### **Day 17**

### **Project 01**

### **Deploy a Database Server with Backup Automation**

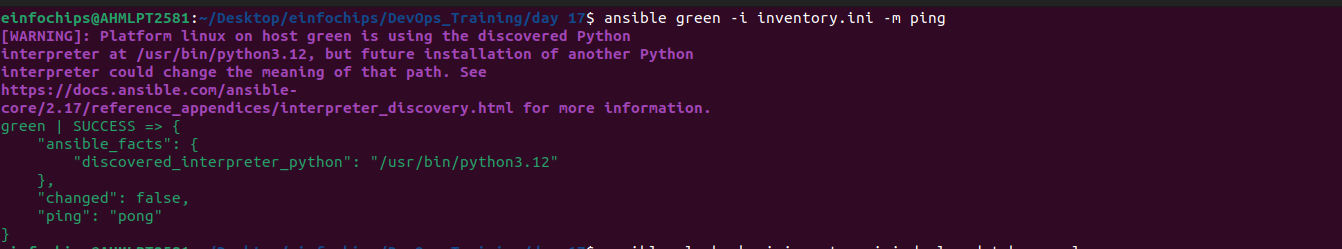
### **Problem Statement**

**Objective**: Automate the deployment, configuration, and backup of a MySQL database server on an Ubuntu instance using Ansible.

**Requirements**:

1. **AWS Ubuntu Instance**: You have an Ubuntu server instance running on AWS.
2. **Database Server Deployment**: Deploy and configure PostgreSQL on the Ubuntu instance.
3. **Database Initialization**: Create a database and a user with specific permissions.
4. **Backup Automation**: Set up a cron job for regular database backups and ensure that backups are stored in a specified directory.
5. **Configuration Management**: Use Ansible to handle the deployment and configuration, including managing sensitive data like database passwords.

### **Deliverables**

1. **Ansible Inventory File**
   * **Filename**: inventory.ini

.

1. **Ansible Playbook**
   * **Filename**: deploy\_database.yml

```

- hosts: web

become: yes

vars:

mysql\_custom\_settings:

max\_connections: 200

query\_cache\_size: 16M

tasks:

- name: update cache as apt-get update

apt:

update\_cache: yes

- name: install mysql and its dependencies

apt:

name: ['mysql-server', 'mysql-client', 'python3-mysqldb', 'libmysqlclient-dev']

state: present

- name: enable the MySQL

service:

name: mysql

state: started

enabled: yes

- name: create a user in the database

mysql\_user:

name: utsav

password: utsav1234

priv: '\*.\*:ALL'

host: '%'

state: present

- name: changing configuration file using template

template:

src: "/home/einfochips/Desktop/einfochips/DevOps\_Training/day 17/templates/sql\_hba.conf.j2"

dest: "/etc/mysql/sql\_hba.conf.j2"

notify:

- restart mysql

- name: copy the script

ansible.builtin.copy:

src: "/home/einfochips/Desktop/einfochips/DevOps\_Training/day 17/scripts/backup.sh"

dest: "/home/ubuntu/backup.sh"

mode: '777'

- name: database backup using cron job

cron:

name: "MySQL Backup"

minute: "\*/30"

job: /home/ubuntu/backup.sh

handlers:

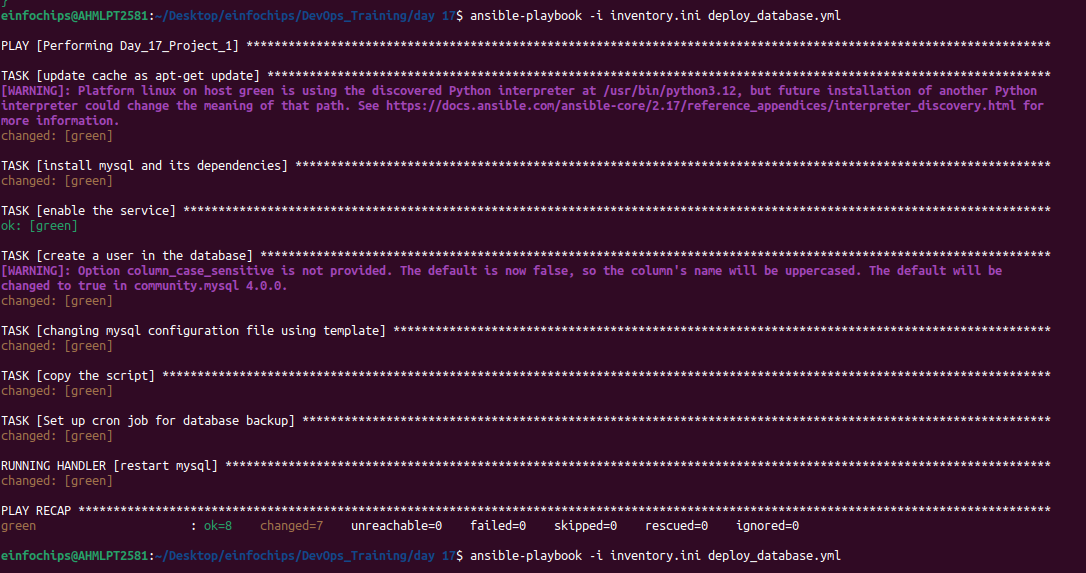
- name: restart mysql

service:

name: mysql

state: restarted

