Infoblox
NEXT LEVEL NETWORKING

DEPLOYMENT GUIDE

Automate Infoblox Infrastructure Using Ansible

TABLE OF CONTENTS

Overview

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.

Ansible

Ansible is a popular open-source automation tool, or platform, used for IT tasks such as configuration management, application deployment, intra-service orchestration and provisioning. It 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.

Ansible Platforms

Ansible has three offerings:

- **Ansible**: A free, open source automation 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**: This refers to the Galaxy website where users can share roles, and to a command line tool for installing, creating and managing roles.

In this deployment guide, we use the main Ansible product.

Usage

You can leverage the capabilities of Ansible in multiple ways:

- **Ad-Hoc**: Issue ansible tasks direct from the command line. This is a good place to start to understand the basics of what Ansible can do prior to learning the playbooks language – ad-hoc commands can also be used to do quick things that you might not necessarily want to write a full playbook for.
- **Playbooks**: These are automation scripts. Playbooks are Ansible's configuration, deployment, and orchestration language. They can describe a policy you want your remote systems to enforce, or a set of steps in a general IT process.
- **Automation Framework**: Requires the Ansible Tower.
- **Check Mode**: An option for running ad-hoc commands or playbooks without making changes.

This deployment guide walks you through working with playbooks.

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.
- **Host**: In Ansible, a host is a remote machine that is assigned to individual variables and they

are further grouped together. Each host has a dedicated name or unique IP address to make its identification easy and quick. They can be given simple port number too if you don't have to access them over SSH connection.

- **Modules**: Also referred to as 'task plugins' or 'library plugins', these control system resources and provide integrations with other services. Modules are the main components that help to install packages, allow APIs to interact together and plan actions for system files too. There are a variety of modules in Ansible that are programmed to automate almost everything inside the tool.
- **Plug-ins**: They are the special pieces of code that help to write code quickly. Plug-ins automate the development tasks and help to speed up the deployment work to the maximum level.
- **Control machine**: Also referred to as a 'control node'. The system where you run your Ansible playbooks from.

Infoblox Integration

The Ansible 2.5 open source project release added support for the Infoblox Network Identity Operating System (NIOS) enablement. For network professionals, this means that existing networking Ansible Playbooks can utilize existing Infoblox infrastructure for IP Address Management (IPAM), using Infoblox for tracking inventory and more. Ansible 2.8 supports 16 modules, covered in the later sections.

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 that 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. You can leverage this module for autoscaling grid members based on DNS traffic.

Deployment

Requirements

Ansible is available for Linux based operating systems (include MacOS) and can be installed on physical or virtual hosts.
This section lists the (minimum) system requirements for installing and using Ansible. For more details refer to the official documentation present here:

- For the 'Control' machine, any distribution of Linux with Python 2.7 or newer, or 3.5 or newer.
- For the 'managed nodes', you need Python 2.6 or newer, or 3.5 or newer.
- PIP, the package management system for Python. If not already present, this can be installed

```
sudo apt install python-pip
```

- The infoblox-client WAPI package for python.

```
sudo pip install infoblox-client
```

- If using MacOS, run the following command to avoid the error "Too many files open":

```
sudo launchctl limit maxfiles unlimited
```

- 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.

Initial Setup

Ansible

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.

Picking an Ansible Version

Ansible is available in two versions:

- Latest Release
- Development Version

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

Installation

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 following command:

```
ansible --version
```

```
root@ansible-demo:~# ansible --version
ansible 2.8.1
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/sh
are/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, Nov 12 2018, 14:36:49) [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:

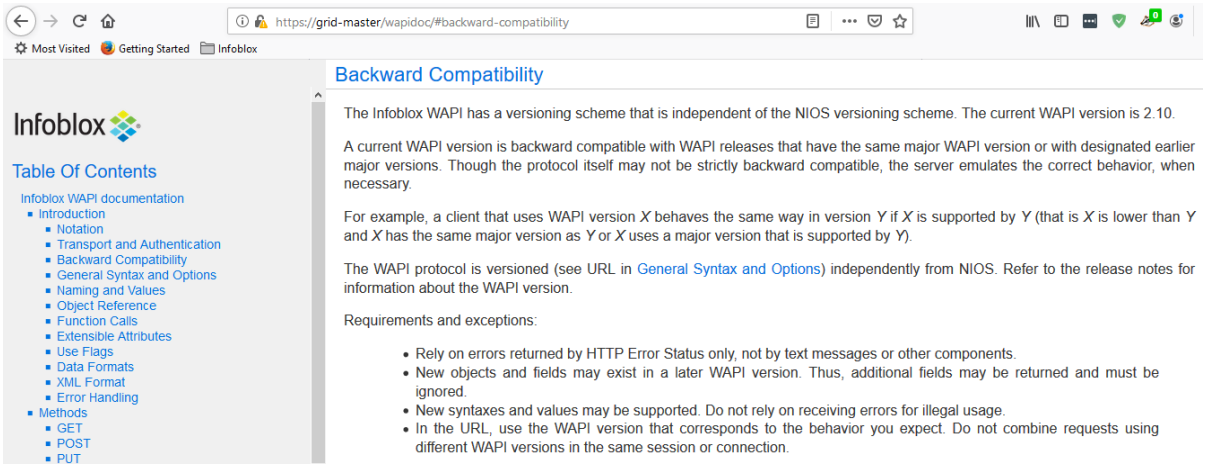
```
1. sudo vi /etc/ansible/hosts
2. <shift-G> (move to the bottom of the file)
3. i (to enter interactive mode)
4. Type the name or IP address for your Infoblox
   appliance.
5. <esc>
6. :wq
```

Infoblox

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 **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://grid-master/wapidoc/#backward-compatibility
```



In Ansible 2.8, 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. For example, the minimum version required for the NIOS member module to work is 2.2

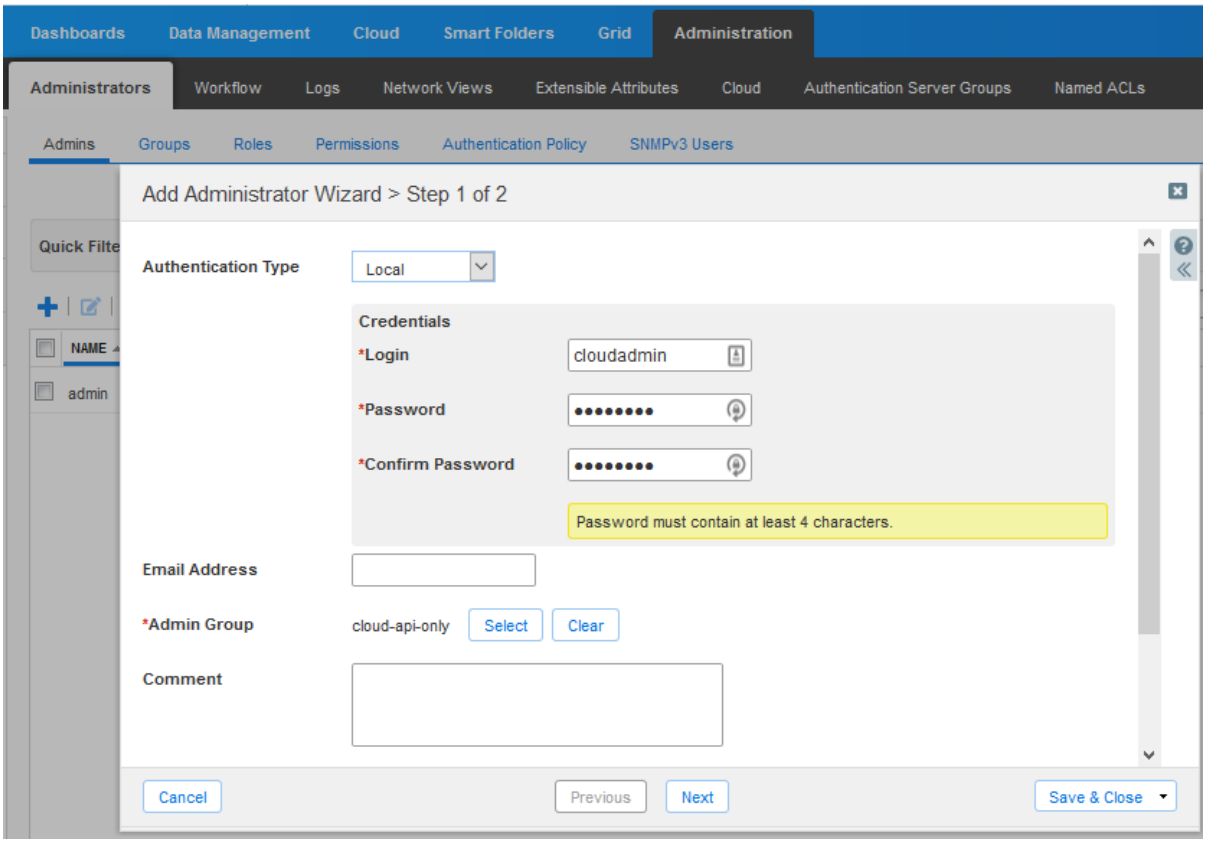
Cloud Admin user (Optional)

