5/8/2025

Cloud and On-Premises Harmony

The Proxmox, Kubernetes, Ansible, and OCI Journey

Rinku Patel
Trios College

Table of Contents

Introduction

The hybrid cloud project integrates on-premises infrastructure with public cloud resources to create a scalable, efficient environment for managing and deploying applications. It combines virtualization, configuration management, and containerized technologies to ensure seamless interaction between private and public cloud components.

Objectives

The main objectives of this project are:

1. Configure a Hybrid Cloud Environment:
    - Set up both on-premises private cloud systems using Proxmox and Oracle Cloud Infrastructure (OCI) public cloud resources.
2. Automate Configuration Management:
    - Use Ansible to provision and manage systems uniformly across the hybrid cloud environment.
3. Deploy and Document Key Services:
    - Deploy containerized web applications and document the configurations, performance data, and architecture design.

Scope

This project includes five systems:

1. Ansible Control Node: Hosted on the Proxmox server to manage configurations.
2. Windows Server Core VM: Provides web services via IIS in the private cloud.
3. Linux Web Server Container: Hosts local web services using Apache.
4. Linux Kubernetes VM: Serves as a staging/testing area for deploying apps with Kubernetes (k3s).
5. OCI Linux VM: Provides public web services hosted in Oracle Cloud Infrastructure.

This comprehensive environment ensures smooth development workflows, robust backup strategies, and scalable production deployments.

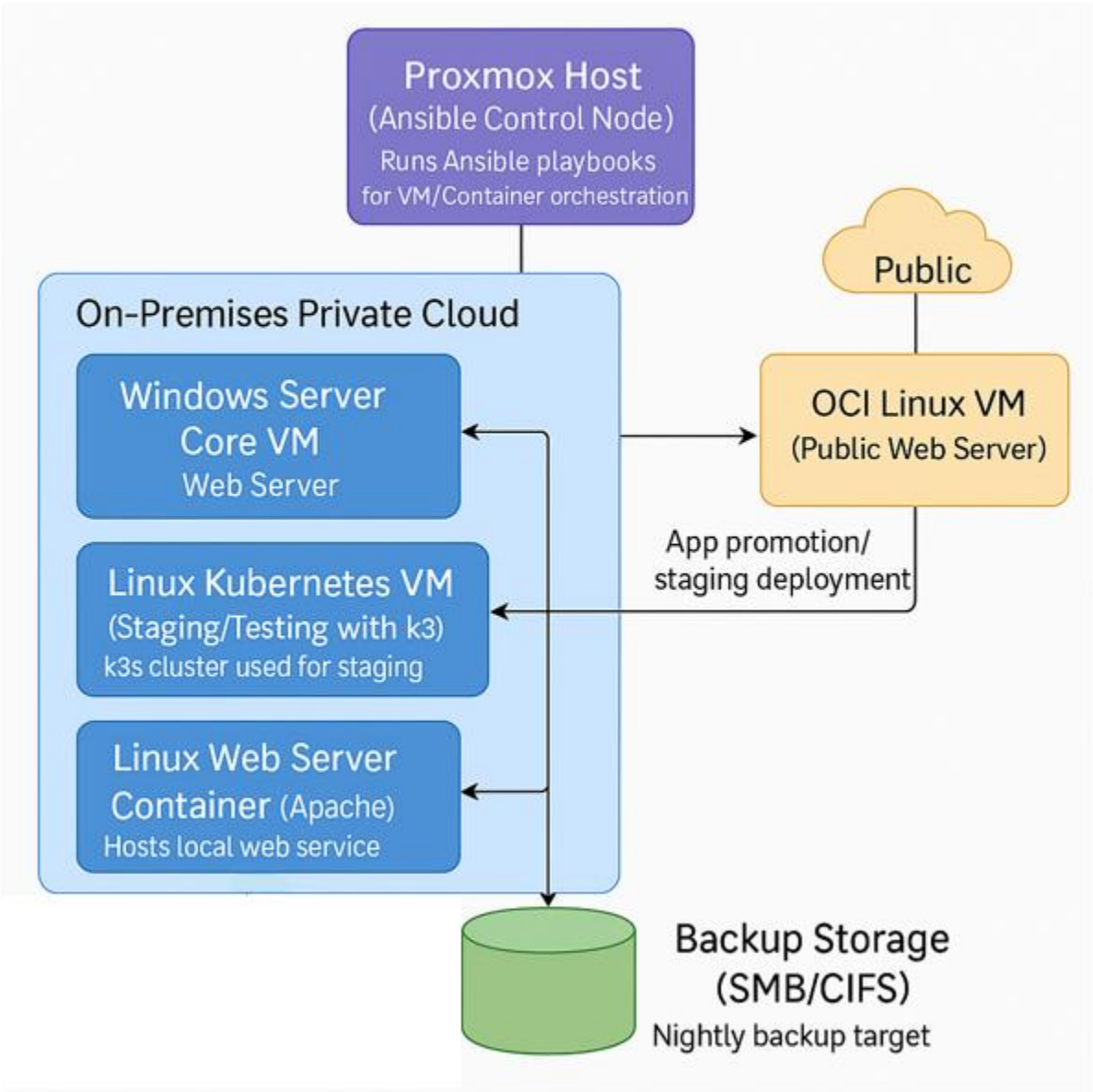
Project Overview

By leveraging technologies like Proxmox, Ansible, Kubernetes, and OCI, the project demonstrates how a hybrid cloud can be deployed to balance control and scalability while providing a practical solution for both private and public hosting needs. The final deliverable includes detailed documentation of the architecture, configuration, and performance metrics, presented in an easy-to-follow report format.

Hybrid Cloud Architecture Design

Introduction

This diagram represents the hybrid cloud architecture designed for the project, showcasing the interaction between on-premises systems and the Oracle Cloud Infrastructure (OCI). The architecture is structured to optimize functionality across both private and public environments, ensuring seamless integration for provisioning, configuration management, and automated deployment processes.



Hybrid Cloud Architecture Diagram

Architecture Components

The diagram illustrates the following components:

1. **Proxmox Host (Ansible Control Node)**:
    - Acts as the central node for virtualization and configuration management using Ansible.
    - Hosts the private cloud systems, including VMs and containers.
2. **Private Cloud Systems**:
    - **Windows Server Core VM**:
        - Configured with IIS to provide internal web services.
    - **Linux Web Server Container**:
        - Runs Apache for hosting local web content within the private cloud.
    - **Linux Kubernetes VM**:
        - Acts as a staging/testing environment for deploying a sample web app using Kubernetes (k3s).
3. **Public Cloud Component**:
    - **OCI Linux VM**:
        - Deployed in Oracle Cloud Infrastructure to host public-facing web services, allowing scalability and accessibility.
4. **Backup Storage**:
    - SMB/CIFS storage integrated with Proxmox to provide backups for all private cloud systems.
    - Ensures data security and availability in case of system failures.

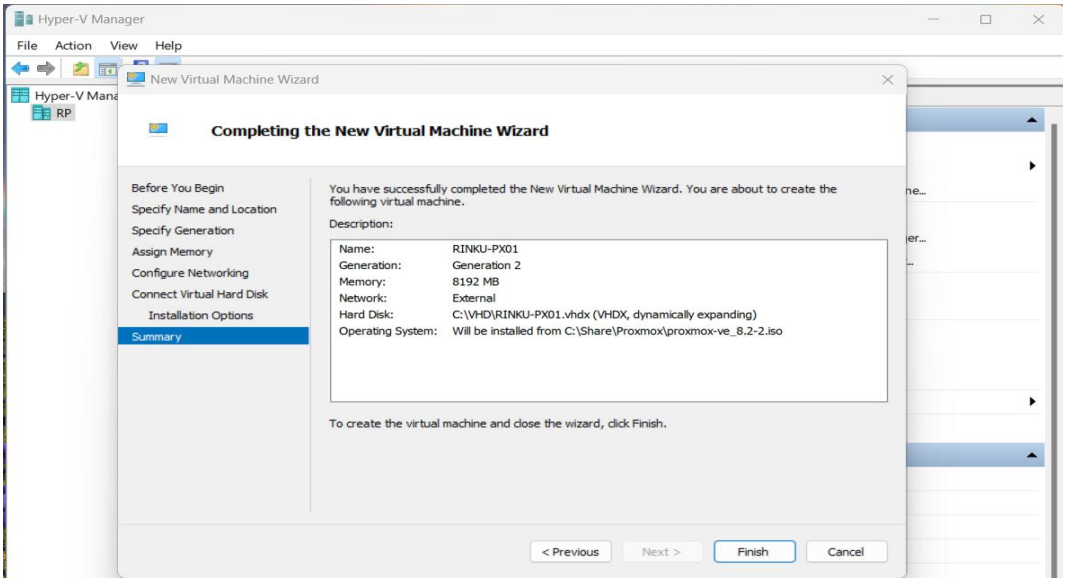
Systems Overview

System Name	Role/Description	IP Address	User	Memory	Core
RINKU-PVE01	Proxmox Host	192.168.2.230	Root	8 GB	8
RINKU-WEB01	Ubuntu 22.04 + Apache	192.168.2.233	Ansible	1 GB	2
RINKU-K8S01	Ubuntu 24.04 + K3s	192.168.2.236	Ansible	3 GB	2
RINKU-WIN01	Windows Server 2019 + IIS, Python, Chocolatey	192.168.2.232	Ansible	4 GB	2
RINKU-OCI01	Oracle Cloud Linux VM/ Canonical Ubuntu 20.04 Minimal	Public IP 129.153.49.245 Private IP 10.0.0.58	Ubuntu	1 GB	1
Backup_Storage	SMB/CIFS Backup Storage	192.168.2.65	Backup_user	Not Applicable	Not Applicable

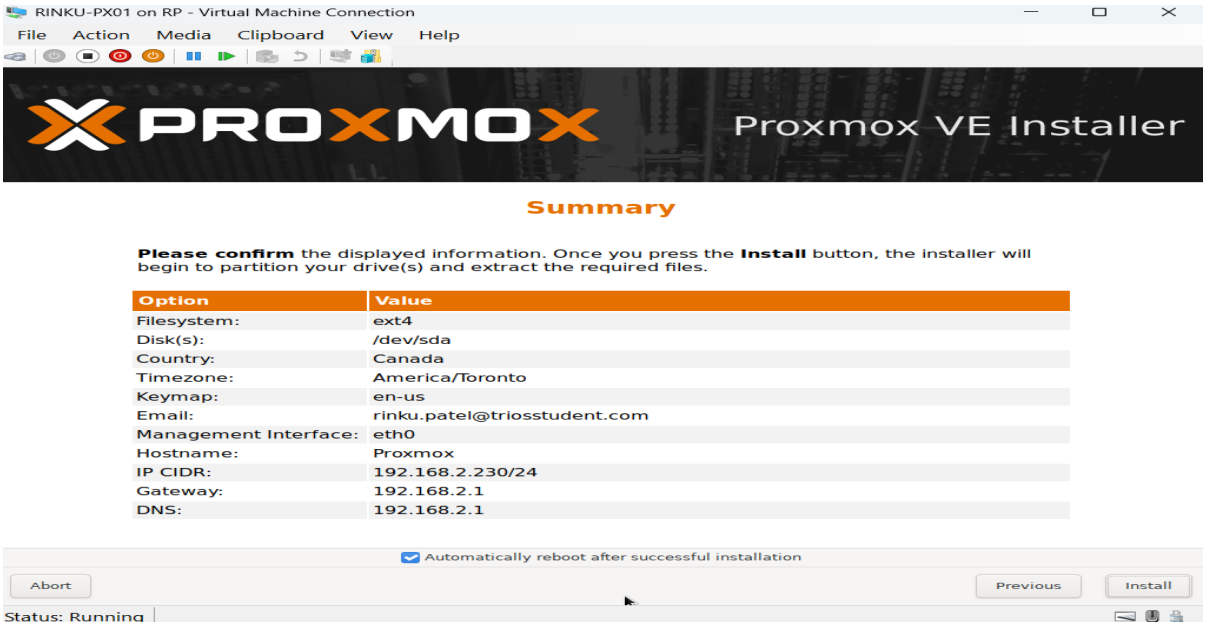
Proxmox Host (Ansible Control Node):

