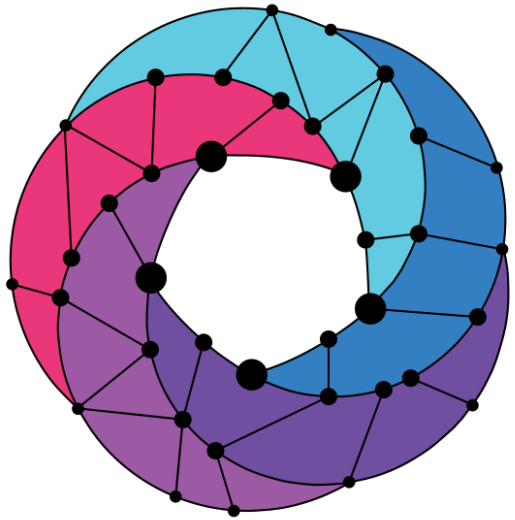


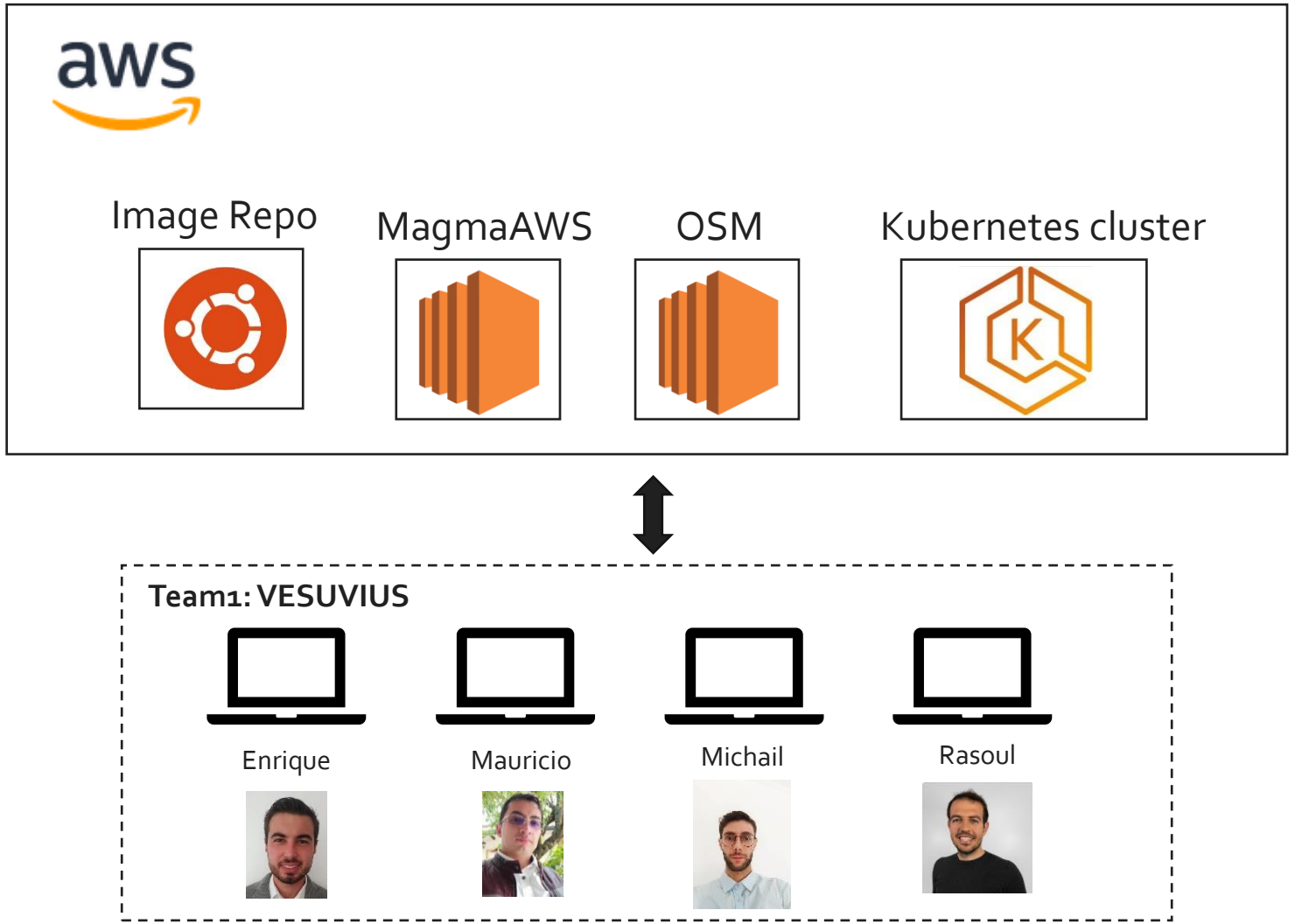
Open Source MANO

OSM#13 Hackfest
Magma 5G Core Onboarding Challenge

Team1: VESUVIUS
Enrique Lluesma Martí (ATOS R&D), Mauricio Alexander Vargas Rodríguez (SENA - CEET), Michail Dalgitsis (VICOMTECH), Rasoul Nikbakht (CTTC)

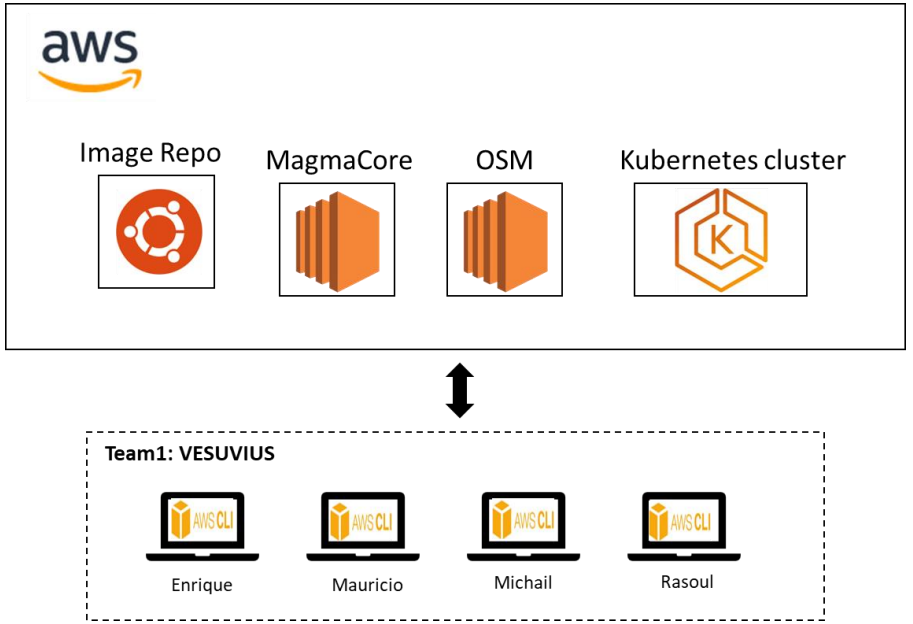
Table of contents

Infrastructure provided by the organizators



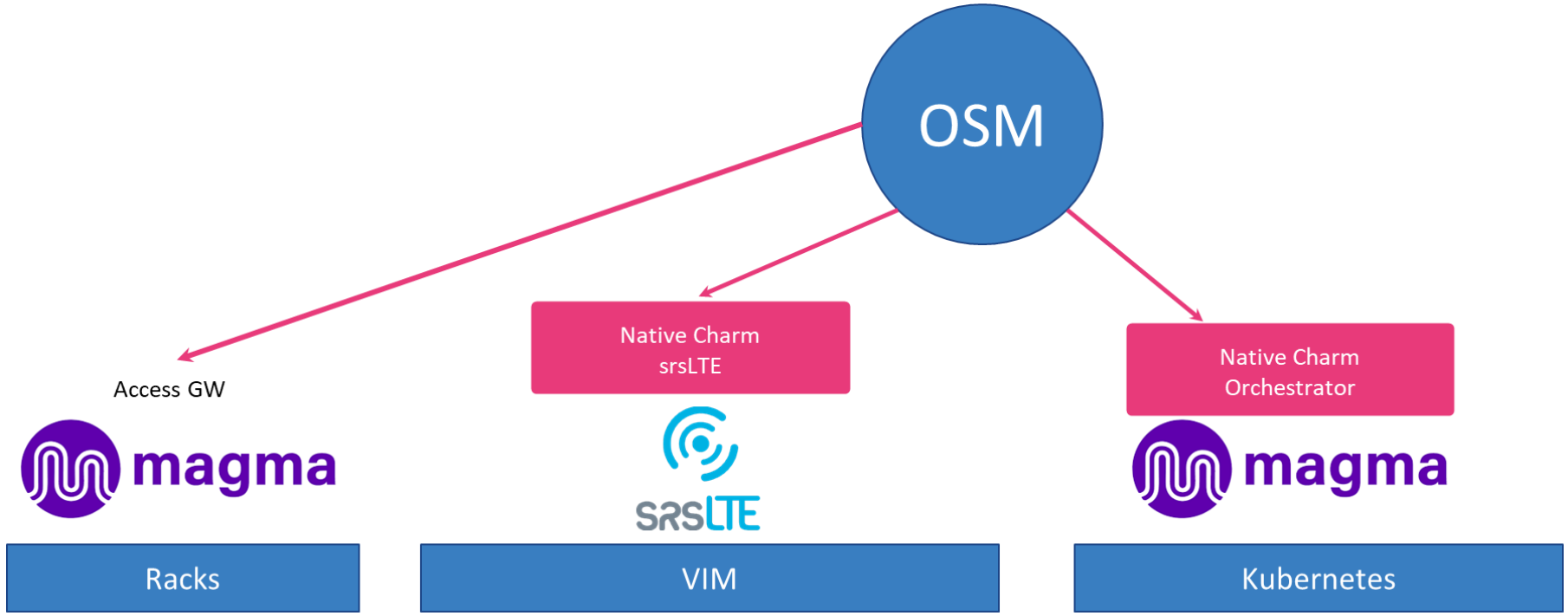
Steps to access the Amazon cloud workspace (AWS)

- Go to Amazon console https://aws.amazon.com/es/console/
- Sign-in with IAM user
  - Account ID: 891210054201
  - IAM user: team1
  - Password: password123!
- Install AWS CLI in a local machine
  - *curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"*
  - *sudo unzip awscliv2.zip*
  - *sudo ./aws/install*
- Configure AWS
  - *aws configure*




```
mdalgitsis@CPU00426:~$ aws configure
AWS Access Key ID [None]: AKIA47ACCHI44VB5FKF7
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
```

Configuration scenario & overview



2.Onboard Magma AGW PNF into OSM to orchestrate Magma AWG

3.Onboard srsLTE eNB and UE simulator VNF into OSM to deploy the eNB

1.Onboard Magma Orchestrator KNF into OSM to deploy Magma Orchestrator into the EKS cluster
Configure AWS Route53
Install admin_operator.pfx in the web browser
Configure Magma Orchestrator through web UI

Open Source
MANO

Deploy Magma Orchestrator

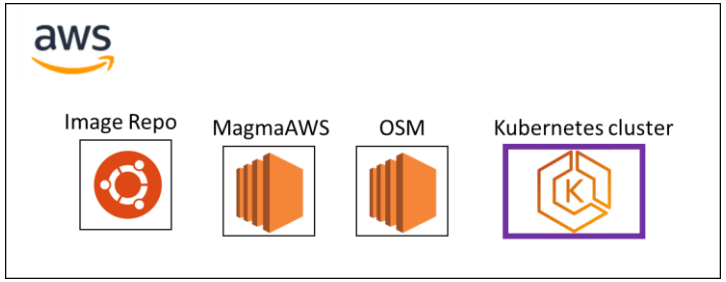
ETSI

Access Kubernetes cluster - EKS

- Install eksctl https://docs.aws.amazon.com/eks/latest/userguide/eksctl.html
- Install kubectl https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/
- Since the K8s cluster is already provided, we need to export the configuration file that corresponds to Team1
  - Make a folder .kube: *mkdir ~/.kube*
  - Copy – paste in .kube the kube_config file from google drive
  - Export configuration: *export KUBECONFIG=~/.kube/kube_config*
  - Check the nodes of the cluster: *kubectl get nodes*

```
mdalgitsis@CPU00426:~$ kubectl get nodes
NAME                           STATUS   ROLES    AGE     VERSION
ip-192-168-40-86.ec2.internal   Ready    <none>   5d20h   v1.22.6-eks-7d68063
ip-192-168-8-194.ec2.internal   Ready    <none>   5d20h   v1.22.6-eks-7d68063
```

  - Other commands to check what is in the EKS cluster:
    - Get a list of all the running pods : *kubectl get pods -A*
    - Get a list of all namespaces: *kubectl get namespaces*

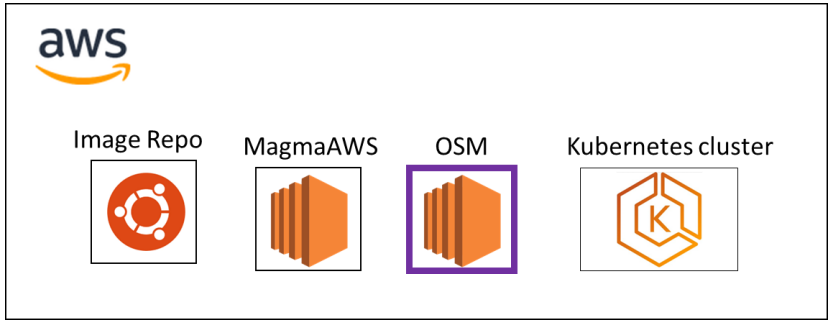


Access OSM EC2 instance (1/2)

- sudo ssh -i team1-key.pem ubuntu@184.73.40.173
- With osm -h we can see all the available commands we can execute through osm machine
  - Check the added VIMs: *osm vim-list*
  - Check the added Kubernetes clusters: *osm k8scluster-list*
- Also, since OSM has been installed upon another Kubernetes cluster with microk8s.kubectl we can interact with it: *microk8s.kubectl get namespaces*

```
ubuntu@ip-192-168-32-40:~$ osm vim-list
+----------+--------------------------------------+-------------------+
| vim name | uuid                                 | operational state |
+----------+--------------------------------------+-------------------+
| aws-site | afd90edc-bb75-4b98-8f39-9ab2a136a624 | ENABLED           |
+----------+--------------------------------------+-------------------+
ubuntu@ip-192-168-32-40:~$ osm k8scluster-list
+-------------+--------------------------------------+----------+-------------------+-------------------+
| Name        | Id                                   | VIM      | Operational State | Op. state details |
+-------------+--------------------------------------+----------+-------------------+-------------------+
| eks-cluster | ae81d925-9656-4ad7-b168-a6a9c9ee9c44 | aws-site | ENABLED           | Helm: ENABLED     |
|             |                                      |          |                   | Juju: ENABLED     |
+-------------+--------------------------------------+----------+-------------------+-------------------+
```

```
ubuntu@ip-192-168-32-40:~$ microk8s.kubectl get namespaces
NAME                 STATUS   AGE
kube-system          Active   4d1h
kube-public          Active   4d1h
kube-node-lease      Active   4d1h
default              Active   4d1h
metallb-system       Active   4d1h
ingress              Active   4d1h
controller-osm-vca   Active   4d1h
osm                  Active   4d1h
```



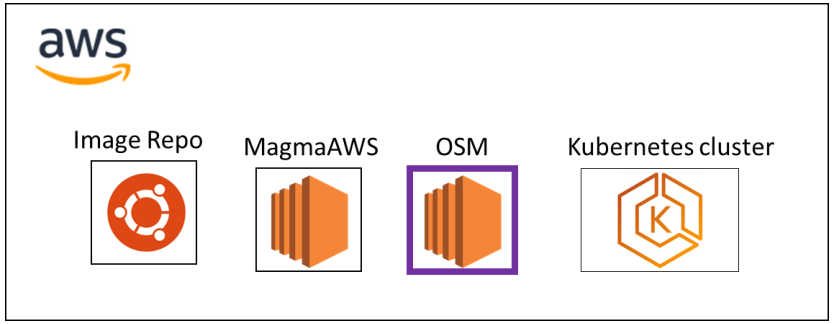
Access OSM EC2 instance (2/2)

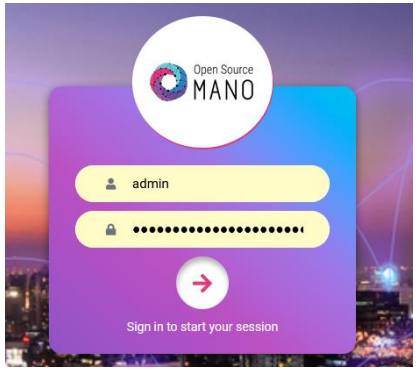
- Lastly, we can access OSM GUI´s on:

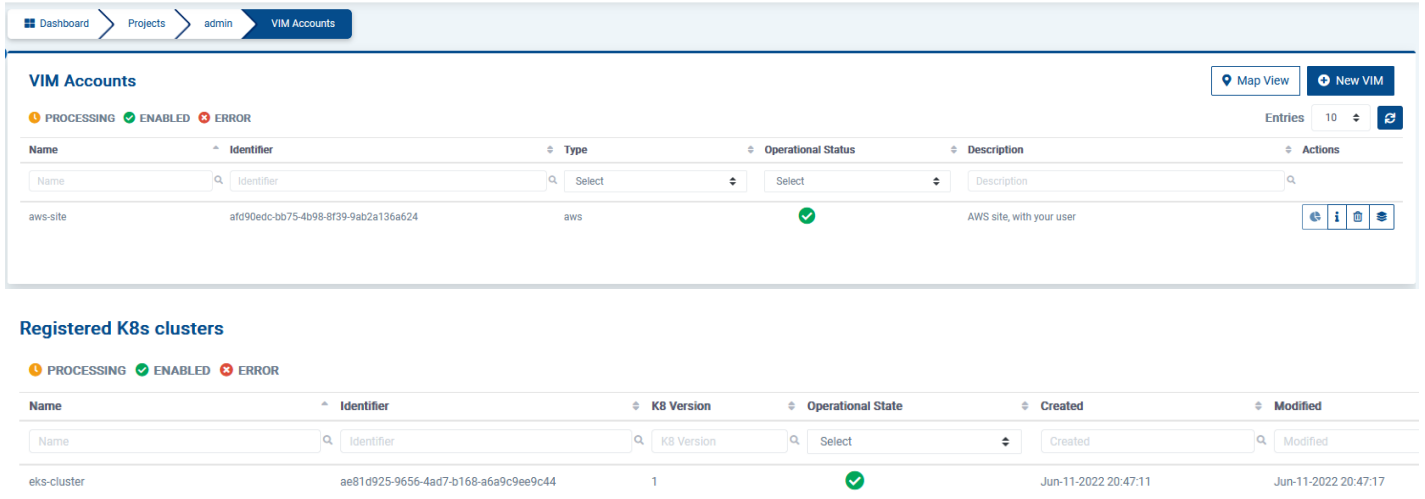
https://ui.184.73.40.173.nip.io/login

  - Username: admin
  - ubuntu@ip-192-168-32-40:~$ env | grep OSM_PASSWORD

OSM_PASSWORD=8975766776cb4a543213b6ec122bdc91

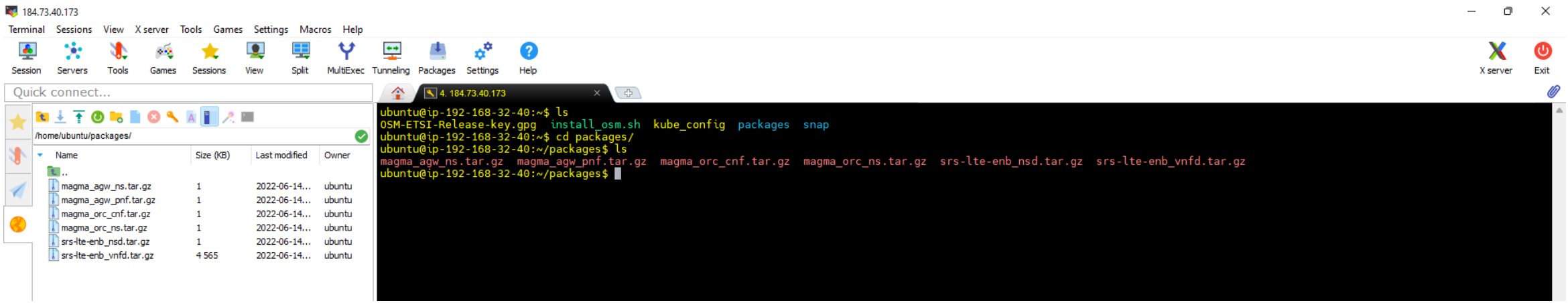






Magma orchestrator Onboarding to OSM (1/3)

