
University of Colorado Boulder

Network Management and Automation

DevOps & Ansible

Levi Perigo, Ph.D.
University of Colorado Boulder
Department of Computer Science
Network Engineering

Review

- **Challenge**

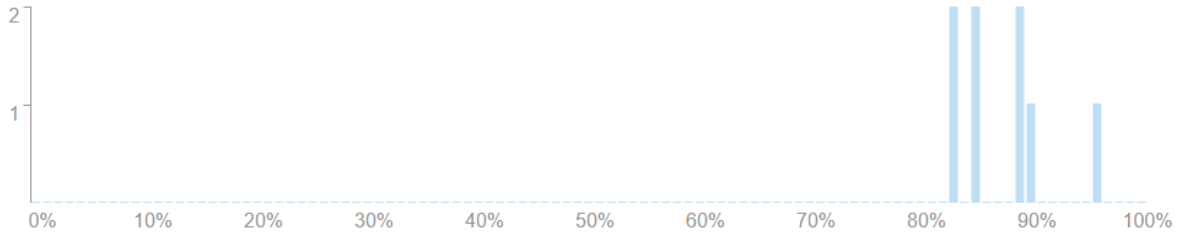
- **Review**
  - Network Automation
  - DevOps
  - NSOT
  - IAC
  - CI/CD
  - NMAS

Midterm Exam

Quiz Summary

Section Filter ▾ | Student Analysis | Item Analysis

Average Score	High Score	Low Score	Standard Deviation	Average Time
88%	96%	83%	4.9	01:10:18



Crisis of Confidence: Motivation

- It's never too late to be what you might have been.

- What do I want to do?

  - ***What do I want to become?***
    - When you know *who* you are, you'll know *what* to do.

Who do you want to be?

- **What do you need in order to be the person you want to be?**
- **What makes you feel balanced?**
- **What makes you come alive?**

What Matters Most?

Out of Balance

- **1. We're doing too many things**
- **2. We're not doing enough things**
- **3. We're doing the wrong things**
- **4. We're not doing the right things**

What is Out of Balance?

- **Tired – doesn't mean out of balance**
- **Stressed – doesn't mean out of balance**
- **Overwhelmed – doesn't mean out of balance**
  - Season, Week, Each Day
    - ***Believing everything needs to get done is keeping you from getting anything done.***

Prioritize

- **“Determine the order for dealing with (a series of items or tasks) according to their relative importance”**
  - Season, Week, Each Day
    - ***Believing everything needs to get done is keeping you from getting anything done.***

Time Management – The Path to Balance

- **1: Decide What Matters**
- **2: Stop Doing What Doesn't Matter**
- **3: Create a Schedule That Reflects What Matters**
- **4: Protect What Matters**
- **5: Be Present for What Matters**

Not enough time?

- **24x7 = 168 Hours in a week**
  - Subtract 10 hours of sleep per night
    - ***98 Hours***
      - Subtract 60 hours for Work/School
        - » **38 hours**
          - Subtract 5 hours for exercise per week

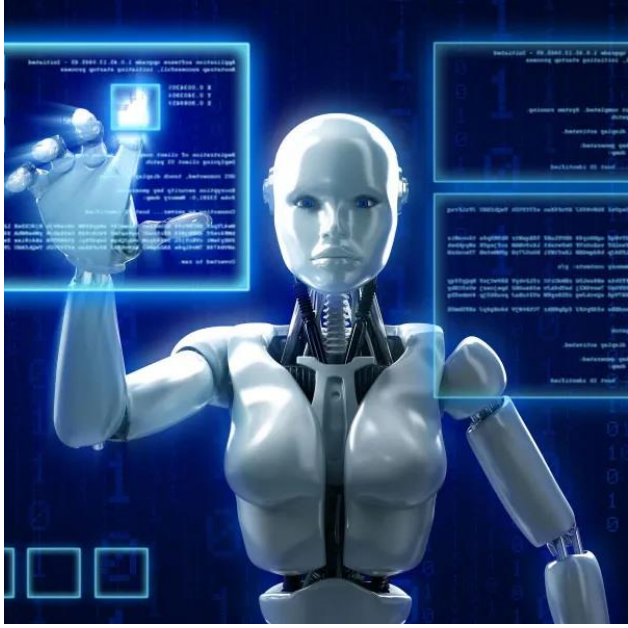
- **You have 33 hours of free time per week! (*almost 5 hours per day*)**
  - You don't have a time problem; you have a priority problem.

