LAB: Network Documentation with NetBox

Login credential is given in the LAB guide, please follow it to access the VM machine.

Note

- We will perform the LAB on the same machine as it is, where SNMP-LibreNMS lab is done.
- Server Hostname will have an extension (netbox) in this LAB, example – `groupX-server-netbox.apnictraining.net`
- '$' means a general user is having the CLI (command line interface)
- '#' means root user is having the CLI
- Do not attempt to perform the LAB with '#' root user unless it is explicitly mentioned.
- Read the instructions carefully before execute any commands.

Lab target

- Setup NetBox service
- Explore Data Center Managemnet - DCIM
- Explore IP address management - IPAM

Requirements

- HTTP server running Nginx or Apache. As like the previous LAB modules, here we will use Apache.
- Python version 3.6 or greater.
- Python extensions – setuptools, graphviz, libpq-dev, and xml2
- PostgreSQL database version 9.6 or greater. (According to the official document, till December 20, 2020, MySQL and other relational databases are not supported at NetBox.)
- Redis server version 4 or greater.

Setup NetBox

1. Update the System Repository

```
$ sudo apt update && sudo apt upgrade
```

2. Install Required Packages

First we have to install necessary dependencies and Python modules for the project:

```
$ sudo apt-get install -y git gcc redis python3-setuptools  graphviz python3 \
python3-pip python3-venv python3-dev build-essential \
libxml2-dev libxslt1-dev libffi-dev libpq-dev libssl-dev zlib1g-dev
```

3. Install and create Database

we are going to install PostgreSQL, and create the database. Here we will use, `netbox` as the username and `training` as the password.

```
$ sudo apt-get install -y postgresql libpq-dev
$ sudo -u postgres psql

psql (10.15 (Ubuntu 10.15-0ubuntu0.18.04.1))
Type "help" for help.

postgres=# CREATE DATABASE netbox;
CREATE DATABASE

postgres=# CREATE USER netbox WITH PASSWORD 'training';
CREATE ROLE

postgres=# GRANT ALL PRIVILEGES ON DATABASE netbox TO netbox;
GRANT

postgres=# \q
```

4. Install and test Redis

```
$ sudo apt install -y redis-server
$ sudo systemctl start redis-server
$ sudo systemctl enable redis-server
$ sudo systemctl status redis-server
```

Check the service response

```
$ redis-cli ping
PONG
```

5. Download NetBox

Clone the netbox repo at `/opt` directory.

```
$ cd /opt
$ sudo git clone -b master https://github.com/netbox-community/netbox.git

Cloning into 'netbox'...
remote: Enumerating objects: 136, done.
remote: Counting objects: 100% (136/136), done.
remote: Compressing objects: 100% (103/103), done.
remote: Total 58055 (delta 63), reused 65 (delta 33), pack-reused 57919
Receiving objects: 100% (58055/58055), 28.38 MiB | 573.00 KiB/s, done.
Resolving deltas: 100% (45302/45302), done.
```

Create NetBox system user.

```
$ sudo adduser --system --group netbox
$ sudo chown --recursive netbox /opt/netbox/netbox/media/
```

6. Configure and install NetBox service

```
$ cd /opt/netbox/netbox/netbox/
$ sudo cp configuration.example.py configuration.py
```

We need our focus into 4 part here.

- ALLOWED_HOSTS
  `we have to put the server hostname or IP; we will use hostname, as we are hosting multiple HTTP service in a single host.`
- DATABASE `fillup the database name, password`
- REDIS `default configuration is enough here for the LAB`
- SECRET_KEY

```
$ sudo vim configuration.py

ALLOWED_HOSTS = ['groupX-server-netbox.apnictraining.net']

DATABASE = {
'NAME': 'netbox',         # Database name
'USER': 'netbox',         # Username that creared
'PASSWORD': 'training',   # Password we assigned
'HOST': 'localhost',      # Database server
'PORT': '',               # Default is leaving it blank
'CONN_MAX_AGE': 300,      # Max database connection age
}

REDIS = {
'tasks': {
'HOST': 'localhost',
'PORT': 6379,
'PASSWORD': '',
'DATABASE': 0,
'SSL': False,
},
'caching': {
'HOST': 'localhost',
'PORT': 6379,
'PASSWORD': '',
'DATABASE': 1,
'SSL': False,

:x  (save & exit)
```

We need to generate the secret key here, and then input to the file.

```
$ python3 /opt/netbox/netbox/generate_secret_key.py
kXw)2e07tO+Q(Eio^qcBF*&KgIYapnyC@HWAUj3vr96RfMz1Ll
```

Copy the key and paste it to the `SECRET_KEY` in the `configuration.py` file.

```
SECRET_KEY ='kXw)2e07tO+Q(Eio^qcBF*&KgIYapnyC@HWAUj3vr96RfMz1Ll'
```

Now its time to make the run. We will execute the `upgrade.sh` script. That is going to do few task for us.

- It will create a Python virtual environment
- All the required Python packages/modules will be installed
- Database schema will be migrated

```
$ sudo /opt/netbox/upgrade.sh

Creating a new virtual environment at /opt/netbox/venv...
Installing Python system packages (pip3 install wheel)...
...
...

Finished.
Collecting static files (python3 netbox/manage.py collectstatic --no-input)...

961 static files copied to '/opt/netbox/netbox/static'.
Removing stale content types (python3 netbox/manage.py remove_stale_contenttypes --no-input)...
Removing expired user sessions (python3 netbox/manage.py clearsessions)...
Clearing cache data (python3 netbox/manage.py invalidate all)...
------------------------------------------------------------------------
WARNING: No existing virtual environment was detected. A new one has
been created. Update your systemd service files to reflect the new
Python and gunicorn executables. (If this is a new installation,
this warning can be ignored.)

netbox.service ExecStart:
    /opt/netbox/venv/bin/gunicorn

netbox-rq.service ExecStart:
   /opt/netbox/venv/bin/python

After modifying these files, reload the systemctl daemon:
  > systemctl daemon-reload
------------------------------------------------------------------------
Upgrade complete! Don't forget to restart the NetBox services:
  > sudo systemctl restart netbox netbox-rq

...
...
```

It will take few minutes to complete the task.

As NetBox doesnt create its user, we have to do it manually. Enter the python environment and use `apnic` as the user and `training` as the password.

Type password two times to confirm

```
$ source /opt/netbox/venv/bin/activate

(venv) $ cd /opt/netbox/netbox
(venv) $ python3 manage.py createsuperuser
        Username: apnic
        Email address: training@apnictraining.net
        Password:
        Password (again):

        Superuser created successfully.
```

7. Setup the middleware - Gunicorn

For NetBox gunicorn is automatically installed with Django. Next, we are going to setup the service. We will keep the default settings for the LAB.

```
$ sudo cp /opt/netbox/contrib/gunicorn.py /opt/netbox/gunicorn.py
```

8. Startup configuration for NetBox

Copy the systemd files to the respective directory.

```
$ sudo cp -v /opt/netbox/contrib/*.service /etc/systemd/system/
$ sudo systemctl daemon-reload
```

Restart and check the NetBox services.

```
$ sudo systemctl start netbox
$ sudo systemctl start netbox-rq
$ sudo systemctl enable netbox
$ sudo systemctl enable netbox-rq
$ sudo systemctl status netbox netbox-rq
```

9. Configure the HTTP service

```
$ sudo apt install -y apache2
```

Skip `apache2` installation, if you have done LibreNMS LAB successfully.

To avoid the complexity we will use the default configuration file. But make sure you modify the `ServerName` portion, and put `#` before the `SSL` configuration lines and change the default port from `443` to `80`, as we are not using HTTPS in this LAB.

```
$ sudo cp /opt/netbox/contrib/apache.conf /etc/apache2/sites-available/netbox.conf
$ sudo vim /etc/apache2/sites-available/netbox.conf
```

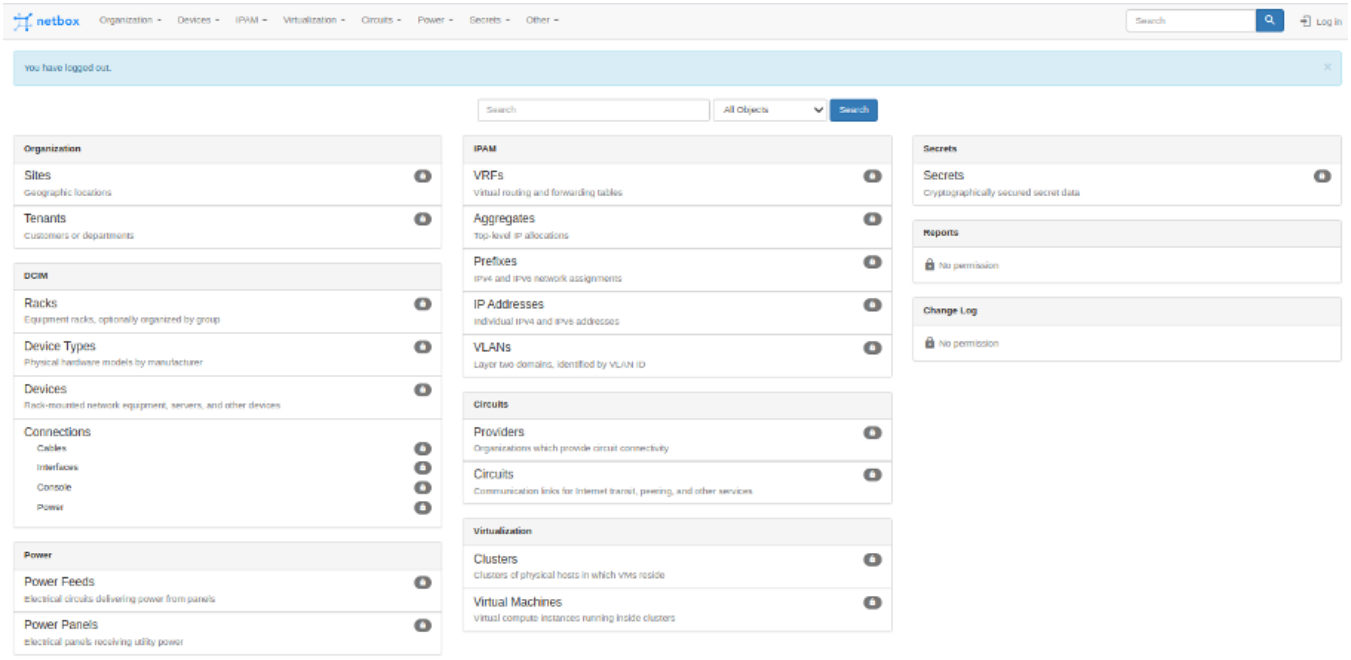
```
        <VirtualHost *:80>
            ProxyPreserveHost On
            ServerName groupX-server-netbox.apnictraining.net

            #    SSLEngine on
            #    SSLCertificateFile /etc/ssl/certs/netbox.crt
            #    SSLCertificateKeyFile /etc/ssl/private/netbox.key

            Alias /static /opt/netbox/netbox/static

            <Directory /opt/netbox/netbox/static>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Require all granted
            </Directory>

            <Location /static>
                ProxyPass !
            </Location>

            RequestHeader set "X-Forwarded-Proto" expr=%{REQUEST_SCHEME}
            ProxyPass / http://127.0.0.1:8001/
            ProxyPassReverse / http://127.0.0.1:8001/
        </VirtualHost>

        :x (save & exit)
```

Enable the `netbox` site, and restart the apache service.

```
$ sudo a2enmod ssl proxy proxy_http headers
$ sudo a2ensite netbox
$ sudo systemctl restart apache2
```

Next, hit the browser - `http://groupX-server-netbox.apnictraining.net`

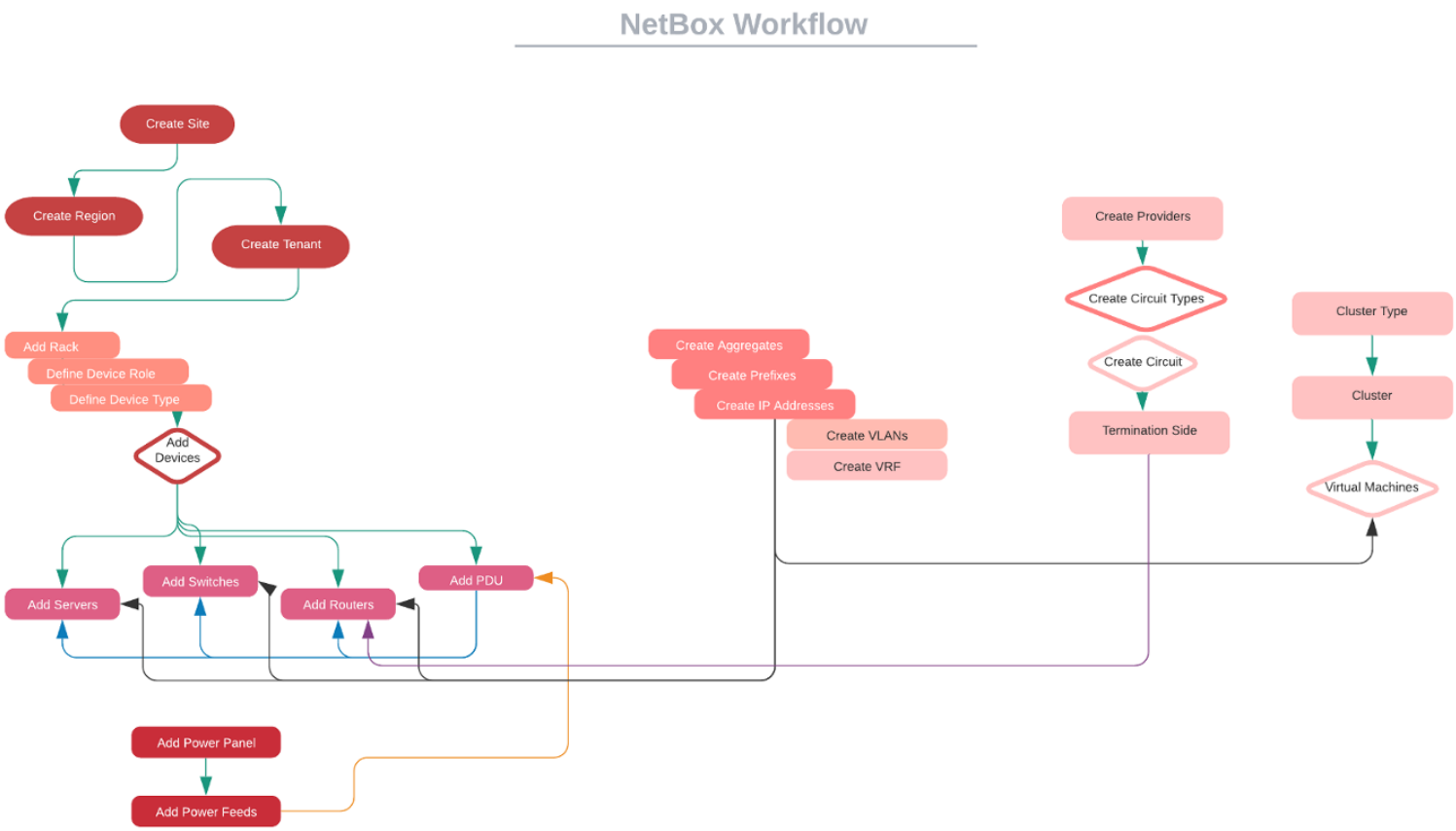
You will get a page like below.



