



Assessment - 22

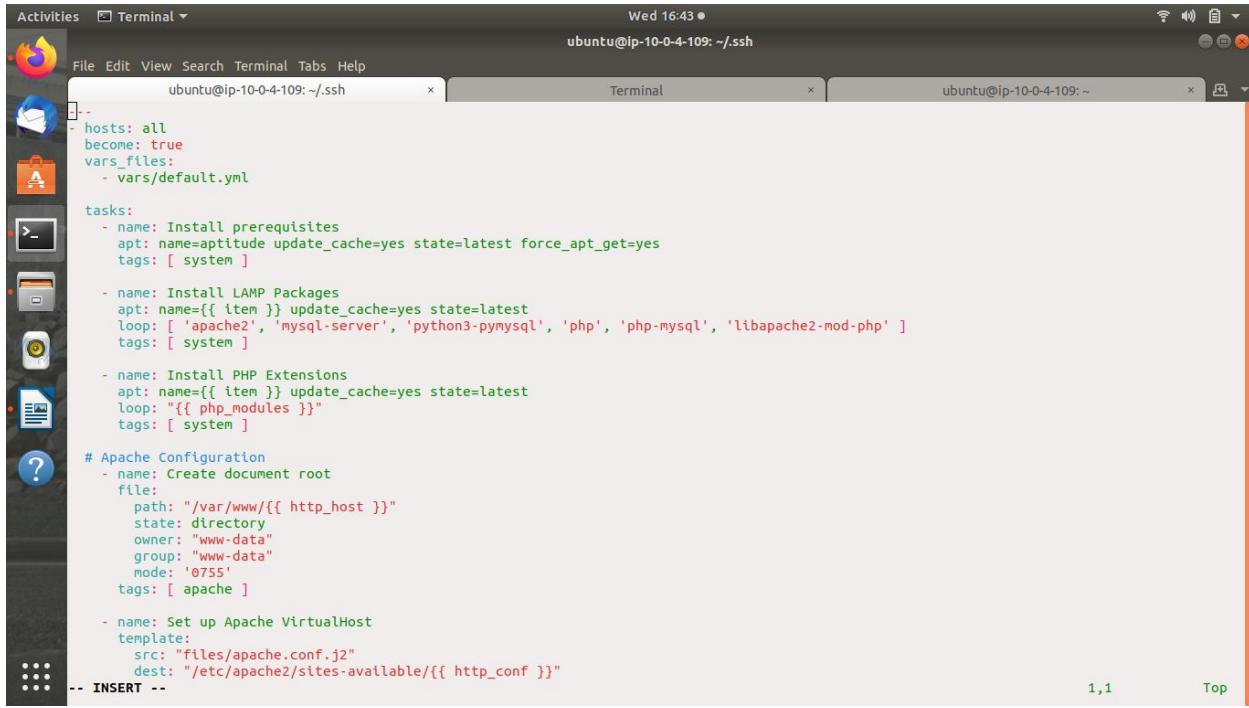
Doubt Resolving-2

Trainee Name : Arun Parmar

Mentor Name : Ravi Kumar

College: UPES

- Project 1
 - Create an ansible playbook to setup WordPress stack on a remote machine



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal" and the command line shows "ubuntu@ip-10-0-4-109: ~". The terminal content displays an Ansible playbook script:

```

hosts: all
become: true
vars_files:
  - vars/default.yml

tasks:
  - name: Install prerequisites
    apt: name=aptitude update_cache=yes state=latest force_apt_get=yes
    tags: [ system ]

  - name: Install LAMP Packages
    apt: name=[ apache2, 'mysql-server', 'python3-pymysql', 'php', 'php-mysql', 'libapache2-mod-php' ]
    tags: [ system ]

  - name: Install PHP Extensions
    apt: name={{ item }} update_cache=yes state=latest
    loop: "{{ php_modules }}"
    tags: [ system ]

# Apache Configuration
- name: Create document root
  file:
    path: "/var/www/{{ http_host }}"
    state: directory
    owner: "www-data"
    group: "www-data"
    mode: '0755'
  tags: [ apache ]

- name: Set up Apache VirtualHost
  template:
    src: "files/apache.conf.j2"
    dest: "/etc/apache2/sites-available/{{ http_conf }}"

```

The terminal window has three tabs: "Activities", "Terminal", and another "Terminal" tab. The status bar at the bottom right shows "1,1 Top".

```
Activities Terminal ▾
File Edit View Search Terminal Tabs Help
ubuntu@ip-10-0-4-109: ~/.ssh x Terminal x ubuntu@ip-10-0-4-109: ~ x
Wed 16:43 ●
ubuntu@ip-10-0-4-109: ~/.ssh
- name: Set up Apache VirtualHost
  template:
    src: "files/apache.conf.j2"
    dest: "/etc/apache2/sites-available/{{ http_conf }}"
  notify: Reload Apache
  tags: [ apache ]

- name: Enable rewrite module
  shell: /usr/sbin/a2enmod rewrite
  notify: Reload Apache
  tags: [ apache ]

- name: Enable new site
  shell: /usr/sbin/a2ensite {{ http_conf }}
  notify: Reload Apache
  tags: [ apache ]

- name: Disable default Apache site
  shell: /usr/sbin/a2dissite 000-default.conf
  notify: Restart Apache
  tags: [ apache ]

# MySQL Configuration
- name: Set the root password
  mysql_user:
    name: root
    password: "{{ mysql_root_password }}"
    login_unix_socket: /var/run/mysql/mysqld.sock
  tags: [ mysql, mysql-root ]

- name: Remove all anonymous user accounts
  mysql_user:
    name: ''
    host_all: yes
  tags: [ mysql ]
-- INSERT --
```

```
Activities Terminal ▾
File Edit View Search Terminal Tabs Help
ubuntu@ip-10-0-4-109: ~/.ssh x Terminal x ubuntu@ip-10-0-4-109: ~ x
Wed 16:43 ●
ubuntu@ip-10-0-4-109: ~/.ssh
- name: Remove all anonymous user accounts
  mysql_user:
    name: ''
    host_all: yes
    state: absent
    login_user: root
    login_password: "{{ mysql_root_password }}"
  tags: [ mysql ]

- name: Remove the MySQL test database
  mysql_db:
    name: test
    state: absent
    login_user: root
    login_password: "{{ mysql_root_password }}"
  tags: [ mysql ]

- name: Creates database for WordPress
  mysql_db:
    name: "{{ mysql_db }}"
    state: present
    login_user: root
    login_password: "{{ mysql_root_password }}"
  tags: [ mysql ]

- name: Create MySQL user for WordPress
  mysql_user:
    name: "{{ mysql_user }}"
    password: "{{ mysql_password }}"
    priv: "{{ mysql_db }}.*:ALL"
    state: present
    login_user: root
    login_password: "{{ mysql_root_password }}"
  tags: [ mysql ]
-- INSERT --
```

Activities Terminal ▾

File Edit View Search Terminal Tabs Help

ubuntu@ip-10-0-4-109: ~/.ssh

Terminal

ubuntu@ip-10-0-4-109: ~

```
Wed 16:43 ●  
ubuntu@ip-10-0-4-109: ~/.ssh  
  
- name: Create MySQL user for WordPress  
  mysql_user:  
    name: "{{ mysql_user }}"  
    password: "{{ mysql_password }}"  
    priv: "{{ mysql_db }}.*:ALL"  
    state: present  
    login_user: root  
    login_password: "{{ mysql_root_password }}"  
    tags: [ mysql ]  
  
# UFW Configuration  
- name: "UFW - Allow HTTP on port {{ http_port }}"  
  ufw:  
    rule: allow  
    port: "{{ http_port }}"  
    proto: tcp  
    tags: [ system ]  
  
# WordPress Configuration  
- name: Download and unpack latest WordPress  
  unarchive:  
    src: https://wordpress.org/latest.tar.gz  
    dest: "/var/www/{{ http_host }}"  
    remote_src: yes  
    creates: "/var/www/{{ http_host }}/wordpress"  
    tags: [ wordpress ]  
  
- name: Set ownership  
  file:  
    path: "/var/www/{{ http_host }}"  
    state: directory  
    recurse: yes  
    owner: www-data  
    group: www-data  
-- INSERT --
```

115,1 76%

Activities Terminal ▾

File Edit View Search Terminal Tabs Help

ubuntu@ip-10-0-4-109: ~/.ssh

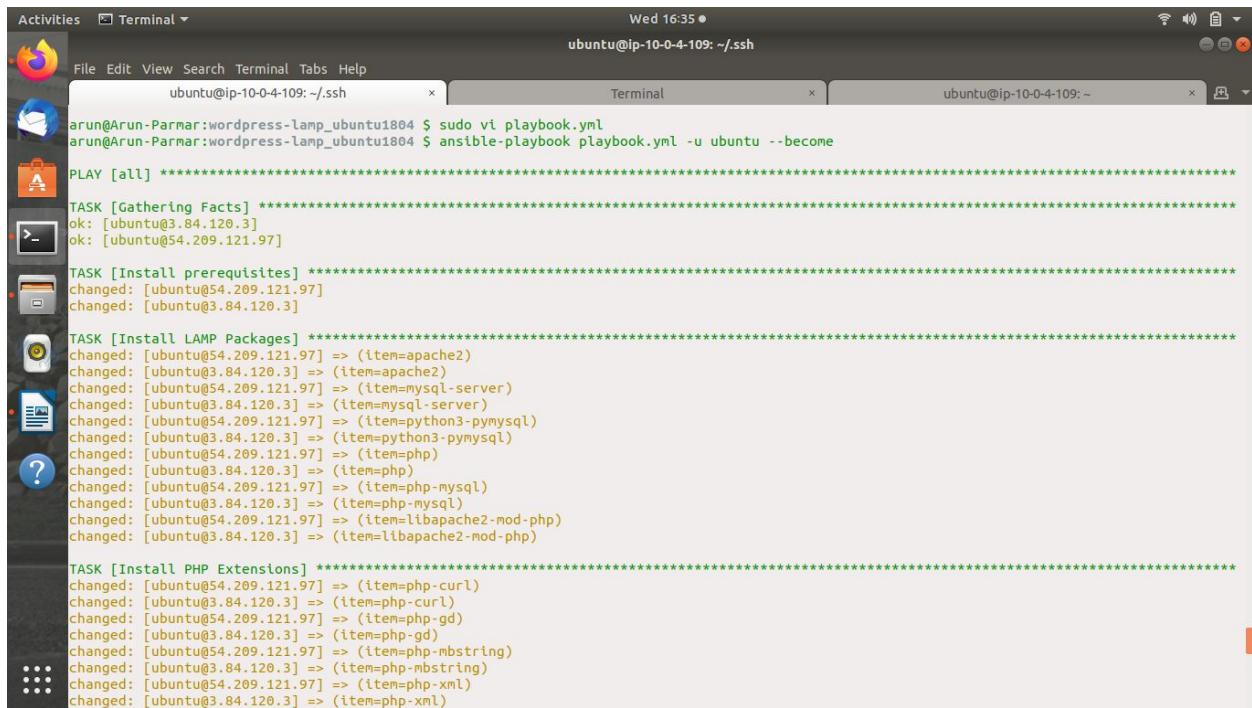
Terminal

ubuntu@ip-10-0-4-109: ~

```
Wed 16:43 ●  
ubuntu@ip-10-0-4-109: ~/.ssh  
  
tags: [ wordpress ]  
  
- name: Set ownership  
  file:  
    path: "/var/www/{{ http_host }}"  
    state: directory  
    recurse: yes  
    owner: www-data  
    group: www-data  
  tags: [ wordpress ]  
  
- name: Set permissions for directories  
  shell: "/usr/bin/find /var/www/{{ http_host }}/wordpress/ -type d -exec chmod 750 {} \\;"  
  tags: [ wordpress ]  
  
- name: Set permissions for files  
  shell: "/usr/bin/find /var/www/{{ http_host }}/wordpress/ -type f -exec chmod 640 {} \\;"  
  tags: [ wordpress ]  
  
- name: Set up wp-config  
  template:  
    src: "files/wp-config.php.j2"  
    dest: "/var/www/{{ http_host }}/wordpress/wp-config.php"  
  tags: [ wordpress ]  
  
handlers:  
- name: Reload Apache  
  service:  
    name: apache2  
    state: reloaded  
  
- name: Restart Apache  
  service:  
    name: apache2  
    state: restarted  
-- INSERT --
```

146,1 Bot

- Run playbook from local



A screenshot of an Ubuntu desktop environment. In the top bar, there are icons for Activities, Terminal, and a window titled "Terminal". The main window is a terminal session titled "Terminal" with the command prompt "ubuntu@ip-10-0-4-109: ~\$". The terminal window displays the output of an Ansible playbook run:

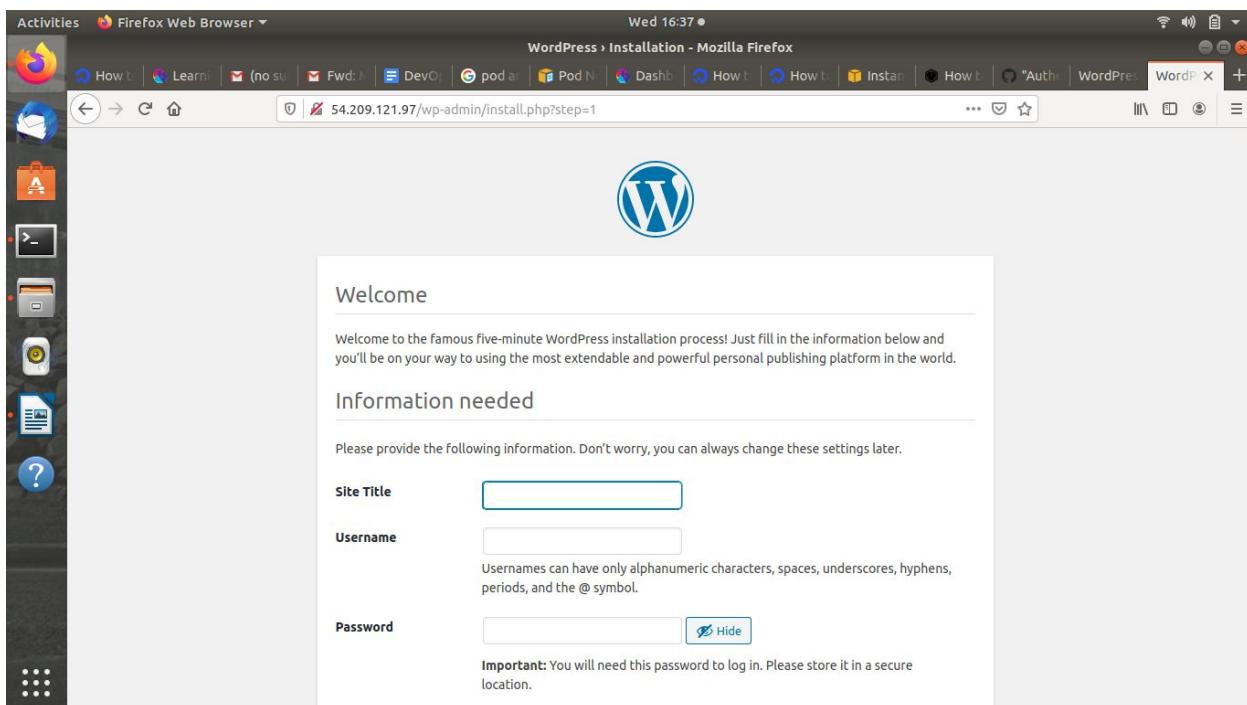
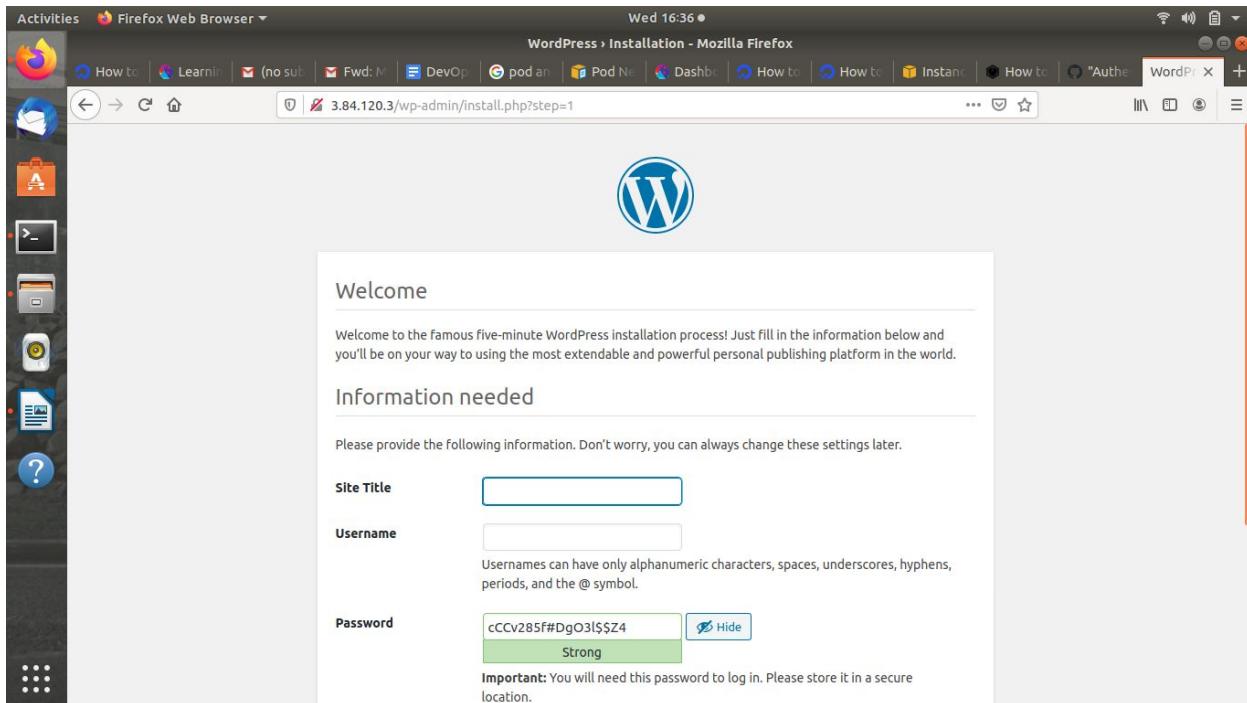
```
ubuntu@ip-10-0-4-109: ~$ sudo vi playbook.yml
arun@Arun-Parmar:~/wordpress-lamp_ubuntu1804$ ansible-playbook playbook.yml -u ubuntu --become

PLAY [all] ****
TASK [Gathering Facts]
ok: [ubuntug3.84.120.3]
ok: [ubuntug54.209.121.97]

TASK [Install prerequisites]
changed: [ubuntug54.209.121.97]
changed: [ubuntug3.84.120.3]

TASK [Install LAMP Packages]
changed: [ubuntug54.209.121.97] => (item=apache2)
changed: [ubuntug3.84.120.3] => (item=apache2)
changed: [ubuntug54.209.121.97] => (item=mysql-server)
changed: [ubuntug3.84.120.3] => (item=mysql-server)
changed: [ubuntug54.209.121.97] => (item=python3-pymysql)
changed: [ubuntug3.84.120.3] => (item=python3-pymysql)
changed: [ubuntug54.209.121.97] => (item=php)
changed: [ubuntug3.84.120.3] => (item=php)
changed: [ubuntug54.209.121.97] => (item=php-mysql)
changed: [ubuntug3.84.120.3] => (item=php-mysql)
changed: [ubuntug54.209.121.97] => (item=libapache2-mod-php)
changed: [ubuntug3.84.120.3] => (item=libapache2-mod-php)

TASK [Install PHP Extensions]
changed: [ubuntug54.209.121.97] => (item=php-curl)
changed: [ubuntug3.84.120.3] => (item=php-curl)
changed: [ubuntug54.209.121.97] => (item=php-gd)
changed: [ubuntug3.84.120.3] => (item=php-gd)
changed: [ubuntug54.209.121.97] => (item=php-mbstring)
changed: [ubuntug3.84.120.3] => (item=php-mbstring)
changed: [ubuntug54.209.121.97] => (item=php-xml)
changed: [ubuntug3.84.120.3] => (item=php-xml)
```



- Create user sam and mike on your local Linux system and sync dynamically their ssh keys to the target instance.
- Create 2 users and generate their ssh keys.

Activities Terminal ▾

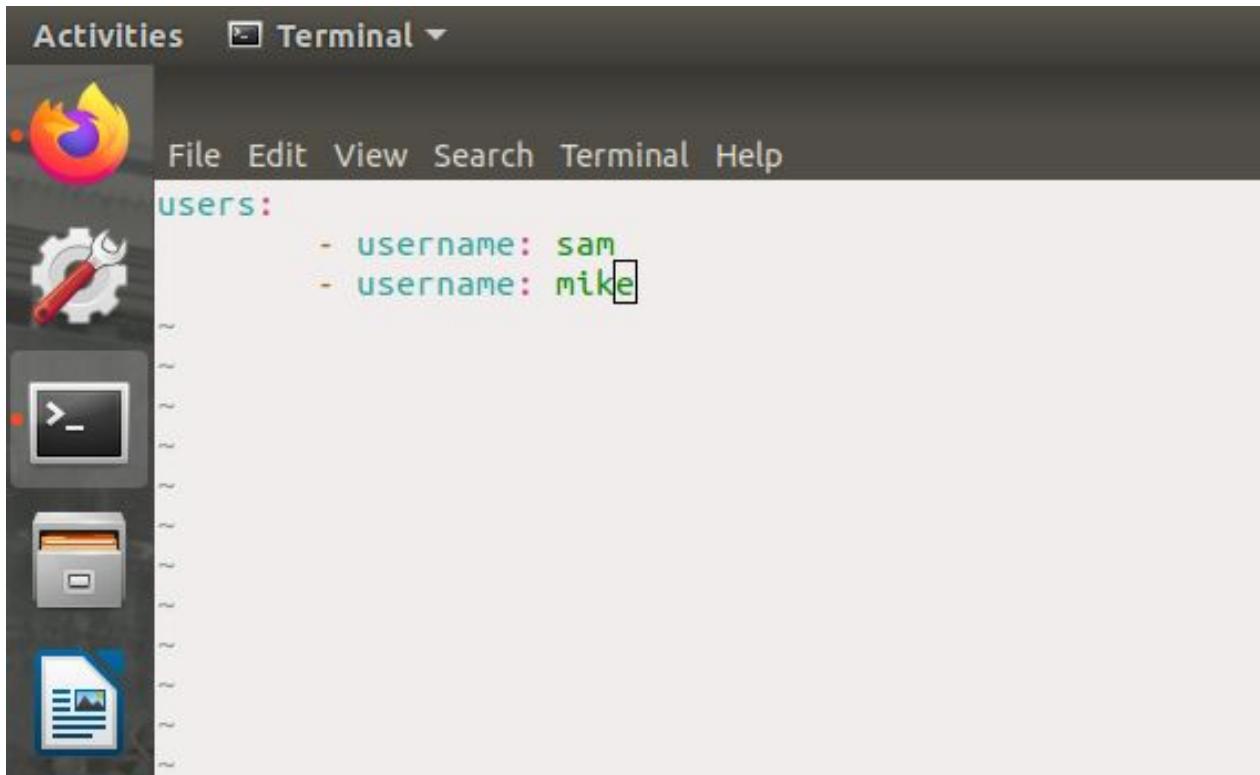
Mon 16:32

sam@Arun-Parmar: ~

```
File Edit View Search Terminal Help
arun@Arun-Parmar:~ $ sudo useradd -m mike
arun@Arun-Parmar:~ $ sudo passwd mike
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
arun@Arun-Parmar:~ $ sudo useradd -m sam
arun@Arun-Parmar:~ $ sudo passwd sam
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
arun@Arun-Parmar:~ $ sudo usermod -s /bin/bash mike
arun@Arun-Parmar:~ $ sudo usermod -s /bin/bash sam
arun@Arun-Parmar:~ $ su sam
Password:
sam@Arun-Parmar:/home/arun$ cd
sam@Arun-Parmar:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/sam/.ssh/id_rsa):
Created directory '/home/sam/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sam/.ssh/id_rsa.
Your public key has been saved in /home/sam/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:dUFo+o9V9h8agYfA0axaJixi1L0q8WXEq9FT7eXvDqQ sam@Arun-Parmar
The key's randomart image is:
+---[RSA 2048]---+
. ..+oo
. o ooo .
. . + .o+ +
o = +.* = oo
. = + S.o..o..
. o * . .+... ..
+ = . E =. o o
. o . oo .
. oo
+---[SHA256]---+
sam@Arun-Parmar:~$
```

```
mike@Arun-Parmar:~/home/sam$ cd
mike@Arun-Parmar:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mike/.ssh/id_rsa):
Created directory '/home/mike/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mike/.ssh/id_rsa.
Your public key has been saved in /home/mike/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:wSXp43k5ur5J7Z3JLDJh2fxbnphwJt43gg97zSpNyE mike@Arun-Parmar
The key's randomart image is:
+---[RSA 2048]---+
. ..
..o
.o .
o. o
.S* ++
*.0 E +
..+.* X+o
.+o.++=0.
.=.=.0Bo.
+---[SHA256]---+
mike@Arun-Parmar:~$
```

Users file



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window is titled "Terminal". The content of the terminal window is as follows:

```
File Edit View Search Terminal Help
users:
  - username: sam
  - username: mike
```

The terminal window has a dark background with light-colored text. The menu bar at the top includes "File", "Edit", "View", "Search", "Terminal", and "Help". To the left of the terminal window, there is a vertical dock with several icons: a Firefox browser icon, a gear and wrench icon, a terminal icon, a folder icon, and a document icon.

