Lab 2: Scale Cloud Services using Orchestration

Adam Guled

University of Arizona

NETV 379: Cloud Computing

Henry Werchan

06/23/2024

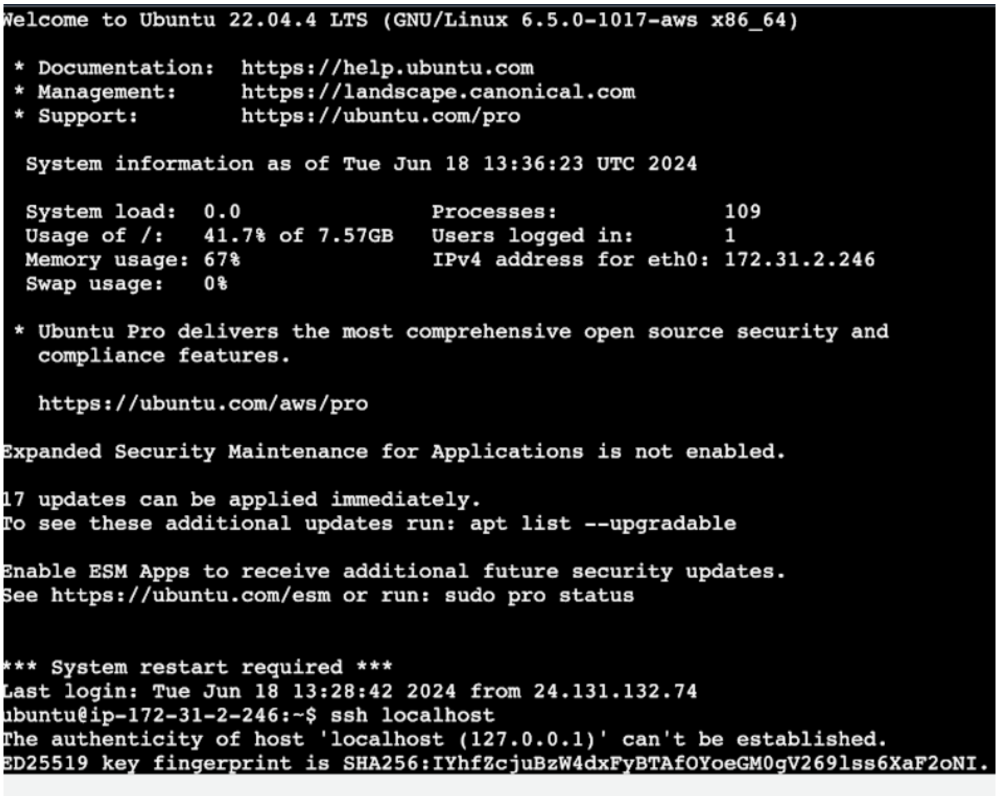
Lab 2: Scale Cloud Services using Orchestration

Using automation and orchestration tools to grow web services in a cloud environment was the goal of this lab. It focused on creating an AWS LAMP instance for Ansible-based configuration management, learning how to automate setups with Ansible, and running ad hoc commands to manage cloud infrastructure. Ansible is an open-source automation tool that makes it easier to provide software, manage configurations, and deploy applications.

Discussions

Step 1: Connecting to the EC2 Instance

- **Description:** Connected to the EC2 instance using AWS Instance Connect.