Login to the system, with the username `apnic` and password `training`

Data Center Management

We will cover the fundamental part of NetBox from the flow diagram.



10. Explore the NetBox service - DCIM

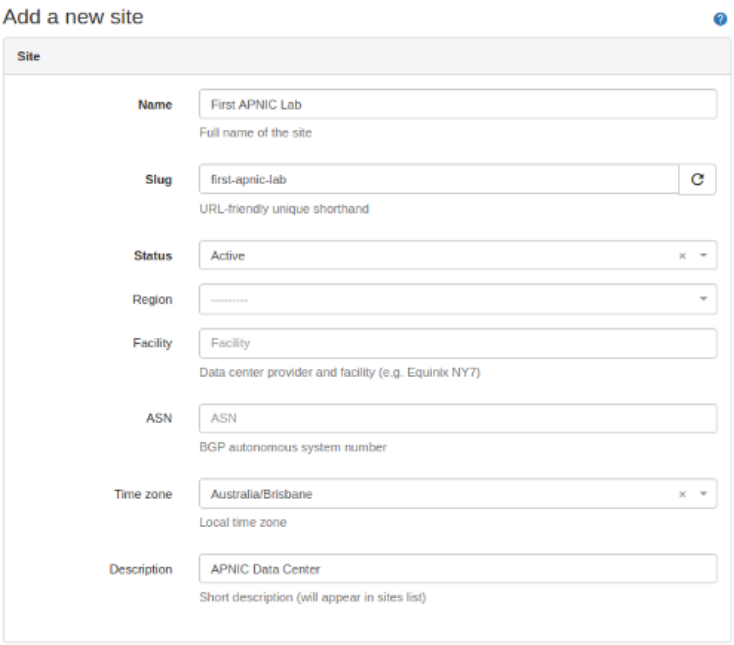
We will practice the following steps to work on NetBox, assuming we are configuring Network-Documenation for a Data Center.

- Create the first site
- Create the first region
- Connect the region with the site
- Create the tenant to define the department
- Connect the tenant with the site
- Add new Rack
- Define device role
- Define device type
- Add devices
- Add power-panels
- Add power-feeds
- Add power distribution unit (PDU)
- Addup few component to the server

Create the first site:

Go to the homepage and click `Sites` under the `Organization` tab.

New page will comeup, click on the `+Add` button and fillup the gap.

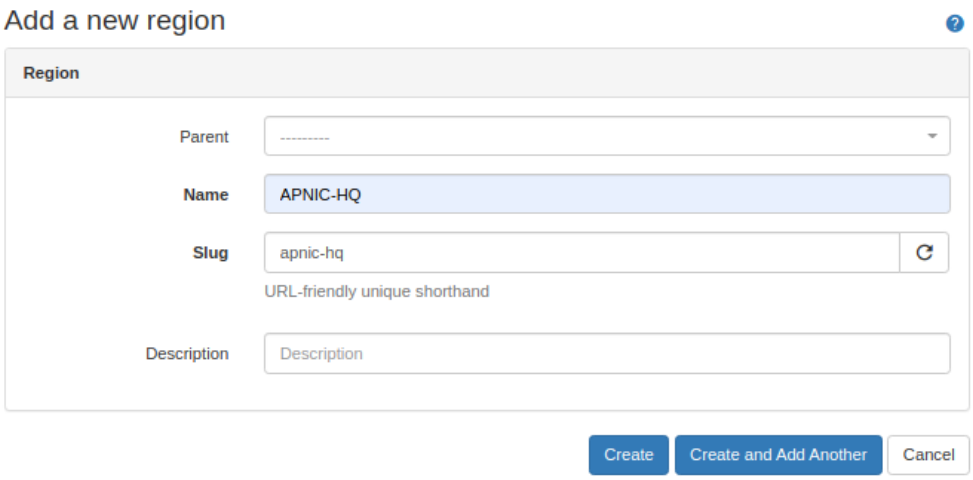


Create the first region:

Next, we have to create the region to complete the first part of the `Organization` tab.

To do that, again go to the homepage and click `Regions` under the `Organization` tab.

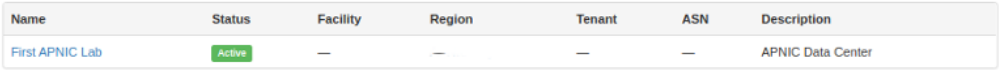
New page will comeup, click on the `+Add` button and fillup the gap.

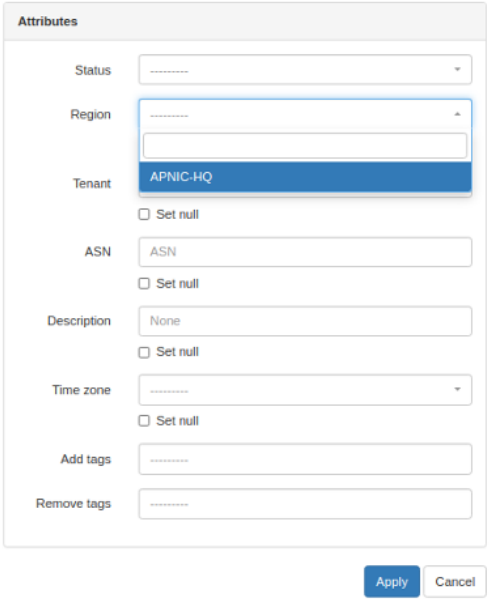


Connect the region with the site:

Now we need connect `Regions` to `Sites`. Go to `Sites` from the `Organization` tab, Select the `First APNIC Lab` and click on `Edit Selected`.

A new window come up. Next, right side of the window there are few options, select `APNIC-HQ` from the drop down menu at `Region` field, and click on `Apply`



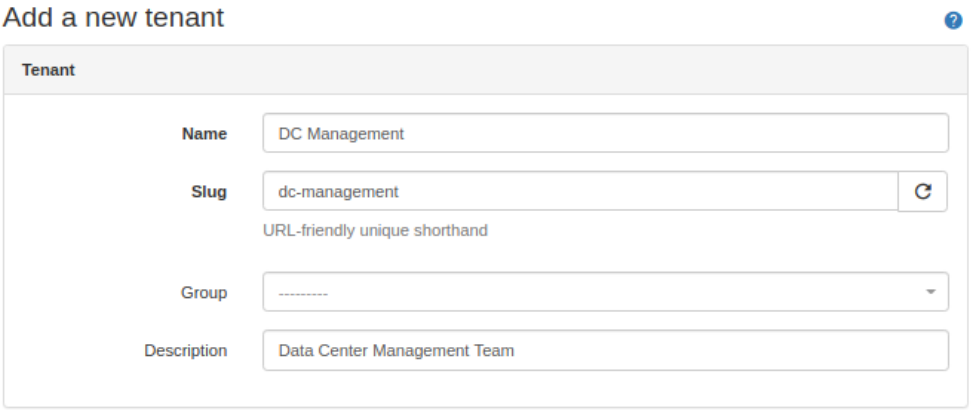


Now it should look like this.

Create a tenant:

Lets create a tenant to define the department.

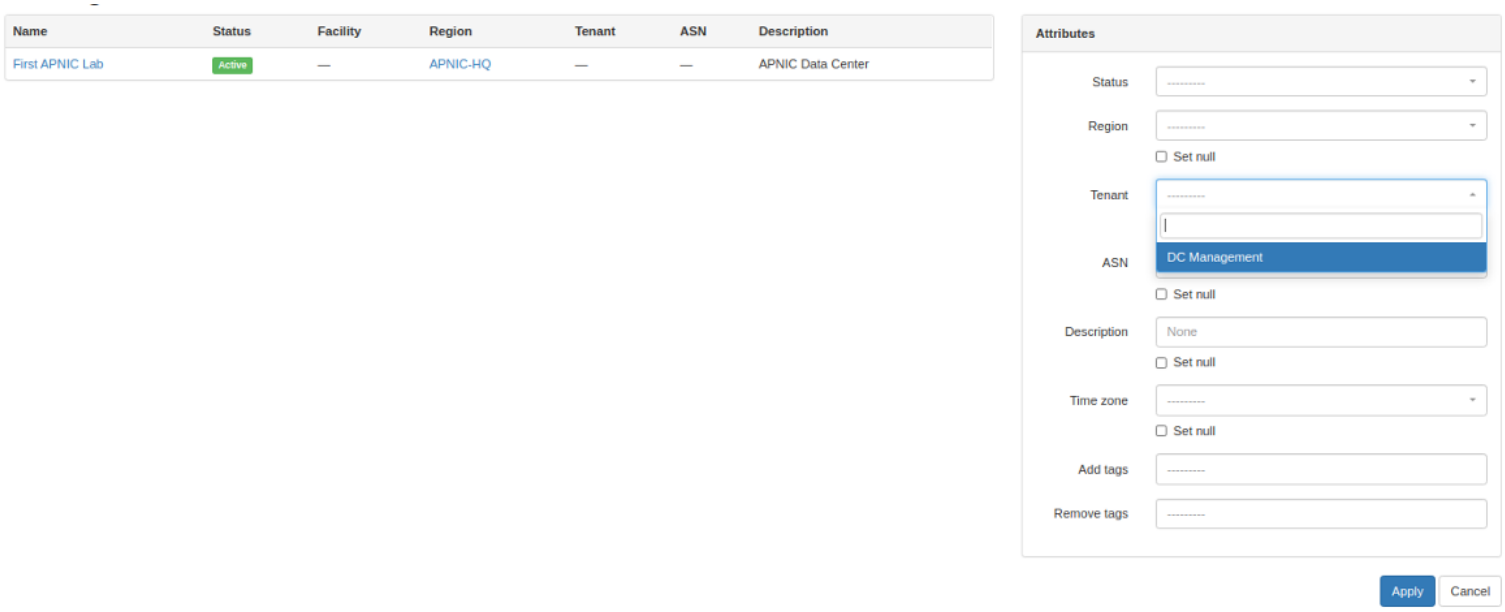
Go to the home page, and nevigate `Tenants` option from the `Organization` block. Click it, new window will popup, click on the `+Add` button to add a new one. Fillup the gap and click on `Create` button.



Connect the tenant with the site:

Go to `Sites` from the `Organization` tab, Select the `First APNIC Lab` and click on `Edit Selected` .

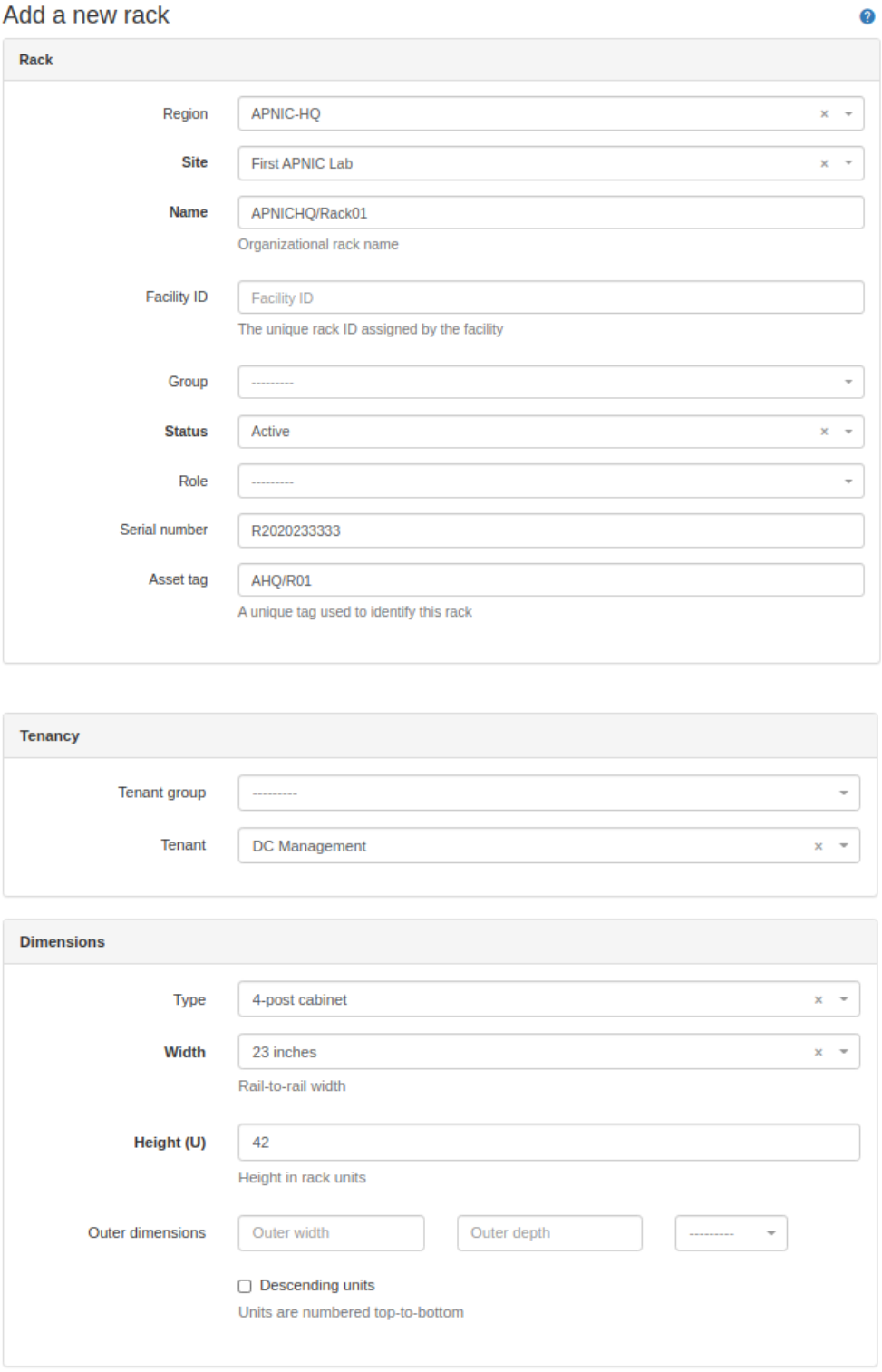
Next, right side of the window there are few options, select `DC Management` from the drop down menu at `Tenant` field, and click on `Apply`



