
System Administration Final Project

Topic: Ansible

Project Report

Section: BSCS 7A

By

Syed Hurrar Hasan Rizvi (1812135)
Syed Mustafa Imam (1812134)
Hirdesh Kumar (1812114)
Elliott Francis Joseph (1812110)

Date: 23-12-2021

TABLE OF CONTENT

1. Product Information	1
2. Use Cases	1
3. Architecture	2
3.1. Application Communication	2
3.2. Back-End Architecture	3
3.3. Protocols And Uses	4
4. Installation and configuration Snapshots	4
4.1.Master Node Configuration	4
4.1.1. Now installing the Ansible on the Master node	5
4.2. Node 1 Configuration	8
4.3. Node 2 Configuration	9
5. Implementation	10
5.1 Ping The Nodes	11
5.2 Playbook For Deploying Client On Node2	11
5.3 Playbook For Deploying Server On Node1	12
5.4 Running ansible playbook for deploying server on Node1	12
5.5 Running ansible playbook for deploying client on Node2	13
5.6 Node 1 is nodejsfiles server directory is created	13
5.7 Node 2 is nodejsfileclient directory is created	13
5.8 Scalaplex is running on 192.168.107.134 on 3000 port	14
5.9 Entering Credentials for Testing Purposes	15
5.10 Server POST request is successful on port 5000	15
5.11 Client Side GET request for Admin portal	16
5.12 Client side GET request successful for the movie list	16
6. References	17

1. Product Information

Ansible is an open-source automation tool, or platform, used for IT tasks such as configuration management, application deployment, infraservice orchestration, and life cycle automation. Automation is crucial these days, with IT environments that are too complex and often need to manage scalability in order to balance load management, otherwise it will be difficult for system administrators to manually configure each machine of the server rack. It is agentless which means we can access all nodes or servers from a single node known as the master node without installing it on the other nodes. It is developed by using the Python programming language. Automation simplifies complex tasks, not just making developers' jobs more manageable but allowing them to focus attention on other tasks that add value to an organization. In other words, it increases efficiency and time. It also gives you the power of deploying multi-tier applications. Ansible, as noted above, is rapidly rising to the top in the world of automation tools

2. Use Cases

Ansible as a tool is used in various different applications and situations.

A.Provisioning

Provisioning is the process for creating a suitable environment for the application to have them running.

B.Continuous Delivery

Ansible provides a simple way to automatically deploy applications. All required services for deployment can be configured from a single system. Continuous Integration is a tool that can be used to run Ansible playbooks, this can be used to test and automatically deploy the application to production if tests are passed.

C.Application Deployment

Ansible provides a simple way to deploy applications across the infrastructure. Deployment of multi-tier applications can be simplified and the infrastructure can be easily changed over the course of time.

D.Ansible for Cloud Computing

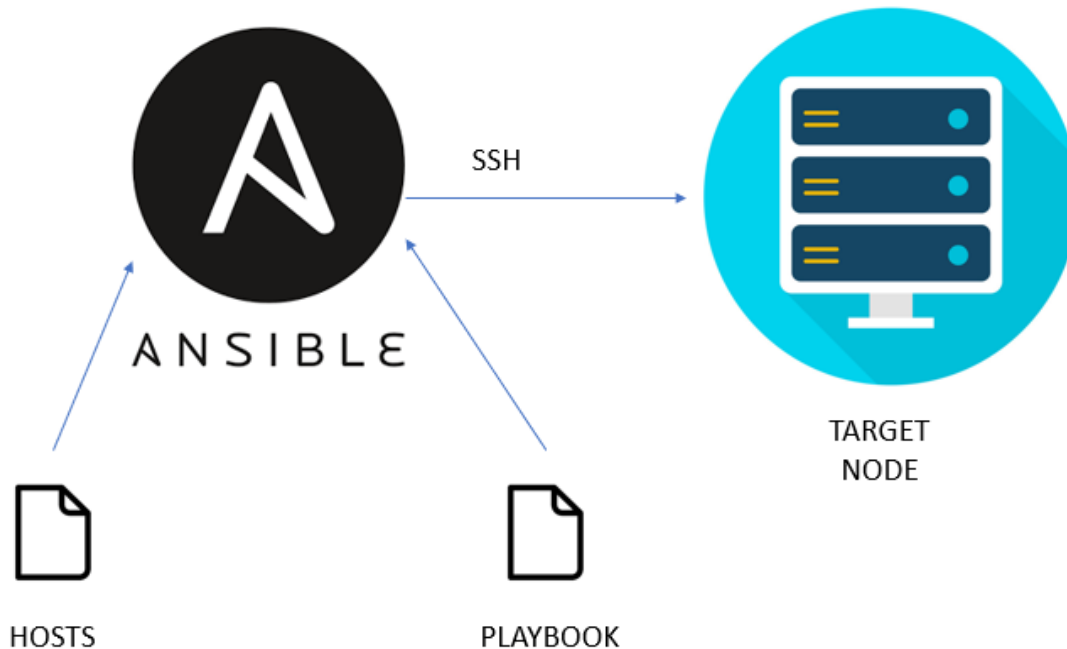
Ansible makes it simple to arrange examples across all cloud suppliers. Ansible contains various modules and permits the administration of enormous cloud foundations across the public-private and half breed cloud.

