OcNOS NOS Installation with Ansible

Created by:
Ron Wilhelmson
ron.wilhelmson@deltaww.com

Version:
Draft 1

Introduction

The basic installation of Network Operating Systems can be automated with scripts and the use of Ansible Playbooks. This guide will outline the installation of OcNOS NOS configuration on Delta Networks switches.

Objective

The objective of this guide is to document the basic steps required to install a verified OcNOS NOS installer remotely in an automated way using Ansible and configuration scripts. Complementary information for basic setup and Ansible use can be referenced at Getting Started with Ansible.

Pre-install Connectivity and Setup

Network and Systems required

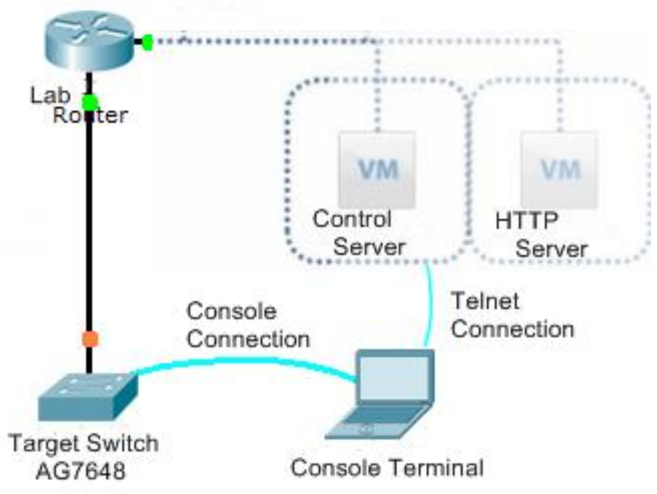
The basic systems required for building and running Ansible playbooks consist of the following:

Control Server – Linux server that runs Ansible and contains playbooks.

Target Switch – Switch that is to be configured running ONIE and may be many.

Web Server – HTTP location where update, license, and install files are located.

Console Terminal – PC with console connection to switch and telnet to control server.



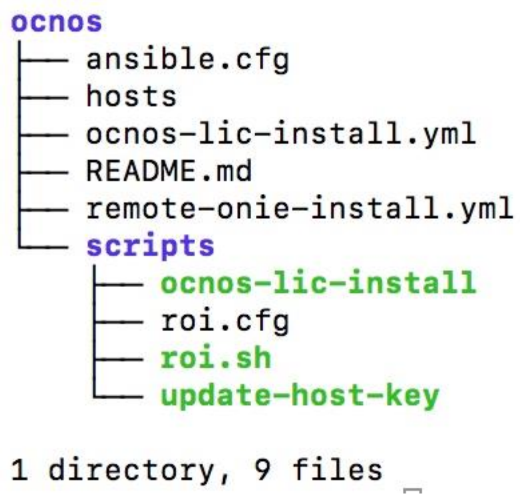
Network Diagram

Note: Please reference Getting Started with Ansible for additional details on system, switch, and server setups.

Ansible Files

The following describes the primary files you will need to get the ONIE playbook running for OcNOS playbooks and scripts.

Ansible Directory and Files



```
ocnos
├── ansible.cfg
├── hosts
├── ocnos-lic-install.yml
├── README.md
├── remote-onie-install.yml
└── scripts
    ├── ocnos-lic-install
    ├── roi.cfg
    ├── roi.sh
    └── update-host-key

1 directory, 9 files
```

ansible.cfg

The ansible.cfg file contains all of the configuration variables that can be set and Ansible will read this configuration file when it is initiated

hosts

The hosts file is a reference file used to define the switches and systems that Ansible will be executing tasks on.

