Ansible Deployment Guide for Check Point

Ryan Rasmuss
https://github.com/ryanrasmuss

Prerequisites

1. Have an Ubuntu server up and running with internet access
2. Have a R80.10 (or higher) Check Point Management Server ready with first time wizard completed.
3. Make sure the Ubuntu server (our Ansible Server) has connectivity to the Checkpoint Management Server (i.e can ping and ssh one another)
4. The text in **bold** represents directories, files, or commands to be run.
5. Please set bash as the default shell on the Management server with **set user admin shell /bin/bash** (remember to **save config**)
6. This guide assumes the reader knows how to edit files in Linux via vi/vim/nano/emacs/whatever.
7. This guide assumes the reader knows basic Linux commands such as **mv, cp, ssh,** etc..

Install Ansible & Supporting Software

1. You will need a machine that supports Ansible. In my case, I am running Ubuntu 16.04.4 LTS 64-bit.
2. Run the following commands (apt or yum depending on your distro):

a. sudo apt-get update
b. sudo apt-get upgrade
c. sudo apt-get install openssh-server
d. sudo apt-get install software-properties-common
e. sudo apt-add-repository ppa:ansible/ansible
f. sudo apt-get install ansible
g. sudo apt-get install git
h. sudo apt-get install python2.7

Your Ubuntu Machine is now an unconfigured Ansible Server and will now be referred to as the Ansible Server for the rest of the guide. For more deployment options refer to: http://docs.ansible.com/ansible/latest/intro_installation.html