E.Ansible for Security and Compliance

It can characterize security strategies which will computerize security strategies across all machines in the organization. Security jobs once arranged in an Ansible node will be installed across all machines in the organization automatically.

So, Ansible is a must if we are working in DevOps, IT Automation, and Cloud Infrastructure.

3. Architecture



- **Inventory**

Ansible peruses data about the machines you oversee from the stock. Stock is recorded in the document which contains IP locations, data sets, and servers. It is present as a hosts file in the ansible directory present in /etc/ansible

- **Playbook**

Playbooks are files written in YAML. Playbooks depict the task to be finished by announcing setups to carry an oversaw hub into the ideal state.

- **Modules**

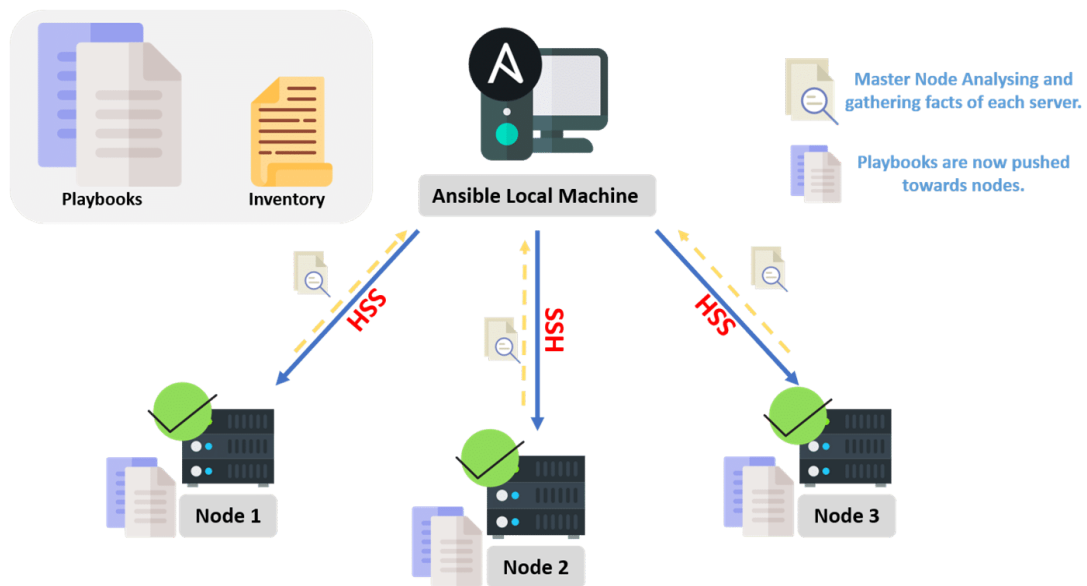
Modules are script-like programs written to determine the ideal condition of the framework. These are ordinarily written in a code editorial manager. Modules are composed by the engineer and executed through SSH. Modules are important for a bigger program called Playbook. Ansible module is an independent content that can be utilized inside an Ansible Playbook.

3.1. Application Communication

Ansible communicates with remote machines over the SSH protocol. By default, it uses native OpenSSH and connects to remote machines using the current user name, just as SSH does. It works by associating with the hubs and pushing out little projects, called "Ansible modules" to them. These projects are composed to be asset

models of the ideal condition of the framework. Ansible then, at that point, executes these modules (over SSH naturally), and eliminates them when wrapped up. The library of modules can live on any machine, and there are no servers, daemons, or data sets required. Typically the work will be done on any terminal program, a text editor, or probably a version control system to keep track of changes on the content. Passwords are supported, but SSH keys with ssh-agent are one of the best ways to use Ansible.