Example hosts file:

```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups
#
# DPR Lab Agema Switch AG7648
#
[switches]
10.62.10.40
[http_server]
10.62.10.22
[control_server]
10.62.10.22
```

playbook

Ansible playbooks are written in the YAML language. For this OcNOS ONIE Install playbook our playbook will have the following sections:

`hosts` – defines the systems to pass commands to. In this case it is "`localhost`"

`tasks` – commands that will be executed by the playbook

For this playbook we are also using the `user: root` and `sudo: yes` as required.

Note: Please see the references at the end of this document for additional details on commands and modules.

Ansible Playbook NOS Installation

The following YAML format playbook is used to initiate the installation of the OcNOS NOS installer.

ONIE Install Playbook

The following playbook along with a helper script and associated configuration file are utilized to initiate and run the OcNOS NOS installer on the target switch running ONIE. Edits to the configuration file and helper script may be required for your specific network details and the version of installer file being loaded. The actual playbook is simple and uncluttered. The playbook initiates the helper shell script with the command task. The Remote ONIE Install script, `roi.sh`, initiates a download of the OcNOS installation file from the HTTP server to the target switch and then executes it. The playbook, script, and configuration file are shown below.

ONIE Install Playbook: remote-onie-install.yml

```
---

- hosts: localhost
  user: root
  sudo: yes
  tasks:
  - name: Execute script
    command: '/bin/bash ./scripts/roi.sh'
```

Ansible utilizes SSH to connect to the target switch to execute the commands defined in the playbook tasks. It does this by creating a series of tasks and then uses `sftp`, by default, to put the Ansible created commands in the `.ansible/tmp` directory on the target switch. Once the command file is on target system Ansible issues an SSH command to execute the temporary command file and then cleans up by deleting the temporary file. The target switch though in our case has ONIE installed and running. ONIE by default does not support `sftp` transfers so it is difficult for Ansible to move the tasks file to the target switch for execution. Therefore we have a helper script, roi.sh, which is called by the command: task as shown below and sources the associated roi.cfg file for specific details for the installation.

Helper Script remote-onie-install: roi.sh

```
#!/bin/bash

#  Copyright (C) 2018 Ron Wilhelmson <ron.wilhelmson@deltaww.com>
#
#

##
## remote-onie-install
##
## History: 12APR2018 Ron.Wilhelmson Initial Creation
##          04MAY2018 Ron.Wilhelmson Update for NOS install options cumulus or icos
##          25MAY2018 Ron.Wilhelmson Update for NOS install option ocnos
##          06JUL2018 Ron.Wilhelmson Update for NOS install option onl
##          19JUL2018 Ron.Wilhelmson Update for NOS install option cisco
#
# Dependencies: roi.cfg in same directory

# Set environment variables from roi.cfg for switch and server names/IPs
source /root/ansible/scripts/roi.cfg

echo ""
echo "Target switch set to: $target_switch"
echo ""
echo "HTTP server set to: $http_server"
echo ""
echo "NOS to be installed on Target switch: $NOS_install"

echo "Sending install command to switch"


case $NOS_install in
  cisco) /usr/bin/ssh -a -l root $target_switch /bin/onie-nos-install http://"$http_server"/iosxrwb-full-x.installer
    echo ""
    ;;

  cumulus) /usr/bin/ssh -a -l root $target_switch /bin/onie-nos-install http://"$http_server"/cumulus-linux-3.5.0-bcm-amd64.bin
    echo ""
    ;;

  icos) /usr/bin/ssh -a -l root $target_switch /bin/onie-nos-install http://"$http_server"/onie-icos-installer-x86_64-ag7648
    echo ""
    ;;

  ocnos) /usr/bin/ssh -a -l root $target_switch /bin/onie-nos-install http://"$http_server"/DELTA_AG_7648-OcNOS-1.3.4.260-
DC_MPLS_ZEBM-S0-P0-installer
    echo ""
    ;;
```

```
   onl) {
        sleep 10
        echo /bin/install_url http://"$http_server"/AG9032v1/ONL-2.0.0_ONL-OS_2018-01-17.0840-
6f80df8_AMD64_INSTALLED_INSTALLER
        sleep 60
        echo exit
        } | telnet $target_switch
esac

echo "Waiting for onie to download and install"

sleep 60

echo "Rebooting switch"
```