RINKU-PVE01	Proxmox Host/ Proxmox.rinkupatel.net	192.168.2.230	Root	8 GB Memory	8 Core

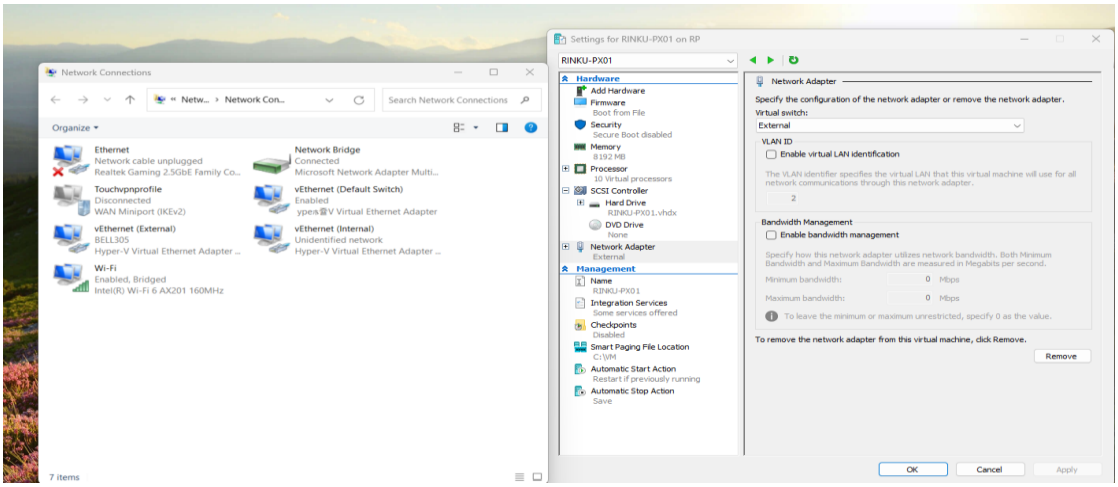
Proxmox Hyper V Installation Configuration



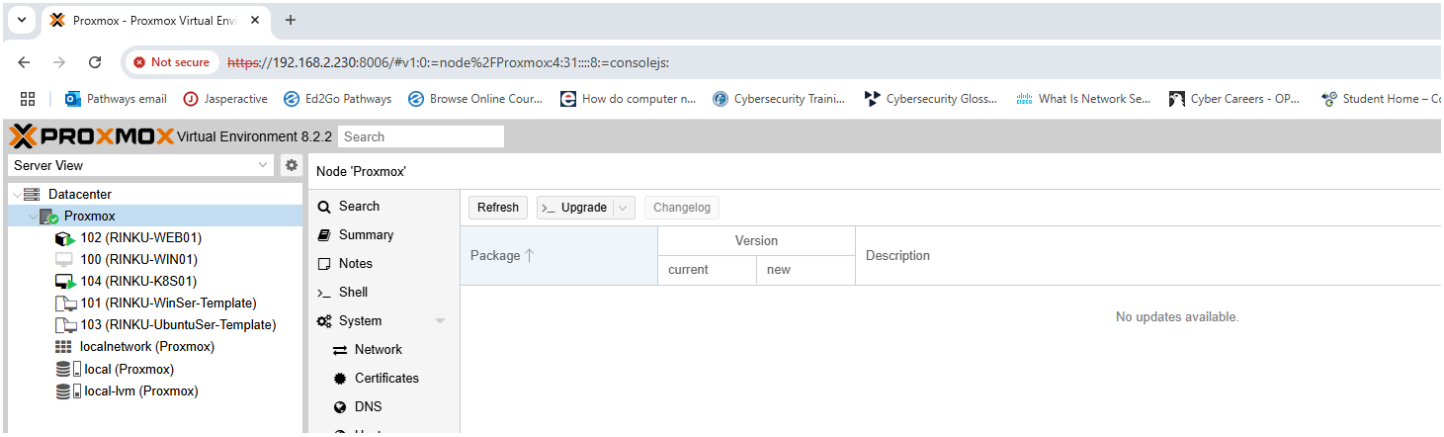
Proxmox Installation Configuration



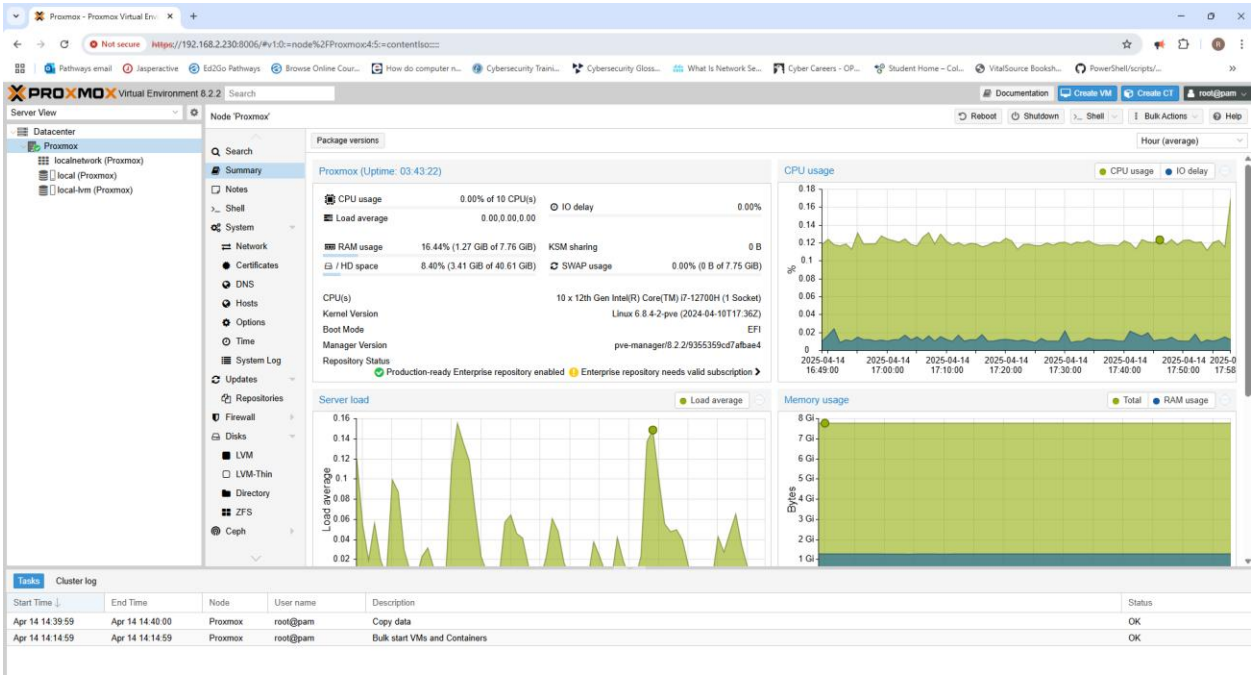
Proxmox Hyper V Network External Switch



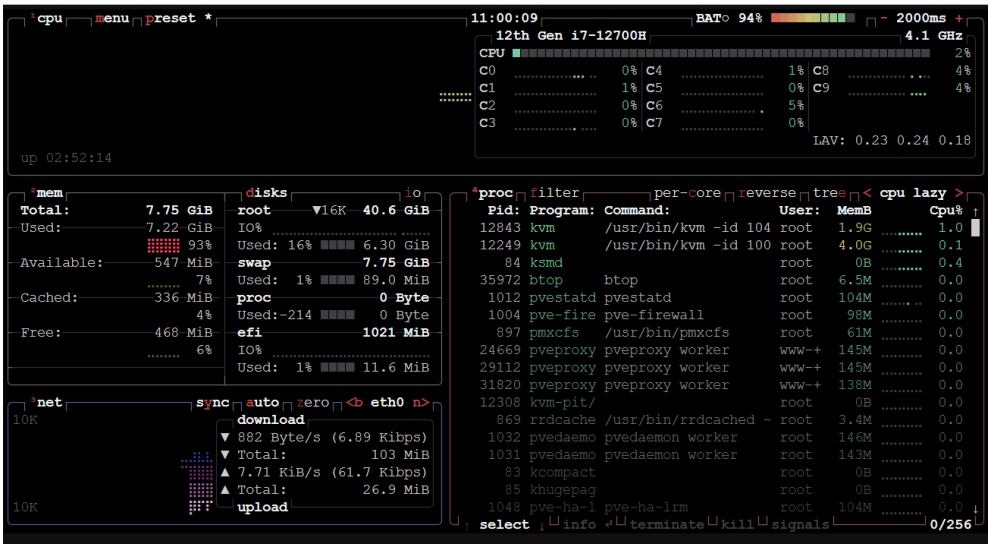
Proxmox Updates – All updates are performed before installing Ansible



