

Project 01

Deploy a Database Server with Backup Automation

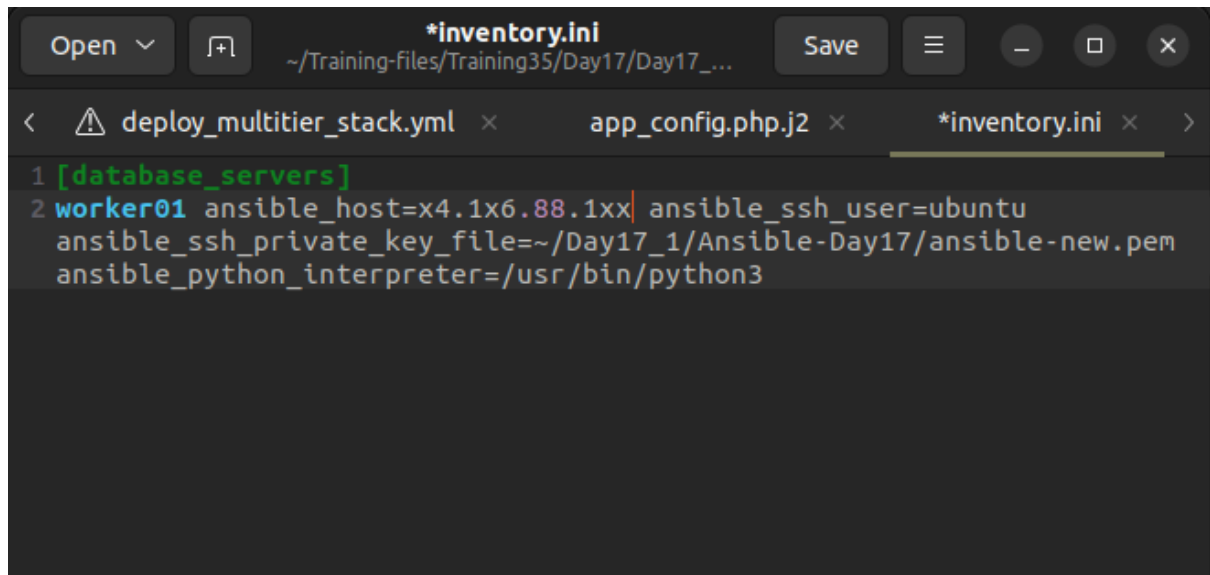
Objective: Automate the deployment and configuration of a PostgreSQL database server on an Ubuntu instance hosted on AWS, and set up regular backups.

Problem Statement

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

1. Ansible Inventory File

- **Filename:** `inventory.ini`
- **Content:** Defines the AWS Ubuntu instance and connection details for Ansible.

A screenshot of a code editor window. The title bar shows the file name as `*inventory.ini` and the path as `~/Training-files/Training35/Day17/Day17_...`. The editor has tabs for `deploy_multitier_stack.yml`, `app_config.php.j2`, and `*inventory.ini`. The `*inventory.ini` tab is active, showing the following content:

```
1 [database_servers]
2 worker01 ansible_host=x4.1x6.88.1xx| ansible_ssh_user=ubuntu
   ansible_ssh_private_key_file=~/.Day17_1/Ansible-Day17/ansible-new.pem
   ansible_python_interpreter=/usr/bin/python3
```

2. Ansible Playbook

- **Filename:** `deploy_database.yml`
- **Content:** Automates the installation of MySQL, sets up the database, creates a user, and configures a cron job for backups. It also includes variables for database configuration and backup settings.

