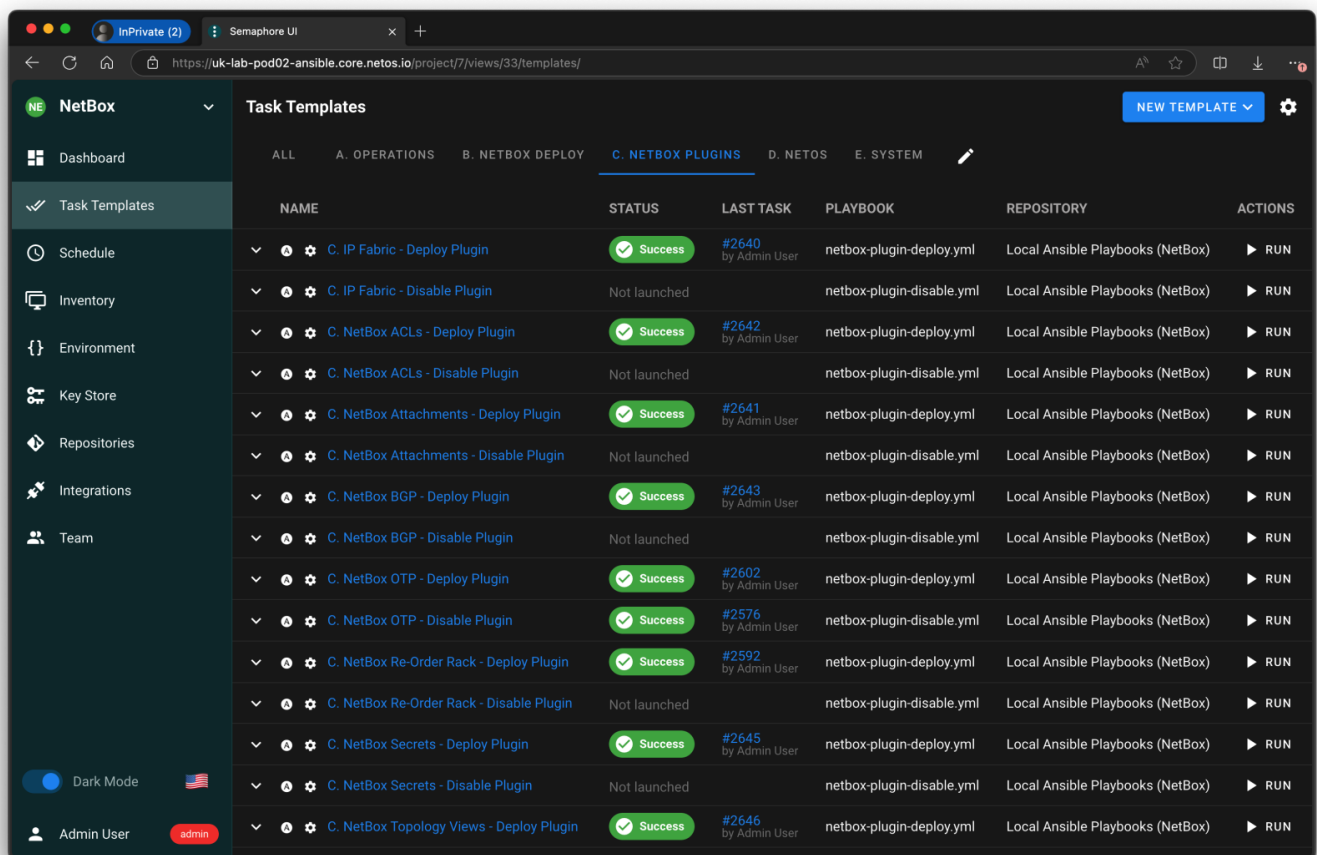


Deploy NetBox Community Plugins

This feature is a work-in-progress. Testing has been completed deploying the plugins in the screenshot to NetBox `3.7.8`, and NetBox `4.0.11`, and also upgrading NetBox, and then re-deploying using this tool.