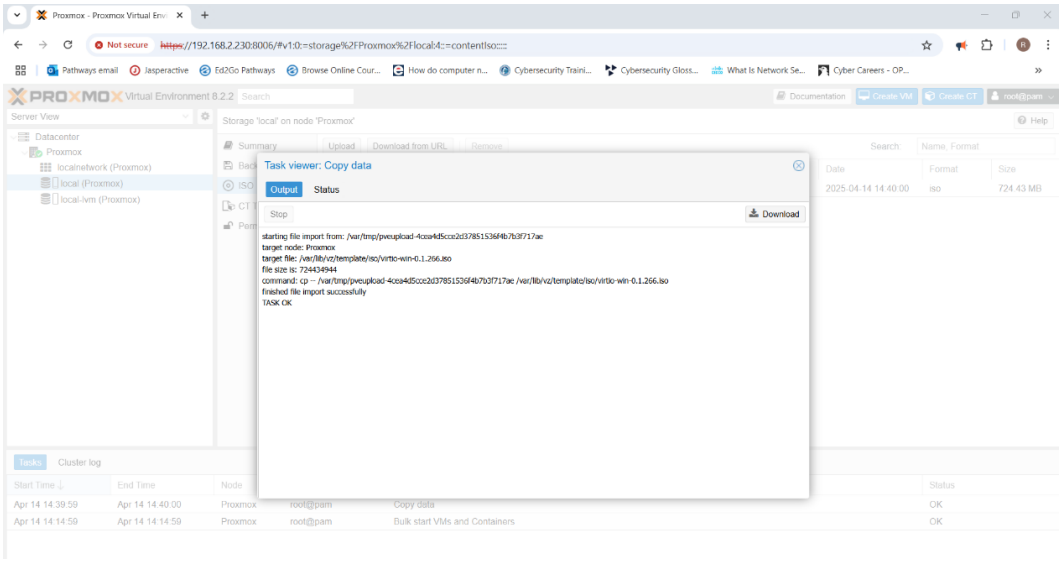
Proxmox Baseline



Proxmox Btop



Proxmox VirtIO Drivers Uploaded



Proxmox Install Ansible Success

```
Processing triggers for man-db (2.11.2-2) ...
root@Proxmox:~# ansible --version
ansible [core 2.14.18]
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.11.2 (main, Nov 30 2024, 21:22:50) [GCC 12.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
root@Proxmox:~#
```

Proxmox SSH Keygen Success

```
Linux Proxmox 6.8.4-2-pve #1 SMP PREEMPT_DYNAMIC PMX 6.8.4-2 (2024-04-10T17:36Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 16 13:58:36 EDT 2025 on tty1
root@Proxmox:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:w3nTaxRb2M99D0j86mdqQzwEcKj5xQyv5jGjghVqFds root@Proxmox
The key's randomart image is:
+---[RSA 3072]----+
|        .o.      |
|    .   o... o   |
|     + o = .= o  |
|    + E. .=o.* o.|
|   o . .Sooo= o =|
|  o .   Bo o+o .o|
| . o   + + .+.  .|
|  . . . .  oo o  |
|     .     .o=   |
+----[SHA256]-----+
root@Proxmox:~#
```

Proxmox Ubuntu CT SSH Login Success

```
Linux Proxmox 6.8.4-2-pve #1 SMP PREEMPT_DYNAMIC PMX 6.8.4-2 (2024-04-10T17:36Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 16 16:00:53 EDT 2025 on pts/0
root@Proxmox:~# ssh-copy-id ansible@192.168.2.233
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '192.168.2.233 (192.168.2.233)' can't be established.
ED25519 key fingerprint is SHA256:Ytbs8zxa6OjpAXTEMcIWXoz1szDQ+Axda27an7GN2ZM.
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: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@192.168.2.233's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ansible@192.168.2.233'"
and check to make sure that only the key(s) you wanted were added.

root@Proxmox:~# ssh ansible@192.168.2.233
Welcome to Ubuntu 22.04 LTS (GNU/Linux 6.8.4-2-pve x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Apr 16 19:58:30 2025 from 192.168.2.65
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ansible@RINKU-WEB01:~$
```

Proxmox Ubuntu Server SSH Login Success

```
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 16 16:08:52 EDT 2025 on pts/0
root@Proxmox:~# ssh-copy-id ansible@192.168.2.236
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '192.168.2.236 (192.168.2.236)' can't be established.
ED25519 key fingerprint is SHA256:DsxmK23DsLmNV08S8HmfLzyjmOhbaj3zVHq7KgY4LlY.
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: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@192.168.2.236's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ansible@192.168.2.236'"
and check to make sure that only the key(s) you wanted were added.

root@Proxmox:~# ssh ansible@192.168.2.236
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-57-generic x86_64)

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

 System information as of Thu Apr 17 02:20:01 AM UTC 2025

  System load:  0.0               Processes:             109
  Usage of /:   36.6% of 14.66GB  Users logged in:       1
  Memory usage: 6%                IPv4 address for ens18: 192.168.2.236
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Wed Apr 16 18:37:57 2025 from 192.168.2.65
ansible@RINKU-K8S01:~$
```

Proxmox Ubuntu Cloud SSH Login Success

```
root@Proxmox:~# ssh ubuntu@129.153.49.245
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1077-oracle x86_64)

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

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Thu May  8 12:40:12 2025 from 174.95.195.13
ubuntu@rinku-oci01:~$
```

Proxmox Ping All Success

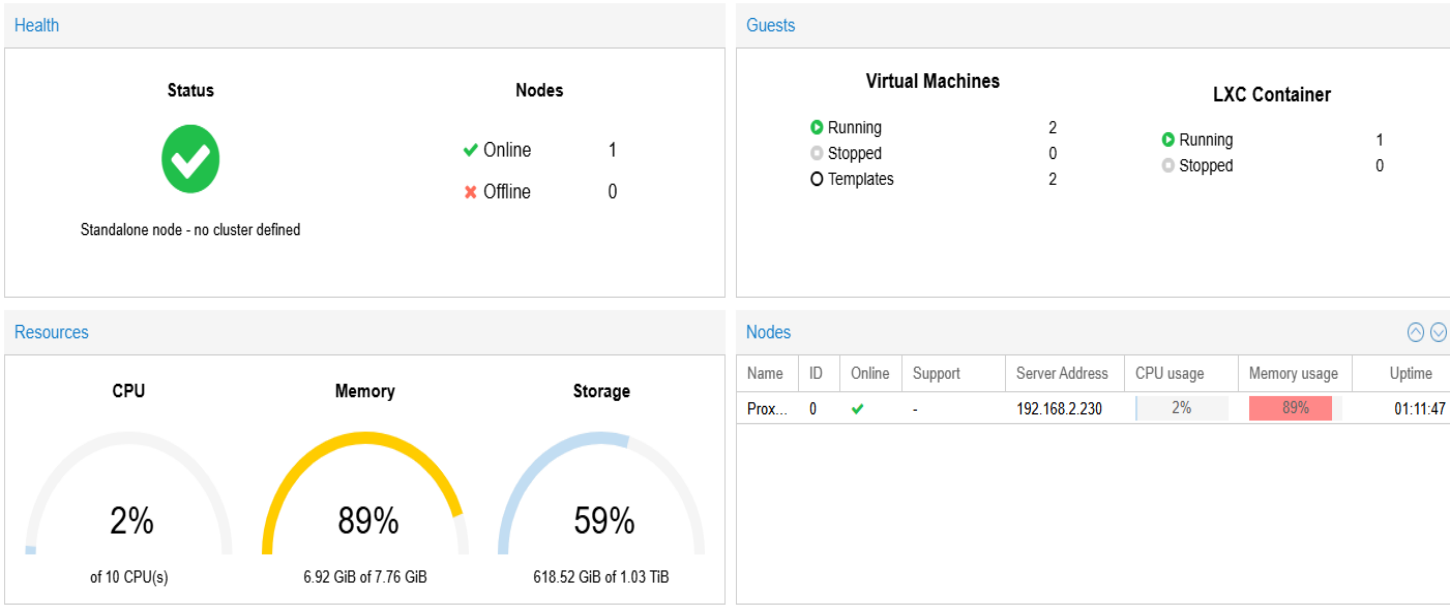
```
permitted by applicable law.
Last login: Thu May  8 09:09:57 EDT 2025 on pts/0
root@Proxmox:~# ansible ubuntu -m ping
192.168.2.233 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
192.168.2.236 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
129.153.49.245 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
root@Proxmox:~# ansible windows -m win_ping
192.168.2.232 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
root@Proxmox:~#
```

Proxmox Configurations Files and Playbooks – Ansible.cfg, hosts.inventory, Depolyapache.yml, Managewindows.yml

```
root@Proxmox:~# cat ansible.cfg
[defaults]
inventory = hosts.inventory
remote_user = ansible
root@Proxmox:~# cat hosts.inventory
[ubuntuweb]
192.168.2.233

[ubuntuk8s]
192.168.2.236

[ubuntuoci]
129.153.49.245 ansible_user=ubuntu

[ubuntu:children]
ubuntuweb
ubuntuk8s
ubuntuoci

[windows]
192.168.2.232

[windows:vars]
ansible_password=Secret555
ansible_connection=winrm
ansible_winrm_port=5985
ansible_winrm_transport=ntlm
ansible_winrm_scheme=http
ansible_winrm_kerberos_delegation=false

[ubuntu:vars]
ansible_sudo_pass=Secret555
root@Proxmox:~#
```

```
root@Proxmox:~# ls
ansible.cfg  deployapache.yml  hosts.inventory  managewindows.yml
root@Proxmox:~# cat deployapache.yml
---
- name: Deploy Apache Web Server on Ubuntu hosts
  hosts: ubuntu
  become: yes
  become_method: sudo

  tasks:
  - name: Install Apache
    apt:
      name: apache2
      state: present

  - name: Ensure Apache is running and started at boot time
    service:
      name: apache2
      state: started
      enabled: yes

root@Proxmox:~# cat managewindows.yml
---
- name: Install web server
  hosts: windows

  tasks:
  - name: Install the IIS web server
    win_feature:
      state: present
      name: Web-Server
```