Playbook:

The screenshot shows a Linux desktop environment with a terminal window open. The terminal title is "Activities Terminal". The date and time are "Mon 17:36" and the prompt is "ubuntu@ip-172-31-48-219: ~/ssh".

```

hosts: all
become: yes
gather_facts: no
tasks:
  - name: Set authorized key mike
    authorized_key:
      user: ubuntu
      state: present
      key: "{{ lookup('file', '/home/mike/.ssh/id_rsa.pub') }}"

  - name: Set authorized key sam
    authorized_key:
      user: ubuntu
      state: present
      key: "{{ lookup('file', '/home/sam/.ssh/id_rsa.pub') }}"

```

Execution output:

```

arun@Arun-Parmar:playbooks $ sudo vi users.yml
arun@Arun-Parmar:playbooks $ ansible-playbook -i hosts new.yml

PLAY [all] ****
TASK [Set authorized key mike] ****
ok: [ubuntu@54.210.163.188]

TASK [Set authorized key sam] ****
changed: [ubuntu@54.210.163.188]

PLAY RECAP ****
ubuntu@54.210.163.188 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
arun@Arun-Parmar:playbooks $ 

```

SSH session output:

```

zLTUeM5ea1P2hMdB7aoW5zU4c3fJIDjydVz4RodEsWBPKkYZwSxAmiQaqpMFs6+Zg+QYI079Uh+kRQAXJJ1aB0dJ9HLv9ZKP sam@Arun-Parmar
ubuntu@ip-172-31-48-219:~/ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCo8fZFJ/Qz0fZdElkhdrV0mHFQ9gPBj/WxeQx2BwW9j7IAycnW3KCkVhqIXS/BkPN8I2c7b8aioKp0DR0xRMTQvoJrUdC+7rgc0hyUh
G6BUaMVHbu/RU619PamE6UasRtbKB5QwZwEj6v17oBHDGzqmr85b+so+2uRNg+YCr8Fs1WwC0lYve9T5yvBFiqld38YnSFUWAMnzZ0g6ab6+IxNtHEF6f/+tgMUR7EM+1gy4C/5M9u2s
WXXV0N+97RxxE1ylZao0ZC+fNEYsFYZUTiEuw9Ve0NGIMaoaEjySackx2W4AgjC201enHTrn8uq1kba5ks161laJwuCC7 mike@Arun-Parmar
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCo8fZFJ/Qz0fZdElkhdrV0mHFQ9gPBj/WxeQx2BwW9j7IAycnW3KCkVhqIXS/BkPN8I2c7b8aioKp0DR0xRMTQvoJrUdC+7rgc0hyUh
G6BUaMVHbu/RU619PamE6UasRtbKB5QwZwEj6v17oBHDGzqmr85b+so+2uRNg+YCr8Fs1WwC0lYve9T5yvBFiqld38YnSFUWAMnzZ0g6ab6+IxNtHEF6f/+tgMUR7EM+1gy4C/5M9u2s
WXXV0N+97RxxE1ylZao0ZC+fNEYsFYZUTiEuw9Ve0NGIMaoaEjySackx2W4AgjC201enHTrn8uq1kba5ks161laJwuCC7 mike@Arun-Parmar
ubuntu@ip-172-31-48-219:~/ssh$ 

```

- Project 2
 - Create a docker image to run nginx
 - Create an image of tomcat with a sample war file.
- Docker file for nginx

Activities Terminal ▾

Mon 17:59
ubuntu@ip-172-31-48-219: ~/.ssh

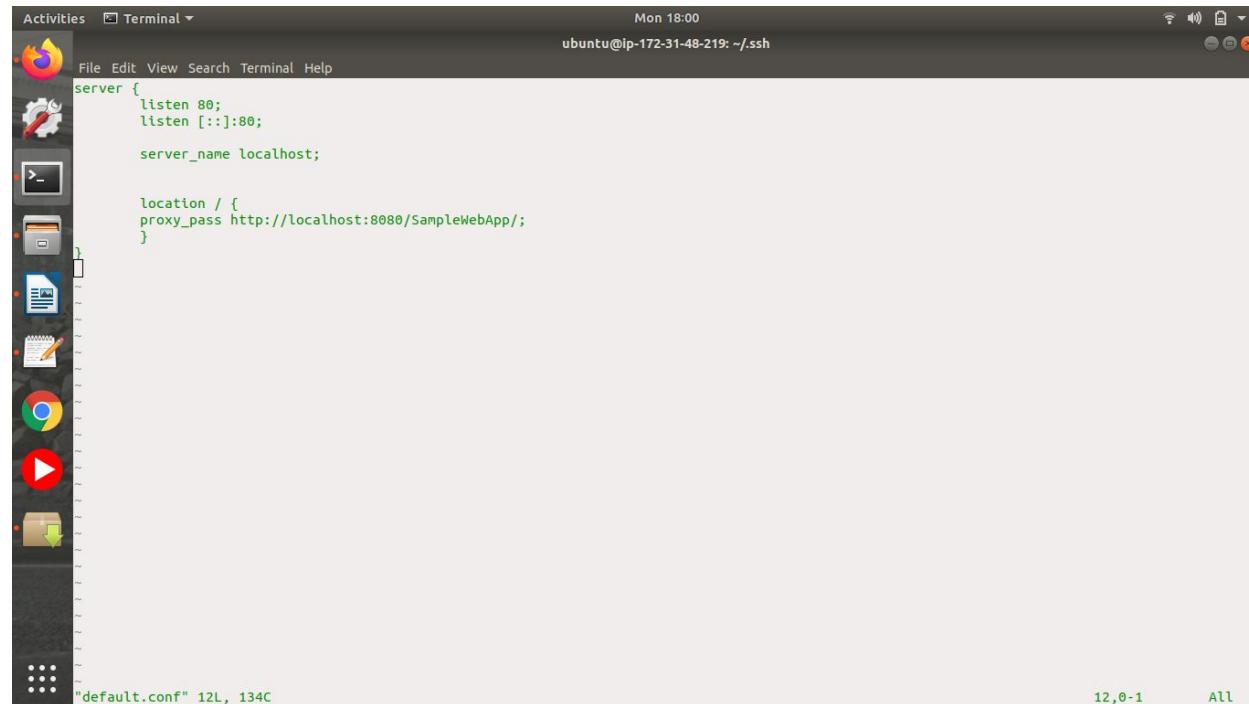
The terminal window displays a Dockerfile with the following content:

```
FROM nginx
RUN rm /etc/nginx/conf.d/default.conf
COPY default.conf /etc/nginx/conf.d
```

The terminal interface includes a vertical application menu on the left and a status bar at the bottom.

"Dockerfile" 4L, 86C

Conf file



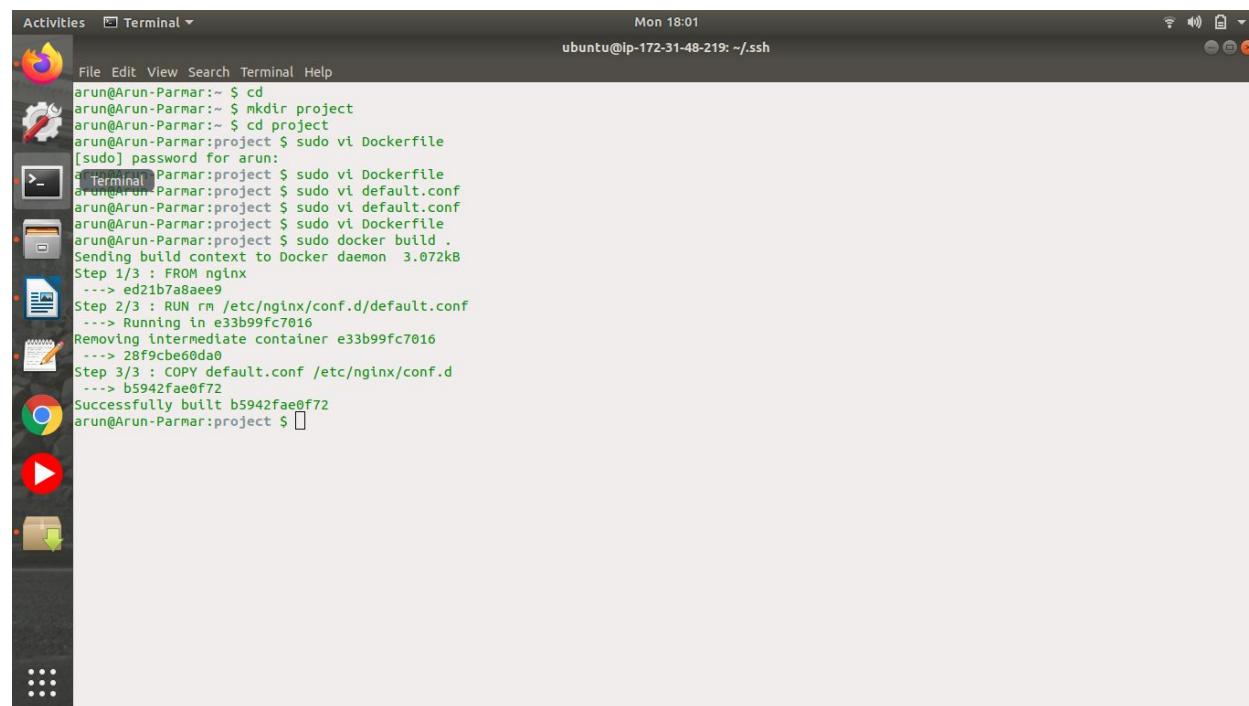
```
Activities Terminal Mon 18:00
ubuntu@ip-172-31-48-219: ~/.ssh

File Edit View Search Terminal Help
server {
    listen 80;
    listen [::]:80;

    server_name localhost;

    location / {
        proxy_pass http://localhost:8080/SampleWebApp;
    }
}

"default.conf" 12L, 134C
```



```
Activities Terminal Mon 18:01
ubuntu@ip-172-31-48-219: ~/.ssh

File Edit View Search Terminal Help
arun@Arun-Parmar:~ $ cd
arun@Arun-Parmar:~ $ mkdir project
arun@Arun-Parmar:~ $ cd project
arun@Arun-Parmar:project $ sudo vi Dockerfile
[sudo] password for arun:
arun@Arun-Parmar:project $ sudo vi Dockerfile
arun@Arun-Parmar:project $ sudo vi default.conf
arun@Arun-Parmar:project $ sudo vi default.conf
arun@Arun-Parmar:project $ sudo vi Dockerfile
arun@Arun-Parmar:project $ sudo docker build .
Sending build context to Docker daemon 3.072kB
Step 1/3 : FROM nginx
--> ed21b7aa8ae9
Step 2/3 : RUN rm /etc/nginx/conf.d/default.conf
--> Running in e33b99fc7016
Removing intermediate container e33b99fc7016
--> 28f9cbe60da0
Step 3/3 : COPY default.conf /etc/nginx/conf.d/
--> b5942fae0f72
Successfully built b5942fae0f72
arun@Arun-Parmar:project $
```

```
Activities Terminal Mon 18:08
ubuntu@ip-172-31-48-219: ~/.ssh

File Edit View Search Terminal Help
arun@Arun-Parmar:project $ sudo vi Dockerfile
arun@Arun-Parmar:project $ sudo vi default.conf
arun@Arun-Parmar:project $ sudo vi default.conf
arun@Arun-Parmar:project $ sudo vi Dockerfile
arun@Arun-Parmar:project $ sudo docker build .
Sending build context to Docker daemon 3.072kB
Step 1/3 : FROM nginx
--> ed21b7a8ae9
Step 2/3 : RUN rm /etc/nginx/conf.d/default.conf
--> Running in e3b99fc7016
Removing intermediate container e3b99fc7016
--> 28f9cbe0da0
Step 3/3 : COPY default.conf /etc/nginx/conf.d
--> b5942fae0f72
Successfully built b5942fae0f72
arun@Arun-Parmar:project $ docker tag b5942fae0f72 arun/nginx:nginx_task
WARNING: Error loading config file: /home/arun/.docker/config.json: stat /home/arun/.docker/config.json: permission denied
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.40/images/b5942fae0f72/tag?repo=arun%2Fnginx&tag=nginx_task: dial unix /var/run/docker.sock: connect: permission denied
arun@Arun-Parmar:project $ sudo docker tag b5942fae0f72 arun/nginx:nginx_task
arun@Arun-Parmar:project $ sudo docker push arun/nginx:nginx_task
The push refers to repository [docker.io/arun/nginx]
3186314b7a52: Preparing
0$bd2d9adfc3: Preparing
d37eeb5b769: Preparing
99134ec7f247: Preparing
c3a984abe8a8: Preparing
denied: requested access to the resource is denied
arun@Arun-Parmar:project $ sudo docker tag b5942fae0f72 kunwararan/nginx:nginx_task
arun@Arun-Parmar:project $ sudo docker push kunwararan/nginx:nginx_task
The push refers to repository [docker.io/kunwararan/nginx]
3186314b7a52: Pushed
0$bd2d9adfc3: Pushed
d37eeb5b769: Pushed
99134ec7f247: Pushed
c3a984abe8a8: Pushed
nginx_task: digest: sha256:f40e17f27d397a14f54f891f81338602c70631571427cbbb9513bd64d8b12429 size: 1362
arun@Arun-Parmar:project $
```

```
Activities Terminal Mon 18:08
ubuntu@ip-172-31-48-219: ~/.ssh

File Edit View Search Terminal Help
Successfully built b5942fae0f72
arun@Arun-Parmar:project $ docker tag b5942fae0f72 arun/nginx:nginx_task
WARNING: Error loading config file: /home/arun/.docker/config.json: stat /home/arun/.docker/config.json: permission denied
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.40/images/b5942fae0f72/tag?repo=arun%2Fnginx&tag=nginx_task: dial unix /var/run/docker.sock: connect: permission denied
arun@Arun-Parmar:project $ sudo docker tag b5942fae0f72 arun/nginx:nginx_task
The push refers to repository [docker.io/arun/nginx]
3186314b7a52: Preparing
0$bd2d9adfc3: Preparing
d37eeb5b769: Preparing
99134ec7f247: Preparing
c3a984abe8a8: Preparing
denied: requested access to the resource is denied
arun@Arun-Parmar:project $ sudo docker tag b5942fae0f72 kunwararan/nginx:nginx_task
arun@Arun-Parmar:project $ sudo docker push kunwararan/nginx:nginx_task
The push refers to repository [docker.io/kunwararan/nginx]
3186314b7a52: Pushed
0$bd2d9adfc3: Pushed
d37eeb5b769: Pushed
99134ec7f247: Pushed
c3a984abe8a8: Pushed
nginx_task: digest: sha256:f40e17f27d397a14f54f891f81338602c70631571427cbbb9513bd64d8b12429 size: 1362
arun@Arun-Parmar:project $ docker images
WARNING: Error loading config file: /home/arun/.docker/config.json: stat /home/arun/.docker/config.json: permission denied
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.40/images/json: dial unix /var/run/docker.sock: connect: permission denied
arun@Arun-Parmar:project $ sudo docker images
REPOSITORY          TAG           IMAGE ID        CREATED         SIZE
arun/nginx          nginx_task    b5942fae0f72   7 minutes ago   127MB
kunwararan/nginx    nginx_task    b5942fae0f72   7 minutes ago   127MB
<none>              <none>        ecc19dd8a464   3 hours ago    127MB
nginx               latest        ed21b7a8ae9    13 days ago    127MB
docker              latest        fc43bbebe6abe  6 weeks ago    221MB
ubuntu              latest        72300a873c2c  7 weeks ago    64.2MB
hello-world         latest        fce289e99eb9   15 months ago   1.84KB
tomcat              8.0          ef6a7c98d192   19 months ago   356MB
arun@Arun-Parmar:project $
```