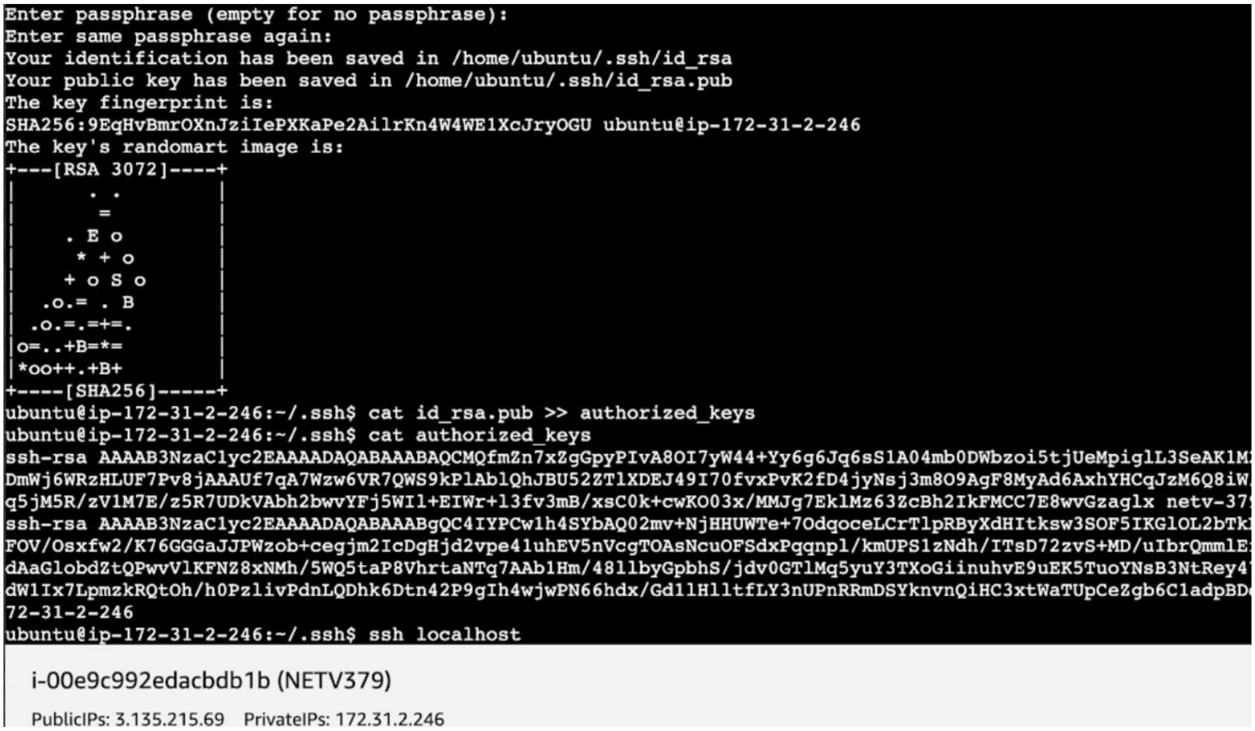
```
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Tue Jun 18 13:36:23 UTC 2024

  System load:  0.0               Processes:             109
  Usage of /:   41.7% of 7.57GB   Users logged in:       1
  Memory usage: 67%               IPv4 address for eth0: 172.31.2.246
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

17 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


*** System restart required ***
Last login: Tue Jun 18 13:28:42 2024 from 24.131.132.74
ubuntu@ip-172-31-2-246:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:IYhfZcjuBzW4dxFyBTAfOYoeGM0gV269lss6XaF2oNI.
```

Step 2: Generating RSA Key Pair

- **Commands:**
  cd .ssh

```
ssh-keygen
cat id_rsa.pub >> authorized_keys
cat authorized_keys
```

- **Description:** Generated an RSA key pair and appended the public key to authorized_keys for passwordless SSH login.



```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:9EqHvBmrOXnJziIePXKaPe2AilrKn4W4WE1XcJryOGU ubuntu@ip-172-31-2-246
The key's randomart image is:
+---[RSA 3072]----+
|       . .       |
|        =        |
|     . E o       |
|      * + o      |
|     + o S o     |
|   .o.= . B      |
|  .o.=.=+=.      |
|o=..+B=*=        |
|*oo++.+B+        |
+----[SHA256]-----+
ubuntu@ip-172-31-2-246:~/.ssh$ cat id_rsa.pub >> authorized_keys
ubuntu@ip-172-31-2-246:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCMQfmZn7xZgGpyPIvA8OI7yW44+Yy6g6Jq6sS1A04mb0DWbzoi5tjUeMpiglL3SeAK1M
DmWj6WRzHLUF7Pv8jAAAUf7qA7Wzw6VR7QWS9kPlAblQhJBU52ZTlXDEJ49I70fvxPvK2fD4jyNsj3m8O9AgF8MyAd6AxhYHCqJzM6Q8iW
q5jM5R/zV1M7E/z5R7UDkVAbh2bwvYFj5WIl+EIWr+l3fv3mB/xsC0k+cwKO03x/MMJg7EklMz63ZcBh2IkFMCC7E8wvGzaglx netv-37
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC4IYPCw1h4SYbAQ02mv+NjHHUWTe+7OdqoceLCrTlpRByXdHItksw3SOF5IKGlOL2bTk
FOV/Osxfw2/K76GGGaJJPWzob+cegjm2IcDgHjd2vpe41uhEV5nVcgTOAsNcuOFSdxPqqnpl/kmUPS1zNdh/ITsD72zvS+MD/uIbrQmmlE
dAaGlobdZtQPwvVlKFNZ8xNMh/5WQ5taP8VhrtaNTq7AAb1Hm/48llbyGpbhS/jdv0GTlMq5yuY3TXoGiinuhvE9uEK5TuoYNsB3NtRey4
dW1Ix7LpmzkRQtOh/h0PzlivPdnLQDhk6Dtn42P9gIh4wjwPN66hdx/GdllHlltfLY3nUPnRRmDSYknvnQiHC3xtWaTUpCeZgb6C1adpBD
72-31-2-246
ubuntu@ip-172-31-2-246:~/.ssh$ ssh localhost
```

