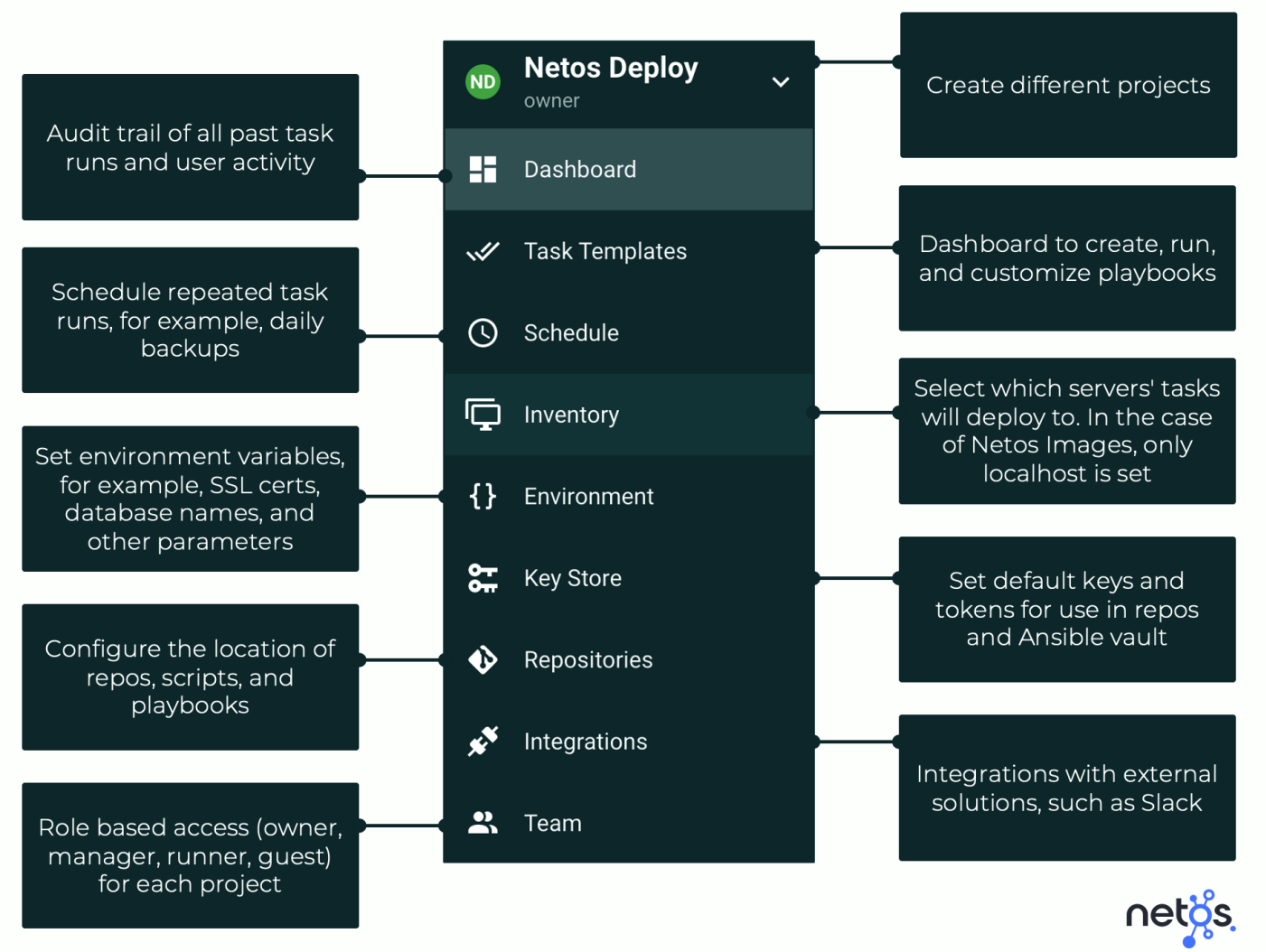
Key Semaphore Principles

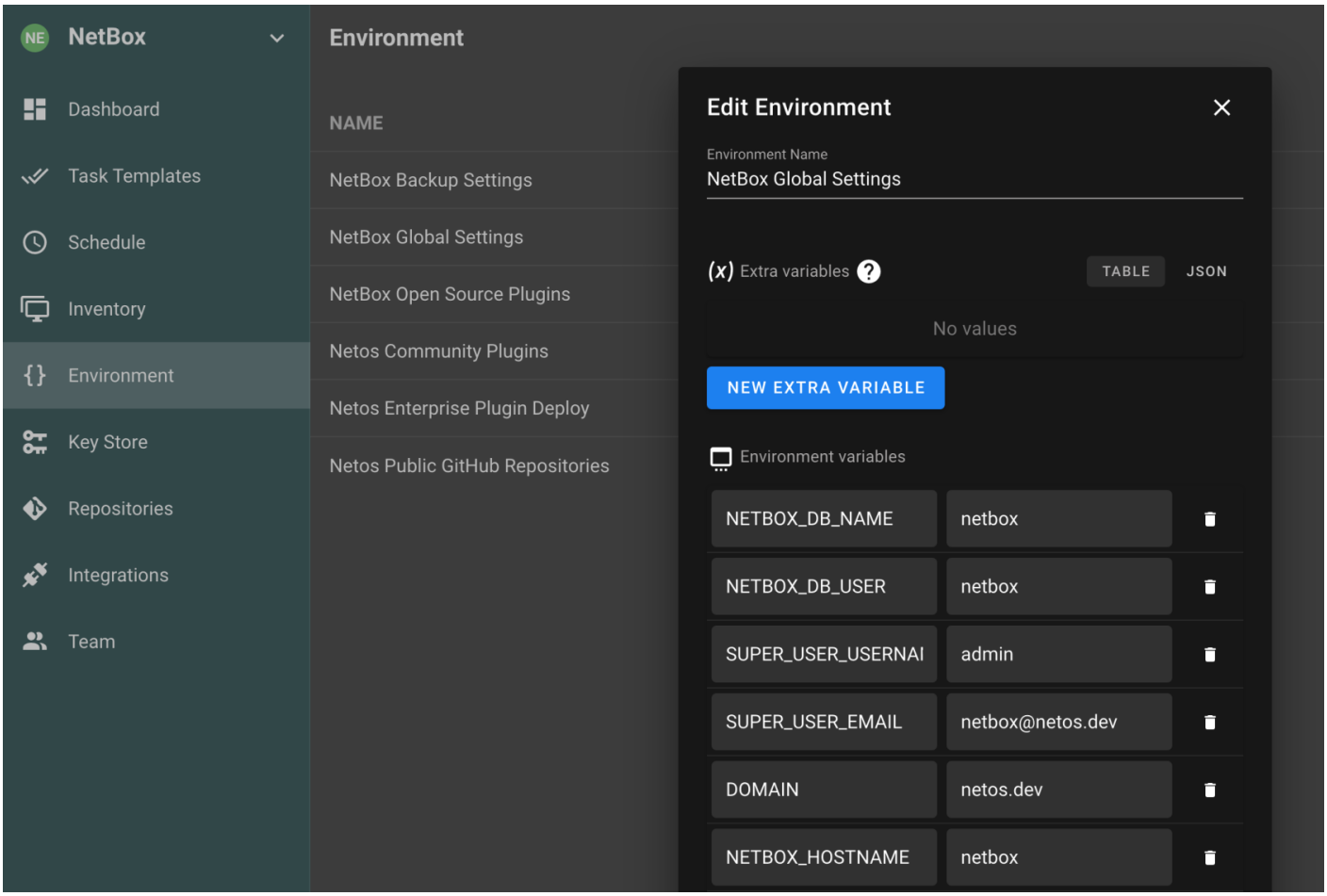
Introduction

The following diagram illustrates the purpose of each menu item in Semaphore, in the context of the Netos deployment.