Config File remote-onie-install: roi.cfg

```
#  Copyright (C) 2018 Ron Wilhelmson <ron.wilhelmson@deltaww.com>
#
#

##
## remote-onie-install variable definitions
##
## History: 12APR2018 Ron.Wilhelmson Initial Creation
##          04MAY2018 Ron.Wilhelmson Updated to set NOS install variable

# Reliancies: roi.sh

# set network operating system (NOS) to be installed on target switch
# current options are: cumulus
#                      OcNOS

NOS_install=OcNOS

# set the switch and server names/IPs according to your systems

# target_switch is the switch that NOS is being installed on

target_switch=10.62.10.40

# http_server is the URL server where the NOS bin and license files are located

http_server=10.62.10.22
```

Playbook Execution

The Ansible command to run this playbook installer is as follows:

```
ansible-playbook remote-onie-install.yml --ask-pass -v
```

Note: The password that will be requested, with the `--ask-pass` option, is for the ONIE root account which is blank so a return is required when the playbook runs and asks for “SSH password:” and the `-v` option increases the verbosity of the execution and provides a more detailed response.

The following playbook execution will be run on the control server through a terminal connection from your PC or directly if working on a workstation and the resulting output is shown.

Example Playbook Output: remote-onie-install.yml

```
root@DPR-LABVM-01:/root/ansible# ansible-playbook remote-onie-install.yml --ask-
pass -v
Using /root/ansible/ansible.cfg as config file
SSH password:

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

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

TASK [Execute script]
***********************************************************************************
*********
changed: [localhost] => changed=true
  cmd:
  - /bin/bash
  - ./scripts/roi.sh
  delta: '0:01:16.455921'
  end: '2018-09-08 14:51:27.078459'
  rc: 0
  start: '2018-09-08 14:50:10.622538'
  stderr: ''
  stderr_lines: []
  stdout: |2-

    Target switch set to: 10.62.10.26

    HTTP server set to: 10.62.10.22

    NOS to be installed on Target switch: ocnos
    Sending install command to switch
    Info: Fetching http://10.62.10.22/DELTA_AG_7648-OcNOS-1.3.4.260-DC_MPLS_ZEBM-
S0-P0-installer ...
    Connecting to 10.62.10.22 (10.62.10.22:80)
    installer             22% |******                         | 72157k  0:00:03
ETAinstaller             58% |******************             |   182M  0:00:01
ETAinstaller             93% |*****************************  |   294M  0:00:00
ETAinstaller            100% |*******************************|   313M  0:00:00 ETA

    Waiting for onie to download and install
    Rebooting switch
  stdout_lines: <omitted>

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

root@DPR-LABVM-01:/root/ansible#
```

The console connection to the target switch will display the following screens when the playbook is executed:

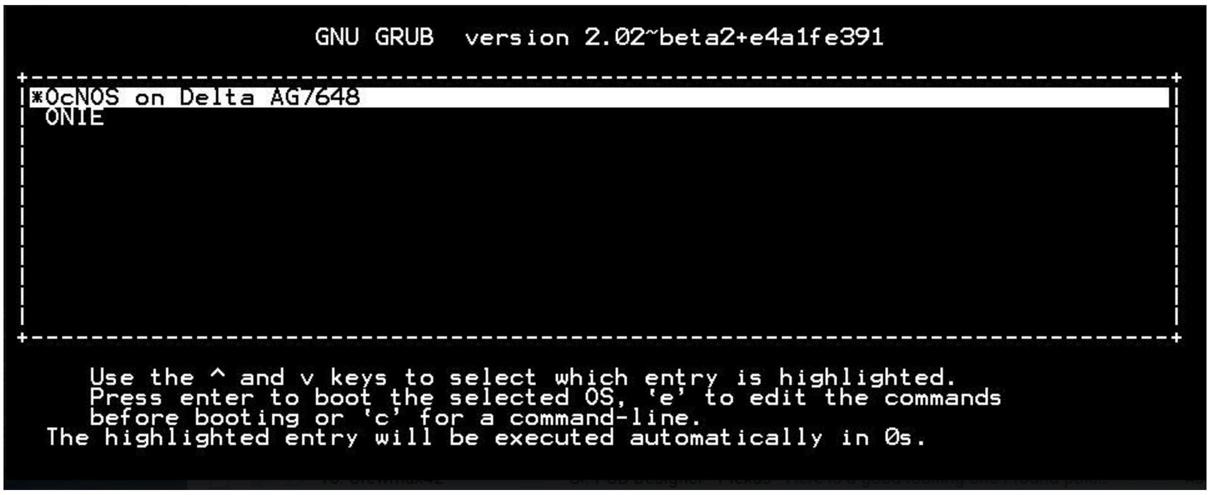
```
ONIE:/ #
ONIE:/ # discover: installer mode detected.
Stopping: discover… done.
ONIE: Executing installer:  http://10.62.10.22/DELTA_AG7648-OcNOS-1.3.4.260-
DC_MPLS_ZEBM-S0-P0-installer
Verifying image checksum ... Done.
Reinstalling the ONIE GRUB in MBR ... Done.
Do not interrupt the following operations. Device will reboot automatically with
OcNOS

Installing OcNOS on Delta AG7648 Switch ...
INFO: This device would be accessible after reboot with IP: 10.62.10.47 or as IP
assigned by the DHCP server
Creating new OcNOS Config partition /dev/sda3 ... Done.
Creating file system to use for OcNOS Configs ... Done.
INFO: To access device, Use serial console or ssh/telnet.
      Default username/password: OcNOS/OcNOS

Creating OcNOS rootfs partition /dev/sda4 ... Done.
Creating file system to use for OcNOS Root File System ... Done.
Mounting the partition to use for OcNOS ... Done.
Extracting rootfs. This may take a few seconds ... Done.
Updating boot loader ... Done.
Performing postinstallation steps ... Mounting the partition to use for OcNOS
Configuration ... Done.

Done.
```

Once the installer completes downloading from the HTTP server it will install and reboot with the default option to load OcNOS:



```
                 GNU GRUB  version 2.02~beta2+e4a1fe391

+----------------------------------------------------------------------------+
|*OcNOS on Delta AG7648                                                      |
| ONIE                                                                       |
|                                                                            |
+----------------------------------------------------------------------------+

     Use the ^ and v keys to select which entry is highlighted.
     Press enter to boot the selected OS, `e' to edit the commands
     before booting or `c' for a command-line.
  The highlighted entry will be executed automatically in 0s.
```

When the OcNOS software completes loading the switch will boot into OcNOS and display the OcNOS login prompt. The default OcNOS login ID is “ocnos” with password “ocnos”.

OcNOS License Install Playbook

The ocnos-lic-install.yml playbook also uses a helper script to install a valid OcNOS license file from the HTTP server and places it on the OcNOS switch. Edits to the helper script may be required for your specific network details and the version of installer file being loaded. The playbook initiates the helper shell script with the command task. The OcNOS install script, `ocnos-lic-install`, connects to the OcNOS switch and requests the current license status. It then initiates a download of the OcNOS license file from the HTTP server to the target switch and then request the license status again to verify any changes. The playbook, script, and output are shown below.

ONIE Install Playbook: ocnos-lic-install.yml

```
---

- hosts: localhost
  user: root
  sudo: yes
  tasks:
  - name: Execute script
    command: '/bin/bash ./scripts/ocnos-lic-install'
```

The ocnos-lic-install script utilizes the linux expect scripting language to send and receive communication commands through a telnet session. The “expect” command may need to be installed on the linux control server if not available.