Introduction

Deploy, manage, and upgrade NetBox community plugins from within Semaphore:



Task Templates							NEW TEMPLATE	⚙
ALL A. OPERATIONS B. NETBOX DEPLOY C. NETBOX PLUGINS D. NETOS E. SYSTEM								
	NAME	STATUS	LAST TASK	PLAYBOOK	REPOSITORY	ACTIONS		
✓	⚙ C. IP Fabric - Deploy Plugin	Success	#2640 by Admin User	netbox-plugin-deploy.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. IP Fabric - Disable Plugin	Not launched		netbox-plugin-disable.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. NetBox ACLs - Deploy Plugin	Success	#2642 by Admin User	netbox-plugin-deploy.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. NetBox ACLs - Disable Plugin	Not launched		netbox-plugin-disable.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. NetBox Attachments - Deploy Plugin	Success	#2641 by Admin User	netbox-plugin-deploy.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. NetBox Attachments - Disable Plugin	Not launched		netbox-plugin-disable.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. NetBox BGP - Deploy Plugin	Success	#2643 by Admin User	netbox-plugin-deploy.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. NetBox BGP - Disable Plugin	Not launched		netbox-plugin-disable.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. NetBox OTP - Deploy Plugin	Success	#2602 by Admin User	netbox-plugin-deploy.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. NetBox OTP - Disable Plugin	Success	#2576 by Admin User	netbox-plugin-disable.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. NetBox Re-Order Rack - Deploy Plugin	Success	#2592 by Admin User	netbox-plugin-deploy.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. NetBox Re-Order Rack - Disable Plugin	Not launched		netbox-plugin-disable.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. NetBox Secrets - Deploy Plugin	Success	#2645 by Admin User	netbox-plugin-deploy.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. NetBox Secrets - Disable Plugin	Not launched		netbox-plugin-disable.yml	Local Ansible Playbooks (NetBox)	▶ RUN		
✓	⚙ C. NetBox Topology Views - Deploy Plugin	Success	#2646 by Admin User	netbox-plugin-deploy.yml	Local Ansible Playbooks (NetBox)	▶ RUN		

You can queue up tasks (providing `concurrency=1` in the Semaphore project/dashboard settings, in order to avoid having multiple tasks running at the same time):

The screenshot shows the NetBox web interface with the 'Task Templates' section selected. The left sidebar contains navigation links: Dashboard, Task Templates, Schedule, Inventory, Environment, Key Store, Repositories, Integrations, and Team. The main content area displays a table of task templates under the 'C. NETBOX PLUGINS' tab. The table has columns for NAME, VERSION, STATUS, LAST TASK, PLAYBOOK, INVENTORY, and ENVIRONMENT. The tasks listed include various plugins like IP Fabric, ACLs, Attachments, BGP, OTP, Re-Order Rack, Secrets, and Topology Views, each with a 'Deploy Plugin' and a 'Disable Plugin' variant. The status of these tasks varies, with some being 'Waiting', 'Not launched', 'Success', or 'Running'.

