Running your own Ansible Driven CA

Oct 30, 2016

Overview & Purpose

As a preparation for running a swarm cluster in production, I needed a way to manage my Root CA and distribute the certificates between my SWARM nodes, configuring their services to use them etc etc

A root CA

There is a bunch of posts / articles out there managing your own CA, none of them offer a free, automated solution which scales.

If running in a public DNS there ss a nice free online solution which can be configured programtically (and via ansible module (https://docs.ansible.com/ansible/letsencrypt_module.html)) called https://letsencrypt.org/ (https://letsencrypt.org/) there are also provides which give a free official SSL certificate which expire every 3 monthes which could be also a suitable solution …

In my case I needed a CA I can create | destroy | redistribute etc so I had in a way to create my own kind of solution …

CA Objectives

1. Install OpenSSL on your CA server host
    1. Configure the CA server options
    2. Generate CA private key
    3. Generate CA certificate generated with that key
2. Generate the required certificate requests for each of your nodes { including the CA server itself }
3. Distribute both the CA cert and the Host certificates to clients

4. Configure my services to use these certs & keys

Materials Needed

1. An inventory of hosts you wish to generate certificates for …
2. Ansible CA role (https://github.com/shelleg/ansible-role-ca/)

How does this work ?

In “shelleg context” the hosts / inventory could be either generated on the fly via a Dynamic Inventor*y (http://docs.ansible.com/ansible/intro_dynamic_inventory.html) or via general group_vars/all/xx_hosts file (more on this in another post …)

- Ansible managed hosts:

Let’s take a look at a part of our group vars which hold our inventory, this example has 1 CA server and 2 nodes like so:

```
shelleg_hosts:
  infra:
   - { cname: "infra01",
       ssl_key: infra01-priv-key.pem,
       ssl_cert: infra01-cert.pem,
     }
   swarm_managers:
     - { cname: "swarm-mgr01",
         ssl_key: swarm-mgr01-priv-key.pem,
         ssl_cert: swarm-mgr01-cert.pem,
       ...
       }
     - { cname: "swarm-node01",
         ssl_key: swarm-node01-priv-key.pem,
         ssl_cert: swarm-node01-cert.pem,
       }
     - { cname: "swarm-node02",
         ssl_key: swarm-node02-priv-key.pem,
         ssl_cert: swarm-node02-cert.pem,
         ...
       }
```

hosts.py (https://gist.github.com/hagzag/5727cd33f710bfbca2c3c6e5d556c8ea#file-hosts-py) hosted with ♡ by **GitHub (https://github.com)** view raw (https://gist.github.com/hagzag/5727cd33f710bfbca2c3c6e5d556c8ea/raw/c47a46de245395a75191d3e1e074fa2c74e2e065/hosts.py)

- Ansible CA role -> https://github.com/shelleg/ansible-role-ca/ (https://github.com/shelleg/ansible-role-ca/) whic has the following steps:

```
--- # tasks file for ansible-role-ca
include: ca-init.yml
when: ca_init is defined and ca_force_create == true

include: certify_nodes.yml
when: ca_certify_nodes is defined and ca_force_certify_nodes

include: fetch_keys.yml
when: ca_fetch_keys is defined

include: distribute_keys.yml
when: ca_distribute_keys is defined
```

main.yml (https://gist.github.com/hagzag/2d38958efd4669b61f97624d54fa0078#file-main-yml) hosted with ♡ by GitHub (https://github.com) view raw (https://gist.github.com/hagzag/2d38958efd4669b61f97624d54fa0078/raw/4b0913d6530a46b2964cdfe1cf12ddbf5fcc981c/main.yml)

- Setting up the CA server:

```
- name: "Ensure openssl is installed"
  apt: name=openssl state=latest

- name: "Delete ca-certs directory"
  file:
    path: "{{ item }}"
    state: absent
    owner: root
    group: root
  with_items:
  - "{{ ca_certs_dir }}"
- name: "Make configuration directory"
  file:
    path: "{{ item }}"
    state: directory
    owner: root
    group: root
  with_items:
  - "{{ ca_certs_dir }}"

- name: "Deploy configuration items"
  template:
    src: "{{ item }}.j2"
    dest: "{{ ca_certs_dir }}/{{ item }}"
    owner: root
    group: root
```

```
    with_items:
    - serial
    - ca.conf

  - name: "set CA_SUBJECT var"
    set_fact:
      ca_subject: '/C={{ ca_country }}/ST={{ ca_state }}/L={{ ca_locality }}/O={{ ca_organization }}/OU={{ ca_organizationalunit }}/CN={{ ca_commonname }}/emailAddress={{ ca_email }}'
     when: ca_subject is not defined

  - name: "Generate private key && Create root CA files"
    shell: "{{ item }}"
    args:
      chdir: "{{ ca_certs_dir }}"
    with_items:
    - "openssl genrsa -out {{ ca_key }} 2048"
    - "openssl req -config /usr/lib/ssl/openssl.cnf -new -key {{ ca_key }} -x509 -days 1825 -out {{ ca_cert }} -passin pass:{{ ca_rootca_password }} -subj \"{{ ca_subject }}\""
```

ca-init.yml (https://gist.github.com/hagzag/89e32ff019abb4713d8d8a5da5152ea0#file-ca-init-yml) hosted with ♡ by GitHub (https://github.com) view raw (https://gist.github.com/hagzag/89e32ff019abb4713d8d8a5da5152ea0/raw/46f33a404caef80176451d70095ceb030551722d/ca-init.yml)