Docker file for tomcat

Activities Terminal ▾

Mon 19:57

Laptop battery low
Approximately 41 minutes remaining (10%)

File Edit View Search Terminal Tabs Help

ubuntu@ip-172-31-48-219: ~/.ssh

```
FROM tomcat
WORKDIR /usr/local/tomcat/webapps
RUN curl -O -L https://github.com/AKSarav/SampleWebApp/raw/master/dist/SampleWebApp.war
```

"Dockerfile" 4L, 135C

4,0-1 All

Activities Terminal ▾

Mon 18:17

File Edit View Search Terminal Help

ubuntu@ip-172-31-48-219: ~/.ssh

```
inspect  Display detailed information on one or more images
load    Load an image from a tar archive or STDIN
ls      List images
prune   Remove unused images
pull    Pull an image or a repository from a registry
push    Push an image or a repository to a registry
rm      Remove one or more images
save    Save one or more images to a tar archive (streamed to STDOUT by default)
tag     Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

Run 'docker image COMMAND --help' for more information on a command.
arun@Arun-Parmar:tomcat $ sudo docker images
REPOSITORY          TAG           IMAGE ID            CREATED             SIZE
kunwararun/nginx   nginx_task   d4d61c0cf82f        2 minutes ago       528MB
arunp/nginx        nginx_task   b5942fae0f72        14 minutes ago      127MB
<none>              <none>       ecc19ddba464        3 hours ago        127MB
tomcat              latest        6ab907c973d2        3 days ago         528MB
nginx               latest        ed21b7a8aee9        13 days ago        127MB
docker              latest        fc43bbebe6abe       6 weeks ago        221MB
ubuntu              latest        72300a873c2c        7 weeks ago        64.2MB
hello-world         latest        fce289e99eb9        15 months ago      1.84kB
tomcat              8.0          ef6a7c98d192       19 months ago      356MB
arun@Arun-Parmar:tomcat $ sudo docker tag d4d61c0cf82f kunwararun/docker_task
arun@Arun-Parmar:tomcat $ sudo docker images
REPOSITORY          TAG           IMAGE ID            CREATED             SIZE
kunwararun/docker   docker_task  d4d61c0cf82f        4 minutes ago       528MB
kunwararun/nginx   nginx_task   d4d61c0cf82f        4 minutes ago       528MB
arunp/nginx        nginx_task   b5942fae0f72        16 minutes ago      127MB
<none>              <none>       b5942fae0f72        16 minutes ago      127MB
<none>              <none>       ecc19ddba464        3 hours ago        127MB
tomcat              latest        6ab907c973d2        3 days ago         528MB
nginx               latest        ed21b7a8aee9        13 days ago        127MB
docker              latest        fc43bbebe6abe       6 weeks ago        221MB
ubuntu              latest        72300a873c2c        7 weeks ago        64.2MB
hello-world         latest        fce289e99eb9        15 months ago      1.84kB
tomcat              8.0          ef6a7c98d192       19 months ago      356MB
arun@Arun-Parmar:tomcat $
```

```
Activities Terminal ▾ Mon 18:19
ubuntu@ip-172-31-48-219: ~/.ssh

File Edit View Search Terminal Help
arun/nginx      nginx_task      b5942fae0f72   14 minutes ago  127MB
arunwararun/nginx <none>        b5942fae0f72   14 minutes ago  127MB
<none>          <none>        ecc19dd8a464   3 hours ago   127MB
tomcat          latest         6ab9907c973d2   3 days ago    528MB
nginx           latest         ed21b7a8aee9   13 days ago   127MB
nginx           latest         fc43bebe64be   6 weeks ago   221MB
Terminal        latest         72300a873c2c   7 weeks ago   64.2MB
ubuntu          latest         fce289e99eb9   15 months ago  1.84kB
hello-world     latest         fce289e99eb9   15 months ago  1.84kB
tomcat          8.0           efa7c98d192   19 months ago  356MB
arun@Arun-Parmar:tomcat $ sudo docker tag d4d61c0cf82f kunwararun/docker_task
arun@Arun-Parmar:tomcat $ sudo docker images
REPOSITORY          TAG           IMAGE ID      CREATED       SIZE
kunwararun/docker  docker_task  d4d61c0cf82f  4 minutes ago  528MB
kunwararun/nginx   nginx_task  d4d61c0cf82f  4 minutes ago  528MB
arun/nginx          nginx_task  b5942fae0f72   16 minutes ago  127MB
kunwararun/nginx   <none>       b5942fae0f72   16 minutes ago  127MB
<none>              <none>       ecc19dd8a464   3 hours ago   127MB
tomcat             latest         6ab9907c973d2   3 days ago    528MB
nginx              latest         ed21b7a8aee9   13 days ago   127MB
docker              latest         fc43bebe64be   6 weeks ago   221MB
ubuntu             latest         72300a873c2c   7 weeks ago   64.2MB
hello-world         latest         fce289e99eb9   15 months ago  1.84kB
tomcat             8.0           efa7c98d192   19 months ago  356MB
arun@Arun-Parmar:tomcat $ sudo docker push kunwararun/docker:docker_task
The push refers to repository [docker.io/kunwararun/docker]
9a2b6ab05822: Pushed
fb8f657b05a7: Mounted from library/tomcat
87af55901360: Mounted from library/tomcat
81349fc07565: Mounted from library/tomcat
892007193bb6: Mounted from library/tomcat
e811ee12aa10: Mounted from library/tomcat
23fd8d486123a: Mounted from library/tomcat
afae6f50abb9: Mounted from library/tomcat
136a15f81f25: Mounted from library/tomcat
185574602537: Mounted from library/tomcat
24efcd549ab5: Mounted from library/tomcat
docker_task: digest: sha256:996117bb0377ccbc67c6649b5182e7c652e0bcccf25c396d3f37c450fa5c8a68 size: 2629
arun@Arun-Parmar:tomcat $
```

- Create an EKS cluster.
- Serve the application in tomcat using nginx and EKS service.

Activities Terminal ▾ Mon 19:02
ubuntu@ip-172-31-48-219: ~/ssh

```
[sudo] password for arun:
arun@Arun-Parmar:~ $ sudo vi EKSservice
arun@Arun-Parmar:~ $ eksctl create cluster EKSservice
[1] eksctl version 0.16.0
[1] using region us-east-1
[1] setting availability zones to [us-east-1b us-east-1c]
[1] subnets for us-east-1b - public:192.168.0.0/19 private:192.168.64.0/19
[1] subnets for us-east-1c - public:192.168.32.0/19 private:192.168.96.0/19
[1] nodegroup "ng-429c3007" will use "ami-of15d55736fd476da" [AmazonLinux2/1.14]
[1] using Kubernetes version 1.14
[1] creating EKS cluster "EKSservice" in "us-east-1" region with un-managed nodes
[1] will create 2 separate CloudFormation stacks for cluster itself and the initial nodegroup
[1] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=EKSservice'
[1] CloudWatch logging will not be enabled for cluster "EKSservice" in "us-east-1"
[1] you can enable it with 'eksctl utils update-cluster-logging --region=us-east-1 --cluster=EKSservice'
[1] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "EKSservice" in "us-east-1"
[1] 2 sequential tasks: { create cluster control plane "EKSservice", create nodegroup "ng-429c3007" }
[1] building cluster stack "eksctl-EKSservice-cluster"
[1] deploying stack "eksctl-EKSservice-cluster"
[1] building nodegroup stack "eksctl-EKSservice-nodegroup-ng-429c3007"
[1] --nodes-min=2 was set automatically for nodegroup ng-429c3007
[1] --nodes-max=2 was set automatically for nodegroup ng-429c3007
[1] deploying stack "eksctl-EKSservice-nodegroup-ng-429c3007"
[✓] all EKS cluster resources for "EKSservice" have been created
[✓] saved kubeconfig as "/home/arun/.kube/config"
[✓] adding identity "arn:aws:iam::187632318301:role/eksctl-EKSservice-nodegroup-ng-42-NodeInstanceRole-TKVN6T7QJMGS" to auth ConfigMap
[1] nodegroup "ng-429c3007" has 0 node(s)
[1] waiting for at least 2 node(s) to become ready in "ng-429c3007"
[1] nodegroup "ng-429c3007" has 2 node(s)
[1] node "ip-192-168-1-32.ec2.internal" is ready
[1] node "ip-192-168-51-75.ec2.internal" is ready
[1] kubectl command should work with "/home/arun/.kube/config", try 'kubectl get nodes'
[✓] EKS cluster "EKSservice" in "us-east-1" region is ready
arun@Arun-Parmar:~ $ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-192-168-1-32.ec2.internal Ready <none> 2m14s v1.14.9-eks-1f0ca9
ip-192-168-51-75.ec2.internal Ready <none> 2m14s v1.14.9-eks-1f0ca9
arun@Arun-Parmar:~ $
```

Activities Terminal ▾ Mon 19:05
ubuntu@ip-172-31-48-219: ~/ssh

```
File Edit View Search Terminal Help
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: kunwararun/nginx:nginx_task
          ports:
            - containerPort: 80
          resources:
            requests:
              memory: "200Mi"
              cpu: "350m"
        - name: tomcat
          image: kunwararun/docker:docker_task
          ports:
            - containerPort: 8080
          resources:
            requests:
              memory: "200Mi"
              cpu: "350m"
```

~
~
~
~
~

"deploy.yml" 32L, 616C

25,25 All

Activities Terminal ▾ Mon 19:06
ubuntu@ip-172-31-48-219: ~/.ssh

```
[i] node "ip-192-168-51-75.ec2.internal" is ready
[i] kubectl command should work with "/home/arun/.kube/config", try 'kubectl get nodes'
[!] EKS cluster "EKSservice" in "us-east-1" region is ready
arun@Arun-Parmar:~ $ kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-192-168-51-32.ec2.internal   Ready    <none>   2m14s  v1.14.9-eks-1f0ca9
ip-192-168-51-75.ec2.internal   Ready    <none>   2m14s  v1.14.9-eks-1f0ca9
arun@Arun-Parmar:~ $ kubectl get svc
NAME          TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP  10.100.0.1   <none>        443/TCP   8m36s
arun@Arun-Parmar:~ $ cd project/
arun@Arun-Parmar:project $ ls
default.conf Dockerfile tomcat
arun@Arun-Parmar:project $ sudo vi deploy.yml
[sudo] password for arun:
arun@Arun-Parmar:project $ sudo vi deploy.yml
arun@Arun-Parmar:project $ kubectl apply -f deploy.yml
deployment.apps/nginx created
arun@Arun-Parmar:project $ kubectl get pods
NAME           READY   STATUS      RESTARTS   AGE
nginx-5fcf8d8fc-8zpn  0/2    ContainerCreating   0          14s
nginx-5fcf8d8fc-m87js 0/2    ContainerCreating   0          14s
arun@Arun-Parmar:project $ kubectl get pods
NAME           READY   STATUS      RESTARTS   AGE
nginx-5fcf8d8fc-8zpn  2/2    Running     0          62s
nginx-5fcf8d8fc-m87js 2/2    Running     0          62s
arun@Arun-Parmar:project $
```