NAME	VERSION	STATUS	LAST TASK	PLAYBOOK	INVENTORY	ENVIRONMENT
✓ C. IP Fabric - Deploy Plugin	—	Waiting	#2563 by Admin User	netbox-plugin-deploy.yml	Localhost	NetBox Community
✓ C. IP Fabric - Disable Plugin	—	Not launched		netbox-plugin-disable.yml	Localhost	NetBox Community
✓ C. NetBox ACLs - Deploy Plugin	—	Waiting	#2564 by Admin User	netbox-plugin-deploy.yml	Localhost	NetBox Community
✓ C. NetBox ACLs - Disable Plugin	—	Not launched		netbox-plugin-disable.yml	Localhost	NetBox Community
✓ C. NetBox Attachments - Deploy Plugin	—	Waiting	#2565 by Admin User	netbox-plugin-deploy.yml	Localhost	NetBox Community
✓ C. NetBox Attachments - Disable Plugin	—	Not launched		netbox-plugin-disable.yml	Localhost	NetBox Community
✓ C. NetBox BGP - Deploy Plugin	—	Waiting	#2566 by Admin User	netbox-plugin-deploy.yml	Localhost	NetBox Community
✓ C. NetBox BGP - Disable Plugin	—	Not launched		netbox-plugin-disable.yml	Localhost	NetBox Community
✓ C. NetBox OTP - Deploy Plugin	—	Success	#2567 by Admin User	netbox-plugin-deploy.yml	Localhost	NetBox Community
✓ C. NetBox OTP - Disable Plugin	—	Not launched		netbox-plugin-disable.yml	Localhost	NetBox Community
✓ C. NetBox Re-Order Rack - Deploy Plugin	—	Waiting	#2568 by Admin User	netbox-plugin-deploy.yml	Localhost	NetBox Community
✓ C. NetBox Re-Order Rack - Disable Plugin	—	Not launched		netbox-plugin-disable.yml	Localhost	NetBox Community
✓ C. NetBox Secrets - Deploy Plugin	—	Running	#2569 by Admin User	netbox-plugin-deploy.yml	Localhost	NetBox Community
✓ C. NetBox Secrets - Disable Plugin	—	Not launched		netbox-plugin-disable.yml	Localhost	NetBox Community
✓ C. NetBox Topology Views - Deploy Plugin	—	Waiting	#2570 by Admin User	netbox-plugin-deploy.yml	Localhost	NetBox Community