deploy_multitier_stack.yml

```
1 - name: Deploy and configure MySQL database
2   hosts: db_server
3   become: yes
4   vars:
5     db_name: "my_database"
6     db_user: "my_user"
7     db_password: "user123"
8
9   tasks:
10  - name: Install MySQL server
11    apt:
12      update_cache: yes
13      name: "{{ item }}"
14      state: present
15    with_items:
16      - mysql-server
17      - mysql-client
18      - python3-mysqldb
19      - libmysqlclient-dev
20
21  - name: Ensure MySQL service is running
22    service:
23      name: mysql
24      state: started
25      enabled: yes
26
27  - name: Create MySQL user
28    mysql_user:
29      name: "{{ db_user }}"
30      password: "{{ db_password }}"
31      priv: '*.*:ALL'
32      host: '%'
33      state: present
34
35  - name: Create MySQL database
```

deploy_multitier_stack.yml

```
32     host: '%'
33     state: present
34
35 - name: Create MySQL database
36   mysql_db:
37     name: "{{ db_name }}"
38     state: present
39
40 - name: Deploy and configure web server and application
41   hosts: web_server
42   become: yes
43
44   vars:
45     db_host: "host_ip"
46     db_name: "my_database"
47     db_user: "my_user"
48     db_password: "user123"
49
50   tasks:
51     - name: Install web server
52       apt:
53         name: nginx
54         state: present
55         update_cache: yes
56
57     - name: Ensure web server is running
58       service:
59         name: nginx
60         state: started
61         enabled: yes
62
63     - name: Deploy application files
64       copy:
65         src: files/index.html
66         dest: /var/www/html/index.html
```

```
68 - name: Configure application
69   template:
70     src: templates/app_config.php.j2
71     dest: /var/www/html/app_config.php
72
73 - name: Restart web server to apply changes
74   service:
75     name: nginx
76     state: restarted
```

3. Jinja2 Template

- **Filename:** `templates/mysql.cnf.j2`
- **Content:** Defines the MySQL configuration file (`mysql.conf`) using Jinja2 templates to manage access controls dynamically.

```
mysql.cnf.j2
1 # Here is entries for some specific programs
2 # The following values assume you have at least 32M ram
3
4 !includedir /etc/mysql/conf.d/
5 !includedir /etc/mysql/mysql.conf.d/
6
```

4. Backup Script

- **Filename:** `scripts/backup.sh`
- **Content:** A script to perform the backup of the MySQL database. This script should be referenced in the cron job defined in the playbook.

```
backup.sh
1 #!/bin/bash
2
3 DB_NAME=$1
4 DB_USER=$2
5 DB_PASSWORD=$3
6 BACKUP_DIR=$4
7 DATE=$(date +%Y%m%d)
8 BACKUP_FILE="${BACKUP_DIR}/${DB_NAME}_${DATE}.sql.gz"
9
10 # Create backup
11 mysqldump -u $DB_USER -p$DB_PASSWORD $DB_NAME | gzip > $BACKUP_FILE
12
13 # Optional: Remove backups older than 7 days
14 find $BACKUP_DIR -type f -name "*.gz" -mtime +7 -exec rm {} \;
15
```

```

einfochips@AHMLPT2484:~/Day17_1/Ansible-Day17$ nano inventory.ini
einfochips@AHMLPT2484:~/Day17_1/Ansible-Day17$ nano deploy_database.yml
einfochips@AHMLPT2484:~/Day17_1/Ansible-Day17$ ansible-playbook -i inventory.ini deploy_database.yml

PLAY [Deploy and configure MySQL database server] *****

TASK [Gathering Facts] *****
ok: [worker01]

TASK [Install MySQL] *****
ok: [worker01] => (item=mysql-server)
ok: [worker01] => (item=mysql-client)
ok: [worker01] => (item=python3-mysqldb)
ok: [worker01] => (item=libmysqlclient-dev)

TASK [Copy MySQL configuration file] *****
ok: [worker01]

TASK [Ensure MySQL service is running] *****
ok: [worker01]

TASK [Create MySQL database] *****
ok: [worker01]

TASK [Create MySQL user and grant privileges] *****
[WARNING]: Option column_case_sensitive is not provided. The default is now false, so the column's name will be uppercased. The default will
be changed to true in community.mysql 4.0.0.
ok: [worker01]

TASK [Create backup directory] *****
ok: [worker01]

TASK [Copy MySQL backup script] *****
ok: [worker01]

TASK [Set up cron job for daily backups] *****
ok: [worker01]

PLAY RECAP *****
worker01                : ok=9    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

einfochips@AHMLPT2484:~/Day17_1/Ansible-Day17$

```