Bharatiya Vidya Bhavans'

Sardar Patel Institute of Technology

Munshinagar, Andheri(W), Mumbai-400058

(Autonomous College Affiliated to University of Mumbai)

Academic Year: 2025 26 **Semester: III** **Class: MCA**

Course Code: MC520 **Course Name: Cloud Computing**

Experiment No.3

Date: 24.09.25

Aim: Ubuntu: Development of Automation using Ansible

CO Mapping – OECS1.4

Objective:

To learn and implement IT automation in Ubuntu using Ansible for configuration management, application deployment, and task orchestration, enabling efficient, repeatable, and scalable system administration.

Concept:

Ansible is an open-source IT automation tool used for configuration management, application deployment, and task automation.

- It uses a declarative YAML-based language (Playbooks) to define tasks.
- Operates agentlessly — communicates with target systems via SSH, so no special software is needed on clients.
- Ensures consistency, scalability, and repeatability of system configurations.

Lab Exercise:

1) **Create a Spring Boot Project**
   - Develop a normal Spring Boot application (for API).
   - Build the project using Maven/Gradle to generate the JAR file.

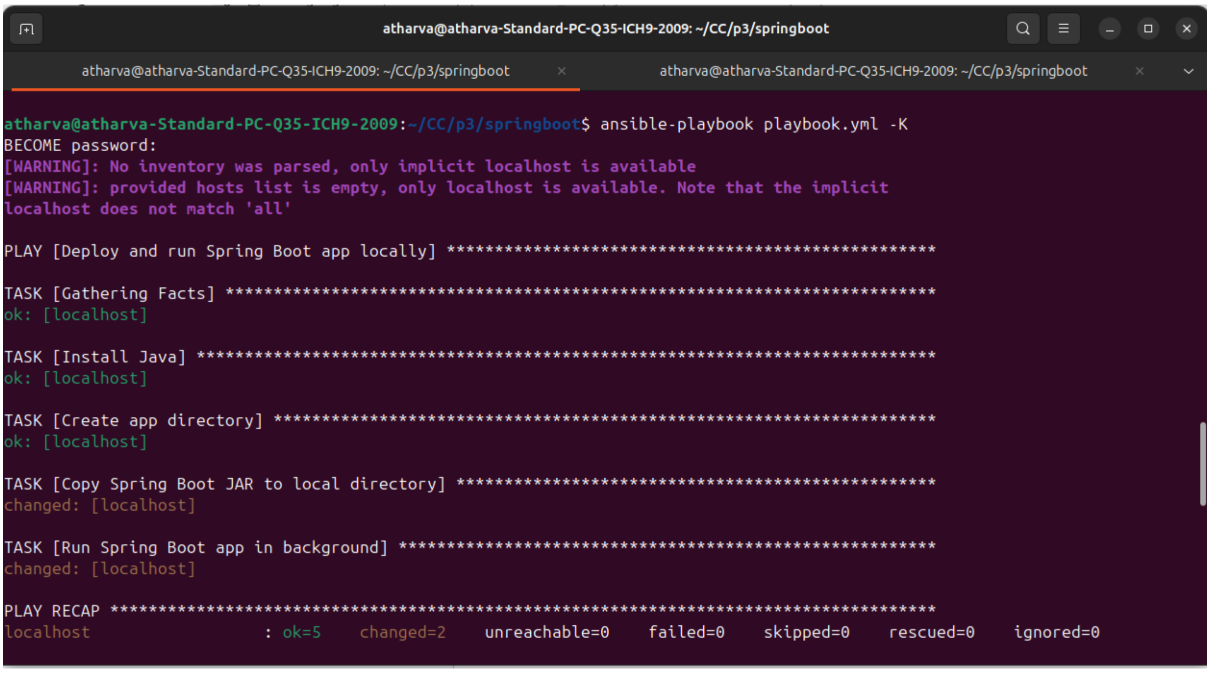
2) **Create Inventory File (inventory.ini)**
   Add the ip address of your machine in **inventory.ini.** you can get it using: - ip addr

