David Polis - Comcast Pre-Hire "DevOps Project"

Step-by-step supplement to the git repo at

1. Build Ansible server

AWS console:

Launch AWS instance (Using RHEL7)

Configure Default Security Group to allow all internal traffic/SSH only from my workstation IP (Ansible Server)

Generate/set key pair (id\_rsa)

Connect via SSH:

Configure Putty (IP address, convert key to PPK, xterm, login)

Add EPEL yum repository

>wget http://dl.fedoraproject.org/pub/epel/7/x86\_64/e/epel-release-7-8.noarch.rpm

>rpm -ivh epel-release-7-8.noarch.rpm

Install Git

>sudo yum install git

Install Ansible

>sudo yum install ansible -y

2. Provision webserver

Launch AWS instance on same subnet as ansible

Note: AWS RHEL7 instances are experiencing a bug that prevents default yum load balancers from communicating properly without an elastic IP.

Workaround: Switched to Ubuntu for this demo to remain in free tier.

Note: Ubuntu 16.04 does not have Python 2.7 installed OOB. Ansible does not yet support Python 3.

Workaround: Install Python 2.7 on target manually, via apt for the purposes of this demo. Could also leverage something like this if scalability were an issue: https://stackoverflow.com/questions/37802230/provisioning-ubuntu-16-04-vagrant-with-ansible-fails-on-chown

Create "webserver" security group for Apache server; allows HTTP and HTTPS In, All out

Add this instance to both security groups.

Now ansible server only accept SSH In from my IP, webserver accept SSH from my only my IP; HTTP and HTTPS from any IP. All outbound traffic allowed, all internal traffic allowed, both servers.

3. Configure ansible hosts

Add target host internal IP to /etc/ansible/hosts

[webserver]

172.31.55.187

Test connectivity to target:

[ec2-user@ip-172-31-55-187 ~]$ ansible webserver -m ping -u ubuntu

172.31.48.216 | SUCCESS => {

"changed": false,

"ping": "pong"

4. Stage project files on ansible server

Create folder structure/playbook file

/etc/ansible/ccdemo

/etc/ansible/ccdemo/roles

/etc/ansible/ccdemo.yaml

Clone apache role from git (found and vetted via ansible galaxy)

cd /etc/ansible/roles/

git clone https://github.com/geerlingguy/ansible-role-apache.git

Copy role to project

cp -r /etc/ansible/roles/ansible-role-apache /etc/ansible/ccdemo/roles/apache

5. Configure project files, modify role to suit (See davedev/ccdemo repo for full content)

ccdemo.yml

-defined host, login, privelege elevation and role

roles/apache/defaults/main.yml

- modified this to use a server\_fqdn variable in vhost entries, defined below

- added vhost parameters to force the HTTPS redirect, using the preferred vhost method, rather than a ReWrite rule. - configured true/false statements as required.

roles/apache/tasks/main.yml

- I added a task to the role here that generates the self-signed certificate that was not in the git master. I enabled a check so the play will fail if the certificate is not configured properly.

- I would normally add a task here to write out or copy a preconfigured index.html and related site structure. I might do this differently in a more complex scenario. E.g. break it out into its own role. I'm short on time so for this demo I just used vim to paste the html into once I verified apache was running and serving up the default index.

roles/apache/vars/main.yml

- I defined the server\_fqdn variable here and removed vars i was not using. The eliminated vars could be used to define multiple vhosts, to deploy in several runs to servers with different names, doc roots, ports, etc. Not necessary here.

6. Run playbook:

[ec2-user@ip-172-31-55-187 ccdemo]$ ansible-playbook site.yaml

PLAY [webserver] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [setup] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.48.216]

TASK [apache : Include OS-specific variables.] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.48.216]

TASK [apache : Define apache\_packages.] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.48.216]

TASK [apache : include] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

included: /etc/ansible/ccdemo/roles/apache/tasks/setup-Debian.yml for 172.31.48.216

TASK [apache : Update apt cache.] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.48.216]

TASK [apache : Ensure Apache is installed on Debian.] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.48.216] => (item=[u'apache2', u'apache2-utils'])

TASK [apache : Get installed version of Apache.] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.48.216]

TASK [apache : Create apache\_version variable.] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.48.216]

TASK [apache : include\_vars] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

skipping: [172.31.48.216]

TASK [apache : include\_vars] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.48.216]

TASK [apache : include] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

included: /etc/ansible/ccdemo/roles/apache/tasks/configure-Debian.yml for 172.31.48.216

TASK [apache : Configure Apache.] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.48.216] => (item={u'regexp': u'^Listen ', u'line': u'Listen 80'})

TASK [apache : Enable Apache mods.] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.48.216] => (item=rewrite.load)

ok: [172.31.48.216] => (item=ssl.load)

TASK [apache : Disable Apache mods.] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [apache : Check whether certificates defined in vhosts exist.] \*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.48.216] => (item={u'servername': u'ccdemo.polis.tech', u'certificate\_file': u'/etc/ssl/certs/server.crt', u'certificate\_key\_file': u'/etc/ssl/certs/server.key', u'documentroot': u'/var/www/html'})

TASK [apache : Add apache vhosts configuration.] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [172.31.48.216]

TASK [apache : Add vhost symlink in sites-enabled.] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.48.216]

TASK [apache : Remove default vhost in sites-enabled.] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.31.48.216]

TASK [apache : Generate a self-signed SSL certificate] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [172.31.48.216]

TASK [apache : Ensure Apache has selected state and enabled on boot.] \*\*\*\*\*\*\*\*\*\*

changed: [172.31.48.216]

RUNNING HANDLER [apache : restart apache] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [172.31.48.216]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

172.31.48.216 : ok=19 changed=19 unreachable=0 failed=0

[ec2-user@ip-172-31-55-187 ccdemo]$

7. Success!

-Note the self-checks occuring... In a more complex scenario I would add more checks.

8. DNS stuff

-added A record "ccdemo" to my polis.tech domain and pointed it at the AWS-provided external IP.

Thanks for your time. I hope this demontrates what you need to see... worth noting I hadn't used Ansible at all until last week. I like it... a lot. Much potential here... super scalable, IMO. Looking forward to your feedback!