- On the OSM instance we have created a directory called packages and uploaded all the files provided for Magma orchestrator, AGW and srsLTE though SSH:



Magma orchestrator Onboarding to OSM (2/3)

- Onboarding to OSM
  - *osm nfpkg-create magma_orc_cnf.tar.gz*

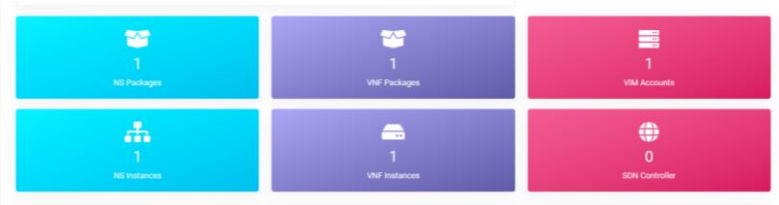
magma_orc_cnf	c9517635-a813-44f6-b235-f10d4fff9066	1.0	Canonical	K8s container deployment of Magma Orchestrator

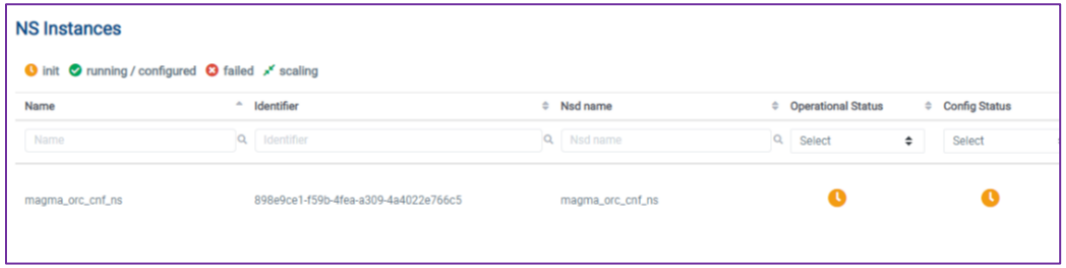
  - *osm nspkg-create magma_orc_ns.tar.gz*

magma_orc_cnf_ns	7ddb5e05-fa4f-4c58-9ade-50dbf493e3f0	1.0	Canonical	NS with 1 KDU connected to the mgmtnet VL

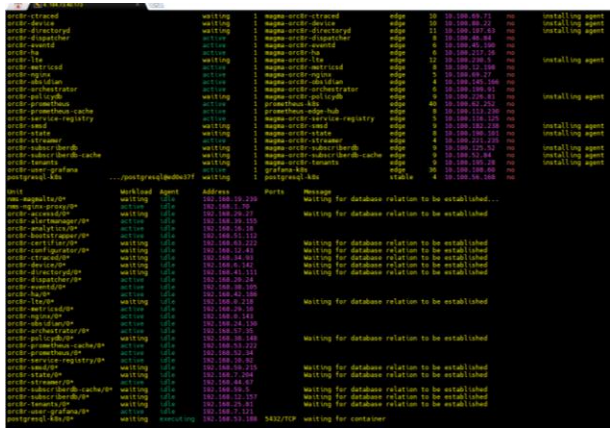
- Instantiating
  - *osm ns-create --ns_name magma_orc_cnf_ns --nsd_name magma_orc_cnf_ns --vim_account aws-site*
  - Magma ORC takes around 15 min to be deployed:

juju status -m magma-orc-kdu-898e9ce1-f59b-4fea-a309-4a4022e766c5 --watch 10s



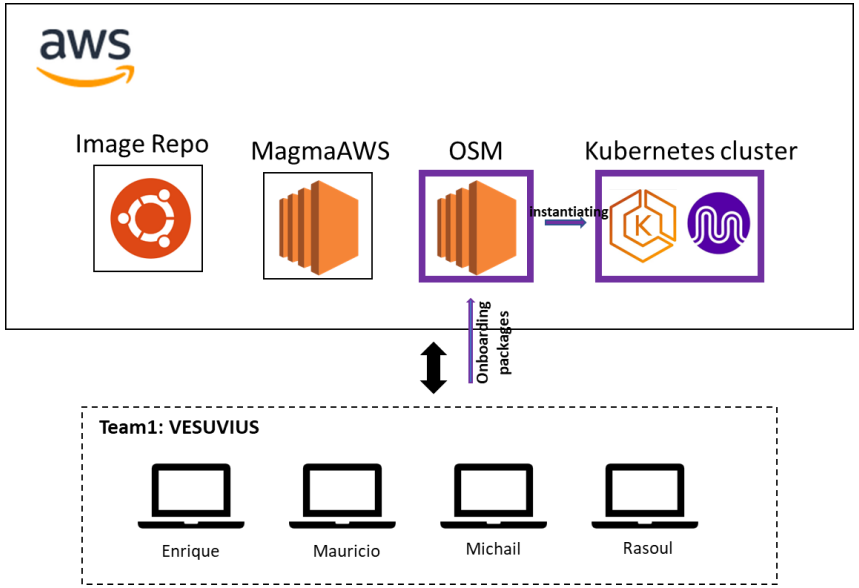


magma_orc_cnf_ns	898e9ce1-f59b-4fea-a309-4a4022e766c5	magma_orc_cnf_ns	✓	✓	Done



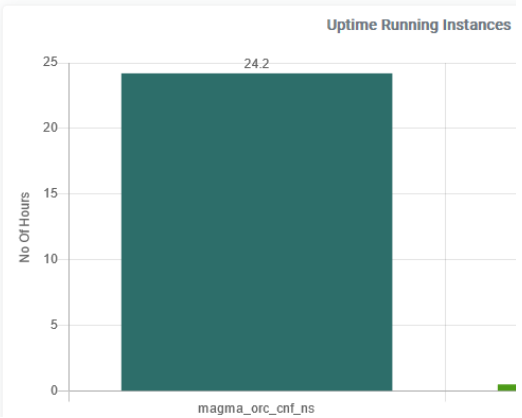
Magma orchestrator Onboarding to OSM (3/3)

- After onboarding the KNF and NS packages and instantiating the network service, the Magma Orchestrator is running on the EKS




```
mdalgitsis@CPU00426:~$ kubectl get pods -n magma-orc-kdu-898e9ce1-f59b-4fea-a309-4a4022e766c5
NAME                                READY   STATUS    RESTARTS   AGE
modeloperator-6c87ffbb48-7bzlf      1/1     Running   0          19h
nms-magmalte-0                      2/2     Running   0          19h
nms-nginx-proxy-0                   2/2     Running   0          19h
orc8r-accessd-0                     2/2     Running   0          19h
orc8r-alertmanager-0                2/2     Running   0          19h
orc8r-analytics-0                   2/2     Running   0          19h
orc8r-bootstrapper-0                2/2     Running   0          19h
orc8r-certifier-0                   2/2     Running   0          19h
orc8r-configurator-0                2/2     Running   0          19h
orc8r-ctraced-0                     2/2     Running   0          19h
orc8r-device-0                      2/2     Running   0          19h
orc8r-directoryd-0                  2/2     Running   0          19h
orc8r-dispatcher-0                  2/2     Running   0          19h
orc8r-eventd-0                      2/2     Running   0          19h
orc8r-ha-0                          2/2     Running   0          19h
orc8r-lte-0                         2/2     Running   0          19h
orc8r-metricsd-0                    2/2     Running   0          19h
orc8r-nginx-0                       2/2     Running   0          19h
orc8r-obsidian-0                    2/2     Running   0          19h
orc8r-orchestrator-0                2/2     Running   0          19h
orc8r-policydb-0                    2/2     Running   0          19h
orc8r-prometheus-0                  2/2     Running   0          19h
orc8r-prometheus-cache-0            2/2     Running   0          19h
orc8r-service-registry-0            2/2     Running   0          19h
orc8r-smsd-0                        2/2     Running   0          19h
orc8r-state-0                       2/2     Running   0          19h
orc8r-streamer-0                    2/2     Running   0          19h
orc8r-subscriberdb-0                2/2     Running   0          19h
orc8r-subscriberdb-cache-0          2/2     Running   0          19h
orc8r-tenants-0                     2/2     Running   0          19h
orc8r-user-grafana-0                2/2     Running   0          19h
postgresql-k8s-0                    1/1     Running   0          19h
postgresql-k8s-operator-0           1/1     Running   0          19h
mdalgitsis@CPU00426:~$
```

Kubernetes EKS cluster



OSM EC2 machine

AWS Route53 configuration (1/3)

- After instantiating Magma orchestrator, there are services running in the EKS cluster related to it.
- With *kubectl get services -n magma-orc-kdu-898e9ce1-f59b-4fea-a309-4a4022e766c5* we can see all these services
- However we are interested in to the LoadBalancer service type
  - *kubectl get services -n magma-orc-kdu-898e9ce1-f59b-4fea-a309-4a4022e766c5 | grep LoadBalancer*
- Here we can find the Internal and External IP addresses

```
mdalgitsis@CPU00426:~$ kubectl get services -n magma-orc-kdu-898e9ce1-f59b-4fea-a309-4a4022e766c5 | grep LoadBalancer
nginx-proxy                         LoadBalancer   10.100.142.226   a9595c63431634e0ba3286a6f36bbf4b-579836936.us-east-1.elb.amazonaws.com   443:30037/TCP
                                        25h
orc8r-bootstrap-nginx               LoadBalancer   10.100.255.13    a86135fcab10f49f2bb0cf01832a0931-1815485410.us-east-1.elb.amazonaws.com   80:31200/TCP,443:3
0747/TCP,8444:30618/TCP                 25h
orc8r-clientcert-nginx              LoadBalancer   10.100.231.81    a8b9cef61fb434ca6b93d6a54e1b76af-477984819.us-east-1.elb.amazonaws.com   80:32284/TCP,443:3
1742/TCP,8443:32118/TCP                 25h
orc8r-nginx-proxy                   LoadBalancer   10.100.145.38    a74e7ff9c1c57419f8629f226fc5e593-640115595.us-east-1.elb.amazonaws.com   80:32199/TCP,8443:
31395/TCP,8444:32671/TCP,443:30338/TCP   25h
```

AWS Route53 configuration (2/3)

- To configure AWS Route53 appropriately we need to create A records.
- One way is through the AWS console and another is through the AWS CLI.
- Through the AWS CLI, we need to create an arecords.json file like the one on the figure
- The DNSName field must be filled according to the services from the previous slide using the external IP.
- We apply this file with the command:
  - *aws route53 change-resource-record-sets --hosted-zone-id Z0309839BG6MW88K4E1V --change-batch file://arecords.json*
- **NOTE**: We execute this command from our local terminal, neither from EKS nor from any EC2 instance

```
mdalgitsis@CPU00426:~$ cat arecords.json
{
"Comment": "Create A Records for VESUVIUS ",
"Changes": [
   {
     "Action": "CREATE",
     "ResourceRecordSet": {
       "Name": "*.nms.team1.osmhackfest.com",
       "Type": "A",
       "AliasTarget": {
         "HostedZoneId": "Z35SXDOTRQ7X7K",
         "DNSName": "a9595c63431634e0ba3286a6f36bbf4b-579836936.us-east-1.elb.amazonaws.com",
         "EvaluateTargetHealth": true
       }
     }
   },
   {
     "Action": "CREATE",
     "ResourceRecordSet": {
       "Name": "controller.team1.osmhackfest.com",
       "Type": "A",
       "AliasTarget": {
         "HostedZoneId": "Z35SXDOTRQ7X7K",
         "DNSName": "a8b9cef61fb434ca6b93d6a54e1b76af-477984819.us-east-1.elb.amazonaws.com",
         "EvaluateTargetHealth": true
       }
     }
   },
   {
     "Action": "CREATE",
     "ResourceRecordSet": {
       "Name": "api.team1.osmhackfest.com",
       "Type": "A",
       "AliasTarget": {
         "HostedZoneId": "Z35SXDOTRQ7X7K",
         "DNSName": "a74e7ff9c1c57419f8629f226fc5e593-640115595.us-east-1.elb.amazonaws.com",
         "EvaluateTargetHealth": true
       }
     }
   },
   {
     "Action": "CREATE",
     "ResourceRecordSet": {
       "Name": "bootstrapper-controller.team1.osmhackfest.com",
       "Type": "A",
       "AliasTarget": {
         "HostedZoneId": "Z35SXDOTRQ7X7K",
         "DNSName": "a86135fcab10f49f2bb0cf01832a0931-1815485410.us-east-1.elb.amazonaws.com",
         "EvaluateTargetHealth": true
       }
     }
   }
 ]
}
```

AWS Route53 configuration (3/3)

- We can confirm that all run well either by AWS CLI or by AWS Console

Confirm by AWS CLI

```
mdalgitsis@CPU00426:~$ aws route53 change-resource-record-sets --hosted-zone-id Z0309839BG6MW88K4E1V --change-batch file://arecords.json
{
    "ChangeInfo": {
        "Id": "/change/C0542281RQZKUSPRZUH4",
        "Status": "PENDING",
        "SubmittedAt": "2022-06-15T12:38:42.566000+00:00",
        "Comment": "Create A Records for VESUVIUS "
    }
}
```

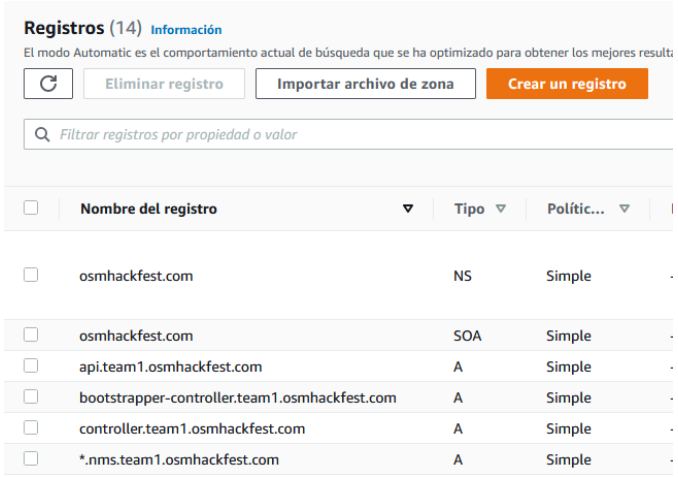
```
mdalgitsis@CPU00426:~$ aws route53 get-change --id /change/C0542281RQZKUSPRZUH4
{
    "ChangeInfo": {
        "Id": "/change/C0542281RQZKUSPRZUH4",
        "Status": "INSYNC",
        "SubmittedAt": "2022-06-15T12:38:42.566000+00:00",
        "Comment": "Create A Records for VESUVIUS "
    }
}
```

Description

Returns the current status of a change batch request. The status is one of the following values:

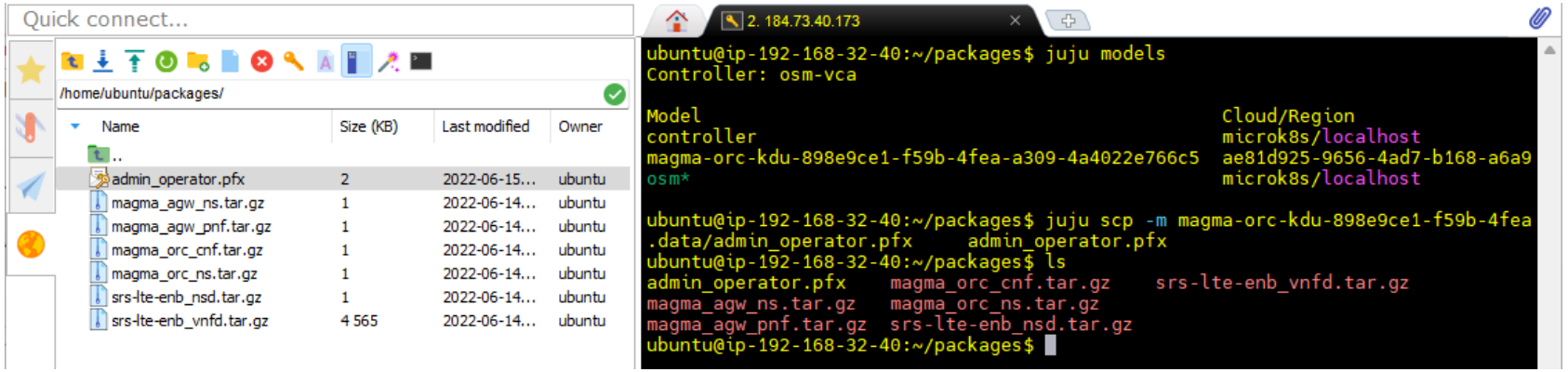
- `PENDING` indicates that the changes in this request have not propagated to all Amazon Route 53 DNS servers. This is the initial status of all change batch requests.
- `INSYNC` indicates that the changes have propagated to all Route 53 DNS servers.

Confirm by AWS Console



Download certificates – Part 1 (1/2)

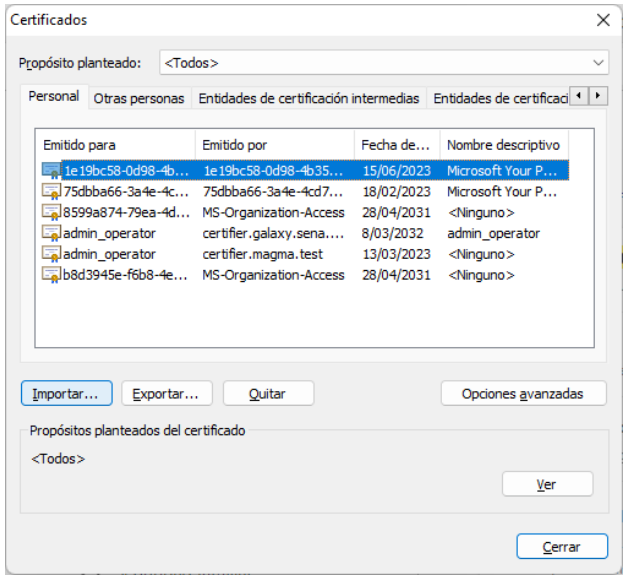
- Magma Orchestrator installation provides us a GUI. To access it we need to use a certificate
- We need to check the models and download the Orchestrator´s certificates for our browser from the **OSM machine** with the juju command:

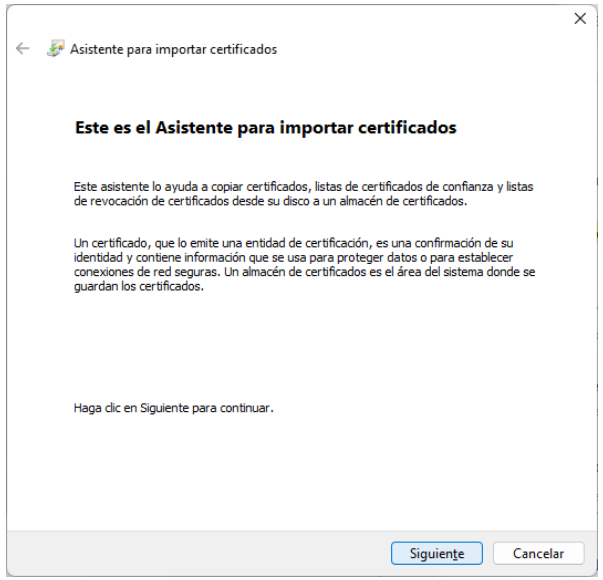


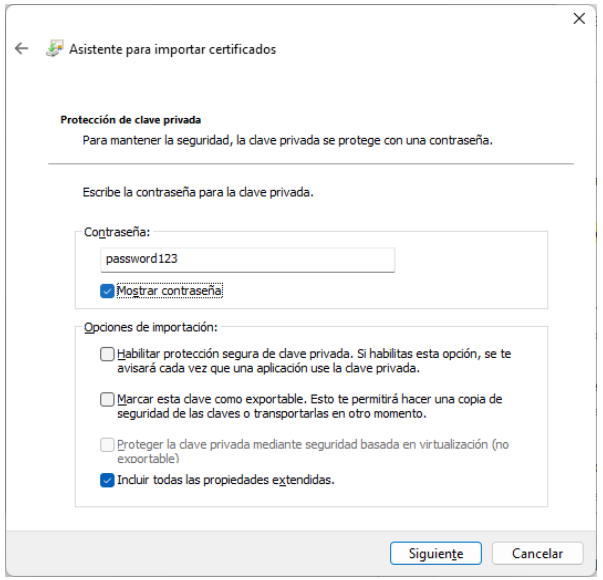
- Export it to our local environment

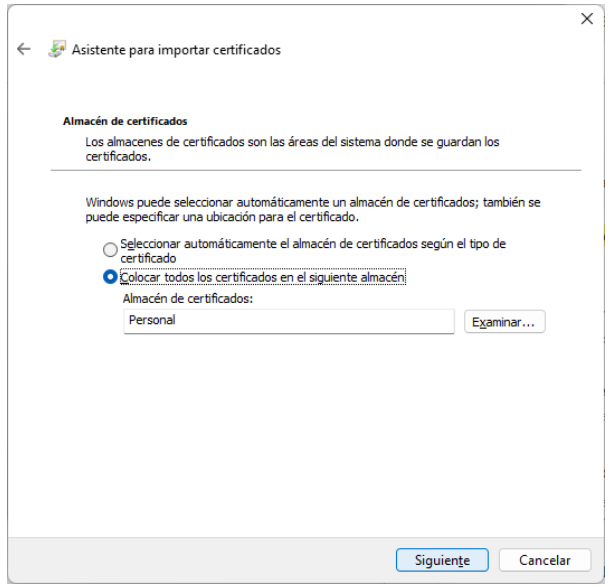
Download certificates – Part 1 (2/2)

