

# **CLOUD ARCHITECTURE MANAGEMENT (CAM) LAB REPORT**

## **Submitted By:**

Name: Sneha S Shetty

Reg Number: 251100680004

Branch: Cloud Computing

## **Submitted To:**

Faculty Name: Dr Prathviraj N

Associate professor at the Manipal School  
of Information Sciences

**Academic Year: 2025 – 2026**

251100680004

## Experiment 1: Passwordless Authentication & NFS Setup

### Theory:

This experiment demonstrates secure access between two Linux systems using RSA-based public-key cryptography. RSA allows passwordless login by pairing a private key with a public key.

NFS (Network File System) shares directories across networked systems, enabling multiple machines to mount and use the same folder as local storage.

### Procedure:

The steps followed for this experiment along with screenshots are as follows:

- Generate RSA keys using ssh-keygen.

```
ssh-keygen -t rsa -b 4096
```

```
msis@k8s-master-node:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/msis/.ssh/id_rsa): ssh
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ssh
Your public key has been saved in ssh.pub
The key fingerprint is:
SHA256:300500Na5xzKo+/SouaQDyMT4zVdGHRul6eV56WxPDM msis@node1
The key's randomart image is:
+---[RSA 4096]----+
| . |
| . o |
| . o . |
| + . o . |
| o o S + O |
| . + + . * X + |
| + = . =O * . |
| o =..o=.=E= |
| o+.o=o =+ |
+---[SHA256]-----+
msis@k8s-master-node:~$
```

- Start SSH agent and add private key.

```
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_rsa
```

251100680004

- **Copy public key to remote systems using ssh-copy-id.**

```
ssh-copy-id msis@slaveip1
ssh-copy-id msis@slaveip2
```

```
msis@k8s-master-node:~$ ssh-copy-id msis@172.18.181.50
The authenticity of host '172.18.181.50 (172.18.181.50)' can't be established.
ED25519 key fingerprint is SHA256:E7UHfqaD3PvAlqBwx7SrM2plvz0DgRlMB1S9llY+d5M.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 2 key(s) remain to be installed -- if you are prompted now it is to install the new keys
msis@172.18.181.50's password:

Number of key(s) added: 2

Now try logging into the machine, with:  "ssh 'msis@172.18.181.50'"
and check to make sure that only the key(s) you wanted were added.

msis@k8s-master-node:~$
```

- **Test passwordless login.**

```
ssh msis@slaveip1
ssh msis@slaveip2
```

// it successfully allows to login without asking for password

```
msis@k8s-master-node:~$ ssh msis@172.18.181.50
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.11.0-17-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

439 updates can be applied immediately.
237 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

4 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Sat Nov 15 09:26:18 2025 from 172.18.181.38
msis@msis:~$
```

251100680004

- **Install NFS packages on all systems(master and slaves).**

```
sudo apt update
sudo apt install nfs-kernel-server -y //on the master
sudo apt install nfs-common -y      //on the slaves
```

```
msis@k8s-master-node:~$ sudo apt install nfs-kernel-server -y
[sudo] password for msis:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nfs-kernel-server is already the newest version (1:2.6.4-3ubuntu5.1).
0 upgraded, 0 newly installed, 0 to remove and 423 not upgraded.
msis@k8s-master-node:~$
```

```
msis@msis: $ sudo apt install nfs-common -y
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 1033718 (apt-get)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 1033718 (apt-get)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 1033718 (apt-get)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 1033718 (apt-get)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 1033718 (apt-get)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 1033718 (apt-get)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 1033718 (apt-get)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 1033718 (apt-get)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 1033718 (apt-get)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 1033718 (apt-get)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 1033718 (apt-get)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 1033718 (apt-get)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 1033718 (apt-get)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nfs-common is already the newest version (1:2.6.4-3ubuntu5.1).
0 upgraded, 0 newly installed, 0 to remove and 450 not upgraded.
msis@msis: $
```

## 6. Create a Shared Directory on the master

```
sudo mkdir -p /nfs/shared
sudo chmod 777 /nfs/shared
```

## 7. Configure and edit the NFS exports file on the master

```
sudo nano /etc/exports
```

//Add the following lines

```
/nfs/shared 172.18.181.38(rw,sync,no_subtree_check)          //here 172.18.181.38 is slaveip1
/nfs/shared 172.18.181.50(rw,sync,no_subtree_check)          //here 172.18.181.50 is slaveip2
```

251100680004

```
GNU nano 7.2                                         /etc/exports *
# /etc/exports: the access control list for filesystems which may be exported
#           to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4       gss/krb5i(rw,async,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,async,no_subtree_check)
#
/nfs/shared 172.18.181.38(rw,async,no_subtree_check)
/nfs/shared 172.18.181.50(rw,async,no_subtree_check)
```

## 8. Export the Shared Directory ,Apply and verify configuration on master

```
sudo exportfs -a
sudo systemctl restart nfs-kernel-server
sudo exportfs -v
```

```
sis@k8s-master-node:~$ sudo mkdir -p /nfs/shared
sis@k8s-master-node:~$ sudo chmod 777 /nfs/shared
sis@k8s-master-node:~$ sudo nano /etc/exports
sis@k8s-master-node:~$ sudo exportfs -a
sis@k8s-master-node:~$ sudo systemctl restart nfs-kernel-server
sis@k8s-master-node:~$ sudo exportfs -v
nfs/shared      172.18.181.38(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
nfs/shared      172.18.181.50(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
sis@k8s-master-node:~$ █
```

## 9. Mount the NFS directory on Slave Nodes

//On slave1

```
sudo mkdir -p /nfs/shared
sudo mount 172.18.181.38:/nfs/shared /nfs/shared
```

//On slave2

```
sudo mkdir -p /nfs/shared
sudo mount 172.18.181.65:/nfs/shared /nfs/shared
```

## 10. Verify the Mount

//Run on any slave

```
df -h | grep nfs
```

//Output would be 172.18.181.38:/nfs/shared 100G 2G 98G 2% /nfs/shared

```
msis@control:~$ df -h | grep nfs
172.18.181.45:/nfs/shared 937G 247G 643G 28% /nfs/shared
msis@control:~$ █
```

251100680004

//Test by creating a file from one slave and checking its visibility on other slave or master

*cd /nfs/shared //from slave1  
touch testfile*

```
msis@msis: $ cd /nfs/shared  
msis@msis:/nfs/shared$ touch testsfile  
msis@msis:/nfs/shared$ touch demo  
msis@msis:/nfs/shared$
```

// Next check if it appears on master or slave2

*cd /nfs/shared //from master  
ls //testfile will appear*

```
msis@k8s-master-node:~$ cd /nfs/shared  
msis@k8s-master-node:/nfs/shared$ ls  
demo testsfile  
msis@k8s-master-node:/nfs/shared$ █
```

251100680004

## Experiment 2: Java Web Services – Calculator Program

### Theory:

Java web services allow distributed communication using SOAP or REST.

SOAP uses XML-based messaging and WSDL, while REST uses lightweight HTTP methods. The calculator web service exposes arithmetic operations remotely.

### Procedure:

The steps followed for this experiment along with screenshots are as follows:

#### 1. Check if Apache Tomcat 9 is present

```
sudo systemctl restart tomcat  
go to localhost:<tomcat_port_no> // to verify tomcat is running  
sudo systemctl status tomcat //this should show status active
```

The screenshot shows the Apache Tomcat 9.0.108 homepage. The URL is <http://localhost:8081>. The page displays a success message: "If you're seeing this, you've successfully installed Tomcat. Congratulations!" It features a cartoon cat logo and links to recommended reading like Security Considerations How-To, Manager Application How-To, and Clustering/Session Replication How-To. The Developer Quick Start section includes links for Tomcat Setup, First Web Application, Realms & AAA, JDBC DataSources, Examples, and Servlet Specifications/Tomcat Versions. The Documentation section links to Tomcat 9.0 Documentation, Tomcat 9.0 Configuration, and Tomcat Wiki. The Getting Help section links to FAQ and Mailing Lists, mentioning tomcat-announce, tomcat-users, tomcat-dev, and tomcat-javadoc mailing lists. Other sections include Other Downloads, Other Documentation, Get Involved, Miscellaneous, and Apache Software Foundation links.

```
nsis@node1:~$ sudo systemctl status tomcat
● tomcat.service - Tomcat
   Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-11-16 11:30:38 IST; 15min ago
     Process: 1626 ExecStart=/opt/tomcat/bin/startup.sh (code=exited, status=0/S>
    Main PID: 1670 (java)
       Tasks: 31 (limit: 18567)
      Memory: 179.9M (peak: 191.1M)
        CPU: 4.296s
       CGroup: /system.slice/tomcat.service
               └─1670 /usr/lib/jvm/java-1.17.0-openjdk-amd64/bin/java -Djava.util.

Nov 16 11:30:38 node1 systemd[1]: Starting tomcat.service - Tomcat...
Nov 16 11:30:39 node1 startup.sh[1626]: Tomcat started.
Nov 16 11:30:38 node1 systemd[1]: Started tomcat.service - Tomcat.
```

251100680004

## 2. Install JDK and IntelliJ IDEA Ultimate

```
sudo apt update  
sudo apt install openjdk-8-jdk
```

```
sudo snap install intellij-idea-ultimate --classic
```

```
msis@node1:~$ sudo apt update  
      sudo apt install openjdk-8-jdk  
  
      sudo snap install intellij-idea-ultimate --classic  
[sudo] password for msis:  
Hit:1 https://download.docker.com/linux/ubuntu noble InRelease  
Ign:2 https://pkg.jenkins.io/debian binary/ InRelease  
Hit:3 https://pkg.jenkins.io/debian binary/ Release  
Hit:5 http://archive.ubuntu.com/ubuntu noble InRelease  
Get:7 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Get:8 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Hit:9 http://apt.puppet.com jammy InRelease  
Hit:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease  
Get:10 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]  
Get:11 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]  
Get:12 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]  
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,585 kB]  
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.3 kB]  
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]  
Get:16 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]  
Get:17 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]  
Get:18 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1,499 kB]  
Get:19 http://archive.ubuntu.com/ubuntu noble-updates/universe i386 Packages [988 kB]  
Get:20 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [378 kB]  
Get:21 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]  
Get:22 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7,140 B]  
Get:23 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]  
Get:24 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11.0 kB]  
Get:25 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]  
Fetched 5,097 kB in 7s (753 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
423 packages can be upgraded. Run 'apt list --upgradable' to see them.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
openjdk-8-jdk is already the newest version (8u462-ga-us1-0ubuntu2~24.04.2).  
0 upgraded, 0 newly installed, 0 to remove and 423 not upgraded.
```

## 3. Setting Up Tomcat in IntelliJ IDEA

- Open Settings  
*File → Settings → Build, Execution, Deployment*
- Open Application Servers  
*On the left side, click: Application Servers*
- Add a New Server  
*Click the '+' icon : Select Tomcat Server*
- Select Tomcat Home Directory  
*In the Tomcat Home field, enter the Tomcat installation path - /opt/tomcat  
If there is Error / Warning then fix Permission Issue in Terminal*

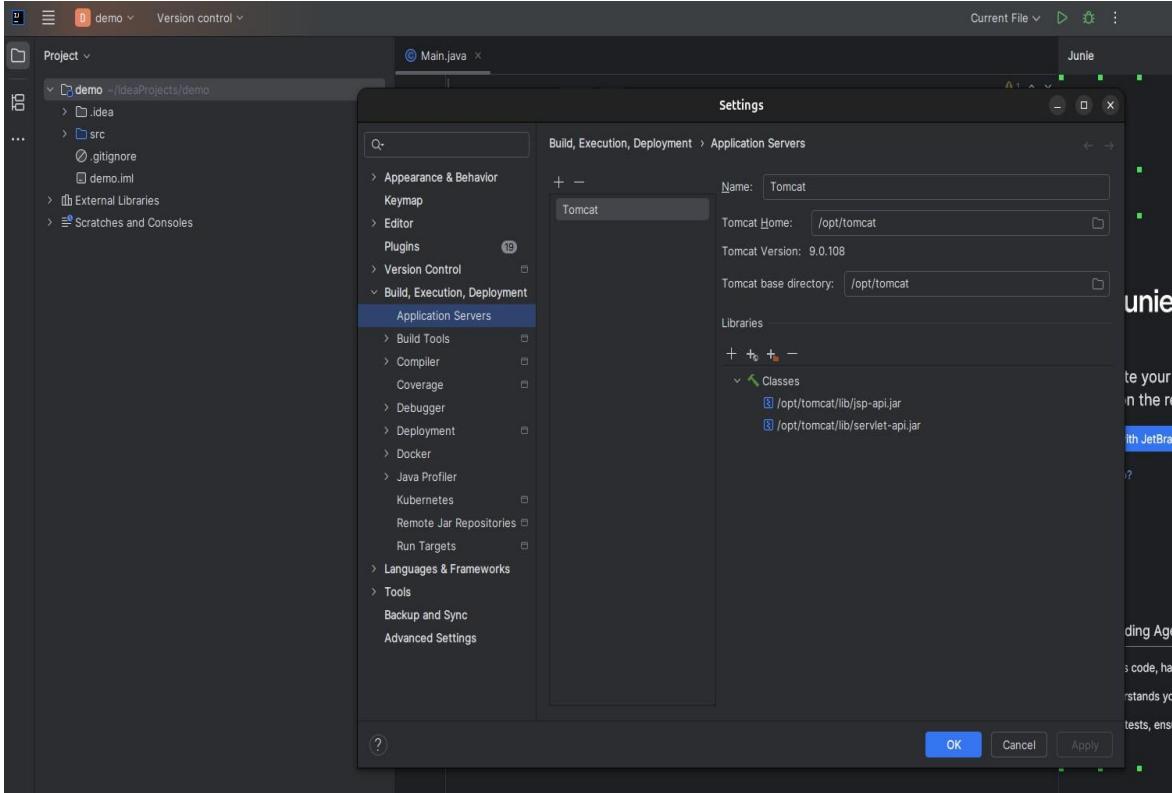
251100680004

Open terminal and run:

