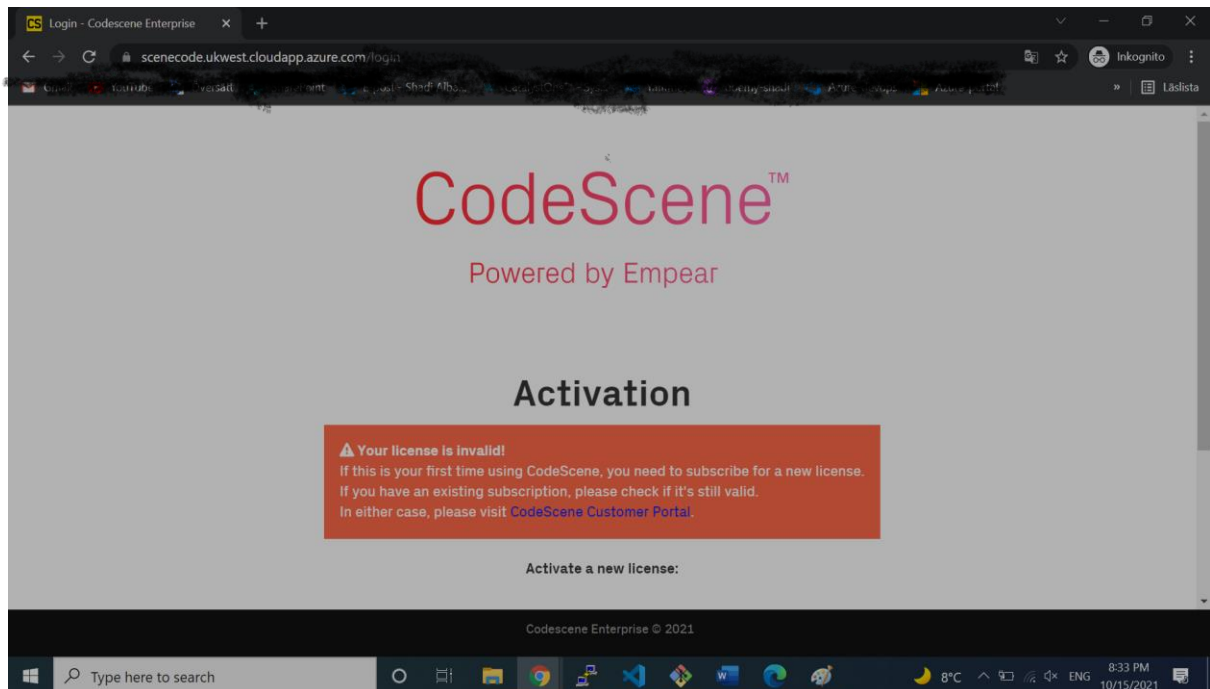


This guide is for you to set up CodeScene in a linux environment, (Azure vm)



Prerequisites:

A running azure vm, with ports open for http, https

A DNS assigned to the vm and ip

A way to access the vm and run commands inside the vm, tex: ssh

Overview

- Creating a shell script, making it executable and running it:
 - It will prepare the environment.
 - It will install required packages.
 - It will clone CodeScene repo.
 - It will issue your ssl-certificate.
 - It will add your DNS to nginx configuration file.
 - It will copy the docker-compose file to your directory with a Codescene folder.
 - Commands to start the docker and docker-compose service.
 - Commands to give privileges to volumes created by docker-compose.
 - Commands to terminate the instance.
 - Command to recreate the docker container with claimed privileges.
- Explanation
- Troubleshooting
- Manual steps

Let's go..

The shell script

First of, start by, heading to the user directory, aka run `cd /home/$USER`

create a file, call it `init.sh`, aka(nano): `nano init.sh`

copy the following lines and paste it inside `init.sh`

copy everything including **first line in green** until you reach:

```
("-----FINISH-----")
```