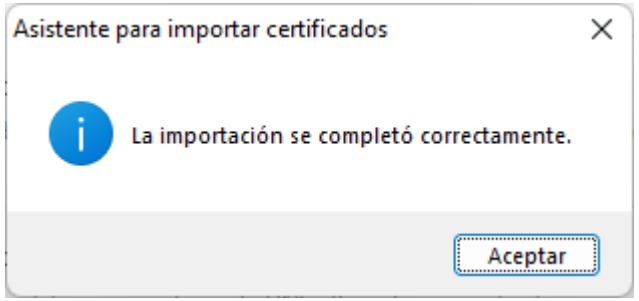
- Install the admin_operator.pfx certificate on the browser with the password: password123





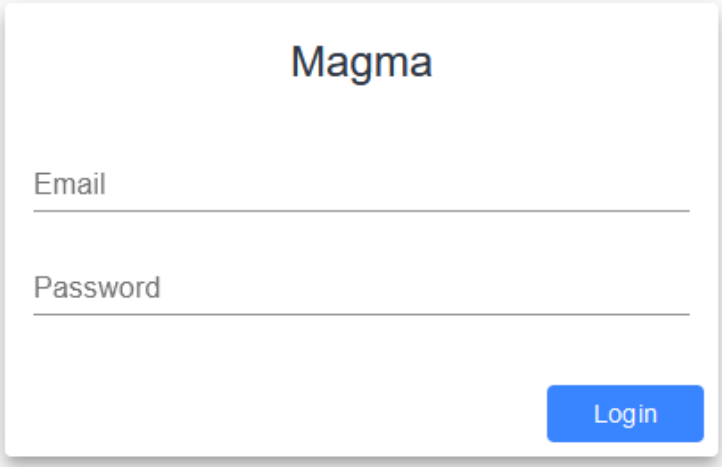






Day-2 actions to get credentials for Magma Orchestrator GUI (1/2)

- Get the GUI credentials
- $ osm ns-action magma_orc_cnf_ns \
- --vnf_name magma_orc_cnf \
- --kdu_name magma-orc-kdu \
- --action_name get-admin-credentials
- $ osm ns-op-show 989205c5-c081-4957-9cbd-c11e532578c3

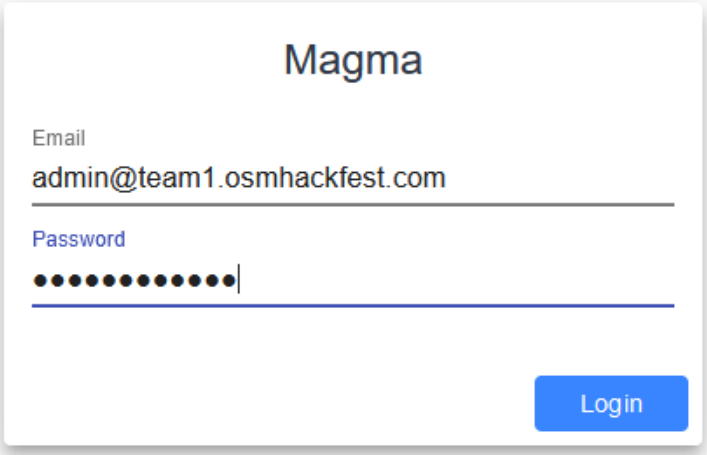


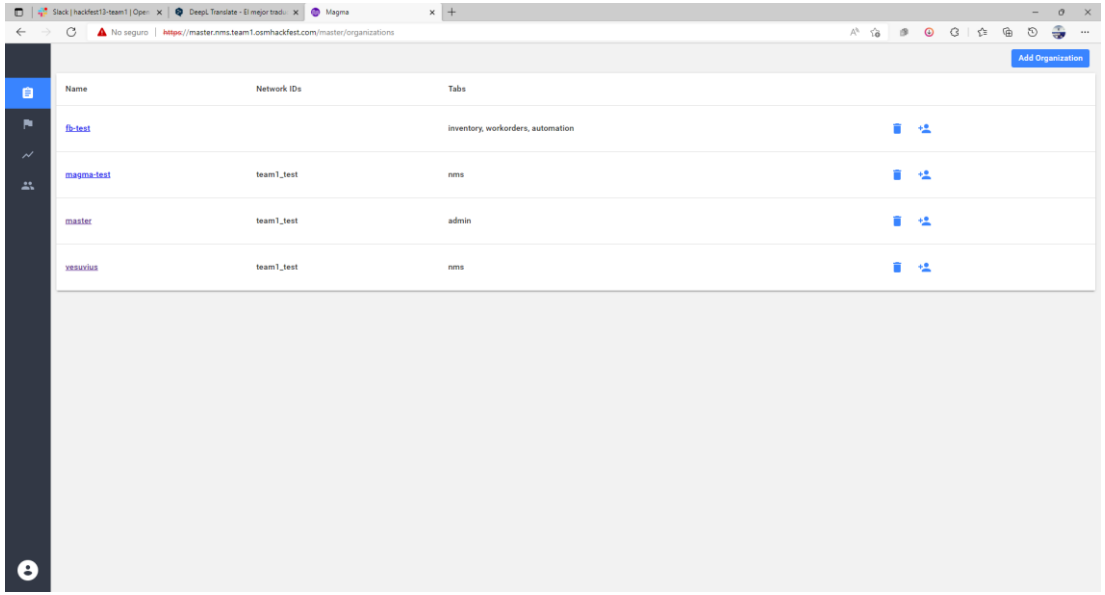

```
ubuntu@ip-192-168-32-40:~$ osm ns-action magma_orc_cnf_ns \
>        --vnf_name magma_orc_cnf \
>        --kdu_name magma-orc-kdu \
>        --action_name get-admin-credentials

989205c5-c081-4957-9cbd-c11e532578c3
ubuntu@ip-192-168-32-40:~$
ubuntu@ip-192-168-32-40:~$ osm ns-op-show 989205c5-c081-4957-9cbd-c11e532578c3
+----------------------+-----------------------------------------------------------------------------------------+
| field                | value                                                                                   |
+----------------------+-----------------------------------------------------------------------------------------+
| _id                  | "989205c5-c081-4957-9cbd-c11e532578c3"                                                  |
| id                   | "989205c5-c081-4957-9cbd-c11e532578c3"                                                  |
| operationState       | "COMPLETED"                                                                             |
| queuePosition        | 0                                                                                       |
| stage                | ""                                                                                      |
| errorMessage         | ""                                                                                      |
| detailedStatus       | null                                                                                    |
| statusEnteredTime    | 1655321980.273547                                                                       |
| nsInstanceId         | "898e9ce1-f59b-4fea-a309-4a4022e766c5"                                                  |
| lcmOperationType     | "action"                                                                                |
| startTime            | 1655321971.1564357                                                                      |
| isAutomaticInvocation| false                                                                                   |
| operationParams      | {                                                                                       |
|                      |   "member_vnf_index": "magma_orc_cnf",                                                  |
|                      |   "kdu_name": "magma-orc-kdu",                                                          |
|                      |   "primitive": "get-admin-credentials",                                                 |
|                      |   "primitive_params": {},                                                               |
|                      |   "lcmOperationType": "action",                                                         |
|                      |   "nsInstanceId": "898e9ce1-f59b-4fea-a309-4a4022e766c5"                                |
|                      | }                                                                                       |
| isCancelPending      | false                                                                                   |
| links                | {                                                                                       |
|                      |   "self": "/osm/nslcm/v1/ns_lcm_op_occs/989205c5-c081-4957-9cbd-c11e532578c3",          |
|                      |   "nsInstance": "/osm/nslcm/v1/ns_instances/898e9ce1-f59b-4fea-a309-4a4022e766c5"        |
|                      | }                                                                                       |
| _admin               | {                                                                                       |
|                      |   "created": 1655321971.1564708,                                                        |
|                      |   "modified": 1655321980.2735484,                                                       |
|                      |   "projects_read": [                                                                    |
|                      |     "148d26145a1c48b3be135a71d28fdf3d"                                                  |
|                      |   ],                                                                                    |
|                      |   "projects_write": [                                                                   |
|                      |     "148d26145a1c48b3be135a71d28fdf3d"                                                  |
|                      |   ],                                                                                    |
|                      |   "worker": "619bc1924b80"                                                              |
|                      | }                                                                                       |
| detailed-status      | {                                                                                       |
|                      |   "Code": "0",                                                                          |
|                      |   "admin-password": "eFFMGLqloElv",                                                     |
|                      |   "admin-username": "admin@team1.osmhackfest.com"                                       |
|                      | }                                                                                       |
+----------------------+-----------------------------------------------------------------------------------------+
ubuntu@ip-192-168-32-40:~$
```

Day-2 actions to get credentials for Magma Orchestrator GUI (2/2)

- Access the https://master.nms.team1.osmhackfest.com with the credentials

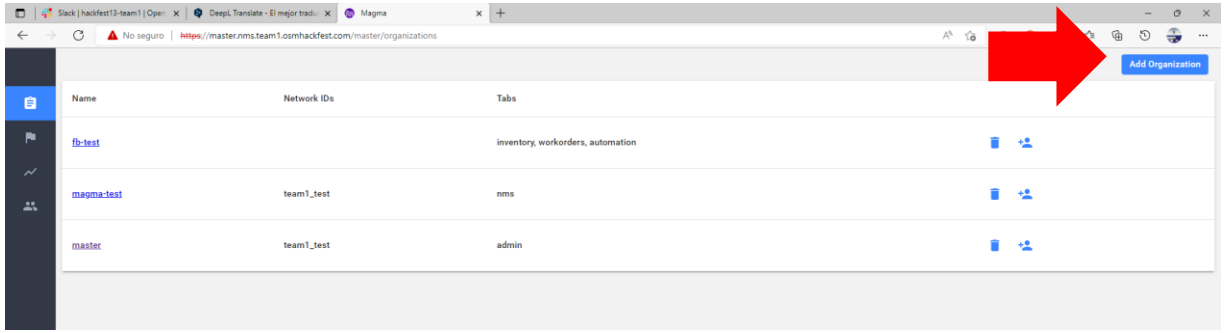
```
|                      |   "worker": "619bc1924b80"
|                      | }
| detailed-status      | {
|                      |   "Code": "0",
|                      |   "admin-password": "eFFMGLqloElv",
|                      |   "admin-username": "admin@team1.osmhackfest.com"
|                      | }
+----------------------+
```

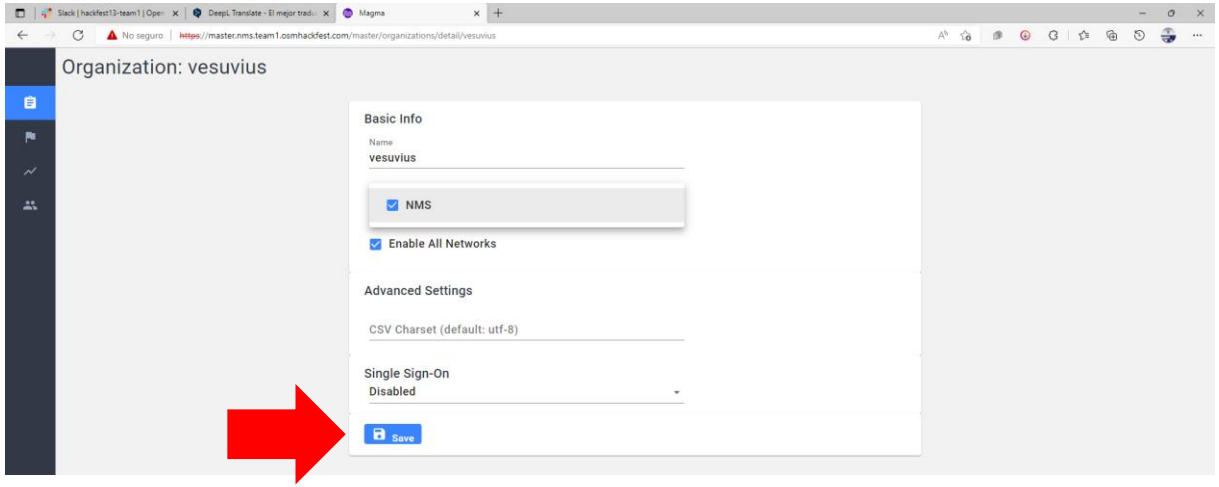


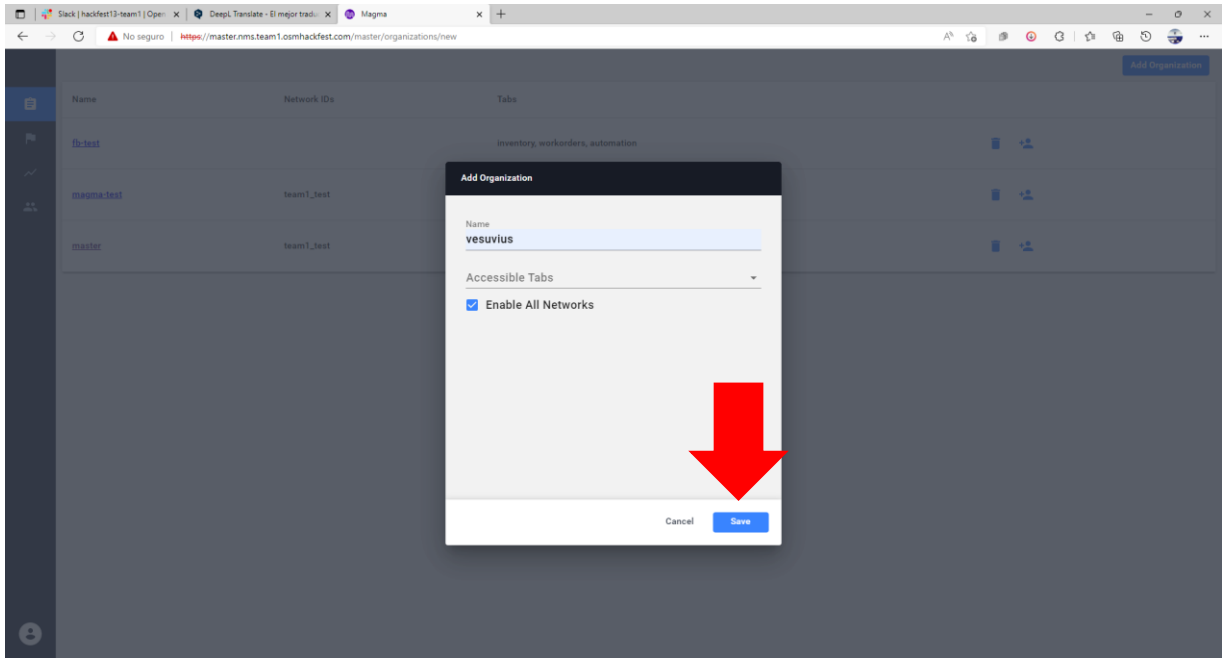


Configuration of Magma Orchestrator (1/7)

- Create new tenant in the NMS

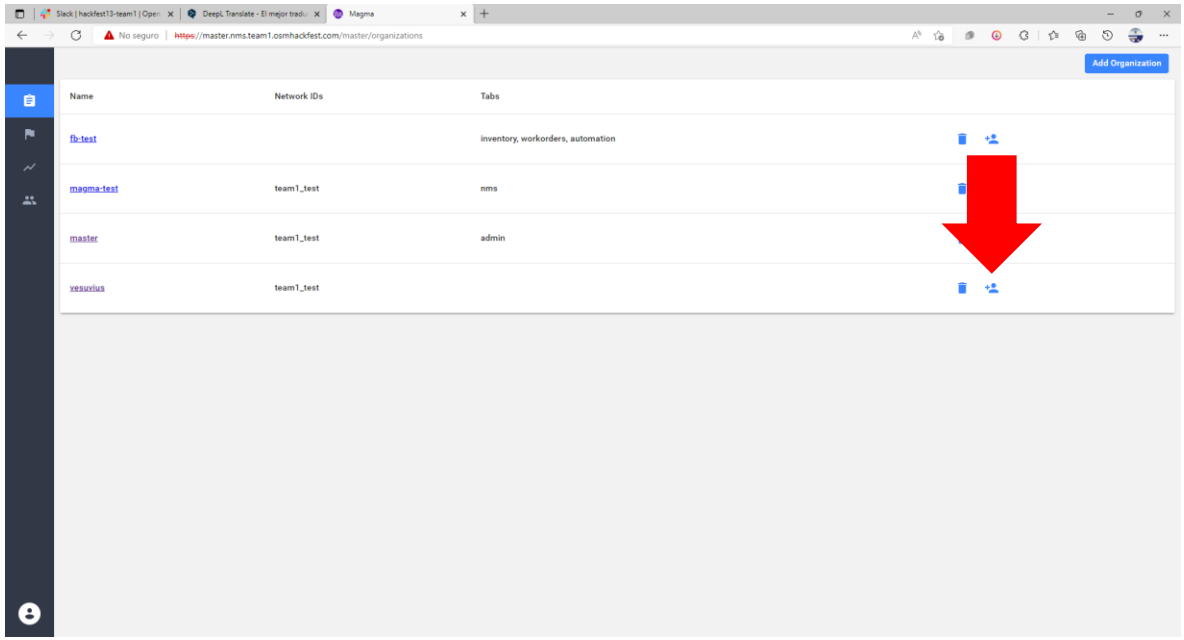


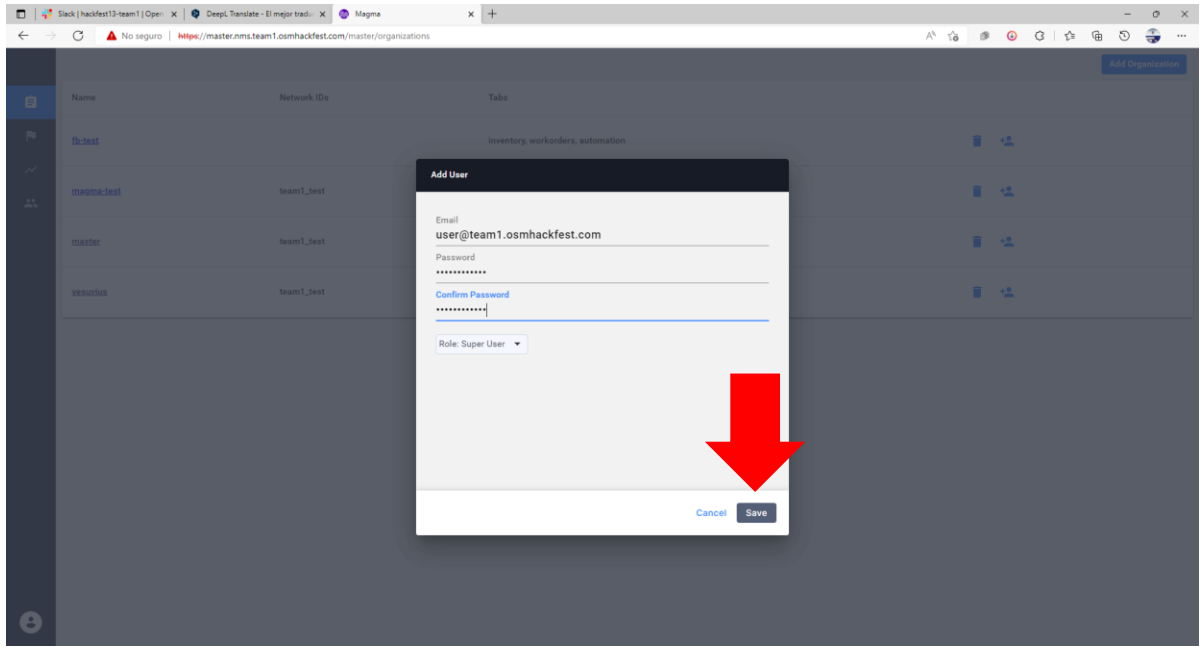




Configuration of Magma Orchestrator (2/7)

