Malaysian Open Source Conference 2017

(The) Multi Facets of the Open Source Tools

Muhammad Najmi Ahmad Zabidi

About me

- Linux Administrator, End Point Corporation (remote staff from home)
- Holds a Master degree from USM, Penang (grad 2007) in Computer Science

Rough idea about today

- Targeting students/normal user without intense experience in system admin
- Getting more organized in work for file management
- How to save time to work with multi machines, automate where possible

What I will cover today

- Tmux/screen
- Git
- AIDE
- Ansible

tmux

- Terminal multiplexer
- Allow us to send session to the background
- We could resume our work if the connection was bad
- Supports keybinding

Gnu screen

- Almost similar with tmux
- Used to be my favorite

git

- For versioning purpose
  - System admin could use this to give “versions” to your file changes (especially text file)
  - Probably not so advanced for system admins, but very useful to revert/track changes

Git – possible use for system admins

- Tracking DNS zone file changes – serial, CNAME, A/AAAA, TXT, MX records etc
- Tracking configuration file changes for network monitoring tools, IDS, webservers etc

Git basics

- git init - initialize repo
- git checkout - changing branch
- git pull - taking files from remote sources
- git clone - “cloning” remote resource
- git cherry-pick - to import certain feature

AIDE (for file change monitoring)

- File changes monitoring is part of PCI/DSS compliance (a concern for e-commerce business since they’re dealing with credit card details)
- We can include/exclude folder/file to monitored

AIDE config

- Located in /etc/aide.conf (for Centos)
- Have to use regular expression to exclude/include files or folders

AIDE – first time execution

- Use aide - - init for the first time execution (this will take time since AIDE will generate a first database as a base

```
[root@mosc-centos aide]# mv aide.db.new.gz aide.db.gz

[root@mosc-centos aide]# aide
AIDE 0.15.1 found differences between database and filesystem!!
Start timestamp: 2017-05-18 00:38:13

Summary:
  Total number of files: 70090
  Added files:           10
  Removed files:         0
  Changed files:         1
```

```
added: /home/najmi
added: /home/najmi/.bash_history
added: /home/najmi/.bash_logout
added: /home/najmi/.bash_profile
added: /home/najmi/.bashrc
added: /home/najmi/.cache
added: /home/najmi/.cache/abrt
added: /home/najmi/.cache/abrt/lastnotification
added: /home/najmi/.config
added: /home/najmi/.config/abrt

---------------------------------------------------
Changed files:
---------------------------------------------------

changed: /etc/aide.conf

---------------------------------------------------
Detailed information about changes:
---------------------------------------------------


File: /etc/aide.conf
 SHA256   : OQih9JPr8QgVKdLVibqiB5sZRhzZZjVA , Y0BGtPqGb/qW2W3Gq5m6qz+TPJjQL5Km
```

AIDE - Tips

- Well, probably not a good idea to monitor /home, as it is expected to have changes every second we’re using it
- But then, it depends on the usage though

Ansible

What is Ansible?

- A configuration management tool
- Heavily depends on a SSH connection
- Uses SSH public key
- Previously uses python-paramiko package
- “agentless” - hence we don’t need to install any daemon on the client side
- since it’s agentless, hence it’s a “push-based tool”

SSH

- Create SSH keypairs first
  - ssh-keygen -t rsa (or dsa)
- Keep the private key in the Ansible head
- Put the public key on the target’s $HOME/.ssh/authorized_keys

- Declare hosts in /etc/ansible/hosts
- We can use range with [var:var] format
- Put label with […] format

Example: “ping” module

```
root@mosc-ubuntu:~# ansible all -m ping
192.168.56.103 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
192.168.56.101 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
192.168.56.104 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Running command

```
root@mosc-ubuntu:~# ansible all -a "/bin/echo Hi"
192.168.56.103 | SUCCESS | rc=0 >>
Hi

192.168.56.104 | SUCCESS | rc=0 >>
Hi

192.168.56.101 | SUCCESS | rc=0 >>
Hi
```

Run from specific hosts

```
root@mosc-ubuntu:~# ansible debian -a "/bin/echo Hi"
192.168.56.103 | SUCCESS | rc=0 >>
Hi


root@mosc-ubuntu:~# ansible centos -a "/bin/echo Hi"
192.168.56.101 | SUCCESS | rc=0 >>
Hi
192.168.56.104 | SUCCESS | rc=0 >>
Hi
```

```
root@mosc-ubuntu:~# cat /etc/ansible/hosts
192.168.56.101
192.168.56.10[3:4]

#ubuntu is the ansible's head, so we omit
this one
#[ubuntu]
#192.168.56.102

[debian]
192.168.56.103

[centos]
192.168.56.101
192.168.56.104
```

ansible-playbook

- Uses an input file (YAML format) to execute commands

Ansible for package management

- In this example I will show how to use it with apt (Debian flavor) and yum (RedHat flavor)

