Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Computer Science and Engineering (Data Science)

AY: 2025-26

Semester: V

Subject: DevOps Laboratory (DJS23OLOE501)

Experiment 3b

(Configuration Management with Puppet and Ansible)

Name: Soham Kishor Walam **Roll No: D102**

Aim: To Perform Configuration Management using Puppet

Theory:

Introduction to Configuration Management

Configuration Management (CM) is a critical practice in DevOps that automates the process of managing and maintaining software and infrastructure configurations across servers. It ensures that systems are deployed consistently, reducing human error, and allowing for version control, auditing, and repeatability.

Two popular CM tools in this domain are **Puppet** and **Ansible**. Both tools automate configuration tasks, but they differ in architecture and approach.

What is Puppet?

Puppet is an open-source configuration management tool that uses a declarative language to describe system configurations. It follows a client-server model where:

The Puppet master holds the configuration data (manifests and modules).

The Puppet agent runs on the nodes (servers) and communicates with the master to enforce configurations.

Key Concepts:

- Manifest: A file written in Puppet DSL (.pp) describing desired configurations (e.g., install Apache).
- Module: A reusable collection of manifests and data (e.g., LAMP stack setup).
- Resource: The fundamental unit (e.g., package, file, service) that Puppet manages.

Understanding Puppet Architecture

Puppet's architecture provides a complete insight into its operation. Here are the key components of the Puppet primary server environment, as well as their respective functions.

- Manifests – These are the codes used for client configuration
- Templates – Help to combine code and data to create a final document
- Files – These are static content that clients can download.

- Modules – These are important to Puppet architecture since they include manifests, files, and templates.
- Certificate authority– These facilitate the master's signature of various certificates issued by the client.

1. **Install Puppet 7 on a Linux Server**

Step 1: Create a VM Instance in GCP

- Go to **Google Cloud Console → Compute Engine → VM Instances**
- Click **Create Instance**
- Set:
    - Name: puppet-vm
    - Machine type: e2-medium (2 vCPU, 4 GB RAM)
    - OS Image: Ubuntu 20.04 LTS
- Under **Firewall**, check Allow HTTP and Allow HTTPS
- Click **Create**

Step 2: Connect to the VM

- Click **SSH** from the GCP Console to open the VM terminal.

Step 3: Update the system packages

```
sudo apt update
sudo apt upgrade -y
```

Step 4: Add Puppet 7 APT repository

```
wget https://apt.puppet.com/puppet7-release-focal.deb
sudo dpkg -i puppet7-release-focal.deb
sudo apt update
```

Step 5: Install Puppet Agent

```
sudo apt install -y puppet-agent
```

Step 6: Verify Puppet installation

```
ls /opt/puppetlabs/bin/
```

Step 7: Add Puppet to PATH

```
echo 'export PATH=/opt/puppetlabs/bin:$PATH' >> ~/.bashrc
source ~/.bashrc
```

Step 8: Check Puppet version

```
which puppet
puppet --version
```

2. Basic Puppet Manifest to Install a Web Server

A Puppet manifest (.pp file) describes the desired state of the system using Puppet's declarative language. Resources like package, service, and file are used to manage system components.

Step 1: Install nano (if not installed)

```
sudo apt install nano -y
```

Step 2: Create a Puppet manifest file

```
sudo nano ~/webserver.pp
```

Step 3: Write the manifest

```
# Install Apache
package { 'apache2':
  ensure => installed,
}

# Ensure Apache service is running
service { 'apache2':
  ensure => running,
  enable => true,
  require => Package['apache2'],
}

# Create a test HTML page
file { '/var/www/html/index.html':
  ensure  => file,
  content => "<h1>Hello from Puppet Web Server</h1>",
  require => Package['apache2'],
}
```

Explanation:

- package {} installs Apache2
- service {} ensures Apache runs and is enabled at boot
- file {} creates an index.html page with sample text

Step 4: Apply the manifest

```
sudo /opt/puppetlabs/bin/puppet apply ~/webserver.pp
```

Step 5: Test Apache web server

Open your VM's External IP in a browser:
http://<VM-External-IP>

You should see:
Hello from Puppet Web Server

3. Deploy a LAMP Stack using Puppet

LAMP stack is widely used for hosting dynamic web applications. Puppet can automate its setup by managing packages, services, and configuration files.
To deploy a complete LAMP stack (Linux, Apache, MySQL, PHP) using a single Puppet manifest.