Activities Terminal ▾ Mon 19:08
ubuntu@ip-172-31-48-219: ~/.ssh

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
  - port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  type: NodePort
```

-- INSERT --

Activities Terminal ▾ Mon 19:08
ubuntu@ip-172-31-48-219: ~/.ssh

```
arun@Arun-Parmar:~ $ kubectl get nodes
NAME           STATUS  ROLES   AGE    VERSION
ip-192-168-1-32.ec2.internal  Ready   <none>  2m14s  v1.14.9-eks-1f0ca9
ip-192-168-51-75.ec2.internal  Ready   <none>  2m14s  v1.14.9-eks-1f0ca9
arun@Arun-Parmar:~ $ kubectl get svc
NAME         TYPE      CLUSTER-IP  EXTERNAL-IP  PORT(S)   AGE
kubernetes   ClusterIP  10.100.0.1  <none>        443/TCP   8m36s
arun@Arun-Parmar:~ $ cd
arun@Arun-Parmar:~ $ .ansible/
arun@Arun-Parmar:~ $ ansible-playbooks/
arun@Arun-Parmar:~ $ assignments/
arun@Arun-Parmar:~ $ .aws/
arun@Arun-Parmar:~ $ LibreOffice Writer
arun@Arun-Parmar:~ $ .config/
arun@Arun-Parmar:~ $ Desktop/
arun@Arun-Parmar:~ $ dir/
arun@Arun-Parmar:~ $ docker/
arun@Arun-Parmar:~ $ cd project/
arun@Arun-Parmar:project $ ls
default.conf  Dockerfile  tomcat
arun@Arun-Parmar:project $ sudo vi deploy.yml
[sudo] password for arun:
arun@Arun-Parmar:project $ sudo vi deploy.yml
arun@Arun-Parmar:project $ kubectl apply -f deploy.yml
deployment.apps/nginx created
arun@Arun-Parmar:project $ kubectl get pods
NAME          READY  STATUS    RESTARTS  AGE
nginx-5fcf8d8fc-8zpz  0/2   ContainerCreating  0   14s
nginx-5fcf8d8fc-m87js 0/2   ContainerCreating  0   14s
arun@Arun-Parmar:project $ kubectl get pods
NAME          READY  STATUS    RESTARTS  AGE
nginx-5fcf8d8fc-8zpz  2/2   Running   0   62s
nginx-5fcf8d8fc-m87js 2/2   Running   0   62s
arun@Arun-Parmar:project $ sudo vi svc.yml
arun@Arun-Parmar:project $ kubectl apply -f svc.yml
service/nginx created
arun@Arun-Parmar:project $
```

```

Activities Terminal ▾ Mon 19:48
ubuntu@ip-172-31-48-219: ~/ssh
File Edit View Search Terminal Tabs Help
ubuntu@ip-172-31-48-219: ~/ssh x Terminal x
Activities Terminal ▾ Mon 19:48
ubuntu@ip-172-31-48-219: ~/ssh
File Edit View Search Terminal Tabs Help
ubuntu@ip-172-31-48-219: ~/ssh x Terminal x

```

enp4s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
ether 98:fa:b8:ef:64:ef txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 19660 bytes 1804568 (1.8 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 19660 bytes 1804568 (1.8 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp5s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.225.90 netmask 255.255.255.0 broadcast 192.168.225.255
inet6 2409:4053:708:2125:b0a1:83cf:3a57:1480 prefixlen 64 scopeid 0x0<global>
inet6 2409:4053:708:2125:cfa0:ea68:6c44:734a prefixlen 64 scopeid 0x0<global>
inet6 fe80::829b:347b:6098:d7c7 prefixlen 64 scopeid 0x20<link>
ether 7c:b2:7d:8d:f9:a0 txqueuelen 1000 (Ethernet)
RX packets 298298 bytes 303899052 (303.8 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 175717 bytes 101718430 (101.7 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

arun@Arun-Parmar:project \$ kubectl get nodes -o wide
NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION
CONTAINER-RUNTIME
ip-192-168-1-32.ec2.internal Ready <none> 48m v1.14.9-eks-1f0ca9 192.168.1.32 3.223.191.100 Amazon Linux 2 4.14.173-137.229
.ipmn2.x86_64 docker://18.9.9
ip-192-168-51-75.ec2.internal Ready <none> 48m v1.14.9-eks-1f0ca9 192.168.51.75 3.92.176.125 Amazon Linux 2 4.14.173-137.229
.ipmn2.x86_64 docker://18.9.9
arun@Arun-Parmar:project \$

```

Activities Terminal ▾ Mon 19:48
ubuntu@ip-172-31-48-219: ~/ssh
File Edit View Search Terminal Tabs Help
ubuntu@ip-172-31-48-219: ~/ssh x Terminal x
Activities Terminal ▾ Mon 19:48
ubuntu@ip-172-31-48-219: ~/ssh
File Edit View Search Terminal Tabs Help
ubuntu@ip-172-31-48-219: ~/ssh x Terminal x

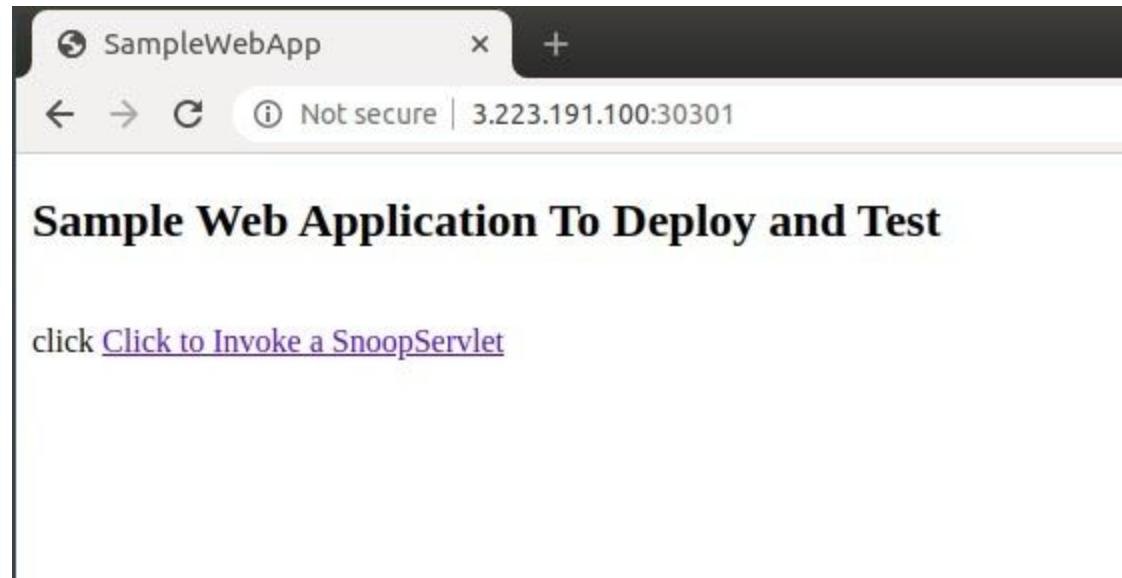
```

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp5s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.225.90 netmask 255.255.255.0 broadcast 192.168.225.255
inet6 2409:4053:708:2125:b0a1:83cf:3a57:1480 prefixlen 64 scopeid 0x0<global>
inet6 2409:4053:708:2125:cfa0:ea68:6c44:734a prefixlen 64 scopeid 0x0<global>
inet6 fe80::829b:347b:6098:d7c7 prefixlen 64 scopeid 0x20<link>
ether 7c:b2:7d:8d:f9:a0 txqueuelen 1000 (Ethernet)
RX packets 298298 bytes 303899052 (303.8 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 175717 bytes 101718430 (101.7 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

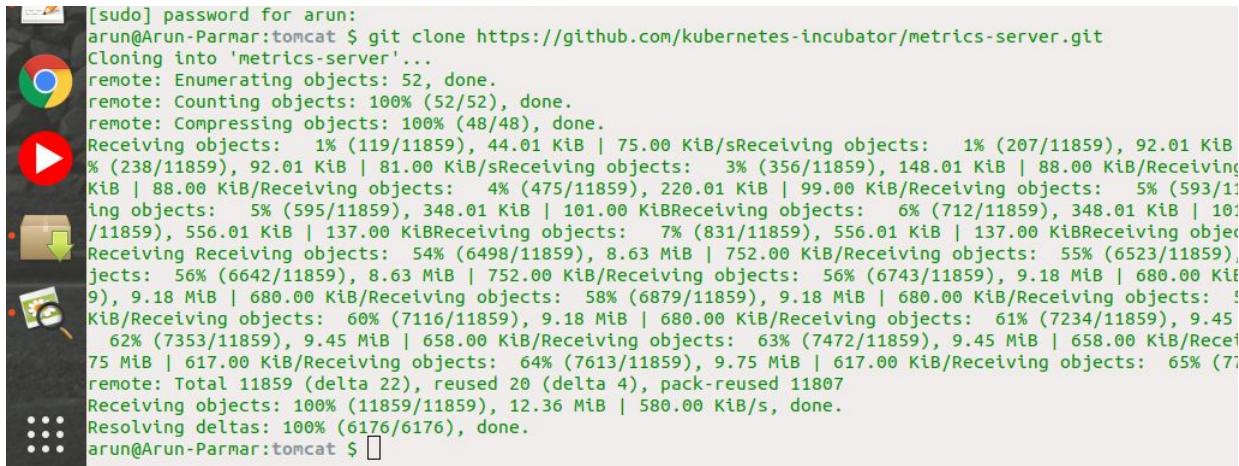
arun@Arun-Parmar:project \$ kubectl get nodes -o wide
NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION
CONTAINER-RUNTIME
ip-192-168-1-32.ec2.internal Ready <none> 48m v1.14.9-eks-1f0ca9 192.168.1.32 3.223.191.100 Amazon Linux 2 4.14.173-137.229
.ipmn2.x86_64 docker://18.9.9
ip-192-168-51-75.ec2.internal Ready <none> 48m v1.14.9-eks-1f0ca9 192.168.51.75 3.92.176.125 Amazon Linux 2 4.14.173-137.229
.ipmn2.x86_64 docker://18.9.9
arun@Arun-Parmar:project \$ kubectl get pods -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
nginx-5fcf8d8fc-8zpzr 2/2 Running 0 42m 192.168.8.210 ip-192-168-1-32.ec2.internal <none> <none>
nginx-5fcf8d8fc-m87js 2/2 Running 0 42m 192.168.60.244 ip-192-168-51-75.ec2.internal <none> <none>
arun@Arun-Parmar:project \$ kubectl get nodes -o wide
NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION
CONTAINER-RUNTIME
ip-192-168-1-32.ec2.internal Ready <none> 48m v1.14.9-eks-1f0ca9 192.168.1.32 3.223.191.100 Amazon Linux 2 4.14.173-137.229
.ipmn2.x86_64 docker://18.9.9
ip-192-168-51-75.ec2.internal Ready <none> 48m v1.14.9-eks-1f0ca9 192.168.51.75 3.92.176.125 Amazon Linux 2 4.14.173-137.229
.ipmn2.x86_64 docker://18.9.9
arun@Arun-Parmar:project \$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 54m
nginx NodePort 10.100.21.151 <none> 80:30301/TCP 39m
arun@Arun-Parmar:project \$

Get port and hit on ip:port



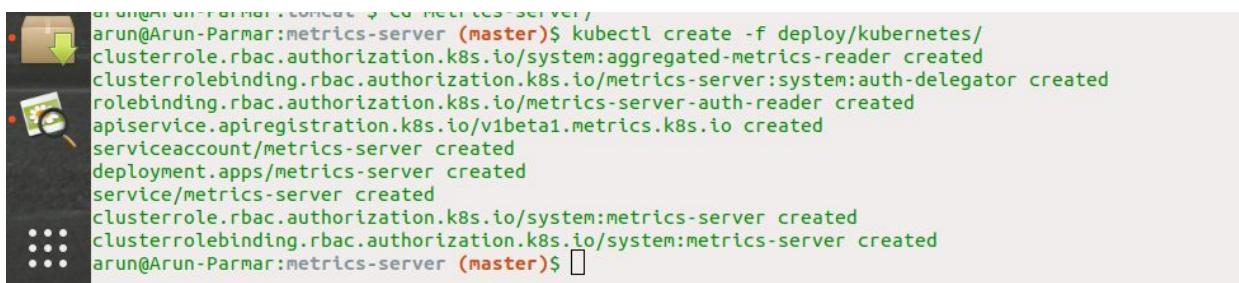
- Create service and hpa on basis of CPU util for this deployment

Clone the GitHub repository of metrics-server