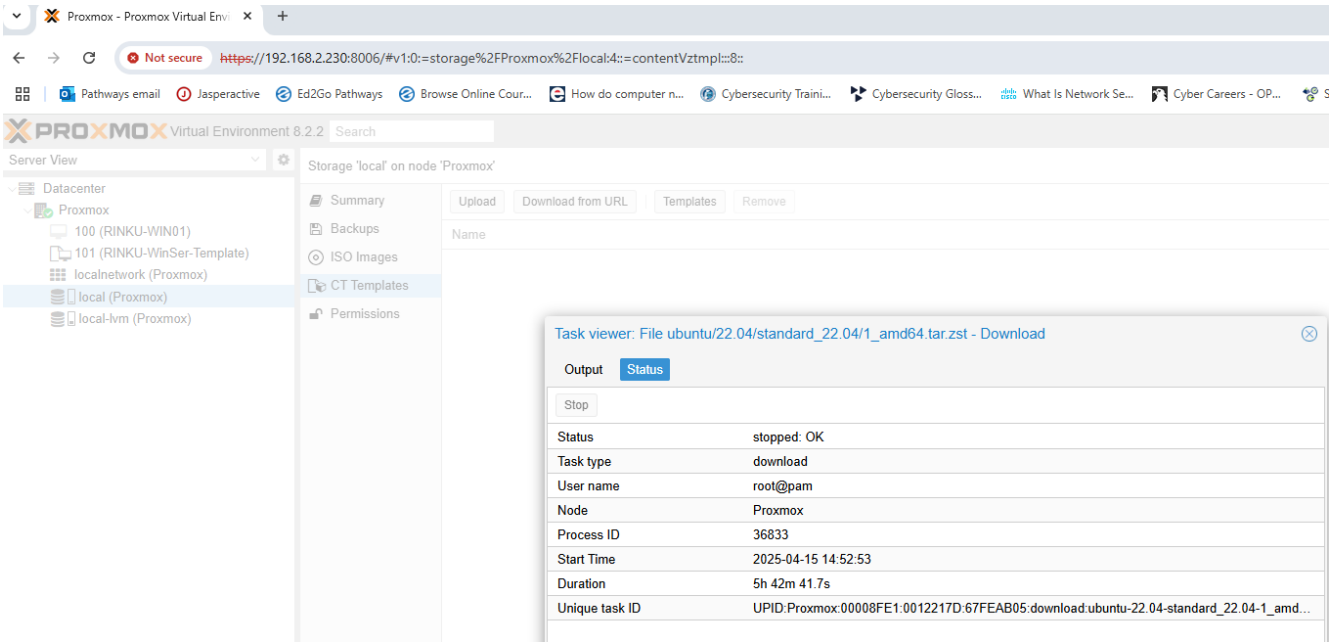
Proxmox Summary



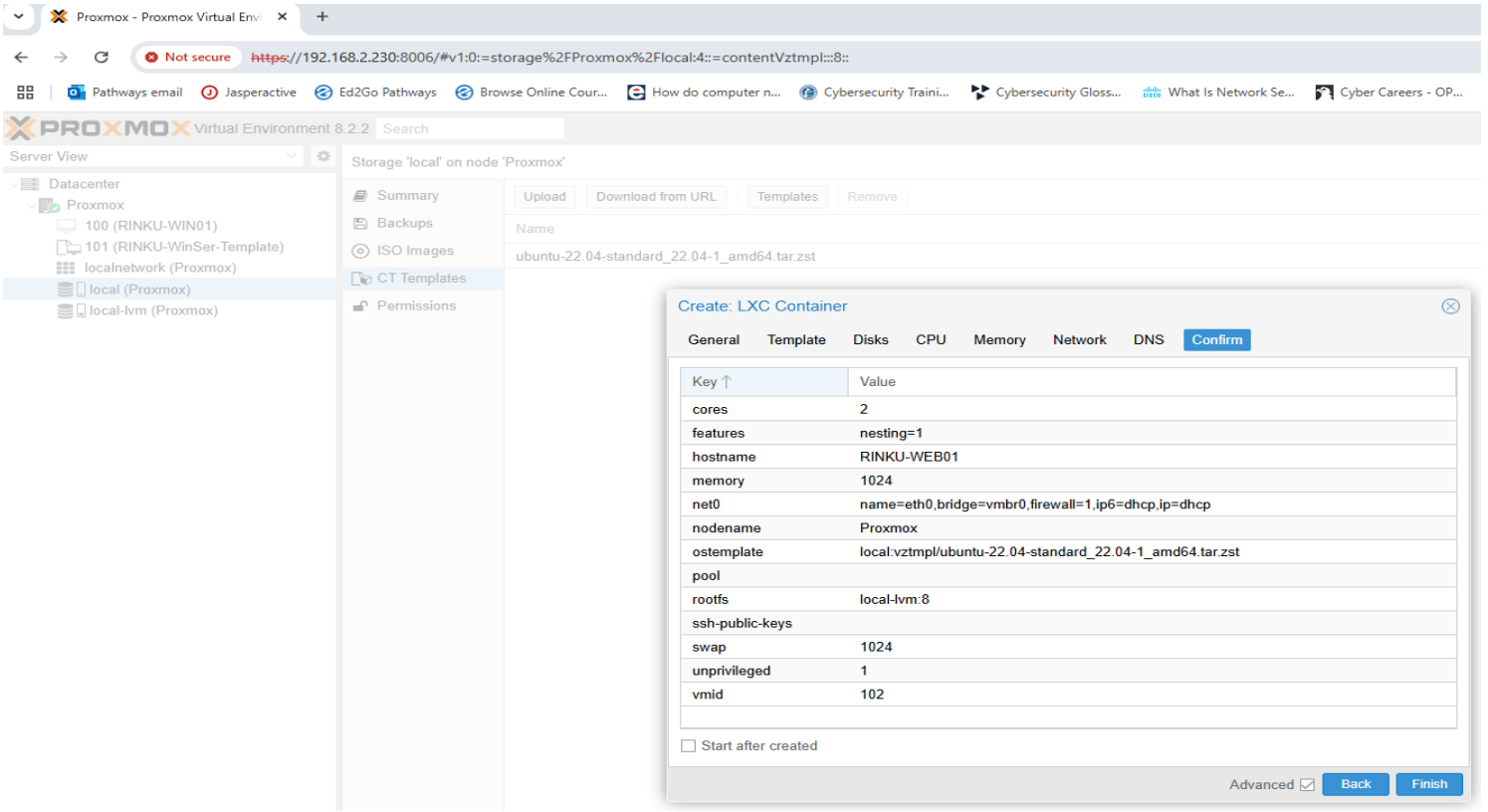
Linux Web Server Container

RINKU-WEB01	Ubuntu 22.04 + Apache	192.168.2.233	Ansible	1 GB Memory	2 core

Ubuntu-22.04-Standard Template Download



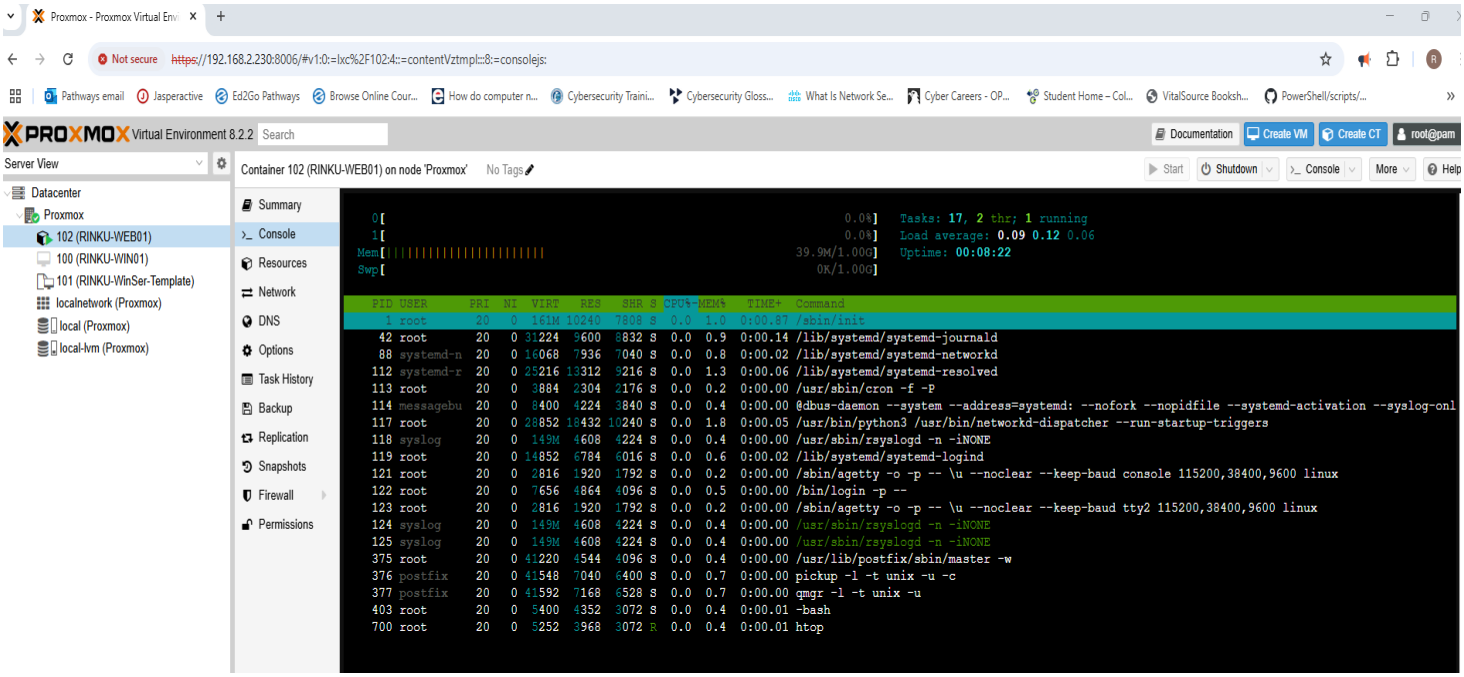
Ubuntu CT Installation Configuration



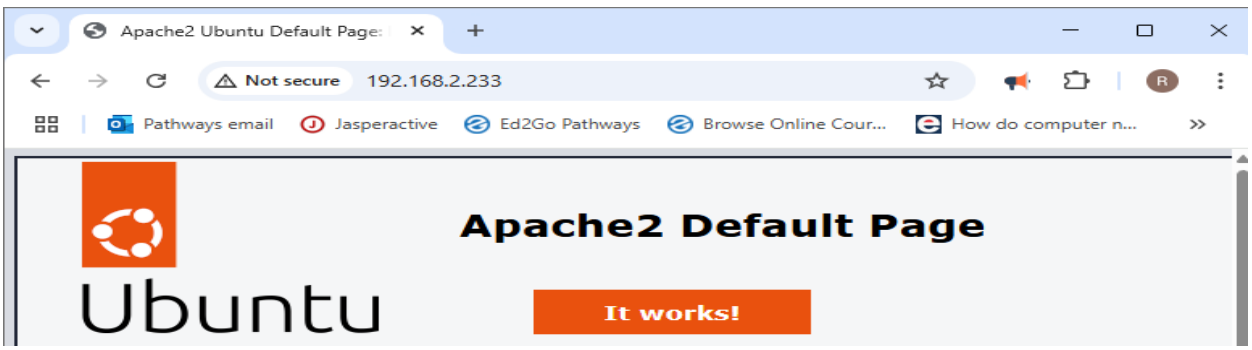
Ubuntu CT Updates

```
root@RINKU-WEB01:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0@if16: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether bc:24:11:07:15:27 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.2.233/24 metric 1024 brd 192.168.2.255 scope global dynamic eth0
       valid_lft 259085sec preferred_lft 259085sec
    inet6 fe80::be24:11ff:fe07:1527/64 scope link
       valid_lft forever preferred_lft forever
root@RINKU-WEB01:~# apt update
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/main Translation-en [510 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 c-n-f Metadata [30.3 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/restricted Translation-en [18.6 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 c-n-f Metadata [488 B]
Get:8 http://archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Reading package lists... Done
E: Release file for http://archive.ubuntu.com/ubuntu/dists/jammy-updates/InRelease is not valid yet (invalid for another 5h 16min 2s). Updates for this repositc
ry will not be applied.
E: Release file for http://archive.ubuntu.com/ubuntu/dists/jammy-security/InRelease is not valid yet (invalid for another 5h 14min 15s). Updates for this reposi
tory will not be applied.
root@RINKU-WEB01:~# apt upgrade
Reading package lists... Done
Building dependency tree... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@RINKU-WEB01:~#
```