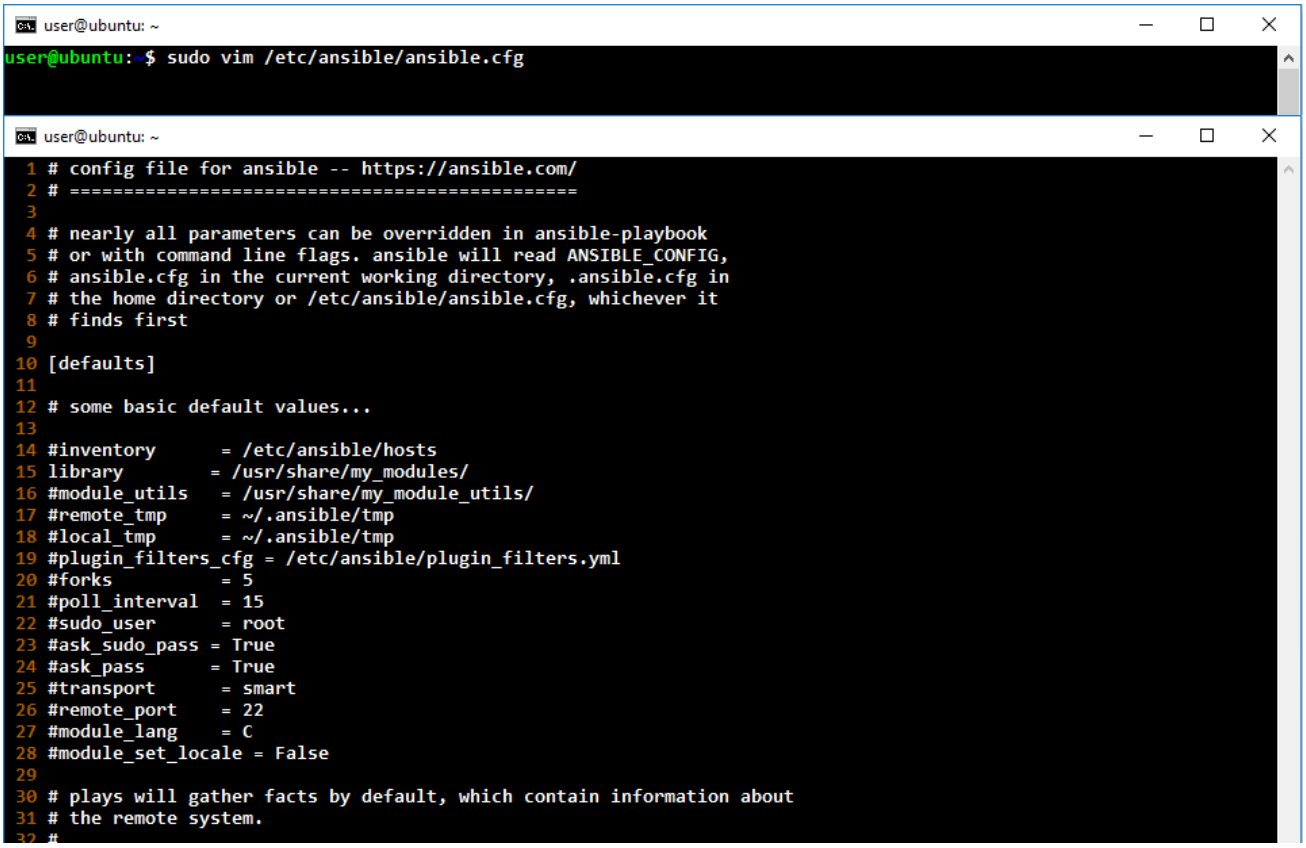
Setup public-key authentication

1. Ansible requires pub-key authentication, so let's generate a key.
2. On the Ansible server (Ubuntu server) , run: **ssh-keygen -t rsa -b 4096** and hit [Enter] for all prompts
3. Now, in **~/.ssh/** you will find an **id_rsa** and **id_rsa.pub**
4. Do not share **id_rsa**

5. Run **ssh-copy-id [ansible_user_name]@127.0.0.1** and reply with yes and enter your password.
6. Run **ssh [ansible_user_name]@127.0.0.1** you should not be prompted a password. You are now done with this section.
7. (Optional) On the Ansible Server, run **ssh-copy-id admin@[ip_of_mgmt_server]** and you will be able to ssh from the Ansible Server to the Management Server without a password prompt.

Install the Check Point Python API SDK & Ansible Module

1. Make sure you have python2.7 installed on the Ubuntu Machine. You can check by calling **python2.7**. Otherwise, install python2.7 via **sudo apt-get install python2.7**.
2. Find out where your **library** directory is located for Ansible. Open your **/etc/ansible/ansible.cfg** (an image of said ansible.cfg is located below). Uncomment the library line (remove # at the front) and make note of the path. In the image below, our **library** is **/usr/share/my_modules/** (remember to uncomment and **save the edit**)