3) **Create Ansible Playbook (playbook.yml)**

```
---
- name: Deploy and run Spring Boot app locally
  hosts: localhost
  connection: local
```

```
become: yes
vars:
  app_name: project-0.0.1-SNAPSHOT.jar
  app_dir: /opt/springboot
  java_package: openjdk-17-jdk
  app_user: atharva
  app_group: atharva

tasks:
  - name: Install Java
    apt:
      name: "{{ java_package }}"
      state: present
      update_cache: yes

  - name: Create app directory
    file:
      path: "{{ app_dir }}"
      state: directory
      owner: "{{ app_user }}"
      group: "{{ app_group }}"
      mode: '0755'

  - name: Copy Spring Boot JAR to local directory
    copy:
      src: "./project/target/{{ app_name }}"
      dest: "{{ app_dir }}/{{ app_name }}"
      owner: "{{ app_user }}"
      group: "{{ app_group }}"
      mode: '0755'

  - name: Run Spring Boot app in background
    shell: "nohup java -jar {{ app_dir }}/{{ app_name }} > /dev/null 2>&1 &"
    args:
      chdir: "{{ app_dir }}"
    async: 0
    poll: 0
```

4) **Run the Playbook**
   Execute the following command in the terminal:
   ansible-playbook playbook.yml -k

