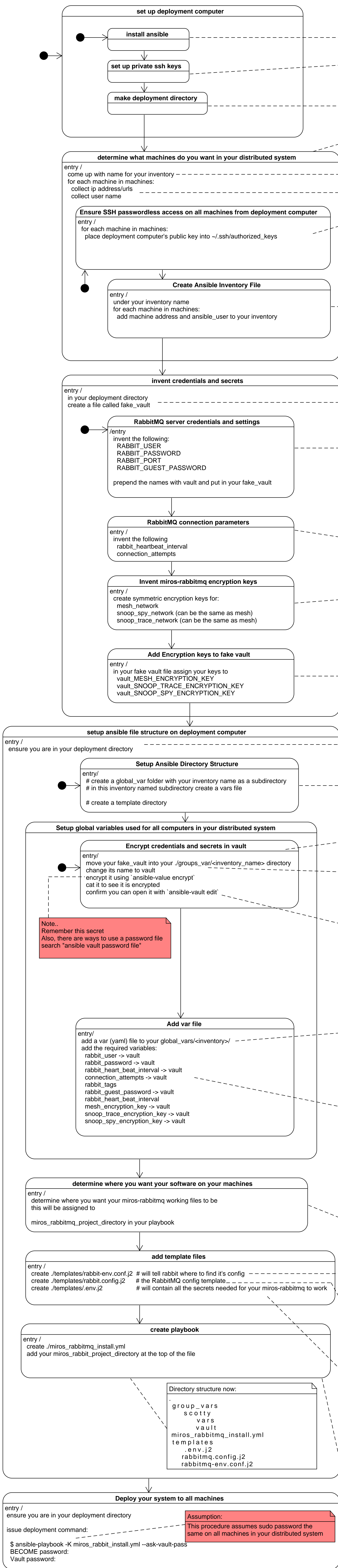


WORKFLOW



EXAMPLES

example of installing ansible:

```
sudo apt-get ansible
```

example of setting local ssh keys:

```
$ sudo ssh-keygen  
$ sudo cat ~/.ssh/id_rsa.pub » ~/.ssh/authorized_keys'
```

example of setting deployment directory:

```
$ mkdir ~/miros_rabbitmq_deployment
```

Assumption:
This procedure assumes sudo password the same on all machines in your distributed system

example inventory name:

scotty

example:

machine	user
192.168.1.71	pi
192.168.1.69	pi

example:

```
cat ~/.ssh/id_rsa.pub | ssh pi@192.168.0.69 'cat » .ssh/authorized_keys'
```

example inventory, on deployment machine in it's /etc/ansible/hosts:

```
[scotty]  
192.168.1.71 ansible_user=pi  
192.168.1.69 ansible_user=pi
```

example:

```
$ cd ~/miros_rabbitmq_deployment  
$ touch fake_vault # this is a yaml file
```

example fake_vault file:

```
---  
vault_RABBIT_USER: peter  
vault_RABBIT_PASSWORD: rabbit  
vault_RABBIT_PORT: 5672  
vault_RABBIT_GUEST_PASSWORD: rabbit567
```

example fake_vault (yaml) file now:

```
---  
vault_RABBIT_HEARTBEAT_INTERVAL: 3600  
vault_CONNECTION_ATTEMPTS: 3  
vault_RABBIT_USER: peter  
vault_RABBIT_PASSWORD: rabbit  
vault_RABBIT_PORT: 5672  
vault_RABBIT_GUEST_PASSWORD: rabbit567
```

example of how to make keys:

```
$ python3  
from cryptography import Fernet  
  
# do this once per needed encryption keys  
encryption_key = Fernet.generate_key()  
print(encryption_key) # => b'u3u...' (copy the string part)
```

Note..
make sure you put " around your keys

example fake_vault (yaml) file now:

```
---  
vault_MESH_ENCRYPTION_KEY: 'u3Uc-qAi9iiCv3fkBfRUAKrM1gH8w51-nVU8M8A73Jg='  
vault_SNOOP_TRACE_ENCRYPTION_KEY: 'u3Uc-qAi9iiCv3fkBfRUAKrM1gH8w51-nVU8M8A73Jg='  
vault_SNOOP_SPY_ENCRYPTION_KEY: 'u3Uc-qAi9iiCv3fkBfRUAKrM1gH8w51-nVU8M8A73Jg='  
vault_RABBIT_HEARTBEAT_INTERVAL: 3600  
vault_CONNECTION_ATTEMPTS: 3  
vault_RABBIT_USER: peter  
vault_RABBIT_PASSWORD: rabbit  
vault_RABBIT_PORT: 5672  
vault_RABBIT_GUEST_PASSWORD: rabbit567
```

example:

```
$ cd ~/miros_rabbitmq_deployment
```

example

```
mkdir group_vars  
mkdir ./group_vars/scotty  
touch ./group_vars/scotty/var  
mkdir templates  
---  
group_vars  
scotty  
vars  
templates
```