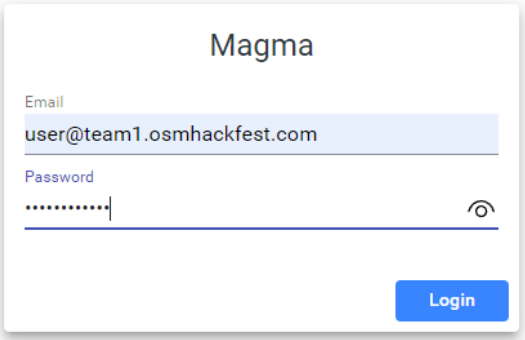
- Create NMS user for tenant:
  User: user@team1.osmhackfest.com
  Password: password1234
  Role: superuser

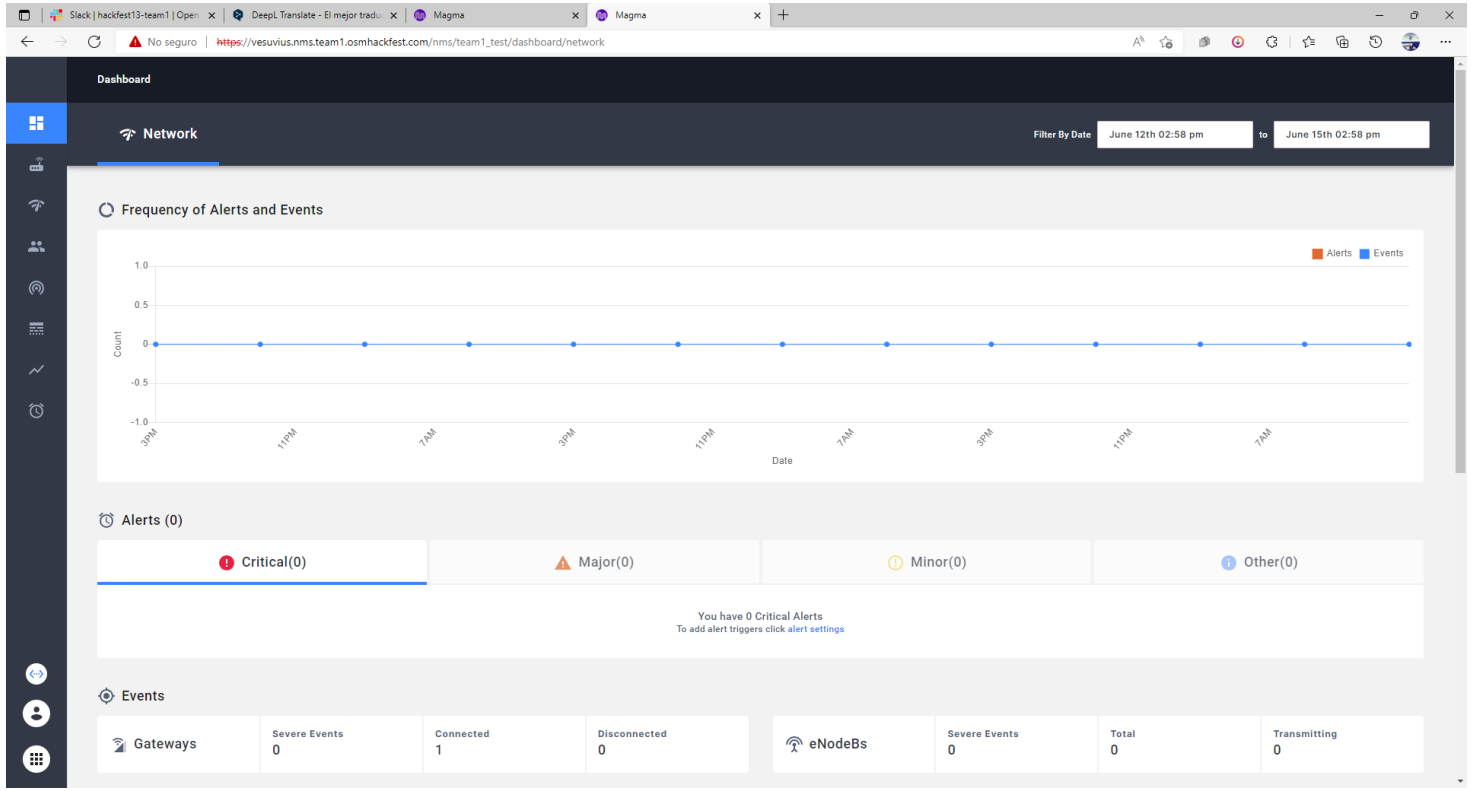




Configuration of Magma Orchestrator (3/7)

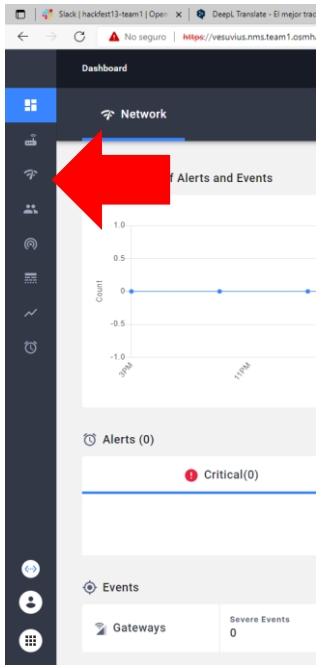
- Log in to the NMS https://vesuvius.nms.team1.osmhackfest.com

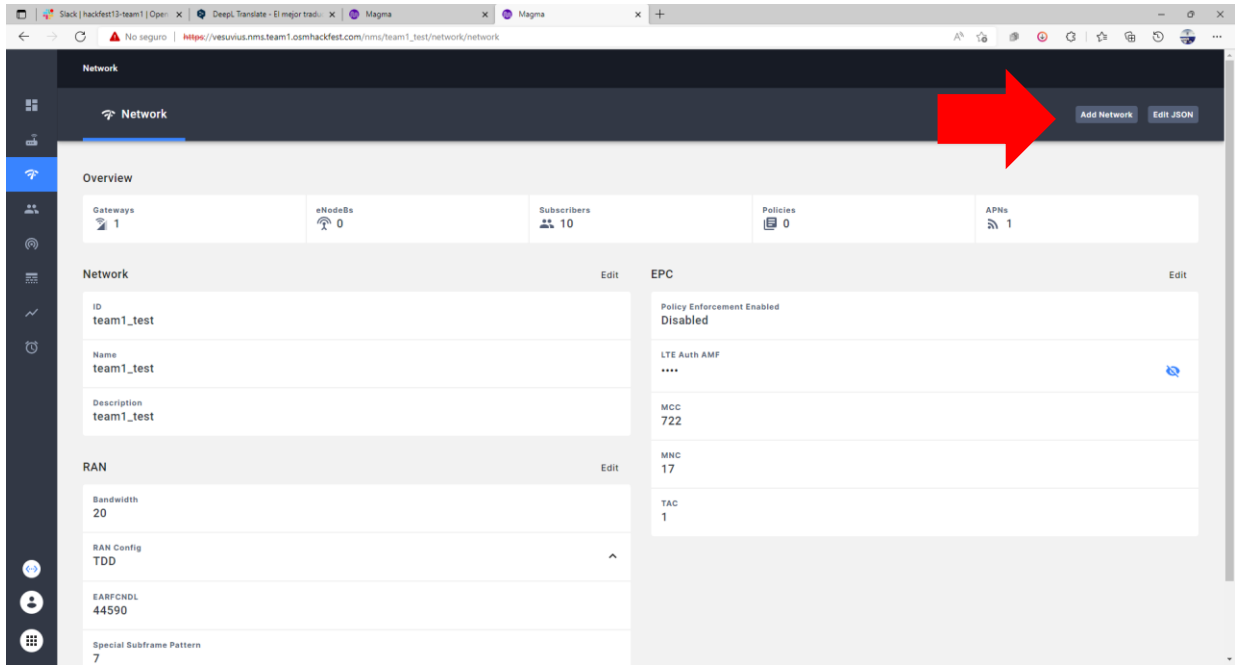


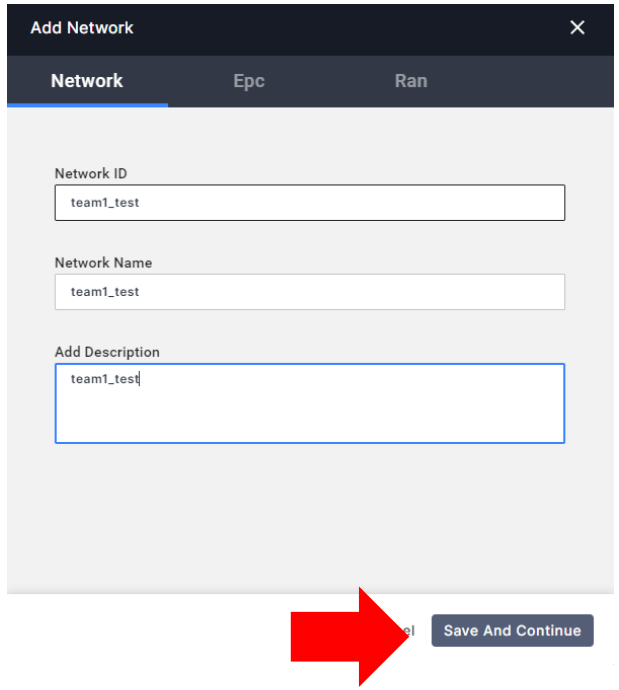


Configuration of Magma Orchestrator (4/7)

- Create an LTE network in tenant's NMS

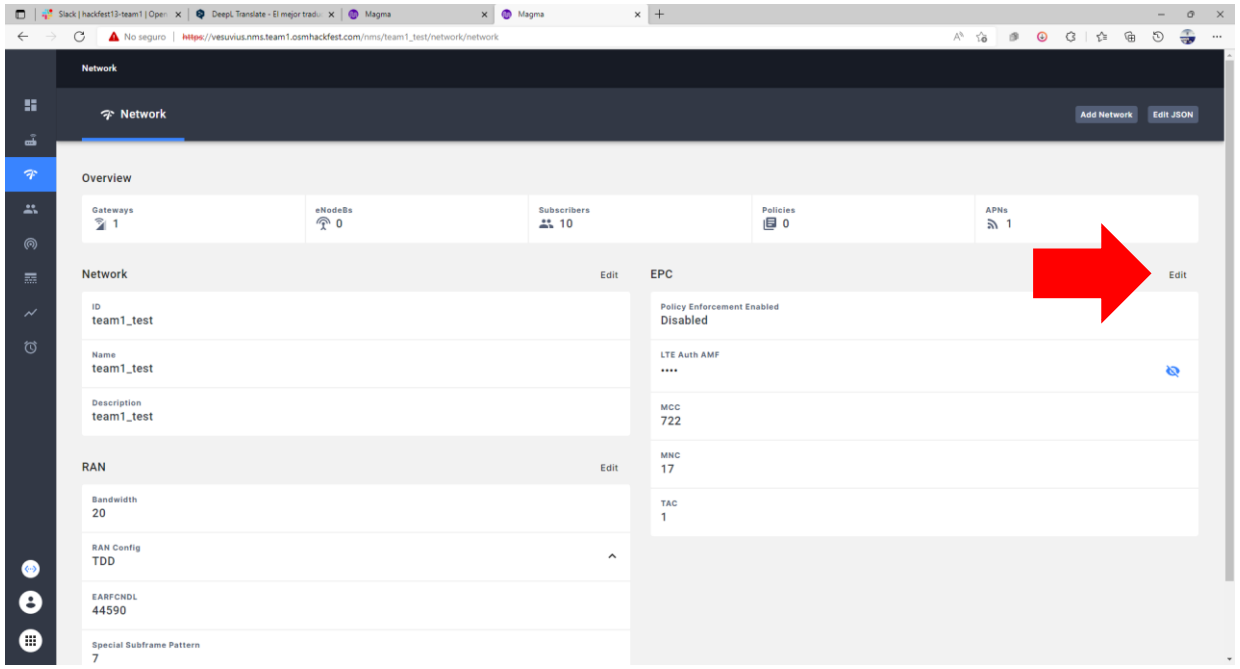


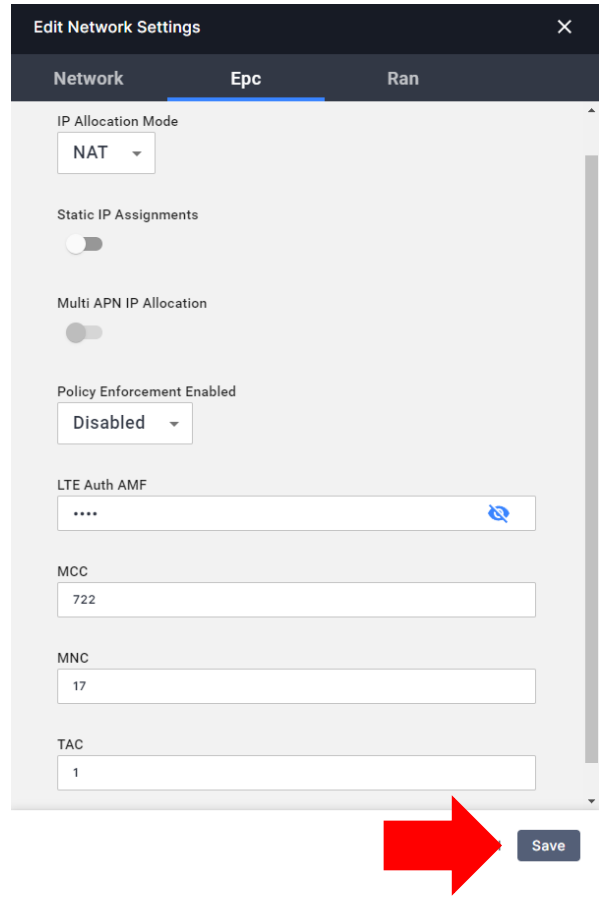




Configuration of Magma Orchestrator (5/7)

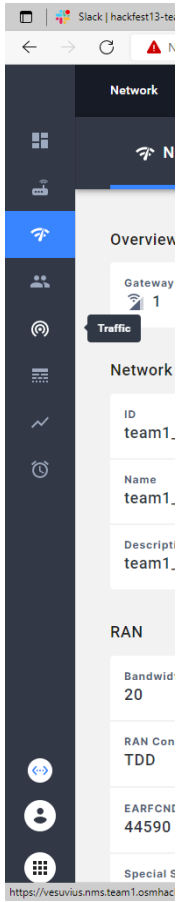
- Configure PLMN (MCC: 722, MNC: 17)

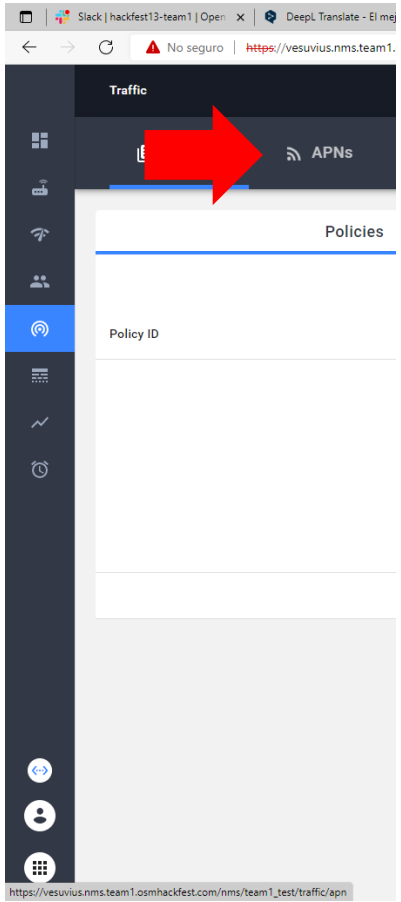


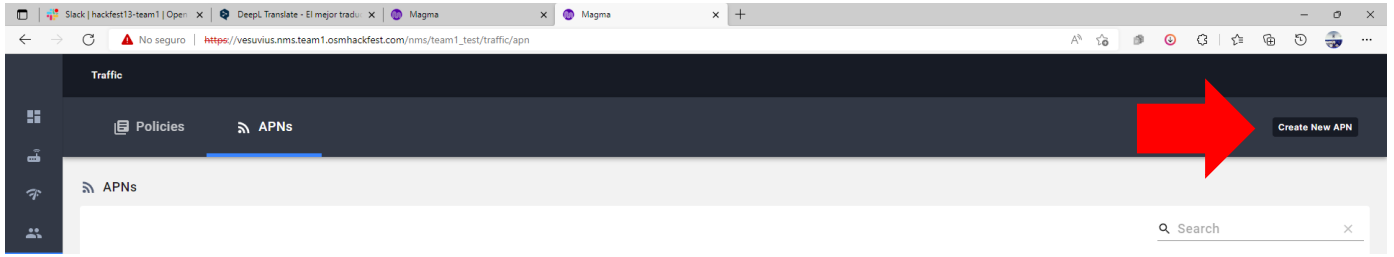


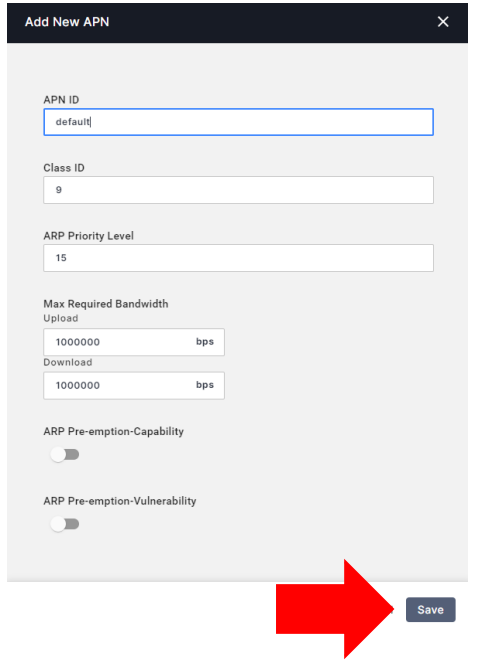
Configuration of Magma Orchestrator (6/7)

- Create APN (name: default)