3.2. Back-End Architecture



In the Back-End Architecture of Ansible, the application operates on the basis of the playbook which is a YAML script created by the main system administrator which is executed on the master node. The application uses SSH protocols to communicate with the target nodes from the master node. On executing the playbook, Ansible first gathers facts about the target nodes, by fact it means valuable information about the remote system, facts are stored in JSON format and are used in making important decisions about the tasks based on their statistics. The Playbook refers to the Inventory which has all the information regarding the nodes which group they belong to and what IP they possess. Ansible will perform operations on the remote machines according to the tasks scripted in the playbook. It is based on a Push Model, so it works by pushing out small programs called modules which will accomplish automation tasks.

3.3. Protocols And Uses

The Protocols most used in Ansible is the SSH protocol. It can be extended to support other transport protocols such as SNMP or message bus by dropping a custom plugin into the connection_plugins directory. Connection plugins can be set globally in the ansible configuration.

4. Installation and configuration Snapshots

4.1.Master Node Configuration

Virtual machine name

MasterNode

Guest operating system










☐ Microsoft Windows

☒ Linux

☐ Other

Version:

Ubuntu 64-bit

Device	Summary
 Memory	2 GB
 Processors	1
 Hard Disk (SCSI)	20 GB
 CD/DVD (SATA)	Auto detect
 Network Adapter	NAT
 USB Controller	Present
 Sound Card	Auto detect
 Printer	Present
 Display	Auto detect

4.1.1. Now installing the Ansible on the Master node

- **Step 1: Update Your Master Node**

```
ubuntumaster@ubuntumaster:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease [18.0 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [35.8 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [65.9 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [277 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,468 B]
Get:10 http://us.archive.ubuntu.com/ubuntu focal-updates/main DEP-11 48x48 Icons [60.8 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [363 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [944 B]
Get:13 http://us.archive.ubuntu.com/ubuntu focal-backports/main amd64 DEP-11 Metadata [7,992 B]
Get:14 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [11.3 kB]
Fetched 1,180 kB in 5s (231 kB/s)
Reading package lists... Done
ubuntumaster@ubuntumaster:~$
```

```
ubuntumaster@ubuntumaster:~$ sudo apt-get install software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
software-properties-common is already the newest version (0.99.9.8).
The following packages were automatically installed and are no longer required:
  ansible-core python3-bcrypt python3-jinja2 python3-markupsafe python3-packaging
  python3-paramiko python3-pyparsing python3-resolvelib sshpass
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntumaster@ubuntumaster:~$
```

#It allows you to easily manage your distribution and independent software vendor software sources

- **Step 2: Install EPEL Repository**

```
ubuntumaster@ubuntumaster:~$ sudo apt-add-repository ppa:ansible/ansible
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications- automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Press [ENTER] to continue or Ctrl-c to cancel adding it.

Hit:1 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
0% [4 InRelease 71.7 kB/114 kB 63%] [3 InRelease 89.0 kB/114 kB 78%]
```

#This allows for easy access to installed packages for commonly used software packages.

- **Step 3: Install Ansible**

\$sudo apt-get update

```
ubuntu-master@ubuntu-master:~$ sudo apt-get update
Hit:1 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:3 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Fetched 336 kB in 4s (83.9 kB/s)
Reading package lists... Done
ubuntu-master@ubuntu-master:~$
```

\$sudo apt-get install ansible -y

```
ubuntu-master@ubuntu-master:~$ sudo apt-get install ansible -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  ansible
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/20.2 MB of archives.
After this operation, 279 MB of additional disk space will be used.
Selecting previously unselected package ansible.
(Reading database ... 164790 files and directories currently installed.)
Preparing to unpack .../ansible_4.9.0-1ppa~focal_all.deb ...
Unpacking ansible (4.9.0-1ppa~focal) ...
Setting up ansible (4.9.0-1ppa~focal) ...
ubuntu-master@ubuntu-master:~$
```

- **Step 4: Configure Our Node 1 & 2 User for SSH Access**

To enable SSH on the remote machines for that we use:

