Chapter 7: Using Ansible with Cloud Services and Containers

1. Amazon Web Services (AWS) with Ansible

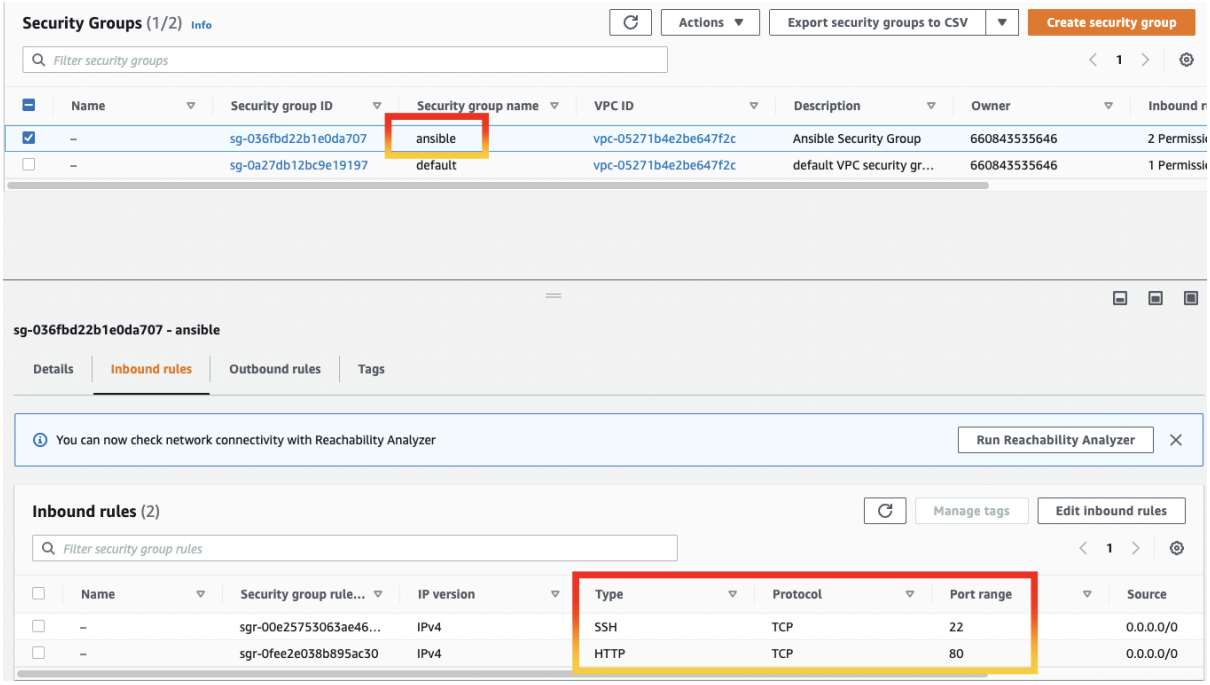
1-1. 환경 구성

```
$ export AWS_ACCESS_KEY_ID='xxxxxxxxxxx'
$ export AWS_SECRET_ACCESS_KEY='xxxxxxxxxxx'
$ sudo pip3 install boto boto3
```