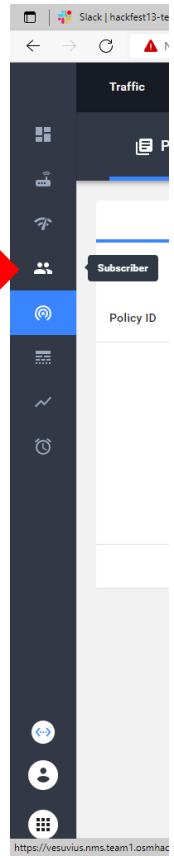


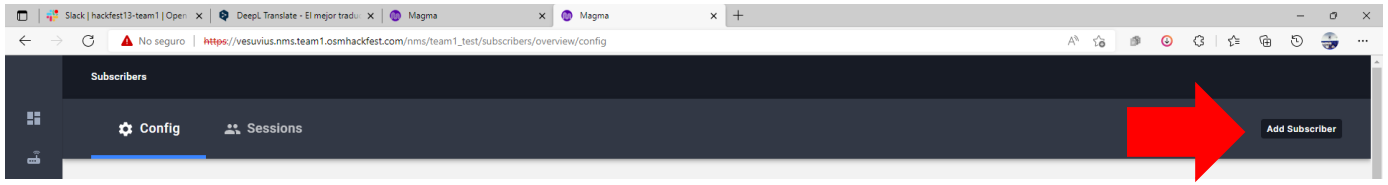


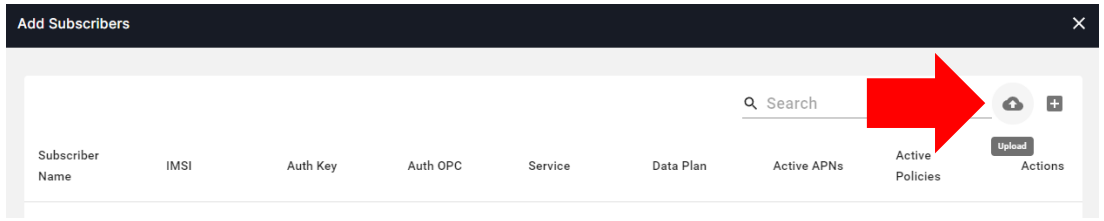


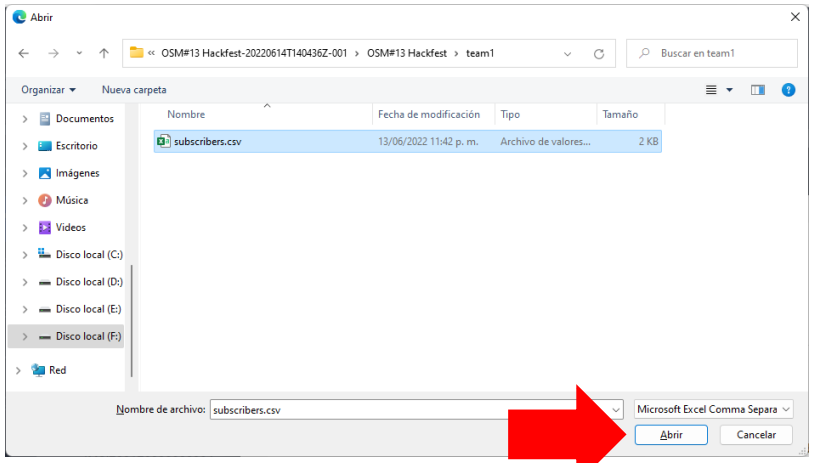
Configuration of Magma Orchestrator (7/7)

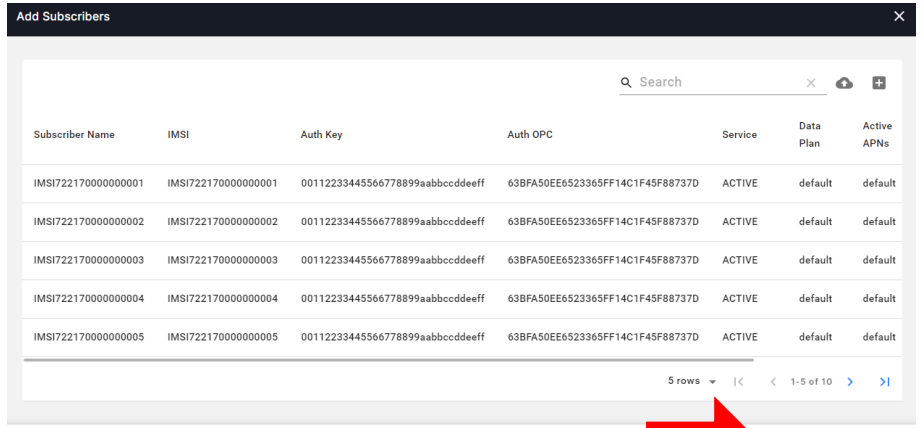
- Import subscribers to the NMS (csv file available on each team's Google Drive)

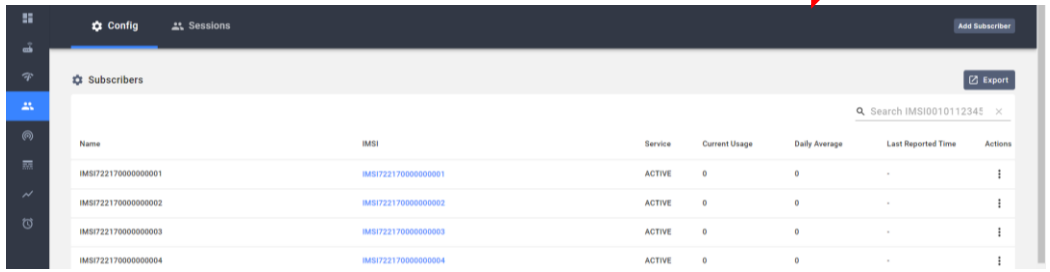












Open Source
MANO

Deploy Magma Access GW

ETSI

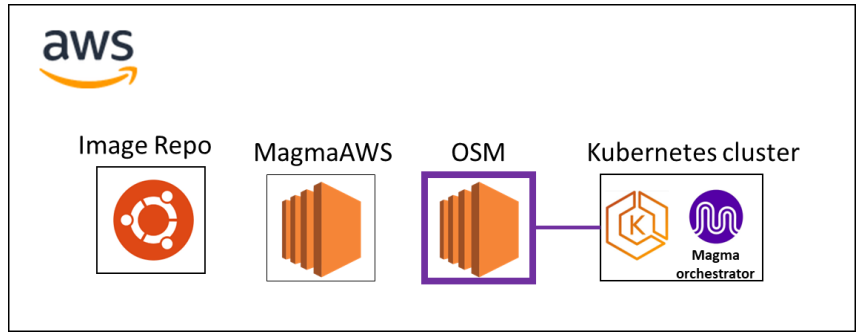
Magma Access GW – PDU

- Before we onboard the PNF into OSM, we need to tell OSM about the PDU. We need the VIM-ID:

```
ubuntu@ip-192-168-32-40:~$ osm vim-list | grep aws | awk '{print $4}'
afd90edc-bb75-4b98-8f39-9ab2a136a624
ubuntu@ip-192-168-32-40:~$
```

- We create a pdu.yaml descriptor file in our working directory. The pdu.yaml file must have a specific structure:
  - vim_account: afd90edc-bb75-4b98-8f39-9ab2a136a624
  - Magma AGW SGi IP: 192.168.51.193
  - Magma AGW S1 IP: 192.168.34.52

```
ubuntu@ip-192-168-32-40:~$ cd packages/
ubuntu@ip-192-168-32-40:~/packages$ cat pdu.yaml
name:           MagmaAGW
description:    Magma Access GW
type:           gateway
vim_accounts:   [ afd90edc-bb75-4b98-8f39-9ab2a136a624 ]
shared:         false
interfaces:
 -  name:       eth0
    ip-address: 192.168.51.193
    mgmt:       true
 -  name:       eth1
    ip-address: 192.168.34.52
    mgmt:       false
ubuntu@ip-192-168-32-40:~/packages$
```




```
ubuntu@ip-192-168-32-40:~/packages$ osm pdu-create --descriptor_file \pdu.yaml
b564062d-6fac-451a-b0f2-29cb3d07c421
ubuntu@ip-192-168-32-40:~/packages$
```

PDU Instances

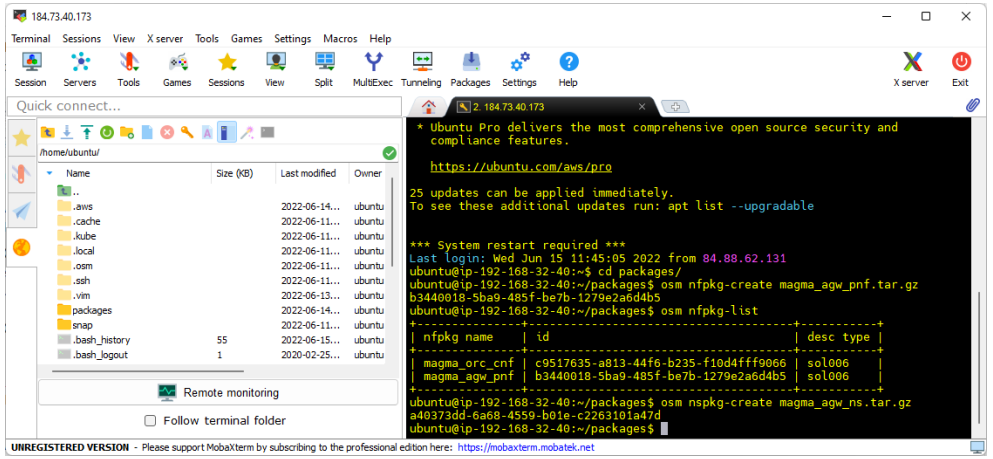
Identifier	Name	Type	UsageState	Created At
81afcc85-2e66-4fb7-97a4-608168b497d5	MagmaAGW	gateway	IN_USE	Jun-15-2022 17:29:48
b564062d-6fac-451a-b0f2-29cb3d07c421	MagmaAGW	gateway	NOT_IN_USE	Jun-15-2022 17:30:13

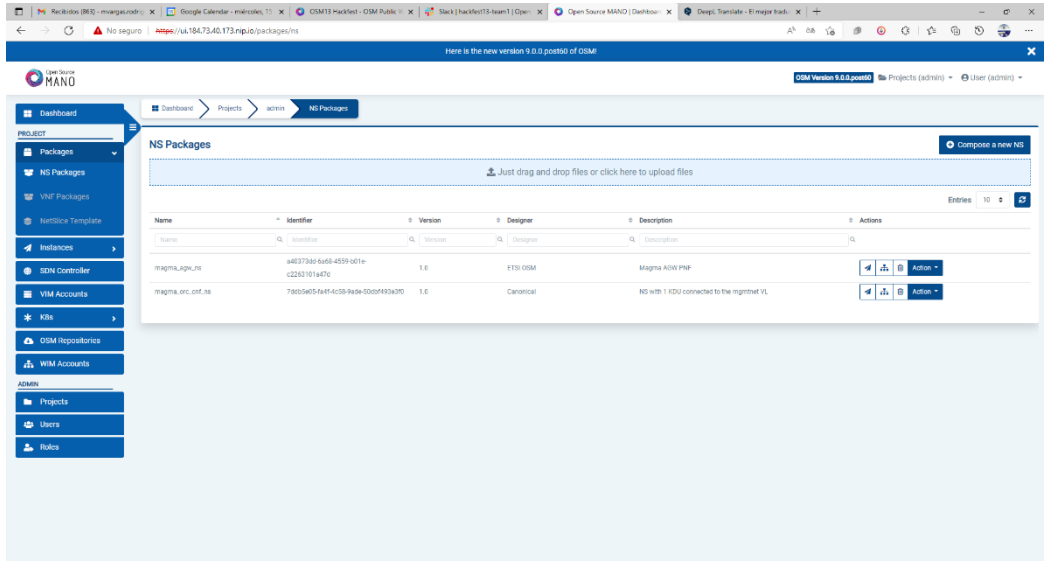
Onboarding Magma Access GW

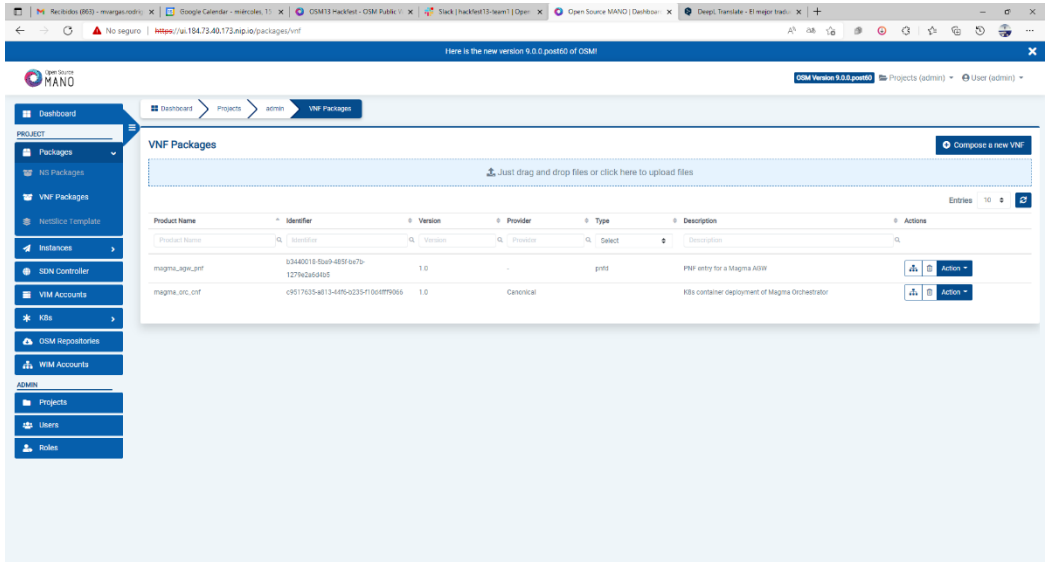
- Creating packages

  $ osm nfpkg-create magma_agw_pnf.tar.gz

  $ osm nspkg-create magma_agw_ns.tar.gz



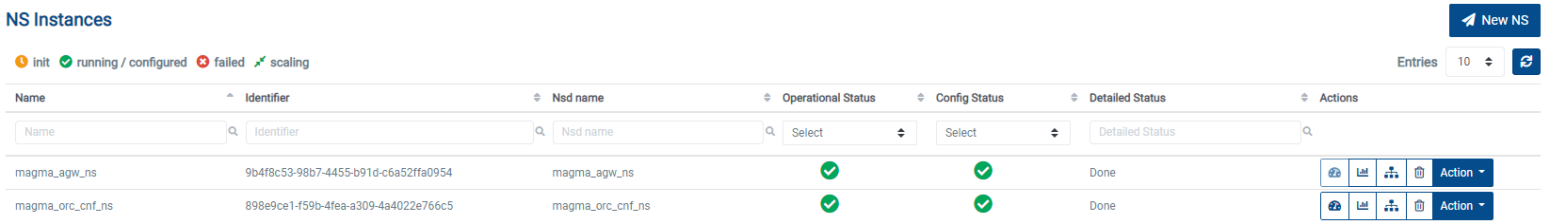


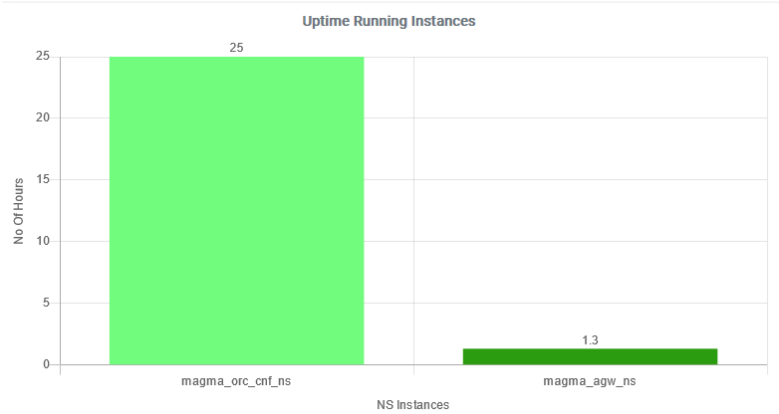


Instantiating Magma Access GW

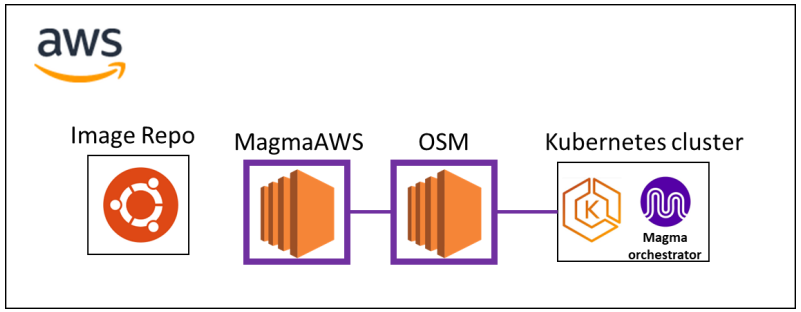
- Instantiating the Network Service

```
ubuntu@ip-192-168-32-40:~/packages$ osm ns-create --ns_name magma_agw_ns \
>                  --nsd_name magma_agw_ns \
>                  --vim_account aws-site

9b4f8c53-98b7-4455-b91d-c6a52ffa0954
ubuntu@ip-192-168-32-40:~/packages$
```





- We can now orchestrate the Magma AGW EC2 machine

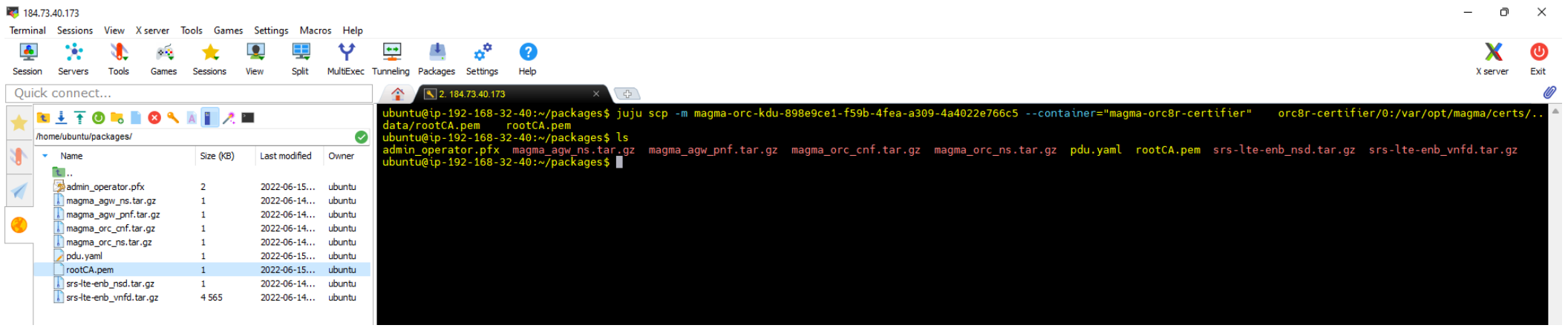


Download certificates – Part 2 (1/2)

- After instantiating the Magma AGW Network Service, from the **OSM machine**, we have to download a key with the following command:
  - juju scp -m $model-name --container="magma-orc8r-certifier"

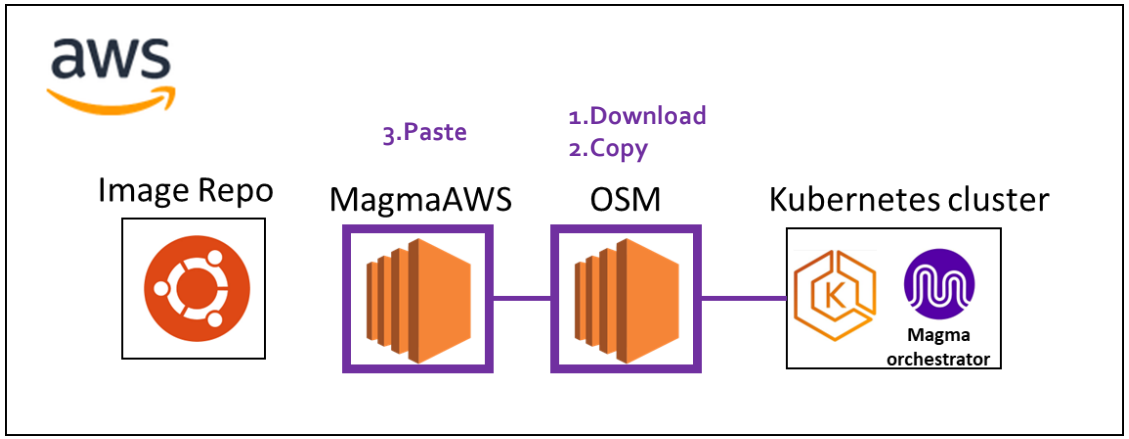
    orc8r-certifier/0:/var/opt/magma/certs/..data/rootCA.pem \

    rootCA.pem



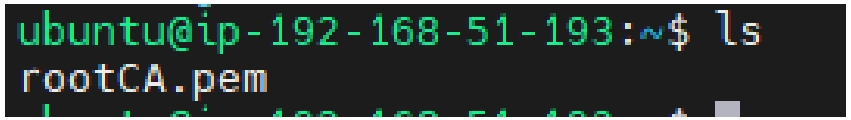
Download certificates – Part 2 (2/2)

- Then, we have to copy & paste it into the Magma AGW EC2 machine.