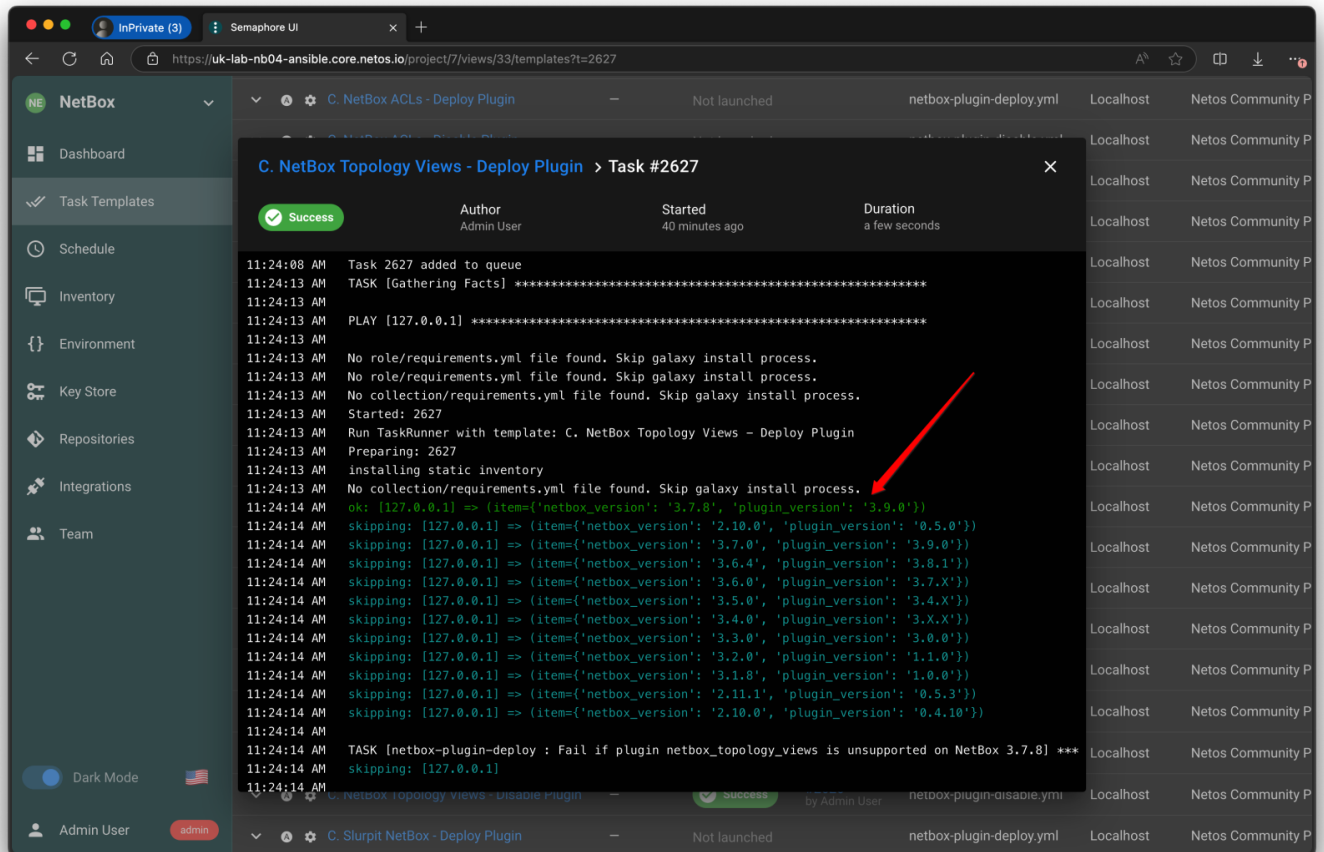
Plugin Mapping Tables

Version Selection

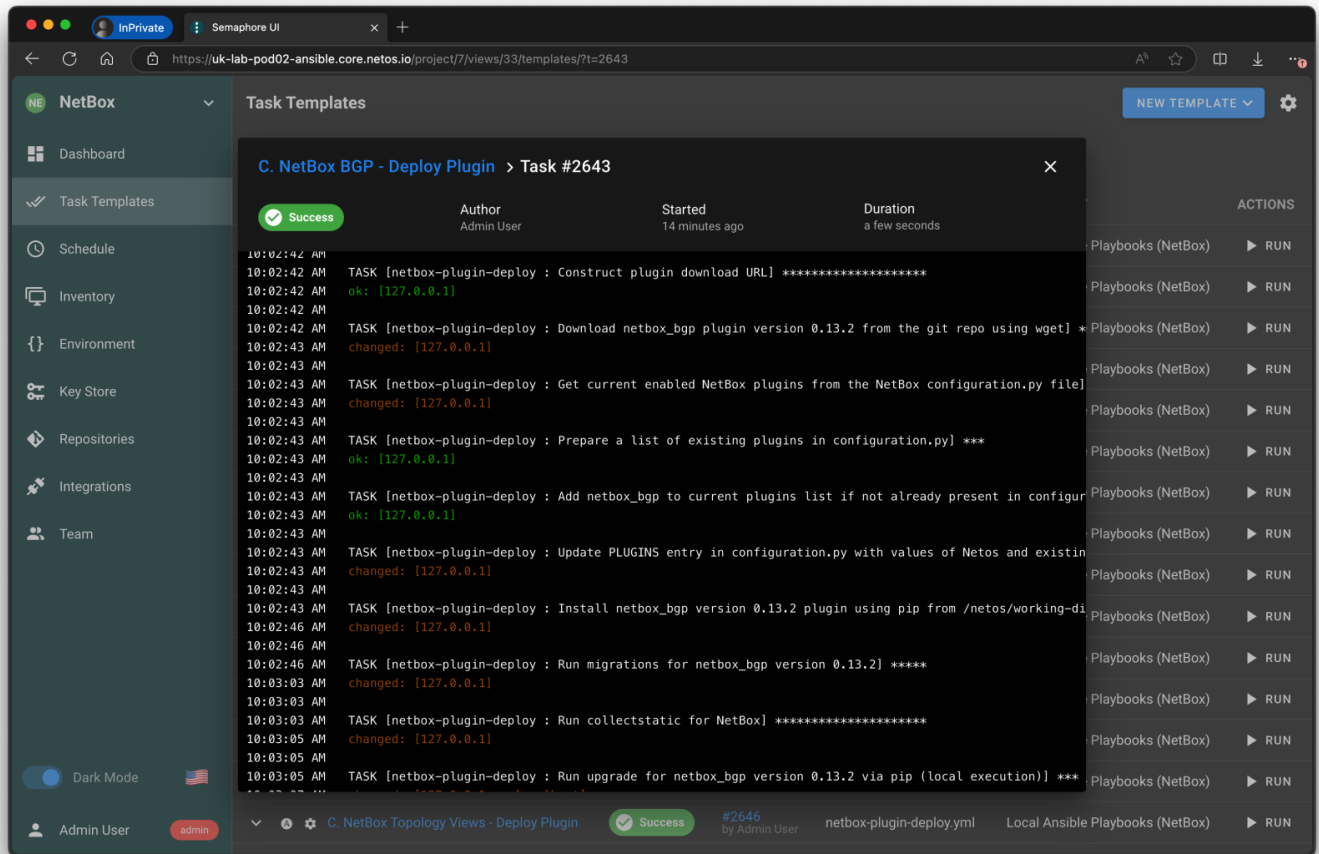
Similar to the mapping logic during the [NetBox installation process](#), each plugin has a defined mapping table that maps the plugin version to the supported NetBox version. For example, in this playbook, the logic is as follows:

