GORILLA TECHNOLOGY - NETBOX

# NetBox - Installation

OVERVIEW

# Populating Data into NetBox

1. **Using the Python API for NetBox**

# NetBox - Installation

This guide will walk you through the process of setting up NetBox using Docker on your local machine. Ensure that you have Docker Community Edition (or compatible) and docker-compose installed. Git should also be installed on your system. This guide is primarily intended for Linux or macOS, but it may also work on Docker for Windows with a Linux backend or Windows Subsystem for Linux (WSL).

**Requirements:**

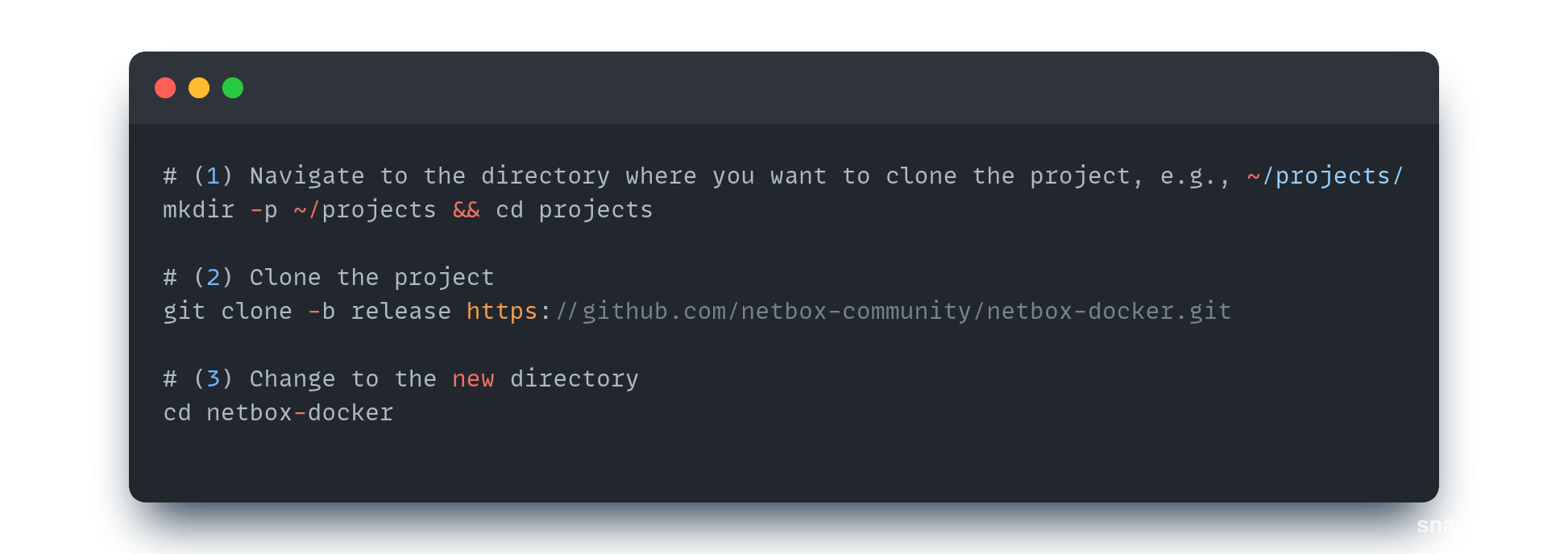
* Docker Community Edition or compatible
* docker-compose
* Git
* Linux or macOS (or Docker for Windows with Linux backend)

# Note:

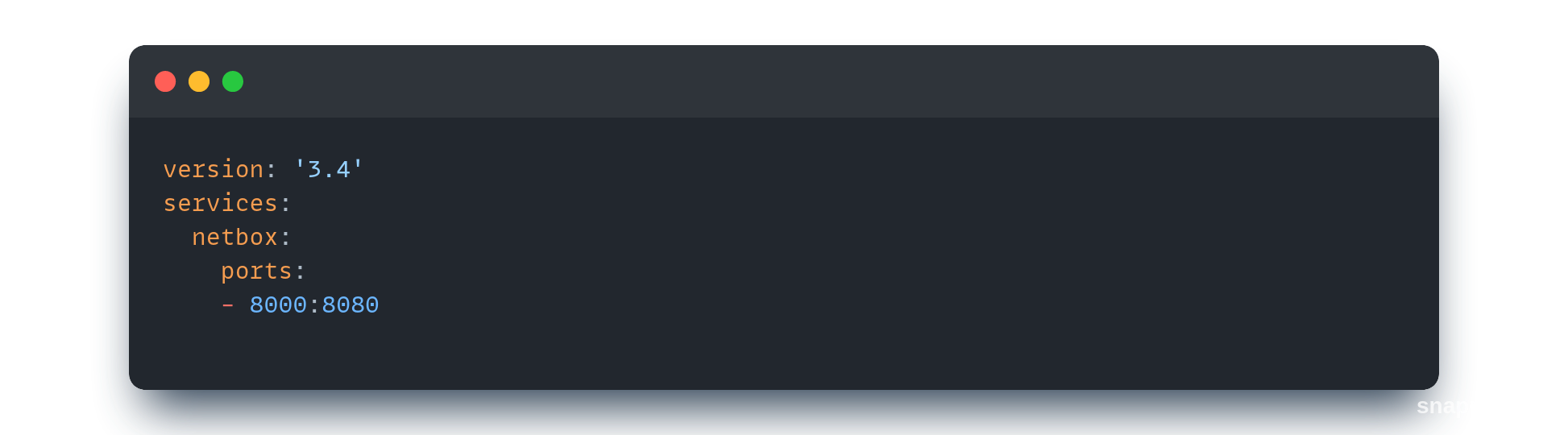
If you encounter permission issues while executing any of the commands mentioned above, you may need to use sudo to run them with elevated privileges.

