

Computer Systems

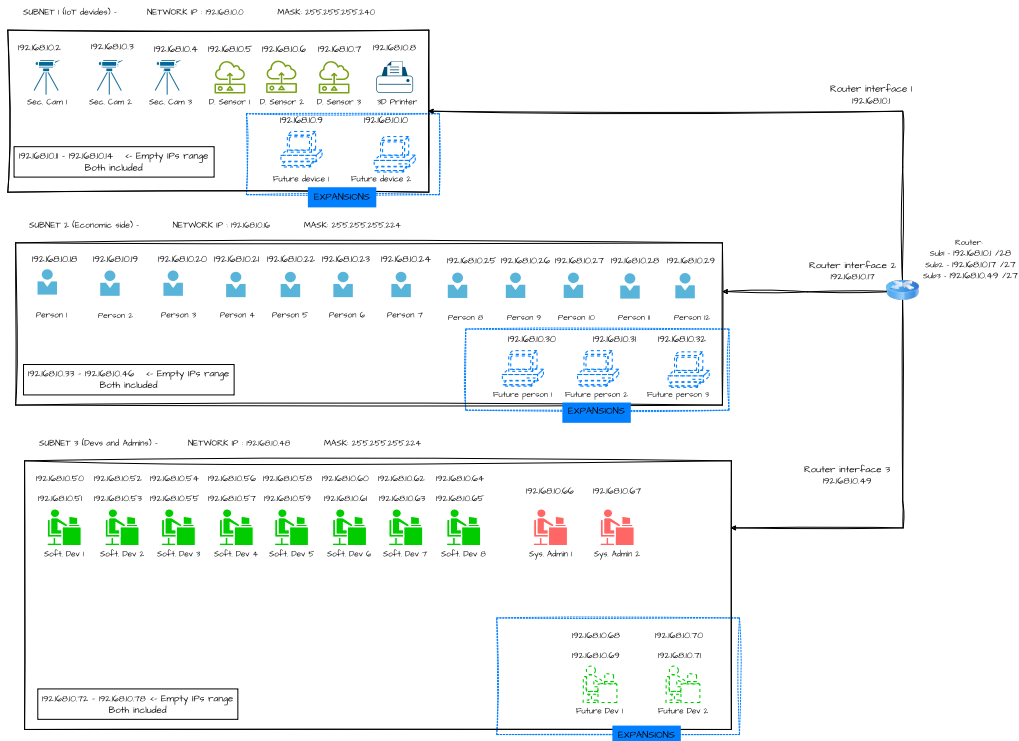
2nd Term Assessable Activity

Author:

Roldán Sanchis Martínez

Exercise 1

Section a



Section b

General info:

Subnet	Network Address	Broadcast Address	Mask	Number of usable hosts
1 (IoT devices)	192.168.10.0	192.168.10.15	/28 - 255.255.255.240	14
2 (economic side)	192.168.10.16	192.168.10.47	/27 - 255.255.255.224	30
3 (devs and admins)	192.168.10.48	192.168.10.79	/27 - 255.255.255.224	30

Network 1 (IoT devices) :

Device	IP
1- Security camera 1	192.168.10.2
2- Security camera 2	192.168.10.3
3- Security camera 3	192.168.10.4
4- Door sensor 1	192.168.10.5
5- Door sensor 2	192.168.10.6
6- Door sensor 3	192.168.10.7
7- 3D Printer	192.168.10.8
8- Future IoT device 1	192.168.10.9
9- Future IoT device 2	192.168.10.10
Router	192.168.10.1

Empty IPs range From 192.168.10.11 to 192.168.10.14

Network 2 (Economical Side):

Device	IP
1- Economical person 1	192.168.10.18
2- Economical person 2	192.168.10.19
3- Economical person 3	192.168.10.20
4- Economical person 4	192.168.10.21
5- Economical person 5	192.168.10.22
6- Economical person 6	192.168.10.23
7- Economical person 7	192.168.10.24
8- Economical person 8	192.168.10.25
9- Economical person 9	192.168.10.26
10- Economical person 10	192.168.10.27
11- Economical person 11	192.168.10.28
12- Economical person 12	192.168.10.29
13- Future Ec. person 1	192.168.10.30
14- Future Ec. person 2	192.168.10.31
15- Future Ec. person 3	192.168.10.32
Router	192.168.10.17
Empty IPs range	From 192.168.10.33 to 192.168.10.46

Network 3 (Devs and Admins):

Device	IP
1- Software dev. 1	192.168.10.50
2- Software dev. 1	192.168.10.51
3- Software dev. 2	192.168.10.52
4- Software dev. 2	192.168.10.53
5- Software dev. 3	192.168.10.54
6- Software dev. 3	192.168.10.55
7- Software dev. 4	192.168.10.56
8- Software dev. 4	192.168.10.57
9- Software dev. 5	192.168.10.58
10- Software dev. 5	192.168.10.59
11- Software dev. 6	192.168.10.60
12- Software dev. 6	192.168.10.61
13- Software dev. 7	192.168.10.62
14- Software dev. 7	192.168.10.63
15- Software dev. 8	192.168.10.64
16- Software dev. 8	192.168.10.65
17- Sys. Admin. 1	192.168.10.66
18- Sys. Admin. 2	192.168.10.67
19- Future soft. dev. 1	192.168.10.68
20- Future soft. dev. 1	192.168.10.69
21- Future soft. dev. 2	192.168.10.70
22- Future soft. dev. 2	192.168.10.71
Router	192.168.10.49
Empty IPs range	From 192.168.10.72 to 192.168.10.48

Exercise 2

Section d

1) *docker run -d --name redis-server -p 8001:8001 --rm redis/redis-stack:latest*

With -d we run the container in detached mode,
with -p 8001:8001 we map the container port 8001 to the machine port 8001.
with --name we establish a name for the container,
and with --rm remove the container when stopped.

2) We can do it locally or using the nano command like this:

nano Dockerfile

And then modifying the content.

3) *docker build -t flask-server /home/rolsanma/CS*

//CS is the base directory where Dockerfile is

With -t we set a tag for the image.

4) *cd app*

to change to the app directory from /home/rolsanma/CS

After that, using

docker run -d -v \$(pwd):/python/app -p 5000:5000 flask-server

With -d we run in detach mode,

with -v we mount the volume that binds /app in the container with /app on the machine,

with -p we map the ports.

Final content of Dockerfile:

FROM python:3.8-slim

COPY requirements.txt /requirements.txt

RUN pip install --no-cache-dir -r /requirements.txt

ENV FLASK_APP=/python/app/app.py

ENV FLASK_RUN_HOST=0.0.0.0

ENV FLASK_ENV=development

ENV REDIS_SERVER=redis-server

CMD ["flask", "run", "--debug"]