Step 1: Create the manifest

```
sudo nano ~/lamp_stack.pp
```

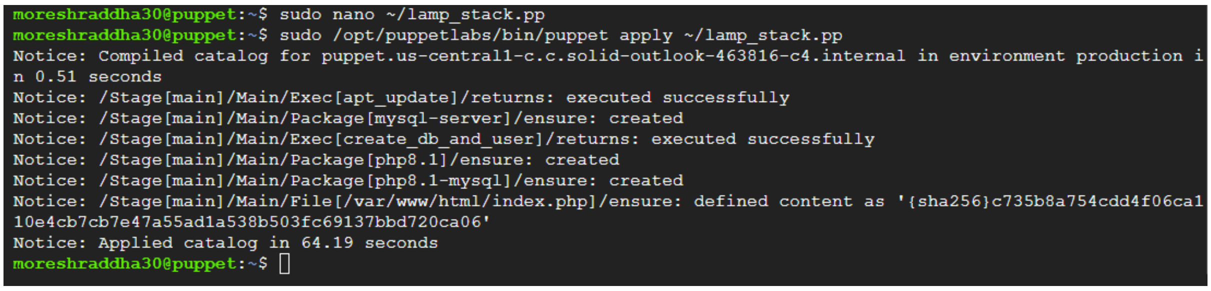
Step 2: Write the manifest

```
# Update package index
exec { 'apt_update':
  command => '/usr/bin/apt update',
  path    => ['/usr/bin', '/usr/sbin'],
}

# Install Apache
package { 'apache2':
  ensure  => installed,
  require => Exec['apt_update'],
}

# Ensure Apache service is running
service { 'apache2':
  ensure    => running,
  enable    => true,
  require   => Package['apache2'],
}

# Install MySQL server
package { 'mysql-server':
  ensure  => installed,
  require => Exec['apt_update'],
}

# Ensure MySQL service is running
service { 'mysql':
  ensure    => running,
  enable    => true,
  require   => Package['mysql-server'],
}

# Set MySQL root password
exec { 'mysql_secure_installation':
  command => "mysql -e \"ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'StrongRootPass'; FLUSH PRIVILEGES;\"",
  path    => ['/usr/bin', '/usr/sbin'],
  unless  => "mysql -uroot -pStrongRootPass -e 'SELECT 1;'",
  require => Service['mysql'],
}

# Create database and user
exec { 'create_db_and_user':
  command => "mysql -uroot -pStrongRootPass -e \"CREATE DATABASE IF NOT EXISTS mydatabase; CREATE USER IF NOT EXISTS 'dbuser'@'localhost' IDENTIFIED BY 'StrongUserPass'; GRANT ALL PRIVILEGES ON mydatabase.* TO 'dbuser'@'localhost'; FLUSH PRIVILEGES;\"",
  path    => ['/usr/bin', '/usr/sbin'],
  require => Exec['mysql_secure_installation'],
}

# Install PHP 8.1 and modules
package { ['php8.1', 'php8.1-mysql', 'libapache2-mod-php8.1']:
  ensure  => installed,
  require => Exec['apt_update'],
}

# Enable PHP in Apache
exec { 'enable_php_module':
  command => '/usr/sbin/a2enmod php8.1',
  path    => ['/usr/bin', '/usr/sbin'],
  unless  => '/bin/ls /etc/apache2/mods-enabled | /bin/grep php8.1.load',
  require => Package['libapache2-mod-php8.1'],
  notify  => Service['apache2'],
}

# Create a PHP test page
file { '/var/www/html/index.php':
  ensure  => file,
  content => "<?php echo '<h1>Hello from Module-Free Puppet LAMP Stack</h1>'; ?>",
  require => [Package['php8.1'], Service['apache2']],
}
```

Explanation:

- exec {} updates APT and sets MySQL root password
- package {} installs Apache, MySQL, PHP
- service {} ensures Apache/MySQL run continuously
- file {} creates a test PHP page

