



DEPLOYMENT GUIDE

# Infoblox and Ansible Integration

Ansible 2.5 | April 2018



# Contents

Overview .....	3
Introduction .....	3
Guide Usage .....	3
Use Cases .....	3
Manage DNS Records, Networks and IP Addresses for VMs .....	3
Automate Deployment of Virtual Infoblox Appliances .....	3
Ansible .....	3
Introduction .....	3
Key Terms .....	3
Ansible Platforms .....	4
Availability .....	4
Requirements .....	4
Usage .....	4
Installation .....	4
Picking an Ansible Version .....	4
Installing Ansible .....	5
Verify Installation .....	5
Inventory .....	5
NIOS .....	5
Supported Versions .....	5
Initial Setup - Cloud Admin .....	6
Infoblox NIOS Module for Ansible .....	7
Overview .....	7
Introduction .....	7
Usage .....	7
Infoblox Modules for Ansible .....	7
Getting Started .....	7
Playbooks .....	8
Preparing Your Playbooks .....	11
Running your Playbooks .....	11
Appendix .....	12
Troubleshooting .....	12
NIOS Module Command Help .....	12
Uninstall the NIOS Module for Ansible .....	12
References and Sources .....	12
Helpful links .....	12

## Overview

### Introduction

Environments are becoming extremely dynamic as virtualization of hardware becomes more and more prevalent. To keep up with that, many organizations depend heavily on tools to automate, or 'orchestrate', tasks as much as possible. These 'orchestration tools' can automate tasks to the point where new applications or servers can be deployed with a single request.

### Guide Usage

In this guide, the following software versions are used:

- Ansible version 2.5.0.
- Ubuntu LTS 16.04.3
- NIOS 8.2.4

### Use Cases

#### Manage DNS Records, Networks and IP Addresses for VMs

Ansible enables the automation for creating and deleting VM's that are deployed across multiple platforms. Integration with Infoblox is powered by the "NIOS module" in Ansible. The NIOS module provides the framework for managing the networks, IP addresses, and DNS records in NIOS.

#### Automate Deployment of Virtual Infoblox Appliances

Organizations can use Ansible to automate the creation (and deletion) of virtual Infoblox appliances. Note: This process will be documented as part of a separate deployment guide.

## Ansible

### Introduction

Ansible is an open source platform that is both light weight and simple to deploy, manage and use. The Ansible platform makes it easy for administrators and developers to automate many tasks, including applying updates to machines on the network to managing devices on the network.

### Key Terms

- **Playbook:** A file used to set configurations and automate any manual processes. These are plain text files written in the Ansible automation language which describe the intended end-state of a deployment or task being executed.
- **Ansible automation language:** The structure used when writing playbooks and other resources for Ansible. The Ansible automation language uses YAML and is intended to be both human and machine readable.
- **Plays:** Operations within a playbook and execute tasks.
- **Tasks:** Used within a play to call modules and run in order.
- **Roles:** A grouping of files and playbooks which makes it easier to organize complex operations.
- **Inventory:** A list of hosts, or servers, that the control machine can use in its orchestration tasks.
- **Modules:** Also referred to as 'task plugins' or 'library plugins', these control system resources and provide integrations with other services.
- **Control machine:** Also referred to as a 'control node'. The system where you run your Ansible playbooks from.

## Ansible Platforms

Ansible has three offerings:

- Ansible: A free, open source product
- Ansible Tower: An enterprise offering which gives you a graphical interface and enables integration with other services and tools. Tower gives permission control and will also save a record of all Ansible playbook activity, useful for auditing purposes.
- Ansible Galaxy: A repository where roles generated by Ansible users can be found.

Ansible is the tool used to orchestrate (automate) the creation and deletion of VM's. In this guide, we use the main Ansible product.

## Availability

Ansible is available for Linux based operating systems (include MacOS) and can be installed on physical or virtual hosts. Windows is NOT a supported operating system for the control machine.

## Requirements

This section lists the (minimum) system requirements for installing and using Ansible:

- For the 'Control' machine, any distribution of **Linux with Python 2.6 or newer**, or 3.5 or newer.
- **PIP**, the package management system for Python.
- The **infoblox-client** WAPI package for python (pip install infoblox-client).
- If using MacOS, run the following command to avoid the error "Too many files open":

```
sudo launchctl limit maxfiles
```

- Internet access and working DNS on the system where Ansible is being installed (the 'Control' machine).
- If using RHEL (or equivalent), enable the 'Extras' and 'Optional' yum repositories.