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**.

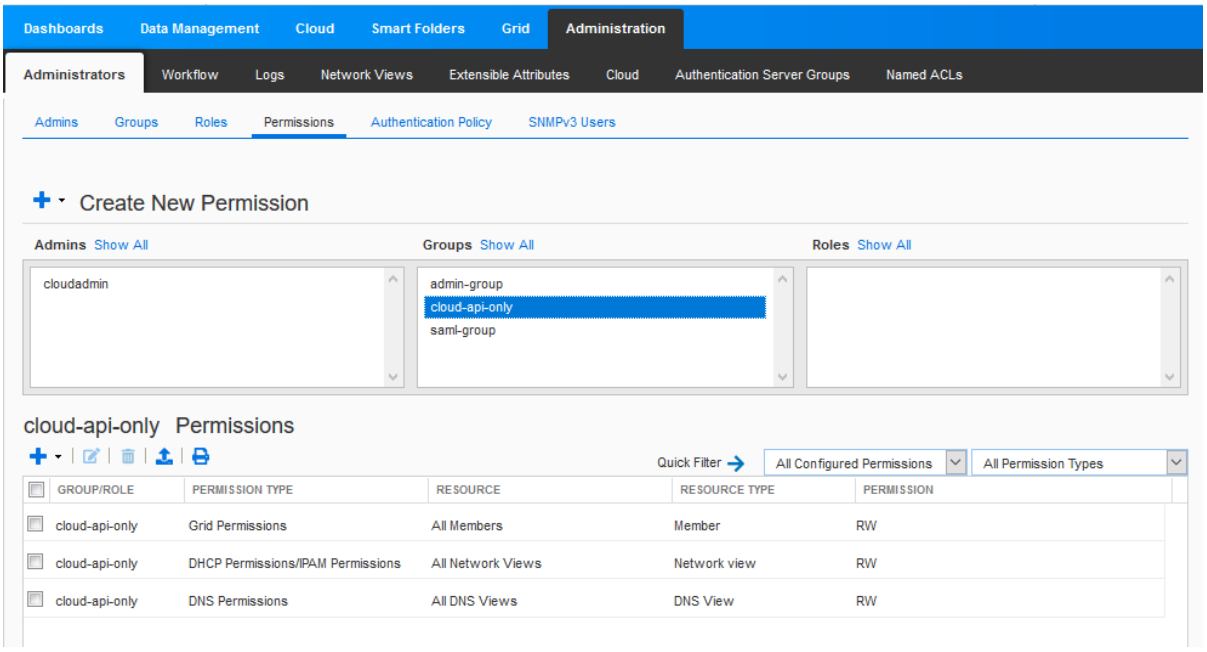
Note: For the **cloud-api-only** group to be available, you need to have the Cloud Network Automation license enabled on your NIOS appliance.