**Reference:** [Click Here](https://github.com/netbox-community/netbox-docker/wiki/Getting-Started)

# Step 1: Clone the NetBox Docker Repository

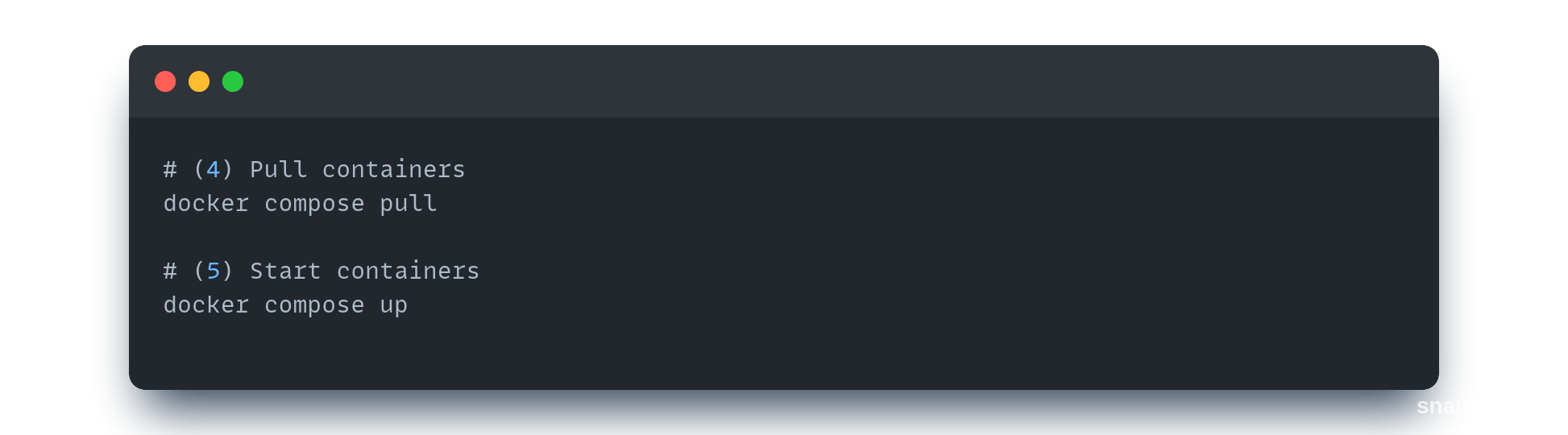
Open a terminal and follow these steps:

# Step 2: Create Docker Compose Override File

Create a new file named docker-compose.override.yml in the project directory with the following content:

This file ensures NetBox always listens on port 8000.

# Step 3: Pull Containers and Start NetBox

Run the following commands to pull containers from the Docker registry and start NetBox:

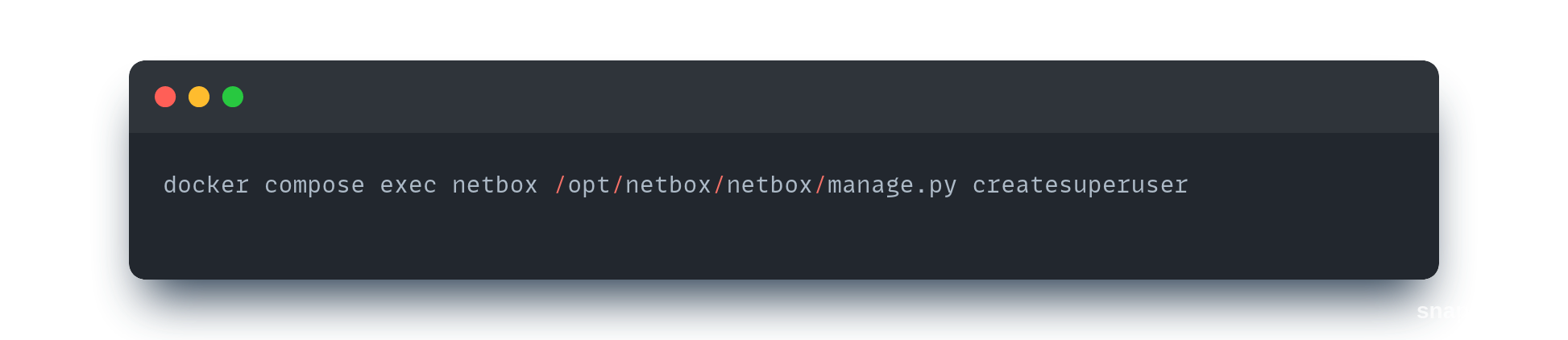
This process may take some time depending on your internet connection. NetBox will initialize the database and start.

# Step 4: Access NetBox

After a few minutes, NetBox will be available. Open http://0.0.0.0:8000/ in a web browser to access the NetBox homepage.