```

.

command in Playbook file to create a directory for MySQL backup file

```

- name: Create backup directory

file:

path: "/var/backups/mysql/"

state: directory

mode: '0755'

```

1. **Jinja2 Template**
   * **Filename**: templates/sql\_hba.conf.j2

```

{% for setting, value in mysql\_custom\_settings.items() %}

{{ setting }} = {{ value }}

{% endfor %}

```

1. **Backup Script**
   * **Filename**: scripts/backup.sh.

```

#!/bin/bash

# Database credentials

USER="utsav"

PASSWORD="utsav1234"

HOST="localhost"

DB\_NAME="my\_database"

# Backup directory

BACKUP\_PATH='/var/backups/mysql/'

DATE=$(date +%F)

# Set default file permissions

umask 177

# Create backup directory if it does not exist

mkdir -p "$BACKUP\_PATH"

# Dump database into SQL file

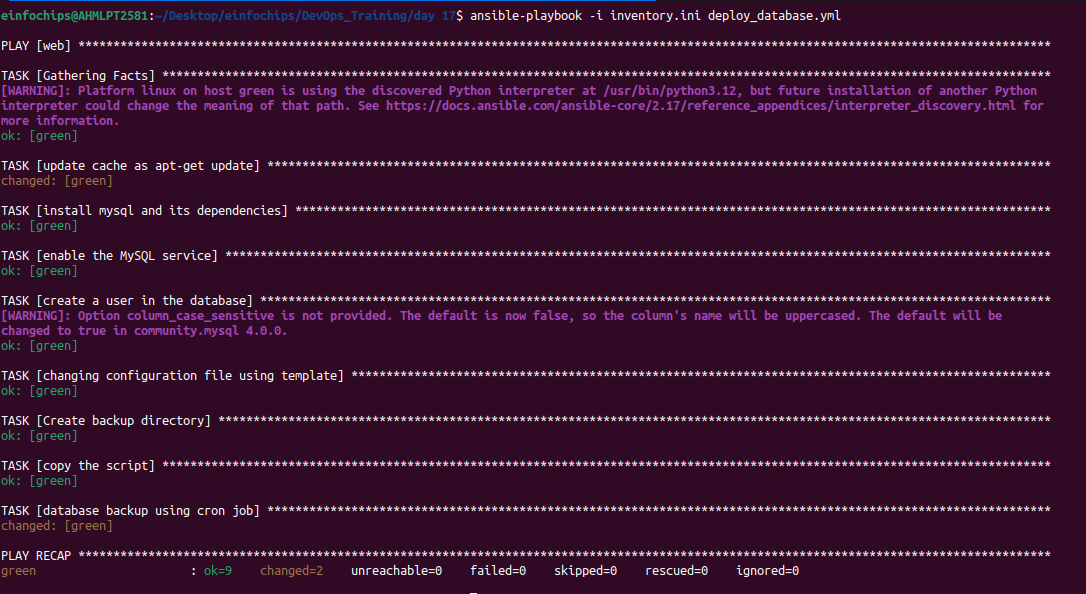
mysqldump --user="$USER" --password="$PASSWORD" --host="$HOST" "$DB\_NAME" > "$BACKUP\_PATH/$DB\_NAME-$DATE.sql"

# Remove backups older than 7 days

find "$BACKUP\_PATH" -type f -name "\*.sql" -mtime +7 -exec rm {} \;

```

Output :

.

File Created:

### 

### **.**

### **Project 02**

**Objective**: Automate the setup of a multi-tier web application stack with separate database and application servers using Ansible.