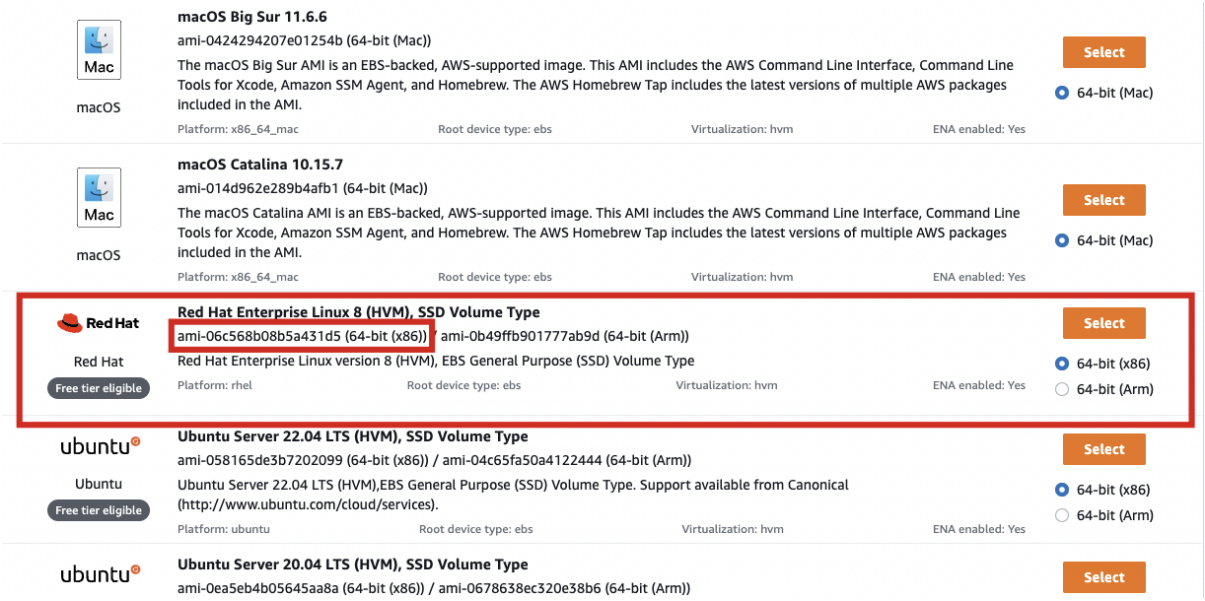
```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: localhost
  connection: local
  gather_facts: false

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Create a security group in AWS for SSH access and HTTP
```

```
      ec2_group:
         name: ansible
         description: Ansible Security Group
         region: ap-northeast-2
         rules:
            - proto: tcp
              from_port: 80
              to_port: 80
              cidr_ip: 0.0.0.0/0
            - proto: tcp
              from_port: 22
              to_port: 22
              cidr_ip: 0.0.0.0/0

# Three dots indicate the end of a YAML document
...
```

```
$ ansible-playbook ec2_playbook.yaml
```



1-2. ec2 생성



ami id : ami-06c568b08b5a431d5

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: localhost
  connection: local
  gather_facts: false

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Create a security group in AWS for SSH access and HTTP
      ec2_group:
         name: ansible
         description: Ansible Security Group
         region: ap-northeast-2
         rules:
            - proto: tcp
              from_port: 80
              to_port: 80
              cidr_ip: 0.0.0.0/0
            - proto: tcp
              from_port: 22
              to_port: 22
              cidr_ip: 0.0.0.0/0

    - name: Provision a set of instances
      ec2:
         key_name: ansible
         group: ansible
         instance_type: t2.micro
         image: ami-06c568b08b5a431d5
         region: ap-northeast-2
         wait: true
         exact_count: 20
         count_tag:
            Name: AnsibleNginxWebservers
         instance_tags:
            Name: Ansible
      register: ec2

    - name: Add all instance public IPs to host group
      add_host:
        hostname: "{{ item.public_ip }}"
        groups: ansiblehosts
      with_items: "{{ ec2.instances }}"

    - name: Show group
      debug:
        var: groups.ansiblehosts
```

```
# Three dots indicate the end of a YAML document
...
```

1-3. ignore_errors

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: localhost
  connection: local
  gather_facts: false

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Create a security group in AWS for SSH access and HTTP
      ec2_group:
         name: ansible
         description: Ansible Security Group
         region: ap-northeast-2
         rules:
            - proto: tcp
              from_port: 80
              to_port: 80
              cidr_ip: 0.0.0.0/0
            - proto: tcp
              from_port: 22
              to_port: 22
              cidr_ip: 0.0.0.0/0

    - name: Provision a set of instances
      ec2:
         key_name: ansible
         group: ansible
         instance_type: t2.micro
         image: ami-06c568b08b5a431d5
         region: ap-northeast-2
         wait: true
         exact_count: 20
         count_tag:
            Name: AnsibleNginxWebservers
         instance_tags:
            Name: Ansible
      register: ec2
      ignore_errors: true

    - name: Add all instance public IPs to host group
      add_host:
        hostname: "{{ item.public_ip }}"
        groups: ansiblehosts
      with_items: "{{ ec2.instances }}"

    - name: Show group
      debug:
        var: groups.ansiblehosts

# Three dots indicate the end of a YAML document
...
```