```
[sudo] password for arun:  
arun@Arun-Parmar:tomcat $ git clone https://github.com/kubernetes-incubator/metrics-server.git  
Cloning into 'metrics-server'...  
remote: Enumerating objects: 52, done.  
remote: Counting objects: 100% (52/52), done.  
remote: Compressing objects: 100% (48/48), done.  
Receiving objects:  1% (119/11859), 44.01 KiB | 75.00 KiB/sReceiving objects:  1% (207/11859), 92.01 KiB  
% (238/11859), 92.01 KiB | 81.00 KiB/sReceiving objects:  3% (356/11859), 148.01 KiB | 88.00 KiB/Receiving  
KiB | 88.00 KiB/Receiving objects:  4% (475/11859), 220.01 KiB | 99.00 KiB/Receiving objects:  5% (593/11859),  
Receiving objects:  5% (595/11859), 348.01 KiB | 101.00 KiB/Receiving objects:  6% (712/11859), 348.01 KiB | 101.00  
KiB/Receiving objects:  7% (831/11859), 556.01 KiB | 137.00 KiB/Receiving objects:  7% (831/11859), 556.01 KiB | 137.00  
KiB/Receiving objects:  54% (6498/11859), 8.63 MiB | 752.00 KiB/Receiving objects:  55% (6523/11859), 8.63 MiB | 752.00  
KiB/Receiving objects:  56% (6642/11859), 8.63 MiB | 752.00 KiB/Receiving objects:  56% (6743/11859), 9.18 MiB | 680.00 KiB/  
9%, 9.18 MiB | 680.00 KiB/Receiving objects:  58% (6879/11859), 9.18 MiB | 680.00 KiB/Receiving objects:  59%  
KiB/Receiving objects:  60% (7116/11859), 9.18 MiB | 680.00 KiB/Receiving objects:  61% (7234/11859), 9.45 MiB |  
62% (7353/11859), 9.45 MiB | 658.00 KiB/Receiving objects:  63% (7472/11859), 9.45 MiB | 658.00 KiB/  
75 MiB | 617.00 KiB/Receiving objects:  64% (7613/11859), 9.75 MiB | 617.00 KiB/Receiving objects:  65% (7753/11859),  
remote: Total 11859 (delta 22), reused 20 (delta 4), pack-reused 11807  
Receiving objects: 100% (11859/11859), 12.36 MiB | 580.00 KiB/s, done.  
Resolving deltas: 100% (6176/6176), done.  
arun@Arun-Parmar:tomcat $
```

Install Metrics Server from the root of the Metrics Server directory.



```
arun@Arun-Parmar:~ $ cd metrics-server/  
arun@Arun-Parmar:metrics-server (master)$ kubectl create -f deploy/kubernetes/  
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created  
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created  
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created  
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created  
serviceaccount/metrics-server created  
deployment.apps/metrics-server created  
service/metrics-server created  
clusterrole.rbac.authorization.k8s.io/system:metrics-server created  
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created  
arun@Arun-Parmar:metrics-server (master)$
```

Confirm that the Metrics Server is running.



```
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created  
arun@Arun-Parmar:metrics-server (master)$ kubectl get pods -n kube-system  
NAME                      READY   STATUS    RESTARTS   AGE  
aws-node-2xr8t            1/1     Running   0          70m  
aws-node-492md            1/1     Running   0          70m  
coredns-56678dcf76-7qxqs  1/1     Running   0          76m  
coredns-56678dcf76-pm95r  1/1     Running   0          76m  
kube-proxy-f9ftj          1/1     Running   0          70m  
kube-proxy-vfgrs          1/1     Running   0          70m  
metrics-server-64c67868bd-g6w6c 1/1     Running   0          3m1s  
arun@Arun-Parmar:metrics-server (master)$ sudo vi hpa.yaml
```

Hpa.yaml file

Activities Terminal ▾

Mon 20:14
ubuntu@ip-172-31-48-219: ~/ssh

```
File Edit View Search Terminal Tabs Help  
ubuntu@ip-172-31-48-219: ~/ssh  
  
apiVersion: autoscaling/v2beta1  
kind: HorizontalPodAutoscaler  
metadata:  
  name: hpa-deployment  
spec:  
  maxReplicas: 10  
  minReplicas: 1  
  scaleTargetRef:  
    apiVersion: apps/v1  
    kind: Deployment  
    name: nginx  
metrics:  
- type: Resource  
  resource:  
    name: cpu  
  targetAverageUtilization: 50
```

Apply file

```
arun@Arun-Parmar:metrics-server (master)$ sudo vi hpa.yml  
arun@Arun-Parmar:metrics-server (master)$ kubectl apply -f hpa.yml  
horizontalpodautoscaler.autoscaling/hpa-deployment created  
arun@Arun-Parmar:metrics-server (master)$ kubectl get svc  
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE  
kubernetes  ClusterIP  10.100.0.1   <none>        443/TCP     77m  
nginx      NodePort   10.100.21.151  <none>        80:30301/TCP  63m  
arun@Arun-Parmar:metrics-server (master)$ kubectl get hpa  
NAME      REFERENCE      TARGETS      MINPODS      MAXPODS      REPLICAS      AGE  
hpa-deployment  Deployment/nginx  0%/50%      1           10          2            28s  
arun@Arun-Parmar:metrics-server (master)$ kubectl get hpa -o wide  
NAME      REFERENCE      TARGETS      MINPODS      MAXPODS      REPLICAS      AGE  
hpa-deployment  Deployment/nginx  0%/50%      1           10          2            34s  
arun@Arun-Parmar:metrics-server (master)$
```

- Project 3 : Deploy Wordpress on this cluster and it should connect to a MySQL installed on Kubernetes. Using PVC

First we launched a cluster with following yml file

Activities Terminal ▾

File Edit View Search Terminal Help

Terminal

Mon

eksctl.io/v1alpha5

kind: ClusterConfig

metadata:

name: zpvc

region: us-east-1

vpc:

id: "vpc-093a4253d4c9ab207"

cidr: "10.0.0.0/16"

subnets:

public:

us-east-1a:

id: "subnet-0af58143cd499547c"

cidr: "10.0.0.0/24"

us-east-1b:

id: "subnet-0342b9c54db410a4e"

cidr: "10.0.2.0/24"

us-east-1c:

id: "subnet-007036a437b689c11"

cidr: "10.0.4.0/24"

us-east-1e:

id: "subnet-060196ddd1ae6f536"

cidr: "10.0.6.0/24"

iam:

serviceRoleARN: "arn:aws:iam::187632318301:role/eks-service-role"

managedNodeGroups:

- name: myng1

instanceType: t3.medium

desiredCapacity: 2

minSize: 1

maxSize: 3

availabilityZones: ["us-east-1a", "us-east-1b", "us-east-1c", "us-east-1e"]

volumeSize: 15

#volumeType: gp2

tags:

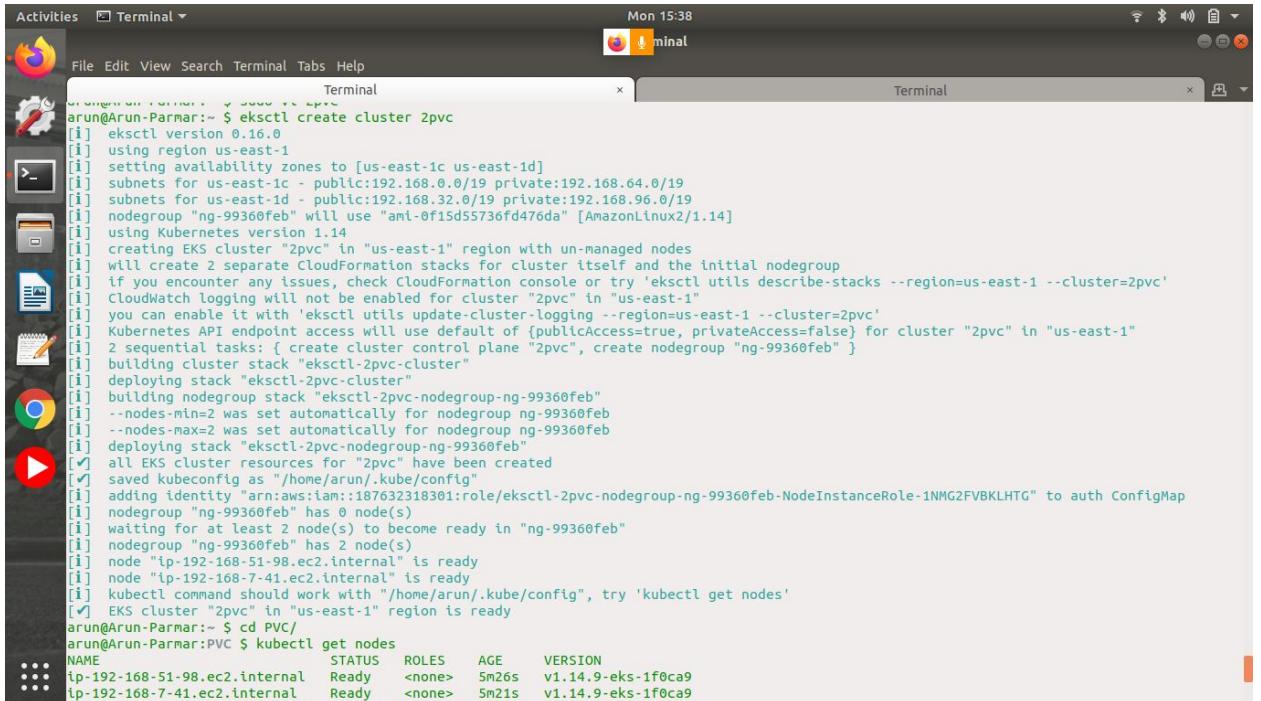
owner: arun

purpose: ekscluster

ssh:

publicKeyName: ttnaccount.pem

allow: true



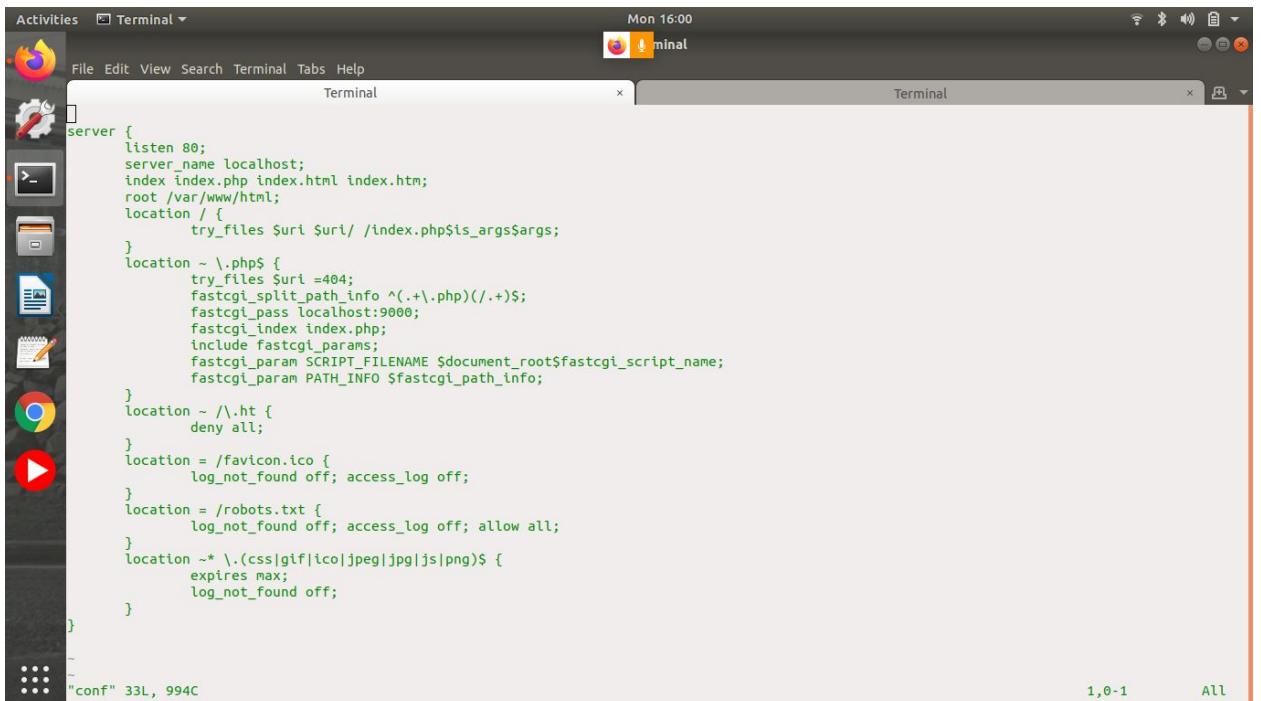
A screenshot of an Ubuntu desktop environment. On the left is a dock with icons for the Dash, Terminal, Home, Applications, and Help. A central terminal window shows the command `eksctl create cluster 2pvc` being run, followed by its output. The output details the creation of an EKS cluster named '2pvc' in the 'us-east-1' region, including the creation of two CloudFormation stacks, nodegroups, and Kubernetes resources. It also shows the cluster becoming ready with two nodes. Below this, a command `kubectl get nodes` is run, displaying two healthy nodes.