`sudo chmod -R a+rX /opt/tomcat` // gives read and execute permission to all users

- Apply and Save

Go back to IntelliJ → **Apply** → **OK**



#### 4. Create a Web Application (Server Side) – Calculator Project

- Open IntelliJ IDEA

Go to **File** → **New Project**

- Choose Java Enterprise / **Jakarta EE**

- Project Name: **calculator**

- On the right side, under templates, choose: **Web Application**

- Under Application Server, choose: **Apache Tomcat** (Tomcat 9.0)

- Keep everything else default and click **Finish** → wait for the project to load

#### 5. Create service package and class

- Under calculator in src/main/java right-click the package `com.example.calculator` → **Select New** → **Java Class** → Create a class named **OperatorService.java** and add below code

251100680004

```
package com.example.calculator;

import java.util.List;
import javax.jws.WebMethod;
import javax.jws.WebService;

@WebService
public class OperatorService {

    @WebMethod
    public int add(int a, int b) { return a + b; }

    @WebMethod
    public int sub(int a, int b) { return a - b; }

    @WebMethod
    public int mul(int a, int b) { return a * b; }

    @WebMethod
    public int div(int a, int b) { return a / b; }
```

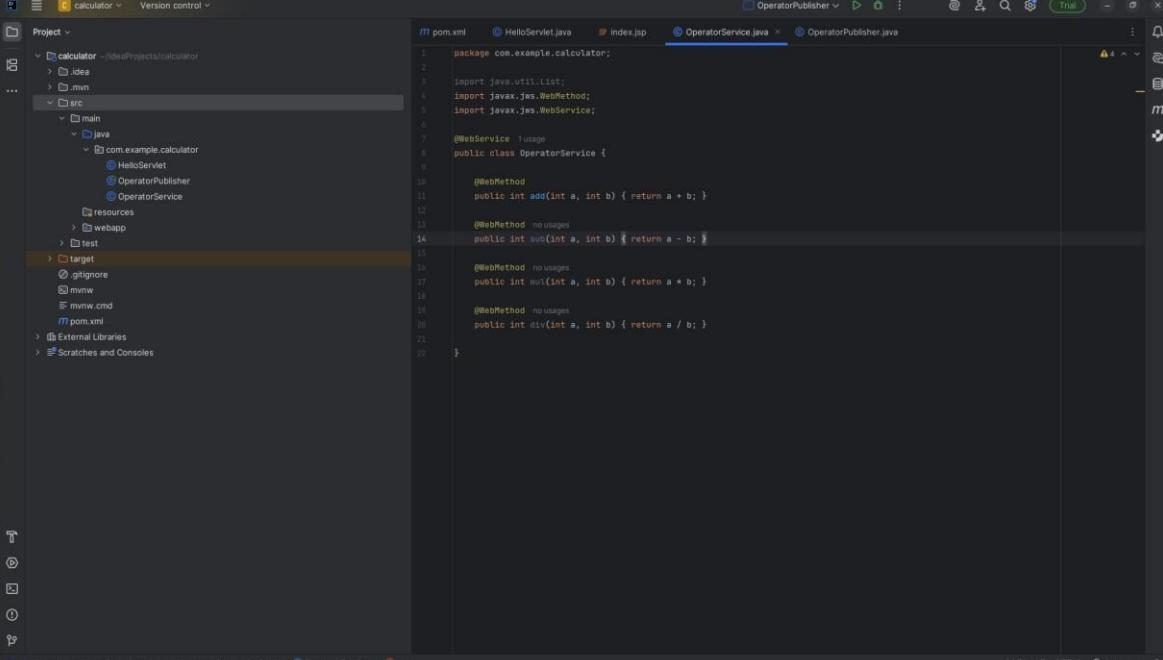
- Similarly in the same package, create **OperatorPublisher.java** and add below code

```
package com.example.calculator;

import javax.xml.ws.Endpoint;

public class OperatorPublisher {
    public static void main(String[] args) {
        String url = "http://localhost:8085/calculator";
        Endpoint.publish(url, new OperatorService());
        System.out.println("Service started at: " + url + "?wsdl");
    }
}
```

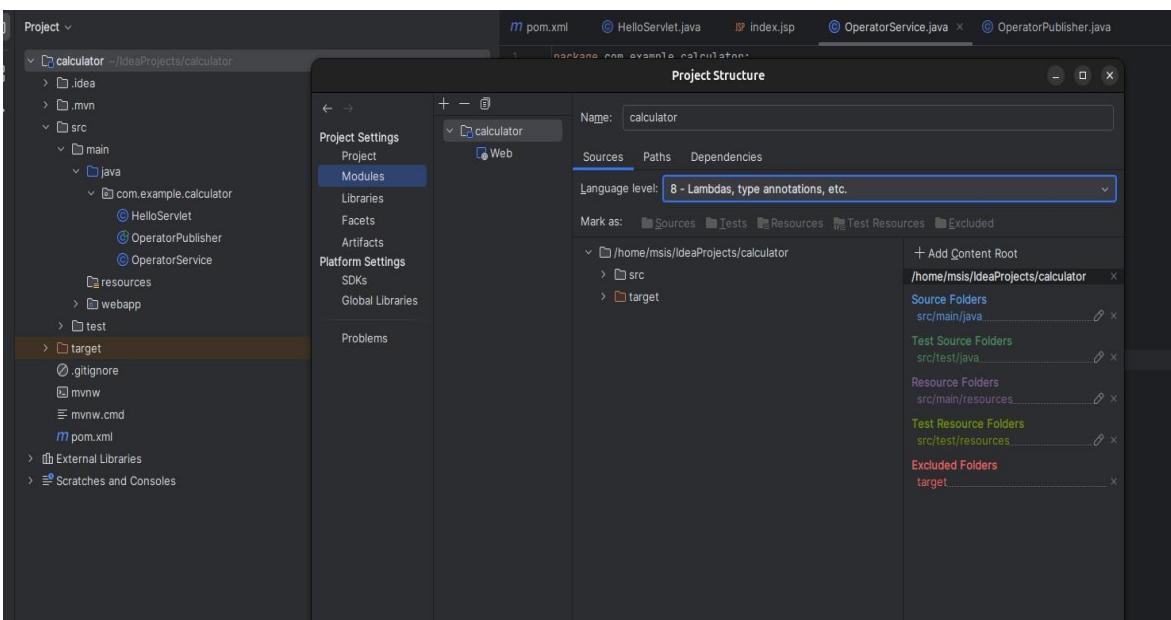
251100680004



```
1 package com.example.calculator;
2
3 import java.util.List;
4 import javax.jws.WebMethod;
5 import javax.jws.WebService;
6
7 /**
8 * Usage
9 */
10 public class OperatorService {
11
12     /**
13      * no usage
14     */
15     public int add(int a, int b) { return a + b; }
16
17     /**
18      * no usage
19     */
20     public int sub(int a, int b) { return a - b; }
21
22     /**
23      * no usage
24     */
25     public int mult(int a, int b) { return a * b; }
26
27     /**
28      * no usage
29     */
30     public int div(int a, int b) { return a / b; }
31 }
```

## 6. Set Language Level in Project Structure

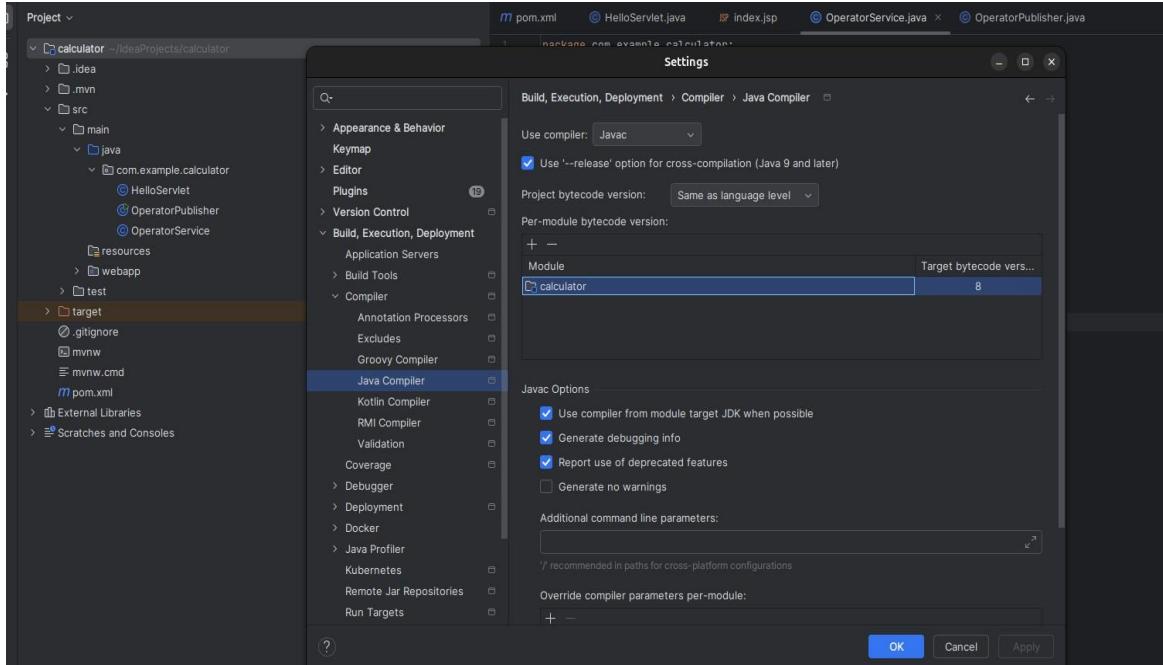
- File → *Project Structure*
- Click **Modules** on the left side.
- Select your module: *calculator*
- **Sources** tab -> Set **Language Level** -> 8 – *Lambdas, type annotations, etc.*
- Open **Dependencies** -> Under **Module SDK** -> **Project SDK (1.8) /Java 1.8**
- Click **Apply** → **OK**.



## 7. Set Target Bytecode Version

251100680004

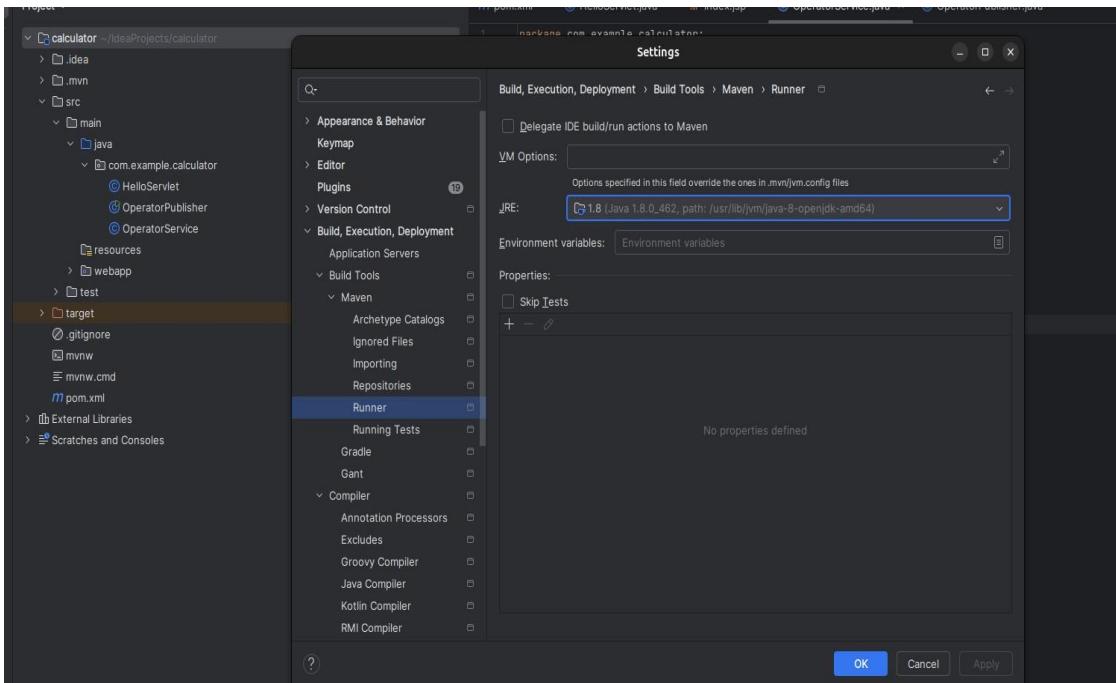
- *File → Settings → Build, Execution, Deployment → Compiler → Java Compiler*
- Under **Per-module bytecode version** → Find your module **calculator** → Set **Target bytecode version = 8**
- Click **Apply** → **OK**.