Environments

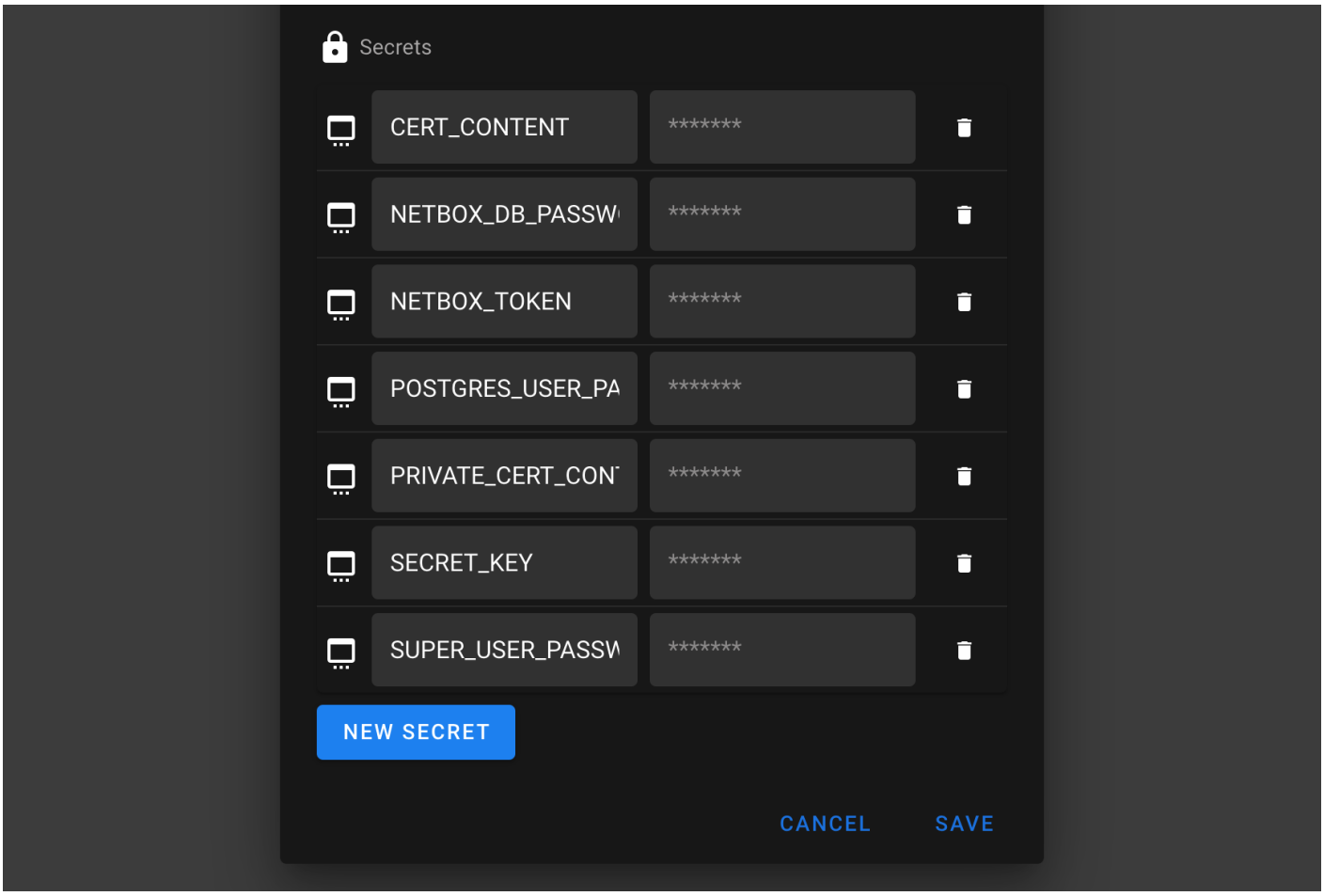
Environments pass variables into Ansible (or other scripts such as Python or Bash).



