

1 European Wireless 2022 Demo instructions

1.1 Demo 1

1. Open a terminal and run the following command:

```
ssh -L 8080:192.168.1.2:80 -N testbed.expeca.proj.kth.se
```

This allows you to access the services running on the testbed by opening a browser window and going to <http://localhost:8080>.

Leave this terminal window running in the background.

2. Open another terminal window:

1. SSH into galadriel:

```
ssh testbed.expeca.proj.kth.se
```

2. cd into the TestbedConfig/ansible folder:

```
cd TestbedConfig/ansible
```

3. Prepare the Python virtual environment:

```
source venv/bin/activate
```

4. Run the cleanup_testbed.sh script:

```
./cleanup_testbed.sh
```

This will cleanup whatever is running on the testbed and prepare it for the demo.

5. Finally, leave this terminal open, as you will need to run the cleanup script again later on.

3. Open a third terminal window:

1. SSH into galadriel:

```
ssh testbed.expeca.proj.kth.se
```

2. cd into the Ainur folder:

```
cd Ainur
```

3. Make sure you are on the correct git branch and you have the latest fixes:

```
git checkout EWDemo2022 && git pull
```

4. Prepare the environment:

```
source venv/bin/activate
source .env
```

5. Run the desired configuration using the `demo1.py` script.

1. To run a setup in which client and server run on the same device:

```
python demo1.py -l local
```

2. To run a setup in which the server runs on the edge:

```
python demo1.py -l edge
```

3. To run a setup in which the server runs on the cloud:

```
python demo1.py -l cloud --region eu-west-2
```

The `--region` option can be any of AWS's compute regions, but I recommend sticking with Stockholm (`eu-north-1`, very low latency), London (`eu-west-2`, low-medium latency), US East Coast (`us-east-1`, high latency), and US West coast (`us-west-1`, very high latency).

4. Any of the above configurations can also be run either over WiFi or Ethernet (default is WiFi). Use the `--phy` flag to specify:

```
python demo1.py --phy ethernet -l cloud --region eu-north-1
```

5. Finally, the `demo1.py` script has a help menu:

```
python demo1.py --help
```

6. Leave the terminal window open during the demo.

4. Open a browser window and head to `http://localhost:8080` once the system is stable. You should see the demo website running there.
5. Finally, when you're done with the demo (or you wish to change the running configuration), run the `cleanup_testbed.sh` script in the second terminal window again. You can close all terminal windows at this point.

1.2 Demo 2

1. Open a terminal window:
 1. SSH into galadriel:
`ssh testbed.expeca.proj.kth.se`
 2. cd into the TestbedConfig/ansible folder:
`cd TestbedConfig/ansible`
 3. Prepare the Python virtual environment:
`source venv/bin/activate`
 4. Run the cleanup_testbed.sh script:
`./cleanup_testbed.sh`
This will cleanup whatever is running on the testbed and prepare it for the demo.
 5. Finally, leave this terminal open, as you may need to run the cleanup script again later on.
2. Open another terminal window:
 1. SSH into galadriel:
`ssh testbed.expeca.proj.kth.se`
 2. cd into the Ainur folder:
`cd Ainur`
 3. Make sure you are on the correct git branch and you have the latest fixes:
`git checkout EWDemo2022 && git pull`
 4. Prepare the environment:
`source venv/bin/activate`
`source .env`
 5. Prepare the demo by editing the file ./demo2_workload_def.yml:
 1. You can tweak cloud instance configuration by editing the cloud section:
 1. `instance_count` changes the amount of cloud instances created.
 2. `region` specifies the AWS where the instances are deployed. Same as above, I recommend sticking with Stockholm (`eu-north-1`, very low latency), London (`eu-west-2`, low-medium latency), US East Coast (`us-east-1`, high latency), and US West coast (`us-west-1`, very high latency).
 2. You can change the type of physical layer deployed using the phy section.
 1. `type` can either be `wifi` or `ethernet`.

3. Next, for the **controller** service, you can tweak the number of controller containers and where they are deployed. These will always be deployed either on the edge or on the cloud. By default, the config deploys 5 containers on the edge, and 5 on the cloud, since the total number of replicas desired is 10, the maximum number of replicas per node is 5, and the only deployment constraint is that the node has a **role** label with the value **backend** (both cloud instances and the edge server have this label). To deploy all the replicas on the cloud, you could do

```
max_replicas_per_node: 10
constraints:
- "node.labels.location==cloud"
- "node.labels.role==backend"
```

To deploy them all to the edge:

```
max_replicas_per_node: 10
constraints:
- "node.labels.location==edge"
- "node.labels.role==backend"
```

Note that you should always deploy at least as many controllers as plants.

4. For the **plant** service, you can tweak the total number of clients by changing the **deploy.replicas** value. Note that we have a total of 10 clients, so 10 is the maximum replicas allowed here. Don't change anything else here.
6. Make sure no old results are stored:

```
sudo rm -rf /opt/expeca/experiments/EWDemo2
```
7. Run the experiment:

```
python demo2.py demo2_workload_def.yml
```

It will run for 3 minutes and then shut down on its own.
8. Once the experiment has finished, you can showcase the resulting CSV files:

```
ls /opt/expeca/experiments/EWDemo2/
```
9. If anything goes wrong, or you decide you want to abort the experiment ahead of time, you can do so by running the **cleanup_testbed.sh** in the other terminal window. Always remember to delete old results between repetitions of the experiment.