# Step 5: Create Admin User

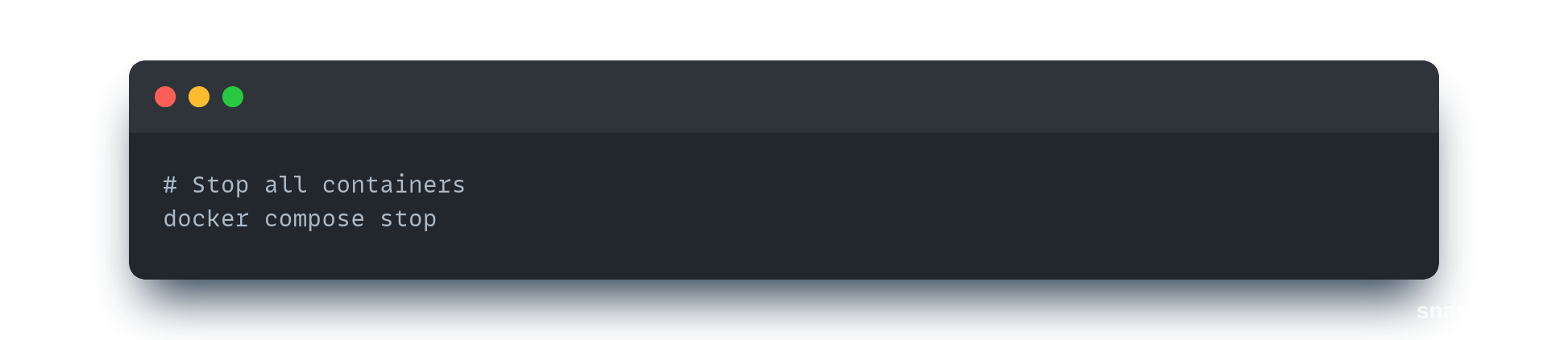
To create the first admin user, run:



# Shutdown:

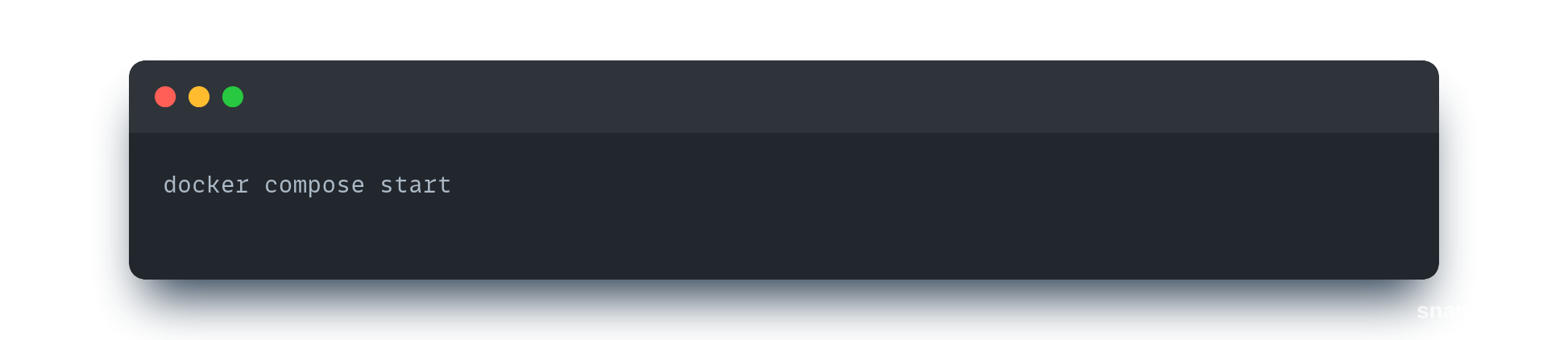
**Stop NetBox**

If you want to stop NetBox to continue work later, use the following command:

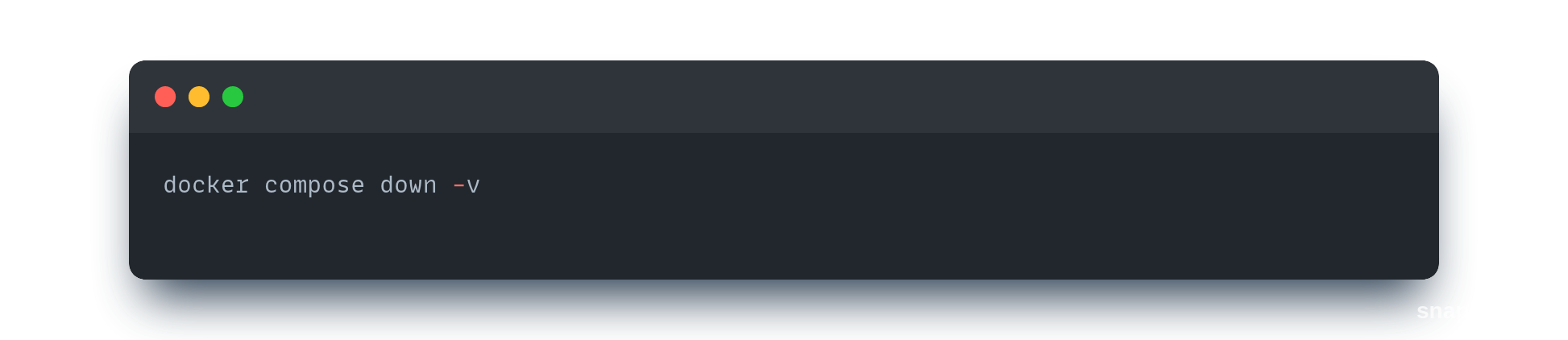


# Start NetBox Again

To start the containers again, use:



# Stop and Clean Up (Attention: Removes Data)

If you want to stop NetBox and clean up resources (database, files, etc.), use:

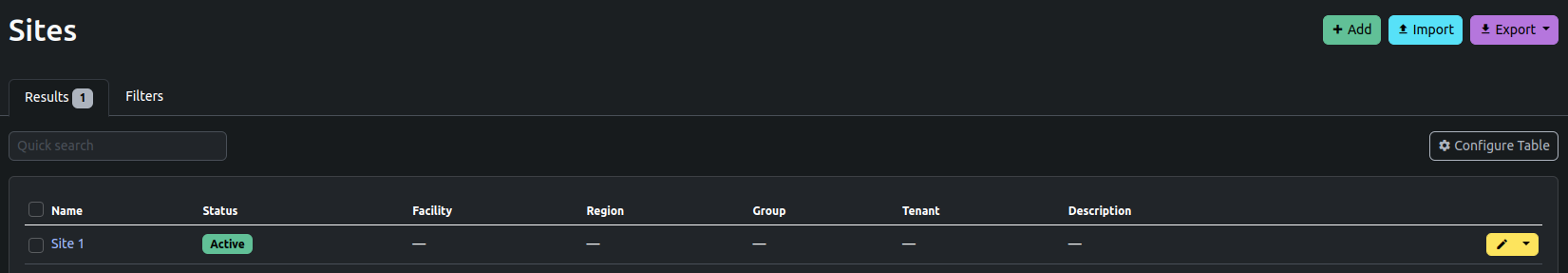
This command will remove any data you have entered into NetBox.

# Populating Data into NetBox

* 1. **Populating using the UI**

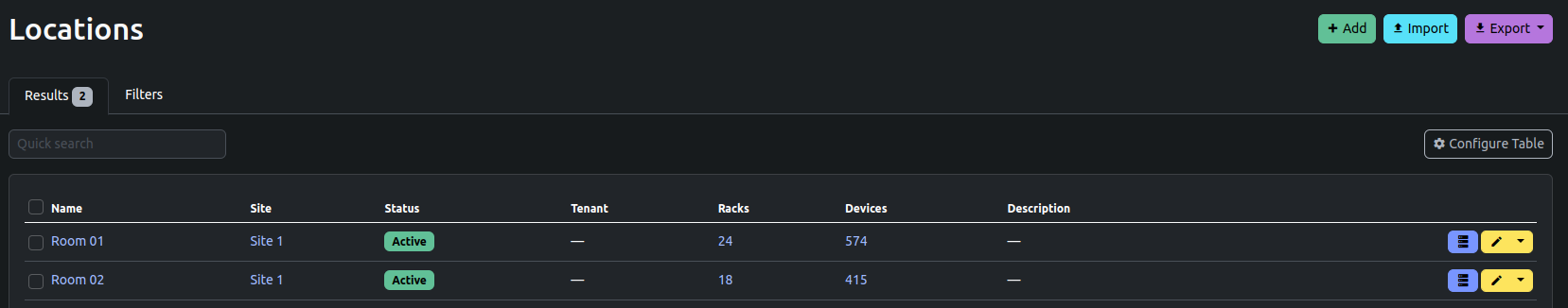
# Create a Site

Go to Sites under Organization and add a new site. Name it “Site 1” for example and provide an appropriate slug.