*** Note how **library** does not have a '#' in the front of library. This tells Ansible that it's library folder is located at **/usr/share/my_modules/** ; you can change the directory location if you feel the need.

3. Run: **git clone --recursive** https://github.com/CheckPoint-APIs-Team/cpAnsible on your Ansible Server (if no git then get it via **sudo apt-get install**

git)

```
user@ubuntu: ~
user@ubuntu:~$ git clone --recursive https://github.com/CheckPoint-APIs-Team/cpAnsible
Cloning into 'cpAnsible'...
remote: Counting objects: 105, done.
remote: Total 105 (delta 0), reused 0 (delta 0), pack-reused 105
Receiving objects: 100% (105/105), 29.39 KiB | 0 bytes/s, done.
Resolving deltas: 100% (59/59), done.
Checking connectivity... done.
Submodule 'check_point_mgmt/cp_mgmt_api_python_sdk' (https://github.com/CheckPoint-APIs-Team/cp_mgmt_api_python_sdk)
registered for path 'check_point_mgmt/cp_mgmt_api_python_sdk'
Cloning into 'check_point_mgmt/cp_mgmt_api_python_sdk'...
remote: Counting objects: 75, done.
remote: Total 75 (delta 0), reused 0 (delta 0), pack-reused 75
Unpacking objects: 100% (75/75), done.
Checking connectivity... done.
Submodule path 'check_point_mgmt/cp_mgmt_api_python_sdk': checked out 'dff38818388a1a7854f9058fa89d908ce85f4fb5'
user@ubuntu:~$
```