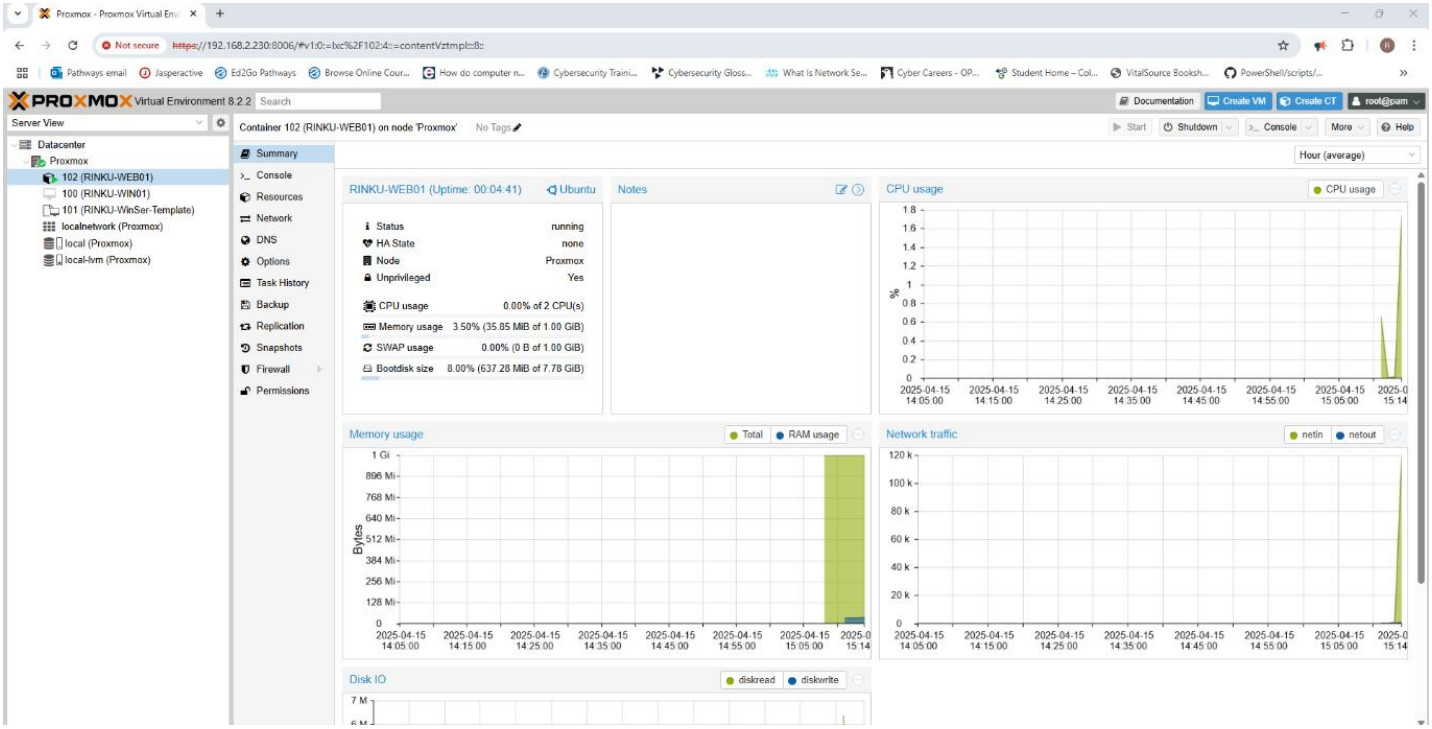
Ubuntu CT Htop



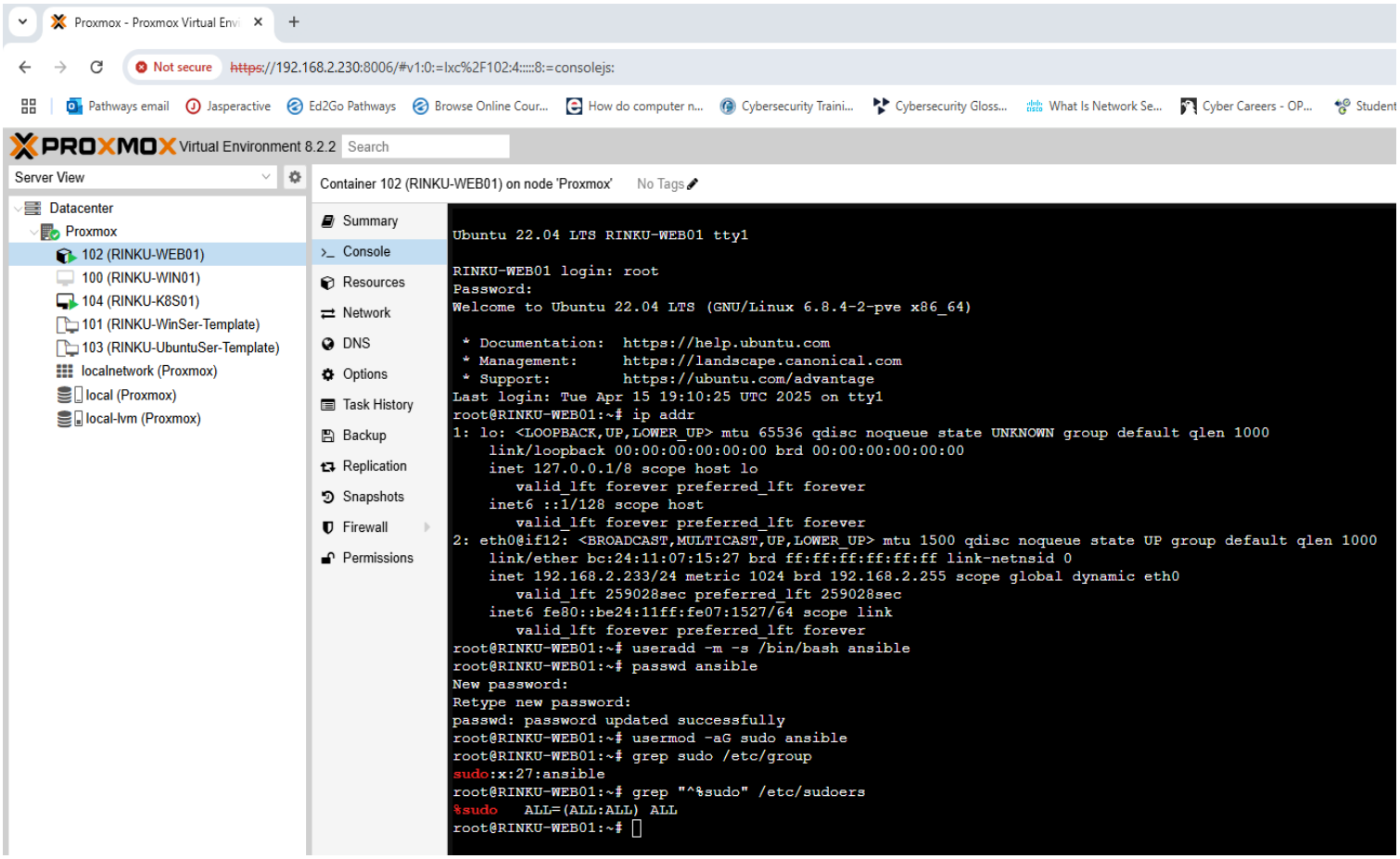
Ubuntu CT Apache2



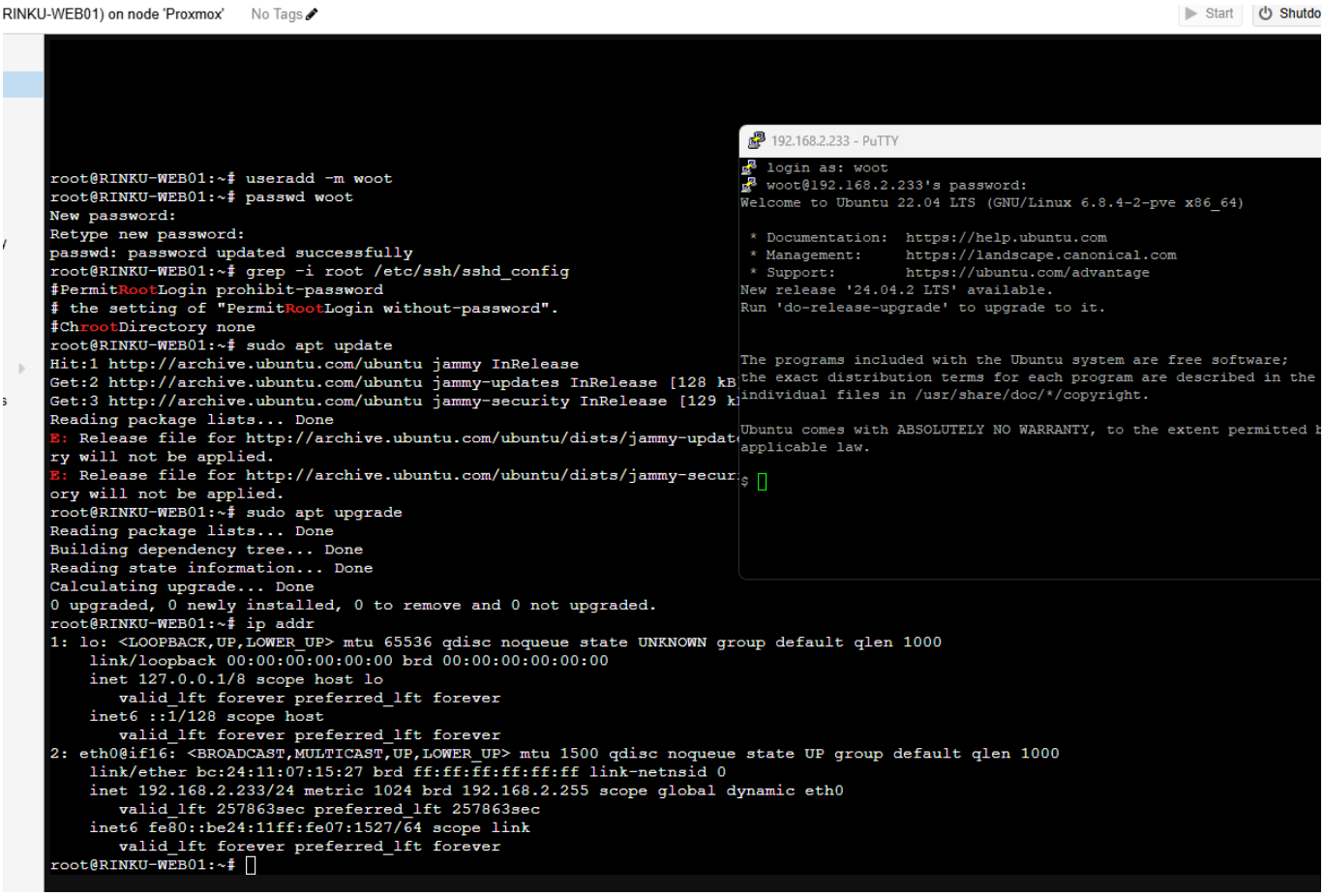
Ubuntu CT Baseline



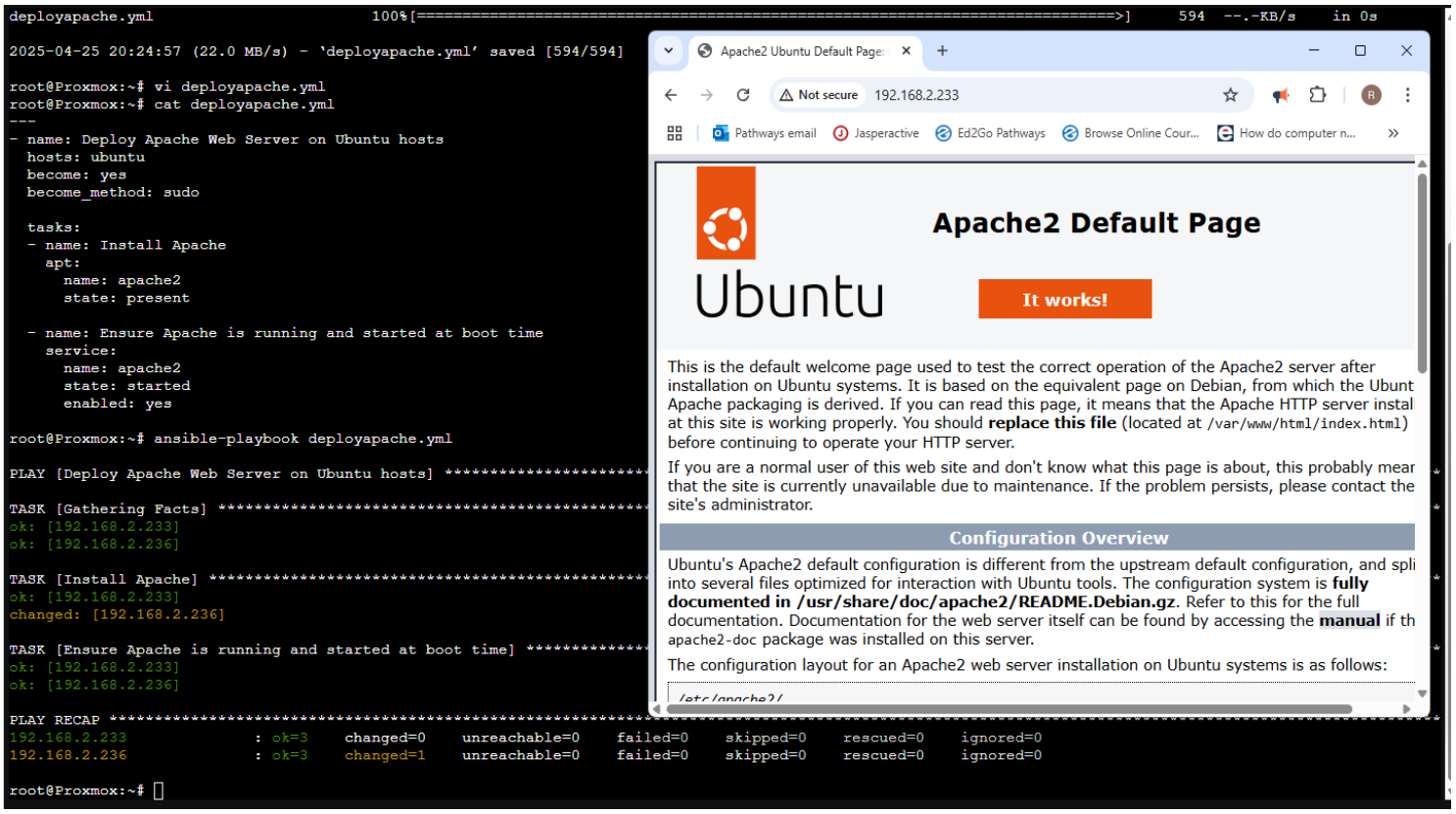
Ubuntu CT IP Configuration and Add Ansible User