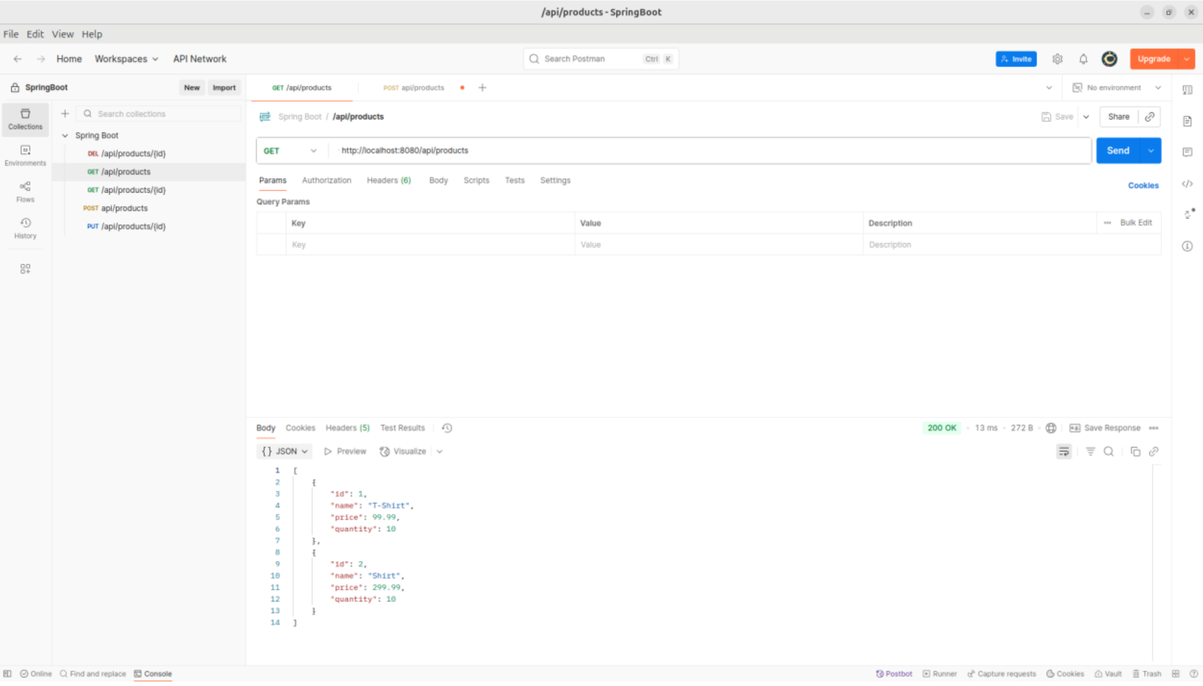

atharva@atharva-Standard-PC-Q35-ICH9-2009: ~/CC/p3/springboot
atharva@atharva-Standard-PC-Q35-ICH9-2009: ~/CC/p3/springboot
atharva@atharva-Standard-PC-Q35-ICH9-2009: ~/CC/p3/springboot

```
atharva@atharva-Standard-PC-Q35-ICH9-2009:~/CC/p3/springboot$ ansible-playbook playbook.yml -K
BECOME password:
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit
localhost does not match 'all'

PLAY [Deploy and run Spring Boot app locally] ***************************************************

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

TASK [Install Java] *****************************************************************************
ok: [localhost]

TASK [Create app directory] *********************************************************************
ok: [localhost]

TASK [Copy Spring Boot JAR to local directory] **************************************************
changed: [localhost]

TASK [Run Spring Boot app in background] ********************************************************
changed: [localhost]

PLAY RECAP **************************************************************************************
localhost                  : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

5) Verify Deployment

Once the playbook runs successfully, check the Spring Boot API on port 8080 (default port specified in the project).

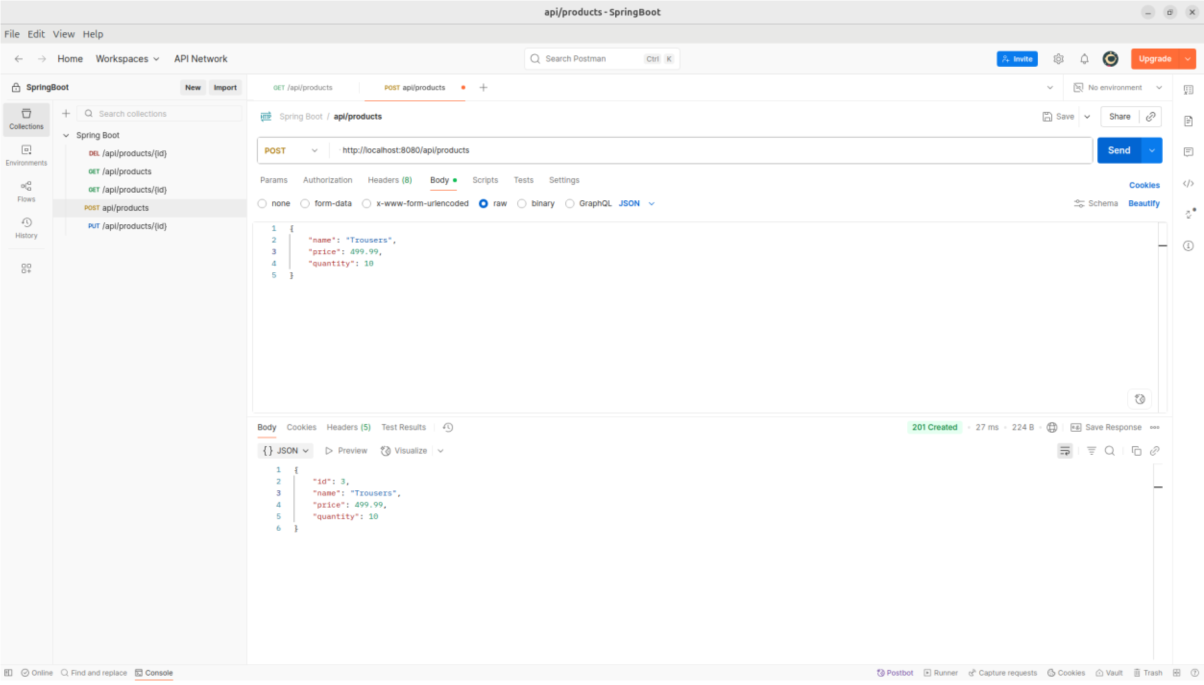
You can use Postman to verify you api.



GET http://localhost:8080/api/products

200 OK · 13 ms · 272 B