## Usage

- **Ad-Hoc**: Direct from the command line.
- **Playbooks**: Automation 'scripts'.
- **Automation Framework**: Requires Ansible Tower.
- **Check Mode**: An option for running ad-hoc commands or playbooks without making changes.

## Installation

Ansible is supported on multiple Linux distributions so the installation steps will vary depending on the flavor that you are installing it on. When getting started, it is recommended to use the OS packages for EPEL and APT; although, Ansible is available through multiple sources, including Pypi and GitHub. For installation instructions for your OS (operating system), refer to

[http://docs.ansible.com/ansible/latest/intro\\_installation.html](http://docs.ansible.com/ansible/latest/intro_installation.html).

In this guide, we demonstrate the installation of the latest release of Ansible on Ubuntu using APT.

## Picking an Ansible Version

Ansible is available in two versions:

- Latest Release
- Development Version

Due to the lightweight nature of Ansible with a short release cycle, many installations will follow the development version. In this guide, we will the latest release.

## Installing Ansible

To install Ansible on Ubuntu, run the following commands:

1. `sudo apt-get update`
2. `sudo apt-get install software-properties-common`
3. `sudo apt-add-repository ppa:ansible/ansible`
4. `sudo apt-get update`
5. `sudo apt-get install ansible`

Note: This process generally only takes a few minutes to complete.

## Verify Installation

To verify that Ansible has been successfully installed, run the command `ansible --version`.

```
infoblox@ansible-test-1:~$ ansible --version
ansible 2.5.0
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/infoblox/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.12 (default, Dec  4 2017, 14:50:18) [GCC 5.4.0 20160609]
```

## Inventory

Ansible uses an 'inventory' to identify all servers that it manages. This can be done using a static 'hosts' file (found in `/etc/ansible/` by default) or a dynamically generated inventory list. To update the static inventory and add your Infoblox appliance, use the following command examples:

1. `sudo vi /etc/ansible/hosts`
2. `<shift-G>` (move to the bottom of the file)
3. `i` (to enter interactive mode)
4. Type the name or IP address for your Infoblox appliance.
5. `<esc>`
6. `:wq`

## NIOS

### Supported Versions

When preparing your playbooks, it is important to set the WAPI version to the version used by your version of NIOS by specifying the with the `wapi_version: x.x` parameter. You can verify the WAPI version used by your Infoblox appliance by appending "`/wapidoc/#backward-compatibility`" to the end of the URL to connect to your Infoblox Grid Manager GUI. Example:

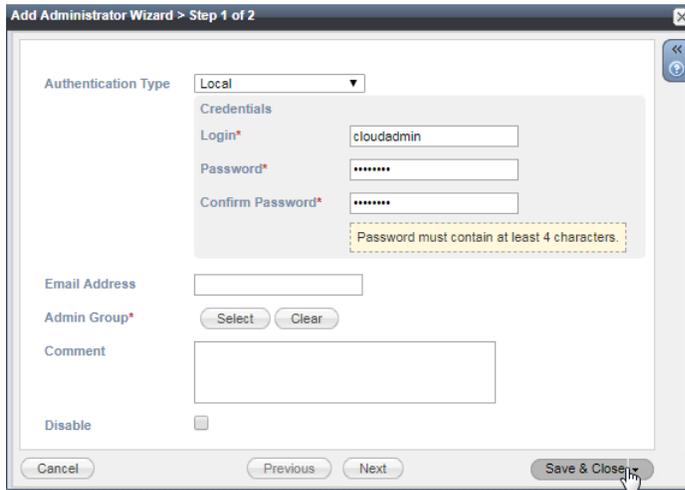
```
https://infoblox.localdomain/wapidoc/#backward-compatibility
```

In Ansible 2.5, the default for the WAPI version is set to 1.4, which corresponds to NIOS version 6.10. Some operations may require WAPI version 2.1 or newer. WAPI version 2.1 corresponds to NIOS version 7.1.