## 8. Set Maven Runner JRE (Java 1.8)

- *File → Settings → Build, Execution, Deployment → Build Tools → Maven → Runner*
- Under **JRE -> Java 1.8**
- Click **Apply** → **OK**.

251100680004



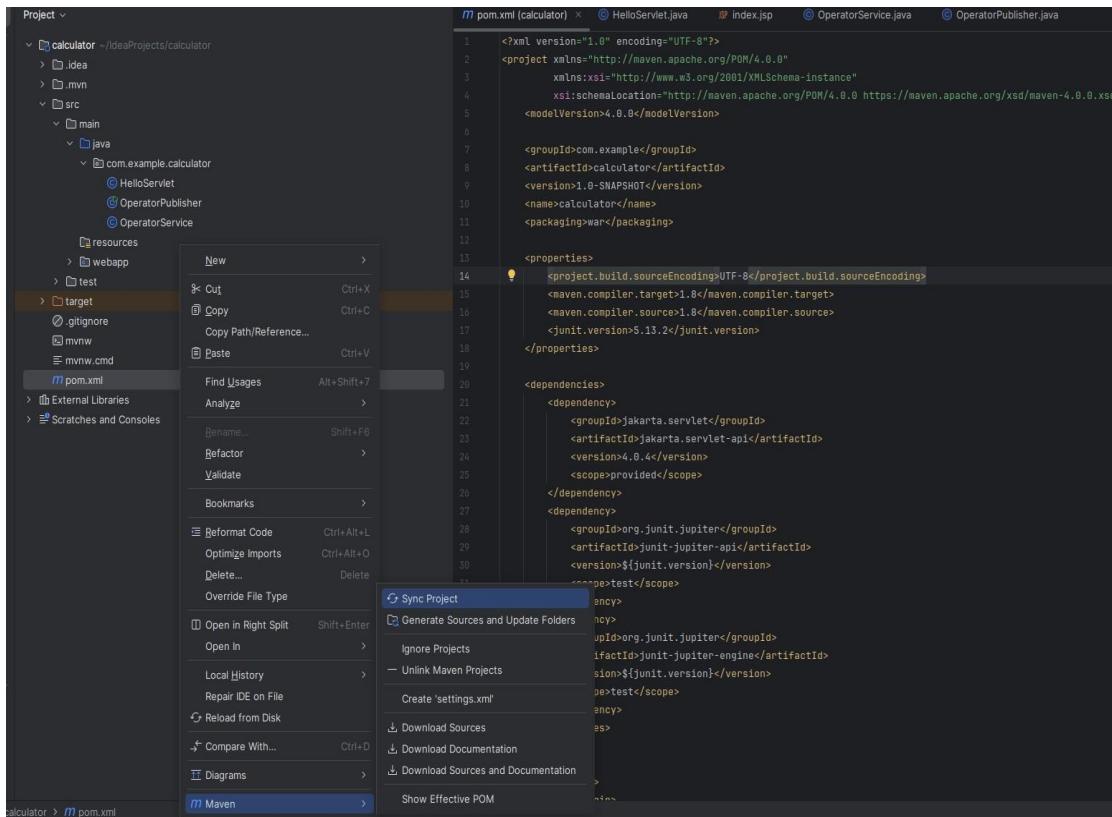
## 9. Update pom.xml Settings (Set Java Version & Servlet Version)

- In IntelliJ, look at the **left navigation** panel -> Click to open **pom.xml**.
- Locate the `<properties>` section.
- Make sure the versions are set to 1.8 like this :

```
<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.target>1.8</maven.compiler.target>      //change to 1.8
    <maven.compiler.source>1.8</maven.compiler.source>    // change to 1.8
    <junit.version>5.13.2</junit.version>
</properties>
```

- Scroll to the **dependencies** section.
- Check the `jakarta.servlet-api` dependency. Make sure version is: **4.0.4**
- If your file had any other version (like 5.x or newer), replace it with **4.0.4**.
- After modifying, right-click **pom.xml** → **Maven** → **Sync Project**.

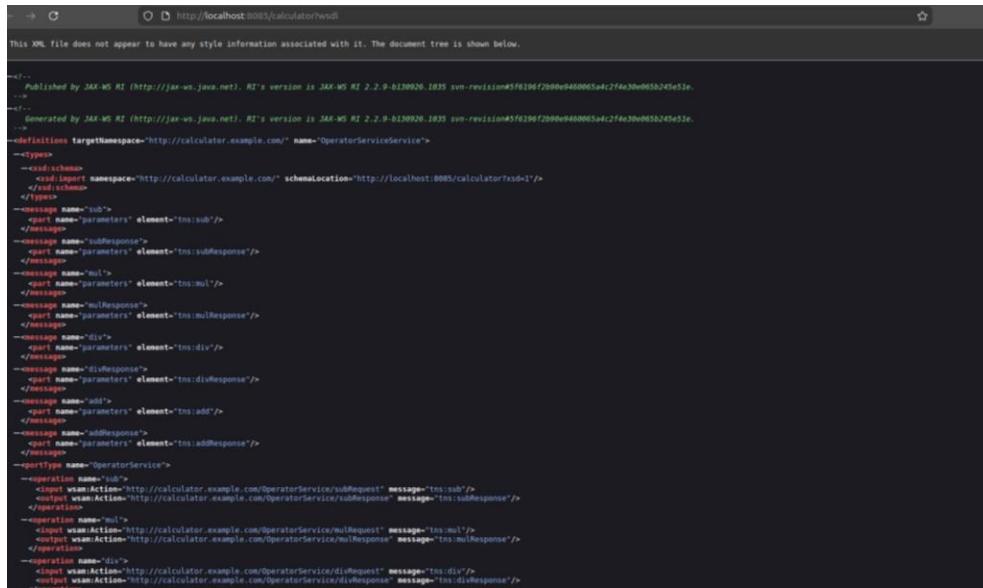
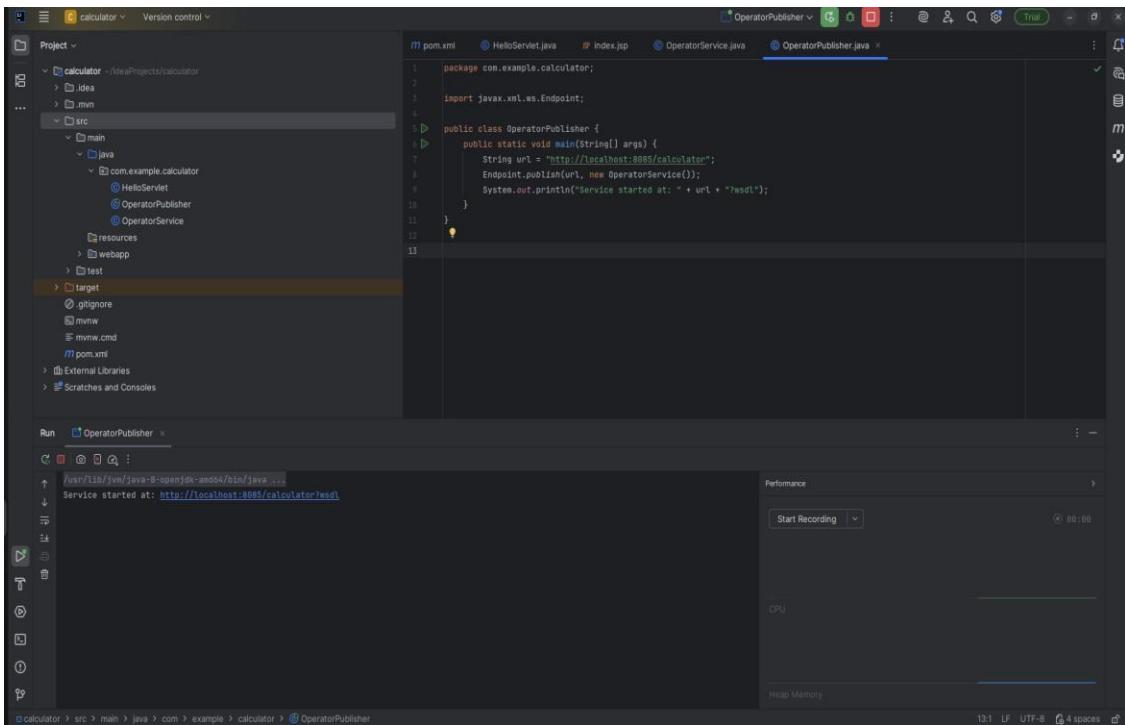
251100680004



## 10. Run the Web Service (Server Application)

- If `HelloServlet.java` shows errors -> open file -> Replace all `jakarta.servlet.*` with `javax.servlet.*`
- Build → Rebuild Project
- Open class **OperatorPublisher.java** → Run 'OperatorPublisher'
- Check Console Output :  
Service started at <http://localhost:8085/calculator>
- Access the WSDL in Browser : open <http://localhost:8085/calculator?wsdl>
- If the WSDL loads successfully -  
Your web service is deployed and running correctly and you will get a page pasted below.

251100680004



## 11. Create a Client Application - Client Project

- Open IntelliJ IDEA  
*Go to File → New Project*
- Do NOT select Java Enterprise / Jakarta EE. -> Choose Java (simple Java project).
- Project Name: **client** -> keep default -> Click **Finish** -> **Open in New Window**.  
*(Do not close the server project.)*

251100680004

- Generate Client Proxy Classes inside src/com/example/client using **wsimport**  
Open system Terminal -> Change directory to the client src folder:  
`cd /IdeaProjects/client/src/`  
Run the **wsimport** command:  
`wsimport -keep -p com.example.client http://localhost:8085/calculator?wsdl`

```
msis@node1:~$ cd IdeaProjects/client/src/
msis@node1:/IdeaProjects/client/src$ wsimport -keep -p com.example.client http://localhost:8085/calculator?wsdl
parsing WSDL...

Generating code...
Compiling code...
msis@node1:/IdeaProjects/client/src$
```

- Create Client File

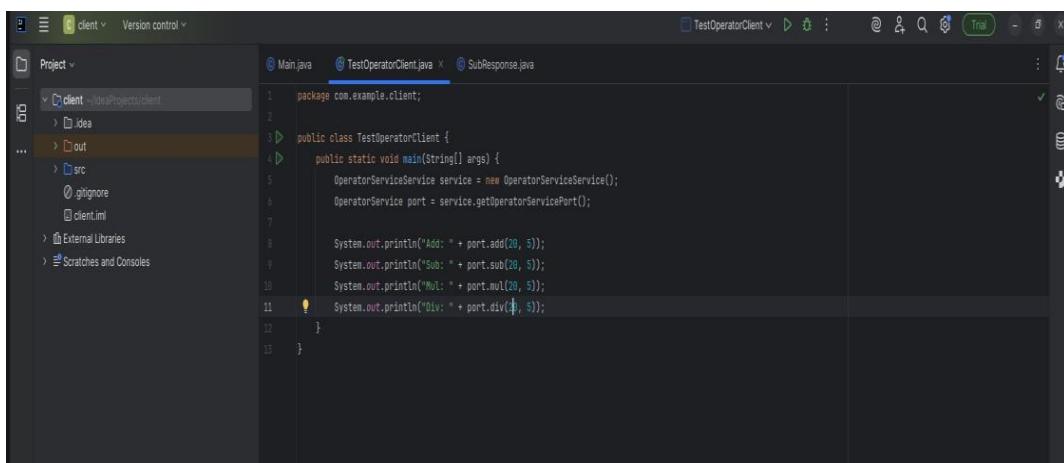
client  
  └── src  
    └── com  
      └── example  
        └── client

Right-click **com.example.client** → New → Java Class → **TestOperatorClient**  
Add the following code:

```
package com.example.client;

public class TestOperatorClient {
    public static void main(String[] args) {
        OperatorServiceService service = new OperatorServiceService();
        OperatorService port = service.getOperatorServicePort();

        System.out.println("Add: " + port.add(20, 5));
        System.out.println("Sub: " + port.sub(20, 5));
        System.out.println("Mul: " + port.mul(20, 5));
        System.out.println("Div: " + port.div(20, 5));
    }
}
```