\$ sudo apt update

```
ubuntu-node1@ubuntu-node1:~$ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
ubuntu-node1@ubuntu-node1:~$
```


\$ sudo apt install openssh-server

```
ubuntunode1@ubuntunode1:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-4ubuntu0.3).
The following packages were automatically installed and are no longer required:
  ansible-core libevent-2.1-7 libutempter0 python3-bcrypt python3-jinja2
  python3-markupsafe python3-packaging python3-paramiko python3-pyparsing
  python3-resolvelib sshpass
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

after installation the SSH service will automatically, to verify it we use:

\$ sudo systemctl status ssh

```
ubuntunode1@ubuntunode1:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-12-23 11:11:47 PST; 17min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 831 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 852 (sshd)
    Tasks: 1 (limit: 2260)
   Memory: 2.3M
    CGroup: /system.slice/ssh.service
            └─852 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
```

#The result should be **Active: active (running)**

● Step 5: Create an Inventory

Ansible hosts file

```
[client]
192.168.107.134
[client:vars]
ansible_user=ubuntunode2
ansible_password=abc.123
[server]
192.168.107.133
[server:vars]
ansible_user=ubuntunode1
ansible_password=abc.123
```

4.2. Node 1 Configuration

Virtual machine name

Node1

Guest operating system










☐ Microsoft Windows

☒ Linux

☐ Other

Version:

Ubuntu 64-bit

Device	Summary
 Memory	2 GB
 Processors	1
 Hard Disk (SCSI)	20 GB
 CD/DVD (SATA)	Auto detect
 Network Adapter	NAT
 USB Controller	Present
 Sound Card	Auto detect
 Printer	Present
 Display	Auto detect

4.3. Node 2 Configuration

Virtual machine name

Node2

Guest operating system









☐ Microsoft Windows

☒ Linux

☐ Other

Version:

Ubuntu 64-bit

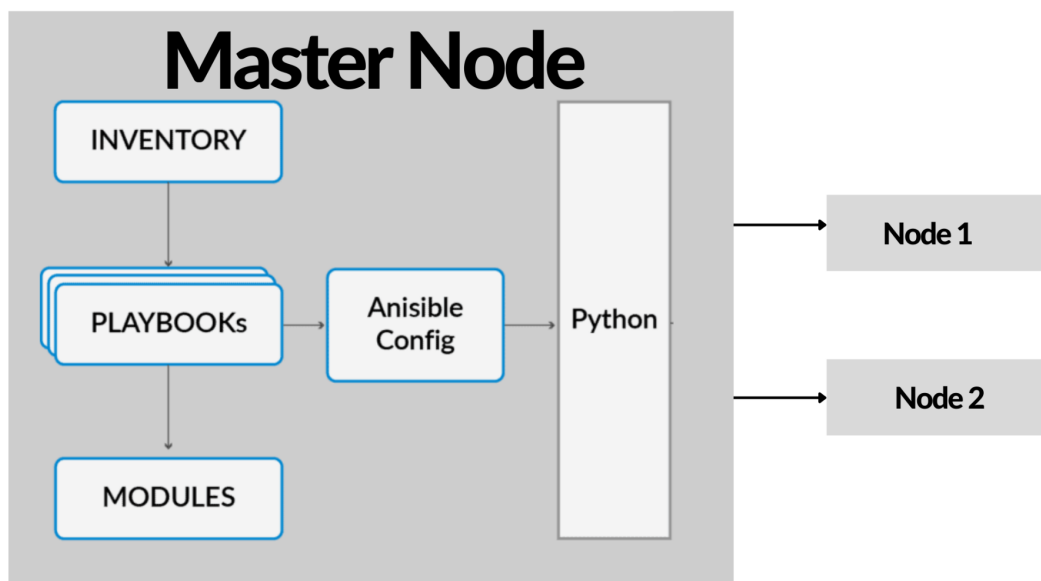
Device	Summary
 Memory	2 GB
 Processors	1
 Hard Disk (SCSI)	20 GB
 CD/DVD (SATA)	Auto detect
 Network Adapter	NAT
 USB Controller	Present
 Sound Card	Auto detect
 Printer	Present
 Display	Auto detect

5. Implementation

In this project we implemented the ansible on three virtual machines with Ubuntu 20.02 as the Operating System, one VM is the Master Node with username ubuntu-master and the other two VMs are Node1 with username ubuntu-node1 and Node2 with username ubuntu-node2 which are the target VM's.

We have made Node1 the Server Node and Node2 the Client Node. We installed all dependencies on both server and client which includes (git, curl, node, npm) packages. We deployed the playbooks to clone the git repository on node1 and node2 where node1 server side dependencies are installed through npm and node2 client side dependencies installed through yarn.

We did this to create a microservice architecture where the client is hosted on Node1 and the server is hosted on Node2, after the playbook is executed the client is deployed on Node 2 and the server is deployed on Node 1. The client communicates with the server through port 50000 where the backend services are active and running on Node1 and client is sending GET and POST requests on the port to access the services, a REST API is established. This service is also accessible through the ip address of the server machine. This all set through the ansible hosts file which can be found on the master node within the /etc/ansible directory.



5.1 Ping The Nodes