Access Magma AGW EC2

- ssh -i team1-key.pem ubuntu@54.161.100.52
- Paste the rootCA.pem file



- Then, run the command:
  - *magma-access-gateway.configure --domain team1.osmhackfest.com --root-ca-pem-path ./rootCA.pem*
- Magma AGW configuration starts
  - Copying files
  - Configures Control Proxy
  - Restart services
- When the configuration is done, we must add the Access Gateway to Magma Orchestrator

Magma AGW configuration

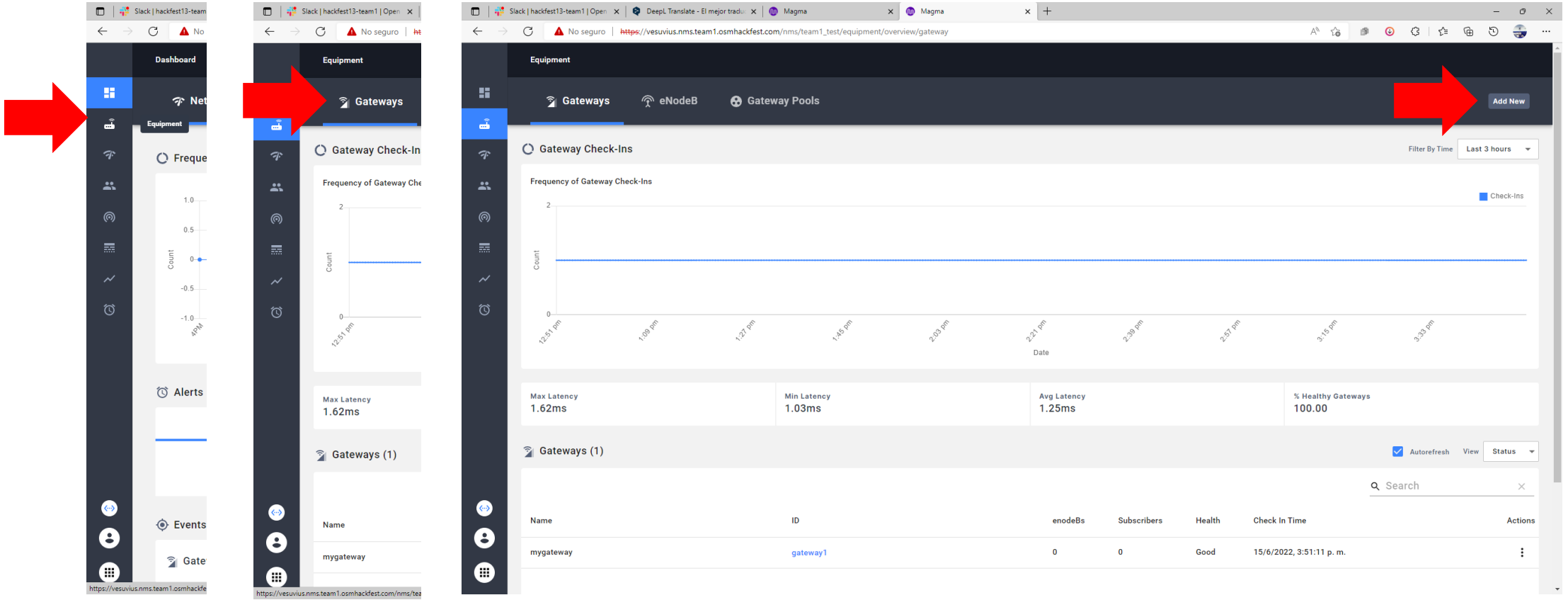
```
root@ip-192-168-51-193:/home/ubuntu# magma-access-gateway.configure \
>         --domain team1.osmhackfest.com \
>         --root-ca-pem-path ./rootCA.pem
Magma AGW Configurator: Starting Magma AGW configuration...
Magma AGW Configurator: Copying Root CA PEM from ./rootCA.pem to /var/opt/magma/tmp/certs...
Magma AGW Configurator: Configuring Control Proxy...
Magma AGW Configurator: Restarting Magma AGW services...
Magma AGW Configurator: Stopping magma@* service...
Magma AGW Configurator: Restarting magma@magmad service...
Magma AGW Configurator: Magma Access Gateway configuration done!
Magma AGW Configurator: To add this Access Gateway to an Orchestrator please use hardware secrets printed below:
Magma AGW Configurator:
Magma AGW Configurator: Hardware ID
Magma AGW Configurator: -----------
Magma AGW Configurator: 7bee1da2-2152-4fec-8c6b-fc69a1f2f07c
Magma AGW Configurator:
Magma AGW Configurator: Challenge key
Magma AGW Configurator: -------------
Magma AGW Configurator: MHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEZn6R6Hmai2vbtf/FzMybVJOm7TMt/5oReJWiOhR/AR42gLEJkuxT2QkEOi3gEROchGw4lqkWIpuyIV1Ebw4iy///hOn+Z98n3ynkvrrJ63B7ENklxCqMjG/76zXWA8uT
Magma AGW Configurator:
Magma AGW Configurator: Build info
Magma AGW Configurator: -------------
Magma AGW Configurator:  Commit Branch: unknown
Magma AGW Configurator:  Commit Tag: unknown
Magma AGW Configurator:  Commit Hash: unknown
Magma AGW Configurator:  Commit Date: unknown
Magma AGW Configurator:
Magma AGW Configurator: Notes
Magma AGW Configurator: -----
Magma AGW Configurator: - Hardware ID is this gateway's unique identifier
Magma AGW Configurator: - Challenge key is this gateway's long-term keypair used for
Magma AGW Configurator:   bootstrapping a secure connection to the cloud
Magma AGW Configurator: - Build info shows git commit information of this build
Magma AGW Configurator:
Magma AGW Configurator:
Magma AGW Configurator: Once Access Gateway is integrated with the Orchestrator, run magma-access-gateway.post-install to verify the installation.
```

STEP1

STEP2

STEP1 - Integrate Access gateway with magma Orchestrator (1/2)

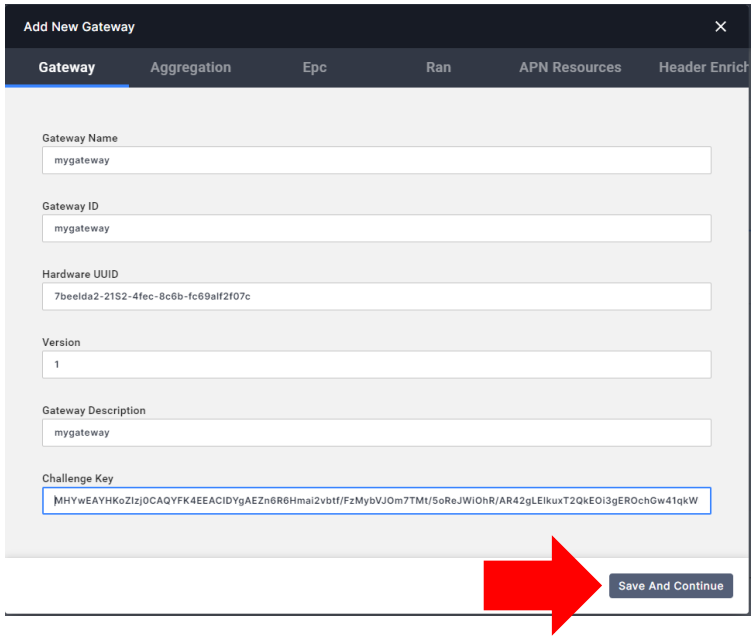
- Access the Magma Orchestrator through web GUI and select equipment, then select gateways



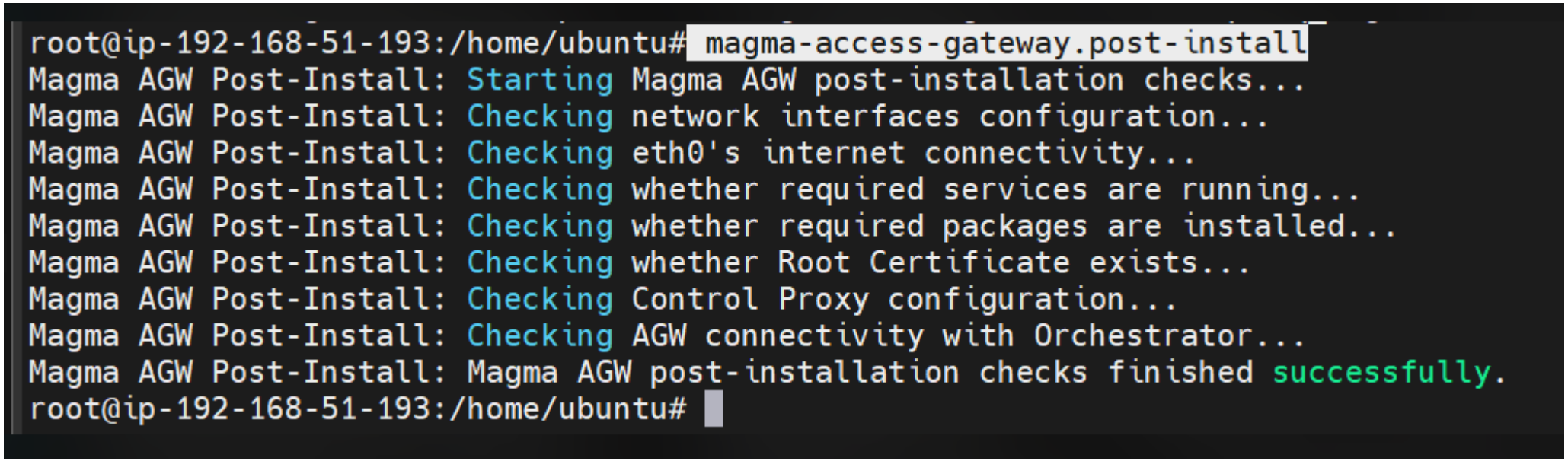
STEP1 - Integrate Access gateway with magma Orchestrator (2/2)

- Add the new gateway with the hardware ID and challenge key from the previous AGW configuration

```
Magma AGW Configurator: Hardware ID
Magma AGW Configurator: -----------
Magma AGW Configurator: 7bee1da2-2152-4fec-8c6b-fc69a1f2f07c
Magma AGW Configurator:
Magma AGW Configurator: Challenge key
Magma AGW Configurator: -------------
Magma AGW Configurator: MHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEZn6R6Hmai2vbtf/FzMybVJOm7TMt/5oReJWiOhR/AR42gLEJkuxT2QkEOi3gEROchGw4lqkWIpuyIV1Ebw4iy///hOn+Z98n3ynkvrrJ63B7ENklxCqMjG/76zXWA8uT
Magma AGW Configurator:
```

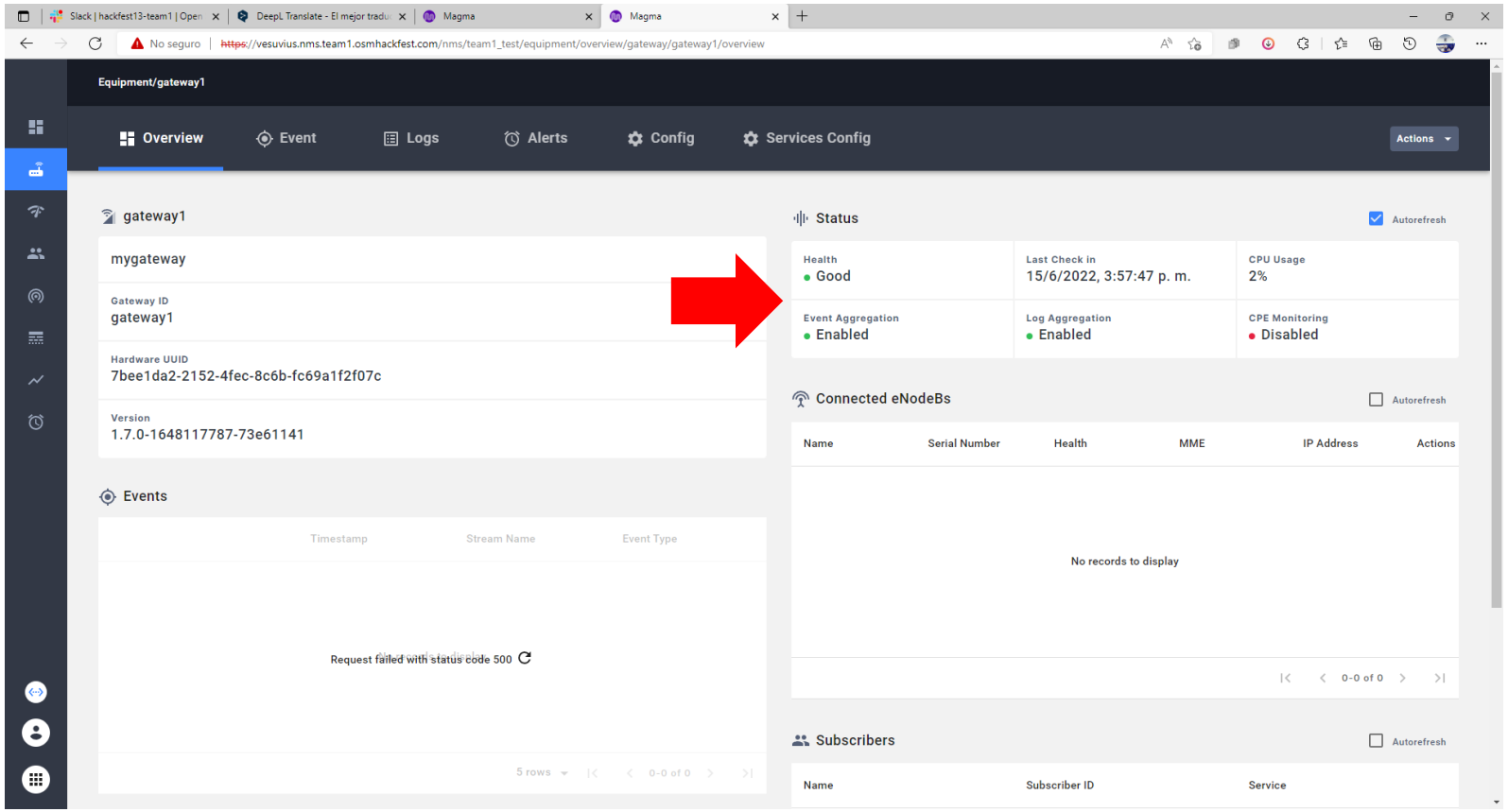


STEP2 - Install magma-access-gateway.post-install



STEP3 –Confirm AG with Magma Orchestrator (Optional)