1. Get the current NetBox version from the NetBox API `/api/status/`
2. Query the plugin-specific mapping table, in this case for `netbox_topology_views`
3. Check the **NetBox Version** to **Plugin Version** mapping, and if there is a match, proceed with the installation.

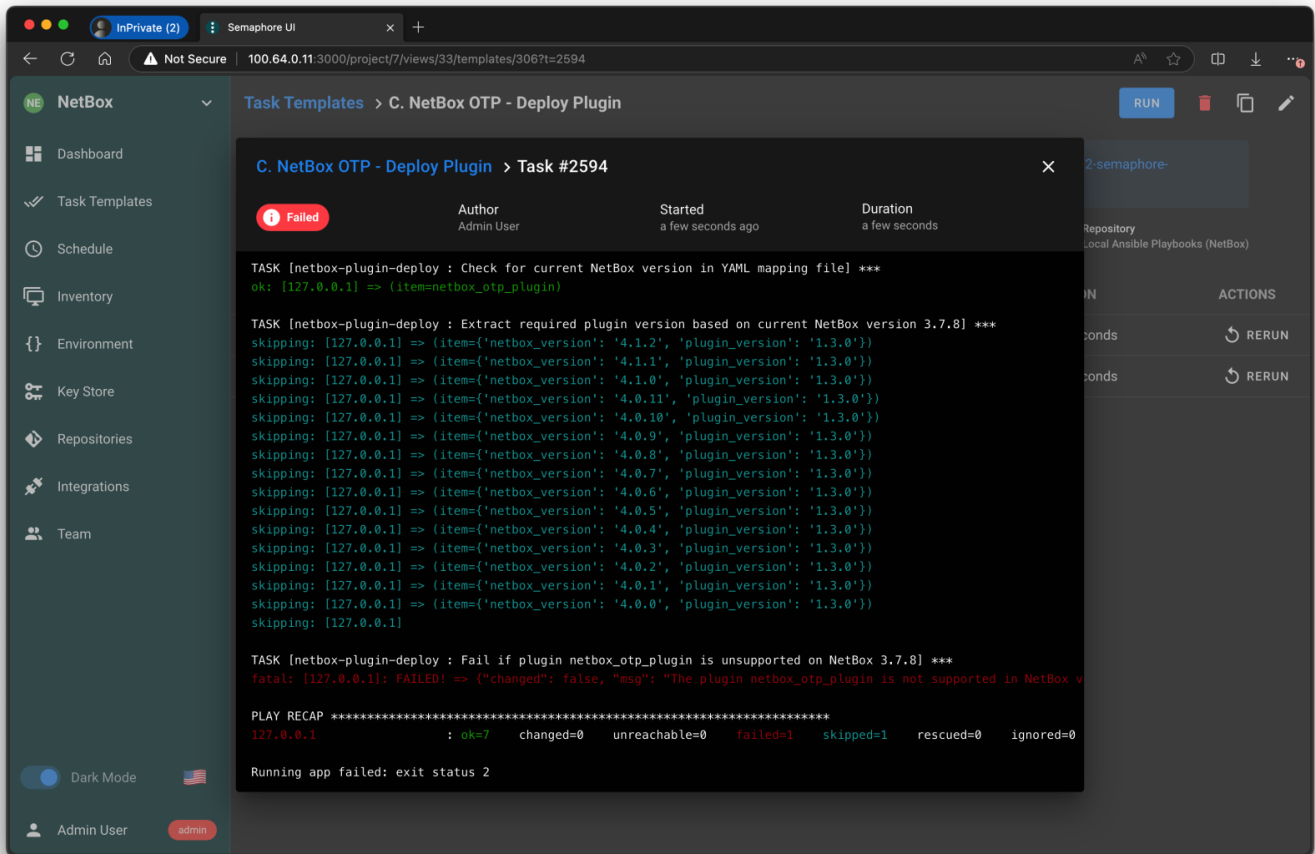
Here you can see that we are running NetBox `3.7.8` and that the playbook has detected that the suitable plugin version is `3.9.0`.