4. You should now have a **cpAnsible/** directory in your working directory
5. Check the contents of this directory, in **cpAnsible** you should see **check_point_mgmt/** and two important file/directory: **check_point_mgmt.py** and **cp_mgmt_api_python_sdk**. If you don't see **cp_mgmt_api_python_sdk**, you can download the Check Point API Python SDK via **git clone** **https://github.com/CheckPointSW/cp_mgmt_api_python_sdk.git**

```
user@ubuntu: ~
user@ubuntu:~$ tree cpAnsible/
cpAnsible/
├── check_point_mgmt
│   ├── check_point_mgmt.py
│   └── cp_mgmt_api_python_sdk
│       ├── add_access_rule
│       │   └── add_access_rule.py
│       ├── clone_host
│       │   └── clone_host.py
│       ├── discard_sessions
│       │   └── discard_sessions.py
│       ├── find_duplicate_ip
│       │   └── find_dup_ip.py
│       ├── __init__.py
│       ├── lib
│       │   ├── api_exceptions.py
│       │   ├── api_response.py
│       │   ├── __init__.py
│       │   └── mgmt_api.py
│       ├── LICENSE
│       └── README.md
├── LICENSE
├── Playbooks
│   ├── demo-playbook.yml
│   └── tutorial-playbook.yml
└── README.md

8 directories, 16 files
user@ubuntu:~$
```

6. Run **sudo mkdir -v /usr/share/my_modules**
7. Move the **cpAnsible/check_point_mgmt/check_point_mgmt.py** file to your library via **sudo mv cpAnsible/check_point_mgmt/check_point_mgmt.py /usr/share/my_modules/**

```
user@ubuntu: ~
user@ubuntu:~$ sudo mkdir -v /usr/share/my_modules
mkdir: created directory '/usr/share/my_modules'
user@ubuntu:~$ sudo mv -v cpAnsible/check_point_mgmt/check_point_mgmt.py /usr/share/my_modules/
'cpAnsible/check_point_mgmt/check_point_mgmt.py' -> '/usr/share/my_modules/check_point_mgmt.py'
user@ubuntu:~$
```

8. **ls /usr/share/my_modules/** should list **check_point_mgmt.py**
9. Move the **cp_mgmt_api_python_sdk** directory into **/usr/lib/python2.7/** via **sudo mv -v cpAnsible/check_point_mgmt/cp_mgmt_api_python_sdk /usr/lib/python2.7/**

```
user@ubuntu: ~
user@ubuntu:~$ sudo mv -v cpAnsible/check_point_mgmt/cp_mgmt_api_python_sdk/ /usr/lib/python2.7/
'cpAnsible/check_point_mgmt/cp_mgmt_api_python_sdk/' -> '/usr/lib/python2.7/cp_mgmt_api_python_sdk'
```

10. **ls /usr/lib/python2.7/** should list **cp_mgmt_api_python_sdk/**
11. You should be able to invoke python2.7 and import the package like this:

a. Call **python2.7**

b. In the interpreter >> **import cp_mgmt_api_python_sdk**
c. If no error comes up (i.e nothing happens) we are good (ctrl-d to exit)

```
user@ubuntu:~$ python2.7
Python 2.7.12 (default, Dec  4 2017, 14:50:18)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cp_mgmt_api_python_sdk
>>>
```

Take a break. Go get a coffee or something.

Let's make an inventory file

1. Ansible has something called an inventory file, which contains the information of the machines you want managed.
2. It is located: **/etc/ansible/hosts**
3. Add the following lines at the end of the hosts file:

```
23 ## 192.168.1.100
24 ## 192.168.1.110
25
26 # If you have multiple hosts following a pattern you can specify
27 # them like this:
28
29 ## www[001:006].example.com
30
31 # Ex 3: A collection of database servers in the 'dbservers' group
32
33 ## [dbservers]
34 ##
35 ## db01.intranet.mydomain.net
36 ## db02.intranet.mydomain.net
37 ## 10.25.1.56
38 ## 10.25.1.57
39
40 # Here's another example of host ranges, this time there are no
41 # leading 0s:
42
43 ## db-[99:101]-node.example.com
44 [sec-checkup-appliance]
45 127.0.0.1
46 [sec-checkup-appliance:vars]
47 ansible_user=user
48 mgmt_server=192.168.1.1
49 appliance_name=Sec-Checkup
50 ansible_python_interpreter=/usr/bin/python2.7
51 mgmt_user=admin
52 mgmt_password=yourmgmtserverppassword
53 fingerprint=BE:BC:4F:A4:E3:6C:C8:13:CF:E8:A0:B4:B4:08:3E:A7:24:E0:A8:C7
                                                              53,71         Bot
```