251100680004

## 12. Run the Client Application

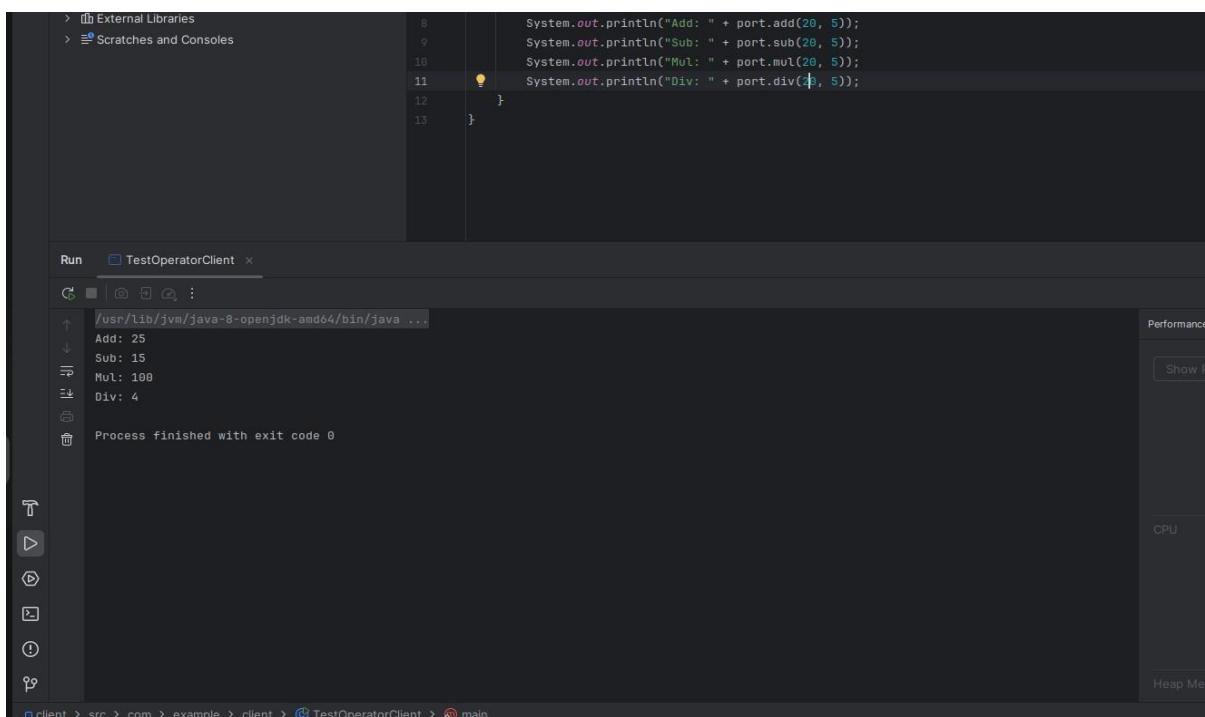
Ensure your **server project** (OperatorPublisher) is still running.

*In client project -> right-click **TestOperatorClient.java**.*

*Select **Run 'TestOperatorClient'**.*

Output : will get the following:

```
Add: 25
Sub: 15
Mul: 100
Div: 4
```



The screenshot shows the IntelliJ IDEA interface during the execution of the `TestOperatorClient` application. The code editor displays the following Java code:

```
8     System.out.println("Add: " + port.add(20, 5));
9     System.out.println("Sub: " + port.sub(20, 5));
10    System.out.println("Mul: " + port.mul(20, 5));
11    System.out.println("Div: " + port.div(20, 5));
12 }
13 }
```

The run tool window at the bottom shows the output of the application:

```
/usr/lib/jvm/java-8-openjdk-amd64/bin/java ...
Add: 25
Sub: 15
Mul: 100
Div: 4
Process finished with exit code 0
```

The navigation bar indicates the current file is `TestOperatorClient`. The bottom status bar shows the full path: `client > src > com > example > client > TestOperatorClient > main`.

251100680004

## Experiment 3: TrueNAS Installation and Setup

### Theory:

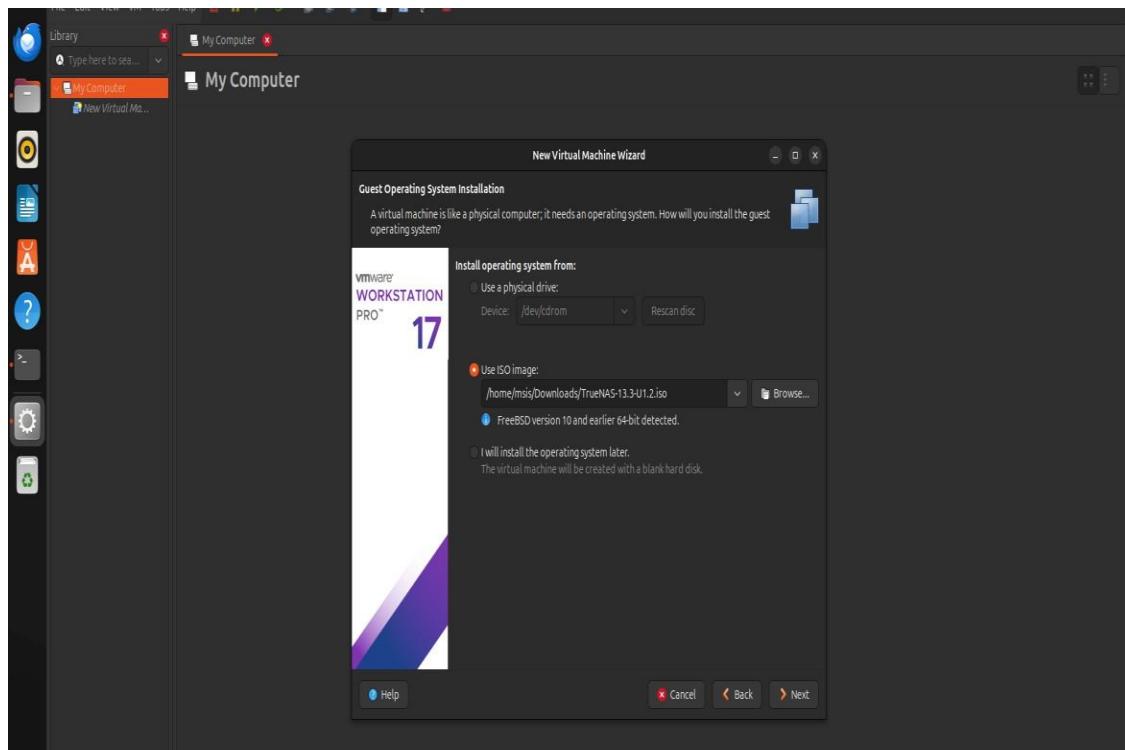
TrueNAS is an enterprise-grade NAS OS built on FreeBSD, offering ZFS storage, snapshots, sharing services (SMB, NFS, WebDAV), and a web dashboard.

## Procedure

The steps followed for this experiment along with screenshots are as follows:

### 1. Installing FREENAS / TRUENAS on VMWARE

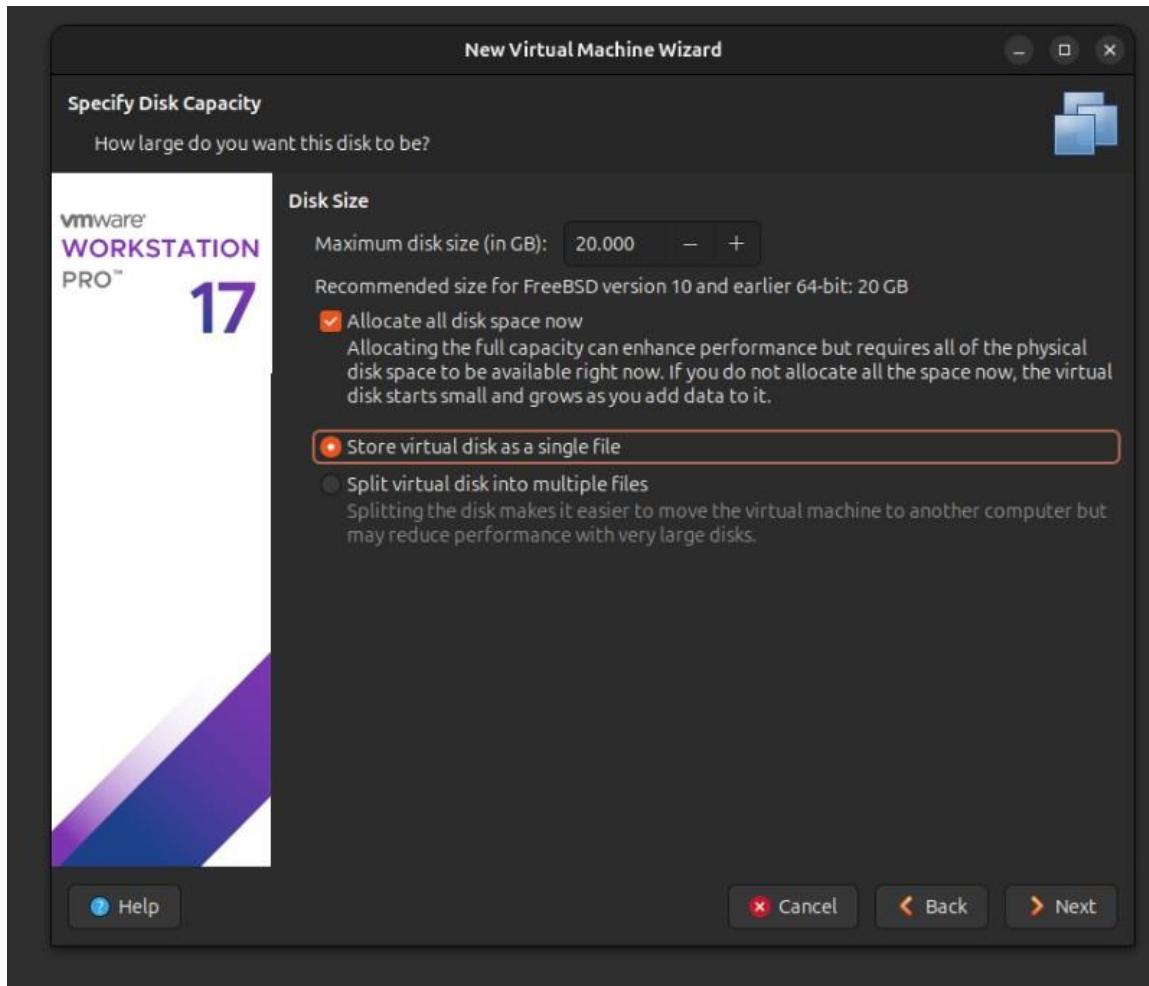
- Download the TrueNAS/FreeNAS installation ISO ([TrueNAS-13.3-U1.2.iso](#)).
- Start VMware and create a **new virtual machine**.
- Pick the **Typical** setup option.
- Choose **Use ISO image** and browse to the downloaded TrueNAS ISO file.



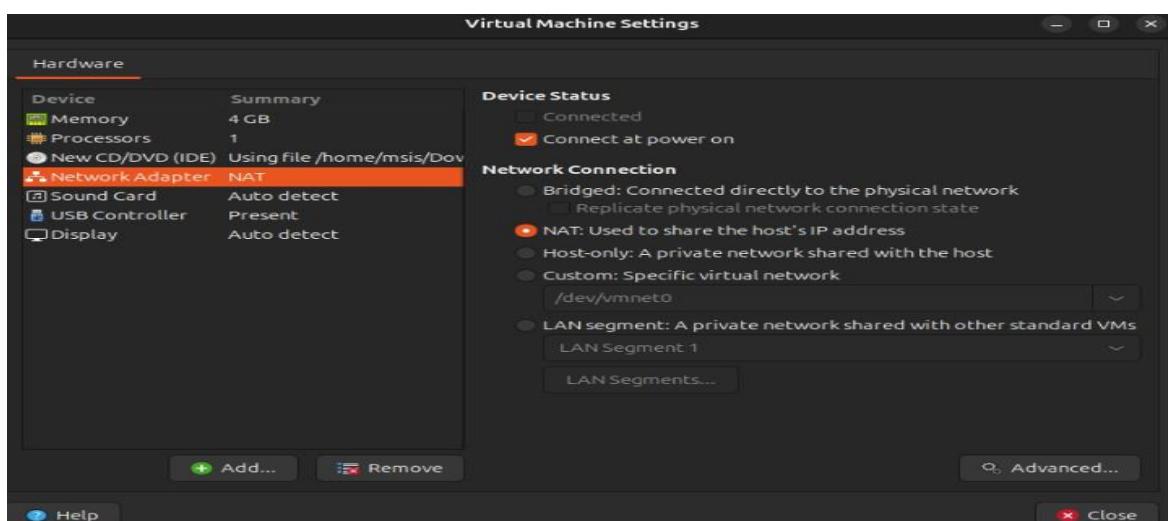
- Set up the virtual disk size according to your needs.
- Choose the option to **store the virtual disk as a single file**.

251100680004

- Click **Next**, then **Finish** to complete the VM setup.



- Adjust the hardware settings by opening **Customize Hardware**.
- Set the **Network Adapter** to **NAT** mode.
- Allocate **4 GB of memory** for the virtual machine.



251100680004

- When the virtual machine starts:
- Choose **Install/Upgrade** from the menu.
- Proceed with the default options by selecting **Yes/OK** when prompted.
- Pick the virtual disk you created as the installation target.