Network Automation & DevOps

- **What is the goal of automation?**
  - We want to be able to drive the network from a single point of control (IAC)
    - ***NMAS controls the network***
      - VLAN to a switch
      - Route to a route table
      - All configuration changes!
  - Versionable, Testable, Repeatable, Rapid Deployment
  - How do we achieve this?
    - ***VM out of band management***
      - Logically centralized, physically distributed
      - “single” point of control = single point of backup
    - ***Templates and Automation tools***

Diagram of DevOps/NMAS – Server – Staging - Production

DevOps Tools



- **Not required**
- **Very helpful**
- **Don’t replace people/brains**
- **Why is this so important?**
  - The robots run the network
    - ***We control the robots***

Methodologies

- **Agentless**
  - Example - Ansible
  - No additional application/software required
  - Relies on SSH to connect to devices
    - ***How do you automate this?***
  - Uses push from NMAS to switch

- **Agent-based**
  - Example - Puppet
  - Requires software on device to communicate to server
  - Device pulls from NMS
  - Not all devices support the agent

Push vs. Pull

- **Push**
  - A user process on server sends commands to the local system
  - What are the advantages/disadvantages?
    - ***Advantage***
      - Immediate remote execution
    - ***Disadvantage***
      - Manually push change (person)
        - » This can be automated
      - Clients have to be available

Push vs. Pull

- **Pull**
  - Client runs software/agent and queries NMS
  - What are advantages/disadvantages?
    - ***More scalable, but harder to control when clients are polling***
      - What is best? (30 secs, 3 min, 30 min, 3 hours)
    - ***Advantage – not done manually***
      - Timer expires – client queries “any updates?”
    - ***Disadvantage – not immediate***

Automation and DevOps Tools

- **SALTSTACK**
  - Agent/pull & push
    - ***Salt state (SLS)***
- **Ansible**
  - Agentless/push
    - ***Playbook***
- **Puppet**
  - Agent/pull
    - ***Manifest***
- **CHEF**
  - Agent/pull
    - ***Cookbook***
- **Network Automation and Programmability Abstraction Layer with Multivendor support (NAPALM)**
- **Modules from app store**
  - configuring specific task (IP address)


ANSIBLE

Installation of Ansible

- **Ubuntu VM = Ansible server (NMAS)**
  - pip
    - ***apt install python-pip***
    - ***pip install ansible***
      - Note: installs current version
  - apt
    - ***apt install ansible***
    - ***Ansible package for disto running***
      - Note: not necessarily getting newest version

- **No agent needed on client!**
  - Good for servers AND network devices!

Configuration Files

- **Tree structure**

- **Create Directory**
  - Name - Company name or project
  - Create files in directory
  - ansible.cfg file
    - ***Basic configuration file***
      - Starting point

```
playbook.yml
roles/
  > common/
    >>  files/
    >>  templates/
    >>  tasks/
    >>  handlers/
    >>  vars/
    >>  defaults/
    >>  meta/
```

ansible.cfg

- **Review this file**
  - /etc/ansible# cat ansible.cfg
- **All options are commented out**
  - This will give you an idea of what to change for the future

```
root@netman:/etc/ansible# cat ansible.cfg
# config file for ansible -- https://ansible.com/
# ===============================================

# nearly all parameters can be overridden in ansible-playbook
# or with command line flags. ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory or /etc/ansible/ansible.cfg, whichever it
# finds first

[defaults]

# some basic default values...

#inventory      = /etc/ansible/hosts
#library        = /usr/share/my_modules/
#module_utils   = /usr/share/my_module_utils/
#remote_tmp     = ~/.ansible/tmp
#local_tmp      = ~/.ansible/tmp
#plugin_filters_cfg = /etc/ansible/plugin_filters.yml
#forks          = 5
#poll_interval  = 15
#sudo_user      = root
#ask_sudo_pass = True
#ask_pass      = True
#transport      = smart
#remote_port    = 22
#module_lang    = C
#module_set_locale = False
```