Ubuntu CT Putty SSH Woot Login



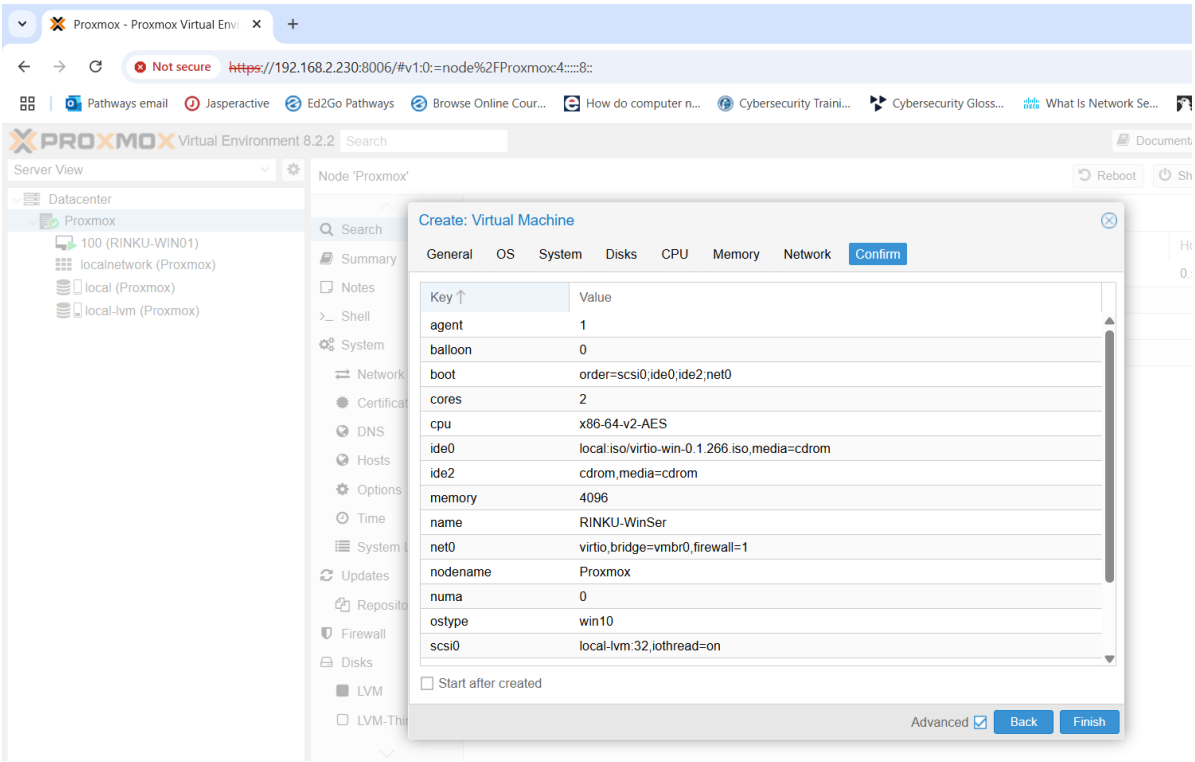
Ubuntu CT Deployapache.yml and Apache2 Works



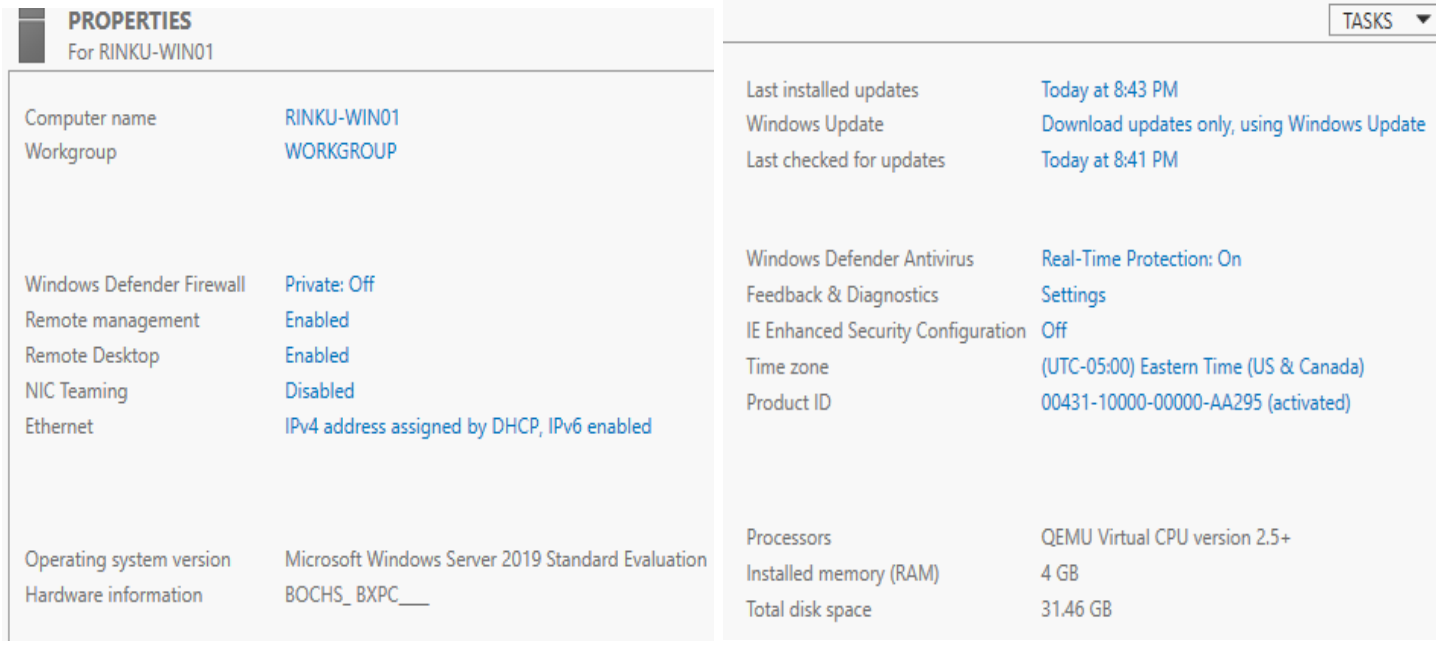
Windows Server Core VM

RINKU-WIN01	Windows Server 2019 + IIS, Python, Chocolatey	192.168.2.232	Ansible	4 GB Memory	2 Core