Where…
[sec-checkup-appliance] name you want to give your management server for Ansible to use
Note that with Ansible and Checkpoint, the IP address is always 127.0.0.1.
[sec-checkup-appliance:vars] this is how ansible ties variables to the Management Server
ansible_user=(your ansible server's username)
mgmt_server=(ip address of your management server)
mgmt_user=(username of your management server)

mgmt_password=(password of management server)*
fingerprint=(fingerprint of the management server)** **(info on fingerprint provided below)**
Don't mess up any spelling here, or you will get awful errors and waste a great deal of time debugging.

* Make sure the **/etc/ansible/hosts** is secured with permissions. A breached ansible server means a breached management server!
** To get the fingerprint you have two options:

1. Run **api fingerprint** on the management server and copy manually
2. Or, use my script provided at https://gist.github.com/cp-ryan/7a4ef7693ccb9b5fd72e9c7f7347acb5

a. **git clone https://gist.github.com/7a4ef7693ccb9b5fd72e9c7f7347acb5.git**
b. **cd 7a4ef7693ccb9b5fd72e9c7f7347acb5/**
c. Run **chmod 700 get_fingerprint.sh**
d. (Optional) Run **ssh-copy-id [mgmt_user]@[mgmt_server_ip]** enter yes and provide password (this sets up keyless ssh & makes things faster)
e. Before running the **get_fingerprint.sh** script, make sure that the Management Server's default shell is bash. For example, I'm using **admin** and I used **set user admin shell /bin/bash** with **save config**. The script will not work otherwise. You can change your default shell back to clish after running the script.
f. Run **sudo ./get_fingerprint.sh [mgmt_user_name] [mgmt_ip_address]**
g. The script will leave behind a fingerprint.txt file in working directory and automatically append the fingerprint line in your **/etc/ansible/hosts**

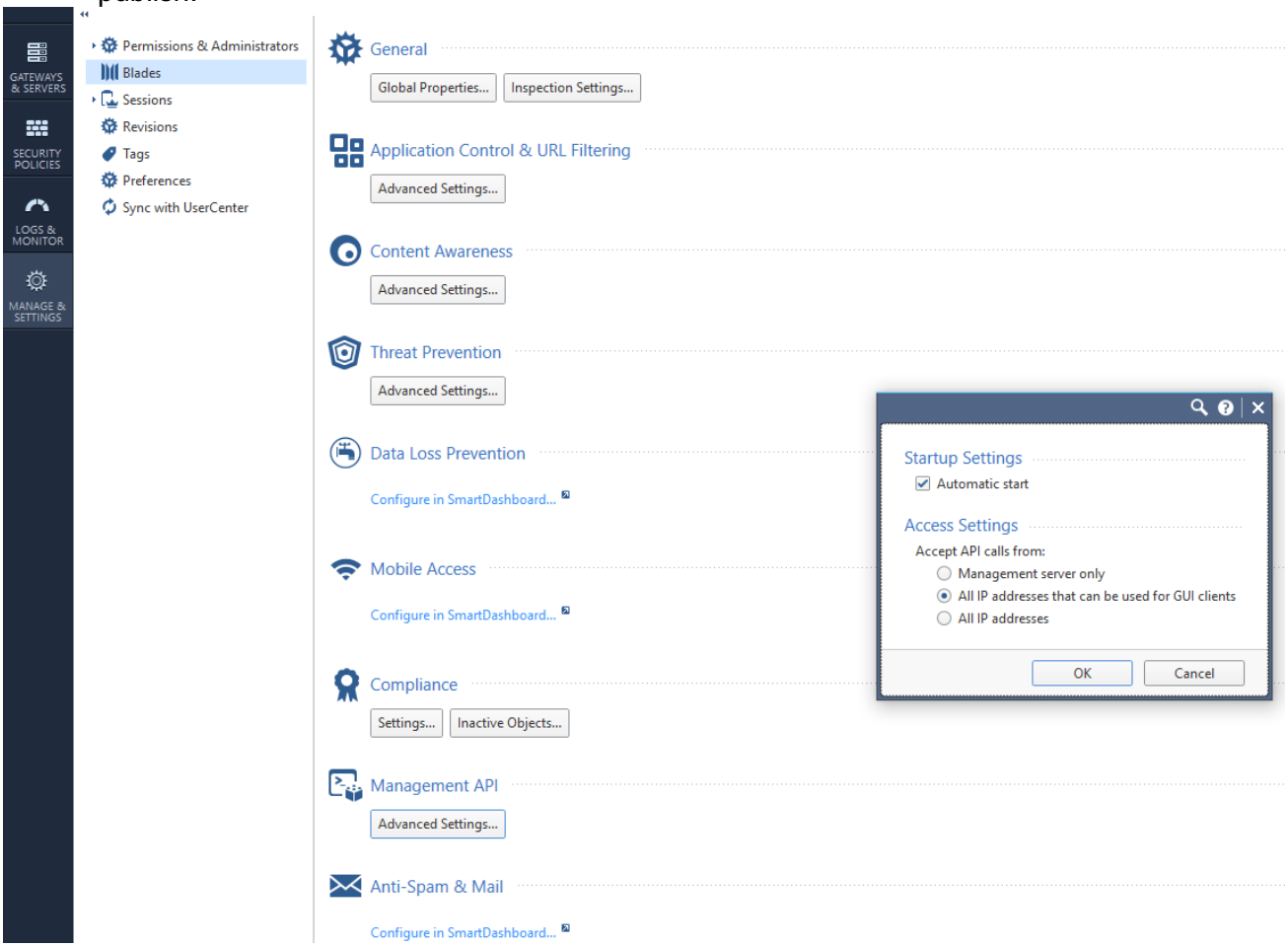
The next page has another screenshot of my **/etc/ansible/hosts** file.

```
42
43 ## db-[99:101]-node.example.com
44 [mgmt-server]
45 127.0.0.1
46 [mgmt-server:vars]
47 ansible_user=user
48 mgmt_server=192.168.26.100
49 appliance_name=Mgmt
50 ansible_python_interpreter=/usr/bin/python2.7
51 mgmt_user=admin
52 mgmt_password=vpn123
53 fingerprint=AE:99:6E:8D:FE:6D:DA:37:3C:68:63:4D:4E:E3:13:5A:68:F3:89:A2
```