### **Problem Statement**

**Objective**: Automate the deployment and configuration of a multi-tier web application stack consisting of:

1. **Database Server**: Set up a PostgreSQL database server on one Ubuntu instance.
2. **Application Server**: Set up a web server (e.g., Apache or Nginx) on another Ubuntu instance to host a web application.
3. **Application Deployment**: Ensure the web application is deployed on the application server and is configured to connect to the PostgreSQL database on the database server.
4. **Configuration Management**: Use Ansible to automate the configuration of both servers, including the initialization of the database and the deployment of the web application.

### **Deliverables**

1. **Ansible Inventory File**
   * **Filename**: inventory.ini

```

[db\_server]

green ansible\_host=3.\*\*\*.\*.\*\* ansible\_user=ubuntu ansible\_ssh\_private\_key\_file=/home/einfochips/.ssh/ansible-worker.pem

[app\_server]

green ansible\_host=3.\*\*\*.\*.\*\* ansible\_user=ubuntu ansible\_ssh\_private\_key\_file=/home/einfochips/.ssh/ansible-worker.pem

```

1. **Ansible Playbook**
   * **Filename**: deploy\_multitier\_stack.yml

```

---

- hosts: db\_server

become: yes

vars:

mysql\_root\_password: "utsav1234"

mysql\_database: "webapp\_db"

mysql\_user: "webapp\_user"

mysql\_password: "utsav1234"

tasks:

- name: update cache as apt-get update

apt:

update\_cache: yes

- name: install mysql and its dependencies

apt:

name: ['mysql-server', 'mysql-client', 'python3-mysqldb', 'libmysqlclient-dev']

state: present

- name: enable the service

service:

name: mysql

state: started

enabled: yes

- name: Create MySQL database

mysql\_db:

name: "{{ mysql\_database }}"

state: present

- name: Create MySQL user

mysql\_user:

name: "{{ mysql\_user }}"

password: "{{ mysql\_password }}"

priv: "{{ mysql\_database }}.\*:ALL"

state: present

- hosts: app\_server

become: yes

vars:

app\_repo\_url: "/home/einfochips/Desktop/einfochips/DevOps\_Training/day 17/Task2/files/index.html"

app\_directory: "/var/www/html/"

db\_host: "{{ hostvars['db\_server']['ansible\_host'] }}"

db\_name: "webapp\_db"

db\_user: "webapp\_user"

db\_password: "secure\_user\_password"

tasks:

- name: Update apt cache

apt:

update\_cache: yes

- name: Install Apache2

apt:

name: apache2

state: present

- name: Install PHP and extensions

apt:

name:

- php

- php-mysql

- libapache2-mod-php

state: present

- name: Ensure Apache is started and enabled

service:

name: apache2

state: started

enabled: yes

- name: Copy Web Application

ansible.builtin.copy:

src: "{{ app\_repo\_url }}"

dest: "{{ app\_directory }}"

- name: Deploy Apache configuration file

template:

src: "/home/einfochips/Desktop/einfochips/DevOps\_Training/day 17/Task2/templates/apache\_server.conf.j2"

dest: "/etc/apache2/apache\_server.conf"

owner: root

group: root

mode: 0644

notify:

- Restart Apache

- name: Ensure appropriate permissions for application directory

file:

path: "{{ app\_directory }}"

owner: www-data

group: www-data

state: directory

recurse: yes

handlers:

- name: Restart Apache

service:

name: apache2

state: restarted