The script then proceeds to download the plugin (to `/netos/working-dir/netbox-plugins`) and install it.



If a match isn't found, i.e. the plugin is not supported on that version of NetBox (according to the mapping table) the installation will fail like this:



Mapping File Example

Each plugin has a mapping file similar to this in the `netos-netbox` repository. They are manually constructed based on the individual plugin recommendations to ensure that for each NetBox version the latest plugin is installed.

The `git_url`, `version_prefix`, and `file_type` variables are used to construct the URL to pull the plugin from the repo, as URL structures vary between plugins.

```
netbox_topology_views:
  name: "NetBox Topology Views"
  description: "A NetBox plugin that provides a graphical topology view for devices in NetBox."
  git_url: "https://github.com/netbox-community/netbox-topology-views"
  version_prefix: "v"
  file_type: "zip"
  version_mapping:
    - netbox_version: "4.1.0"
      plugin_version: "4.1.0"

    - netbox_version: "4.0.0"
      plugin_version: "4.0.0"
```

```
- netbox_version: "3.7.8"
  plugin_version: "3.9.0"
```

Every NetBox version is mapped to a specific plugin version based on the plugin documentation.

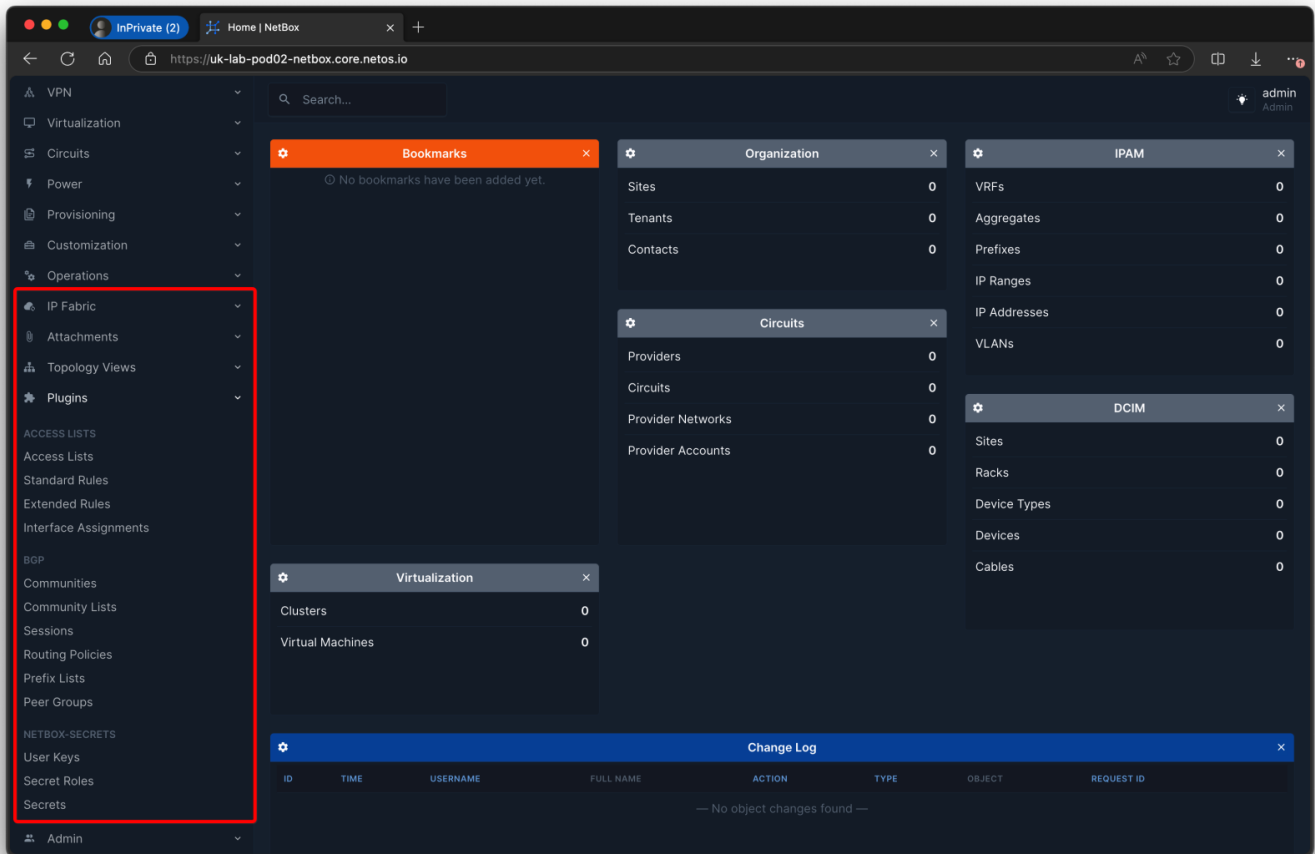
Updating Configuration.py

Plugins must be registered in `configuration.py`, for example; `PLUGINS = ['netbox_topology_views']`. The playbook will check if the plugin name is included, and if not, add it.

If there is an issue with the plugin, it can cause NetBox to fail to start, i.e. you'll get an NGINX 502 error. If this happens you can use the [Health Check](#) script to see if there is anything obvious in the logs. You can also [Disable NetBox Plugins](#) to remove the plugin entry from `configuration.py`.