Step 3: Apply the manifest
sudo /opt/puppetlabs/bin/puppet apply ~/lamp_stack.pp

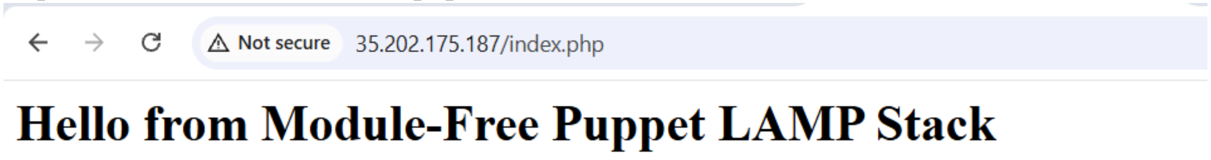


```
moreshraddha30@puppet:~$ sudo nano ~/lamp_stack.pp
moreshraddha30@puppet:~$ sudo /opt/puppetlabs/bin/puppet apply ~/lamp_stack.pp
Notice: Compiled catalog for puppet.us-central1-c.c.solid-outlook-463816-c4.internal in environment production i
n 0.51 seconds
Notice: /Stage[main]/Main/Exec[apt_update]/returns: executed successfully
Notice: /Stage[main]/Main/Package[mysql-server]/ensure: created
Notice: /Stage[main]/Main/Exec[create_db_and_user]/returns: executed successfully
Notice: /Stage[main]/Main/Package[php8.1]/ensure: created
Notice: /Stage[main]/Main/Package[php8.1-mysql]/ensure: created
Notice: /Stage[main]/Main/File[/var/www/html/index.php]/ensure: defined content as '{sha256}c735b8a754cdd4f06ca1
10e4cb7cb7e47a55ad1a538b503fc69137bbd720ca06'
Notice: Applied catalog in 64.19 seconds
moreshraddha30@puppet:~$
```

Step 4: Test LAMP Stack

1. Open your VM's external IP in a browser:

http://<VM-External-IP>/index.php



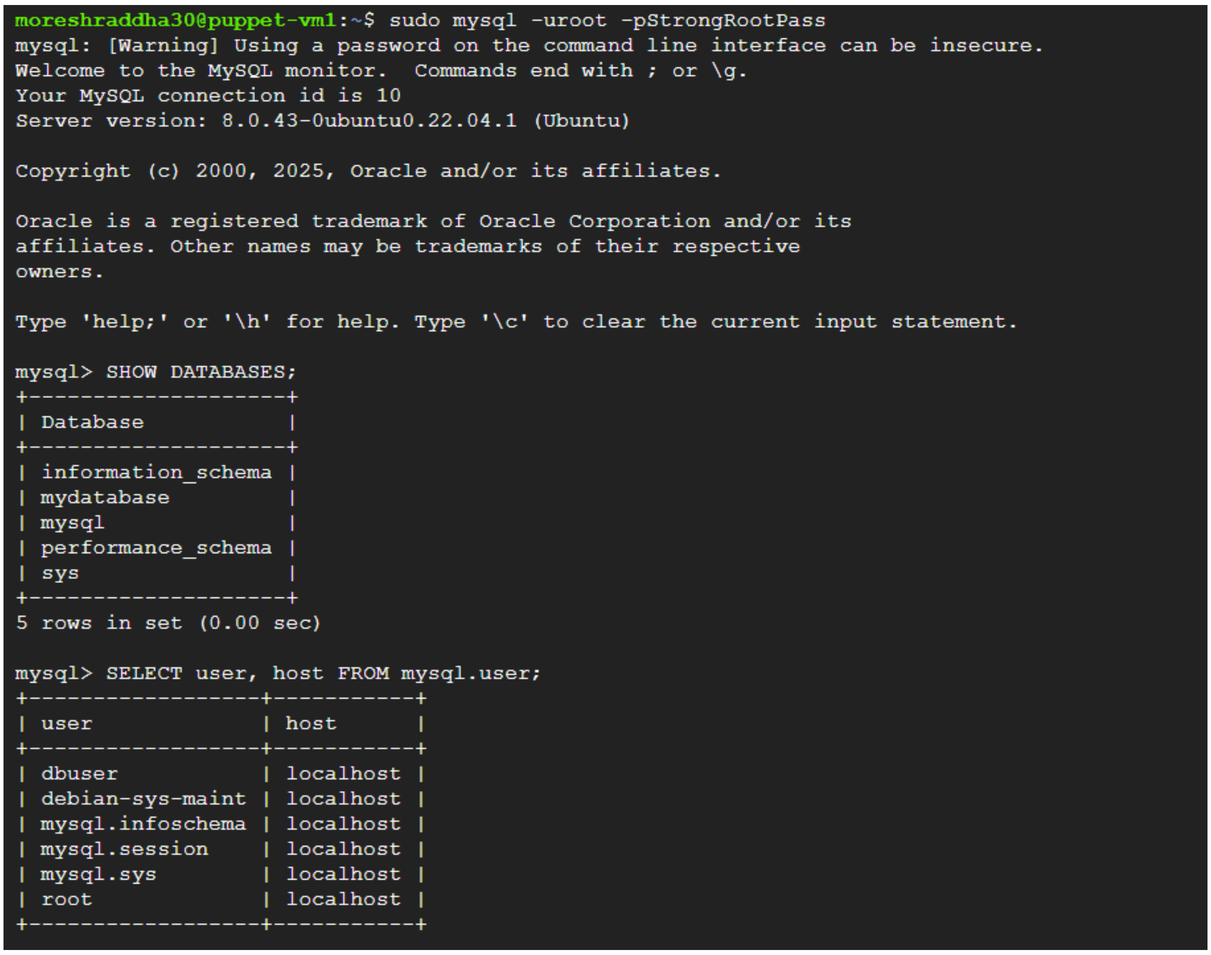
Hello from Module-Free Puppet LAMP Stack