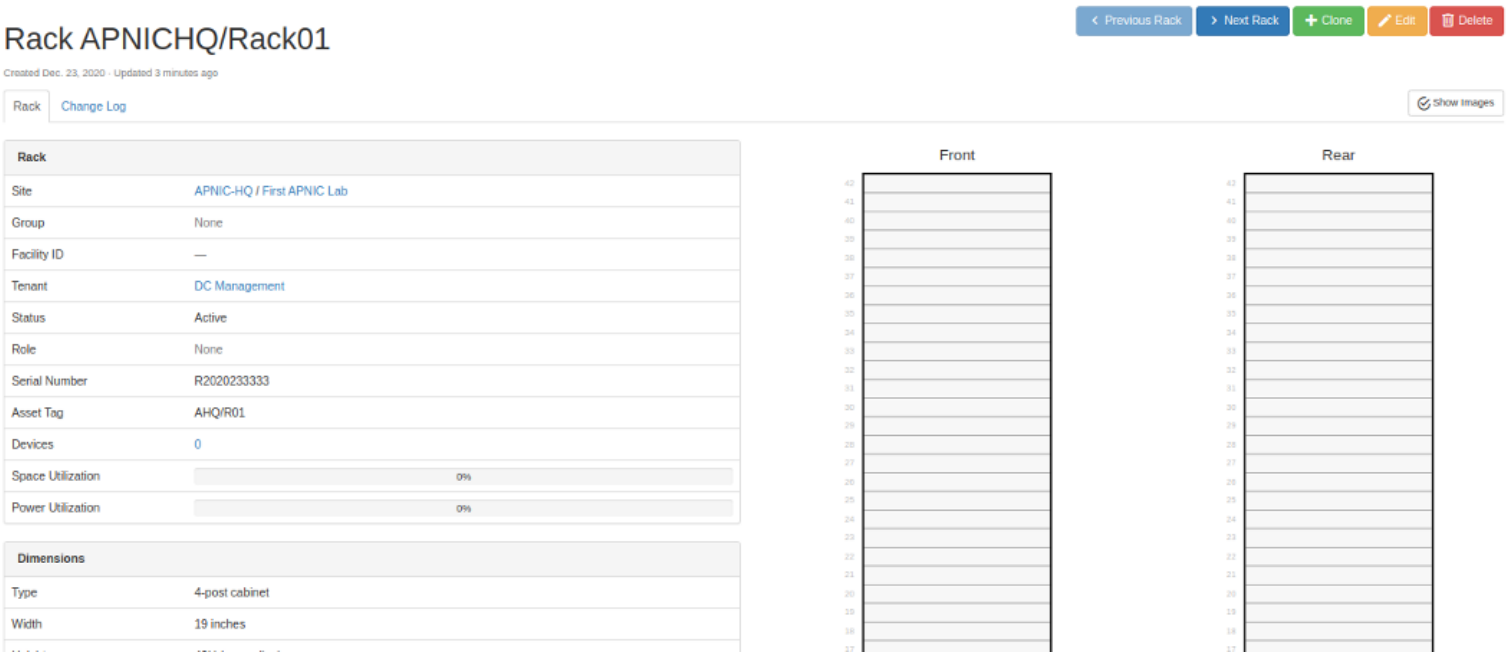
Now it should it look like this.

Add new RACK:

To add a new rack, go to the home page, and nevigate `Racks` option from the `DCIM` block. Click it, new window will popup, click on the `+Add` button to add a new one. Fillup the gap, use `First APNIC Lab` as site, use `APNIC-HQ` for region, `DC Management` for tenant, from the drop down option respectively; use the name of the rack as `APNICHQ/Rack01` and click on `Create` button.



The output will be like this.



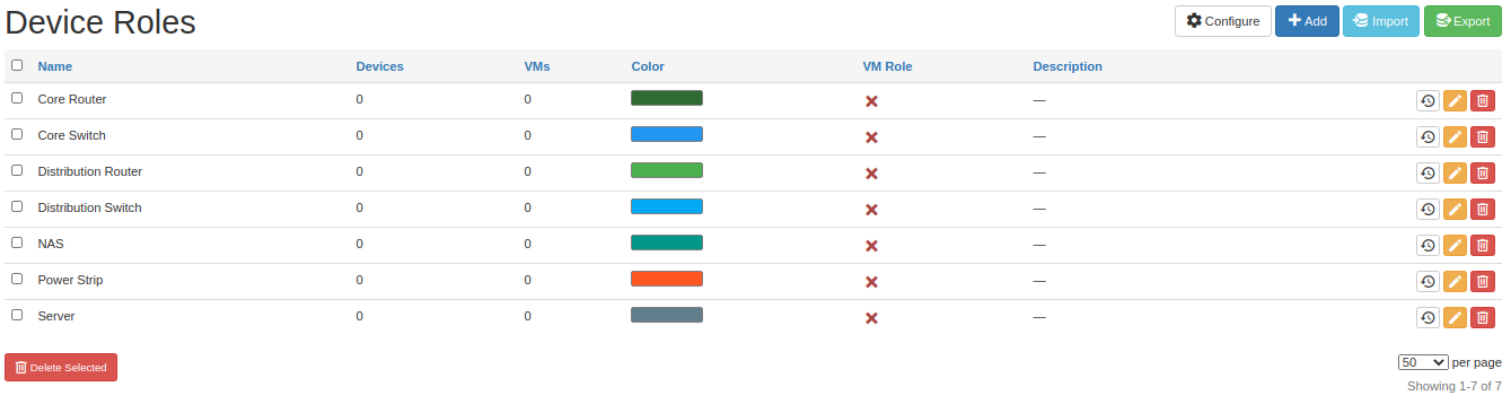
Define device role:

Before adding new device, we have to create few necessary definition for devices.

To create device role, nevigate `Device Roles` option from the drop down menu `Devices`. Click it, new window will popup, click on the `+Add` button to add a new one. Add router, server, etc with assigning the color code.

Note: make sure you uncheck the option `VM Role`

it should look like this.

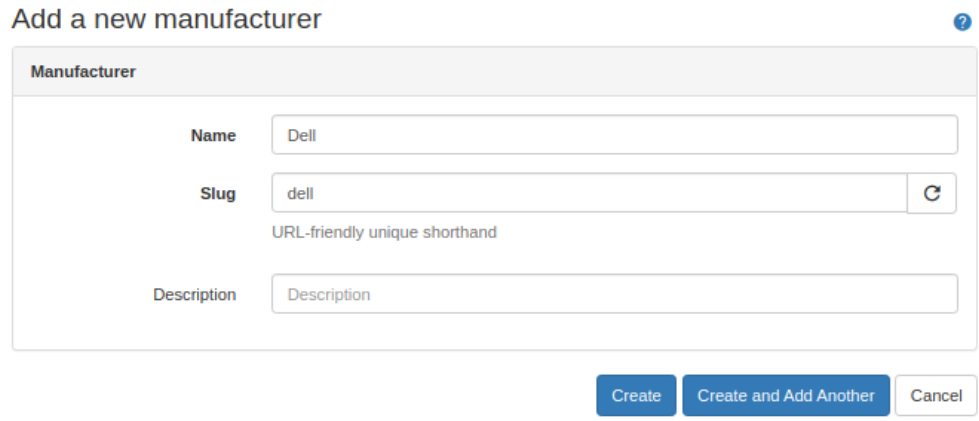


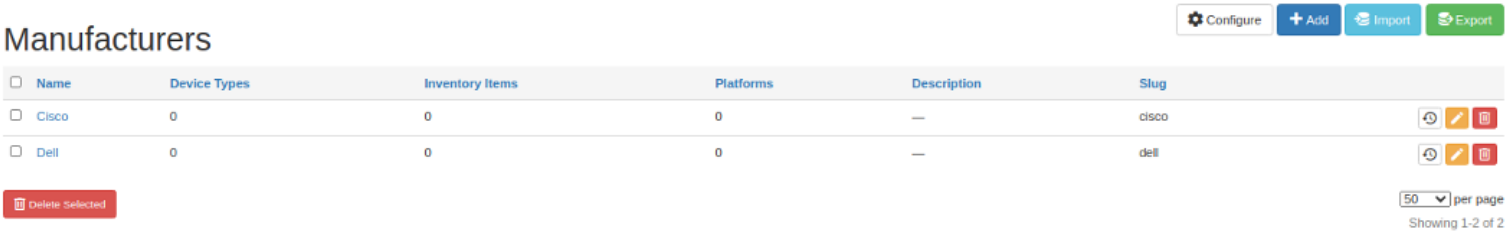
Define device type:

Need to create the `Manufacturers` first, before creating device types.

From the `Devices` drop down menu, select `Manufacturers` and then click `+Add` from the new window.

Create `Cisco` and `Dell`, by fill in the gap, and it will look like below.

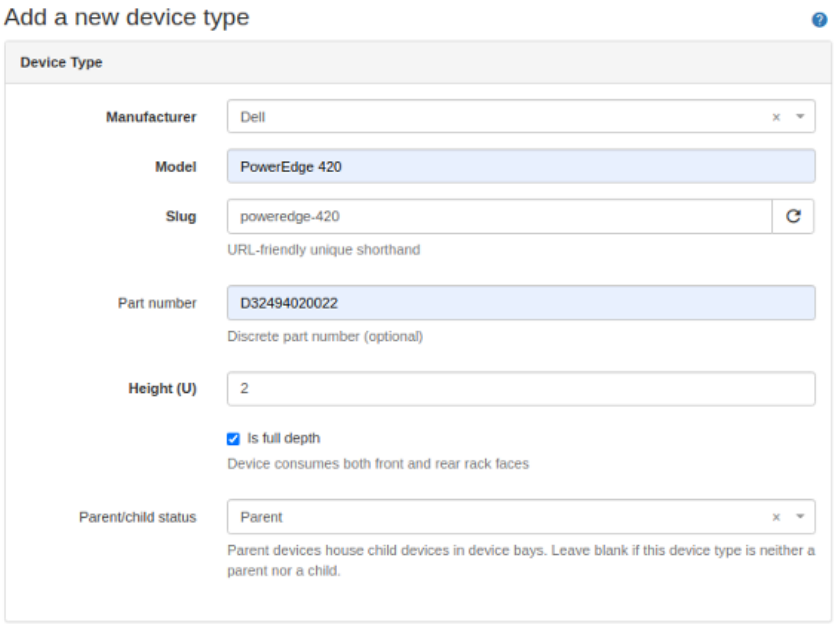


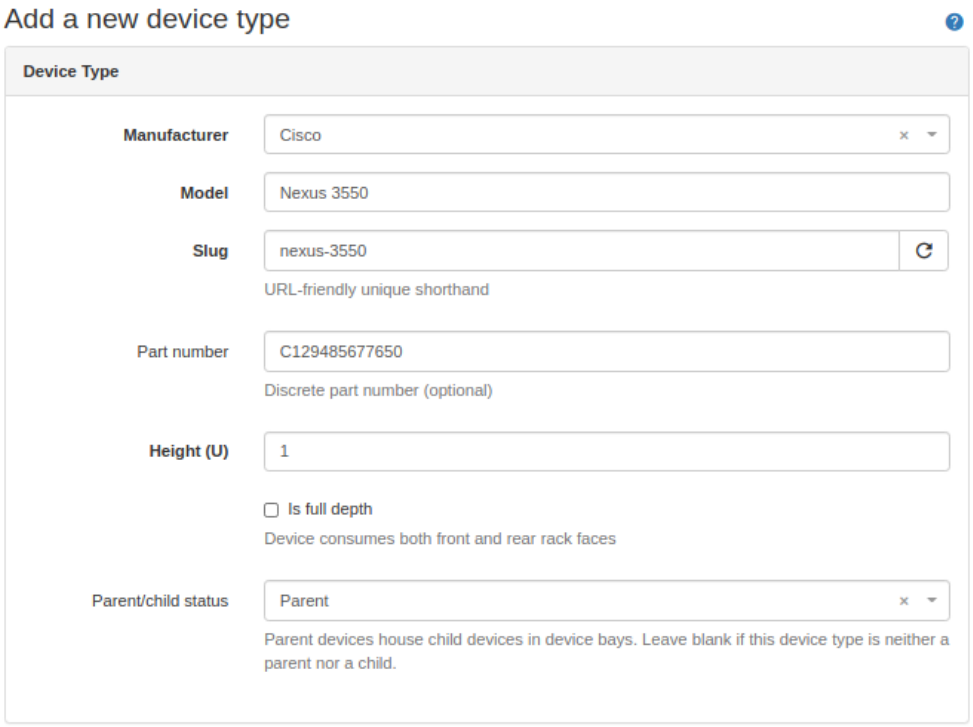


Next, from the `Devices` drop down menu, select `Device Types` and then click `+Add` from the new window.

Fill the gap with sample specification of Dell. Assuming the server is Dell PowerEdge 420, which is 2U rack.

Note: As this is 2U rack server, it will cover from `Front` to `Rear`, and that is why option `Full Depth` should be checked. But for the Cisco switch, `Full Depth` option should be unchecked.