For example, in this environment file we pass through variables from the Semaphore Environment, but also set default variables where applicable in the `/vars/` files in the Ansible project.

```
CERT_CONTENT: "{{ lookup('env', 'CERT_CONTENT') | default('must-be-set-in-semaphore-variable')
}}"
CERT_DIR: "{{ lookup('env', 'CERT_DIR') | default('/netos/certs/netbox') }}"
DOMAIN: "{{ lookup('env', 'DOMAIN') | default('netos.dev') }}"
NETBOX_DB_NAME: "{{ lookup('env', 'NETBOX_DB_NAME') | default('netbox') }}"
```

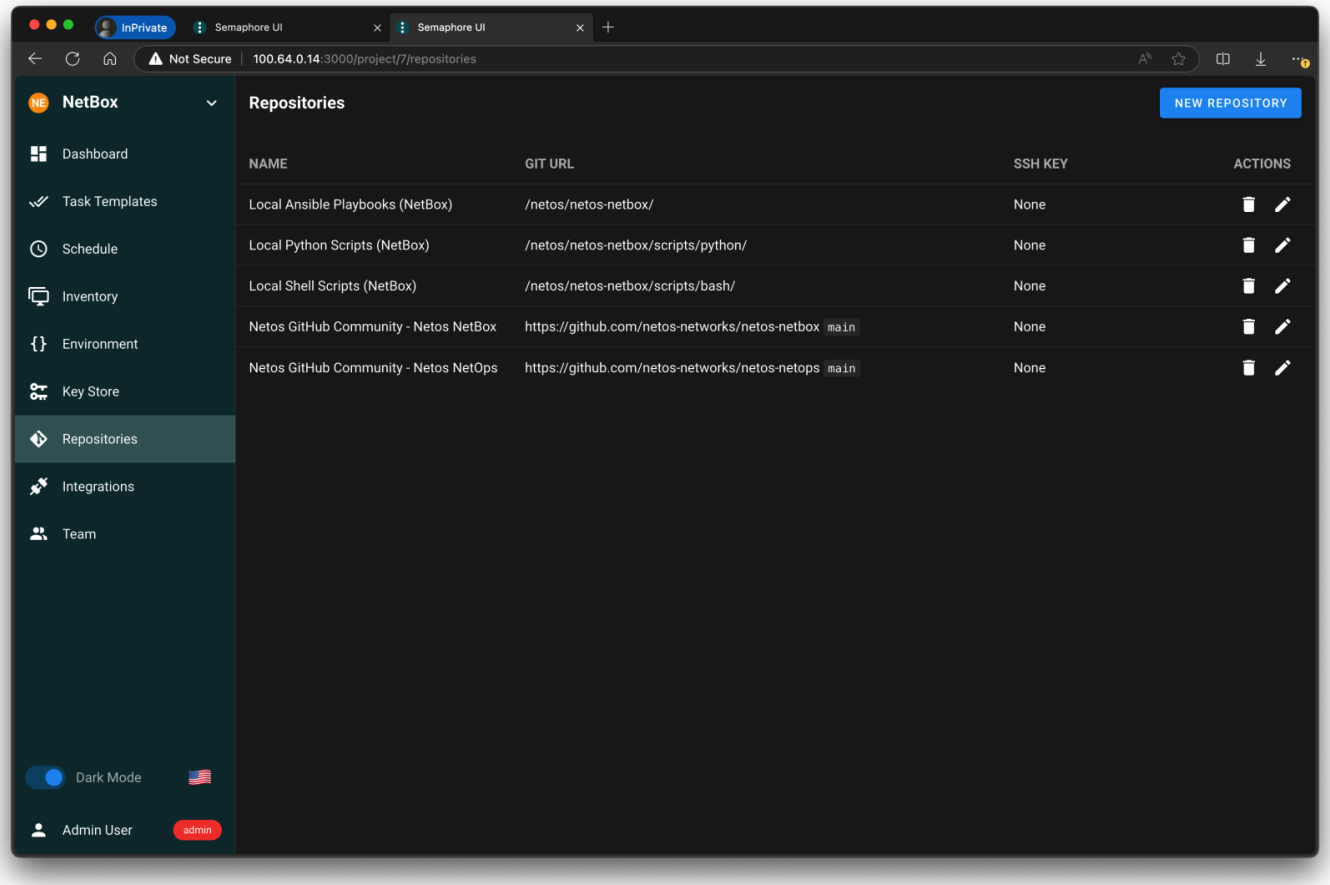
Semaphore also supports encrypted secrets as variables. We decided to use this feature throughout rather than Ansible Vault because it ensures that all data is in one place, rather than being split across different vaults and environments.