## Initial Setup - Cloud Admin

The plugin will authenticate with NIOS using an account specified in its config file/playbook or environment parameters. Before this will work, this account must first be created in NIOS. This can be a regular admin account, or a cloud-api enabled account, with the appropriate permissions. To create a cloud-api enabled admin account:

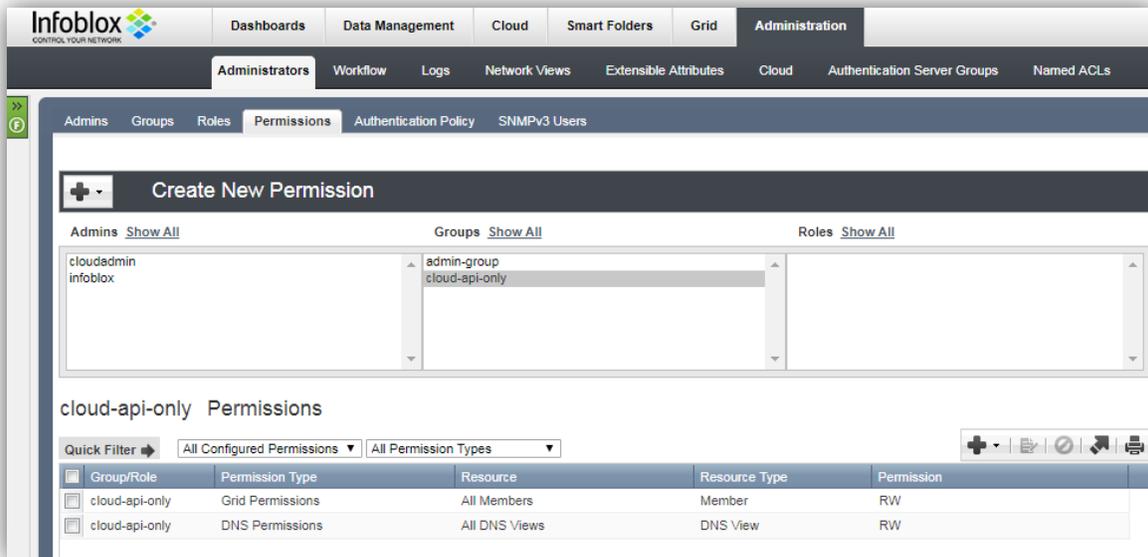
1. Login to your Infoblox Grid Manager GUI if not already logged in.
2. Navigate to **Administration** -> **Administrators** -> **Admins**.
3. Click on the **+** (Add) button.
4. Specify the user name in the Login field, along with the desired password in the two corresponding text boxes.
5. Click **Select** and choose the **cloud-api-only** group.
6. Click **Save & Close**.



Permissions must also be defined which will allow the plugin to make changes. To set the permissions:

1. Navigate to **Administration** -> **Administrators** -> **Permissions**.
2. Under the **Groups** column, select **cloud-api-only**.
3. Click on the **+** (Add) button. If the menu expands, select **Global Permissions** (clicking on the icon itself will default to this menu option).
4. Set the permissions as required. For lab purposes and getting started, allow Read/Write access for the following:
  - a. **DNS Permissions** -> **All DNS Views**
  - b. **DHCP Permissions** -> **All Network Views**
  - c. **Grid Permissions** -> **All Members** (to enable service restarts)

Note: Permissions are inherited. Unless overridden at a lower level, they apply to all objects underneath.



## Infoblox NIOS Module for Ansible

### Overview

#### Introduction

Beginning with Ansible version 2.5, five Infoblox NIOS modules are included with the default installation:

- nios\_dns\_view
- nios\_host\_record
- nios\_network
- nios\_network\_view
- nios\_zone

Documentation outlining the requirements, parameters and examples can be found at [http://docs.ansible.com/ansible/latest/modules/list\\_of\\_net\\_tools\\_modules.html#nios](http://docs.ansible.com/ansible/latest/modules/list_of_net_tools_modules.html#nios).

#### Usage

When using the Infoblox NIOS modules for Ansible, it is recommended to do so with playbooks. In this guide, we demonstrate these modules using playbooks.

### Infoblox Modules for Ansible

#### Getting Started

Starting with Ansible version 2.5, the NIOS modules are included with Ansible. Before being able to use these, you must install the infoblox-client WAPI package for Python and is compatible with Python version 2.6 or newer. To install the infoblox-client package, run the following command:

```
pip install infoblox-client
```

Note: This process may take a few minutes to complete.

## Playbooks

Developing playbooks that use the Infoblox NIOS modules can enable complex operations when automating IPAM functions for device management. Infoblox maintains a repository of simple playbooks that can be used for reference and can be cloned directly from GitHub.