Directory structure now

```
group_vars  
scotty  
vars  
vault  
templates
```

example:

```
$ mv fake_vault ./global_vars/scotty/vault  
$ ansible-vault encrypt ./global_vars/scotty/vault
```

example:

```
$ cat ./global_vars/scotty/vault  
  
$ANSIBLE_VAULT;1.1;AES256  
34363736353133336561626464646437613...
```

Note..
If you see something like this you can add your vault to your revision control system

example:

```
$ ansible-vault edit ./global_vars/scotty/vault  
  
---  
vault_MESH_ENCRYPTION_KEY: 'u3Uc-qAi9iiCv3fkBfRUAKrM1gH8w51-nVU8M8A73Jg='  
vault_SNOOP_TRACE_ENCRYPTION_KEY: 'u3Uc-qAi9iiCv3fkBfRUAKrM1gH8w51-nVU8M8A73Jg='  
vault_SNOOP_SPY_ENCRYPTION_KEY: 'u3Uc-qAi9iiCv3fkBfRUAKrM1gH8w51-nVU8M8A73Jg='  
vault_RABBIT_HEARTBEAT_INTERVAL: 3600  
vault_CONNECTION_ATTEMPTS: 3  
vault_RABBIT_USER: peter  
vault_RABBIT_PASSWORD: rabbit  
vault_RABBIT_PORT: 5672  
vault_RABBIT_GUEST_PASSWORD: rabbit567
```

example:

```
$ touch ./global_vars/scotty/var
```

example vars file:

```
---  
# public  
python_packages_to_install:  
- miros-rabbitmq  
rabbit_tags:  
- administrator  
  
# secrets  
rabbit_user: "{{ vault_RABBIT_USER }}"  
rabbit_password: "{{ vault_RABBIT_PASSWORD }}"  
rabbit_port: "{{ vault_RABBIT_PORT }}"  
rabbit_heart_beat_interval: "{{ vault_RABBIT_HEARTBEAT_INTERVAL }}"  
connection_attempts: "{{ vault_CONNECTION_ATTEMPTS }}"  
rabbit_guest_password: "{{ vault_RABBIT_GUEST_PASSWORD }}"  
mesh_encryption_key: "{{ vault_MESH_ENCRYPTION_KEY }}"  
snoop_trace_encryption_key: "{{ vault_SNOOP_TRACE_ENCRYPTION_KEY }}"  
snoop_spy_encryption_key: "{{ vault_SNOOP_SPY_ENCRYPTION_KEY }}"  
rabbit_heart_beat_interval: "{{ vault_RABBIT_HEARTBEAT_INTERVAL }}"
```

Note..
this creates public variable names with secret contents

example:

```
miros_rabbitmq_project_directory: '~/miros-rabbitmq'
```

example ./templates/rabbit-env.conf.j2 file:

```
RABBITMQ_CONFIG_FILE=/etc/rabbitmq/rabbitmq  
NODE_IP_ADDRESS=0.0.0.0
```

example ./templates/rabbit-config.j2 file:

```
{  
  (rabbit,  
    {  
      {loopback_users,[]}  
    }  
  )  
}
```

Directory structure now

```
group_vars  
scotty  
vars  
vault  
templates  
.env.j2  
rabbitmq.config.j2  
rabbitmq-env.conf.j2
```

example ./templates/.env.j2 file:

```
MESH_ENCRYPTION_KEY={{mesh_encryption_key}}  
SNOOP_TRACE_ENCRYPTION_KEY={{snoop_trace_encryption_key}}  
SNOOP_SPY_ENCRYPTION_KEY={{snoop_spy_encryption_key}}  
RABBIT_USER={{rabbit_user}}  
RABBIT_PASSWORD={{rabbit_password}}  
RABBIT_PORT={{rabbit_port}}  
RABBIT_HEARTBEAT_INTERVAL={{rabbit_heart_beat_interval}}  
CONNECTION_ATTEMPTS={{connection_attempts}}  
RABBIT_GUEST_USER={{rabbit_guest_user}}
```

example ./miros_rabbitmq_install.yml

```
---  
- hosts: scotty  
  vars:  
    miros_rabbitmq_project_directory: '~/miros-rabbitmq'  
  tasks:  
    - name: Install rabbitmq-server  
      become: true  
      apt: name={{ item }} state=present update_cache=false  
        with_items:  
          - erlang  
          - rabbitmq-server  
    .  
    .  
    .
```