i-00e9c992edacbdb1b (NETV379)

PublicIPs: 3.135.215.69 PrivateIPs: 172.31.2.246

Step 3: Installing and Configuring Ansible

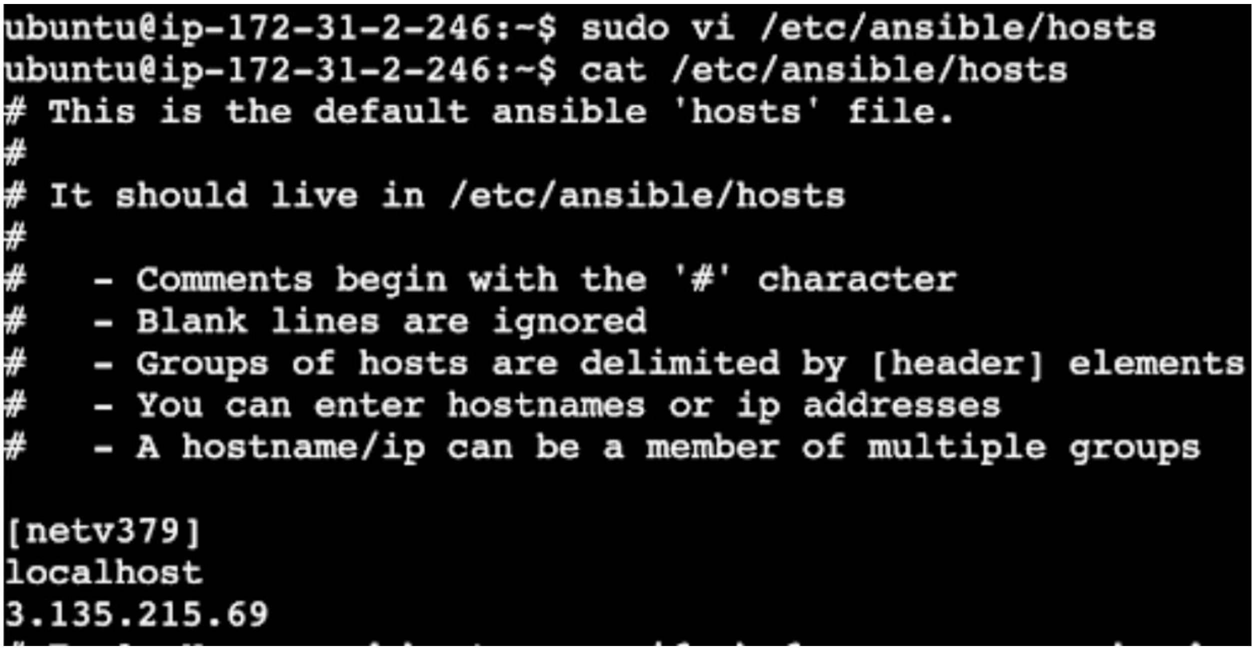
- **Commands:**

```
sudo apt-add-repository ppa:ansible/ansible
sudo apt-get update
sudo apt-get install ansible
sudo vi /etc/ansible/hosts
```

- **Inventory File Content:**

```
[netv379]
localhost
3.135.215.69
```

- **Description:** Added the Ansible PPA, installed Ansible, and created a custom inventory file.