ansible.cfg

- **cat /etc/ansible/ansible.cfg**

```
root@netman:/etc/ansible# cat ansible.cfg
# config file for ansible -- https://ansible.com/
# ===============================================

# nearly all parameters can be overridden in ansible-playbook
# or with command line flags. ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory or /etc/ansible/ansible.cfg, whichever it
# finds first

[defaults]

# some basic default values...

#inventory      = /etc/ansible/hosts
#library        = /usr/share/my_modules/
#module_utils   = /usr/share/my_module_utils/
#remote_tmp     = ~/.ansible/tmp
#local_tmp      = ~/.ansible/tmp
#plugin_filters_cfg = /etc/ansible/plugin_filters.yml
#forks          = 5
#poll_interval  = 15
#sudo_user      = root
#ask_sudo_pass = True
#ask_pass      = True
#transport      = smart
#remote_port    = 22
#module_lang    = C
#module_set_locale = False
```

- **[defaults]**
  - inventory = <file for hosts>
    - ***i.e. - (inventory = /etc/ansible/hosts)***
    - ***List of devices we are going to manage***
    - ***Can change from default of /etc/ansible/hosts***
  - host_key_checking=False
    - ***Lab environment – disables yes/no prompt from SSH***
      - Make sure crypto key generate is greater than 768 (picture below)
      - *If change key or change IP address of device will have to change SSH key on server (picture next slide)

```
learnodl@LearnODL:/etc/ansible$ ansible r1 -u a -m raw -a "show clock" -k -c ssh
SSH password:
192.168.134.101 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: ssh_rsa_verify: RSA modulus t
oo small: 512 < minimum 768 bits\r\nkey_verify failed for server_host_key\r\n",
    "unreachable": true
}
```

```
learnodl@LearnODL:/etc/ansible$ ansible r1 -u a -m raw -a "show clock" -k -c ssh
SSH password:
192.168.134.101 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\r\n@    WARNING: REMOTE HOST IDENTIFICATION HAS C
HANGED!     @\r\n@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\r\n
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!\r\nSomeone could be eavesd
ropping on you right now (man-in-the-middle attack)!\r\nIt is also possible that
 a host key has just been changed.\r\nThe fingerprint for the RSA key sent by th
e remote host is\n4e:6b:7d:d2:2a:f7:50:8c:43:7d:f4:07:ee:ca:ae:e1.\r\nPlease con
tact your system administrator.\r\nAdd correct host key in /home/learnodl/.ssh/k
nown_hosts to get rid of this message.\r\nOffending RSA key in /home/learnodl/.s
sh/known_hosts:3\r\n  remove with: ssh-keygen -f \"/home/learnodl/.ssh/known_hos
ts\" -R 192.168.134.101\r\nPassword authentication is disabled to avoid man-in-t
he-middle attacks.\r\nKeyboard-interactive authentication is disabled to avoid m
an-in-the-middle attacks.\r\nPermission denied (password).\r\n",
    "unreachable": true
}
```

Ansible Hosts File

- **Create groups**
  - Cisco, Arista, Juniper, Webservers, etc.
  - Called by adhoc or playbook
  - Why is this important?

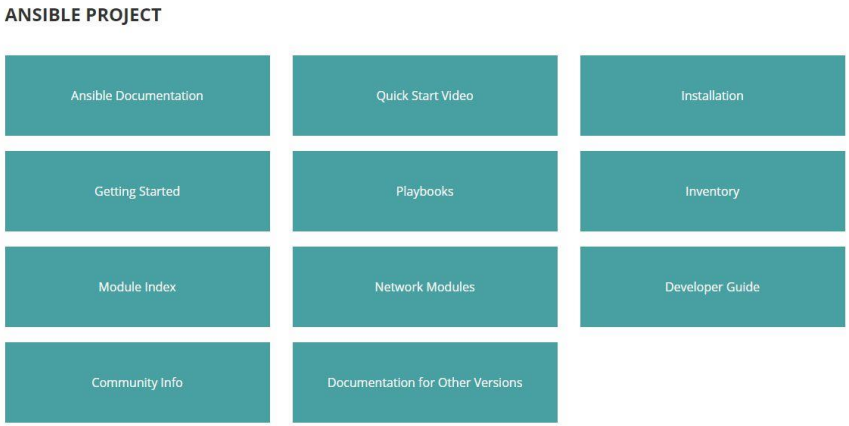
- **Assign IPs or domains/hostnames to group**