- Complete the installation and allow the VM to reboot.
- Set a root password **when asked**



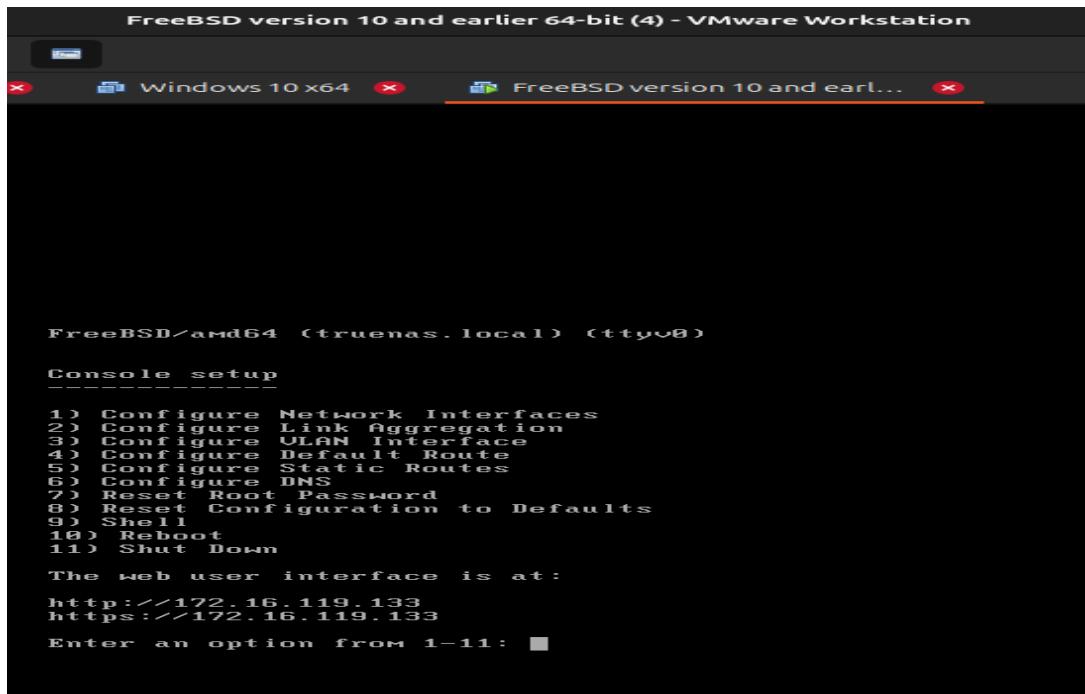
251100680004



251100680004

## 2. Accessing the TrueNAS Web Dashboard

- After the system reboots, TrueNAS will show a LAN IP address on the console. Open a browser and enter that IP in the address bar.
- Log in using: **Username:** root  
**Password:** the password you created during installation

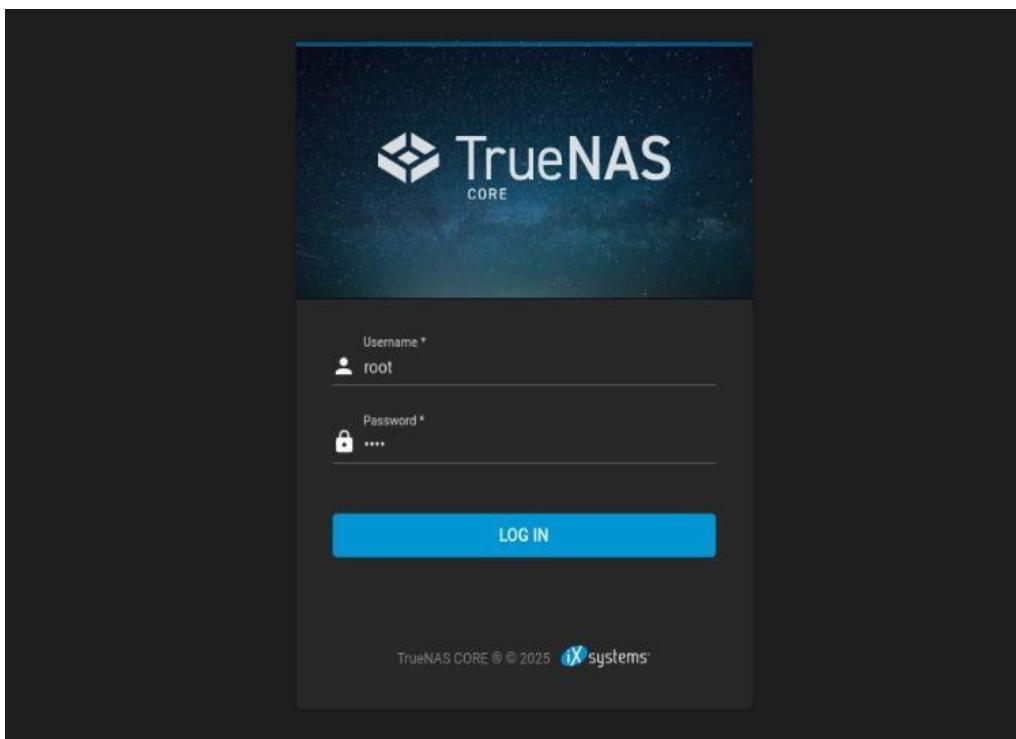


The screenshot shows a FreeBSD terminal window titled "FreeBSD version 10 and earlier 64-bit (4) - VMware Workstation". The window contains a "Console setup" menu with the following options:

```
FreeBSD/amd64 (truenas.local) (tty00)
Console setup
-----
1) Configure Network Interfaces
2) Configure Link Aggregation
3) Configure VLAN Interface
4) Configure Default Route
5) Configure Static Routes
6) Configure DNS
7) Reset Root Password
8) Reset Configuration to Defaults
9) Shell
10) Reboot
11) Shut Down

The web user interface is at:
http://172.16.119.133
https://172.16.119.133

Enter an option from 1-11: █
```



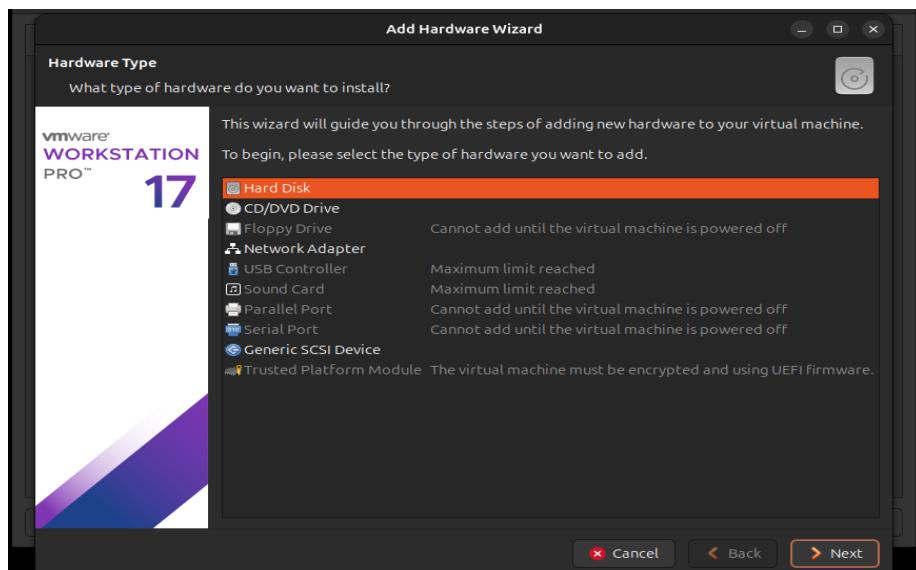
251100680004



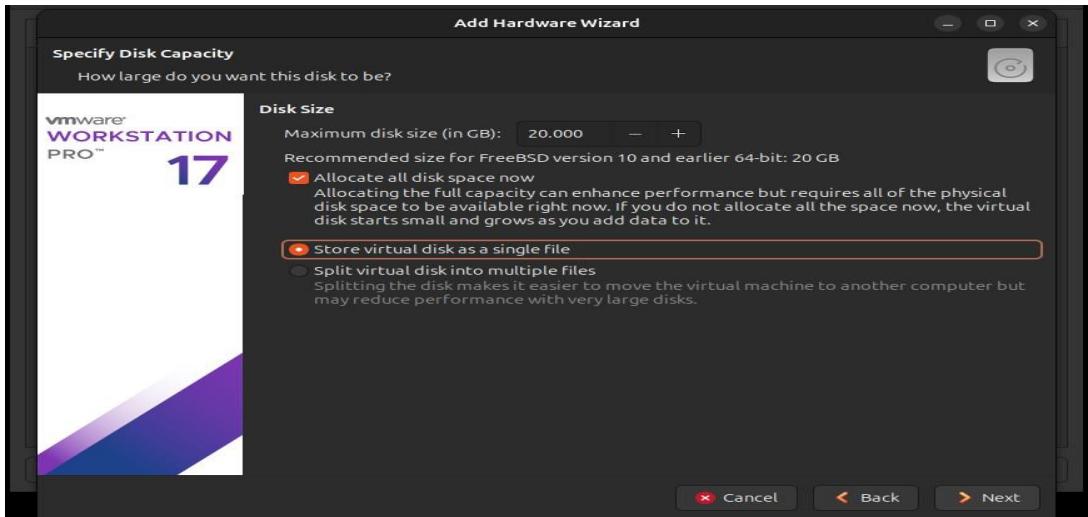
(It is recommended to shutdown truenas and restart the VM again and relogin to truenas)

### 3. Adding Two Virtual Hard Drives to the TrueNAS VM

- Open the VM settings by right-clicking the TrueNAS virtual machine and selecting **Settings**.
- Choose **Add → Hard Disk**, then select **SCSI** as the disk type.
- Create a **new virtual disk** and set its size as needed.
- Enable **Store virtual disk as a single file**.
- Click **Finish** to add the disk.

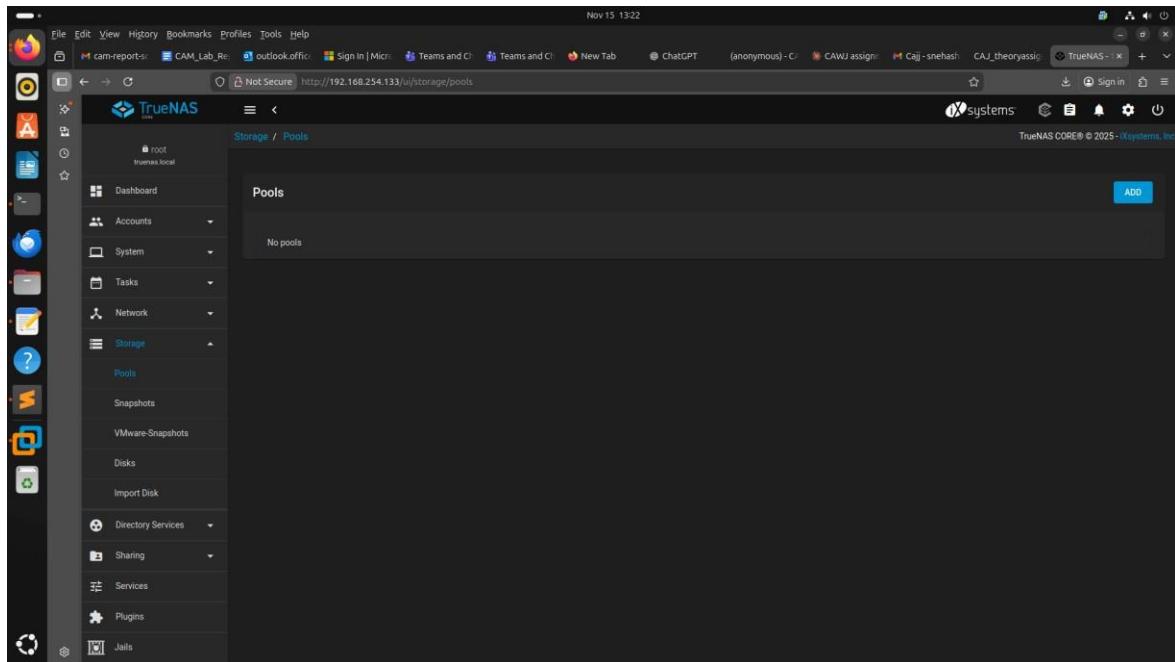


251100680004



#### 4. Creating Two Storage Pools in the TrueNAS Dashboard

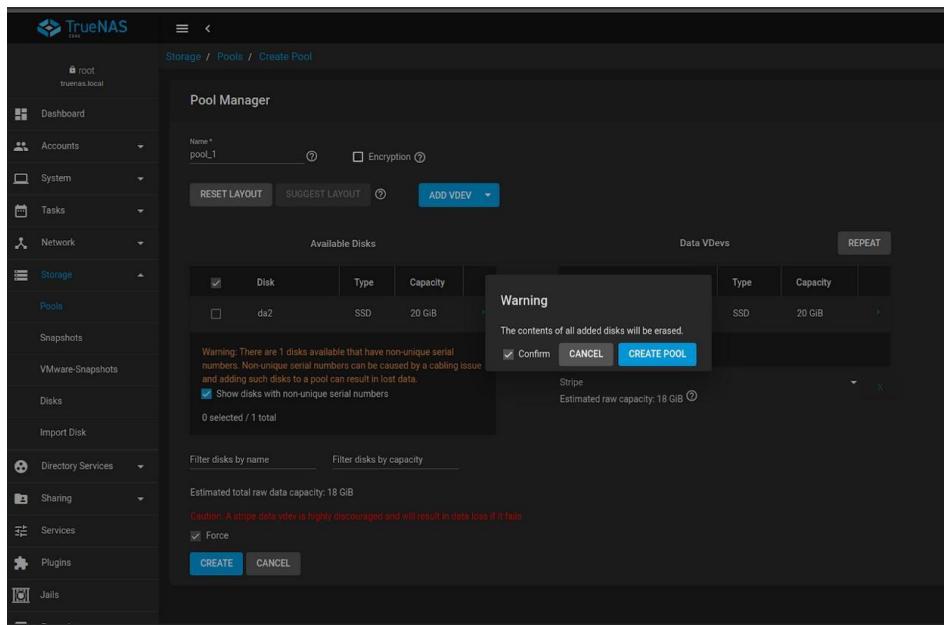
- After signing in to the TrueNAS web interface:
- Open the **Storage** menu.
- In the left sidebar, select **Storage → Pools**