- Generating the node certificates:

```
  - name: "Generate Certs for Infra server || CA server"
    shell: 'openssl genrsa -out {{ item.cname }}-priv-key.pem 2048'
    args:
      chdir: "{{ ca_certs_dir }}"
    with_items:
    - "{{ shelleg_hosts.infra }}"
    - "{{ shelleg_hosts.swarm.swarm_workers }}"
    - "{{ shelleg_hosts.swarm.swarm_managers }}"

  - name: "Create certificate request for Infra server || CA server"
    shell: 'openssl req -subj "/CN={{ item.cname }}" -new -key "{{ item.cname }}"-priv-key.pem -out "{{ item.cname }}".csr'
    args:
     chdir: "{{ ca_certs_dir }}"
    with_items:
    - "{{ shelleg_hosts.infra }}"
    - "{{ shelleg_hosts.swarm.swarm_workers }}"
    - "{{ shelleg_hosts.swarm.swarm_managers }}"

  - name: "Generate the CA trusted certificate"
    shell: 'sudo openssl x509 -req -days 1825 -in "{{ item.cname }}".csr -CA ca.pem -CAkey ca-priv-key.pem -CAcreateserial -out "{{ item.cname }}"-cert.pem -extensions v3_req -extfile /usr/lib/ssl/openssl.cnf'
    args:
     chdir: "{{ ca_certs_dir }}"
    with_items:
    - "{{ shelleg_hosts.infra }}"
    - "{{ shelleg_hosts.swarm.swarm_workers }}"
    - "{{ shelleg_hosts.swarm.swarm_managers }}"
```

certs.yml (https://gist.github.com/hagzag/26f583b5fca83605417ddae2e883898b#file-certs-yml) hosted with ♡ by GitHub (https://github.com) view raw (https://gist.github.com/hagzag/26f583b5fca83605417ddae2e883898b/raw/9eed300090923e08c719ea66b583a307d18b4350/certs.yml)

- Fetching the keys for distribution (copy from CA server to Ansible control machine):

```
- name: "copy keys from infra to ansible machine for distribution"
  fetch: src="{{ ca_certs_dir }}/{{ item.ssl_key }}" dest="{{ ca_distribution_certs_dir }}/{{ item.ssl_key }}" flat=yes
  with_items:
  - "{{ shelleg_hosts.infra }}"
  - "{{ shelleg_hosts.swarm.swarm_workers }}"
  - "{{ shelleg_hosts.swarm.swarm_managers }}"

- name: "copy certs from infra to ansible machine for distribution"
  fetch: src="{{ ca_certs_dir }}/{{ item.ssl_cert }}" dest="{{ ca_distribution_certs_dir }}/{{ item.ssl_cert }}" flat=yes
  with_items:
  - "{{ shelleg_hosts.infra }}"
  - "{{ shelleg_hosts.swarm.swarm_workers }}"
  - "{{ shelleg_hosts.swarm.swarm_managers }}"

- name: "copy ca.pem ca-priv-key.pem"
  fetch: src="{{ ca_certs_dir }}/{{ item }}" dest="{{ ca_distribution_certs_dir }}/{{ item }}" flat=yes
  with_items:
  - "{{ ca_cert }}"
  - "{{ ca_key }}"
```

fetch-certs.yml (https://gist.github.com/hagzag/a7a4a1fefd45541f3c393a20ae5abc54#file-fetch-certs-yml) hosted with ♡ by GitHub (https://github.com) view raw (https://gist.github.com/hagzag/a7a4a1fefd45541f3c393a20ae5abc54/raw/3e1bbcc5e1f029e21e1d60ca2fe3b06658a62654/fetch-certs.yml)