- **[myGroup]**
  - 10.1.1.1
  - router1
  - 2001:db8::1
  - raveninnovation.com

```
learnodl@LearnODL:/etc/ansible$ cat hosts3
[r1r2]
192.168.134.103
192.168.134.104
learnodl@LearnODL:/etc/ansible$ _
```

- **Can have nested groups**
  - [cisco]
    - ***[cisco-routers]***
    - ***[cisco-switches]***
    - ***[cisco-test-network-gns3]***

Ansible Modules



- **Core modules- https://docs.ansible.com/**
  - Basic tasks
    - ***Copy file from one system to another***
    - ***Changing user ID or password***
    - ***Restarting service***
  - Stored here (noted in ansible.cfg)
    - ***#library = /usr/share/my_modules/***

- **Ansible Galaxy**
  - App store
    - ***Install optional modules***
      - MySQL, Cumulus Linux, etc.

```
root@netman:/etc/ansible# cat ansible.cfg
# config file for ansible -- https://ansible.com/
# ===============================================

# nearly all parameters can be overridden in ansible-playbook
# or with command line flags. ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory or /etc/ansible/ansible.cfg, whichever it
# finds first

[defaults]

# some basic default values...

#inventory      = /etc/ansible/hosts
#library        = /usr/share/my_modules/
#module_utils   = /usr/share/my_module_utils/
#remote_tmp     = ~/.ansible/tmp
#local_tmp      = ~/.ansible/tmp
#plugin_filters_cfg = /etc/ansible/plugin_filters.yml
#forks          = 5
#poll_interval  = 15
#sudo_user      = root
#ask_sudo_pass = True
#ask_pass      = True
#transport      = smart
#remote_port    = 22
#module_lang    = C
#module_set_locale = False
```

Ansible Commands

- **Ad hoc - easy**
  - Single command
  - Must have correctly setup hosts file
    - ***AKA "inventory"***
- **Playbook – difficult and more difficult (scale)**
  - Series of commands
  - Stored as YAML file
    - ***The file extension can be either .yml or .yaml, but .yaml is preferred***
  - Difficult - Tasks, variables, etc. in a single file
    - ***More difficult - breaking apart tasks into subfolders/files***
- **Ansible Output - Color Coded Legend**
  - Red = Bad / unreachable / failed
  - Green = Good / reachable / worked
  - Orange = something changed / action taken

Ansible Commands – Ad Hoc

- **Retrieve commands from NMAS to all devices**
- **-u**
  - username
- **-m**
  - Module
    - ***Typically "command" or "raw" or "shell"***
- **-k**
  - Prompting for SSH password
    - ***If SSH keys are not pre-installed on server***
      - Can be used for ZTP
- **-a**
  - Specifies arguments used
    - ***Typically commands "show clock" or "cat /etc/network/interfaces"***
- **-i**
  - Specify hosts file other than default
    - ***"ansible –i hosts2 r1 …"***

```
learnodl@LearnODL:/etc/ansible$ ansible r1 -u a -m raw -a "show clock" -k -c ssh
SSH password:
192.168.134.103 | SUCCESS | rc=0 >>

*01:12:42.271 UTC Fri Mar 1 2002channel_by_id: 2: bad id: channel free
client_input_channel_req: channel 2: unknown channel
Shared connection to 192.168.134.103 closed.
Connection to 192.168.134.103 closed by remote host.
```

Playbook – Linux Example (basic template)

- **YAML**
  - Spacing is specific (like Python)
    - ***Spacing vs tabs vs indentation etc.***