- **Creating Pool 1**
- Click **Add → Create New Pool**

251100680004

- Follow these steps:
- Enter the pool name: **pool\_1**
- In the *Available Disks* section, the newly added virtual disks (e.g., **da1**, **da2**) will appear
- Select one disk and click **Add** to include it in the pool
- Click **Create**, then confirm to finish



- Click **Add** → **Create New Pool**
- Follow these steps:
- Enter the pool name: **pool\_2**
- Select the other virtual disk
- Add the disk to the pool configuration
- Click **Create**, then confirm to complete the pool setup

251100680004

Storage / Pools / Create Pool

Pool Manager

Name: pool\_2

Encryption:

RESET LAYOUT SUGGEST LAYOUT ADD VDEV

Available Disks

No data to display

0 selected / 0 total

Filter disks by name Filter disks by capacity

Data VDEvs

Disk Type Capacity

da2 SSD 29 GiB

Stripe Estimated raw capacity: 18 GiB

Creating a stripe data vdev is highly discouraged and will result in data loss if it fails.

Force

CREATE CANCEL

- Now TrueNAS has two **separate** pools, each using a separate virtual disk as shown below.

Storage / Pools

Pools

pool\_1 (System Dataset Pool) ONLINE ✓ | 6.89 MiB (0%) Used | 16.95 GiB Free

pool\_2 ONLINE ✓ | 384 KiB (0%) Used | 16.95 GiB Free

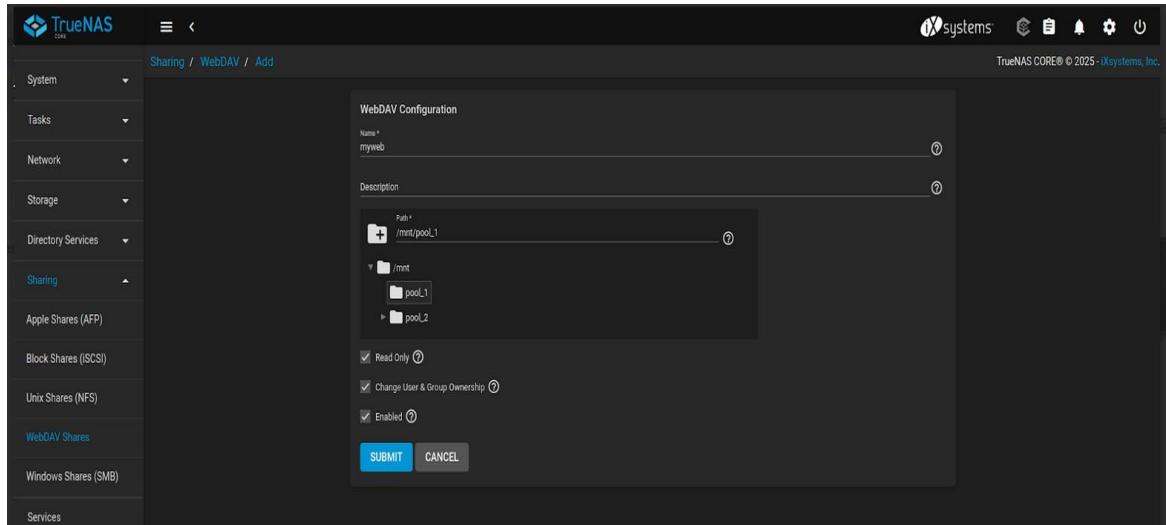
Add

Name	Type	Used	Available	Compression	Compression Ratio	Readonly	Dedup	Comments
pool_1	FILESYSTEM	6.89 MiB	16.95 GiB	lz4	18.97	false	OFF	
pool_2	FILESYSTEM	384 KiB	16.95 GiB	lz4	1.00	false	OFF	

## 5. WebDAV Configuration

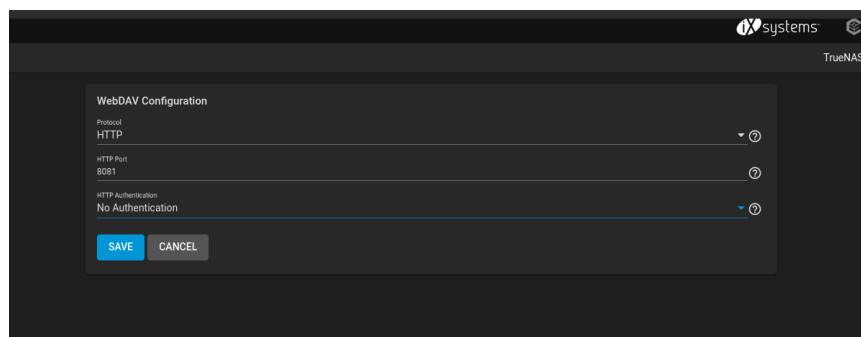
- Creating a WebDAV Share**
- Go to **Sharing → WebDAV → Add**
- Enter the required details:
  - Name:** myweb
  - Directory:** /mnt/pool\_1/myweb

251100680004



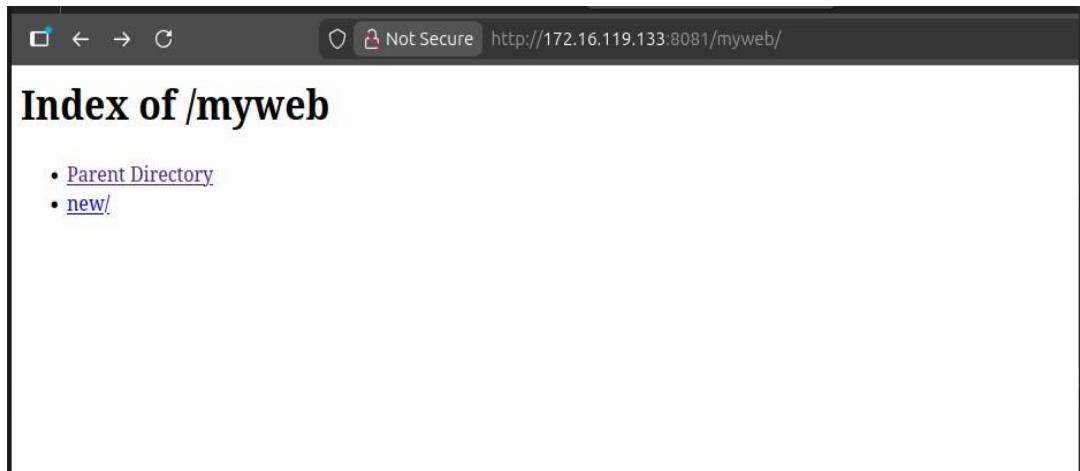
- **Enabling and Setting Up the WebDAV Service**

- Open the **Services** section
- Enable the **WebDAV** service and configure it as follows:
  - **Authentication:** None
  - **Protocol:** HTTP
  - **Port:** 8081



251100680004

- Now access **myweb** From Browser by pasting <truenasip>:8081/myweb



## 6. NFS Configuration and Mounting on TrueNAS

- Create an NFS Share
- Open Sharing → Unix Shares (NFS)
- Click Add
- Set the Path to:  
*/mnt/Pool2/LINUX*
- Save the share and enable the NFS service if asked

A screenshot of the TrueNAS web interface showing the NFS configuration screen. The top section is titled "Paths" and displays a tree view of storage paths:

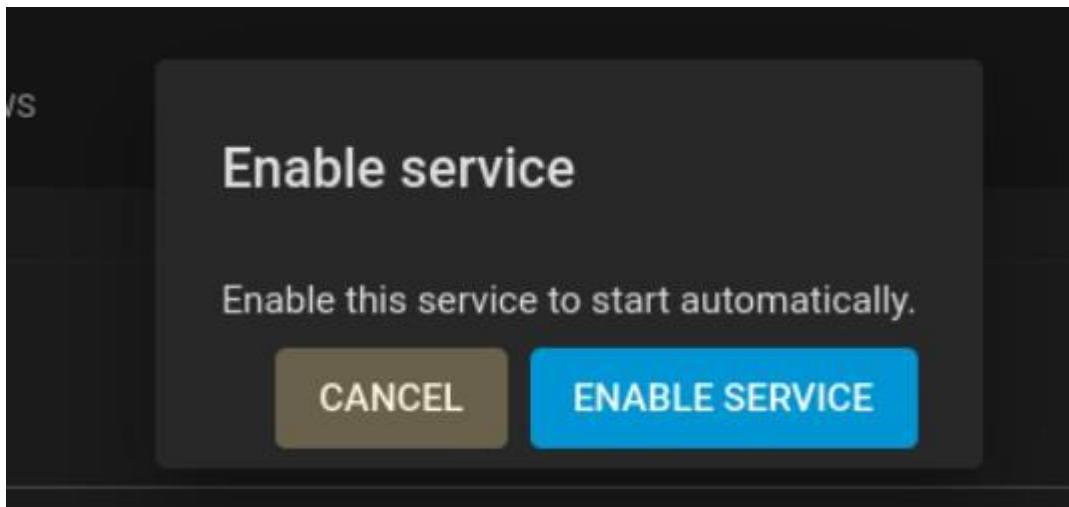
- /mnt/Pool2/LINUX (selected)
- /mnt/Pool1
- /mnt/WebDAV
- /mnt/Pool2
- LINUX (selected)
- WINDOWS

To the right of the tree view is a blue "ADD" button. The bottom section is titled "General Options" and contains the following settings:

Description	(?)
<input checked="" type="checkbox"/> All dirs (?)	(?)
<input type="checkbox"/> Quiet (?)	(?)
<input checked="" type="checkbox"/> Enabled (?)	(?)

At the bottom of the screen are three buttons: "SUBMIT" (blue), "CANCEL" (gray), and "ADVANCED OPTIONS" (blue).

251100680004



- Mount the NFS Share on Linux
- Run the following commands on your Terminal:

```
sudo mkdir /mnt/nfs_check
sudo mount -t nfs -o nfsvers=3 <truenas-ip>:/mnt/Pool2/LINUX /mnt/nfs_check
```

```
msis@msis:~$ sudo mkdir /mnt/nfs_check_
msis@msis:~$ sudo mount -t nfs -o nfsvers=3 172.16.196.131:/mnt/Pool2/LINUX /mnt/nfs_check_
msis@msis:~$
```

- Validation

Pool2								
Name	Type	Used	Available	Compression	Compression Ratio	Readonly	Dedup	Comments
Pool2	FILESYSTEM	912 KB	16.95 GiB	lz4	1.00	false	OFF	⋮
LINUX	FILESYSTEM	192 KB	16.95 GiB	Inherits (lz4)	1.00	false	OFF	⋮
nfstest	FILESYSTEM	96 KB	16.95 GiB	Inherits (lz4)	1.00	false	OFF	⋮

- Create a dataset inside /mnt/Pool2/LINUX in the TrueNAS GUI
- Check the folder /mnt/nfs\_check on the Linux system
- The newly created dataset should appear there, confirming the NFS share is working

251100680004

```
msis@msis:~$ cd /mnt/nfs_check_
msis@msis:/mnt/nfs_check_$ ls
nfstest
msis@msis:/mnt/nfs_check_$ █
```

## 7. Windows (SMB) Sharing Configuration

### Create a User in TrueNAS

- Go to **Accounts** → **Users** → **Add**
- Enter the required details:
  - **Username:** e.g., *winuser*
  - **Password:** choose a secure password
- Select the appropriate directory and set permissions
- Click **Submit** to create the user

The screenshot shows a user creation form titled 'Identification'. It includes fields for 'Full Name' (with placeholder 'winuser'), 'Username' (with placeholder 'winuser'), 'Email' (empty), 'Password' (with placeholder '\*\*\*\*\*'), and 'Confirm Password' (with placeholder '\*\*\*\*\*'). Each field has a question mark icon to its right, likely indicating help or validation rules.