> There is a bug in Semaphore where updates to Secrets are not saved. Problem: not possible to edit secret name · Issue #2293 · semaphoreui/semaphore (github.com)

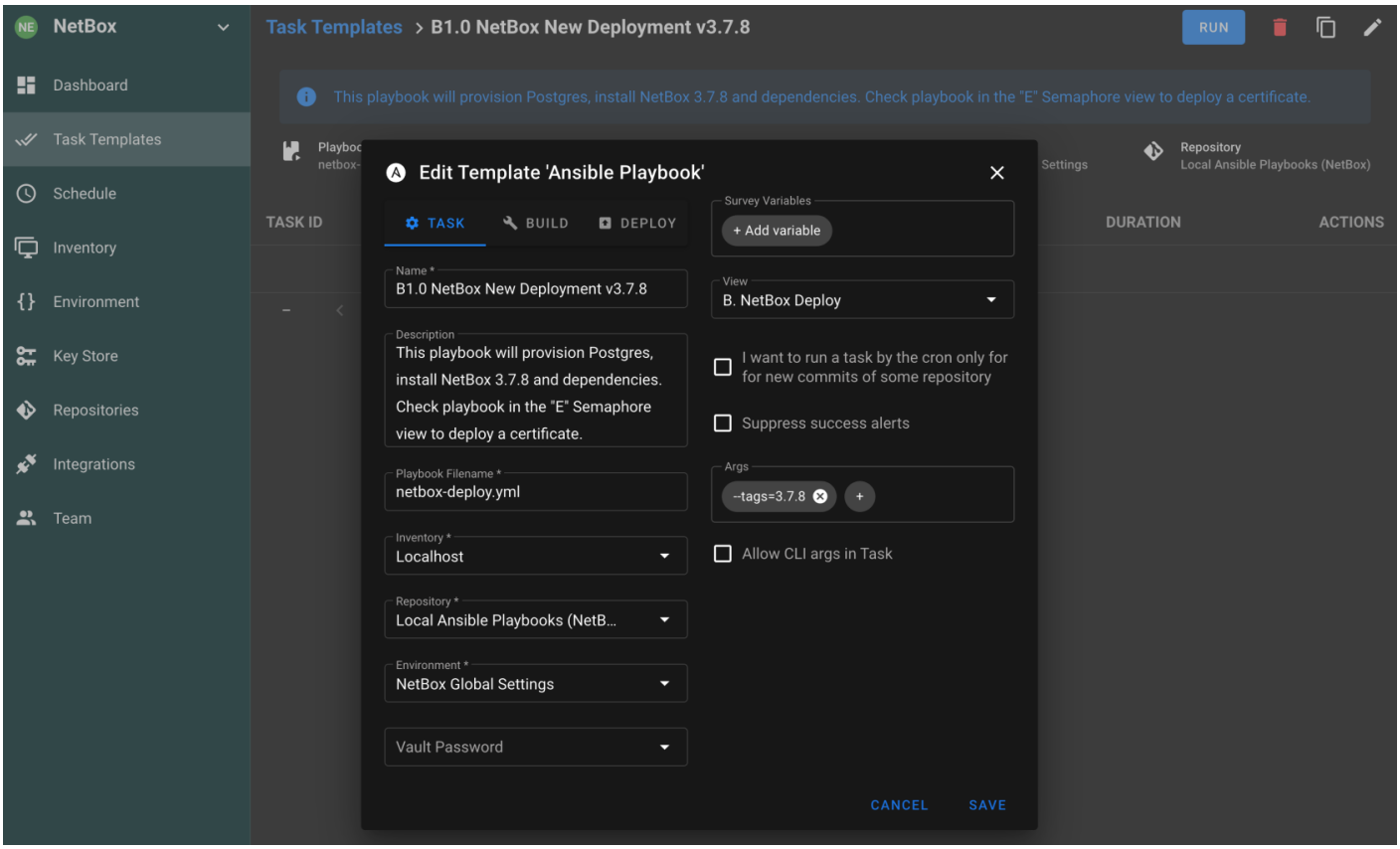
Repositories

Repositories are linked to Templates (e.g. an Ansible playbook) and in general point to a Git repository. In the case of all Netos projects, we instead point to a local file system, and instead have a per-project task to pull the repository from Netos Networks (github.com).