```
[
    {
        "id": 1,
        "name": "T-Shirt",
        "price": 99.99,
        "quantity": 10
    },
    {
        "id": 2,
        "name": "Shirt",
        "price": 299.99,
        "quantity": 10
    }
]
```

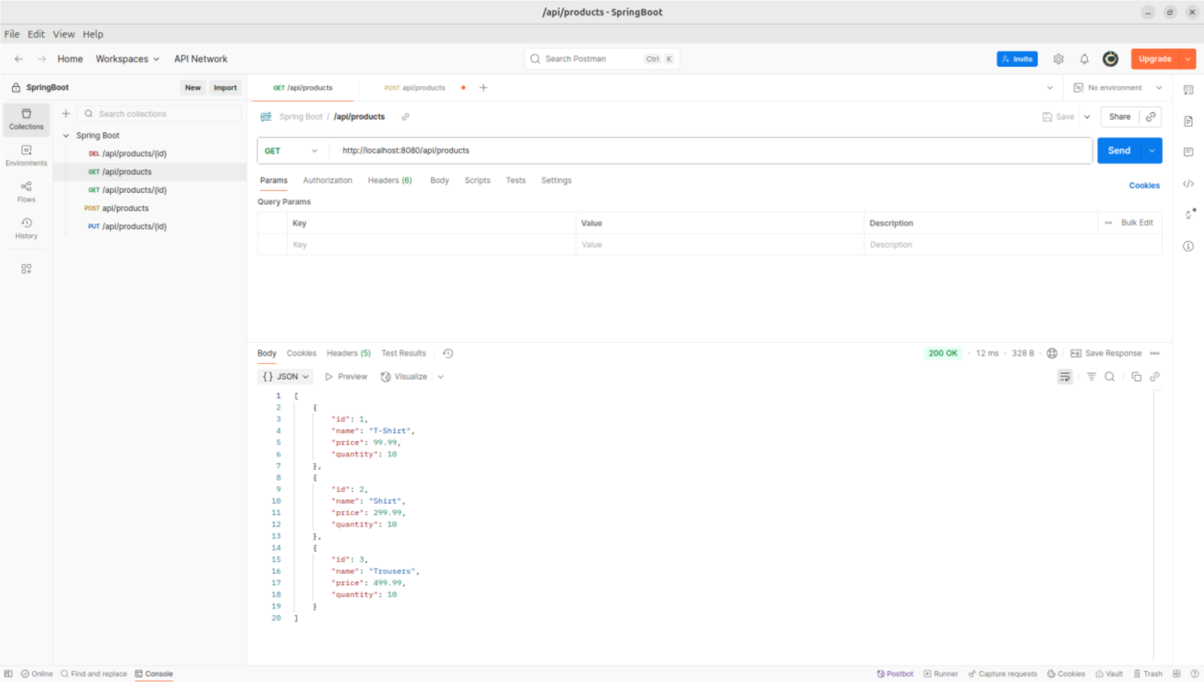

api/products - SpringBoot
File Edit View Help
Home Workspaces API Network
Search Postman Ctrl K
Invite
Upgrade
SpringBoot New Import
Collections
Environments
Flows
History
Search collections
Spring Boot
DEL /api/products/{id}
GET /api/products
GET /api/products/{id}
POST api/products
PUT /api/products/{id}
GET /api/products
POST api/products
No environment
Spring Boot / api/products
Save
Share
Send
Params Authorization Headers (8) Body Scripts Tests Settings
Cookies
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON
Schema Beautify
Body Cookies Headers (5) Test Results
Save Response
{} JSON Preview Visualize
Online Find and replace Console
Postbot Runner Capture requests Cookies Vault Trash

POST http://localhost:8080/api/products

```
{
    "name": "Trousers",
    "price": 499.99,
    "quantity": 10
}
```

201 Created · 27 ms · 224 B