```
root@ubuntumaster:/etc/ansible# ansible all -m ping
192.168.107.134 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.107.133 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
root@ubuntumaster:/etc/ansible#
```

5.2 Playbook For Deploying Client On Node2

```
---
- name: Deploy Client On Node2
  hosts: client
  become: yes
  gather_facts: no
  vars:
    git_repo_url: https://github.com/SyedMustafaIman/Ansible-SclaPlex-Deployment
    repo_path: /home/ubuntu2/nodejsfileclient
    repo_branch: main
  tasks:
    - name: Run the equivalent of "apt get-update"
      apt:
        update_cache: yes
    - name: Ansible shell module multiple commands
      shell: 'curl -FSSL https://deb.nodesource.com/setup_14.x | sudo -E bash -'
    - name: Install required Packages
      apt: name={{ item }} state=present
      with_items:
        - git
        - curl
        - nodejs
        - npm
    - name: Clone a public repository
      git:
        repo: "https://github.com/SyedMustafaIman/Ansible-SclaPlex-Deployment"
        dest: "/home/ubuntu2/nodejsfileclient"
        version: "main"
        accept_hostkey: yes
        force: yes
    - name: Install yarn globally
      community.general.npm:
        name: yarn
        state: latest
        global: yes
    - name: Install packages based on package.json using the yarn
      community.general.yarn:
        path: "/home/ubuntu2/nodejsfileclient/client/"
        state: latest
    - name: start the application
      command: yarn start
      args:
        chdir: "/home/ubuntu2/nodejsfileclient/client/"
```

5.3 Playbook For Deploying Server On Node1

```

---
- name: Deploy Server On Node1
  hosts: server
  become: yes
  gather_facts: no
  vars:
    git_repo_url: https://github.com/SyedMustafaIman/Ansible-SclaPlex-Deployment
    repo_path: /home/ubuntu1/nodejsfileserv
    repo_branch: main

  tasks:
    - name: Run the equivalent of "apt get-update"
      apt:
        update_cache: yes

    - name: Ansible shell module multiple commands
      shell: 'curl -fSSL https://deb.nodesource.com/setup_14.x | sudo -E bash -'

    - name: Install required Packages
      apt: name={{ item }} state=present
      with_items:
        - git
        - curl
        - nodejs
        - npm

    - name: Clone a public repository
      git:
        repo: "https://github.com/SyedMustafaIman/Ansible-SclaPlex-Deployment"
        dest: "/home/ubuntu1/nodejsfileserv"
        version: "main"
        accept_hostkey: yes
        force: yes

    - name: Install nodemon globally
      community.general.npm:
        name: nodemon
        state: latest
        global: yes

    - name: Install packages based on package.json using the npm
      community.general.npm:
        path: "/home/ubuntu1/nodejsfileserv/server/"
        state: latest

    - name: start the application
      command: nodemon
      args:
        chdir: "/home/ubuntu1/nodejsfileserv/server/"
  
```

5.4 Running ansible playbook for deploying server on Node1

```

root@ubuntu1master:/etc/ansible# ansible-playbook deployserveronnode1.yml -K
BECOME password:

PLAY [Deploy Server On Node1] *****

TASK [Run the equivalent of "apt get-update"] *****
changed: [192.168.107.133]

TASK [Ansible shell module multiple commands] *****
changed: [192.168.107.133]