Give it some time until it asks your **DNS**, add your **DNS** carefully and answer the followed question, and it let it finish running:

```
#!/bin/bash -i

echo "CodeScene"
echo "update"
sudo apt update -y
echo "-----"
echo "upgrade"
sudo apt upgrade -y
echo "-----"
echo "get install software-properties-common"
sudo apt-get install software-properties-common
echo "-----"
echo "repository universe"
sudo add-apt-repository universe
echo "-----"
echo "install docker, docker-compose"
sudo apt install docker.io -y && sudo apt install docker-compose -y
echo "-----"
echo "install mariadb-server, mariadb-client"
sudo apt install mariadb-server -y && sudo apt install mariadb-client -y
echo "-----"
echo "sudo snap install core; sudo snap refresh core"
sudo snap install core; sudo snap refresh core
echo "-----"
echo "snap install --classic certbot"
sudo snap install --classic certbot
echo "-----"
echo "ln -s /snap/bin/certbot /usr/bin/certbot"
sudo ln -s /snap/bin/certbot /usr/bin/certbot
echo "mkdir src , enter it"
mkdir source
cd source
echo "mkdir repos , enter it"
mkdir repos
cd repos
echo "-----"
```

```

echo "git clone"
https://gitlab.com/Shadi.Albatal.SyVe/docker-codescene-nginx-self-signed-
ssl.git
echo "cd docker-codescene-nginx-self-signed-ssl"
cd docker-codescene-nginx-self-signed-ssl/
echo "checkout using_letsencrypt_ssl"
git checkout using_letsencrypt_ssl
echo "-----"
read -p "Enter your dns: " dns
echo "issuing your ssl-cert"
echo sudo certbot certonly --standalone -d $dns
sudo certbot certonly --standalone -d $dns
echo "-----"
echo "adding your dns to source/repos/docker-codescene-nginx-self-signed-
ssl/docker-nginx/nginx.conf"
pwd
cd docker-nginx
pwd
ll
cat nginx.conf
sudo replace "**domain_name**" $dns -- nginx.conf
echo "-----"
echo "review the dns if correctly added to file"
cat nginx.conf
echo "-----"
echo "going to /home/user"
cd /home/$USER
pwd
ll
echo "-----"
echo "creating codescene at user dir"
mkdir codescene
ll /home/$USER
echo "-----"
echo "copy docker-compose to user dir"
cp /home/azureuser/source/repos/docker-codescene-nginx-self-signed-
ssl/initial-files/docker-compose.yml .
pwd
ll
echo "-----"
echo "sudo docker build -t reverseproxy source/repos/docker-codescene-nginx-
self-signed-ssl/docker-nginx"
sudo docker build -t reverseproxy source/repos/docker-codescene-nginx-self-
signed-ssl/docker-nginx
echo "-----"
echo "docker-compose up -d"
sudo docker-compose up -d
pwd

```

```
ls
echo "-----"
echo "sudo chown -R 999 docker-codescene/*"
sudo chown -R 999 docker-codescene/*
echo "-----"
echo "sudo docker-compose down"
sudo docker-compose down
echo "-----"
echo "sudo docker-compose up -d --force-recreate"
sudo docker-compose up -d --force-recreate
echo "-----FINISH-----"
```

Quick Check

- check that your **ssl-certificate** was issued successfully, should look like:

```
Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/scenecode.ukwest.cloudapp.azure.com/fullchain.pem
Key is saved at: /etc/letsencrypt/live/scenecode.ukwest.cloudapp.azure.com/privkey.pem
This certificate expires on 2022-01-13.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

-----
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
* Donating to EFF: https://eff.org/donate-le
-----
```

- check that your **DNS** was added to the **nginx.conf**, just review it by running this long command from your same directory, run this long command from same your directory:
`cat source/repos/docker-codescene-nginx-self-signed-ssl/docker-nginx/nginx.conf`

it will list the whole file, you must see it somewhere in the middle, such as:

(Here my DNS is: **scenecode.ukwest.cloudapp.azure.com**)

```
server {
    listen 443;
    ssl on;
    ssl_certificate /etc/letsencrypt/live/scenecode.ukwest.cloudapp.azure.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/scenecode.ukwest.cloudapp.azure.com/privkey.pem;
    ssl_session_cache shared:SSL:10m;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;
    server_name local.codescene.io;
    index index.html index.htm;
    root /opt/www;
    access_log /nginx/tmp/access.log;
```