1-4. dynamic inventory

```
$ cd /home/ansible/diveintoansible/Using Ansible with Cloud Services and Containers/AWS with Ansible/04
$ mkdir inventory
$ cd inventory
$ wget https://raw.githubusercontent.com/ansible/ansible/stable-2.9/contrib/inventory/ec2.py
$ wget https://raw.githubusercontent.com/ansible/ansible/stable-2.9/contrib/inventory/ec2.ini
$ chmod u+x ec2.py
```

ec2.py 수정

```
#!/usr/bin/env python3


# from ansible.module_utils import ec2 as ec2_utils
```

ec2.ini 수정

```
# The number of seconds a cache file is considered valid. After this many
# seconds, a new API call will be made, and the cache file will be updated.
# To disable the cache, set this value to 0
cache_max_age = 0
```

EC2_INI_PATH 설정 및 EC2 목록 정보 얻어오기

```
$ cd ..
$ pwd
/home/ansible/diveintoansible/Using Ansible with Cloud Services and Containers/AWS with Ansible/04
$ export EC2_INI_PATH=inventory/ec2.ini

# 출력되는 정보중 tag_Name_Ansible 이 생성된 EC2 호스트 정보이다.
$ inventory/ec2.py --list
...
...
...
```

ansible.cfg

```
[defaults]
inventory = inventory/ec2.py
host_key_checking = False
forks=20
ansible_managed = Managed by Ansible - file:{file} - host:{host} - uid:{uid}
```

```
---
ansible_ssh_private_key_file: ~/.ssh/ansible.pem
ansible_user: ec2-user
ansible_become: true
...
```

```
# console 에서 생성한 key pair 파일 저장 후 퍼미션 설정
$ chmod 600 ~/.ssh/ansible.pem
$ ansible tag_Name_Ansible -m ping -o
```

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: localhost
  connection: local
  gather_facts: false

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Create a security group in AWS for SSH access and HTTP
      ec2_group:
         name: ansible
         description: Ansible Security Group
         region: ap-northeast-2
         rules:
            - proto: tcp
              from_port: 80
              to_port: 80
              cidr_ip: 0.0.0.0/0
            - proto: tcp
              from_port: 22
```

```
              to_port: 22
              cidr_ip: 0.0.0.0/0

    - name: Provision a set of instances
      ec2:
         key_name: ansible
         group: ansible
         instance_type: t2.micro
         image: ami-096fda3c22c1c990a
         region: ap-northeast-2
         wait: true
         exact_count: 20
         count_tag:
            Name: AnsibleNginxWebservers
         instance_tags:
            Name: Ansible
      register: ec2
      ignore_errors: true

    - name: Refresh inventory to ensure new instances exist in inventory
      meta: refresh_inventory

-

  # Target: where our play will run and options it will run with
  hosts: tag_Name_Ansible

  # Roles: list of roles to be imported into the play
  roles:
    - { role: webapp, target_dir: /usr/share/nginx/html }

# Three dots indicate the end of a YAML document
...
```

(*)webapp 롤은 이미 추가되어 있다.

1-5. pause prompt

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: localhost
  connection: local
  gather_facts: false

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Create a security group in AWS for SSH access and HTTP
      ec2_group:
         name: ansible
         description: Ansible Security Group
         region: ap-northeast-2
         rules:
            - proto: tcp
              from_port: 80
              to_port: 80
              cidr_ip: 0.0.0.0/0
            - proto: tcp
              from_port: 22
              to_port: 22
              cidr_ip: 0.0.0.0/0

    - name: Provision a set of instances
      ec2:
         key_name: ansible
         group: ansible
         instance_type: t2.micro
         image: ami-096fda3c22c1c990a
         region: us-east-1
         wait: true
         exact_count: 20
         count_tag:
            Name: AnsibleNginxWebservers
         instance_tags:
            Name: Ansible
      register: ec2
      ignore_errors: true
```

```
    - name: Refresh inventory to ensure new instances exist in inventory
      meta: refresh_inventory

-

  # Target: where our play will run and options it will run with
  hosts: tag_Name_Ansible

  # Roles: list of roles to be imported into the play
  roles:
    - { role: webapp, target_dir: /usr/share/nginx/html }

-

  # Target: where our play will run and options it will run with
  hosts: tag_Name_Ansible

  # Task: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - debug:
        msg: "Check http://{{ ansible_host }}"

    - pause:
        prompt: "Verify service availability and continue to terminate"

# Three dots indicate the end of a YAML document
...
```