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**

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



Infoblox NIOS Modules for Ansible

There are a total of 16 modules included with Ansible 2.8. They can be currently found in the development branch of the documentation:

- nios_a_record – Configure Infoblox NIOS A records
- nios_aaaa_record – Configure Infoblox NIOS AAAA records
- nios_cname_record – Configure Infoblox NIOS CNAME records
- nios_dns_view – Configure Infoblox NIOS DNS views
- nios_fixed_address – Configure Infoblox NIOS DHCP Fixed Address
- nios_host_record – Configure Infoblox NIOS host records
- nios_member – Configure Infoblox NIOS members
- nios_mx_record – Configure Infoblox NIOS MX records
- nios_naptr_record – Configure Infoblox NIOS NAPTR records
- nios_network – Configure Infoblox NIOS network object
- nios_network_view – Configure Infoblox NIOS network views
- nios_nsgroup – Configure Infoblox DNS Nameserver Groups
- nios_ptr_record – Configure Infoblox NIOS PTR records
- nios_srv_record – Configure Infoblox NIOS SRV records
- nios_txt_record – Configure Infoblox NIOS txt records
- nios_zone – Configure Infoblox NIOS DNS zones
- lookup – Fetch Infoblox NIOS specified objects. The documentation for this can be found here.

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.

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:

```
sudo 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
6. cd /etc/ansible/infoblox-ansible-playbooks/2.7
7. ls
   create_aaaa_record.yaml   create_naptr_record.yaml
   delete_aaaa_record.yaml   delete_naptr_record.yaml
   create_a_record.yaml      create_ptr_record.yaml
   delete_a_record.yaml      delete_ptr_record.yaml
   create_cname_record.yaml  create_srv_record.yaml
   delete_cname_record.yaml  delete_srv_record.yaml
   create_mx_record.yaml     create_txt_record.yaml
   delete_mx_record.yaml     delete_txt_record.yaml
8. cd /etc/ansible/infoblox-ansible-playbooks/2.8
9. ls
   create_ha_member.yaml    create_token_member.yaml
   delete_member.yaml       member_lookup.yaml
   create_member.yaml       delete_ha_member.yaml
   modify_member.yaml
```

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

Create Network View

```
---

- hosts: localhost
  connection: local
  vars:
    nios_provider:
      host: grid-master
      username: cloudadmin
      password: infoblox
      wapi_version: 2.9
  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
  connection: local
  vars:
    nios_provider:
      host: grid-master
      username: cloudadmin
      password: infoblox
      wapi_version: 2.9
  tasks:
   - name: create network view
     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 Host Record

```
---

- hosts: localhost
  connection: local
  vars:
    nios_provider:
      host: grid-master
      username: cloudadmin
      password: infoblox
      wapi_version: 2.9
  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
       provider: "{{ nios_provider }}"
```

Create A Record

```
---

- hosts: localhost
  connection: local
  vars:
    nios_provider:
      host: grid-master
      username: cloudadmin
      password: infoblox
      wapi_version: 2.9
  tasks:
   - name: create A record
     nios_a_record:
       name: a.demo.com
       ipv4: 10.0.0.2
       comment: Created with Ansible
       view: ansibleDnsView
       state: present
       provider: "{{ nios_provider }}"
```

Create member

```
---

- hosts: localhost
  connection: local
  vars:
    nios_provider:
      host: grid-master
      username: cloudadmin
      password: infoblox
      wapi_version: 2.9
  tasks:
   - name: Create Infoblox member
     nios_member:
       host_name: member01.ansible-demo.com
       vip_setting:
        - address: 192.168.1.71
          subnet_mask: 255.255.255.0
          gateway: 192.168.1.1
       config_addr_type: IPV4
       pre_provisioning:
         - hardware_info:
           - hwmodel: IB-VM-820
             hwtype: IB-VNIOS
           licenses:
           - dns
           - dhcp
           - enterprise
           - vnios
           - rpz
       platform: VNIOS
       comment: Created with Ansible
       state: present
       provider: "{{ nios_provider }}"
```

Delete Host Record