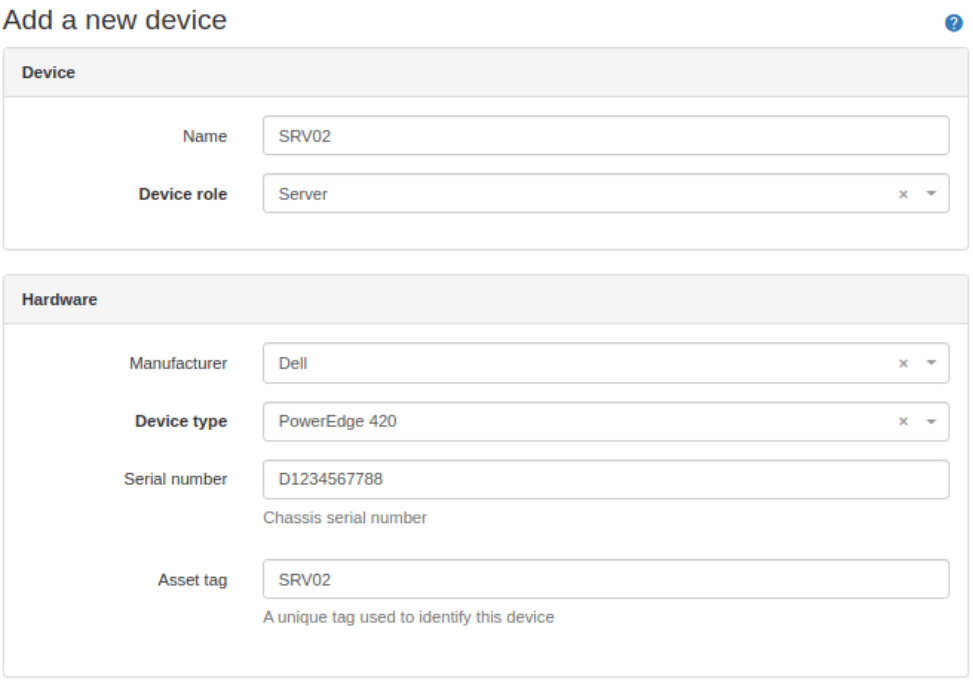


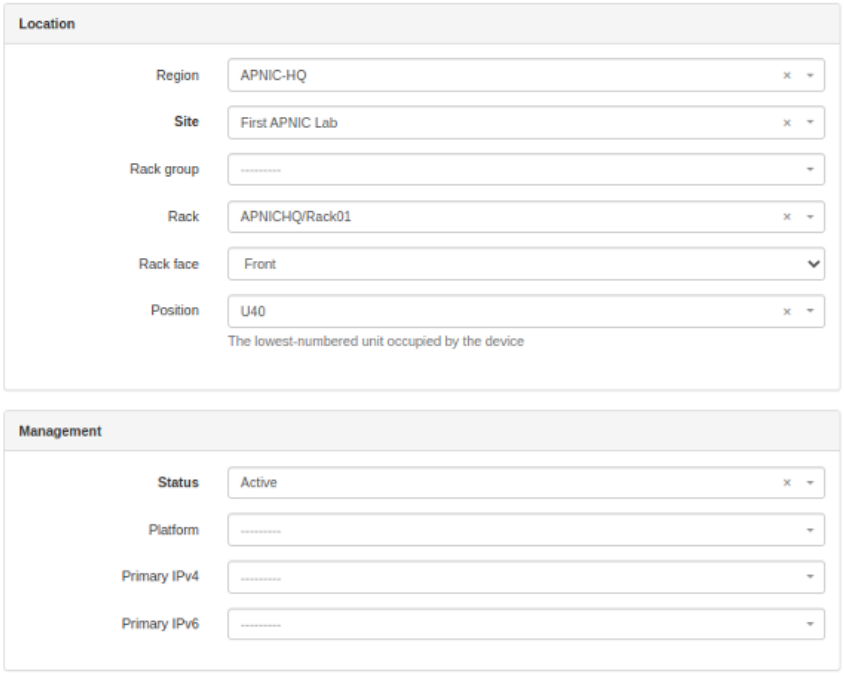


Add devices:

Finally, we are adding devices now.

Select `Devices` option from the drop down menu `Devices` . New window will popup, click on the `+Add` button to add a new one. Add switch and server, and select all the option from the drop down menu accordingly, keep in mind to check, `Hardware` , `Location` `Tenancy` .

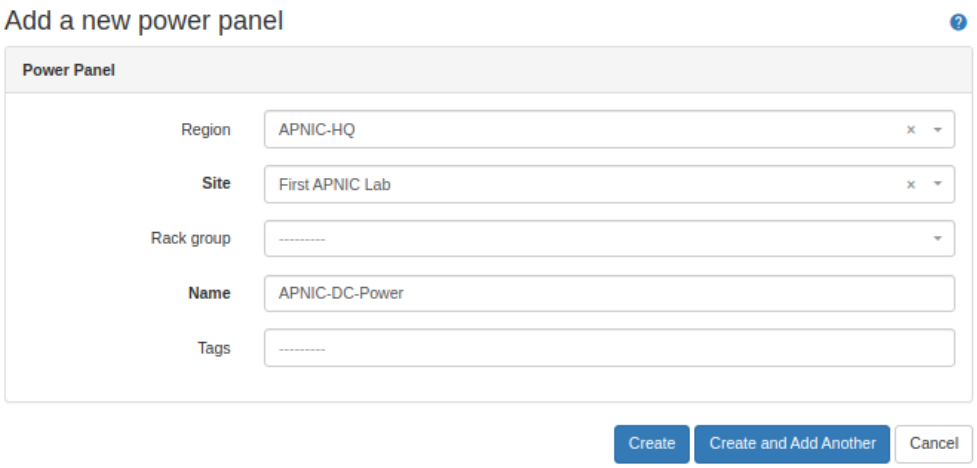




Add power-panels:

Add two different Power Panels to ensure redundant power supply from two separate main power grid. Its the main power source.

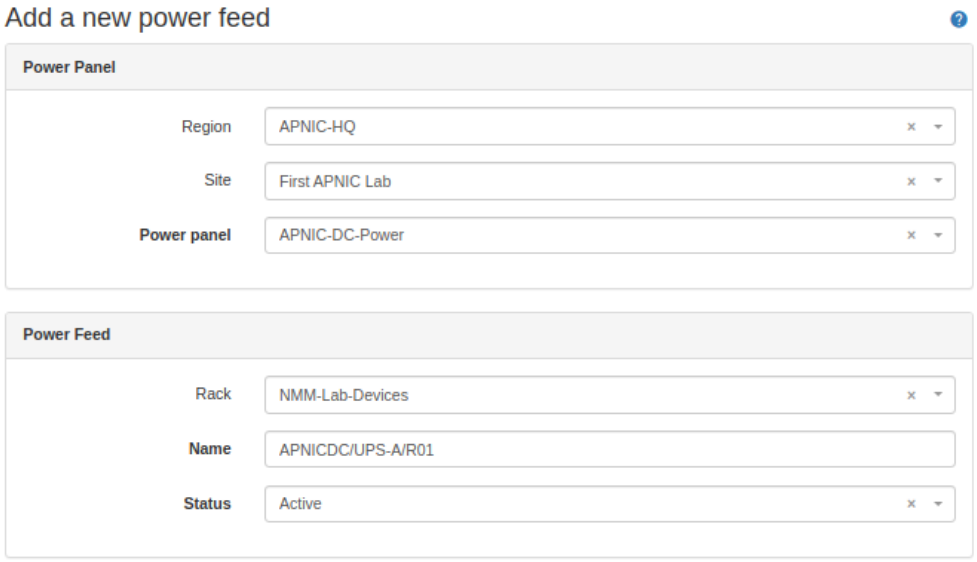
e.g: APNIC-DC-Power and APNIC-DC-Power-Sec.



Add power-feeds:

Every rack should have two different power feeds from two separate power panels, asumming two separate online UPS in place.

e.g: APNICDC/UPS-A/R01 and APNICDC/UPS-B/R01; for second one, change the Power-Panel to `APNIC-DC-Power-Sec` and assign name as `APNICDC/UPS-B/R01`.

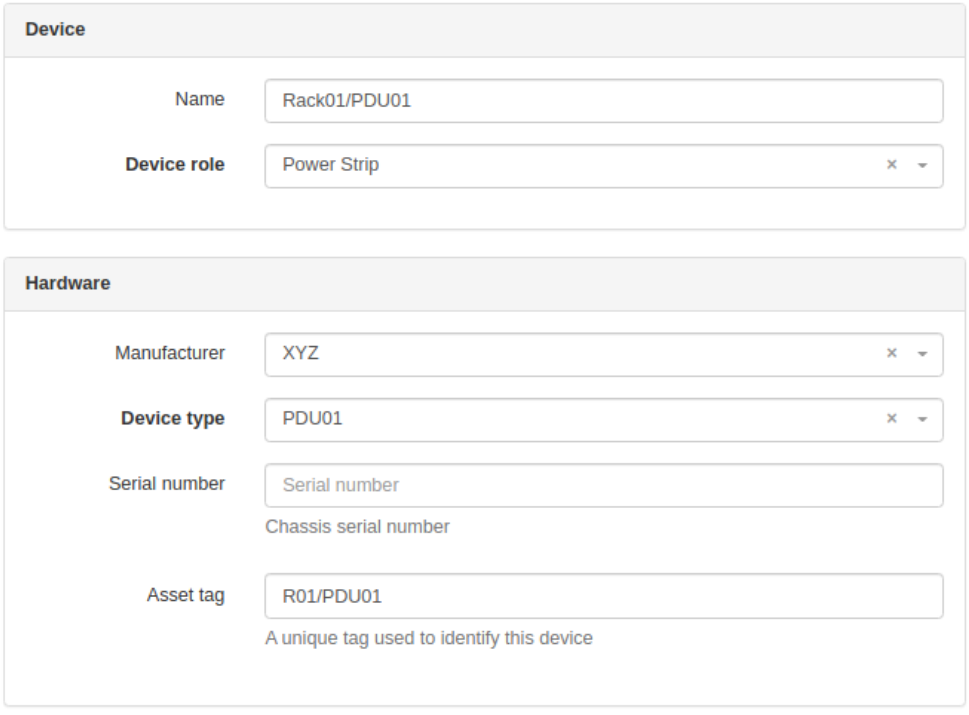


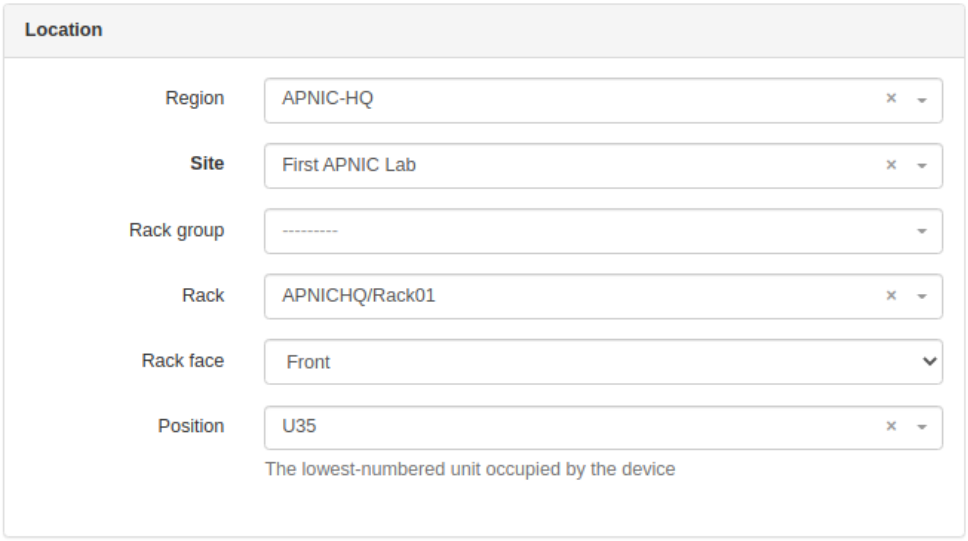
Add PDU

PDU = Power distribution unit

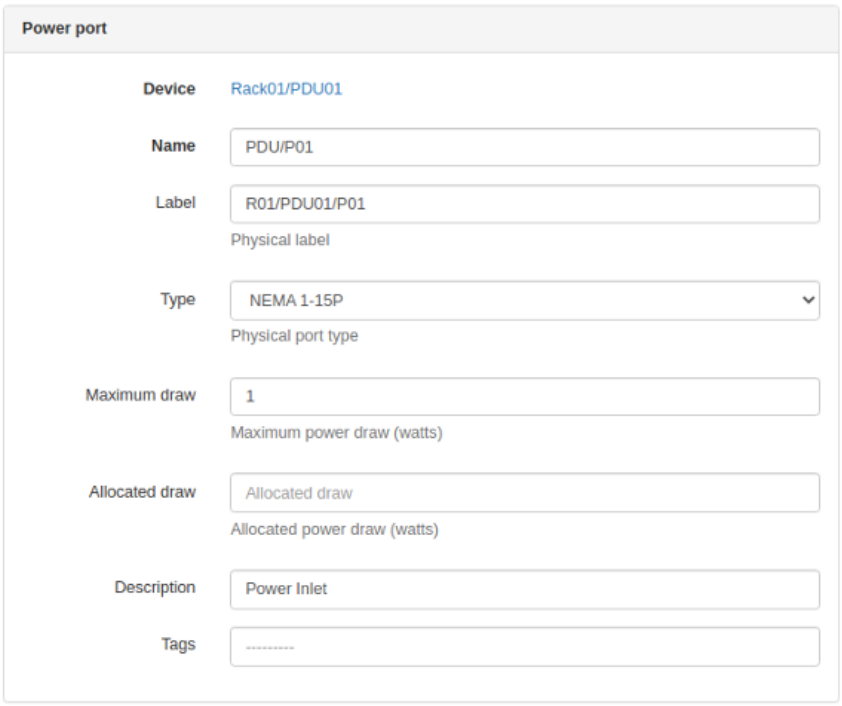
Before creating a new PDU device, create `Manufacturer` as `XYZ`, device role `Power Strip`, and device type `PDU01`, then Go to the `Devices` and click on `+add`.

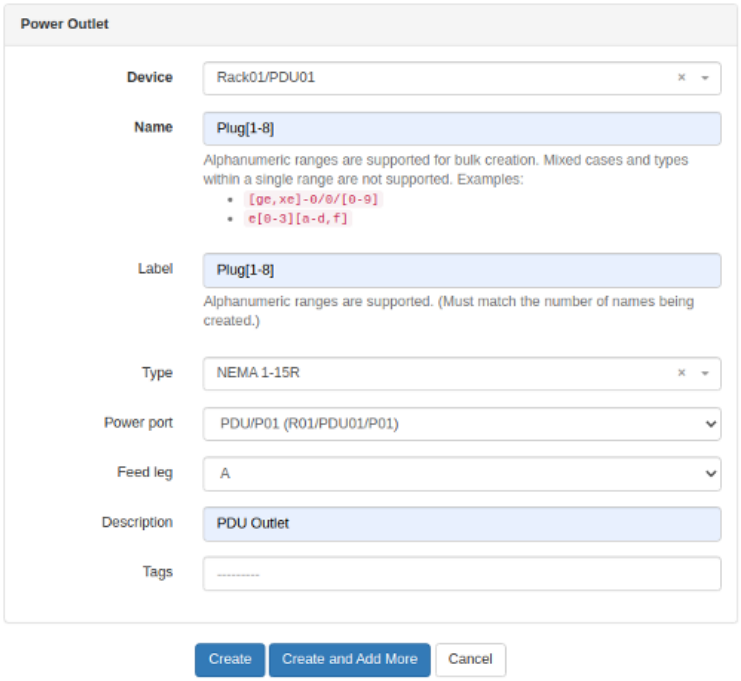
Give the PDU a name `Rack01/PDU01`, `Asset Tag` as `R01/PDU01`, and carefully choose all the options accordingly.