TASK [Install required Packages] *****
changed: [192.168.107.133] => (item=git)
changed: [192.168.107.133] => (item=curl)
changed: [192.168.107.133] => (item=nodejs)
changed: [192.168.107.133] => (item=npm)

TASK [Clone a public repository] *****
changed: [192.168.107.133]

TASK [Install nodemon globally] *****
ok: [192.168.107.133]

TASK [Install packages based on package.json using the npm] *****
changed: [192.168.107.133]

TASK [start the application] *****
  
```

5.5 Running ansible playbook for deploying client on Node2

```
root@ubuntumaster:/etc/ansible# ansible-playbook deployclientonnode2.yml -K
BECOME password:

PLAY [Deploy Client On Node2] *****

TASK [Run the equivalent of "apt get-update"] *****
changed: [192.168.107.134]

TASK [Ansible shell module multiple commands] *****
changed: [192.168.107.134]

TASK [Install required Packages] *****
changed: [192.168.107.134] => (item=git)
changed: [192.168.107.134] => (item=curl)
changed: [192.168.107.134] => (item=nodejs)
changed: [192.168.107.134] => (item=npm)

TASK [Clone a public repository] *****
changed: [192.168.107.134]

TASK [Install yarn globally] *****
ok: [192.168.107.134]

TASK [Install packages based on package.json using the yarn] *****
changed: [192.168.107.134]

TASK [start the application] *****
```

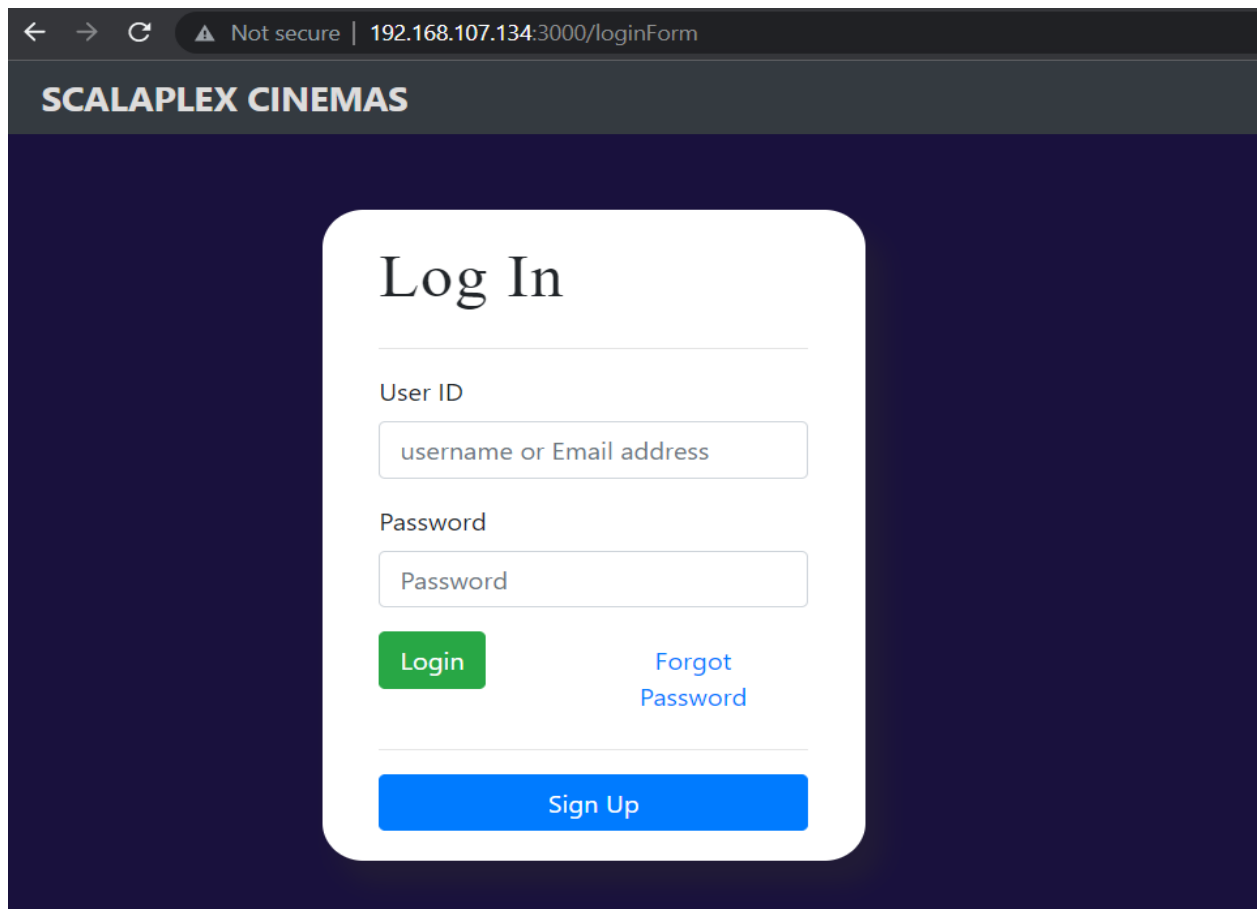
5.6 Node 1 is nodejsfiles server directory is created