```
---

- hosts: localhost
  connection: local
  vars:
    nios_provider:
      host: grid-master
      username: cloudadmin
      password: infoblox
      wapi_version: 2.9
  tasks:
   - name: Delete Host record
     nios_host_record:
       name: host.ansiblezone.com
       view: ansibleDnsView
       state: absent
       provider: "{{ nios_provider }}"
```

Delete Member

```
---

- hosts: localhost
  connection: local
  vars:
    nios_provider:
      host: grid-master
      username: cloudadmin
      password: infoblox
      wapi_version: 2.9
  tasks:
   - name: Delete member
     nios_member:
       host_name: member01.ansible-demo.com
       state: absent
       provider: "{{ nios_provider }}"
```

Sample Lookup plugin

```
---
- hosts: localhost
  connection: local
  vars:
    nios_provider:
      host: grid-master
      username: cloudadmin
      password: infoblox
      wapi_version: 2.9
  tasks:
   - name: get member list
     set_fact:
       members: "{{ lookup('nios', 'member', provider=nios_provider)
}}"
   - name: display all members
     debug:
       msg: "{{ members }}"
```

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 the playbook use the `ansible-playbook` command:

```
root@ansible-demo:~/infoblox-ansible-playbooks/2.7# ansible-playbook create_a_record.yaml

PLAY [localhost] ***************************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [localhost]

TASK [Create Nios A record] ****************************************************************************
changed: [localhost]

PLAY RECAP *********************************************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
  ignored=0
```

Conclusion

The Ansible modules allow you to configure Infoblox, the lookup plugin allows you to grab information from Infoblox to use in subsequent tasks. With these modules, you can now automate your Infoblox infrastructure.

Additional Information

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
https://community.infoblox.com/
Community Blog: Infoblox vNIOS Autoscaling on Openstack using Ansible
Community Blog: What is new with Ansible 2.8

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`
```

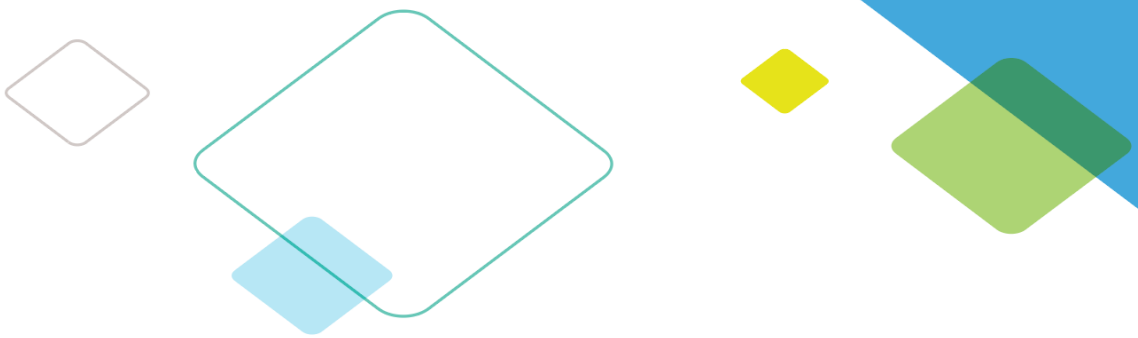
Uninstall the NIOS Module for Ansible

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
```



Infoblox is leading the way to next-level DDI with its Secure Cloud-Managed Network Services. Infoblox brings next-level security, reliability and automation to on-premises, cloud and hybrid networks, setting customers on a path to a single pane of glass for network management. Infoblox is a recognized leader with 50 percent market share comprised of 8,000 customers, including 350 of the Fortune 500.

Corporate Headquarters | 3111 Coronado Dr. | Santa Clara, CA | 95054

+1.408.986.4000 | 1.866.463.6256 (toll-free, U.S. and Canada) | info@infoblox.com | www.infoblox.com