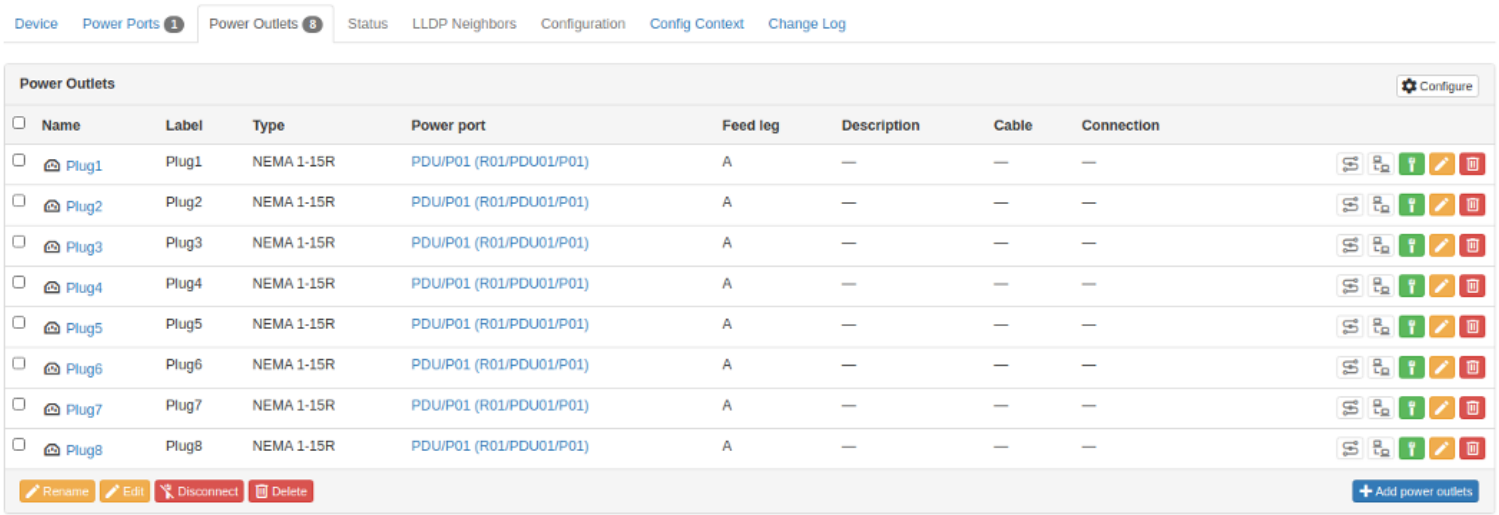


Next, create `power ports` as the power inlets and `power outlets` as power outlet, of the PDU, from the `Add New Components` option.



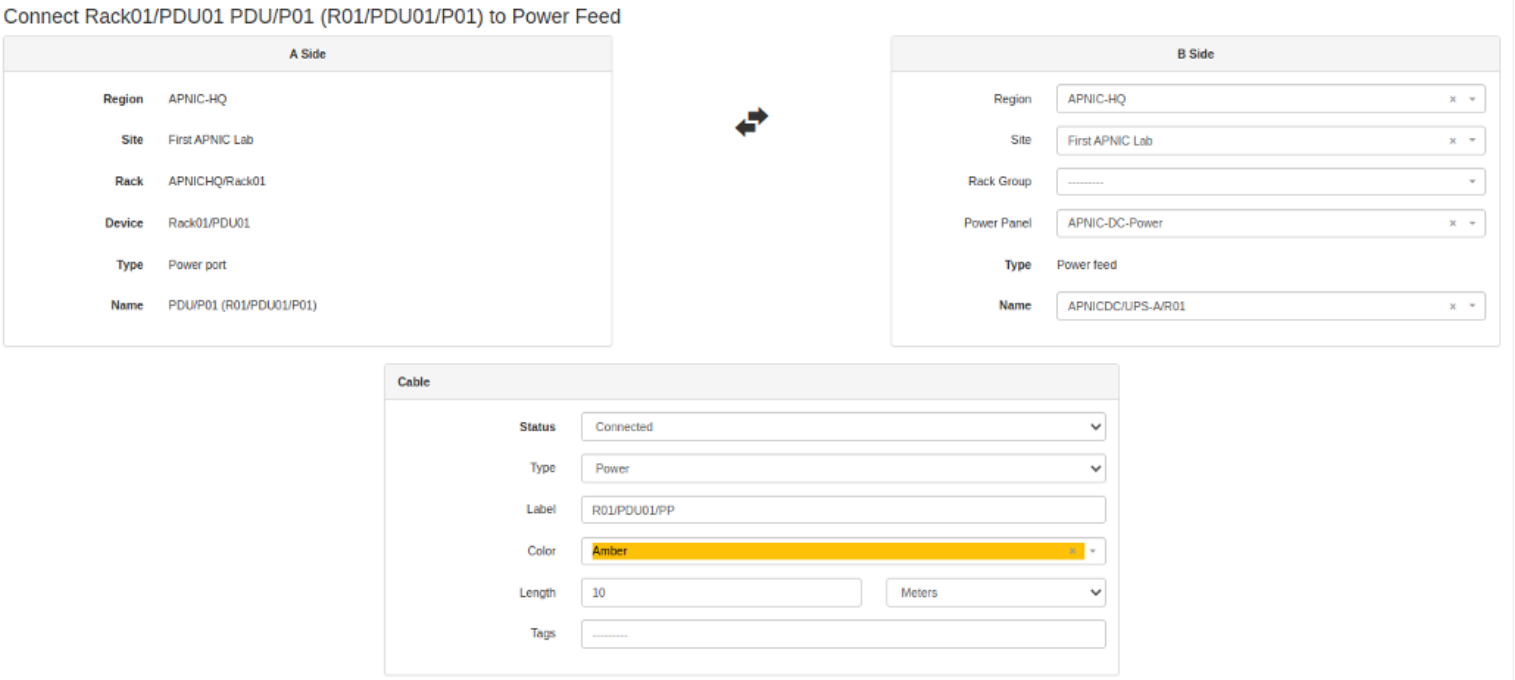


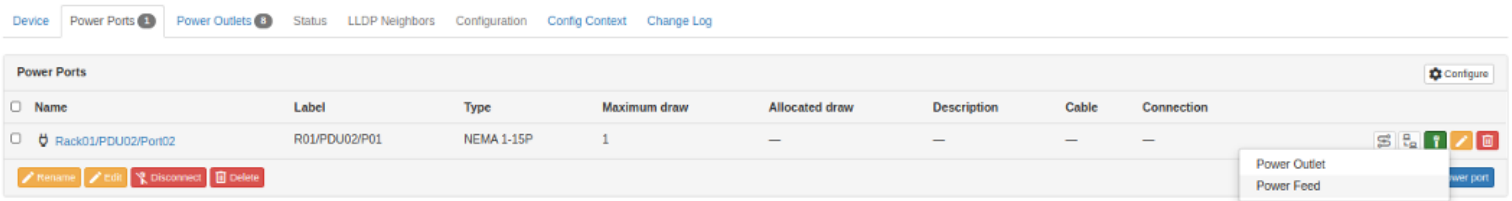
The power outlet should look like this.

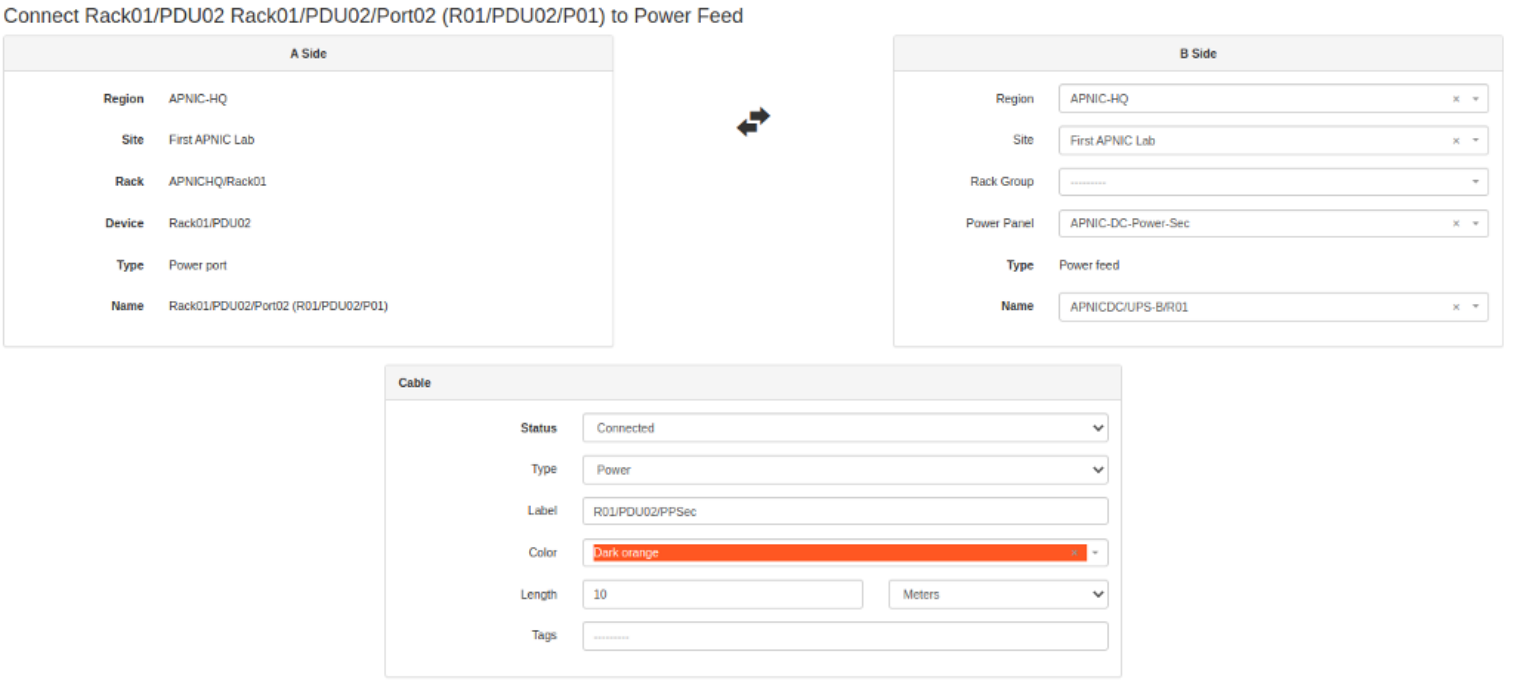


Now create another PDU for the rack01 to get power feed from the different power supply.

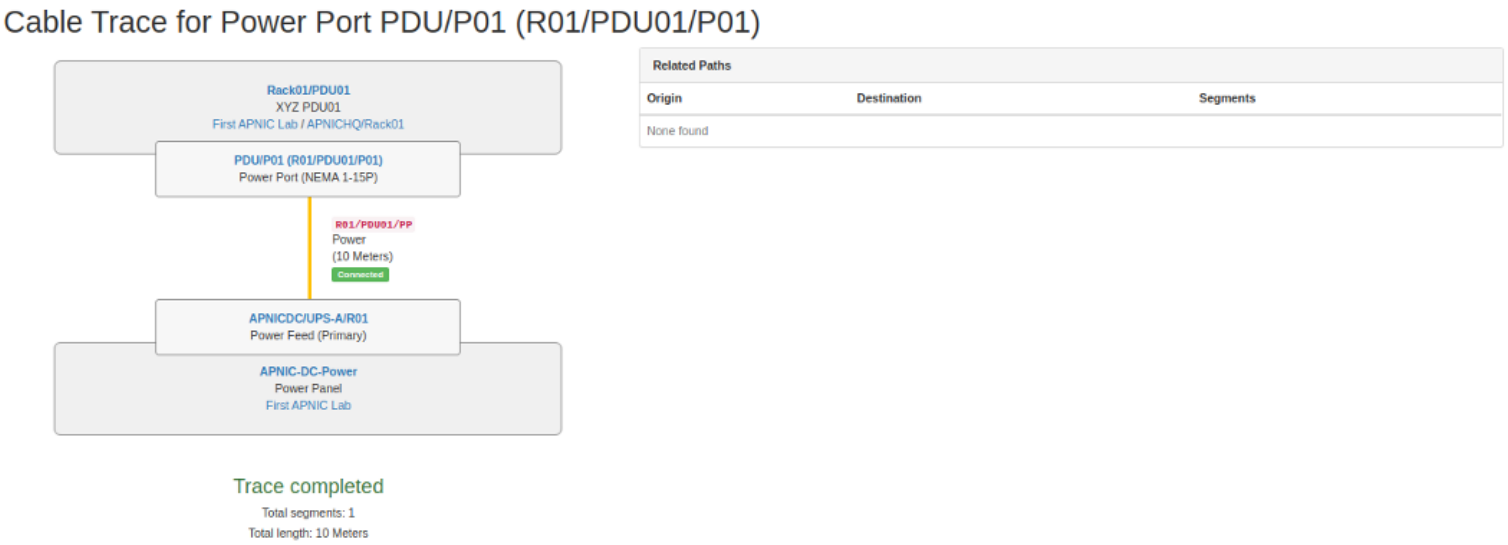
Let us connect the PDU01 and PDU02 inlet to the power feed. Follow the options from the screeshots. You need to click on green colored connection icon and select `Power Feed`.