```
ubuntunode1@ubuntunode1:~$ ls
Desktop  Downloads  nodejsfiles server  Public  Templates
Documents Music      Pictures            stu1    Videos
ubuntunode1@ubuntunode1:~$
```

5.7 Node 2 is nodejsfileclient directory is created

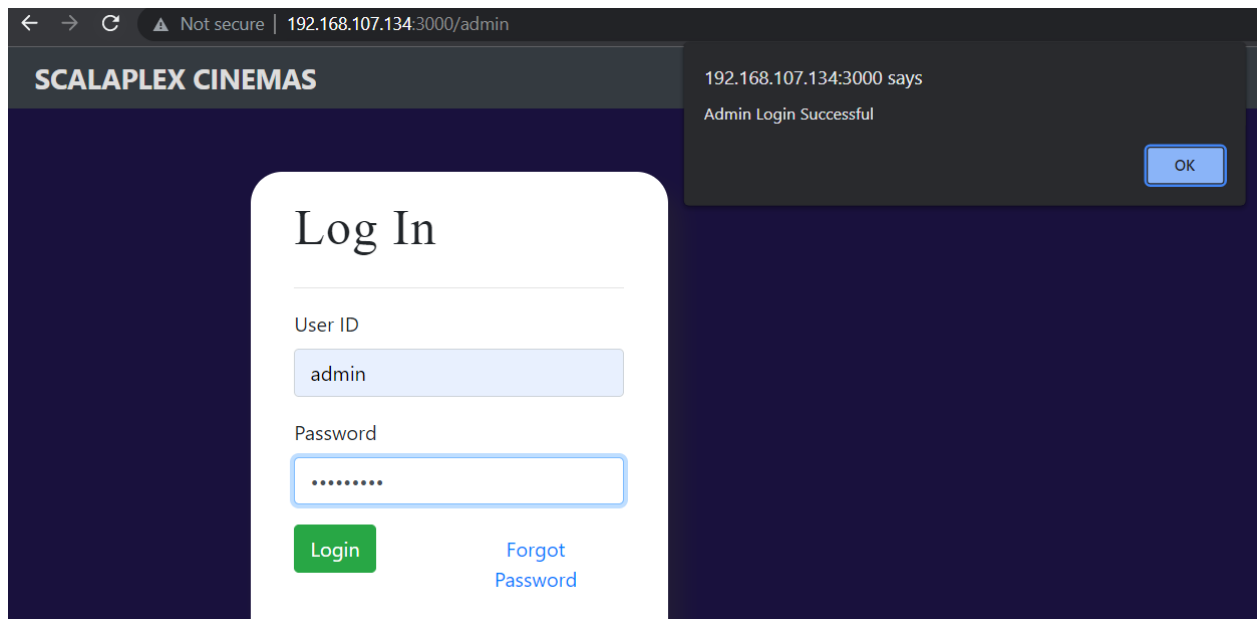
```
ubuntunode2@ubuntunode1:~$ ls
Desktop  Downloads  nodejsfileclient  Public  Templates
Documents Music      Pictures            stu1    Videos
ubuntunode2@ubuntunode1:~$
```

5.8 Scalaplex is running on 192.168.107.134 on 3000 port

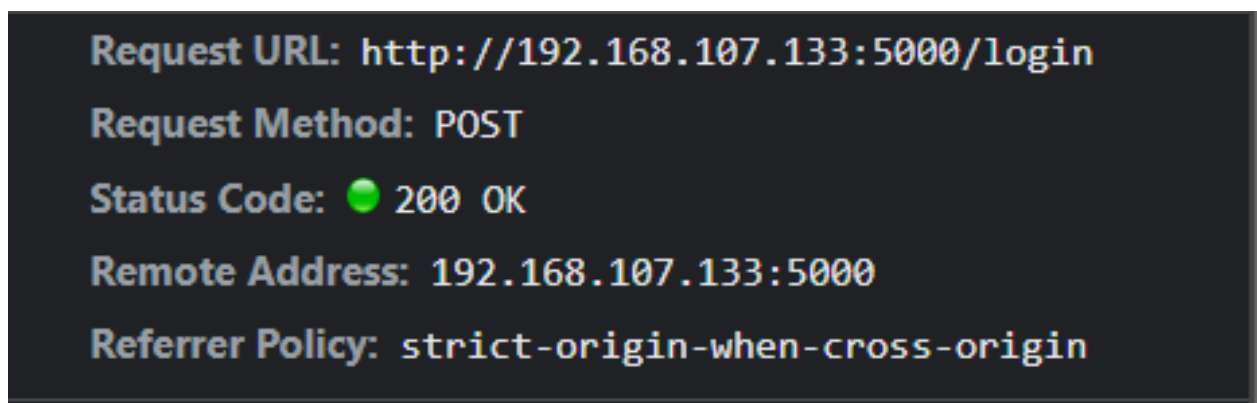


The screenshot shows a web browser window with the address bar displaying "192.168.107.134:3000/loginForm". The page has a dark blue header with the text "SCALAPLEX CINEMAS". The main content area is a dark blue gradient. In the center, there is a white rounded rectangle containing the "Log In" form. The form has a title "Log In" followed by a horizontal line. Below the line, there are two input fields: "User ID" with a placeholder "username or Email address" and "Password" with a placeholder "Password". To the left of the "Password" field is a green "Login" button. To the right of the "Login" button is a blue link "Forgot Password". Below these fields is a blue "Sign Up" button.

5.9 Entering Credentials for Testing Purposes



5.10 Server POST request is successful on port 5000