2. Verify MySQL:

sudo mysql -uroot -pStrongRootPass

SHOW DATABASES;

SELECT user, host FROM mysql.user;

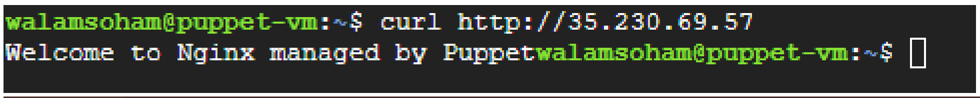


```
moreshraddha30@puppet-vm1:~$ sudo mysql -uroot -pStrongRootPass
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.43-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mydatabase         |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> SELECT user, host FROM mysql.user;
+------------------+-----------+
| user             | host      |
+------------------+-----------+
| dbuser           | localhost |
| debian-sys-maint | localhost |
| mysql.infoschema | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| root             | localhost |
+------------------+-----------+
```

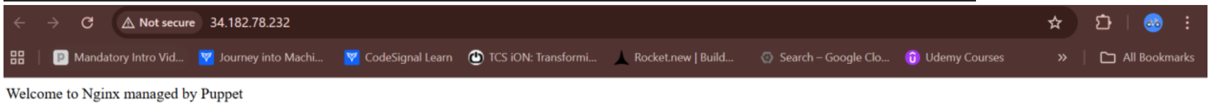
Lab experiment to be performed in this session:

1. Write a Puppet manifest nginx_server.pp to:

1. Install Nginx.
2. Ensure the Nginx service is enabled and running.
3. Create a custom HTML page at /var/www/html/index.html with the text:
   Welcome to Nginx managed by Puppet
4. Apply the manifest.
5. Access the server's external IP in a browser and show the webpage as proof.



```
walamsoham@puppet-vm:~$ curl http://35.230.69.57
Welcome to Nginx managed by Puppetwalamsoham@puppet-vm:~$ 
```