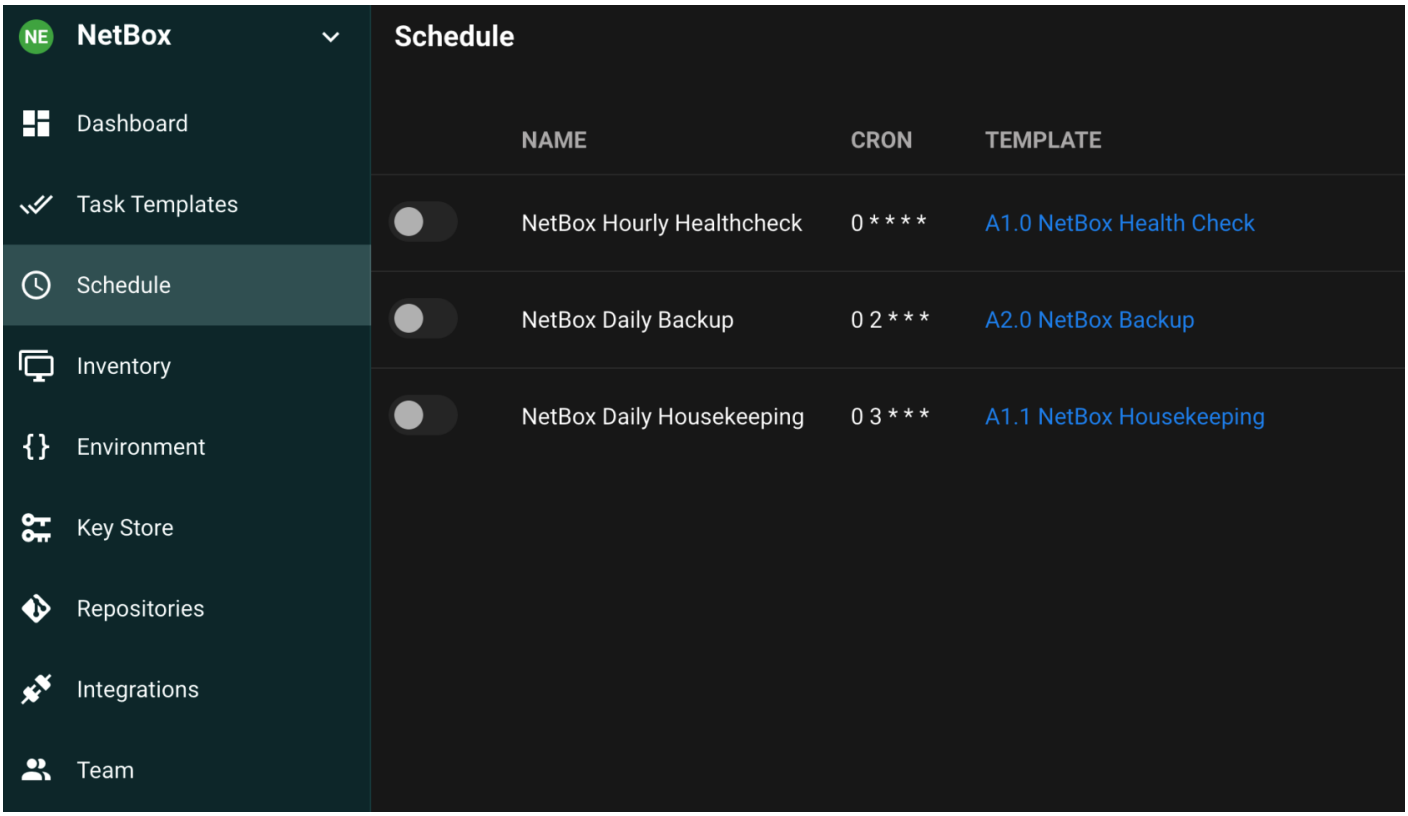
Templates

Everything above is pulled together into a Template.



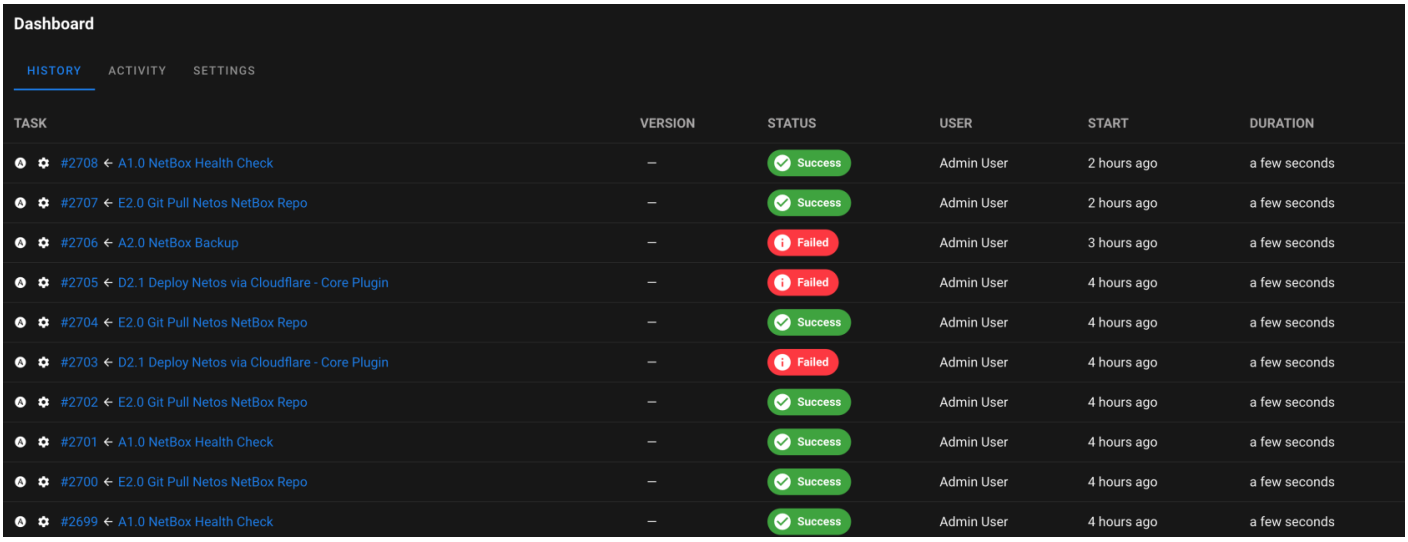
Scheduling

Templates can then be scheduled for repetitive tasks, such as backup and housekeeping tasks.



Tracking Tasks

You can globally track the status of all tasks, as well as within the logs of each Template.



Created 2024-09-22 11:53:58 UTC by Richard Foster
Updated 2024-10-09 13:18:47 UTC by Richard Foster