```

1. **Jinja2 Template**
   * **Filename**: templates/app\_config.php.j2
2. **Application Files**
   * **Filename**: files/index.html (or equivalent application files)

```

<!DOCTYPE html>

<html>

<head>

<title>My Web Application</title>

</head>

<body>

<center>

<h1>Project 2 Web App</h1>

<p>Day 17 Task 2</p>

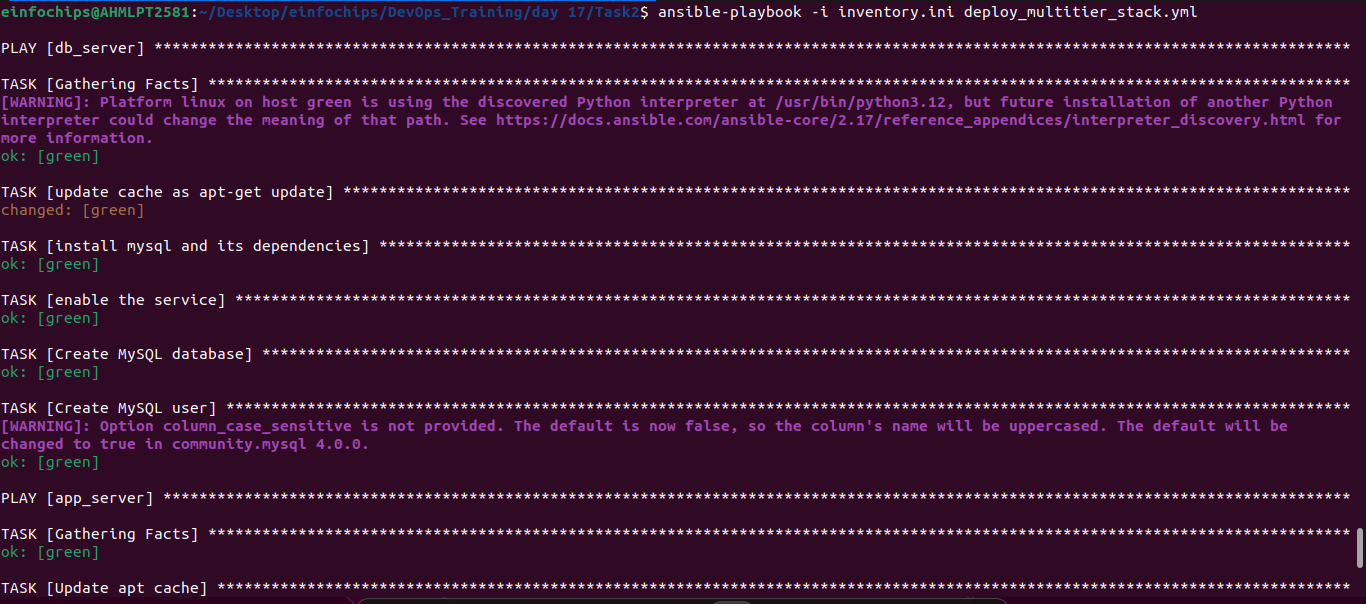
</center>

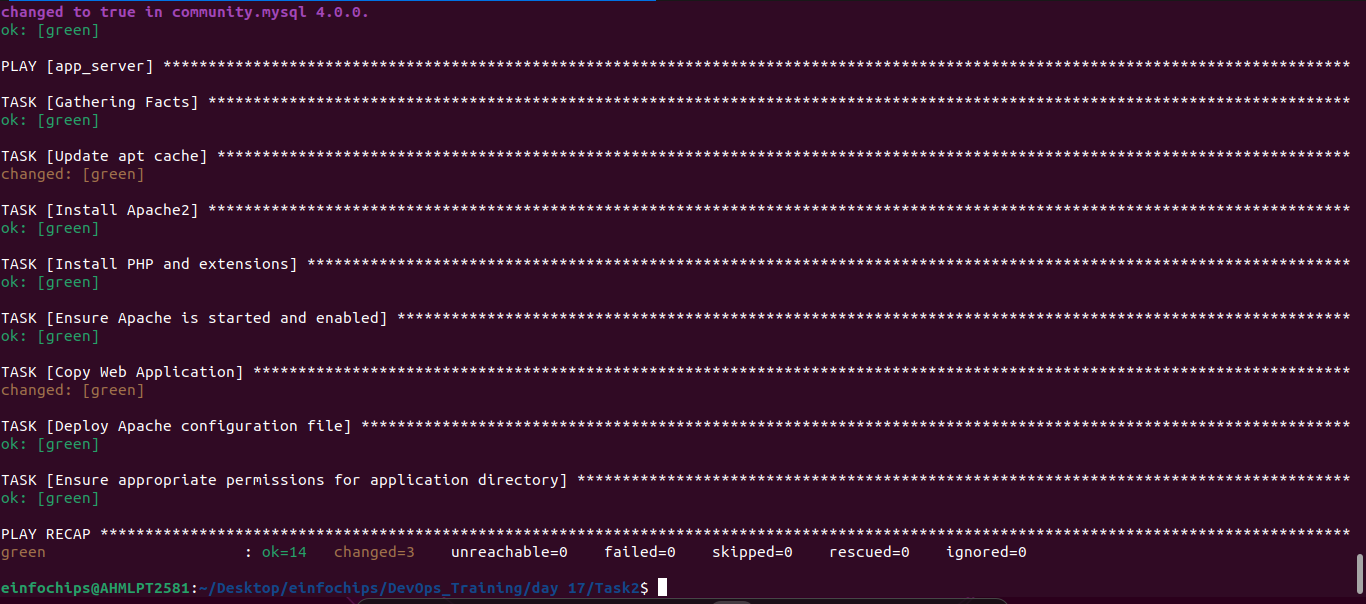
</body>

</html>

```

Output:

.

.