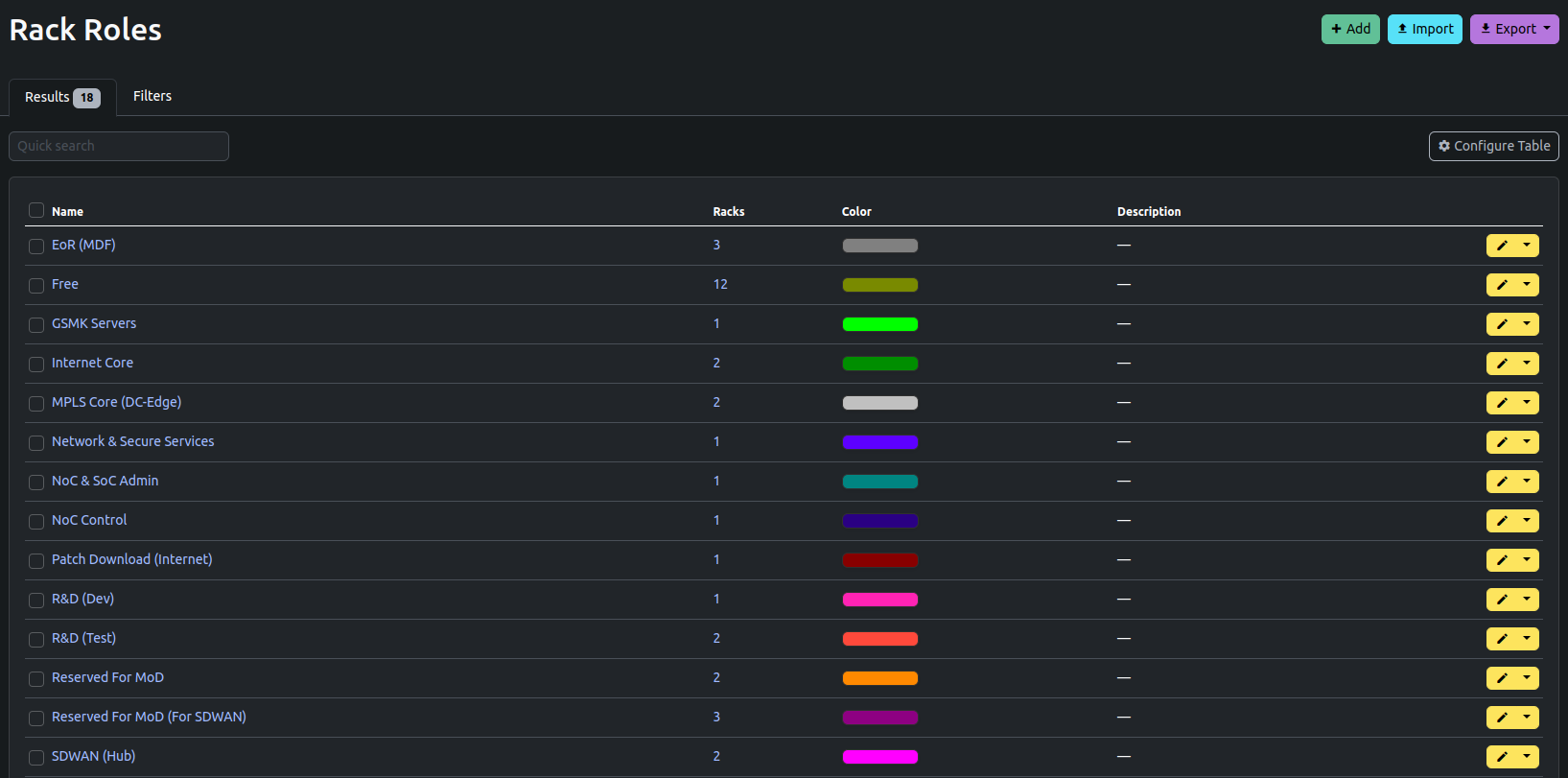
# Create Locations

A location can be a room, ﬂoor, cage, or other organizational unit. Go to Locations under Organization and add new locations. In our example, we have added “Room 01” and “Room 02”.



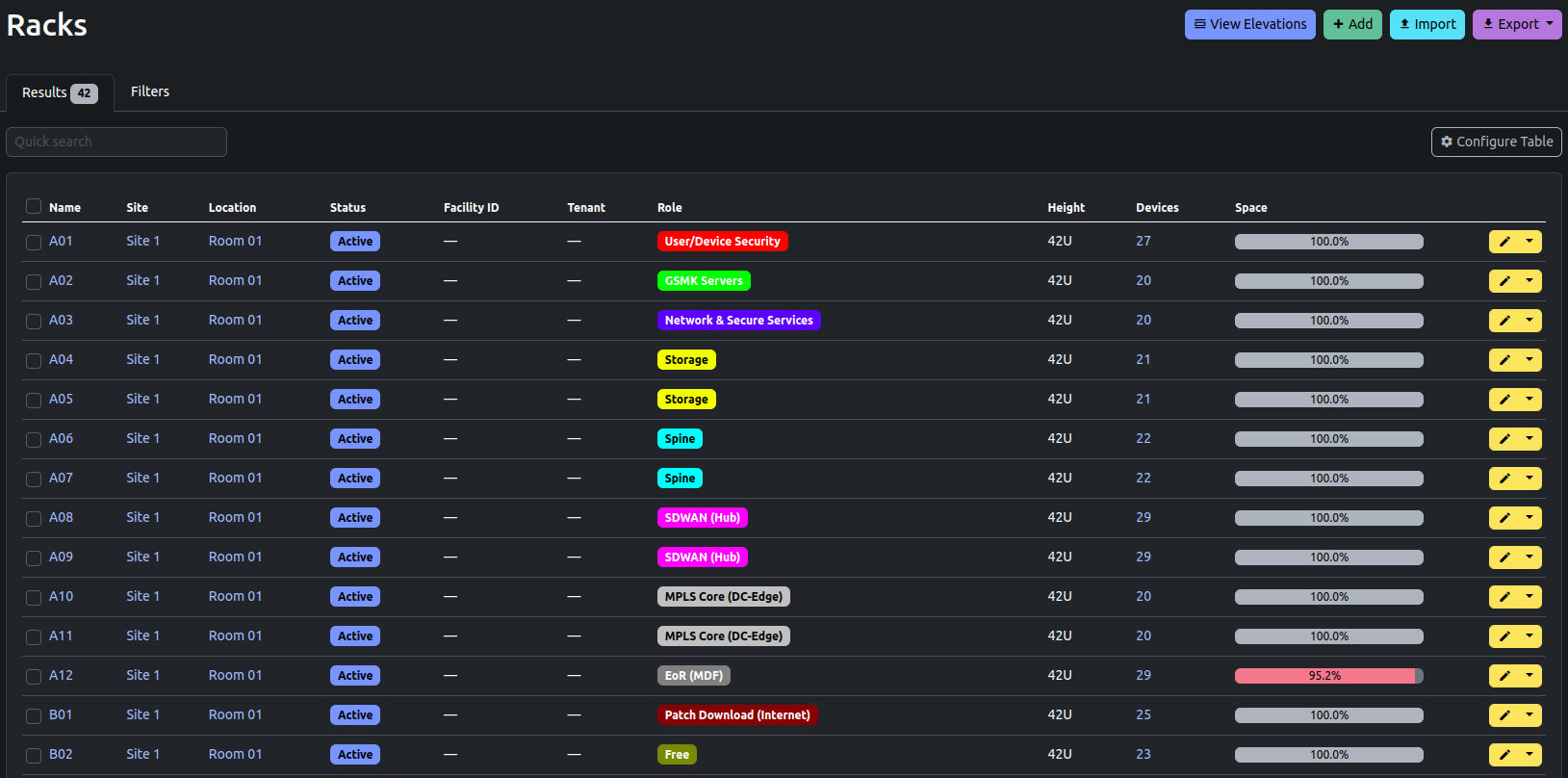
# Create Rack Roles

Define roles for each rack in your site and add all the various rack roles to NetBox. Provide appropriate colour coding. This is available under the Racks section of Organization. Ex: Security; Administration, etc.