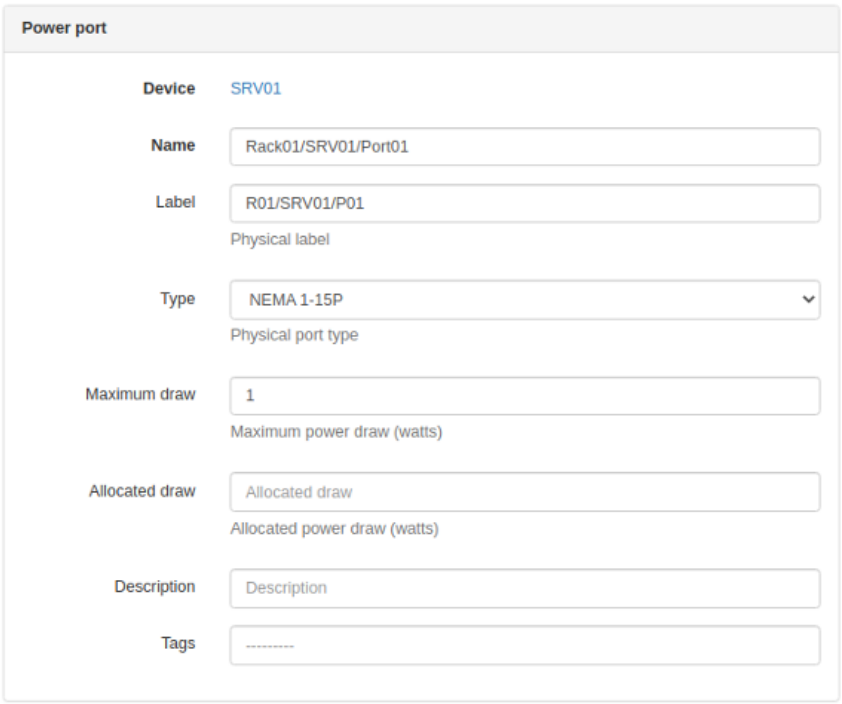
Let us check the connection from PDU to power feed.

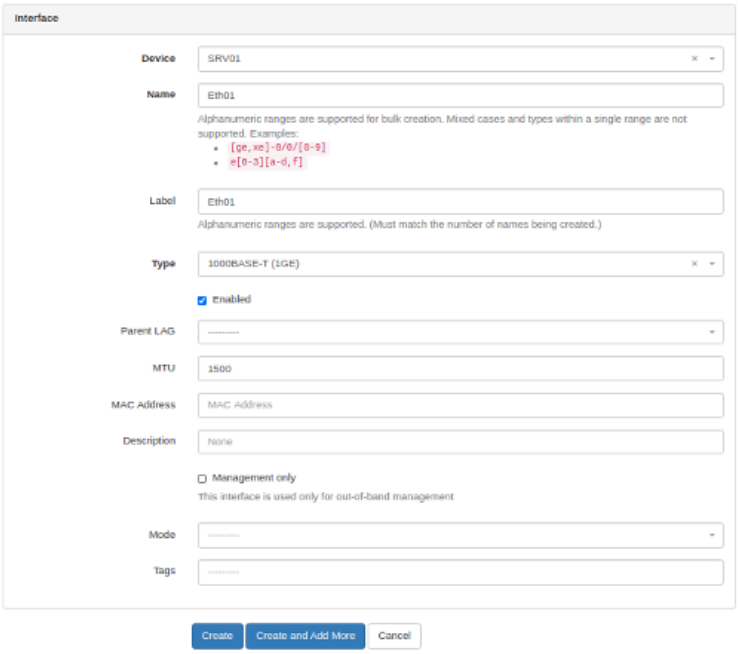


Addup few component to the server

Let us add few components for the device, first we add few components for server-01, Selecting `SRV01` from the `Devices` tab, choose `Power Ports` from the `Add Components` drop down menu. Power ports are named like - `Rack01/SRV01/Port01` .

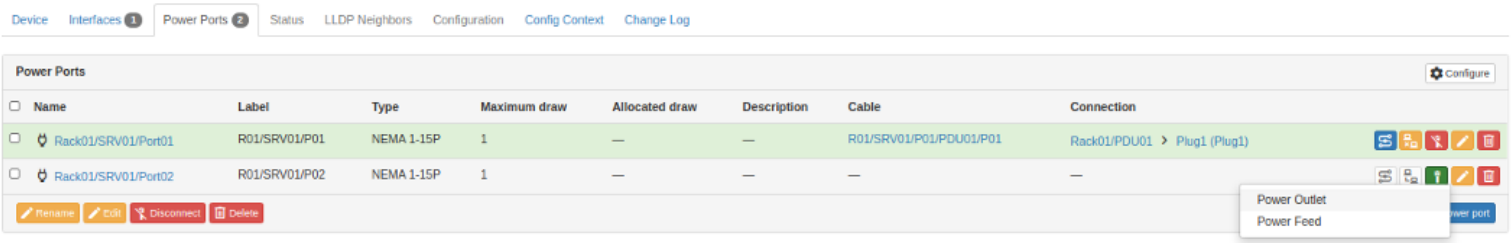
To add `Interfaces` , again click `Add Components` and give it a name `Eth0` with `1GE` from `Types` .

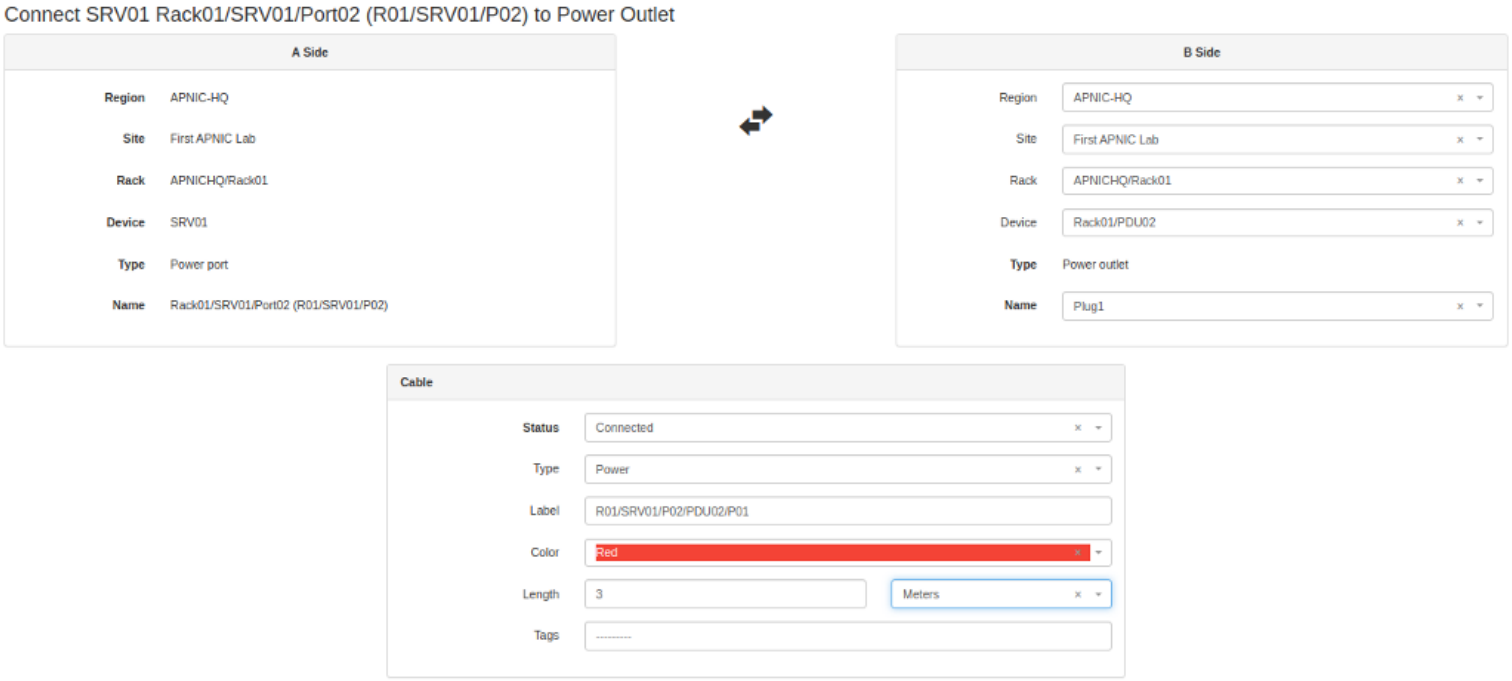




Add one more power port.

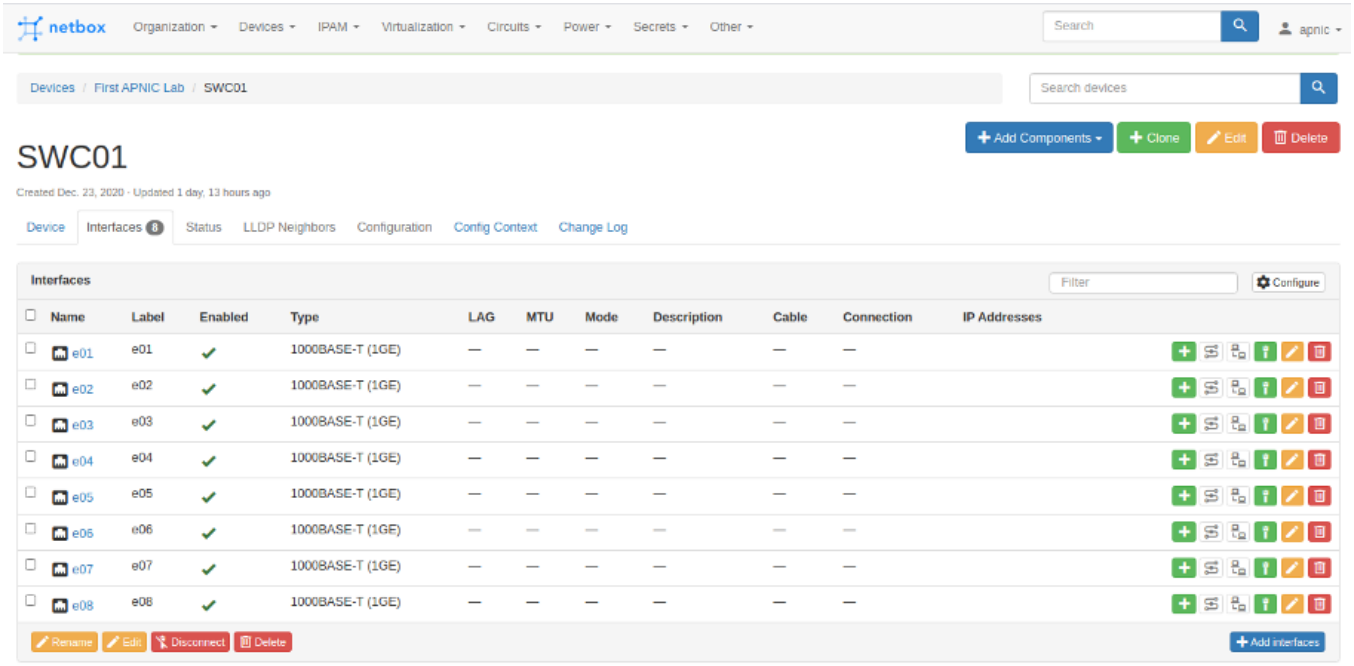
Now connect those two Power Ports with the PDU unit 01 and 02 accordingly, use the option `power outlet` to make the connection and check the status. Power cable tagging can be done following `Rack/Server/Power_port/PDU_number/Port_number`





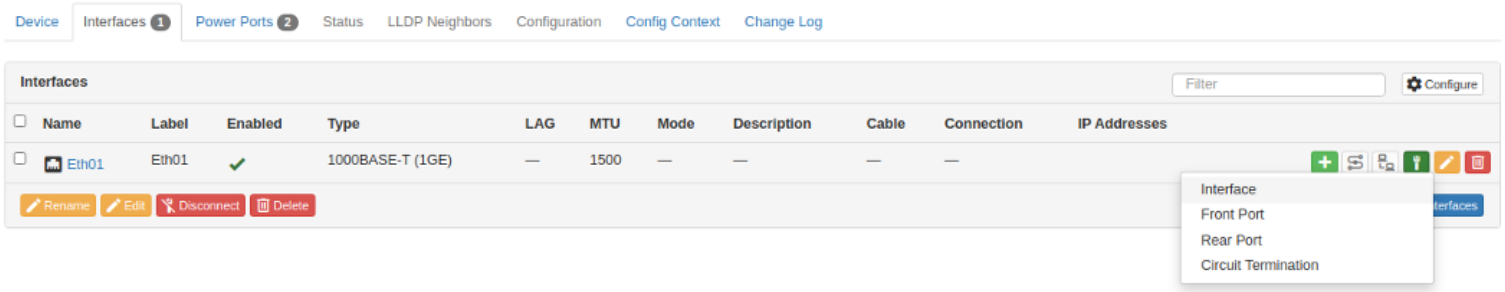
Next, let us add few ports to the switch, that we have created. It will be time consuming to add 24 or 48 ports, to simplify the LAB we will add 8 ports only to the Cisco Nexus 3550 switch, dont forget to add two power port for switch, and connect from two separate PDU as well.

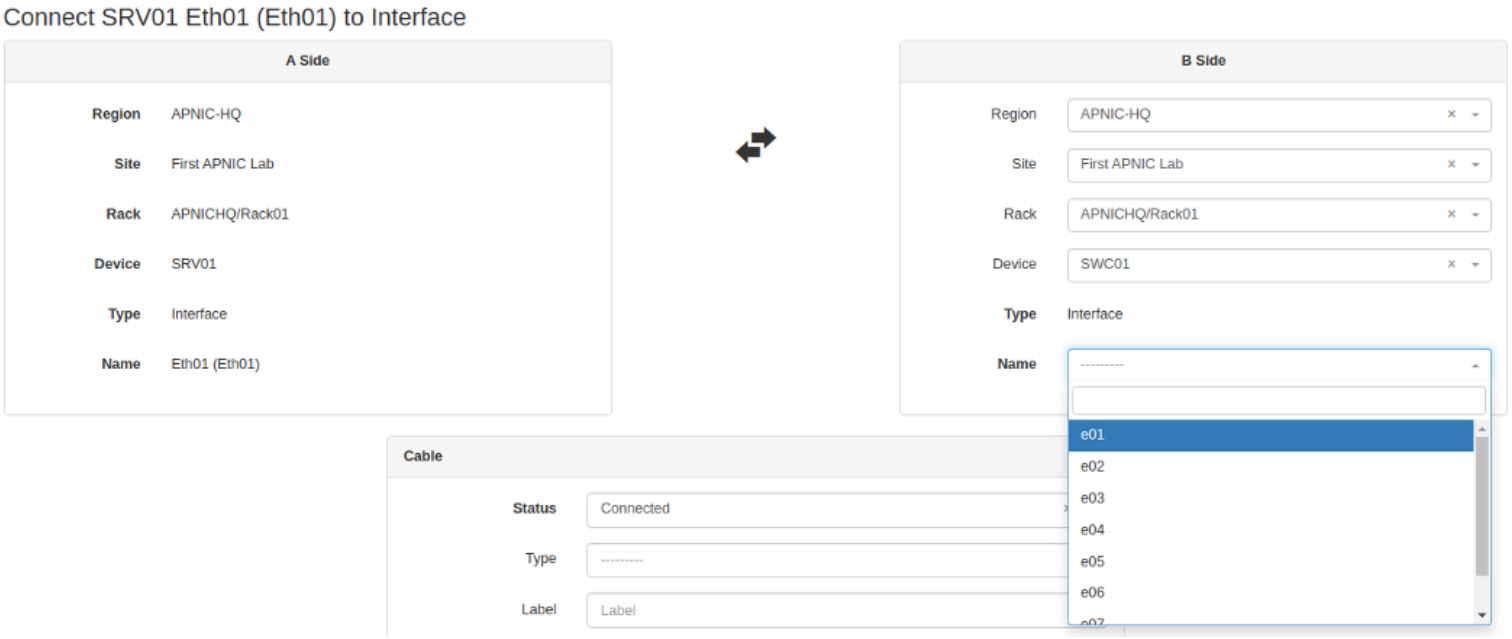
Go to `SWC01` from the `Devices` tab, and click on `Add Components` to add `Interfaces` . Naming can be done `e01, e02, e03` , select `1000BASE-T (1GE)` from the `Type` option. After creating all the 8 ports it should look like -