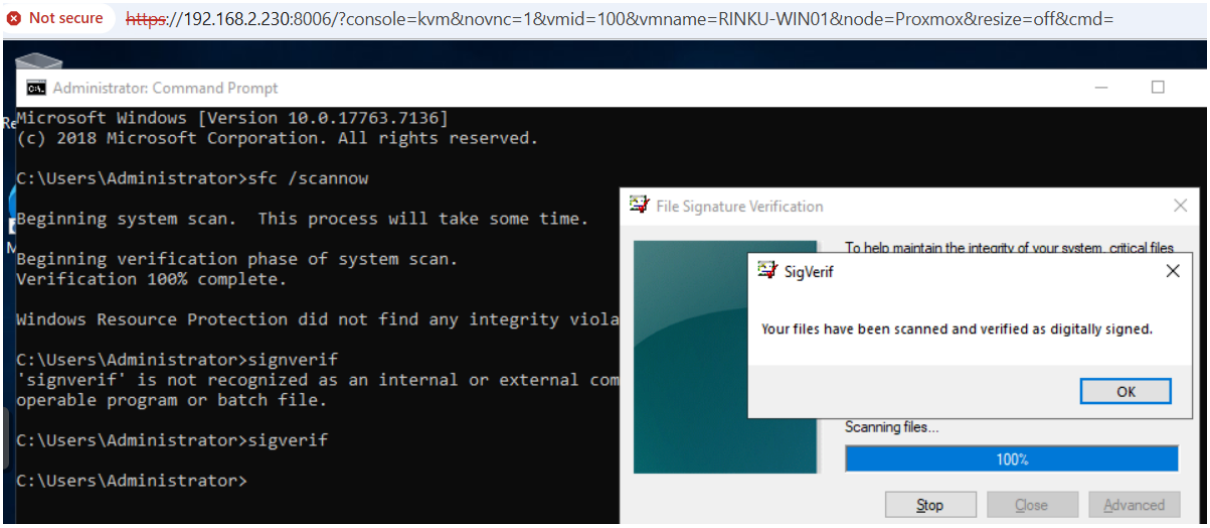
Windows Server 2019 Install Configuration



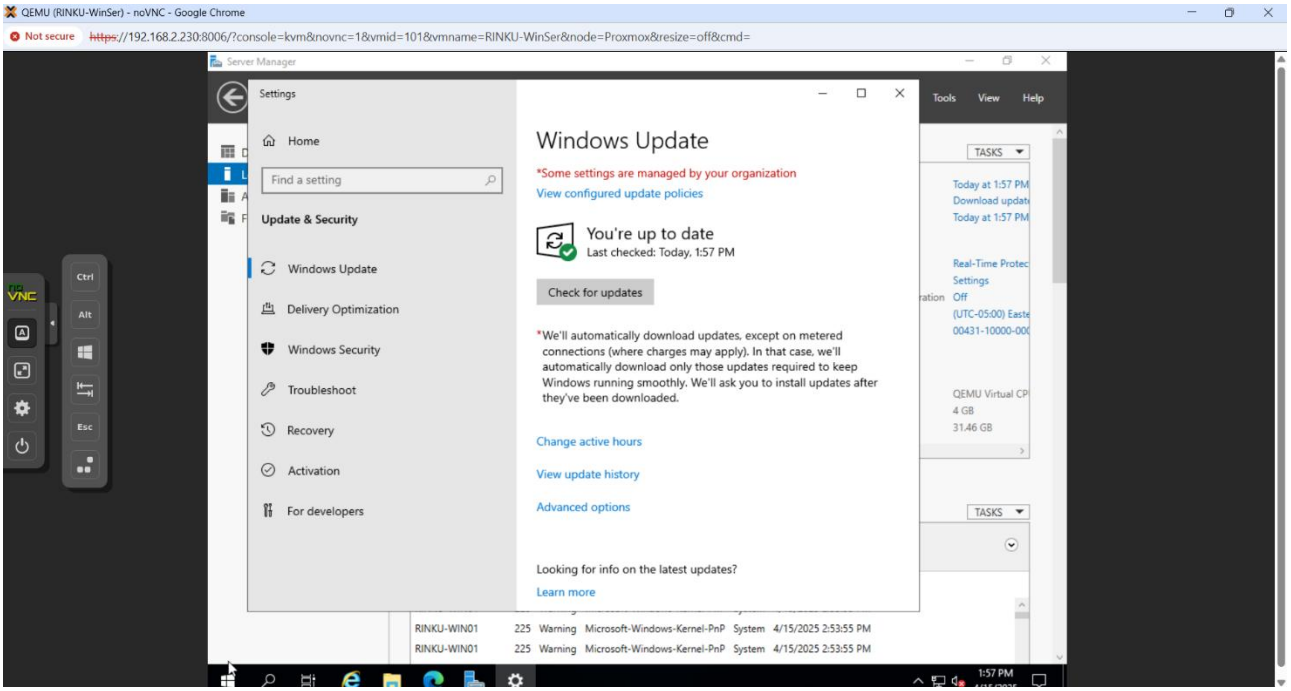
Windows Server Post Installation All Configurations