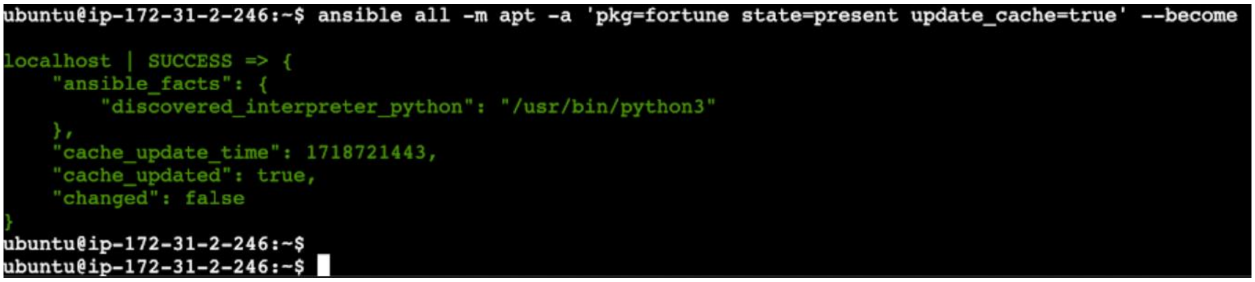


```
ubuntu@ip-172-31-2-246:~$ sudo vi /etc/ansible/hosts
ubuntu@ip-172-31-2-246:~$ cat /etc/ansible/hosts
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

[netv379]
localhost
3.135.215.69
```

Step 4: Running Ansible Commands

- **Installing a Package:**
  - **Command:**

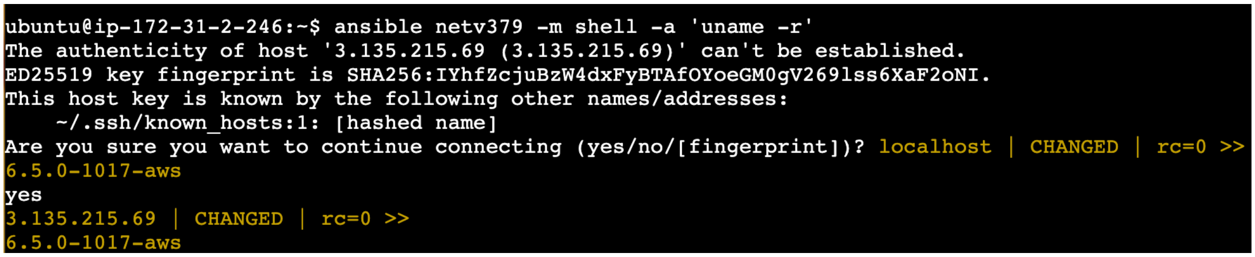
  ansible all -m apt -a 'pkg=cmatrix state=present update_cache=true' --become
  - **Description:** Used Ansible to install the cmatrix package.



```
ubuntu@ip-172-31-2-246:~$ ansible all -m apt -a 'pkg=fortune state=present update_cache=true' --become

localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1718721443,
    "cache_updated": true,
    "changed": false
}
ubuntu@ip-172-31-2-246:~$
ubuntu@ip-172-31-2-246:~$
```

- **Checking Kernel Version:**
  - **Command:**

  ansible netv379 -m shell -a "uname -r"
  - **Description:** Used Ansible to check the kernel version.