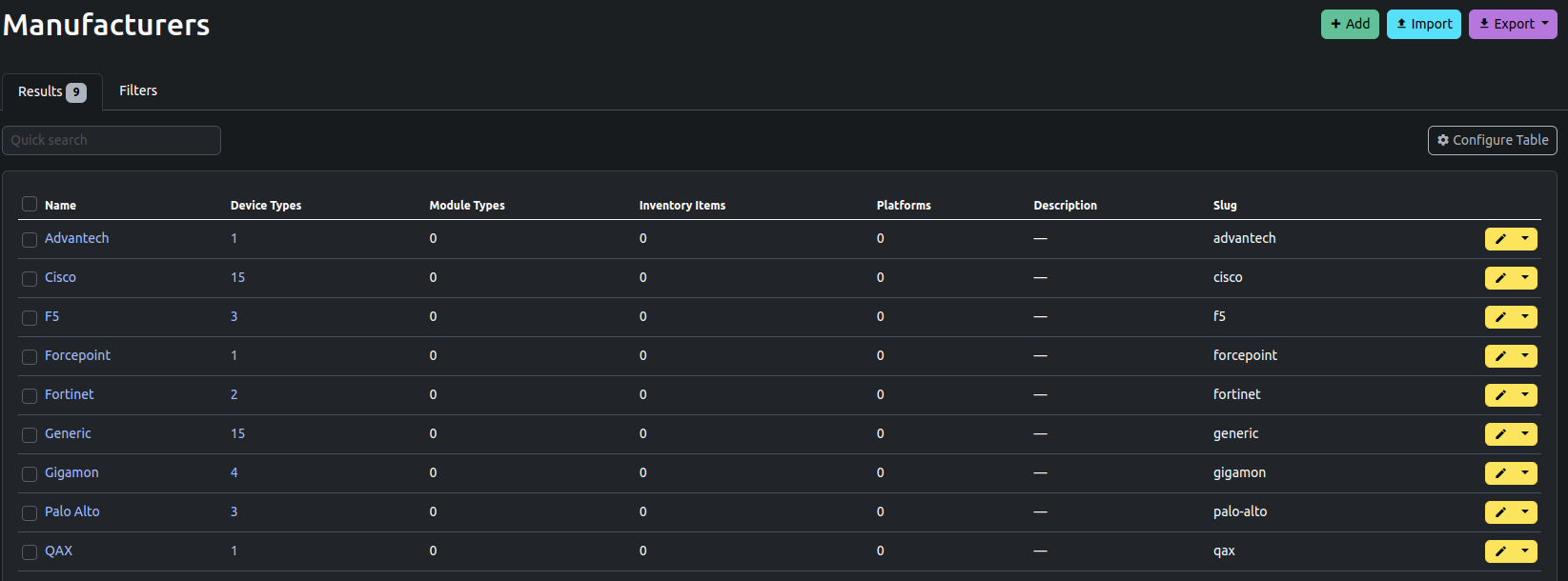
# Create Racks

Add racks to the desired rooms/locations in your site and name them appropriately.