1-6. remove ec2 instances and security groups

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: localhost
  connection: local
  gather_facts: false

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Create a security group in AWS for SSH access and HTTP
      ec2_group:
         name: ansible
         description: Ansible Security Group
         region: us-east-1
         rules:
            - proto: tcp
              from_port: 80
              to_port: 80
              cidr_ip: 0.0.0.0/0
            - proto: tcp
              from_port: 22
              to_port: 22
              cidr_ip: 0.0.0.0/0

    - name: Provision a set of instances
      ec2:
         key_name: ansible
         group: ansible
         instance_type: t2.micro
         image: ami-096fda3c22c1c990a
         region: ap-northeast-2
         wait: true
         exact_count: 20
         count_tag:
            Name: AnsibleNginxWebservers
         instance_tags:
            Name: Ansible
      register: ec2
      ignore_errors: true

    - name: Refresh inventory to ensure new instances exist in inventory
      meta: refresh_inventory

-
```

```
  # Target: where our play will run and options it will run with
  hosts: tag_Name_Ansible

  # Roles: list of roles to be imported into the play
  roles:
    - { role: webapp, target_dir: /usr/share/nginx/html }

-

  # Target: where our play will run and options it will run with
  hosts: tag_Name_Ansible

  # Task: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - debug:
        msg: "Check http://{{ ansible_host }}"

    - pause:
        prompt: "Verify service availability and continue to terminate"

    - name: Remove tagged EC2 instances from security group by setting an empty group
      ec2:
        state: running
        region: "{{ ec2_region }}"
        instance_ids: "{{ ec2_id }}"
        group_id: ""
      delegate_to: localhost

    - name: Terminate EC2 instances
      ec2:
        state: absent
        region: "{{ ec2_region }}"
        instance_ids: "{{ ec2_id }}"
        wait: true
      delegate_to: localhost

-

  hosts: localhost
  connection: local
  gather_facts: false

  tasks:
  - name: Remove ansible security group
    ec2_group:
      name: ansible
      region: ap-northeast-2
      state: absent


# Three dots indicate the end of a YAML document
...
```

2. Docker with Ansible

2-1. Configure our Docker Lab

```
$ cd /home/ansible/diveintoansible/Using Ansible with Cloud Services and Containers/Docker with Ansible/01

$ ping docker
PING docker (172.18.0.6) 56(84) bytes of data.
64 bytes from docker.diveinto.io (172.18.0.6): icmp_seq=1 ttl=64 time=0.263 ms
64 bytes from docker.diveinto.io (172.18.0.6): icmp_seq=2 ttl=64 time=0.518 ms
64 bytes from docker.diveinto.io (172.18.0.6): icmp_seq=3 ttl=64 time=0.587 ms
^C
--- docker ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max/mdev = 0.263/0.456/0.587/0.139 ms

$ cat install_docker.sh
sudo apt update
sudo apt install -y docker.io
pip3 install docker