- run **ls** in same directory, you must see:
 - 1 file:
 - docker-compose.yaml
 - 2 folders:
 - codescene (empty)
 - Sources/repos/docker-codescene-nginx-self-signed-ssl/

```

azureuser@cs-vm:~/source/repos/docker-codescene-nginx-self-signed-ssl$ ll
total 44
drwxrwxr-x 6 azureuser azureuser 4096 Oct 13 13:39 ./
drwxrwxr-x 3 azureuser azureuser 4096 Oct 13 13:33 ../
drwxrwxr-x 8 azureuser azureuser 4096 Oct 13 13:33 .git/
-rw-rw-r-- 1 azureuser azureuser 108 Oct 13 13:33 .gitignore
-rw-rw-r-- 1 azureuser azureuser 8728 Oct 13 13:33 README.md
drwxrwxr-x 3 azureuser azureuser 4096 Oct 13 13:33 docker-codescene/
-rw-rw-r-- 1 azureuser azureuser 748 Oct 13 13:33 docker-compose.yml
drwxrwxr-x 2 azureuser azureuser 4096 Oct 13 13:44 docker-nginx/
drwxrwxr-x 2 azureuser azureuser 4096 Oct 13 13:33 initial-files/
azureuser@cs-vm:~/source/repos/docker-codescene-nginx-self-signed-ssl$

```

- Check that your docker containers up and running, should see 2 containers:
sudo docker container -ps

Done, everything looks fine?

browse to your DNS address and you should see your CODESCENE login page in front of you, enjoy 😊, if your face is not happy, means something went wrong, so let's troubleshooting:

- if you see nginx bad request, close your tab, wait a minute, open a new private tab, browse again to your DNS address
- if you it is showing offline:
 - review the shell script log, check if any error
 - prune your docker, and provision it again, and browse your DNS again
 - make sure your ss-certificate is correctly existing, and make sure your DNS was correctly added to nginx.conf
- check on internet if some updates have happened and some commands are out of date
- if still nothing works, run the command in the script manually (manual steps provided next page)

Manual setup:

Run these steps manually, one by one

```
sudo apt update -y
```

```
sudo apt upgrade -y
```

```
sudo apt-get install software-properties-common
```

```
sudo add-apt-repository universe
```

```
sudo apt install docker.io -y && sudo apt install docker-compose -y
```

```
sudo snap install core; sudo snap refresh core
```

```
sudo snap install --classic certbot
```

```
sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

```
mkdir source
```

```
cd source
```

```
mkdir repos
```

```
cd repos
```

```
git clone https://github.com/ShadiAlbatal/docker-codescene-nginx-self-signed-ssl.git
```

```
cd docker-codescene-nginx-self-signed-ssl/
```

```
git checkout using_letsencrypt_ssl
```

write the following command, don't forget to replace the: >> your DNS << by your actual **DNS**

```
sudo certbot certonly --standalone -d >> your DNS <<
```

```
cd docker-nginx
```

update **nginx.config**, open the file and manually change **domain_name** to you **DNS** address:

```
nano nginx.conf
```

to save, press **CTRL+x**, followed by **y**, followed by **ENTER**

review the file:

```
cat nginx.conf
```

go to your user directory:

```
cd /home/$USER
```

```
mkdir codescene
```

copy docker-compose to your user directory:

```
cp /home/azureuser/source/repos/docker-codescene-nginx-self-signed-ssl/initial-files/docker-compose.yml .
```

make sure you have got both: **docker-compose.yml** and **codescene** folder:

```
ls
```

```
sudo docker build -t reverseproxy source/repos/docker-codescene-nginx-self-signed-ssl/docker-nginx
```

```
sudo docker-compose up -d
```

should have got a new folder called **docker-codescene**, beside **docker-compose.yml** and **codescene** folder, review that:

```
ls
```

```
sudo chown -R 999 docker-codescene/*
```

```
sudo docker-compose down
```

```
sudo docker-compose up -d --force-recreate
```

Finally done, browse to your **DNS** address, it should be working 😊

****NOTE****

after following the guide and getting it all up and running, there is 3 scenarios:

- 1. The instance keeps running → you don't need to do anything.*
- 2. The instance shut down → everything setup, you only need to make your container up and running again.*
- 3. the instance was totally killed and deleted → you need to create a new instance and start over this guide to setup the environment either using the script or by the manual steps.*