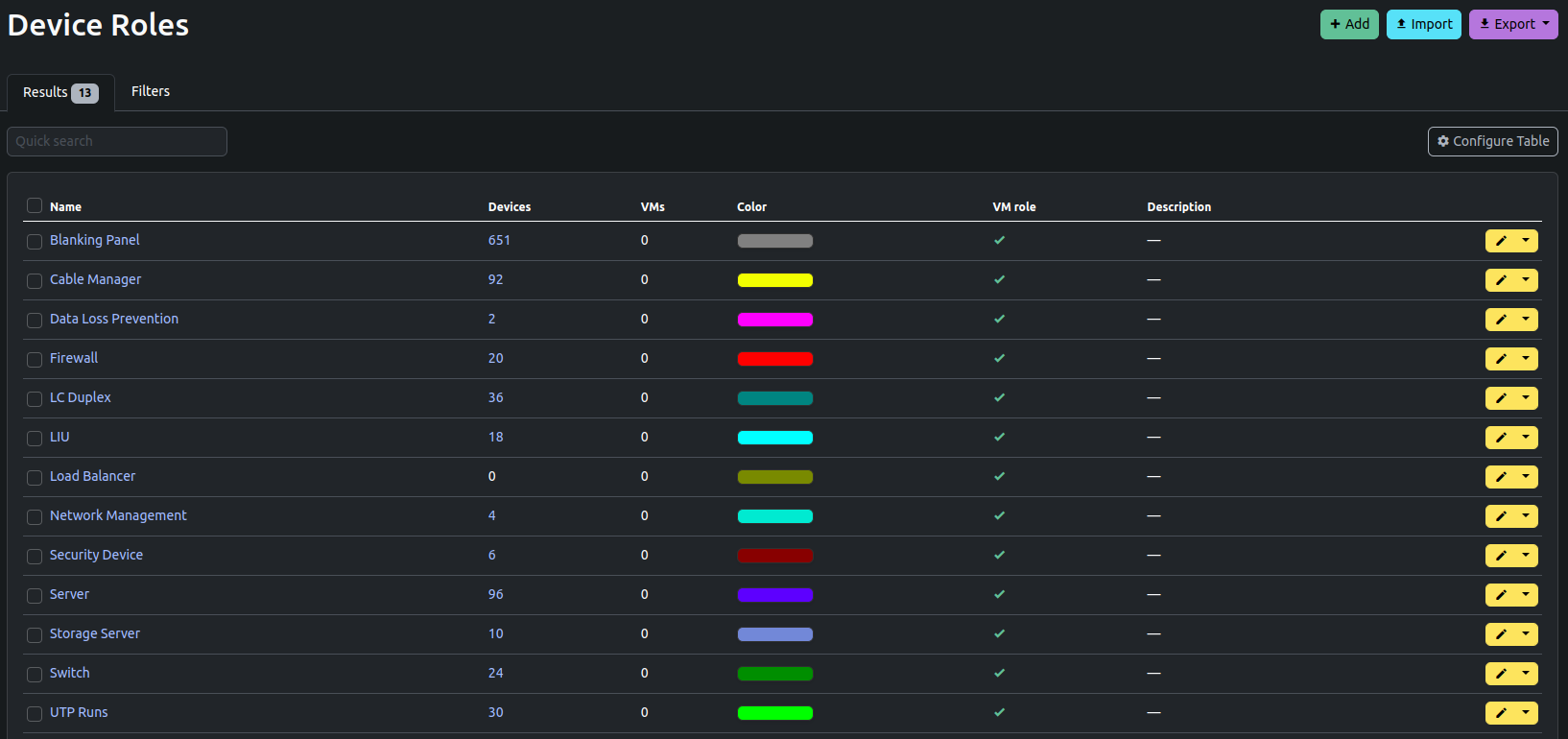
**- Add Manufacturers**

Go to Manufacturers under Devices and add details of the different manufacturers of your devices. Ex: Cisco, Palo Alto, etc.



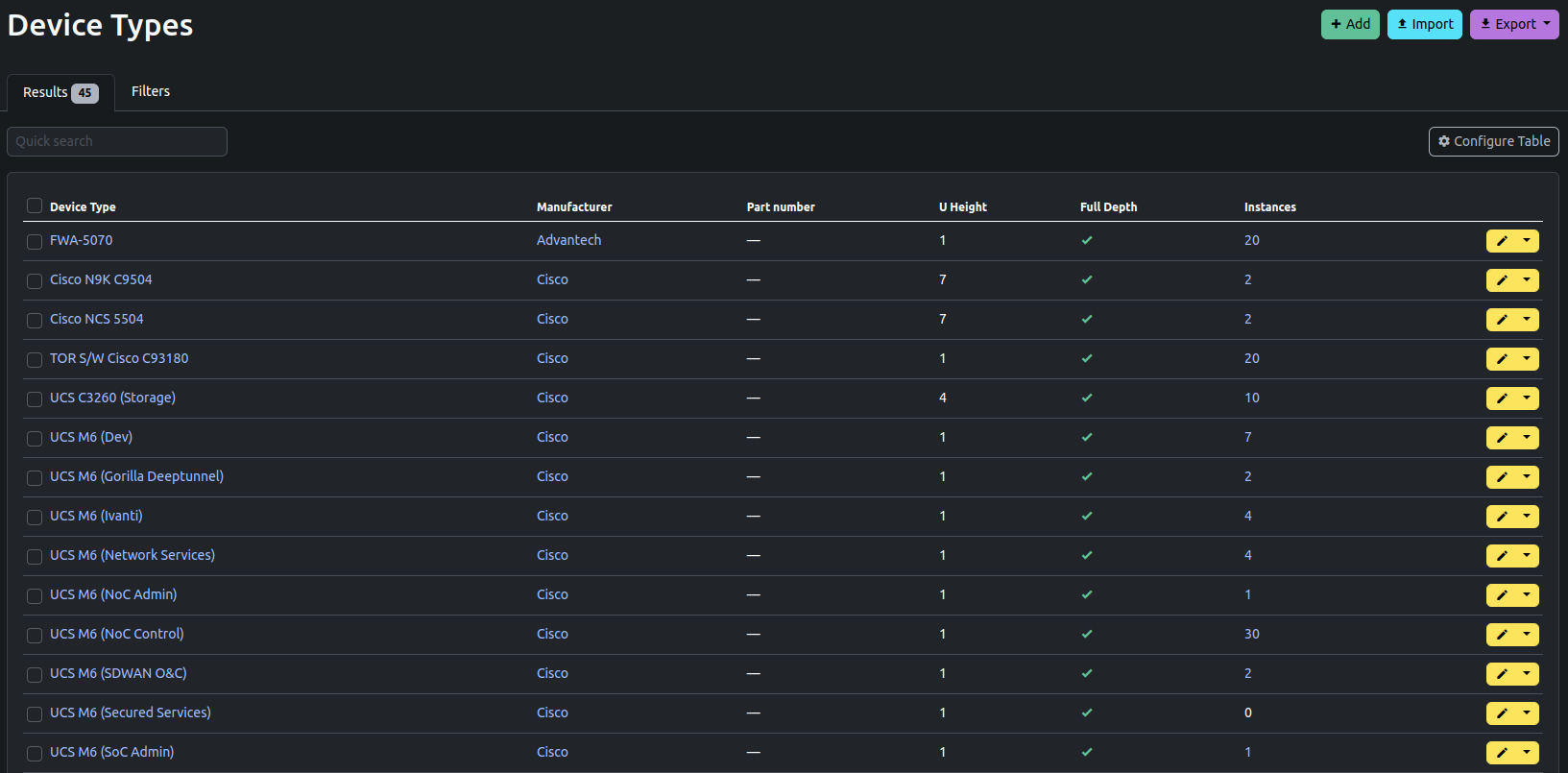
# Create Device Roles

Identify the various roles of devices in all the racks of your site and then add Device Roles under Devices. Define appropriate colour coding for your various device roles. Ex: Server - Blue; Switch - green, etc.