$ bash -x install_docker.sh

$ cat envdocker
export DOCKER_HOST=tcp://docker:2375

$ source envdocker
$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
```

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Pull images
      docker_image:
        docker_host: tcp://docker:2375
        name: "{{ item }}"
        source: pull
      with_items:
        - centos
        - ubuntu
        - redis
        - nginx
        # n.b. large image, >1GB
        - wernight/funbox

# Three dots indicate the end of a YAML document
...
```

```
$ ansible-playbook docker_playbook.yaml

PLAY [ubuntu-c] ****************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [ubuntu-c]

TASK [Pull images] *************************************************************************************************************************************
changed: [ubuntu-c] => (item=centos)
changed: [ubuntu-c] => (item=ubuntu)
changed: [ubuntu-c] => (item=redis)
changed: [ubuntu-c] => (item=nginx)
changed: [ubuntu-c] => (item=wernight/funbox)

PLAY RECAP *********************************************************************************************************************************************
ubuntu-c                   : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

$ docker images
REPOSITORY        TAG       IMAGE ID       CREATED        SIZE
redis             latest    24f9f41dd114   10 days ago    111MB
nginx             latest    cd4e03b35a8e   10 days ago    134MB
```

```
ubuntu            latest    a7870fd478f4   3 weeks ago    69.2MB
centos            latest    e6a0117ec169   9 months ago   272MB
wernight/funbox   latest    538c146646c3   4 years ago    1.12GB

# 도커 실행 테스트
$ cat examples.txt
docker run --rm -it wernight/funbox cmatrix
docker run --rm -it wernight/funbox nyancat
docker run --rm -it wernight/funbox asciiquarium
```

2-2. Build Containers

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Pull images
      docker_image:
        docker_host: tcp://docker:2375
        name: "{{ item }}"
        source: pull
      with_items:
        - centos
        - ubuntu
        - redis
        - nginx
        # n.b. large image, >1GB
        - wernight/funbox

    - name: Create an nginx container
      docker_container:
        docker_host: tcp://docker:2375
        name: containerwebserver
        image: nginx
        ports:
          - 80:80
        container_default_behavior: no_defaults

# Three dots indicate the end of a YAML document
...
```

```
$ ansible-playbook docker_playbook.yaml
$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                NAMES
b6e91c7818dc   nginx     "/docker-entrypoint.…"   3 seconds ago   Up 2 seconds   0.0.0.0:80->80/tcp   containerwebserver
```

2-3. Build Customized Images and Customized Containers

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Pull images
      docker_image:
        docker_host: tcp://docker:2375
        name: "{{ item }}"
        source: pull
      with_items:
        - centos
```

```
        - ubuntu
        - redis
        - nginx
        # n.b. large image, >1GB
        - wernight/funbox

    - name: Create a customised Dockerfile
      copy:
        dest: /shared/Dockerfile
        mode: 0644
        content: |
          FROM nginx

    - name: Build a customised image
      docker_image:
        docker_host: tcp://docker:2375
        name: nginxcustomised:latest
        source: build
        build:
          path: /shared
          pull: yes
        state: present
        force_source: yes

    - name: Create an nginxcustomised container
      docker_container:
        docker_host: tcp://docker:2375
        name: containerwebserver
        image: nginxcustomised:latest
        ports:
          - 80:80
        container_default_behavior: no_defaults
        recreate: yes

# Three dots indicate the end of a YAML document
...
```

```
$ ansible-playbook docker_playbook.yaml
$ docker images
REPOSITORY        TAG       IMAGE ID       CREATED        SIZE
redis             latest    24f9f41dd114   10 days ago    111MB
nginx             latest    cd4e03b35a8e   10 days ago    134MB
nginxcustomised   latest    cd4e03b35a8e   10 days ago    134MB
ubuntu            latest    a7870fd478f4   3 weeks ago    69.2MB
centos            latest    e6a0117ec169   9 months ago   272MB
wernight/funbox   latest    538c146646c3   4 years ago    1.12GB
$ docker ps -a
CONTAINER ID   IMAGE                    COMMAND                  CREATED         STATUS         PORTS                  NAMES
719ad4488936   nginxcustomised:latest   "/docker-entrypoint.…"   8 seconds ago   Up 7 seconds   0.0.0.0:80->80/tcp   containerwebserve
```

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Pull images
      docker_image:
        docker_host: tcp://docker:2375
        name: "{{ item }}"
        source: pull
      with_items:
        - centos
        - ubuntu
        - redis
        - nginx
        # n.b. large image, >1GB
        - wernight/funbox