```
{
    "id": 3,
    "name": "Trousers",
    "price": 499.99,
    "quantity": 10
}
```



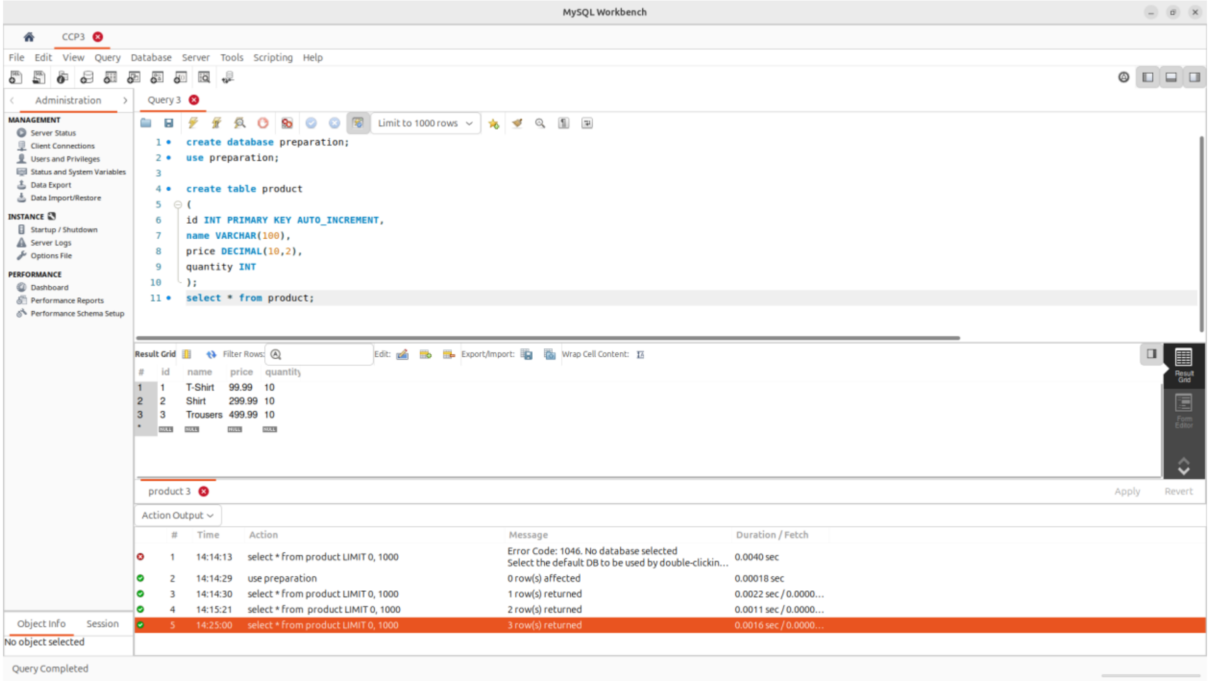
GET http://localhost:8080/api/products

Query Params

Key	Value	Description
Key	Value	Description

200 OK · 12 ms · 328 B

```
[
    {
        "id": 1,
        "name": "T-Shirt",
        "price": 99.99,
        "quantity": 10
    },
    {
        "id": 2,
        "name": "Shirt",
        "price": 299.99,
        "quantity": 10
    },
    {
        "id": 3,
        "name": "Trousers",
        "price": 499.99,
        "quantity": 10
    }
]
```



```sql
create database preparation;
use preparation;

create table product
(
id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(100),
price DECIMAL(10,2),
quantity INT
);
select * from product;
```

#	id	name	price	quantity
1	1	T-Shirt	99.99	10
2	2	Shirt	299.99	10
3	3	Trousers	499.99	10
*	NULL	NULL	NULL	NULL

	#	Time	Action	Message	Duration / Fetch
☒	1	14:14:13	select * from product LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clickin...	0.0040 sec
✓	2	14:14:29	use preparation	0 row(s) affected	0.00018 sec
✓	3	14:14:30	select * from product LIMIT 0, 1000	1 row(s) returned	0.0022 sec / 0.0000...
✓	4	14:15:21	select * from product LIMIT 0, 1000	2 row(s) returned	0.0011 sec / 0.0000...
✓	5	14:25:00	select * from product LIMIT 0, 1000	3 row(s) returned	0.0016 sec / 0.0000...

Observation:

The Ansible playbook executed successfully and automated the installation of Java, creation of the app directory, and deployment of the Spring Boot JAR.

The Spring Boot application started in the background and the API was accessible on port 8080.