251100680004

**Directories and Permissions**

Home Directory /mnt/Pool2/WINDOWS

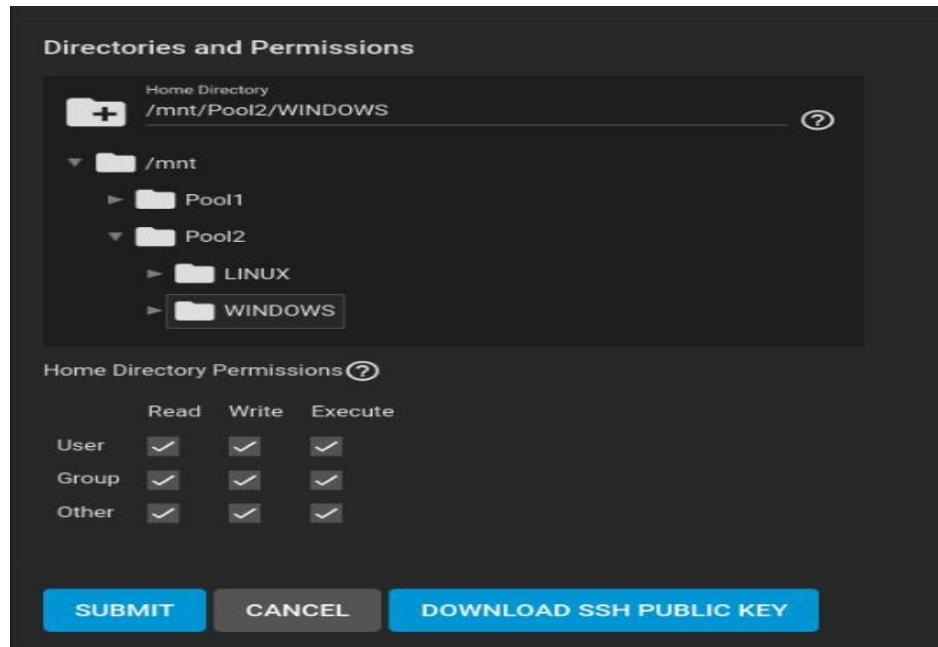
Tree View:

- /mnt
- Pool1
- Pool2
  - LINUX
  - WINDOWS

Home Directory Permissions

	Read	Write	Execute
User	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Group	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Other	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**SUBMIT** **CANCEL** **DOWNLOAD SSH PUBLIC KEY**



## Configure the SMB Service

- Open the **Services** section
- Enable **SMB1 support** if needed
- Under **Advanced Options**, set the newly created user as the **Guest User**
- Save the settings

**NetBIOS**

NetBIOS Name: truenas

NetBIOS Alias

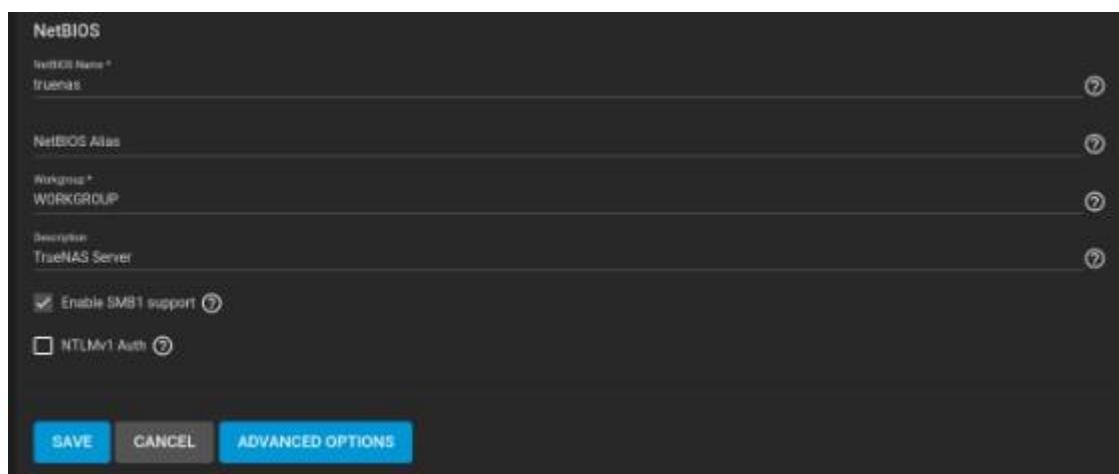
Workgroup: WORKGROUP

Description: TrueNAS Server

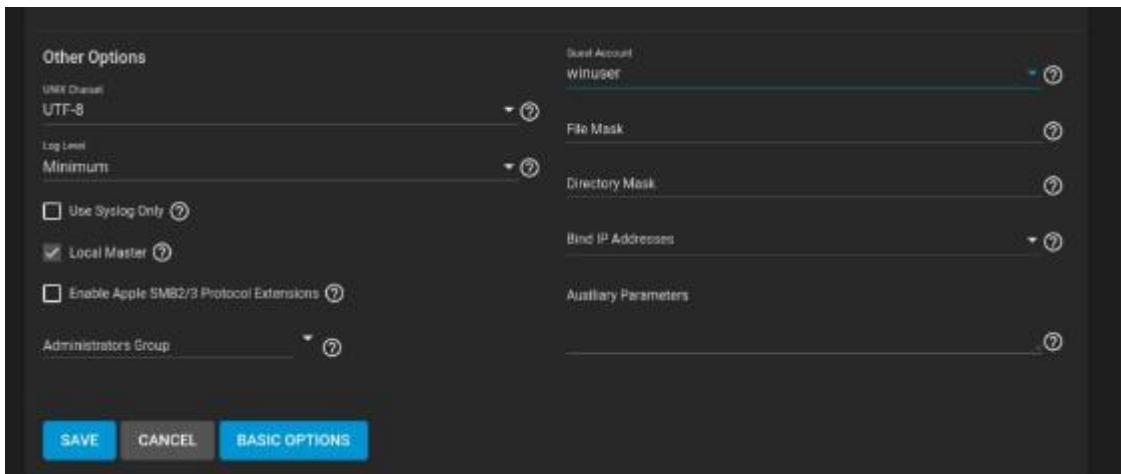
Enable SMB1 support

NTLMv1 Auth

**SAVE** **CANCEL** **ADVANCED OPTIONS**

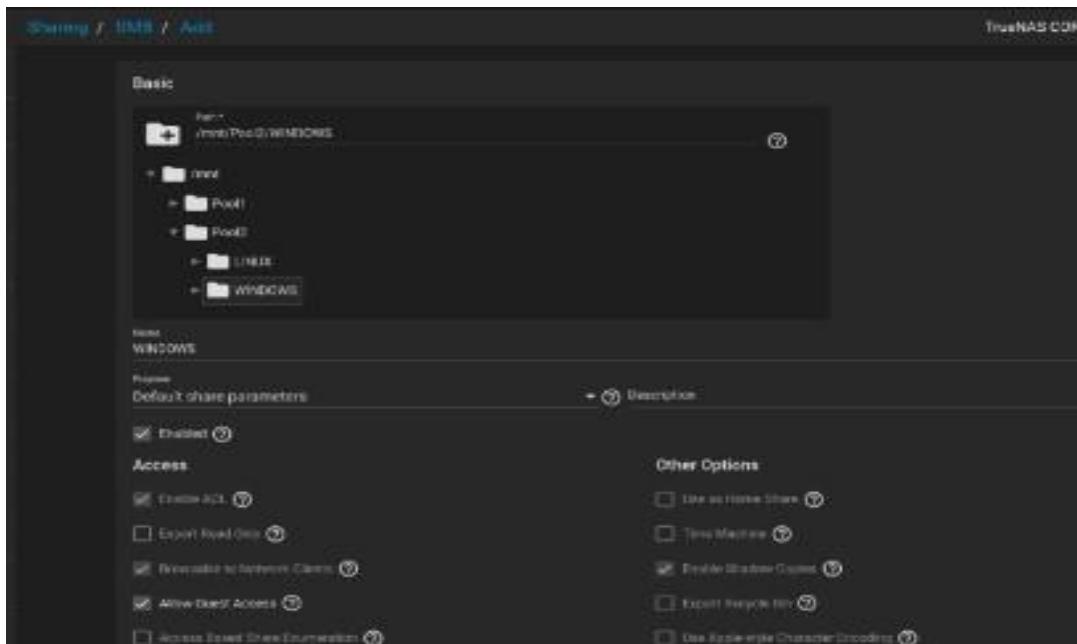


251100680004



## Create the SMB Share

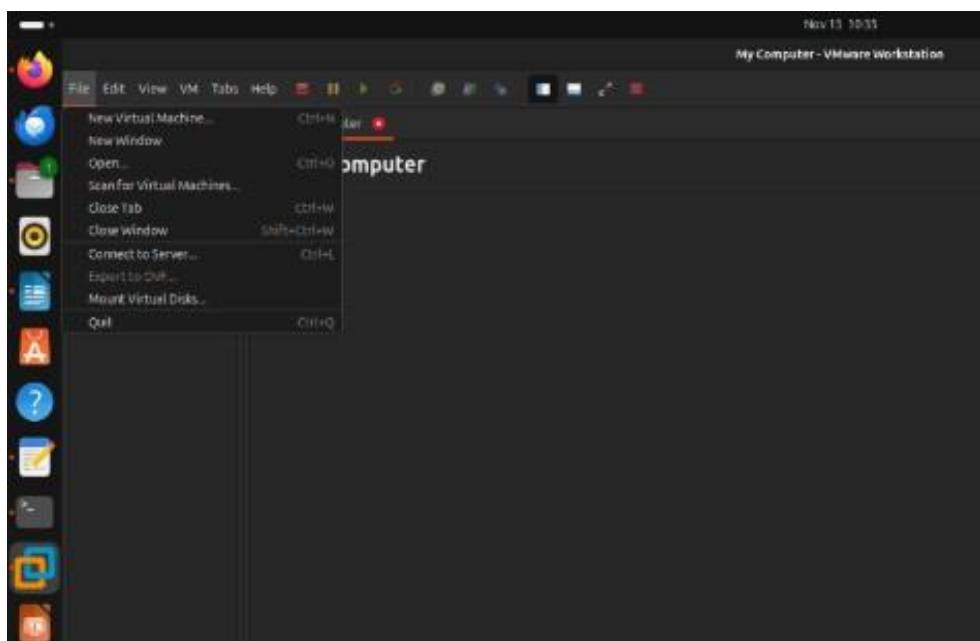
- Navigate to **Sharing → Windows Shares (SMB)**
- Click **Add**
- Set the **Path** to:  
/mnt/Pool2/WINDOWS
- Enable **Allow Guest Access**
- Click **Save** to create the share



251100680004

## Setting Up the Windows Virtual Machine

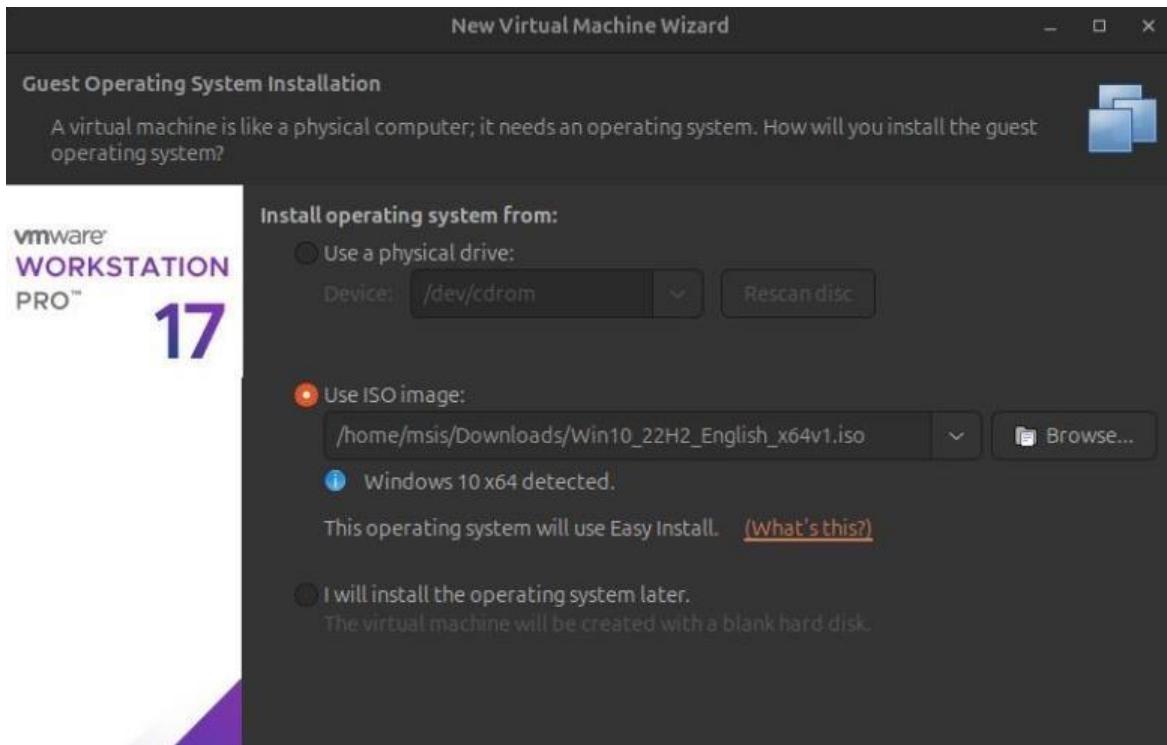
- Download Windows 10 ISO
- Open VMware Workstation