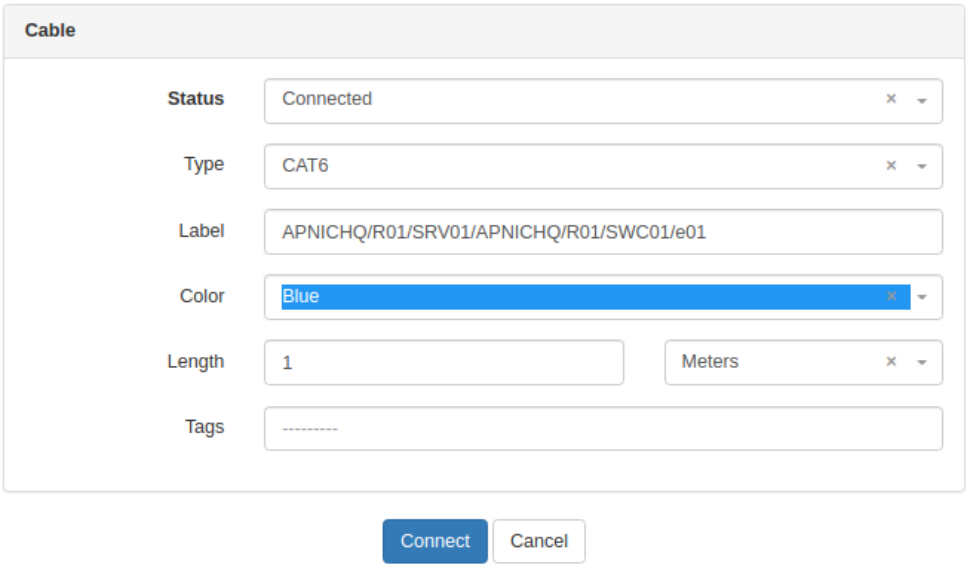


Let us connect the server ethernet port to the switch port. click on the connect icon, choose `interface` , a window will popup, `A Side` is the server side, and `B Side` is the destination side, for us here it is the switch that we have in our rack-01. Choose `swc01` the `Device` option, and then chose one port below to the `Interface` .

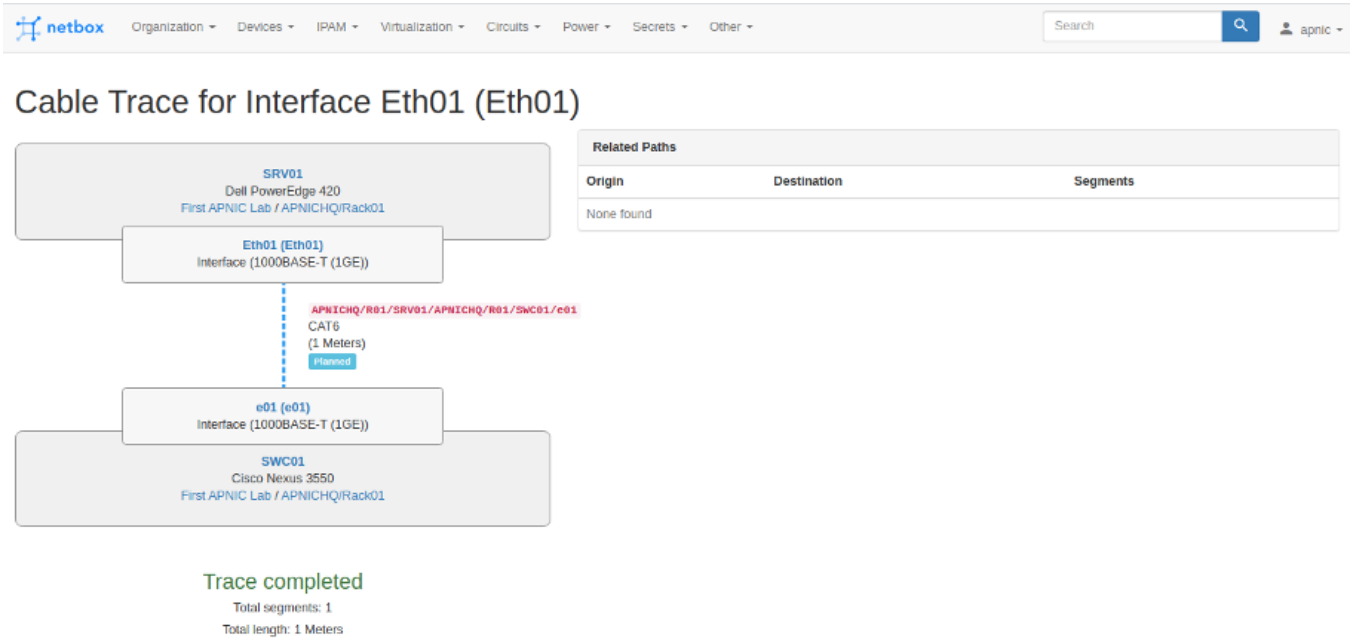
From the `Cable` box, select `CAT6` from the `Type` , use `Label` like we discuss at our presentation slides, `APNICHQ/R01/SRV01/APNICHQ/R01/SWC01/e01` `(Format: Source_Device_ID/Destination_Device_ID-Port_Number/Name)` .







Now it should look like below, to check this status, click on the `Trace` icon from the SRV01 `Interface` details.



So, till now, we have created a server and switch; gave them redundant power supply and network interface, and placed them inside the rack.

IP Address Management

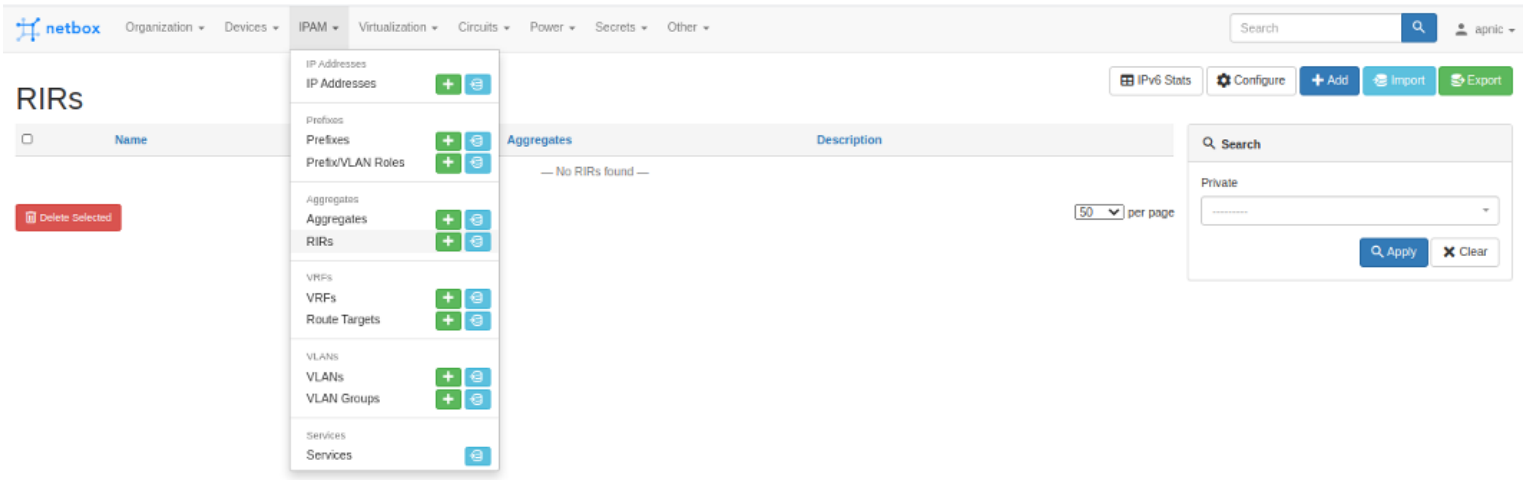
11. Explore the NetBox service - IPAM

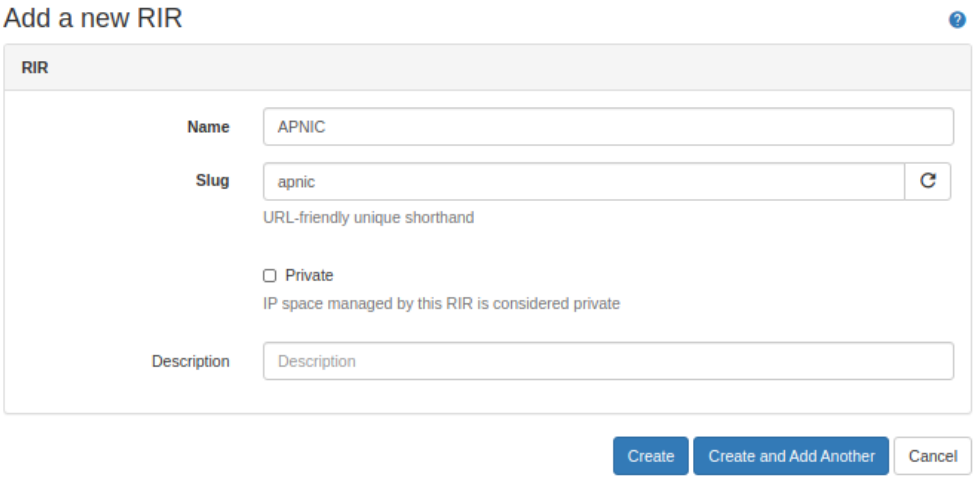
Now we will work on IPAM, and then go back to DCIM to see the full picture.

- Create Aggregates
- Create Prefixes
- Create IP address

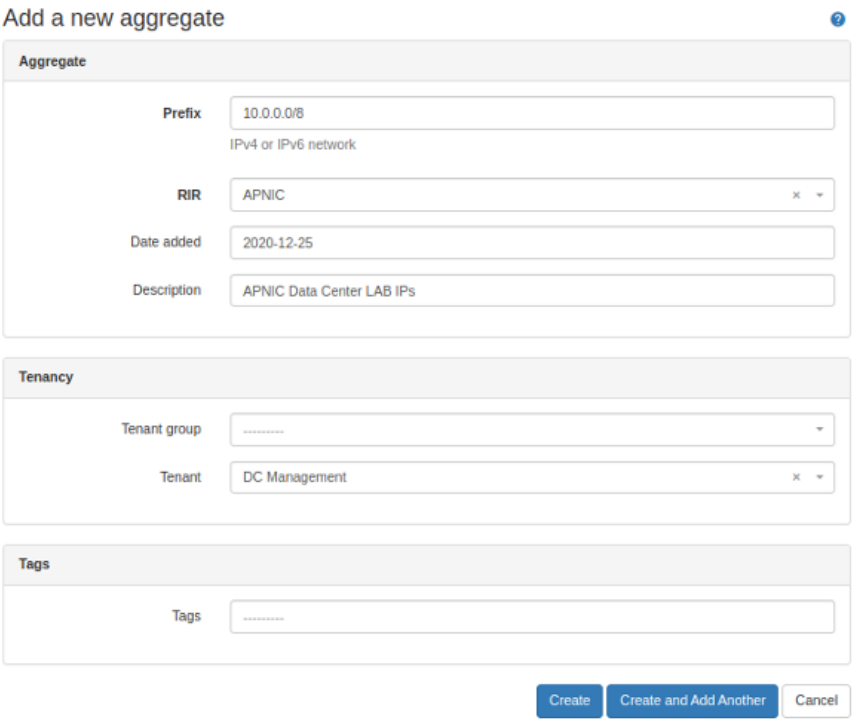
Create Aggregates:

First, select `RIRs` from the `IPAM` tab, and create new one with `APNIC`



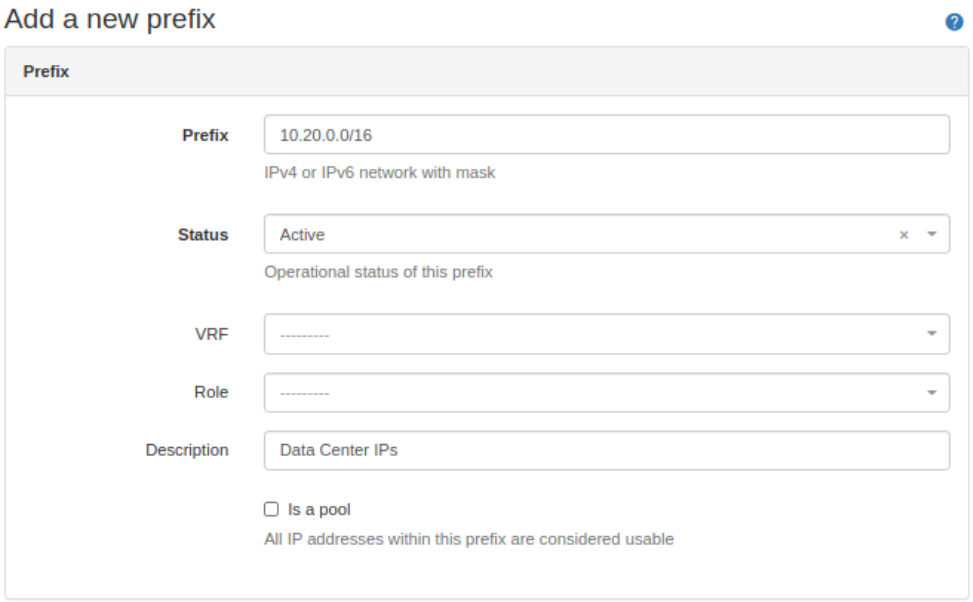


It will take you to a new window, where you have to create Aggregate IPs. Here we assume the IP Prefix is `10.0.0.0/8`.