- Check the AGW status by clicking on the gateway ID



Open Source
MANO

Deploy SRS LTE

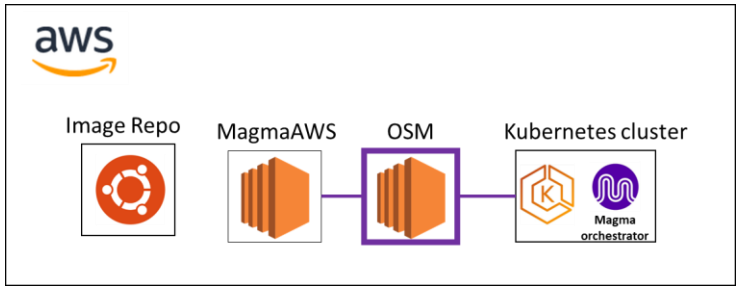
ETSI

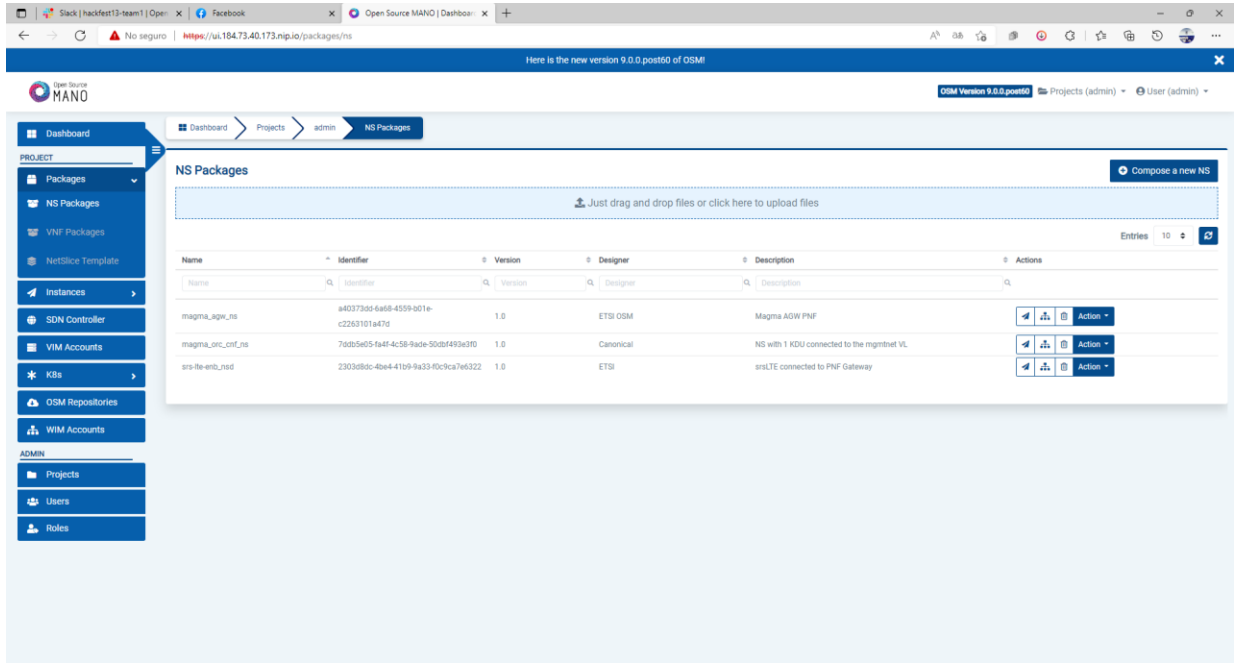
Onboarding srsLTE to OSM

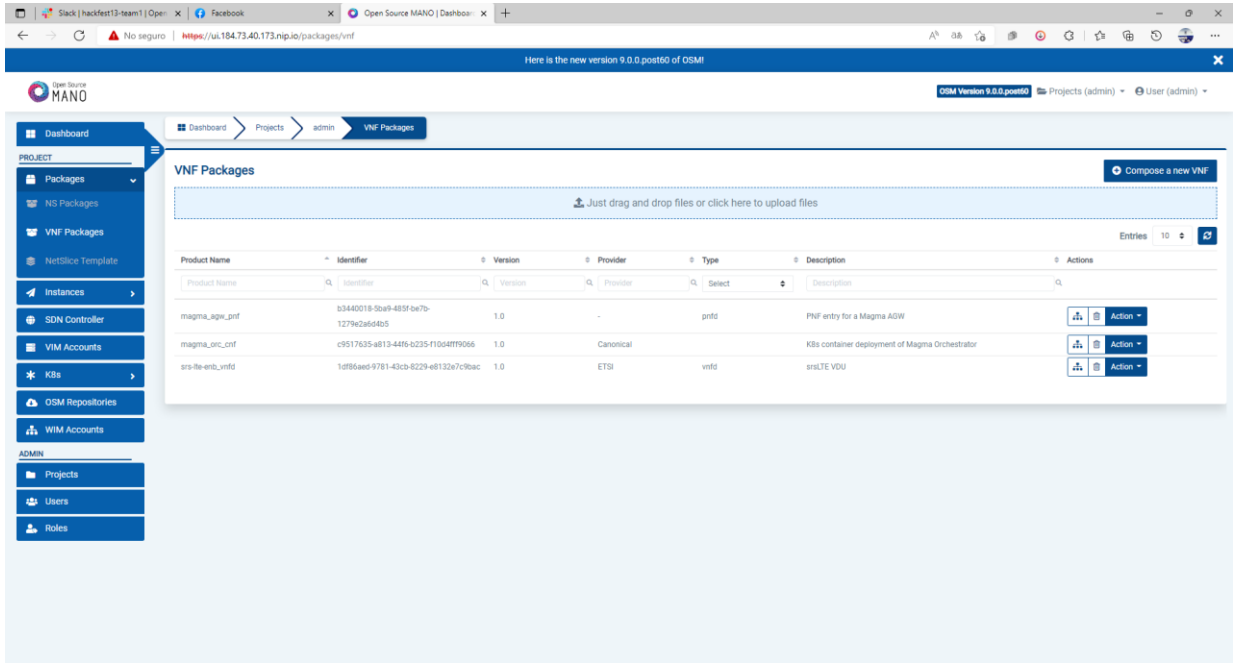
- We start creating the packages

  $ osm nfpkg-create srs-lte-enb_vnfd.tar.gz

  $ osm nspkg-create srs-lte-enb_nsd.tar.gz





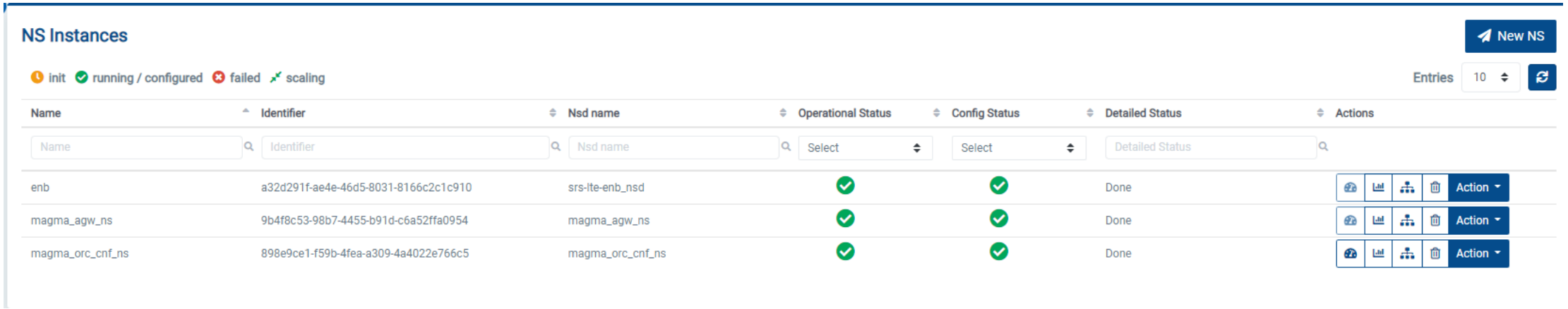


Deploy (Instantiate) the eNB

- Execute the command (Takes some time to deploy the enb instance)

```
$ osm ns-create --ns_name enb --nsd_name srs-lte-enb_nsd --vim_account aws-site \
  --config "{vld: [
             {name: mgmt, vim-network-name: subnet-098c0d6260c1f0a95 }
            ],
            additionalParamsForVnf: [
             {member-vnf-index: 'srsLTE',
              additionalParams: {
                bind_address_subnet: '192.168.32.0/19',
                mme_addr: '192.168.34.52',
                enb_mcc: '722',
                enb_mnc: '17'}}]}"
```

```
ubuntu@ip-192-168-32-40:~$ osm ns-create --ns_name enb --nsd_name srs-lte-enb_nsd --vim_account aws-site \
>   --config "{vld: [
>               {name: mgmt, vim-network-name: subnet-098c0d6260c1f0a95 }
>             ],
>             additionalParamsForVnf: [
>               {member-vnf-index: 'srsLTE',
>                additionalParams: {
>                  bind_address_subnet: '192.168.32.0/19',
>                  mme_addr: '192.168.34.52',
>                  enb_mcc: '722',
>                  enb_mnc: '17'}}]}"
41aa3120-d1b3-4bb6-ba4d-6ac3797d4a68
ubuntu@ip-192-168-32-40:~$
```



Attach UE to Magma (from CLI)

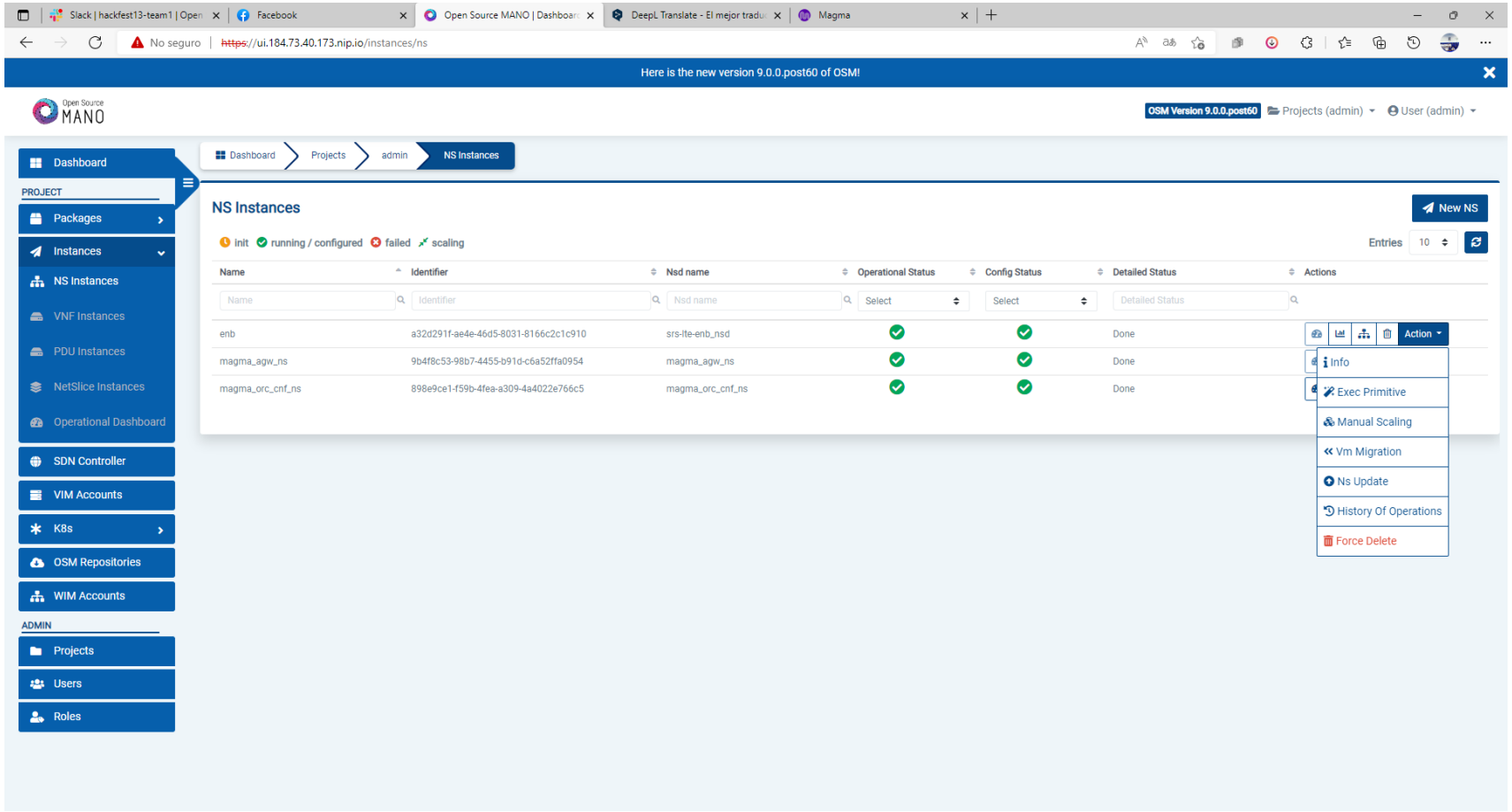
- The attachment of users is a Day-2 OSM action
- Execute the command

```
$ osm ns-action enb --vnf_name "srsLTE" --vdu_id srsLTE-vdu  --action_name attach-ue --params '{
          usim-imsi: "722170000000008",
          usim-k: "00112233445566778899aabbccddeeff",
          usim-opc: "63bfa50ee6523365ff14c1f45f88737d"
        }'

Output:bbb7becc-4a2b-456e-b733-940e829da171
```

- We can also attach users through the OSM dashboard (UI)

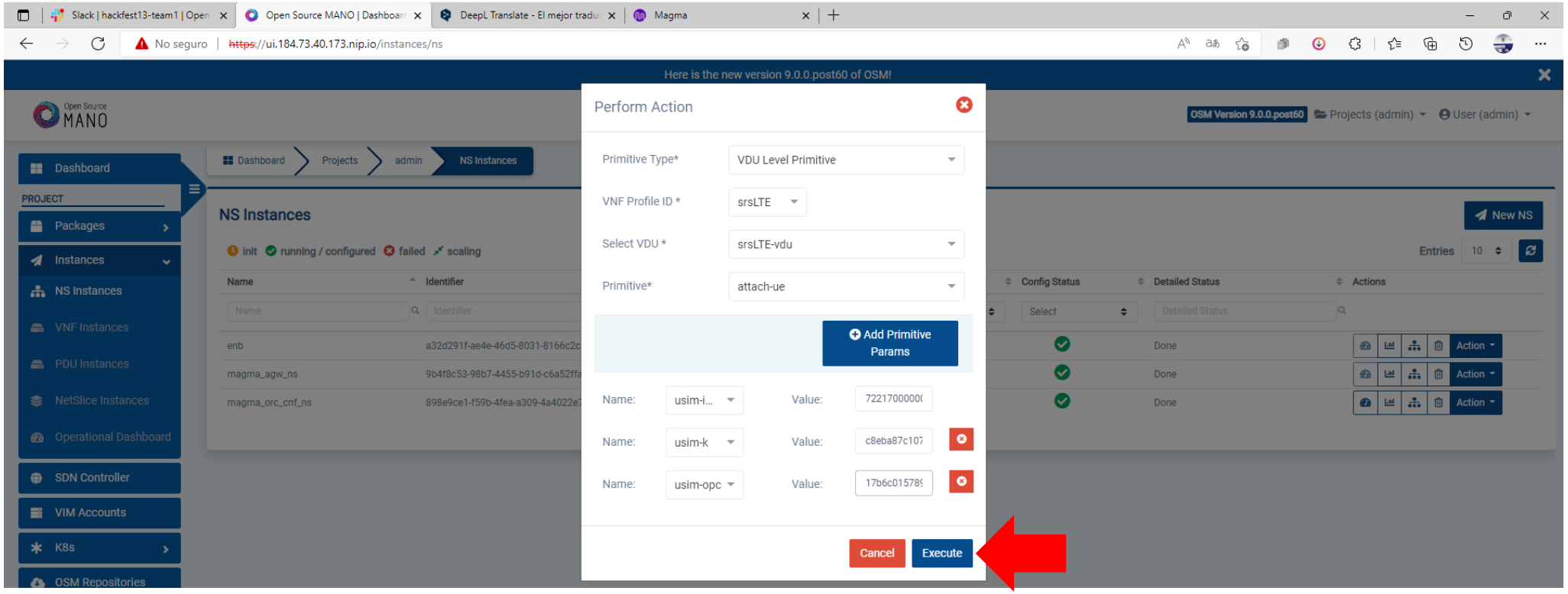
Attach UE to Magma (from UI) (1/4)

- Select “Exec primitive” on the eNB instance on OSM web GUI



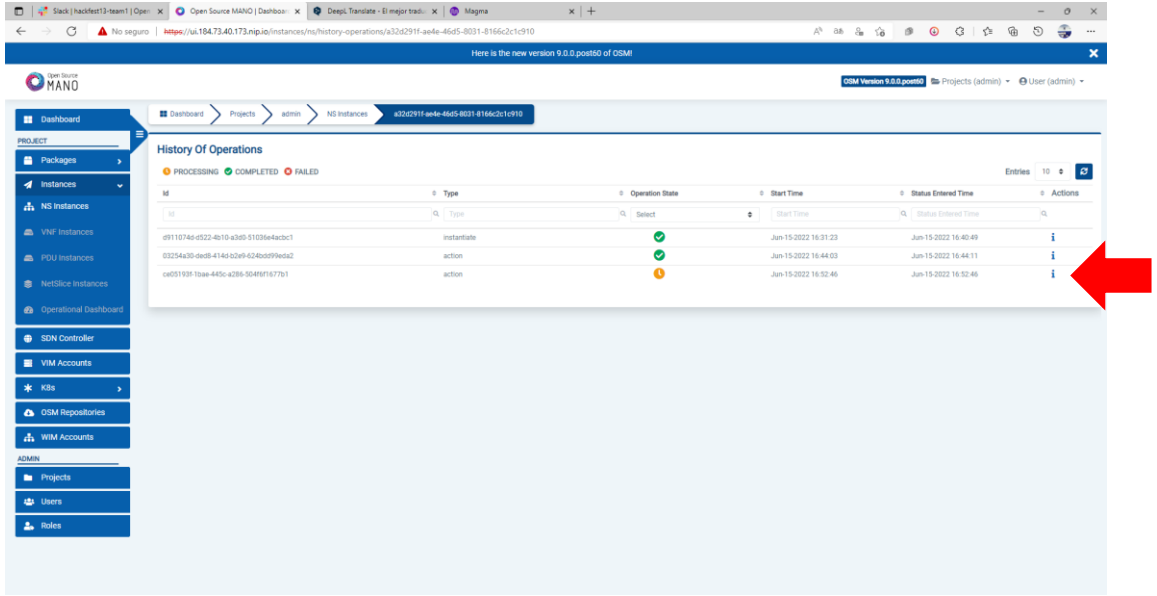
Attach UE to Magma (from UI) (2/4)

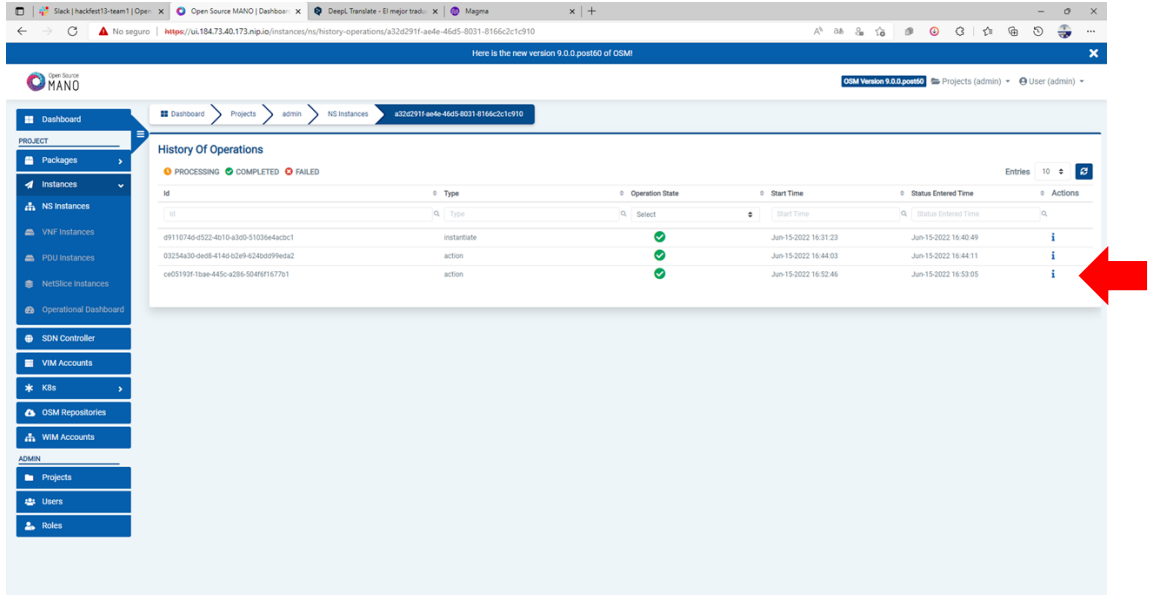
- Select “VDU Level”, VNF Profile ID, VDU primitive and Add Primitive params with the previous UE values, the execute:



Attach UE to Magma (from UI) (3/4)

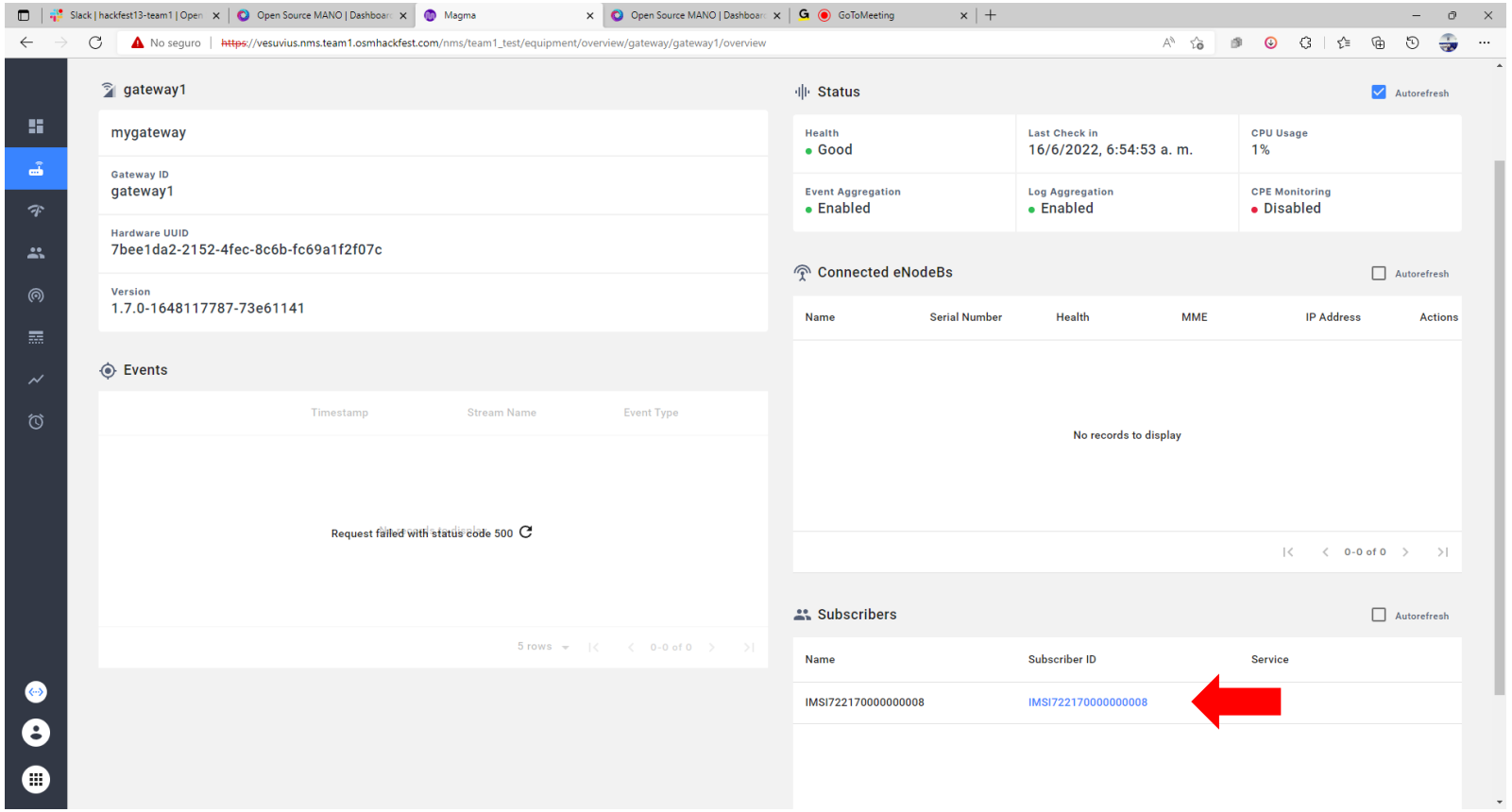
- Check the History on the enb NS instance:





Attach UE to Magma (from UI) (4/4)

- Check the UE connected on the Magma Orchestrator:



One step forward (1/4)