```

arun@Arun-Parmar:~ $ eksctl create cluster 2pvc
[1] eksctl version 0.16.0
[1] using region us-east-1
[1] setting availability zones to [us-east-1c us-east-1d]
[1] subnets for us-east-1c - public:192.168.0.0/19 private:192.168.64.0/19
[1] subnets for us-east-1d - public:192.168.32.0/19 private:192.168.96.0/19
[1] nodegroup "ng-99360feb" will use "ami-of15d55736fd476da" [AmazonLinux2/1.14]
[1] using Kubernetes version 1.14
[1] creating EKS cluster "2pvc" in "us-east-1" region with un-managed nodes
[1] will create 2 separate Cloudformation stacks for cluster itself and the initial nodegroup
[1] if you encounter any issues, check Cloudformation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=2pvc'
[1] CloudWatch logging will not be enabled for cluster "2pvc" in "us-east-1"
[1] you can enable it with 'eksctl utils update-cluster-logging --region=us-east-1 --cluster=2pvc'
[1] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "2pvc" in "us-east-1"
[1] 2 sequential tasks: [ create cluster control plane "2pvc", create nodegroup "ng-99360feb" ]
[1] building cluster stack "eksctl-2pvc-cluster"
[1] deploying stack "eksctl-2pvc-cluster"
[1] building nodegroup stack "eksctl-2pvc-nodegroup-ng-99360feb"
[1] --nodes-min=2 was set automatically for nodegroup ng-99360feb
[1] --nodes-max=2 was set automatically for nodegroup ng-99360feb
[1] deploying stack "eksctl-2pvc-nodegroup-ng-99360feb"
[✓] all EKS cluster resources for "2pvc" have been created
[✓] saved kubeconfig as "/home/arun/.kube/config"
[✓] adding identity "arn:aws:iam::187632318301:role/eksctl-2pvc-nodegroup-ng-99360feb-NodeInstanceRole-1NMG2FVBKLHTG" to auth ConfigMap
[1] nodegroup "ng-99360feb" has 0 node(s)
[1] waiting for at least 2 node(s) to become ready in "ng-99360feb"
[1] nodegroup "ng-99360feb" has 2 node(s)
[1] node "ip-192-168-51-98.ec2.internal" is ready
[1] node "ip-192-168-7-41.ec2.internal" is ready
[1] kubectl command should work with "/home/arun/.kube/config", try 'kubectl get nodes'
[✓] EKS cluster "2pvc" in "us-east-1" region is ready
arun@Arun-Parmar:~ $ cd PVC/
arun@Arun-Parmar:PVC $ kubectl get nodes
NAME           STATUS    ROLES   AGE     VERSION
ip-192-168-51-98.ec2.internal   Ready    <none>   5m26s   v1.14.9-eks-1f0ca9
ip-192-168-7-41.ec2.internal    Ready    <none>   5m21s   v1.14.9-eks-1f0ca9

```

Also, add the ports you will use to the security group.
Till cluster is being up, we will create concerned files.
Wordpress config file:



A screenshot of an Ubuntu desktop environment. On the left is a dock with icons for the Dash, Terminal, Home, Applications, and Help. A central terminal window shows a complex Nginx configuration file. The file defines a server block for port 80, handling PHP requests via FastCGI and static files via location blocks for .php, .ico, .txt, and various image formats like .css, .gif, .jpeg, .jpg, .js, and .png. It includes directives for log levels and file paths.

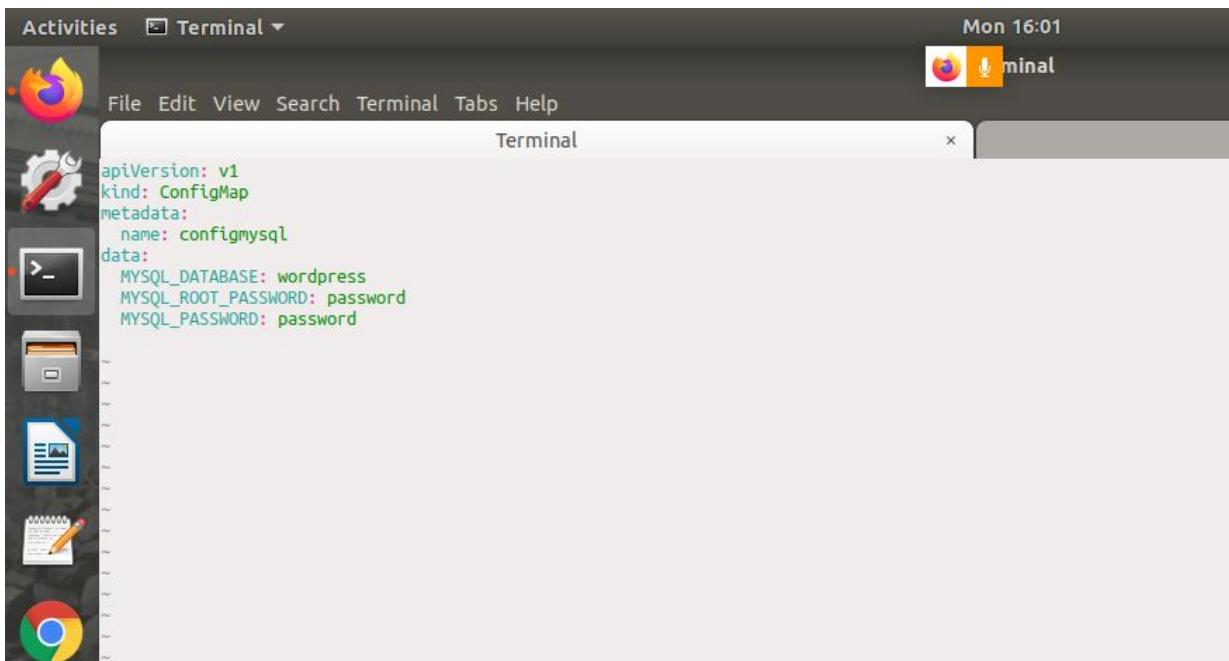
```

server {
    listen 80;
    server_name localhost;
    index index.php index.html index.htm;
    root /var/www/html;
    location / {
        try_files $uri $uri/ /index.php$is_args$args;
    }
    location ~ \.php$ {
        try_files $uri =404;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass localhost:9000;
        fastcgi_index index.php;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
    }
    location ~ /\.ht {
        deny all;
    }
    location = /favicon.ico {
        log_not_found off; access_log off;
    }
    location = /robots.txt {
        log_not_found off; access_log off; allow all;
    }
    location ~* \.(css|gif|ico|jpeg|jpg|js|png)$ {
        expires max;
        log_not_found off;
    }
}

```

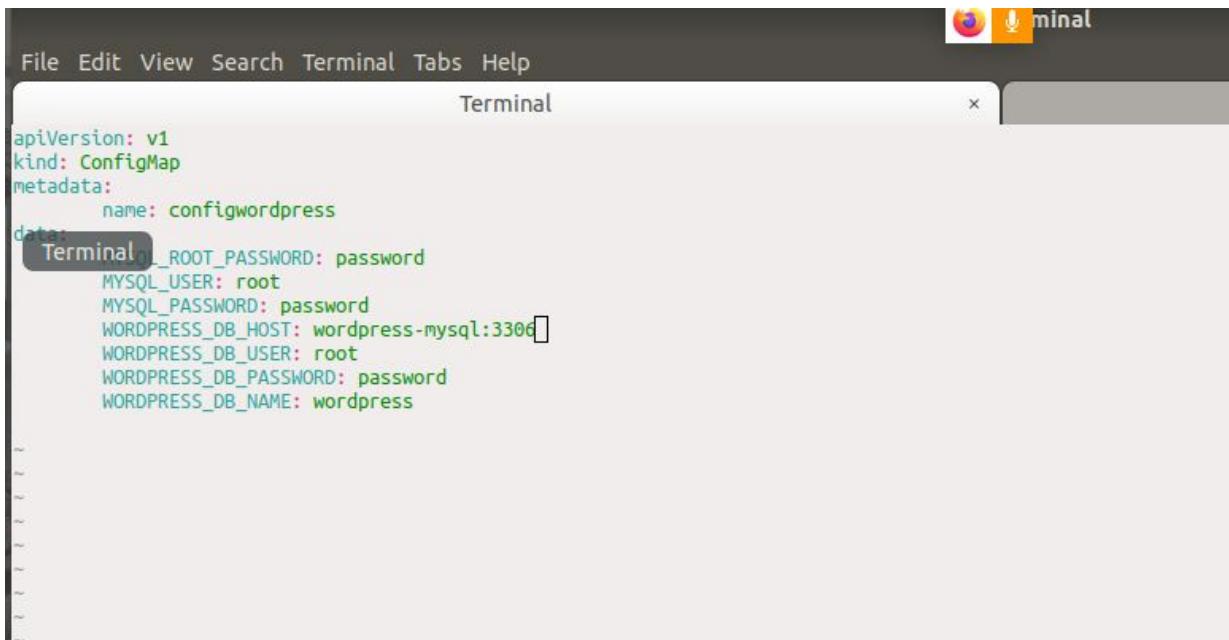
"conf" 33L, 994C

Mysql configmap file:



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: configmysql
data:
  MYSQL_DATABASE: wordpress
  MYSQL_ROOT_PASSWORD: password
  MYSQL_PASSWORD: password
```

Wordpress configmap file:



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: configwordpress
data:
  WORDPRESS_DB_HOST: wordpress-mysql:3306
  WORDPRESS_DB_USER: root
  WORDPRESS_DB_PASSWORD: password
  WORDPRESS_DB_NAME: wordpress
```

Mysql deploying file

Activities Terminal ▾

File Edit View Search Terminal Tabs Help

Terminal

Mon 16:01

Firefox Terminal

```
apiVersion: v1
kind: Service
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  ports:
    - port: 3306
  selector:
    app: wordpress
  clusterIP: None
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  selector:
    matchLabels:
      app: wordpress
  template:
    metadata:
      labels:
        app: wordpress
    spec:
      containers:
        - name: mysql
          image: mysql:5.6
          ports:
            - containerPort: 3306
          envFrom:
            - configMapRef:
                name: configmysql
          volumeMounts:
            - name: mysql-persistent-storage
              mountPath: /var/lib/mysql
          volumes:
            - name: mysql-persistent-storage
              persistentVolumeClaim:
                claimName: mysql-pv-claim
-- INSERT --
```

Wordpress Deploying file

Activities Terminal ▾

File Edit View Search Terminal Tabs Help

Terminal

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wp-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
  ...
  ...
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        ###### PHP Container ######
        - name: dphp
          image: wordpress:5.1.1-fpm-alpine
          ports:
            - containerPort: 9000
          envFrom:
            - configMapRef:
                name: configwordpress
          volumeMounts:
            - name: wordpress-persistent-storage
              mountPath: /var/www/html
        ###### Nginx Container ######
        - name: nginx
          image: 201998/projecteks:wordpress
          volumeMounts:
            - name: wordpress-persistent-storage
              mountPath: /var/www/html
  volumes:
    - name: wordpress-persistent-storage
      persistentVolumeClaim:
        claimName: wp-pv-claim
  ...
  ...
apiVersion: v1
kind: Service
metadata:
  ...
  -- INSERT --
```

Docker file:

Build the image.
Apply all the yml files.