```
Request URL: http://192.168.107.134:3000/home
Request Method: GET
Status Code: 200 OK
Remote Address: 192.168.107.134:3000
Referrer Policy: strict-origin-when-cross-origin
```

```
Request URL: http://192.168.107.134:3000/index/admin/listmovies
Request Method: GET
Status Code: 200 OK
Remote Address: 192.168.107.134:3000
Referrer Policy: strict-origin-when-cross-origin
```

5.11 Client Side GET request for Admin portal

Add New Movies	Movie's List	Customers's List	Add Show Time			
Name	Username/Email	Gender	Contact	City	Address	
Muhammad Asad Rehman	asad@gmail.com	male	15	Karachi	B-103, block-2, Clifton	
Anas Saleem	anas@gmail.com	male	03451234567	Karachi	defence phase 8	
Mustafa Imam	cs1812134@szabist.pk	male	3032845326	Karachi	Ashraf Compound House#A-55, Block "E", North naizmabad, Karachi, Shindh, Pakist	

5.12 Client side GET request successful for the movie list

Movie Name	Movie Bought Date	Genre	Duration	Year
Lila	2021-05-12	Family, Horror	3hr 32min	1985
Avengers	2021-06-15	Family, Comedy	3hr 30min	1999
King Kong	2021-05-05	Fantasy	2hr 30min	2021
Knight Rider	2021-05-06	Fantasy	2hr 30min	2000
Ertugal	2021-06-23	Family, Hoor	3hr 32min	1999
Godliz	2021-06-04	Family, Comedy	3hr 32min	1999
End Game	2021-05-07	Fantasy	3hr 21min	2020
Sniper	2019-06-04	Shooting	2hr 25min	2019
Raiz	2021-05-07	Family	3hr 32min	2021

6. References

- <https://www.ansible.com/contact-us?hsCtaTracking=c179b300-92c4-4bf9-b035-11fe8e147b3e%7Cd89a6445-6b7c-4d68-ab2e-aad1f4ab567f>
- <https://docs.ansible.com/ansible/latest/index.html>
- <https://www.youtube.com/watch?v=5hycyr-8EKs>
- <https://medium.com/nonstopio/deploying-a-node-js-app-using-ansible-cfe7dfeddcac>
- <https://docs.ansible.com/ansible/latest/plugins/connection.html>