- **To run Playbook**
  - ansible-playbook simple-playbook.yaml

simple-playbook.yml

```
---
- hosts: leaf1
  vars:
    loopback_ip: "10.2.1.1"
  remote_user: root
  tasks:
  - name: write the networking config file
    template: src=interfaces.j2 dest=/etc/network/interfaces
    notify:
    - restart networking
  - name: ensure networking is running
    service: name=networking state=started
  handlers:
    - name: restart networking
      service: name=networking state=restarted
```

Interfaces.j2 (template file)

```
#this file has been configured by ansible

auto eth0
iface eth0 inet dhcp

auto lo
iface lo inet loopback
  address {{loopback_ip}}/32
```

Playbook Example

```
PLAY [leaf1] ********************************************************************

GATHERING FACTS *****************************************************************
ok: [leaf1]

TASK: [write the networking config file] ****************************************
changed: [leaf1]

TASK: [ensure networking is running] ********************************************
changed: [leaf1]

NOTIFIED: [restart networking] **************************************************
changed: [leaf1]

PLAY RECAP **********************************************************************
leaf1                      : ok=4    changed=3    unreachable=0    failed=0
```

- Gathering facts: uses core module to login to switch, and gets stats, such as loopback info, because that is what is going to be changed
  - Can view all of these tasks with the “ansible –m setup”

Playbook IOS

```
learnodl@LearnODL:/etc/ansible$ cat hosts3
[r1r2]
192.168.134.103
192.168.134.104
learnodl@LearnODL:/etc/ansible$ _
```

```
---
creds:
  username: a
  password: a
learnodl@LearnODL:/etc/ansible$
learnodl@LearnODL:/etc/ansible$ cat ios.yaml
---
- hosts: r1r2
  connection: local

  tasks:
  - name: OBTAIN LOGIN CREDENTIALS
    include_vars: secrets.yaml

  - name: DEFINE PROVIDER
    set_fact:
      provider:
        host: "{{ inventory_hostname }}"
        username: "{{creds['username'] }}"
        password: "{{creds['password'] }}"

  - name: RUN 'SHOW VERSION'
    ios_command:
      provider: "{{ provider }}"
      commands:
        - show version
    register: version

  - debug: var=version.stdout_lines
learnodl@LearnODL:/etc/ansible$
```

```
learnodl@LearnODL:/etc/ansible$ ansible-playbook -i hosts3 ios.yaml

PLAY [r1r2] ********************************************************************

TASK [setup] *******************************************************************
ok: [192.168.134.103]
ok: [192.168.134.104]

TASK [OBTAIN LOGIN CREDENTIALS] ************************************************
ok: [192.168.134.103]
ok: [192.168.134.104]

TASK [DEFINE PROVIDER] *********************************************************
ok: [192.168.134.103]
ok: [192.168.134.104]

TASK [RUN 'SHOW VERSION'] ******************************************************
changed: [192.168.134.103]
changed: [192.168.134.104]

PLAY RECAP *********************************************************************
192.168.134.103            : ok=4    changed=1    unreachable=0    failed=0
192.168.134.104            : ok=4    changed=1    unreachable=0    failed=0
```

```
---
creds:
  username: a
  password: a
learnodl@LearnODL:/etc/ansible$
learnodl@LearnODL:/etc/ansible$ cat ios.yaml
---
- hosts: r1r2
  connection: local

  tasks:
  - name: OBTAIN LOGIN CREDENTIALS
    include_vars: secrets.yaml

  - name: DEFINE PROVIDER
    set_fact:
      provider:
        host: "{{ inventory_hostname }}"
        username: "{{creds['username'] }}"
        password: "{{creds['password'] }}"

  - name: RUN 'SHOW VERSION'
    ios_command:
      provider: "{{ provider }}"
      commands:
        - show version
    register: version

  - debug: var=version.stdout_lines
learnodl@LearnODL:/etc/ansible$
```

Summary

- **Ansible Config File**
- **Ansible Hosts File**
- **Adhoc vs. Playbook**
  - Playbook .yaml
- **Templates**
- **Next Lab = Python, Ansible, Templates**

Questions?