Welcome to Nginx managed by Puppet

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

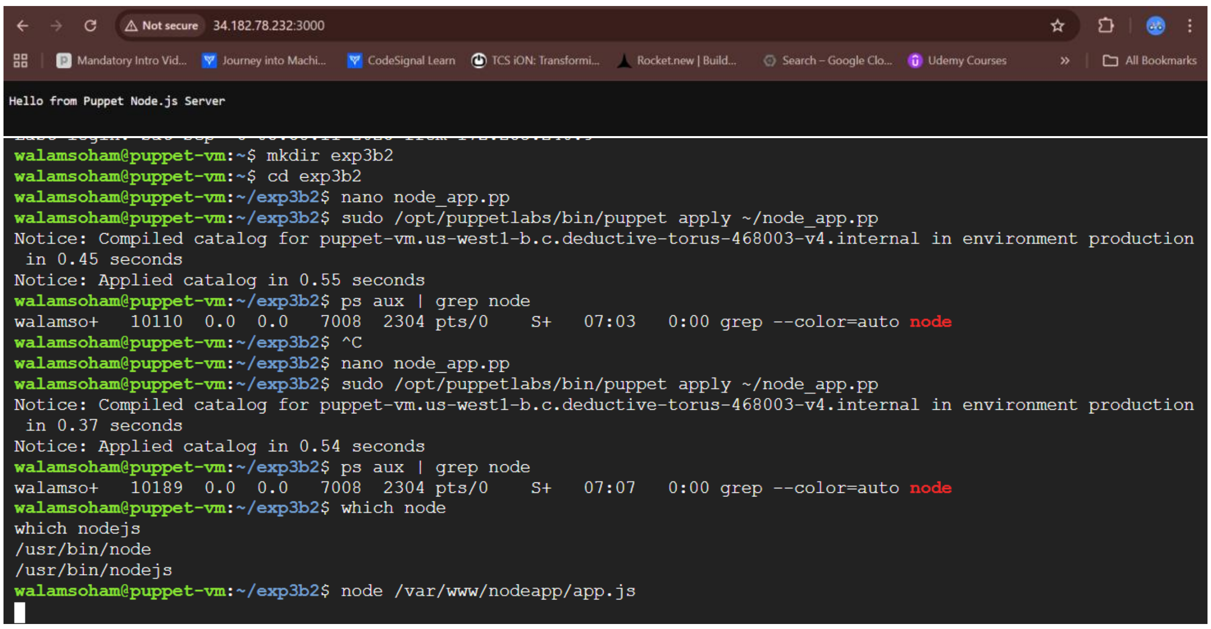
Department of Computer Science and Engineering (Data Science)

2. Write a Puppet Manifest for Node.js Application Setup

- Write a Puppet manifest node_app.pp that:
  1. Installs **Node.js** and **npm**.
  2. Creates a directory /var/www/nodeapp.
  3. Creates a file /var/www/nodeapp/app.js with the following content:

```
const http = require('http');
const server = http.createServer((req, res) => {
  res.end('Hello from Puppet Node.js Server');
});
server.listen(3000);
```

  4. Runs the Node.js app as a background process using nohup.
  5. Verify by accessing:
     http://<VM-IP>:3000
     The output should display:
     Hello from Puppet Node.js Server



```
Hello from Puppet Node.js Server
```

```
walamsoham@puppet-vm:~$ mkdir exp3b2
walamsoham@puppet-vm:~$ cd exp3b2
walamsoham@puppet-vm:~/exp3b2$ nano node_app.pp
walamsoham@puppet-vm:~/exp3b2$ sudo /opt/puppetlabs/bin/puppet apply ~/node_app.pp
Notice: Compiled catalog for puppet-vm.us-west1-b.c.deductive-torus-468003-v4.internal in environment production
 in 0.45 seconds
Notice: Applied catalog in 0.55 seconds
walamsoham@puppet-vm:~/exp3b2$ ps aux | grep node
walamso+   10110  0.0  0.0   7008  2304 pts/0    S+   07:03   0:00 grep --color=auto node
walamsoham@puppet-vm:~/exp3b2$ ^C
walamsoham@puppet-vm:~/exp3b2$ nano node_app.pp
walamsoham@puppet-vm:~/exp3b2$ sudo /opt/puppetlabs/bin/puppet apply ~/node_app.pp
Notice: Compiled catalog for puppet-vm.us-west1-b.c.deductive-torus-468003-v4.internal in environment production
 in 0.37 seconds
Notice: Applied catalog in 0.54 seconds
walamsoham@puppet-vm:~/exp3b2$ ps aux | grep node
walamso+   10189  0.0  0.0   7008  2304 pts/0    S+   07:07   0:00 grep --color=auto node
walamsoham@puppet-vm:~/exp3b2$ which node
which nodejs
/usr/bin/node
/usr/bin/nodejs
walamsoham@puppet-vm:~/exp3b2$ node /var/www/nodeapp/app.js
```