    - name: Create a customised index.html
      copy:
        dest: /shared/index.html
        mode: 0644
        content: |
          Customised page for nginxcustomised
```

```
    - name: Create a customised Dockerfile
      copy:
        dest: /shared/Dockerfile
        mode: 0644
        content: |
          FROM nginx
          COPY index.html /usr/share/nginx/html/index.html

    - name: Build a customised image
      docker_image:
        docker_host: tcp://docker:2375
        name: nginxcustomised:latest
        source: build
        build:
          path: /shared
          pull: yes
        state: present
        force_source: yes

    - name: Create an nginxcustomised container
      docker_container:
        docker_host: tcp://docker:2375
        name: containerwebserver
        image: nginxcustomised:latest
        ports:
          - 80:80
        container_default_behavior: no_defaults
        recreate: yes

# Three dots indicate the end of a YAML document
...
```

```
$ ansible-playbook docker_playbook.yaml

PLAY [ubuntu-c] ****************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [ubuntu-c]

TASK [Pull images] *************************************************************************************************************************************
ok: [ubuntu-c] => (item=centos)
ok: [ubuntu-c] => (item=ubuntu)
ok: [ubuntu-c] => (item=redis)
ok: [ubuntu-c] => (item=nginx)
ok: [ubuntu-c] => (item=wernight/funbox)

TASK [Create a customised index.html] ******************************************************************************************************************
changed: [ubuntu-c]

TASK [Create a customised Dockerfile] ******************************************************************************************************************
changed: [ubuntu-c]

TASK [Build a customised image] ************************************************************************************************************************
changed: [ubuntu-c]

TASK [Create an nginxcustomised container] *************************************************************************************************************
changed: [ubuntu-c]

PLAY RECAP *********************************************************************************************************************************************
ubuntu-c                   : ok=6    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

$ docker images
REPOSITORY        TAG       IMAGE ID       CREATED         SIZE
nginxcustomised   latest    cf3604d437ae   4 seconds ago   134MB
redis             latest    24f9f41dd114   10 days ago     111MB
nginx             latest    cd4e03b35a8e   10 days ago     134MB
ubuntu            latest    a7870fd478f4   3 weeks ago     69.2MB
centos            latest    e6a0117ec169   9 months ago    272MB
wernight/funbox   latest    538c146646c3   4 years ago     1.12GB
```

2-4. Using Ansible to connect running Containers

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
```

```
  hosts: ubuntu-c

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Pull python image
      docker_image:
        docker_host: tcp://docker:2375
        name: python:3.8.5
        source: pull

    - name: Create 3 python containers
      docker_container:
        docker_host: tcp://docker:2375
        name: "python{{ item }}"
        image: python:3.8.5
        container_default_behavior: no_defaults
        command: sleep infinity
      with_sequence: 1-3
-

  # Hosts: where our play will run and options it will run with
  hosts: containers
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Ping containers
      ping:

# Three dots indicate the end of a YAML document
...
```

```
[control]
ubuntu-c

[centos]
centos[1:3]

[ubuntu]
ubuntu[1:3]

[linux:children]
centos
ubuntu

[containers]
python[1:3] ansible_connection=docker ansible_python_interpreter=/usr/bin/python3
```

2-5. Terminate and remove Docker resources

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:

    - name: Remove old containers
      docker_container:
        docker_host: tcp://docker:2375
        name: "{{ item }}"
        state: absent
        container_default_behavior: no_defaults
      with_items:
        - containerwebserver
        - python1
        - python2
        - python3

    - name: Remove images
      docker_image:
        docker_host: tcp://docker:2375
```

```
        name: "{{ item }}"
        state: absent
      with_items:
        - centos
        - ubuntu
        - redis
        - nginx
        - wernight/funbox
        - nginxcustomised
        - python:3.8.5

    - name: Remove files
      file:
        path: "{{ item }}"
        state: absent
      with_items:
        - /shared/Dockerfile
        - /shared/index.html

# Three dots indicate the end of a YAML document
...
```

```
$ ansible-playbook docker_playbook.yaml

$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
```