```
---
- hosts: debian
  tasks:
    - name: Installs nginx web server
      apt: pkg=nginx state=installed update_cache=true
      notify:
        - start nginx

  handlers:
    - name: start nginx
      service: name=nginx state=started
root@mosc-ubuntu:~/ansible-scripts# ansible-playbook nginx.yml

PLAY
****************************************************************
***********

TASK [setup]
****************************************************************
***
ok: [192.168.56.103]
```

Install httpd in Centos

```
---
- hosts: centos
  tasks:
    - name: Install httpd web server
      yum:  pkg=httpd state=latest
      notify:
        - start httpd

  handlers:
    - name: start httpd
      service: name=httpd state=started
```

```
ansible-playbook httpd-centos.yml

PLAY ***************************************

TASK [setup] ***************************************
ok: [192.168.56.104]
ok: [192.168.56.101]

TASK [Install httpd web server]
************************************************
changed: [192.168.56.104]
changed: [192.168.56.101]

RUNNING HANDLER [start httpd]
**************************************************
changed: [192.168.56.104]
changed: [192.168.56.101]

PLAY RECAP ****************************************
192.168.56.101             : ok=3    changed=2    unreachable=0
   failed=0
192.168.56.104             : ok=3    changed=2    unreachable=0
   failed=0
```

Ansible for file copy

copy-file.yml

```
- hosts: all

  tasks:
   - copy:
       src: /root/files/test.conf
       dest: /root/target/target-recieved.conf
       owner: root
       group: root
       mode: 0644
```

Failed! (targer dir is not exist)

```
TASK [copy] ***************************************************************

fatal: [192.168.56.103]: FAILED! => {"changed": false, "checksum": 
"04d06159a29826346c1fd76a889d49c1b7d825d5", "failed": true, "msg": "Destination directory 
/root/target does not exist"}

fatal: [192.168.56.104]: FAILED! => {"changed": false, "checksum": 
"04d06159a29826346c1fd76a889d49c1b7d825d5", "failed": true, "msg": "Destination directory 
/root/target does not exist"}

fatal: [192.168.56.101]: FAILED! => {"changed": false, "checksum": 
"04d06159a29826346c1fd76a889d49c1b7d825d5", "failed": true, "msg": "Destination directory 
/root/target does not exist"}


PLAY RECAP ****************************************************************

192.168.56.101             : ok=1    changed=0    unreachable=0    failed=1

192.168.56.103             : ok=1    changed=0    unreachable=0    failed=1

192.168.56.104             : ok=1    changed=0    unreachable=0    failed=1
```

```
ansible -a "mkdir /root/target/" all

192.168.56.103 | SUCCESS | rc=0 >>


192.168.56.101 | SUCCESS | rc=0 >>


192.168.56.104 | SUCCESS | rc=0 >>
```

```
 ansible-playbook copy-file.yml


PLAY
***************************************************************************

TASK [setup]
*******************************************************************
ok: [192.168.56.103]
ok: [192.168.56.101]
ok: [192.168.56.104]

TASK [copy]
********************************************************************
changed: [192.168.56.103]
changed: [192.168.56.101]
changed: [192.168.56.104]

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

192.168.56.103             : ok=2    changed=1    unreachable=0    failed=0

192.168.56.104             : ok=2    changed=1    unreachable=0    failed=0
```

Install AIDE (Centos)

```
 ansible-playbook install-aide-centos.yml

PLAY
*******************************************************************

TASK [setup]
*******************************************************************
ok: [192.168.56.104]
ok: [192.168.56.101]

TASK [Install AIDE daemon]
*****************************************************
changed: [192.168.56.104]
changed: [192.168.56.101]

PLAY RECAP
*********************************************************************
192.168.56.101             : ok=2    changed=1    unreachable=0
failed=0
192.168.56.104             : ok=2    changed=1    unreachable=0
failed=0
```

```
---
- hosts: debian
  tasks:
    - name: Installs AIDE daemon
      apt: pkg=aide state=installed
update_cache=true
```

Install AIDE (Debian based)

```
ansible-playbook install-aide-debian.yml

PLAY
*******************************************************************
*********

TASK [setup]
*******************************************************************
*
ok: [192.168.56.103]

TASK [Installs AIDE daemon]
*****************************************************
changed: [192.168.56.103]

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

Condition statement in Ansible

- Ansible could be used with multiple types of distro too, but we have to have a condition check on the remote distro
- This will save time to run a script once only rather than typing them multiple times

Multiple distro check

```
---
- hosts: all

  tasks:
  - apt: name=$item state=latest
    with_items:
     - ntp
    when: ansible_distribution == 'Debian' or ansible_distribution == 'Ubuntu'

  - yum: name=$item state=latest
    with_items:
     - ntp
    when: ansible_distribution == 'CentOS' or ansible_distribution == 'Red Hat Enterprise Linux'

  - service: name=ntpd state=started enabled=yes
```

* adapted from : https://raymii.org/s/tutorials/Ansible_-_Only_if_on_specific_distribution_or_distribution_version.html

END

Feel free to e-mail najmi.zabidi@gmail.com or najmi@endpoint.com