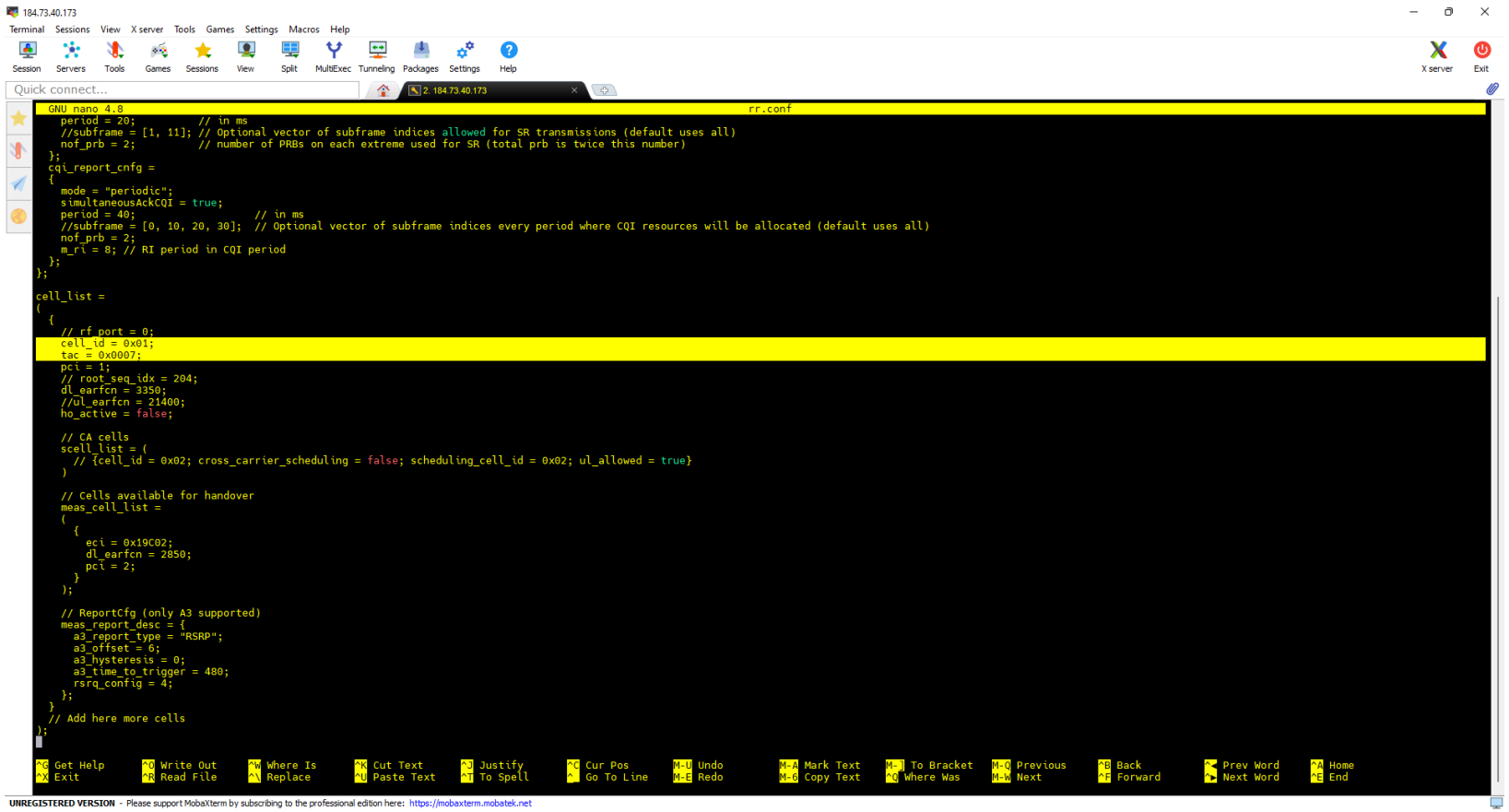
- **Access one active UE**
  - Description: Access a UE and generate traffic to a web site or simply ping it.
    - osm vnf-list
    - ssh 54.80.0.209
    - User: Ubuntu
    - Pass: osm2022
    - Check tun_srsue

```
ubuntu@ip-192-168-32-40:~$ osm vnf-list
+--------------------------------------+------+--------------------------------------+-----------------+-----------------+--------------------------------------+----------------+
| vnf id                               | name | ns id                                | vnf member index | vnfd name       | vim account id                       | ip address     |
+--------------------------------------+------+--------------------------------------+-----------------+-----------------+--------------------------------------+----------------+
| e9ad4766-edc4-4f80-bf9c-357f3e58c77c | -    | 898e9ce1-f59b-4fea-a309-4a4022e766c5 | magma_orc_cnf    | magma_orc_cnf   | afd90edc-bb75-4b98-8f39-9ab2a136a624 | None           |
| 3c1c9654-10f1-4a08-b445-13d74f798106 | -    | 9b4f8c53-98b7-4455-b91d-c6a52ffa0954 | magma-agw-pnf    | magma_agw_pnf   | afd90edc-bb75-4b98-8f39-9ab2a136a624 | 192.168.51.193 |
| 3e50d5bf-01d7-4bfd-8323-8d059de2691c | -    | 41aa3120-d1b3-4bb6-ba4d-6ac3797d4a68 | srsLTE           | srs-lte-enb_vnfd | afd90edc-bb75-4b98-8f39-9ab2a136a624 | 54.80.0.209    |
+--------------------------------------+------+--------------------------------------+-----------------+-----------------+--------------------------------------+----------------+
```

```
ubuntu@ip-192-168-37-34:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 192.168.37.34  netmask 255.255.224.0  broadcast 192.168.63.255
        inet6 fe80::c72:2dff:fe8f:eea9  prefixlen 64  scopeid 0x20<link>
        ether 0e:72:2d:8f:ee:a9  txqueuelen 1000  (Ethernet)
        RX packets 127182  bytes 169027247 (169.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20358  bytes 2240396 (2.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 192.168.181.152  netmask 255.255.255.0  broadcast 192.168.181.255
        inet6 fe80::ce3:99ff:fed1:a0f7  prefixlen 64  scopeid 0x20<link>
        ether 0e:e3:99:d1:a0:f7  txqueuelen 1000  (Ethernet)
        RX packets 137  bytes 5864 (5.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 143  bytes 8062 (8.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 59414169  bytes 1766181415291 (1.7 TB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 59414169  bytes 1766181415291 (1.7 TB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun_srsue: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 192.168.128.26  netmask 255.255.255.0  destination 192.168.128.26
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
```

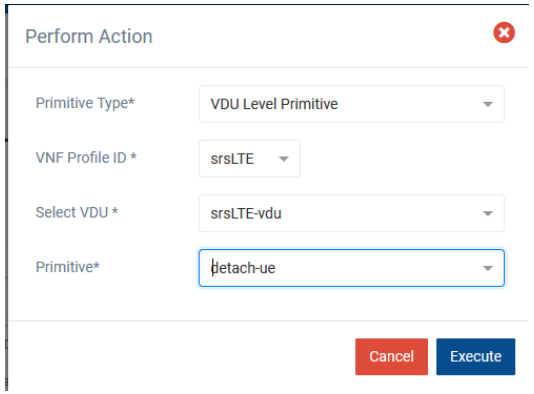
One step forward (2/4)

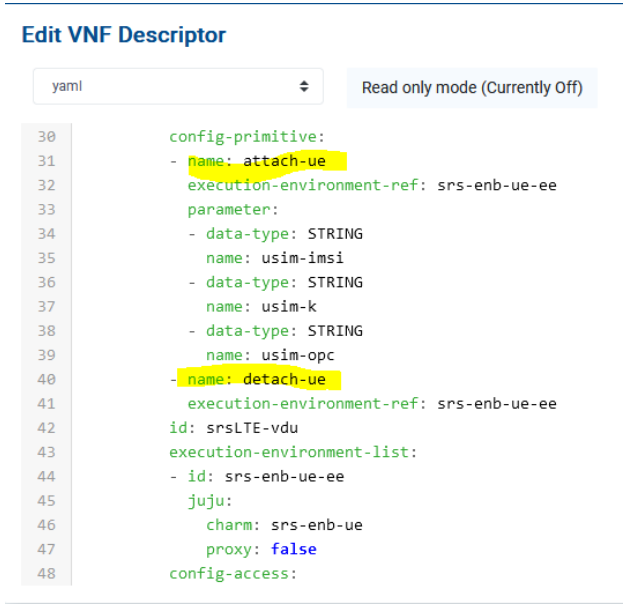
- **Access one active UE**
  - Check the rr.conf file to get the Cell ID and TAC to configure on Magma-Orc

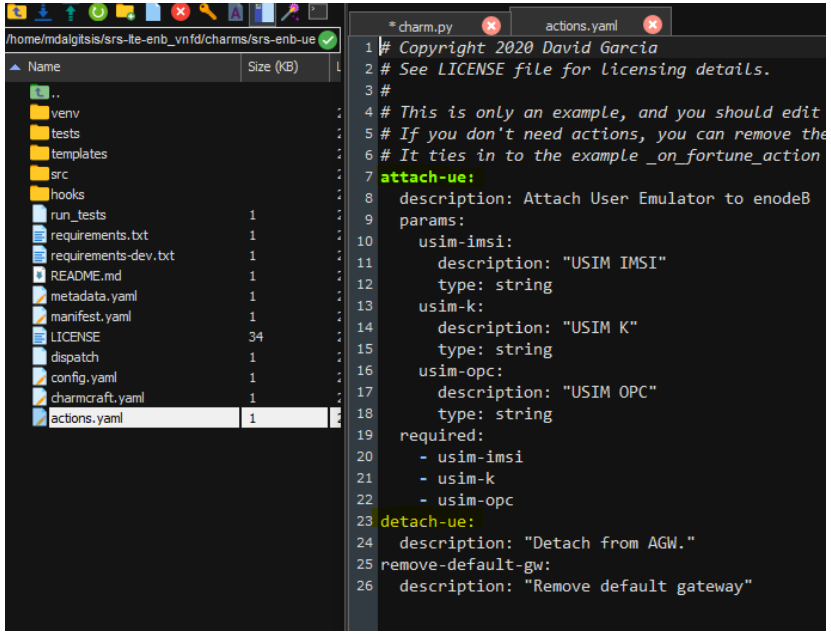


One step forward (3/4)

- **Detach specific UE**
  - Description: The current deployment with the detach action, removes all the active users. The detach_specific_ue action could detach a specific UE based on its IMSI number.

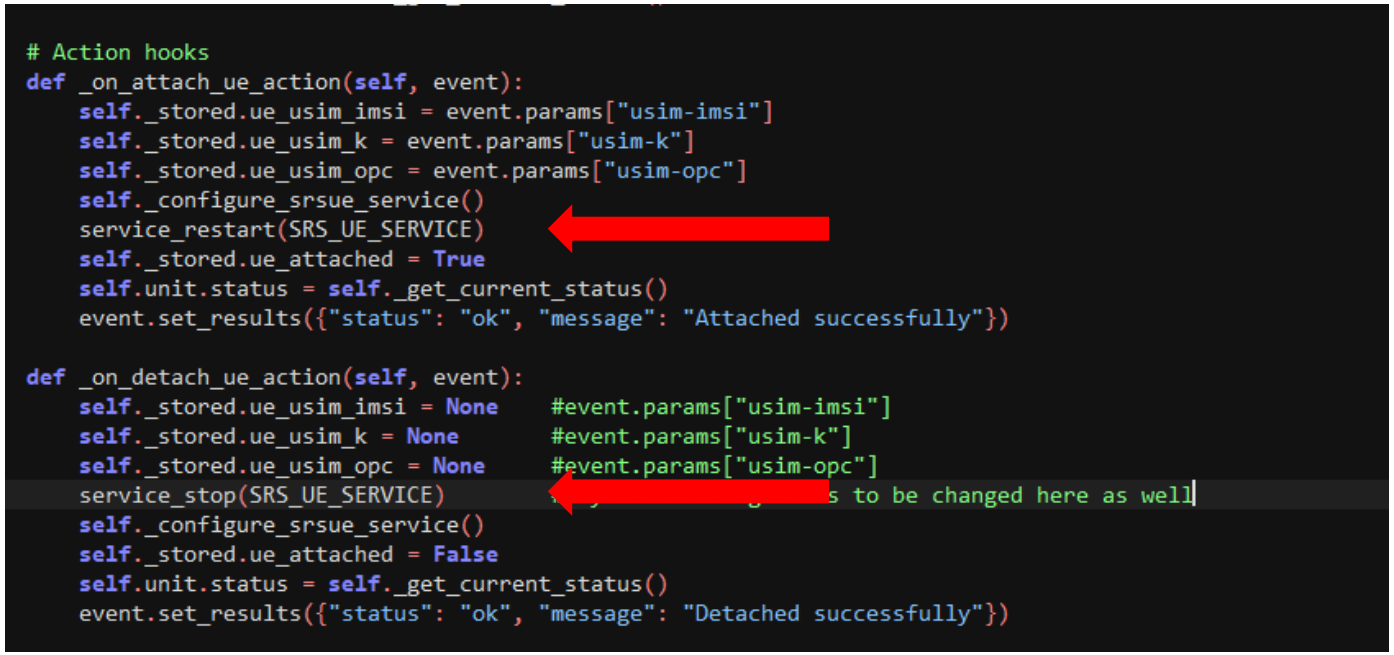






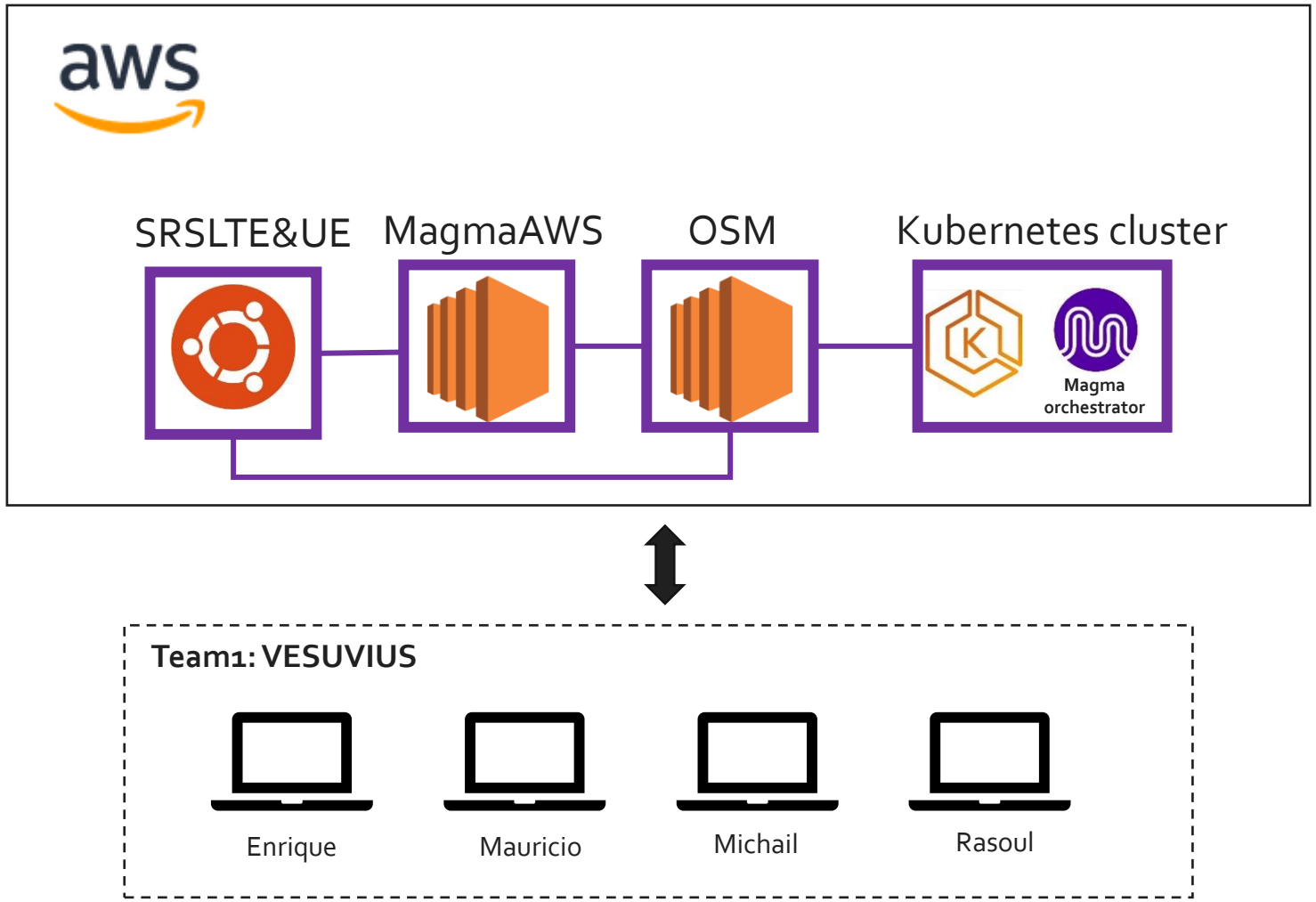
One step forward (4/4)

- **Detach specific UE**
  - Description: The current deployment with the detach action, removes all the active users. The detach_specific_ue action could detach a specific UE based on its IMSI number.

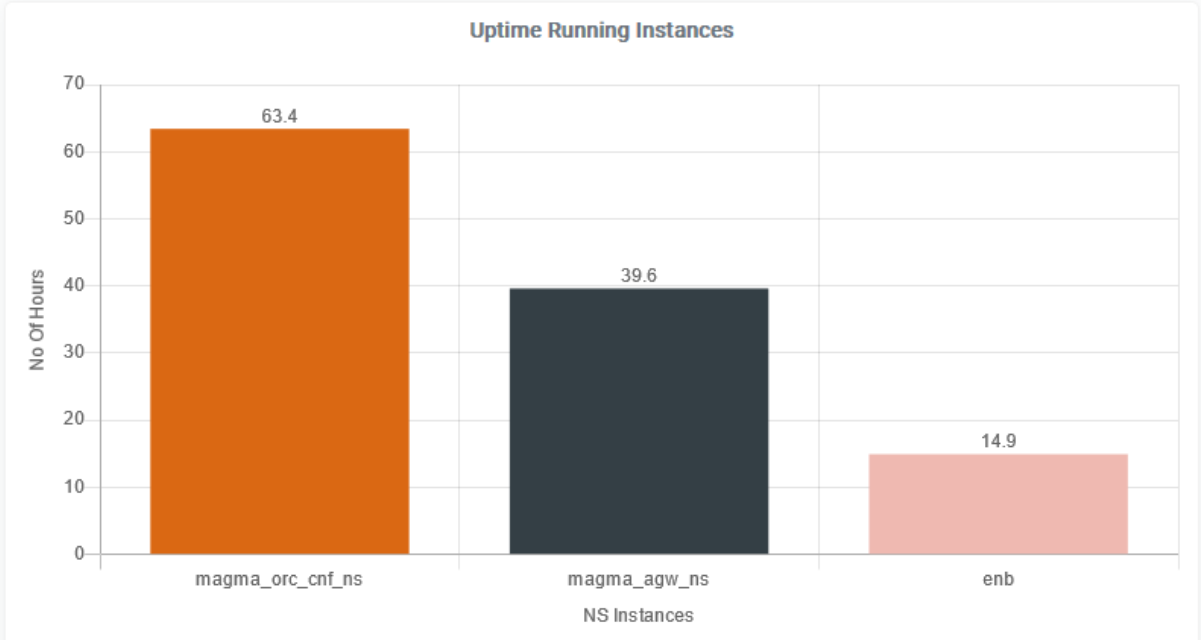


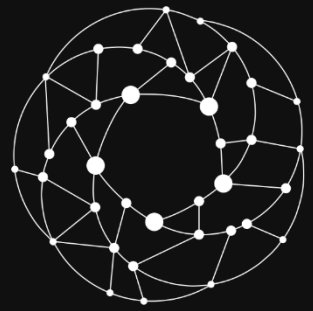
```
# Action hooks
def _on_attach_ue_action(self, event):
    self._stored.ue_usim_imsi = event.params["usim-imsi"]
    self._stored.ue_usim_k = event.params["usim-k"]
    self._stored.ue_usim_opc = event.params["usim-opc"]
    self._configure_srsue_service()
    service_restart(SRS_UE_SERVICE)
    self._stored.ue_attached = True
    self.unit.status = self._get_current_status()
    event.set_results({"status": "ok", "message": "Attached successfully"})

def _on_detach_ue_action(self, event):
    self._stored.ue_usim_imsi = None    #event.params["usim-imsi"]
    self._stored.ue_usim_k = None       #event.params["usim-k"]
    self._stored.ue_usim_opc = None     #event.params["usim-opc"]
    service_stop(SRS_UE_SERVICE)        s to be changed here as well
    self._configure_srsue_service()
    self._stored.ue_attached = False
    self.unit.status = self._get_current_status()
    event.set_results({"status": "ok", "message": "Detached successfully"})
```

Conclusions (1/2)



Conclusions (2/2)





Open Source
MANO

Thank you!

ETSI