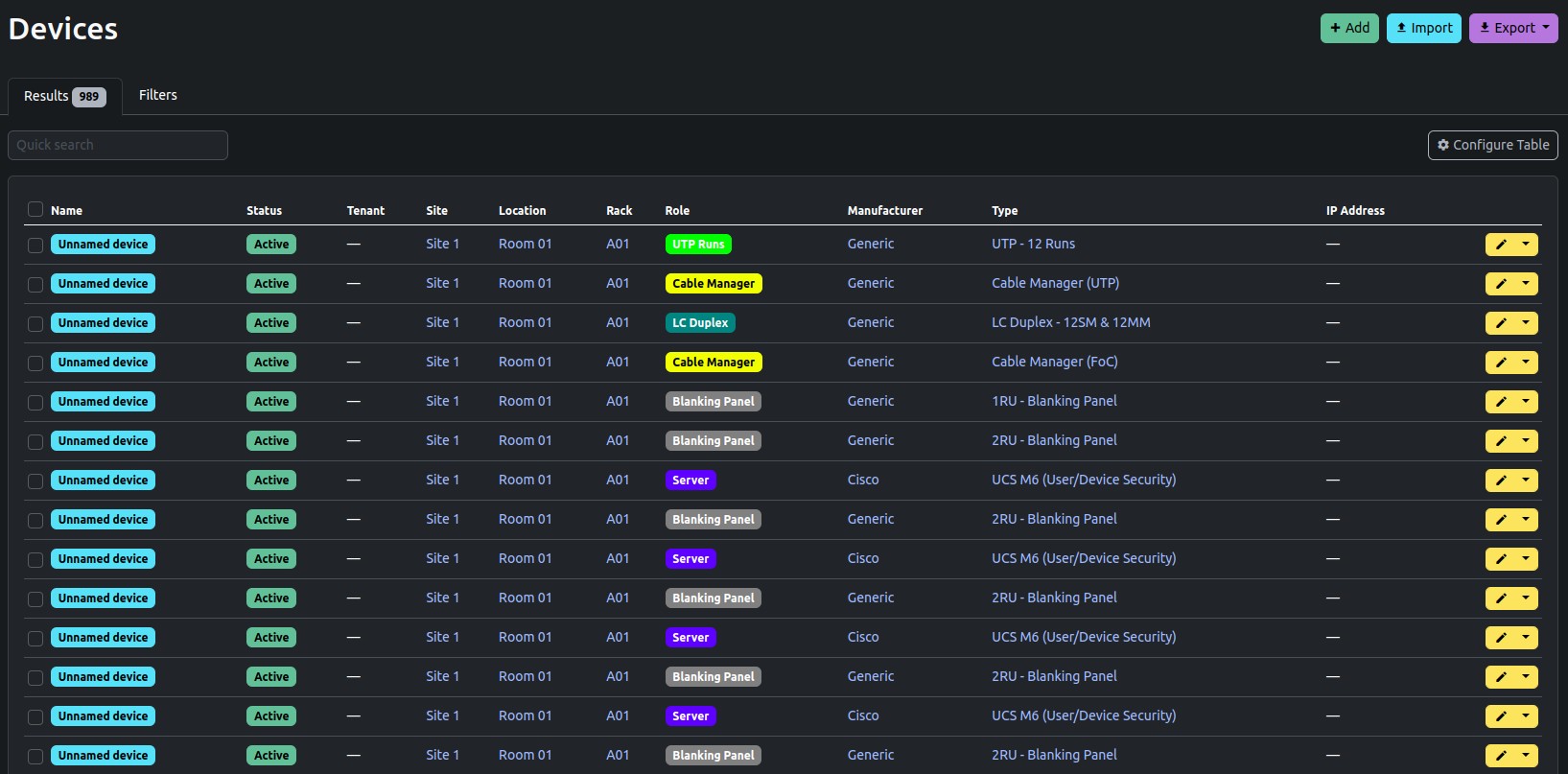
# Add Device Types

Go to Device Types under Devices and add the various device types among the devices in all the racks of your site. While adding a new device type, specify a particular device role and manufacturer from the ones you have already created. Also specify the number of rack units that each device type takes. All devices of the same device type should take up the same amount of rack space. Ex: UCS M6 Server, 2RU Blanking Panel, LC Duplex - 12SM & 12MM, etc.



# Add Devices

Go to Devices and add the various devices present in the racks of your site. While adding a device, specify the device type, device role, the rack that it belongs to, the position in the rack that it belongs to, the location and site that it belongs to, and the face of the rack that it occupies (front/rear). Specify a unique name for the device (optional; if not specified, its name is “unnamed device”).



# Populating through imports

* Follow the same order of population of data as specified in 2.2.
* For each section, use the import button to import CSV/JSON/YAML data in bulk.
* Sample data for each section is provided in the “NetBox-Data” Spreadsheet.
* Download the respective section’s sheet as .csv, and import it into NetBox to populate NetBox with data.

# Using the Python API for NetBox

**Python API: Pynetbox**

# Pynetbox Installation

* To install, run:

pip install pynetbox

* Alternatively, you can clone [this repository](https://github.com/netbox-community/pynetbox) and run:

python setup.py install

# Create NetBox API Key:

* Under the Admin tab in NetBox, go to API Tokens and add a new Token. The API Key field will be automatically filled.
* Select the User for which the API Key applies, and check the “Write enabled” box if you want to allow write permissions.

# Using the API

* Once Pynetbox has been installed, create a .py file and instantiate the API using the following code:

import pynetbox nb = pynetbox.api(

'http://localhost:8000',

token = <Your NetBox API Token>

)

* Ensure to replace <Your NetBox API Token> with your actual NetBox API Key **within quotes**.
* The first argument the .api() method takes is the NetBox URL.
* There are a handful of named arguments you can provide, but in most cases none are required to simply pull data.
* In order to write, the token argument must be provided.

>>> device\_roles = nb.dcim.device\_roles.all()

>>> for role in device\_roles:

... print(role.name)

...

Blanking Panel Cable Manager

Data Loss Prevention Firewall

LC Duplex LIU

Load Balancer Network Management Security Device Server

Storage Server Switch

UTP Runs

>>>