To clone the Infoblox repository of sample Ansible playbooks:

```
1. cd /etc/ansible
2. sudo git clone https://github.com/infobloxopen/infoblox-ansible-  
playbooks.git
3. cd /etc/ansible/infoblox-ansible-playbooks/2.5
4. git status
   On branch master
   Your branch is up-to-date with 'origin/master'.
   nothing to commit, working directory clean
5. ls
   create_dnsview.yml  create_netview.yml  create_zone.yml
   delete_host.yml    delete_network.yml  other_lookups.yml
   create_host.yml     create_network.yml  delete_dnsview.yml
   delete_netview.yml delete_zone.yml
```

A select number of example playbooks are also included here for your reference, including:

### Create Network View

```
---
- hosts: localhost
  vars:
    nios_provider:
      host: 10.60.27.4
      username: admin
      password: infoblox
      wapi_version: 2.1

  connection: local
  tasks:
    - name: create network view
      nios_network_view:
        name: ansibleNetView
        extattrs:
          Site: Test Site
          comment: Created with Ansible
          state: present
        provider: "{{ nios_provider }}"
```

### Create Network

```
---
- hosts: localhost
  vars:
    nios_provider:
      host: 10.60.27.4
      username: admin
      password: infoblox
      wapi_version: 2.1

  connection: local
  tasks:
    - name: create network
      nios_network:
        network: 10.0.0.0/24
        network_view: ansibleNetView
        options:
          - name: domain-name
            value: infoblox.com
        extattrs:
          Site: Test Site
          comment: Created with Ansible
          state: present
        provider: "{{ nios_provider }}"
```

## Create DNS View

```
---
- hosts: localhost
  vars:
    nios_provider:
      host: 10.60.27.4
      username: admin
      password: infoblox
      wapi_version: 2.1

  connection: local
  tasks:
    - name: create DNS view
      nios_dns_view:
        name: ansibleDnsView
        network_view: ansibleNetView
        extattrs:
          Site: Test Site
        comment: Created with Ansible
        state: present
      provider: "{{ nios_provider }}"
```

## Create Host Record

```
---
- hosts: localhost
  vars:
    nios_provider:
      host: 10.60.27.4
      username: admin
      password: infoblox
      wapi_version: 2.1

  connection: local
  tasks:
    - name: create host record
      nios_host_record:
        name: host.ansiblezone.com
        view: ansibleDnsView
        ipv4addrs:
          - ipv4addr:
              "{{ lookup('nios_next_ip',
                '10.0.0.0/24',
                provider=nios_provider)[0] }}"
        ipv6addrs:
          - ipv6addr: fd00::2
        ttl: 3600
        extattrs:
          Site: Test Site
        comment: Created with Ansible
        state: present
```

## Delete Host Record

```
---
- hosts: localhost
  vars:
    nios_provider:
      host: 10.60.27.4
      username: admin
      password: infoblox
      wapi_version: 2.1

  connection: local
  tasks:
    - name: delete host record
      nios_host_record:
        name: host.ansiblezone.com
        view: ansibleDnsView
        state: absent
        provider: "{{ nios_provider }}"
```

## Preparing Your Playbooks

Once your environment has been setup, the first step before running your playbooks is to make sure that all variables are updated for your environment. In the examples provided in this guide, the variables which may require modification have been highlighted in **red**.

## Running your Playbooks

Once your playbooks have been updated with any changes required to make them work in your environment, you are ready to begin working with them. To run a playbook, use the command **ansible-playbook**.

Example:

```
infoblox@ansible-test-1:/etc/ansible/infoblox-ansible-playbooks/2.5$ ansible-playbook delete_netview.yml
PLAY [localhost] *****
TASK [Gathering Facts] *****
ok: [localhost]

TASK [delete network view] *****
changed: [localhost]

PLAY RECAP *****
localhost : ok=2    changed=1    unreachable=0    failed=0

infoblox@ansible-test-1:/etc/ansible/infoblox-ansible-playbooks/2.5$
```

## Appendix

### Troubleshooting

#### NIOS Module Command Help

If the infoblox-client package for Python has not been installed, you will see an error confirming that it is required. Example:

```
$ python infoblox.py
infoblox-client is required but does not appear to be installed. It can be installed using the command
`pip install infoblox-client`
```

#### Uninstall the NIOS Module for Ansible

The NIOS modules themselves are built-in to Ansible so there is nothing to uninstall for these. To uninstall Ansible, run the following command:

```
sudo apt-get purge --auto-remove ansible
```

To remove the infoblox-client package for Python:

```
sudo pip uninstall infoblox-client
```

### References and Sources

#### Helpful links

<https://www.ansible.com/>

<http://docs.ansible.com/ansible/latest/>

<http://docs.ansible.com/ansible/latest/YAMLSyntax.html>

[http://docs.ansible.com/ansible/latest/vmware\\_guest\\_module.html](http://docs.ansible.com/ansible/latest/vmware_guest_module.html)

<https://www.infoblox.com/>

<https://support.infoblox.com/>

<https://community.infoblox.com/>