Activities Terminal ▾

File Edit View Search Terminal Tabs Help

Terminal x Mon 15:38

arun@Arun-Parmar:~ \$ cd PVC/
arun@Arun-Parmar:PVC \$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-192-168-51-98.ec2.internal Ready <none> 5m26s v1.14.9-eks-1f0ca9
ip-192-168-7-41.ec2.internal Ready <none> 5m21s v1.14.9-eks-1f0ca9
arun@Arun-Parmar:PVC \$ kubectl apply -f mysql.yml
configmap/configmysql created
arun@Arun-Parmar:PVC \$ kubectl get configmaps
NAME DATA AGE
configmysql 3 28s
arun@Arun-Parmar:PVC \$ kubectl -f apply mysql_deploy.yml
Error: unknown command "mysql_deploy.yml" for "kubectl"
Run 'kubectl --help' for usage.
unknown command "mysql_deploy.yml" for "kubectl"
arun@Arun-Parmar:PVC \$ kubectl apply -f mysql_deploy.yml
service/wordpress-mysql created
persistentvolumeclaim/mysql-pv-claim created
deployment.apps/wordpress-mysql created
arun@Arun-Parmar:PVC \$ kubectl get svc -o wide
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE SELECTOR
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 14m <none>
wordpress-mysql ClusterIP None <none> 3306/TCP 18s app=wordpress
arun@Arun-Parmar:PVC \$ kubectl apply -f wp_wide
wp_deploy.yml
arun@Arun-Parmar:PVC \$ kubectl apply -f wordpress.yml
configmap/configwordpress created
arun@Arun-Parmar:PVC \$ kubectl apply -f wp_deploy.yml
persistentvolumeclaim/wp-pv-claim created
deployment.apps/nginx-deployment created
service/my-service created
arun@Arun-Parmar:PVC \$ kubectl get svc -o wide
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE SELECTOR
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 14m <none>
my-service NodePort 10.100.206.168 <none> 80:30487/TCP 10s app=nginx
wordpress-mysql ClusterIP None <none> 3306/TCP 63s app=wordpress
arun@Arun-Parmar:PVC \$

Activities Terminal ▾

File Edit View Search Terminal Tabs Help

Terminal x Mon 15:38

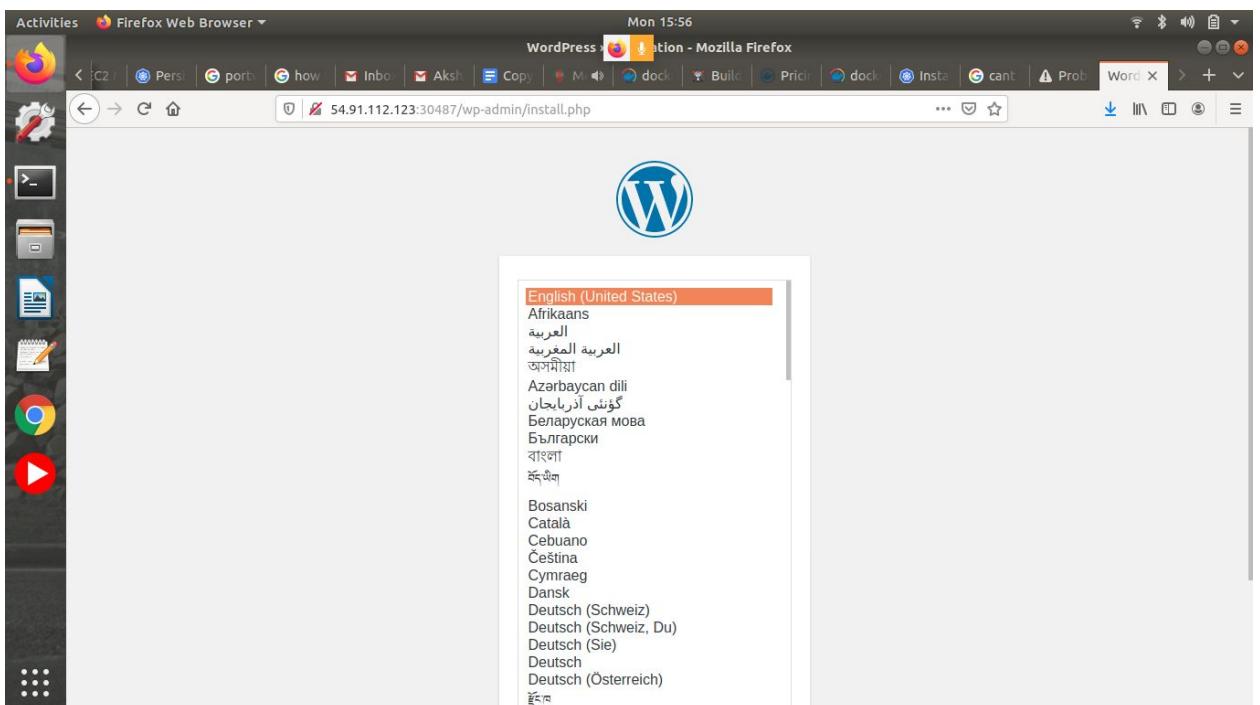
ip-192-168-7-41.ec2.internal Ready <none> 5m21s v1.14.9-eks-1f0ca9
arun@Arun-Parmar:PVC \$ kubectl apply -f mysql.yml
configmap/configmysql created
arun@Arun-Parmar:PVC \$ kubectl get configmaps
NAME DATA AGE
configmysql 3 28s
arun@Arun-Parmar:PVC \$ kubectl -f apply mysql_deploy.yml
Error: unknown command "mysql_deploy.yml" for "kubectl"
Run 'kubectl --help' for usage.
unknown command "mysql_deploy.yml" for "kubectl"
arun@Arun-Parmar:PVC \$ kubectl apply -f mysql_deploy.yml
service/wordpress-mysql created
persistentvolumeclaim/mysql-pv-claim created
deployment.apps/wordpress-mysql created
arun@Arun-Parmar:PVC \$ kubectl get svc -o wide
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE SELECTOR
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 14m <none>
wordpress-mysql ClusterIP None <none> 3306/TCP 18s app=wordpress
arun@Arun-Parmar:PVC \$ kubectl apply -f w
wordpress.yml wp Deploy.y
arun@Arun-Parmar:PVC \$ kubectl apply -f wordpress.yml
configmap/configwordpress created
arun@Arun-Parmar:PVC \$ kubectl apply -f wp Deploy.y
persistentvolumeclaim/wp-pv-claim created
deployment.apps/nginx-deployment created
service/my-service created
arun@Arun-Parmar:PVC \$ kubectl get svc -o wide
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE SELECTOR
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 14m <none>
my-service NodePort 10.100.206.168 <none> 80:30487/TCP 10s app=nginx
wordpress-mysql ClusterIP None <none> 3306/TCP 63s app=wordpress
arun@Arun-Parmar:PVC \$ kubectl get deployment
NAME READY UP-TO-DATE AVAILABLE AGE
nginx-deployment 2/2 2 2 41s
wordpress-mysql 1/1 1 1 92s
arun@Arun-Parmar:PVC \$

Activities Terminal ▾

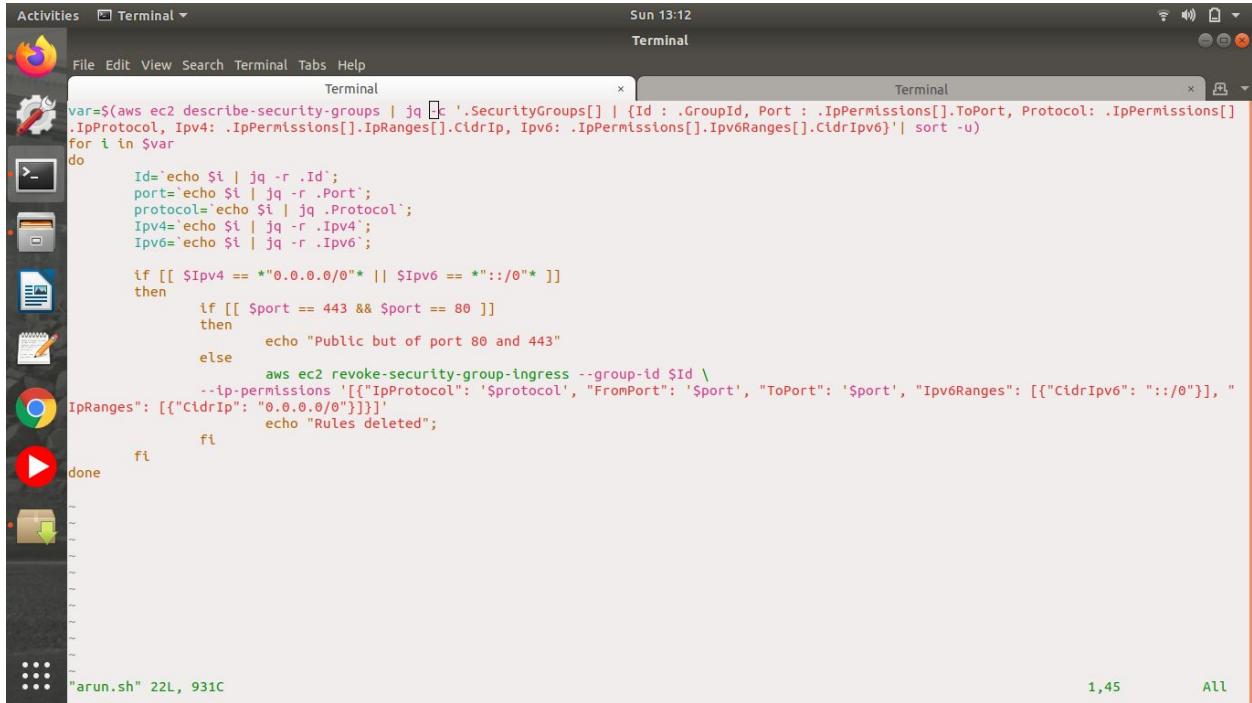
```

configmap/configwordpress created
arun@Arun-Parmar:PVC $ kubectl apply -f wp_deploy.yml
persistentvolumeclaim/wp_pv-claim created
deployment.apps/nginx-deployment created
service/service created
arun@Arun-Parmar:PVC $ kubectl get svc -o wide
NAME          TYPE      CLUSTER-IP      EXTERNAL-IP     PORT(S)        AGE   SELECTOR
kubernetes    ClusterIP  10.100.0.1    <none>        443/TCP       14m   <none>
my-service    NodePort   10.100.206.168 <none>        80:30487/TCP  10s   app=nginx
wordpress-mysql ClusterIP  None         <none>        3306/TCP      63s   app=wordpress
arun@Arun-Parmar:PVC $ kubectl get deployment
NAME        READY  UP-TO-DATE  AVAILABLE  AGE
nginx-deployment  2/2    2          2          41s
wordpress-mysql  1/1    1          1          92s
arun@Arun-Parmar:PVC $ kubectl get nodes -o wide
NAME           STATUS  ROLES   AGE   VERSION
ip-192-168-51-98.ec2.internal  Ready  <none>  8m30s v1.14.9-eks-1f0ca9  192.168.51.98  52.91.222.99  Amazon Linux 2  4.14.173-137.2
ip-192-168-7-41.ec2.internal  Ready  <none>  8m25s v1.14.9-eks-1f0ca9  192.168.7.41   54.91.112.123  Amazon Linux 2  4.14.173-137.2
29.amzn2.x86_64              docker://18.9.9
29.amzn2.x86_64              docker://18.9.9
arun@Arun-Parmar:PVC $ kubectl get svc
NAME          STATUS  VOLUME        CAPACITY  ACCESS MODES  STORAGECLASS  AGE
mysql-pv-claim Bound   pvc-83c84f37-7d6e-11ea-931f-12ba6e2b8915  20Gi   RWO          gp2          7m26s
wp-pv-claim  Bound   pvc-a23720ec-7d6e-11ea-bbc4-0a3fd4961b7b  20Gi   RWO          gp2          6m35s
arun@Arun-Parmar:PVC $ kubectl get pvc
NAME          TYPE      CLUSTER-IP      EXTERNAL-IP     PORT(S)        AGE
kubernetes    ClusterIP  10.100.0.1    <none>        443/TCP       21m
my-service    NodePort   10.100.206.168 <none>        80:30487/TCP  6m41s
wordpress-mysql ClusterIP  None         <none>        3306/TCP      7m34s
arun@Arun-Parmar:PVC $ 
```

Hit browser with node ip (External ip) followed by port



- Project 4: Create a bash script which lists all the security group rules and delete all the rules in which public access is allowed except 80 and 443 ports.



```

Activities Terminal Sun 13:12 Terminal
File Edit View Search Terminal Tabs Help Terminal
var=$(aws ec2 describe-security-groups | jq '.SecurityGroups[] | {Id : .GroupId, Port : .IpPermissions[].ToPort, Protocol: .IpPermissions[].IpProtocol, Ipv4: .IpPermissions[].IpRanges[].CidrIp, Ipv6: .IpPermissions[].IpRanges[].CidrIpv6}' | sort -u)
for i in $var
do
    Id=`echo $i | jq -r .Id`;
    port=`echo $i | jq -r .Port`;
    protocol=`echo $i | jq .Protocol`;
    Ipv4=`echo $i | jq -r .Ipv4`;
    Ipv6=`echo $i | jq -r .Ipv6`;

    if [[ $Ipv4 == *"0.0.0.0/0"* || $Ipv6 == *":/0"* ]]
    then
        if [[ $port == 443 && $port == 80 ]]
        then
            echo "Public but of port 80 and 443"
        else
            aws ec2 revoke-security-group-ingress --group-id $Id \
            --ip-permissions '[{"IpProtocol": '$protocol', "FromPort": '$port', "ToPort": '$port', "IpRanges": [{"CidrIpv6": "::/0"}]}]'
            echo "Rules deleted";
        fi
    done
    fi
done
"arun.sh" 22L, 931C 1,45 All

```


Activities Firefox Web Browser ▾

Sat 22:09

EC2 Management Console - Mozilla Firefox

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#SecurityGroup:sg-04c75ba8dc7acd595-testing

67% arun.parmar@bothenew.com @... N. Virginia Support

AWS Services Resource Groups

New EC2 Experience Tell us what you think

Limits Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Scheduled Instances Capacity Reservations

Instances

Owner 1876532318301

Security group name testing Security group ID sg-04c75ba8dc7acd595 Description testing for script VPC ID vpc-d38d68b7

Inbound rules count: 2 Permission entries Outbound rules count: 1 Permission entry

Inbound rules Outbound rules Tags

Inbound rules

Type Protocol Port range Source Description - optional

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
HTTP	TCP	80	-/0	-

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with various service icons and a navigation menu. The main content area displays the details of a security group named 'testing'. It shows the security group ID (sg-04c75ba8dc7acd595), owner (1876532318301), and two inbound rules for port 80. The interface is in dark mode.