Test Connectivity & Ansible Playbooks

- First, make sure you have basic connectivity from the Ansible Server to the Management Server. From the Ansible Server, ping the Management Server.
- On the Ansible Server, run **ssh [ansible_user_name]@127.0.0.1** which should log you into the ansible server without a password prompt. (If there is a problem refer to the "Setup public-key authentication" section of this guide) Be sure to run **exit**.
- Copy the **ansible-mgmt-test.playbook** to your Ansible Server (can be found here: https://gist.github.com/cp-ryan/71442b35e6e8d830e19f776a7b1b6cc8)
- Or copy the playbook to ansible server via **git clone https://gist.github.com/71442b35e6e8d830e19f776a7b1b6cc8.git**
- Run **cd 71442b35e6e8d830e19f776a7b1b6cc8/**

- Before running the playbook, make sure the Management Server settings are correct and publish.

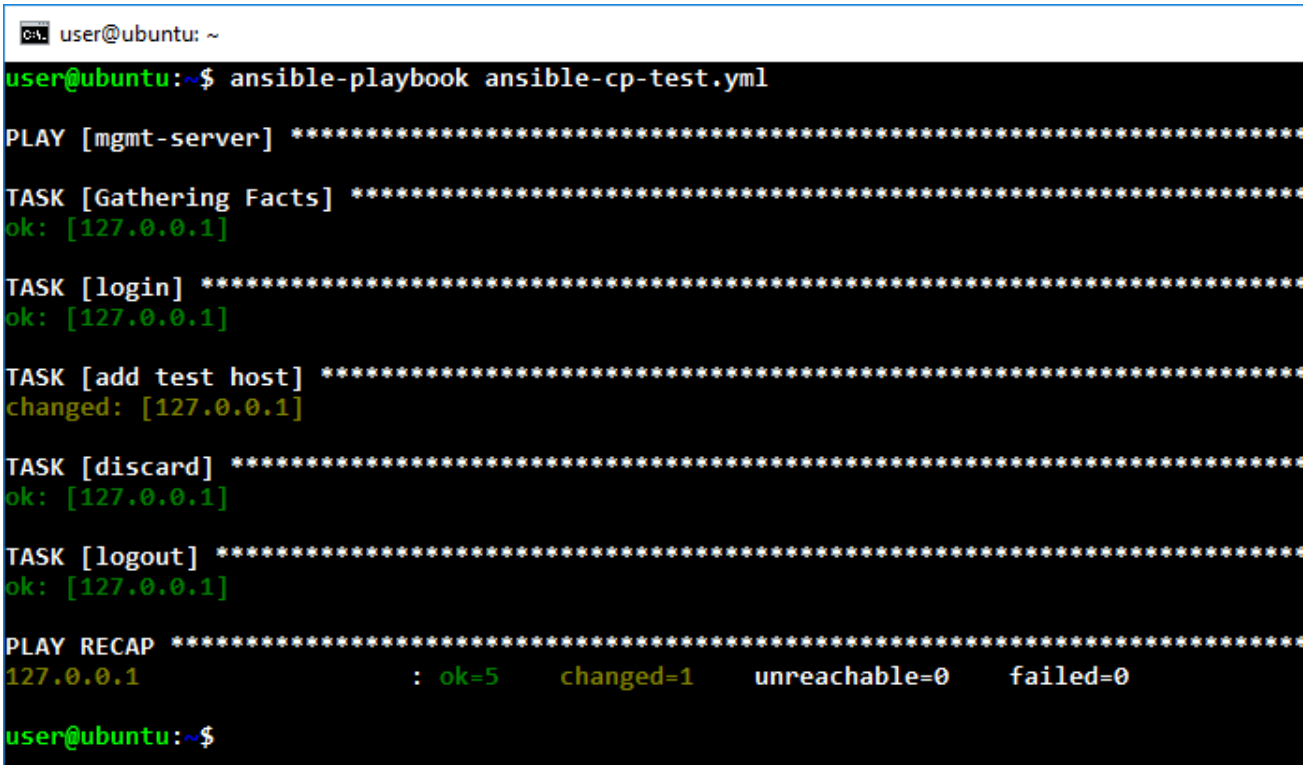


- Run **api status** on the management server to verify the api server is running; otherwise call **api restart** (run **api restart** just to be safe)
- Open the **ansible-cp-test.yml** and make sure the “hosts” section (line 2) matches the name given in in your **/etc/ansible/hosts** file
- For example, my **/etc/ansible/hosts** contains **[mgmt-server]** and **[mgmt-server:vars]** which matches the line 2 in my **ansible-cp-test.yml** file

user@ubuntu: ~/71442b35e6e8d830e19f776a7b1b6cc8

```
---
- hosts: mgmt-server
  tasks:
  - name: "login"
    check_point_mgmt:
      command: login
      parameters:
        username: "{{mgmt_user}}"
        password: "{{mgmt_password}}"
        management: "{{mgmt_server}}"
      fingerprint: "{{fingerprint}}"
    register: login_response
  - name: "add test host"
    check_point_mgmt:
      command: add-host
      parameters:
        name: "test-123abc"
        ip-address: "1.1.1.2"
        color: "red"
      session-data: "{{login_response}}"
  - name: "discard"
    check_point_mgmt:
      command: discard
      session-data: "{{login_response}}"
  - name: "logout"
    check_point_mgmt:
      command: logout
      session-data: "{{login_response}}"
```

- Run **ansible-playbook ansible-cp-test.yml** The results should be all green (no real changes are made on the management server)



Congrats! You made an Ansible Server that can send API calls to Check Point's Management Server.

Next steps

- My github (https://github.com/ryanrasmuss) has a few playbooks dedicated to Ansible-CheckPoint PoCs
- I recommend going over how to make Ansible Playbooks: https://docs.ansible.com/ansible/2.5/user_guide/playbooks.html
- Understand how the inventory works with Ansible: https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html
- Check Point's API: https://sc1.checkpoint.com/documents/latest/APIs/index.html#introduction~v1.1%20

Disclaimer

You can automate this entire setup process with the following script: https://github.com/ryanrasmuss/ansible-chkp-setup. Follow the directions listed in the readme.