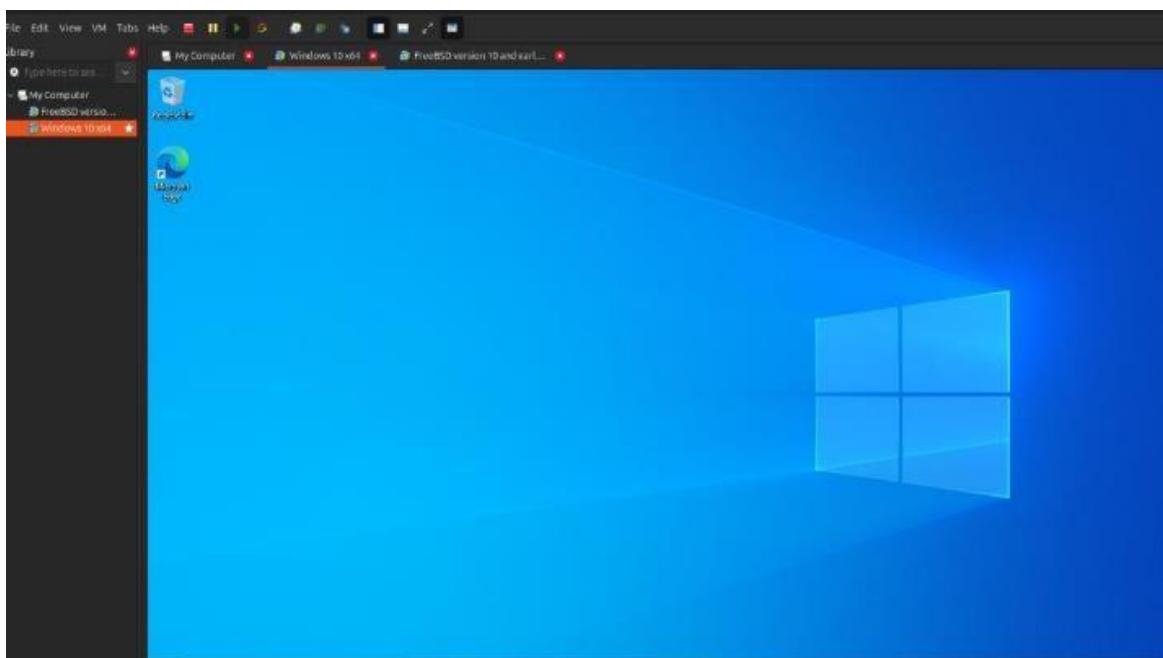
## Select the VM Configuration

- Choose **Typical (Recommended)**
- Under installation options, select **Use ISO image**
- Browse and choose the **Windows 10 ISO** file

251100680004



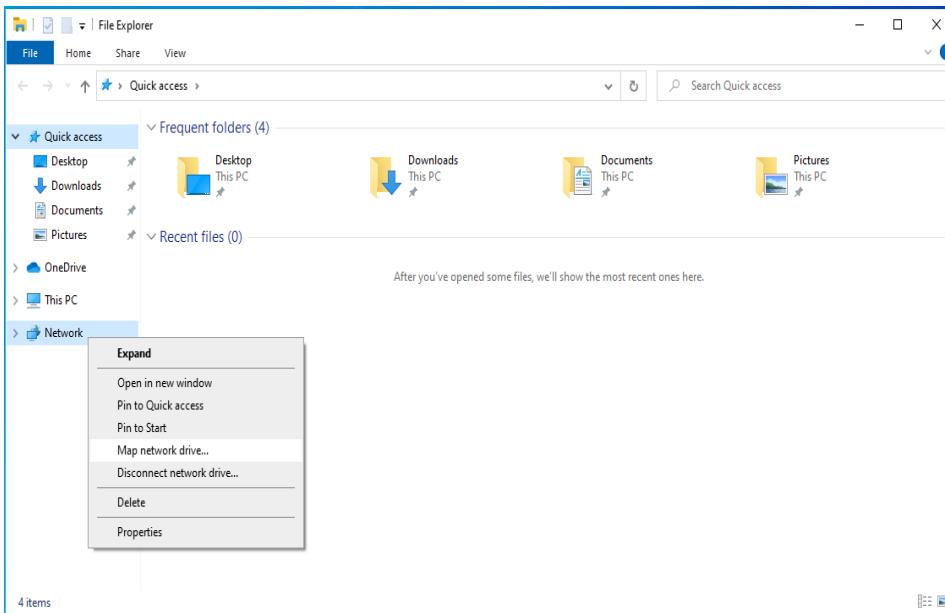
- **Configure the Virtual Disk**
- When setting up storage, select **Store virtual disk as a single file**
- Click **Finish** to complete the VM creation
- **Start the Windows VM**
- Power on the virtual machine to begin the Windows installation process



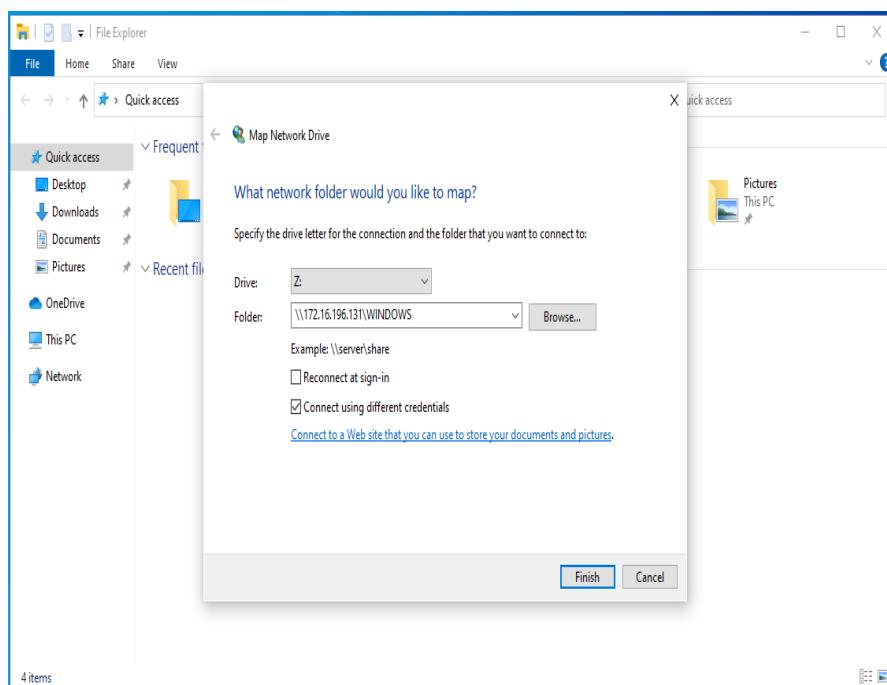
251100680004

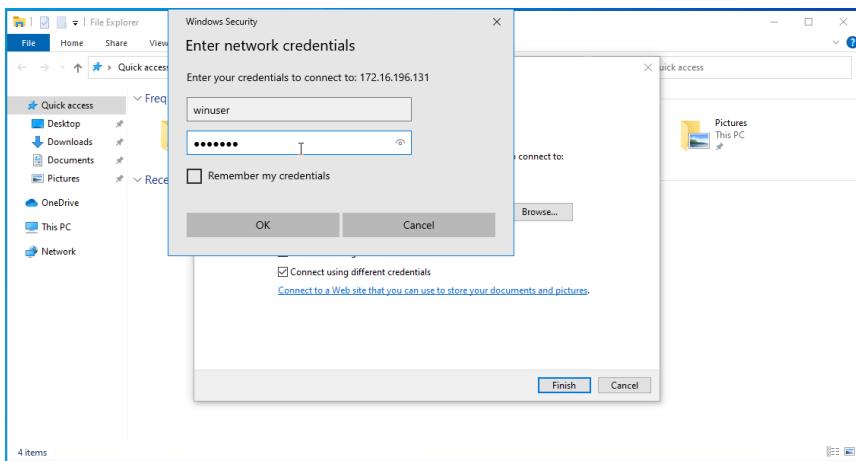
## Validating SMB Share from the Windows Machine

- Start the **Windows VM**
- Open **File Explorer**
- Right-click **Network** and select **Map network drive**
- In the path field, enter:  
\\<truenas-ip>\<SMB\_Share\_Name>
- Uncheck the first option and choose **Connect using different credentials**
- When asked to sign in, enter:
  - **Username:** winuser
  - **Password:** the password you created
- This confirms that the SMB share is accessible from Windows.



251100680004

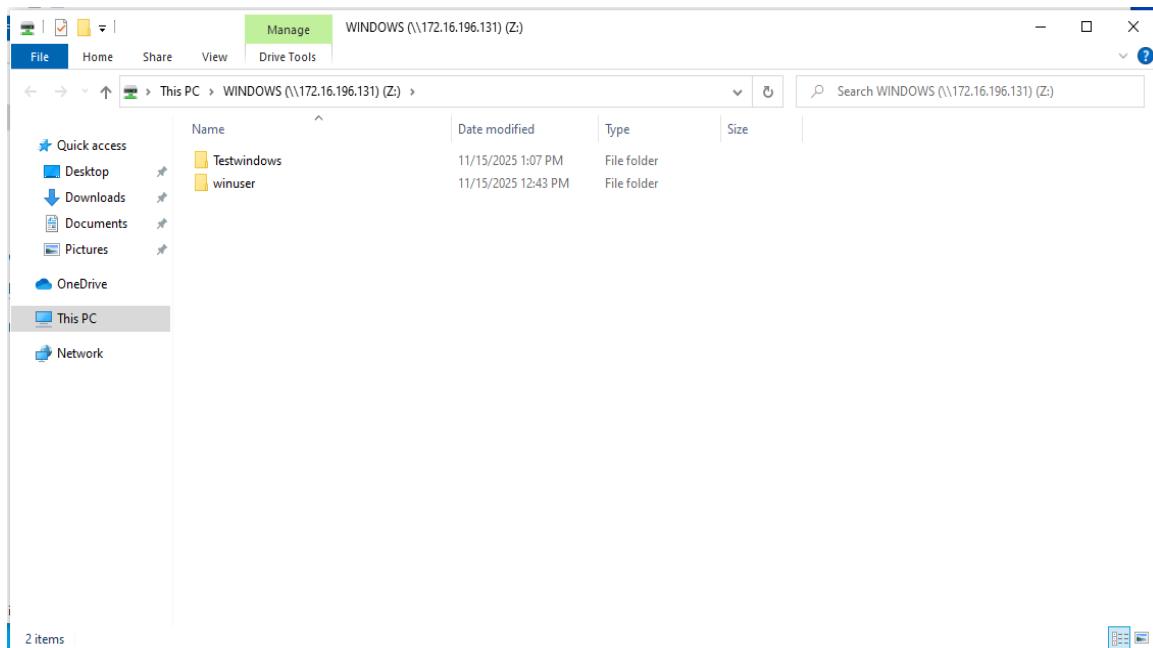




- Create a new dataset under the Pool in Truenas website

Name	Type	Used	Available	Compression	Compression Ratio	Readonly	Dedup	Comments
Pool2	FILESYSTEM	1.05 MiB	16.95 GiB	Iz4	1.00	false	OFF	
LINUX	FILESYSTEM	192 KiB	16.95 GiB	Inherits (Iz4)	1.00	false	OFF	
WINDOWS	FILESYSTEM	224 KiB	16.95 GiB	Inherits (Iz4)	1.00	false	OFF	
Testwindows	FILESYSTEM	96 KiB	16.95 GiB	Inherits (Iz4)	1.00	false	OFF	

- Check the created dataset is reflected in Windows VM . Hence mounting successful.



## **Additional Features Identified in TrueNAS**

During the TrueNAS exploration, an important safety-related feature was discovered:

### **Static IP Configuration (Network -> Interfaces -> Current Interface -> Edit)**

TrueNAS allows assigning a static IP address for improved network safety and reliability.

- Open Network -> Interfaces.
- Select active interface .
- Click **Edit** to modify IP settings.
- Specify a **Static IPv4 address** such as 192.168.254.133/24.
- Save and apply changes -> **test changes**
- Paste the changed truenas ip in the new tab in the browser
- We see that it opens a **new truenas tab** for you.

### **Benefits:**

- Improved security
- Stable LAN access for SMB/NFS/WebDAV
- Easier remote management
- Prevents accidental IP changes

251100680004

## Conclusion

All experiments successfully demonstrated modern cloud-oriented concepts:

- Secure communication using RSA keys.
- Distributed computing using Java Web Services.
- Centralized network storage using TrueNAS .