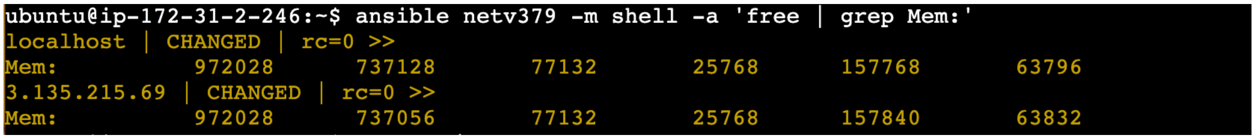
```
ubuntu@ip-172-31-2-246:~$ ansible netv379 -m shell -a 'uname -r'
The authenticity of host '3.135.215.69 (3.135.215.69)' can't be established.
ED25519 key fingerprint is SHA256:IYhfZcjuBzW4dxFyBTAfOYoeGM0gV269lss6XaF2oNI.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? localhost | CHANGED | rc=0 >>
6.5.0-1017-aws
yes
3.135.215.69 | CHANGED | rc=0 >>
6.5.0-1017-aws
```

- **Checking Memory Usage:**

- **Command:**

ansible netv379 -m shell -a "free | grep Mem:"

- **Description:** Used Ansible to check memory usage.

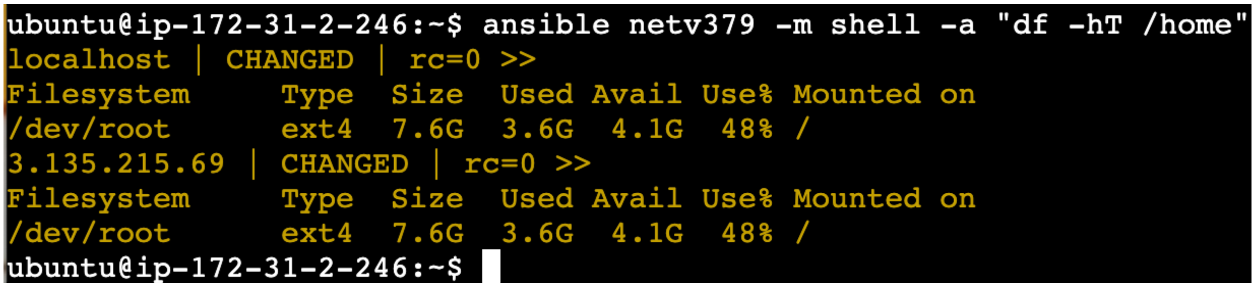


```
ubuntu@ip-172-31-2-246:~$ ansible netv379 -m shell -a 'free | grep Mem:'
localhost | CHANGED | rc=0 >>
Mem:           972028      737128      77132      25768      157768      63796
3.135.215.69 | CHANGED | rc=0 >>
Mem:           972028      737056      77132      25768      157840      63832
```

- **Checking Disk Space:**
  - **Command:**

  ansible netv379 -m shell -a "df -hT /home"

  - **Description:** Used Ansible to query the disk space usage of the /home directory.
  - **Output:**



```
ubuntu@ip-172-31-2-246:~$ ansible netv379 -m shell -a "df -hT /home"
localhost | CHANGED | rc=0 >>
Filesystem     Type  Size  Used Avail Use% Mounted on
/dev/root      ext4  7.6G  3.6G  4.1G  48% /
3.135.215.69 | CHANGED | rc=0 >>
Filesystem     Type  Size  Used Avail Use% Mounted on
/dev/root      ext4  7.6G  3.6G  4.1G  48% /
ubuntu@ip-172-31-2-246:~$
```

Observations & Conclusions

The Ansible commands successfully installed packages and queried system information, demonstrating the ability of Ansible to manage and monitor cloud infrastructure. The process of generating an RSA key pair, installing Ansible, and creating a custom inventory file was straightforward and facilitated the automation of configuration management tasks.

This lab exercise highlights the effectiveness of Ansible in automating configuration management and scaling cloud services. By using Ansible, tasks such as software installation, system updates, and querying system information can be performed consistently and efficiently across multiple instances.

References

Cánepa, G. (2015, October 18). How to automate simultaneous WordPress deployments in multiple Linux servers using Ansible – Part 3. TecMint. Retrieved from https://www.tecmint.com/automate-wordpress-installations-in-multiple-linux-servers-using-ansible/

Servers for Hackers. (2014, August 26). An Ansible tutorial. Servers for Hackers. Retrieved from https://serversforhackers.com/c/an-ansible-tutorial

Werchan, H. (2024). Start: Lab 2 Scale Cloud Services

Werchan, H. (2024). 5.Lab 2.Screenshots

Werchan, H. (2024). Cloud Orchestration & Automation