Helper Script: ocnos-lic-install

```
#!/bin/bash

#  Copyright (C) 2018 Ron Wilhelmson <ron.wilhelmson@deltaww.com>
#
#

##
## ocnos-lic-install
##
## History: 10Sep2018 Ron.Wilhelmson Initial Creation

# Dependencies: expect
#               valid ocnos license file

# Usage: ocnos-lic-install

# Example: ./ocnos-lic-install

switch="ocnos switch"
address="10.62.10.27"
port="22"
user="ocnos"
```

```
pass="ocnos"
license="http://10.62.10.22/IPI-00182330CDFA.bin"

VAR=$(
expect -c "
	set timeout 3
	spawn telnet -l "$user" "$address"
	expect \"Password:\"
	send \"$pass\n\"
	expect \"OcNOS>\"
	send \"show license\n\"
	expect \"OcNOS>\"
	send \"license get $license\n\"
	expect \"OcNOS>\"
	send \"show license\n\"
	expect \"OcNOS>\"
	send \"exit\n\"
	")

echo "$VAR"
```

Playbook Execution

The Ansible command to run ocnos-lic-install.yml playbook is as follows:

```
ansible-playbook ocnos-lic-install.yml -v
```

The resulting output is as follows:

```
root@DPR-LABVM-01:~/ansible# ansible-playbook ocnos-lic-install.yml -v
Using /root/ansible/ansible.cfg as config file

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

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

TASK [Execute script to install OcNOS License on switch]
*******************************************************************************
changed: [localhost] => changed=true
  cmd:
  - /bin/bash
  - ./scripts/ocnos-lic-install
  delta: '0:00:00.804873'
  end: '2018-09-10 23:53:13.863985'
  rc: 0
  start: '2018-09-10 23:53:13.059112'
  stderr: ''
  stderr_lines: []
  stdout: |-
    spawn telnet -l ocnos 10.62.10.27
    Trying 10.62.10.27...
    Connected to 10.62.10.27.
    Escape character is '^]'.
    Password:
    Last login: Tue Sep 11 03:47:40 UTC 2018 on pts/0
    Linux OcNOS 3.16.7-g63251bb-delta-agema7648 #1 SMP Fri May 4 05:49:27 UTC 2018
x86_64

    OcNOS version 1.3.4.260-OCNOS-DC-MPLS-ZEBM IPIRouter 08/26/2018 13:43:17
    OcNOS>show license
    License Type: Trial edition
    Remaining days to expire: 137 day(s)
    Node Identifier: 00182330CDFA
    Active license file:
    OcNOS>license get http://10.62.10.22/IPI-00182330CDFA.bin
    OcNOS>show license
    License Type: Trial edition
    Remaining days to expire: 137 day(s)
    Node Identifier: 00182330CDFA
    Active license file:
    OcNOS>
  stdout_lines: <omitted>

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

root@DPR-LABVM-01:~/ansible#
```

Example Playbooks and Projects:

The following git repository has example playbooks created for this guide and other applications that may be useful:

https://github.com/DeltaProducts/Getting-Started-with-Ansible

References:

Ansible User Guide
http://docs.ansible.com/ansible/devel/user_guide/intro_getting_started.html

Ansible Installation Guide

http://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

Ansible ansible-playbook Guide:

https://docs.ansible.com/ansible/2.4/ansible-playbook.html

Ansible Commands

http://docs.ansible.com/ansible/latest/modules/list_of_commands_modules.html

Ansible Modules

http://docs.ansible.com/ansible/latest/modules/list_of_all_modules.html

Tera Term Guide

https://ttssh2.osdn.jp/index.html.en

Open Network Install Environment (ONIE) Installation Guide

https://github.com/DeltaProducts/SolutionCenter/blob/master/ONIE%20recovery%20from%20bootable%20USB.pdf

Apache Web Server Setup Guide

https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-16-04