Post Installation

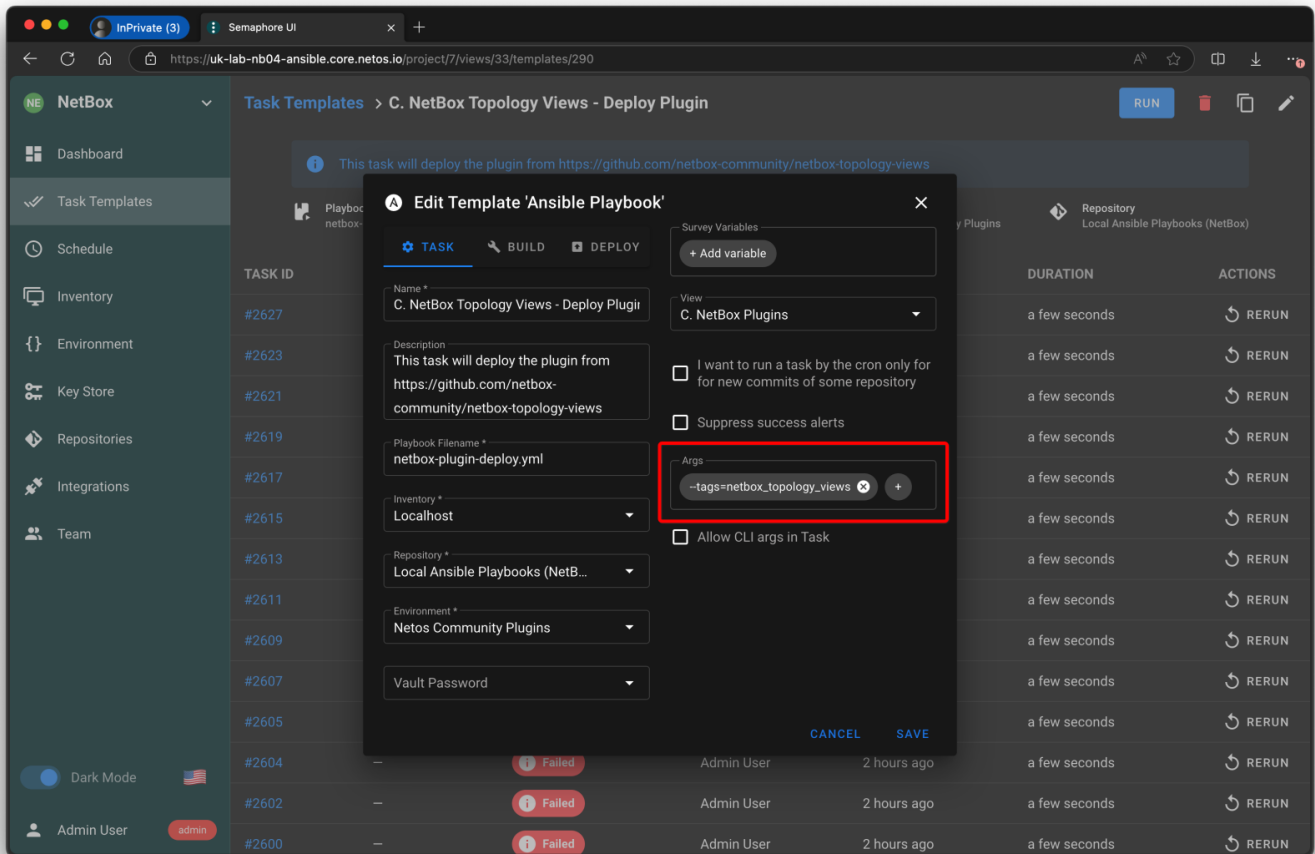
After running the playbooks in the screenshot at the top of this guide, we can see that the plugins have been installed in NetBox:



Add New Plugins to Semaphore

The process for adding a plugin to deploy through Semaphore is as follows:

1. The respective templates must be configured in Semaphore, pointing to the correct playbooks, i.e. `netbox-plugin-deploy.yml` and `netbox-plugin-disable.yml`
2. Within the template, the tag must align with the plugin name, i.e. `-- tags=netbox_topology_views`
3. The plugin name must be added to `netbox-plugin-deploy.yml` and `netbox-plugin-disable.yml`
4. The mapping file must be configured, e.g. in Ansible `/mappings/netbox-plugins/netbox_topology_views.yml`



Some plugins, i.e. `netbox_data_flows`, caused issues with the `venv`, and broke the installation of other plugins. Further investigation is required.

To Do

- Define the optimal sequencing, i.e. upgrade plugin, then NetBox, then migrate, etc.
- Knit everything together, e.g. upgrading NetBox along with 7 different plugins
- Ensuring a critical mass of mapping tables and data to make it useful
- Adding / supporting external/local/private plugins
- Create combined playbooks / checks to test the installation of all NetBox plugins on all NetBox versions

Revision #14

Created 23 September 2024 12:34:53 by Richard Foster

Updated 9 October 2024 14:09:36 by Richard Foster