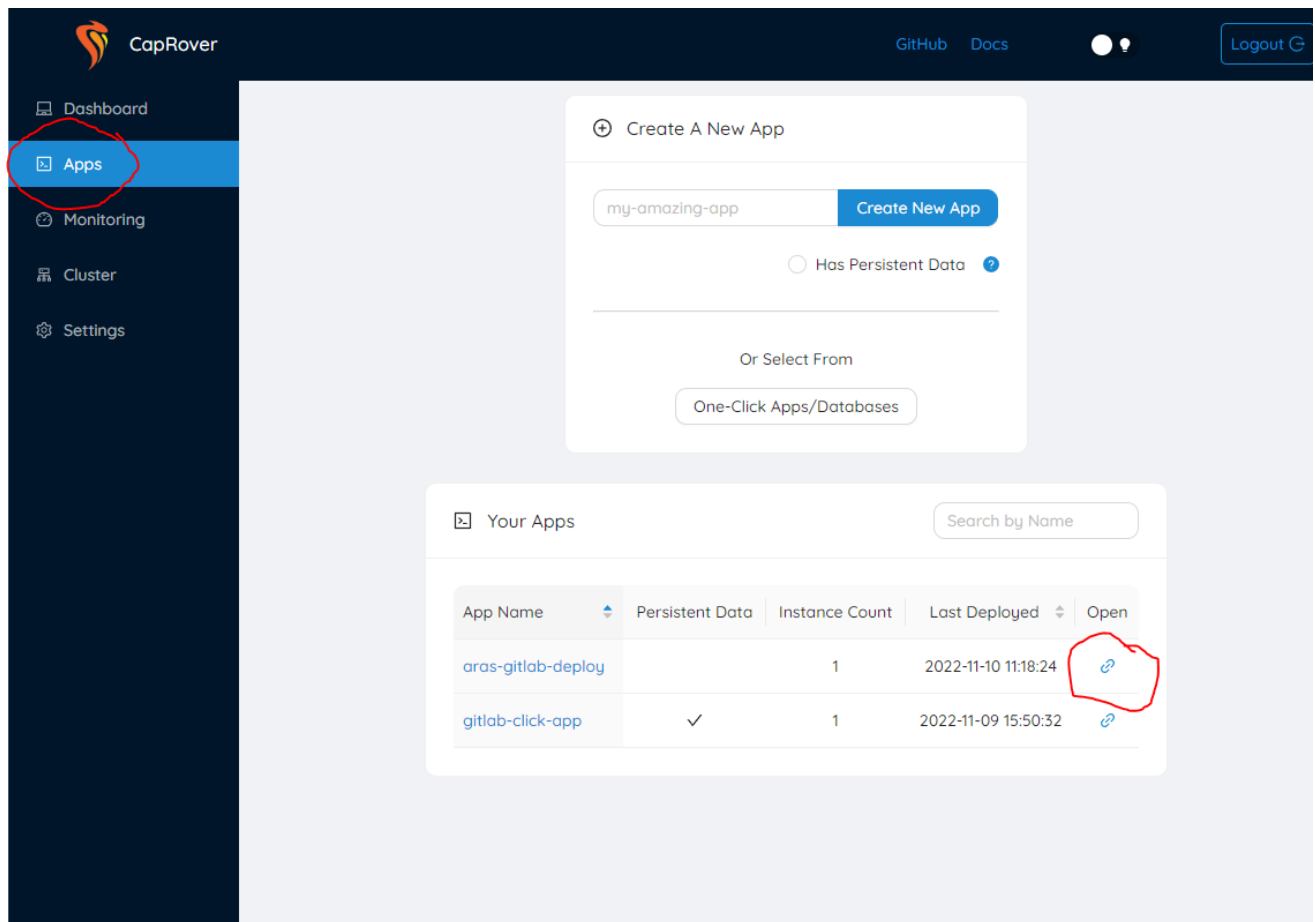
From VSCode and terminal:

- Make some changes in the index.php-file that will be pushed from Gitlab to Caprover app.
- commit and push to your repo, master or main branch. This will trigger the pipeline, CI/CD.

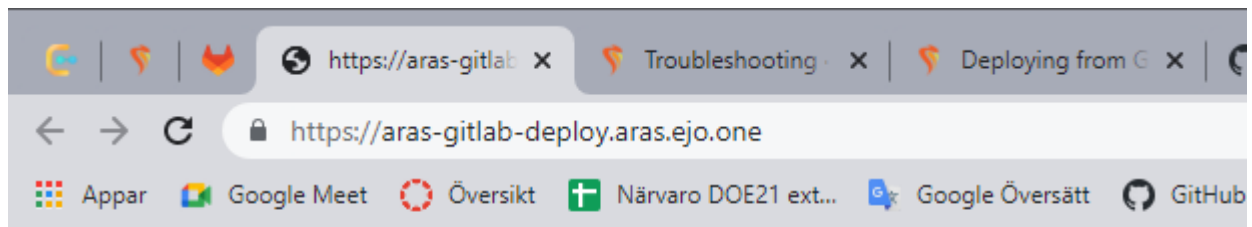
Wait a little bit until your build is finished and deployed automatically! After a few minutes you can be able to see your deployed app on CapRover!

Open CapRover. Log in if needed.

- Navigate to "Apps"
- Click on open on your app.



The output from the index.php-file can now be seen:



PHP output: Hello World from Sara! This is working now=) And today as well #3/221110