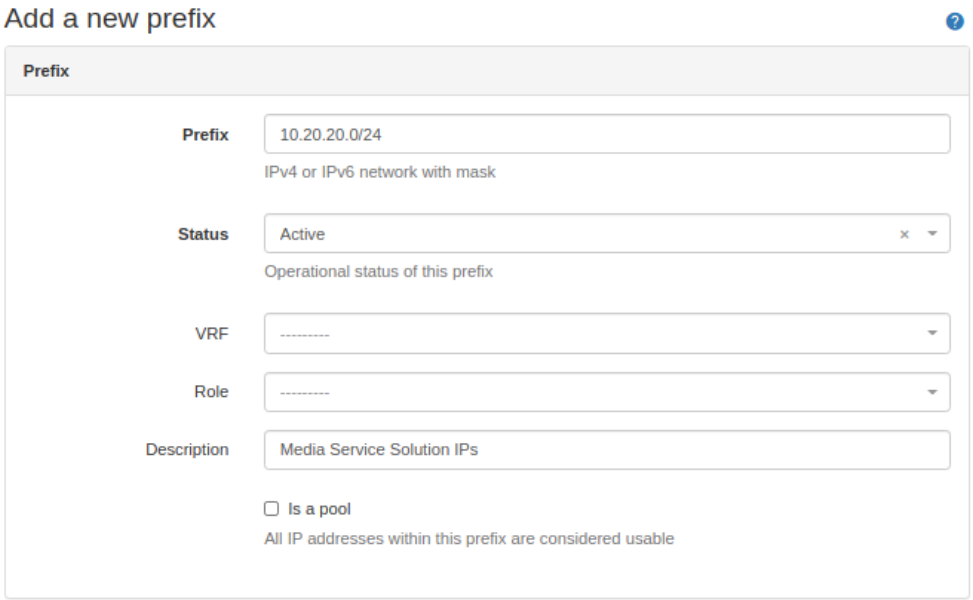


Create Prefixes

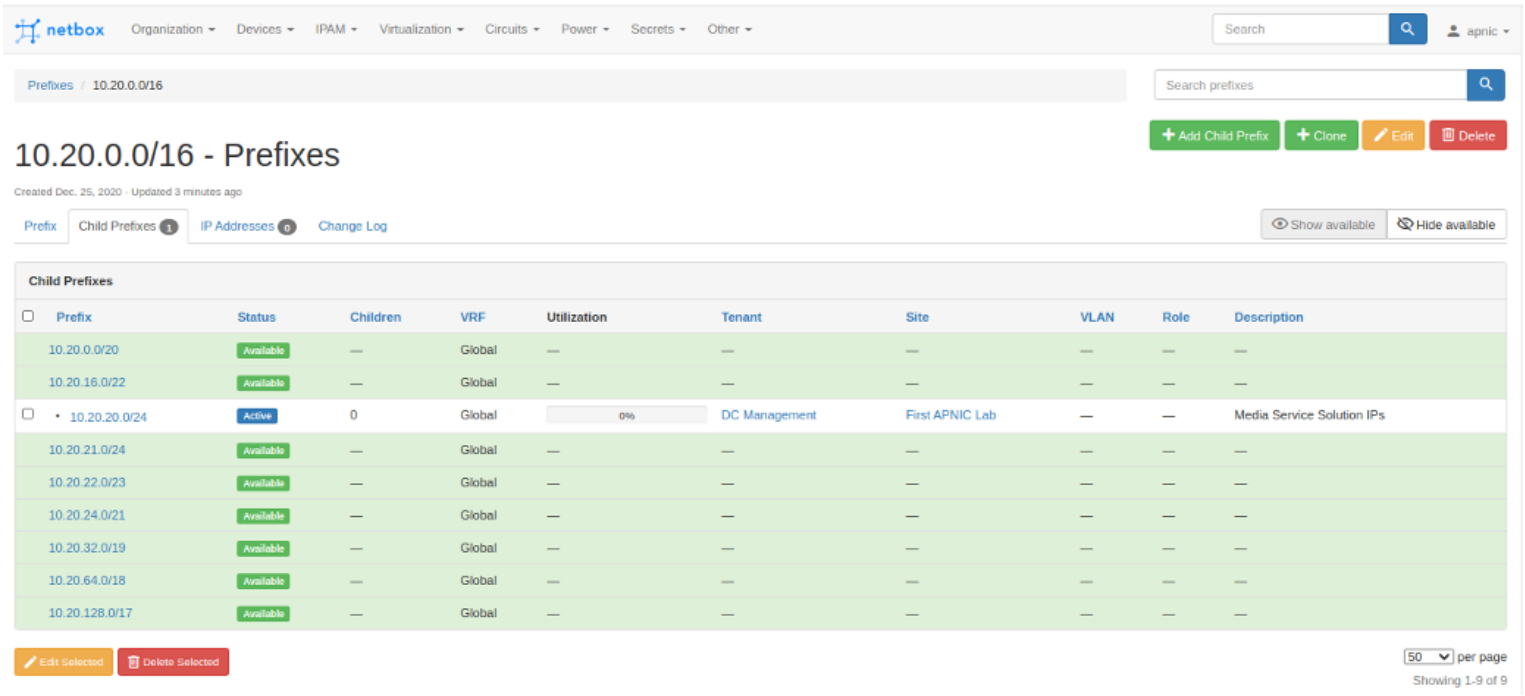
Select `Prefixes` from the `IPAM` tab to add new prefixes. we will use `10.20.0.0/16` as our prefix for the Data Center.



So the prefix window will showup, there you can see different sub-tab; go to the `Child Prefixes`, and create a new one `10.20.20.0/24` for `Media Service Solution`

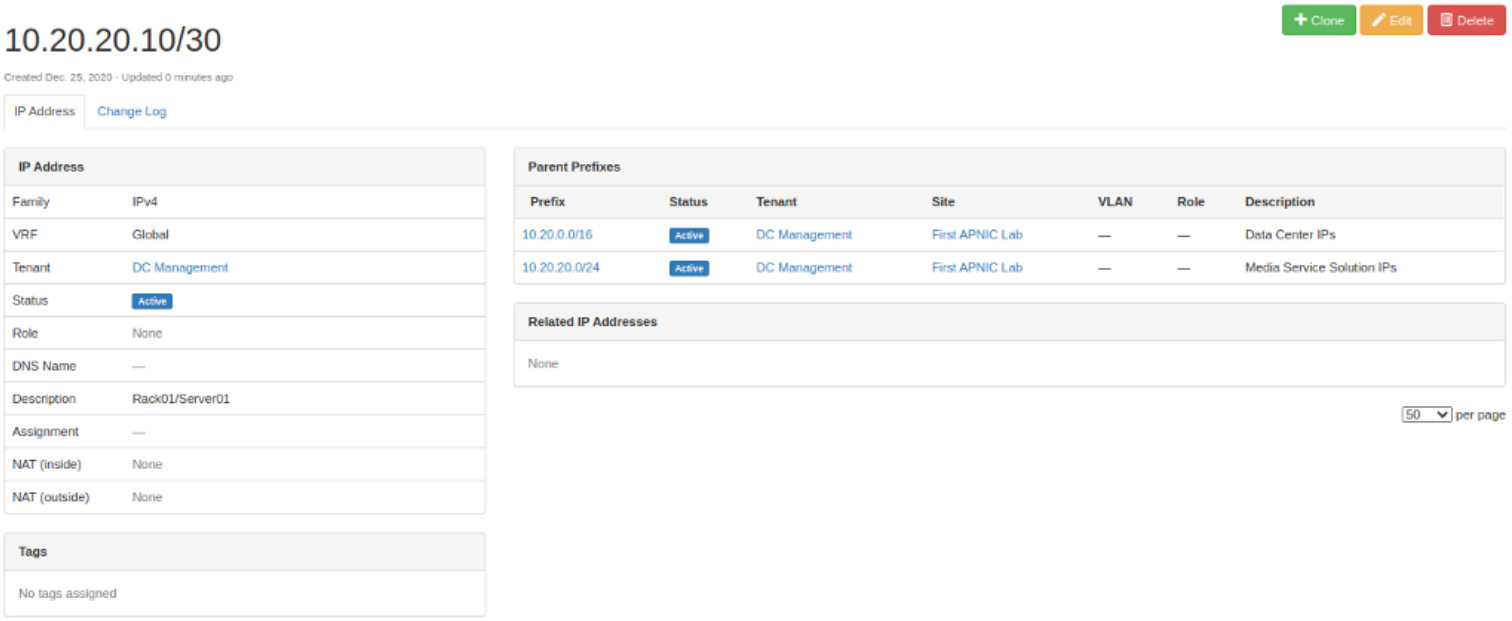


You will see the window like this.



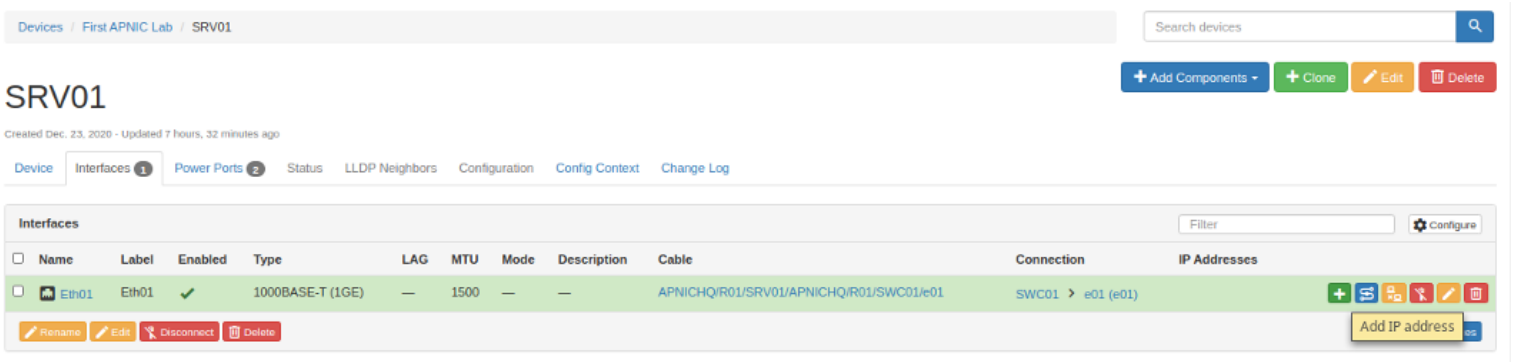
Click on the child-prefix that we just now defined.from the go to the `IP Addresses` sub-tab. And create a IP for the first server as `10.20.20.10/30` .

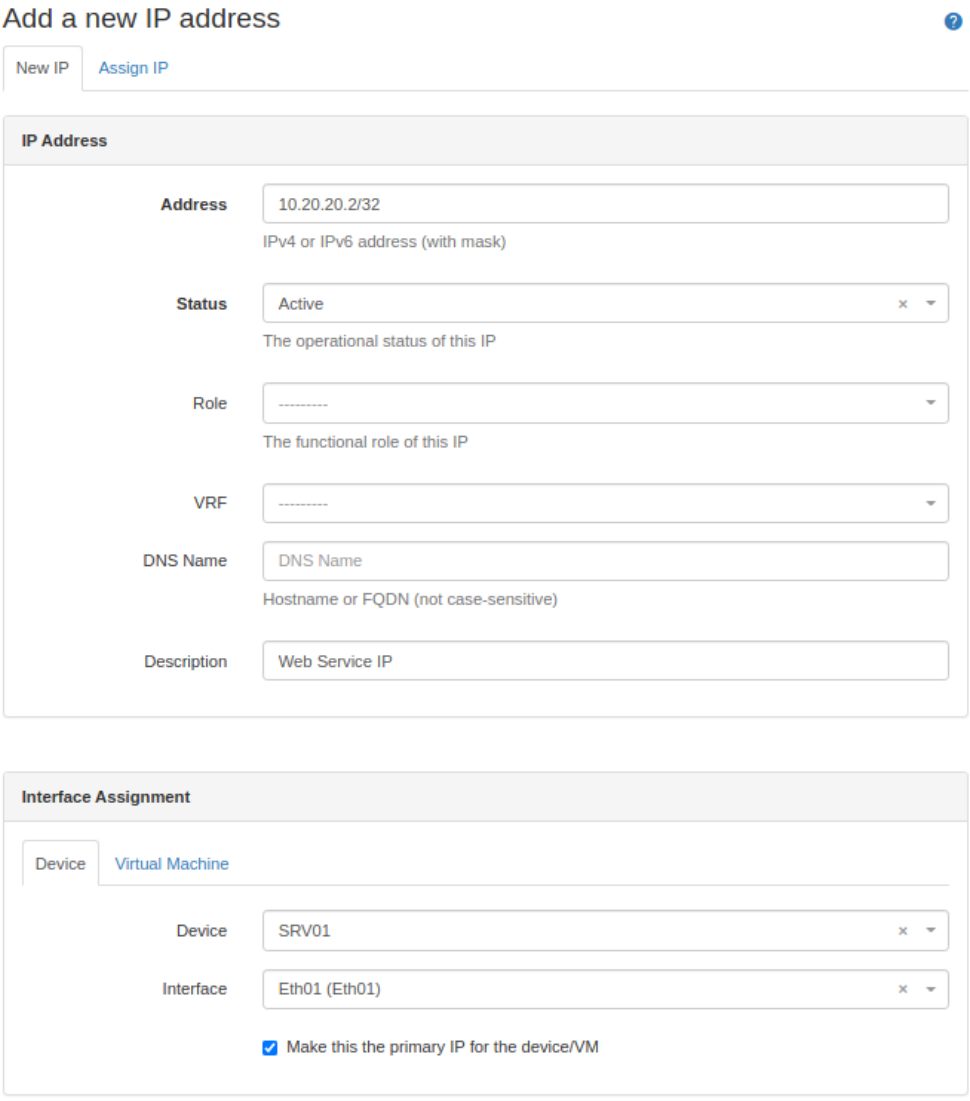
You will get a window like below to see the status.



Now, lets us go back to the DCIM module, and assign an IP to the server 01 interface.

To do that, select srv01 from the `Devices` lists, go to the sub-tab `Interfaces` and click on the `green + sign`, to add the IP address. It will take you to a new page, provide all the info accordingly.





So, as of now our one server is connected with a switch, and both the devices ar connected with the power source.

Exercise:

- Create another server
- Create NAS
- Connect server and NAS with switch
- Create Router, and connect with switch as gateway.
- Create the Circuits, and connect with Rotuer.

End of Lab