- Distribute the Certs & keys to the various nodes:

```
- name: "Ensures {{ ca_defult_ssl_certs_dir }} and {{ ca_defult_ssl_key_dir }} dirs exist"
  file: path="{{ item }}" state=directory owner=root group=root mode=0750 recurse=yes
  with_items:
  - "{{ ca_defult_ssl_certs_dir }}"
  - "{{ ca_defult_ssl_key_dir }}"

- block:

  - name: "copy keys from infra to ansible machine for distribution"
    copy: src="{{ ca_distribution_certs_dir }}/{{ item.ssl_key }}" dest="{{ ca_defult_ssl_key_dir }}/{{ item.ssl_key }}"
    with_items:
    - "{{ shelleg_hosts.infra }}"
    - "{{ shelleg_hosts.swarm.swarm_workers }}"
    - "{{ shelleg_hosts.swarm.swarm_managers }}"

  - name: "copy certs from infra to ansible machine for distribution"
    copy: src="{{ ca_distribution_certs_dir }}/{{ item.ssl_cert }}" dest="{{ ca_defult_ssl_certs_dir }}/{{ item.ssl_cert }}"
```

```
      with_items:
      - "{{ shelleg_hosts.infra }}"
      - "{{ shelleg_hosts.swarm.swarm_workers }}"
      - "{{ shelleg_hosts.swarm.swarm_managers }}"

   when: inventory_hostname == "{{ item.cname }}"

  # Root CA key/cert

  - name: "copy {{ ca_key }} to {{ ca_defult_ssl_key_dir }}"
    copy:
     src: "{{ ca_distribution_certs_dir }}/{{ item }}"
     dest: "{{ ca_defult_ssl_key_dir }}/{{ item }}"
    with_items:
    - "{{ ca_key }}"

  - name: "copy {{ ca_cert }} to {{ ca_defult_ssl_certs_dir }}"
    copy:
     src: "{{ ca_distribution_certs_dir }}/{{ item }}"
     dest: "{{ ca_defult_ssl_certs_dir }}/{{ item }}"
    with_items:
    - "{{ ca_cert }}"
```

gen-certe.yml (https://gist.github.com/hagzag/7beb8acef1fcbdbe90a76f058b4c647c#file-gen-certe-yml) hosted with ♡ by GitHub (https://github.com) view raw (https://gist.github.com/hagzag/7beb8acef1fcbdbe90a76f058b4c647c/raw/53c320a27a6061d3fcfd2c85ec1382cca4594f2d/gen-certe.yml)

Gotchas

This role is still under development …

Currently running the following playbook will result in all the 6 steps unless you set the available vars to prevent them as seen in the main.yml above.

The supporting vars are:

```
# weather or not you wish to:
# install and configure the root CA (from scratch)
ca_init: true
# generate certs for nodes
ca_certify_nodes: true
# copy key to ansible control machine
ca_fetch_keys: true
# force creating even if files exist on the node
ca_force_create: true
# force creating of node certificates
ca_force_certify_nodes: true
```

```
# distribute / copy keys from control machine to nodes
ca_distribute_keys: true
```

vars.yml (https://gist.github.com/hagzag/d730405560c3d68a11810e78bcb5f684#file-vars-yml) hosted with ♡ by GitHub (https://github.com) view raw (https://gist.github.com/hagzag/d730405560c3d68a11810e78bcb5f684/raw/26c36e04bacd79a1bea2ae1e5dab8d0e65291ac0/vars.yml)

An example playbook utilizing the CA role - in the CA server:

```
- hosts: infra01
  become: true
  vars:
   ca_init: true
   ca_certify_nodes: true
   ca_fetch_keys: true
   ca_force_create: true
   ca_force_certify_nodes: true
   ca_distribute_keys: true
  roles:
   - role: ansible-role-ca
     tags: ca
```

playbook-infra.yml (https://gist.github.com/hagzag/cfbc99b4e63a3beb90a50056ce3e2d48#file-playbook-infra-yml) hosted with ♡ by GitHub (https://github.com) view raw (https://gist.github.com/hagzag/cfbc99b4e63a3beb90a50056ce3e2d48/raw/cb81c07532c17408158b0c14188e28f6b7509d27/playbook-infra.yml)

On the nodes which needs certificates …

```
- hosts: shelleg-swarm-instances
  vars:
   ca_distribute_keys: true
  become: yes
  become_user: root
  roles:
    - role: ansible-role-ca
      tags: ca,core
```

deploy-nodes.yml (https://gist.github.com/hagzag/9f55d18246c650213e8d9d6d017e2e7d#file-deploy-nodes-yml) hosted with ♡ by GitHub (https://github.com) view raw (https://gist.github.com/hagzag/9f55d18246c650213e8d9d6d017e2e7d/raw/a71fb5f98ec9f7962564312a859587f47ee070d1/deploy-nodes.yml)

Go ahead and give a try and tell me what you think (open an issue if needed ;))

Going forward

Issue #1: Control the creating of the server kay only when the existing CA kay has expired, unless force create is defined … there is a mechanism in place which needs testing … *Issue #2:* Add support for more hosts / groups of nodes - currently supports only the shelleg.infra and shelleg.swarm.* node groups.

Hope you enjoyed this post at least as much as I enjoyed writing this role …

Comments and findings are welcome.