

ICOS 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 ICOS/Fastpath NOS configuration on Delta Networks switches.

Objective

The objective of this guide is to document the basic steps required to install a verified ICOS 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

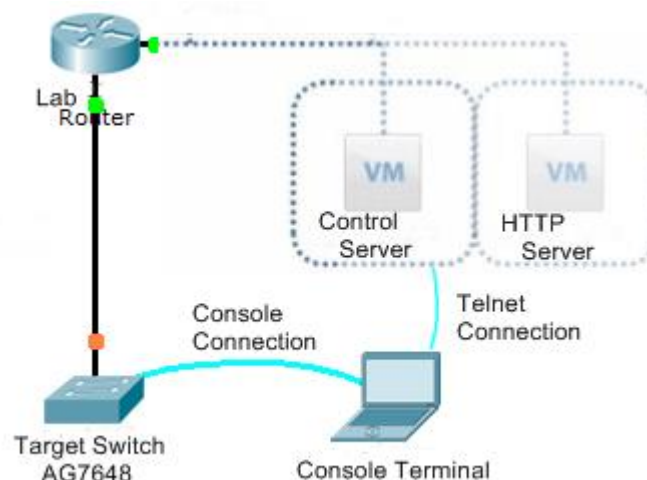
The basic systems required for building and running Ansible playbooks consist of the following:

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.

Ansible Directory and Files

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

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 ICOS 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 ICOS NOS installer.

ONIE Install Playbook

The following playbook along with a helper script and associated configuration file are utilized to initiate and run the ICOS 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 ICOS 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

# 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
    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 ""
          ;;
esac

echo "Waiting for onie to download and install"

sleep 60

echo "Rebooting switch"

/usr/bin/ssh -l root $target_switch /sbin/reboot
```

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
#                      icos

NOS_install=icos

# 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 will be run and shown on the control server through a terminal connection from you PC or directly if working on a workstation.

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": "0:01:15.439526",
"end": "2018-05-03 14:10:18.638688",
"rc": 0,
"start": "2018-05-03 14:09:03.199162",
"stderr": "",
"stderr_lines": [],
"stdout": "",
"Target switch set to: 10.62.10.40",
"",
"HTTP server set to: 10.62.10.22",
"",
"NOS to be installed on Target switch: icos",
"Sending install command to switch",
"Info: Fetching http://10.62.10.22/onie-icos-installer-x86_64-ag7648 ...",
"Connecting to 10.62.10.22 (10.62.10.22:80)",
"",
"installer          29% |*****| 63652k  0:00:02 ETA",
"installer          83% |*****| 174M  0:00:00 ETA",
"installer          100% |*****| 210M  0:00:00 ETA",
"",
"Waiting for onie to download and install",
"Rebooting switch"]}]

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:/ # discover: installer mode detected.
Stopping: discover... done.
ONIE: Executing installer: http://10.62.10.22/onie-icos-installer-x86\_64-ag7648
Verifying image checksum ... OK.
Preparing image archive ... OK.
ICOS Installer: platform: x86_64-ag7648-r0
INFO: Validating the image
INFO: Image is a valid image
```

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

```
GNU GRUB version 2.02~beta2+e4a1fe391

*ICOS
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 ICOS software completes loading the switch will boot into ICOS:

```
Booting 'ICOS'
Loading ICOS ...
Loading ICOS initial randisk ...
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Initializing cgroup subsys cpuacct
[ 0.000000] Linux version 3.16.0-29-generic (root@ubuntu-64bit-host.rtp.broadcom.com) (gcc version 4.6.3 (Ubuntu/Linaro 4.6.3-1ubuntu5)) #39 SMP Fri Jul 31 13:58:07
PDI 2015 (Ubuntu 3.16.0-29.39~14.04.1-generic 3.16.7-ckt2)
[ 0.000000] Command line: BOOT_IMAGE=/vmlinuz console=ttyS0,115200n8 acpi=off root=/dev/sda3 nodprobe.blacklist=igb,i2c-isnt,i2c-i801,i2c-dev
[ 0.000000] KERNEL supported cpus:
[ 0.000000] Intel GenuineIntel
[ 0.000000] AMD AuthenticAMD
```


The install completes when the following Ubuntu login screen is displayed. At this prompt you can access the ICOS environment with user ID **admin** and password **broadcom**:

```
Ubuntu 14.04 LTS localhost ttyS0
localhost login: admin
Password:
Last login: Wed May 2 20:30:36 UTC 2018 on ttyS0
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.16.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
admin@localhost:~$ sudo icos-cli
[sudo] password for admin:

Initializing console session. Press ^z to exit.

<localhost> #
<localhost> #show platform vpd
Operational Code Image File Name..... FastPath-ICOS-esw-ag7648-ag_x86_64-UB14R-CNRFS-BD6IOQhr3v2m2b6
Software Version..... 3.2.2.6
Timestamp..... Sat May 7 09:58:58 IST 2016
<localhost> #
```

The command “icos-cli” is used with the sudo password boradcom again to get to the FastPath prompt. The platform can be shown with the “show platform vpd” command as displayed above to verify ICOS/Faspath release.

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/Ron-delta/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>