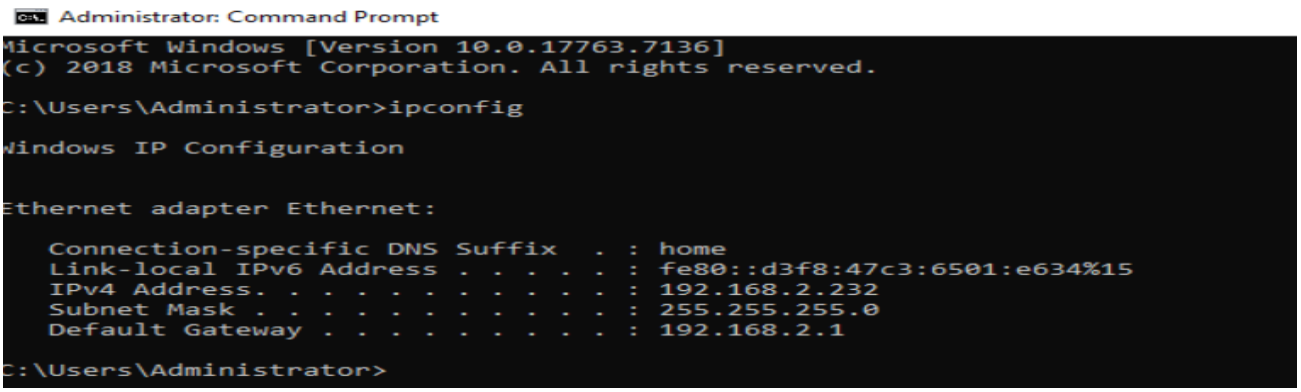
Windows Server SFC Scan and Sigverif



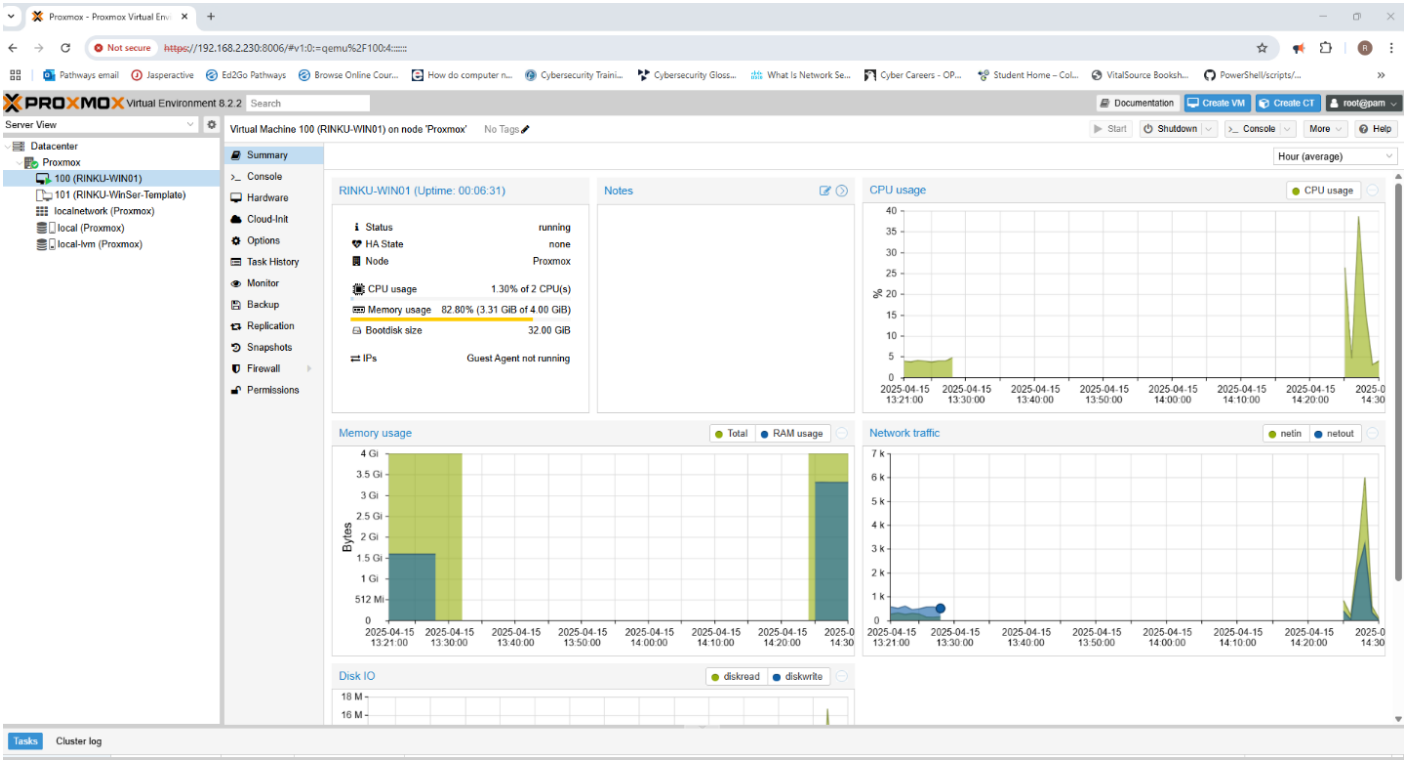
Windows Server Updates



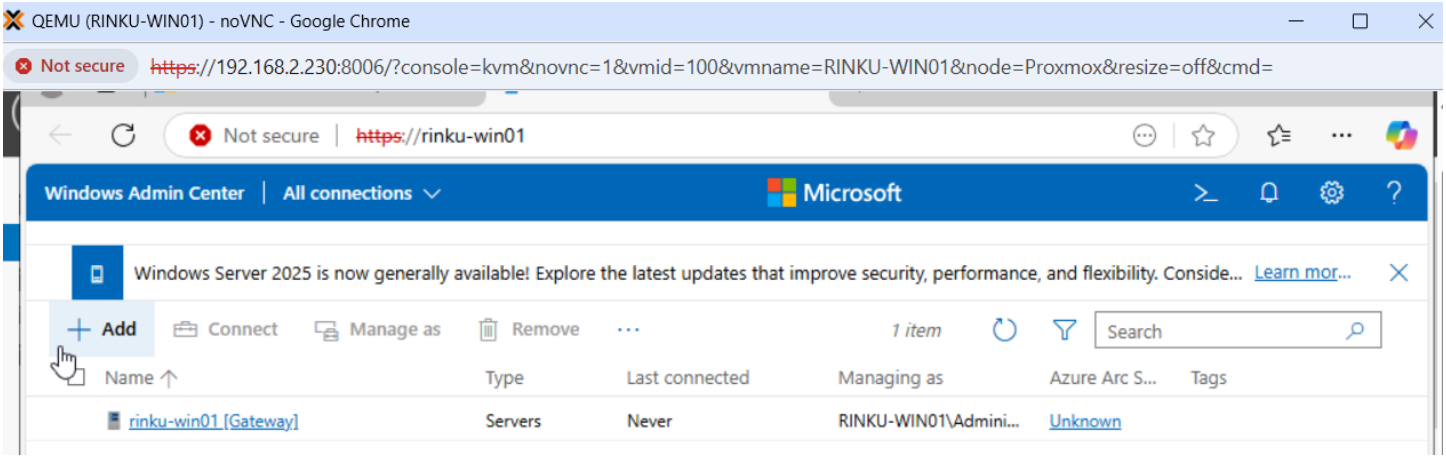
Windows Server IP Configuration



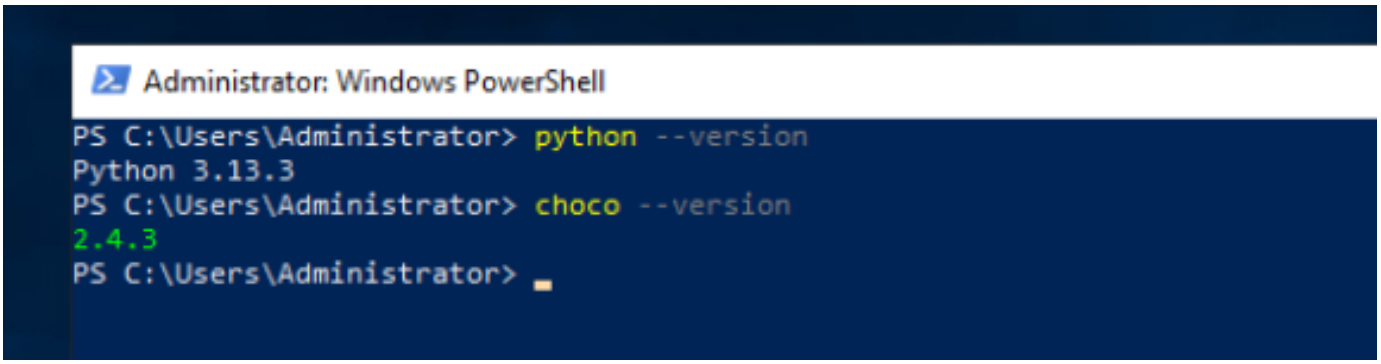
Windows Server Baseline



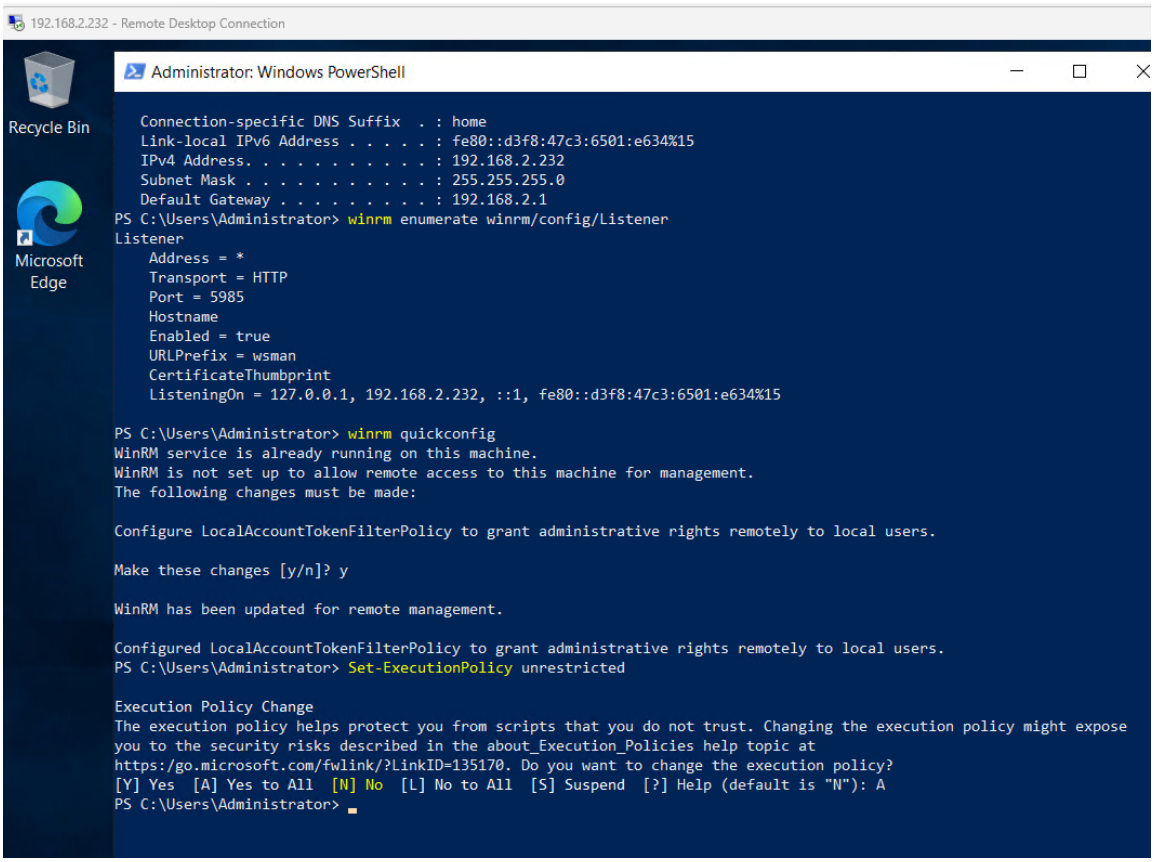
Windows Server Admin Center



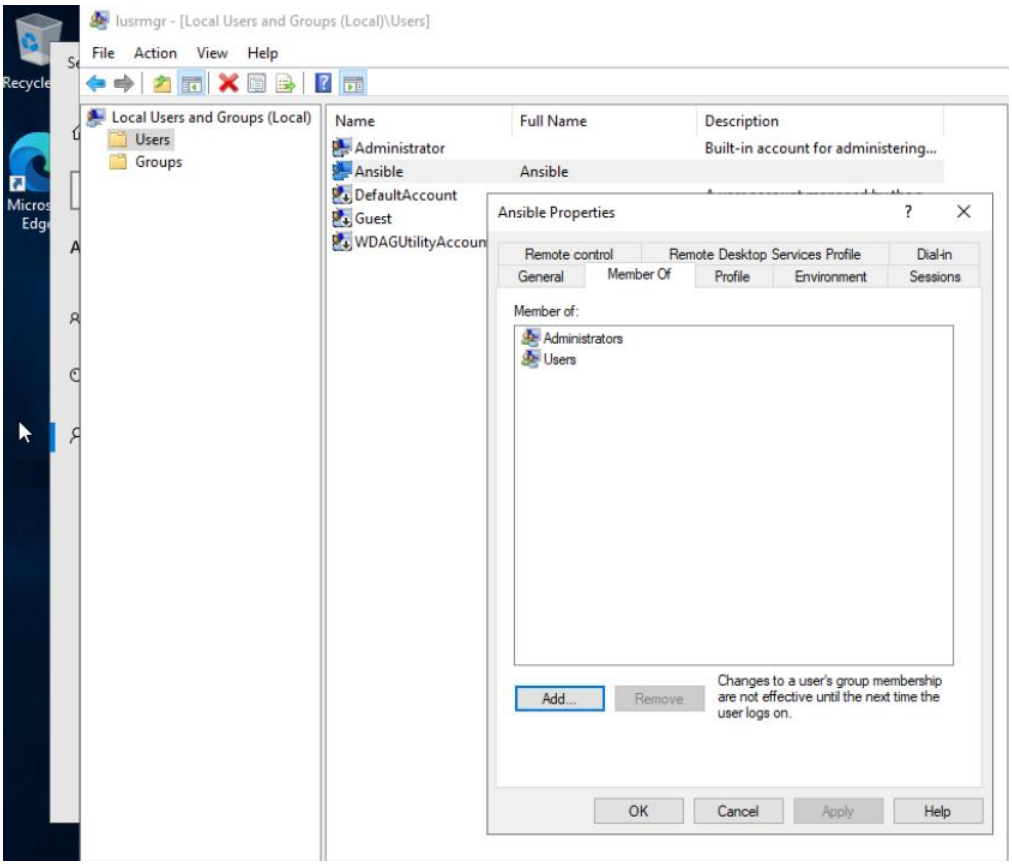
Windows Server Python and Chocolatey



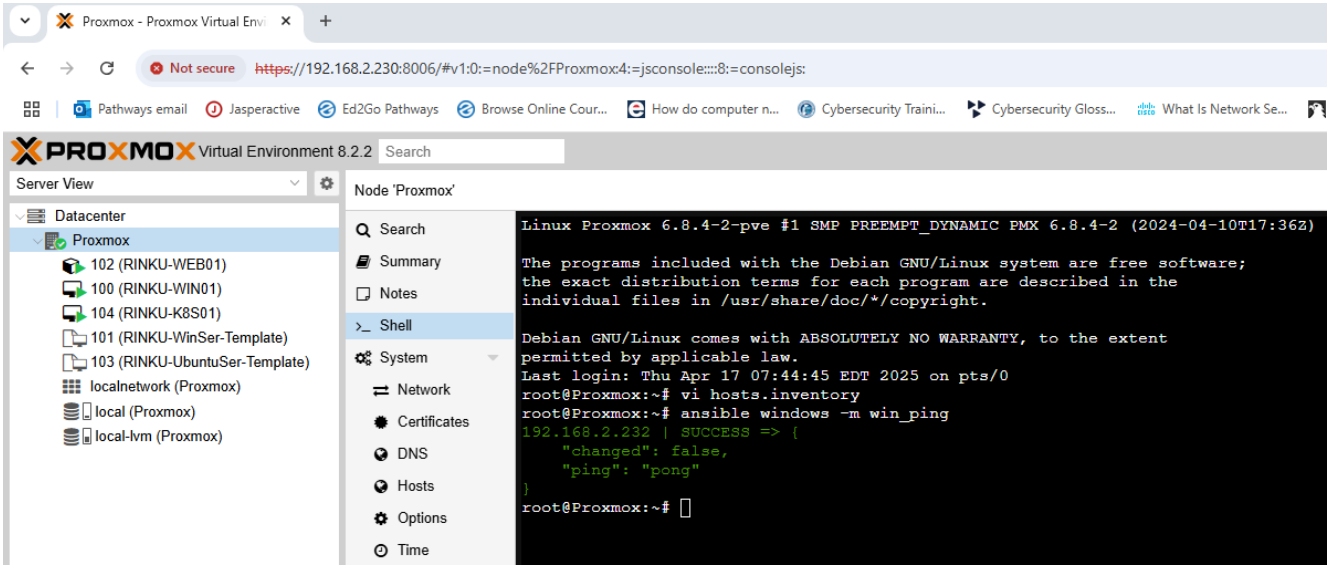
Windows Server IP Configuration and WinRm QuickConfig



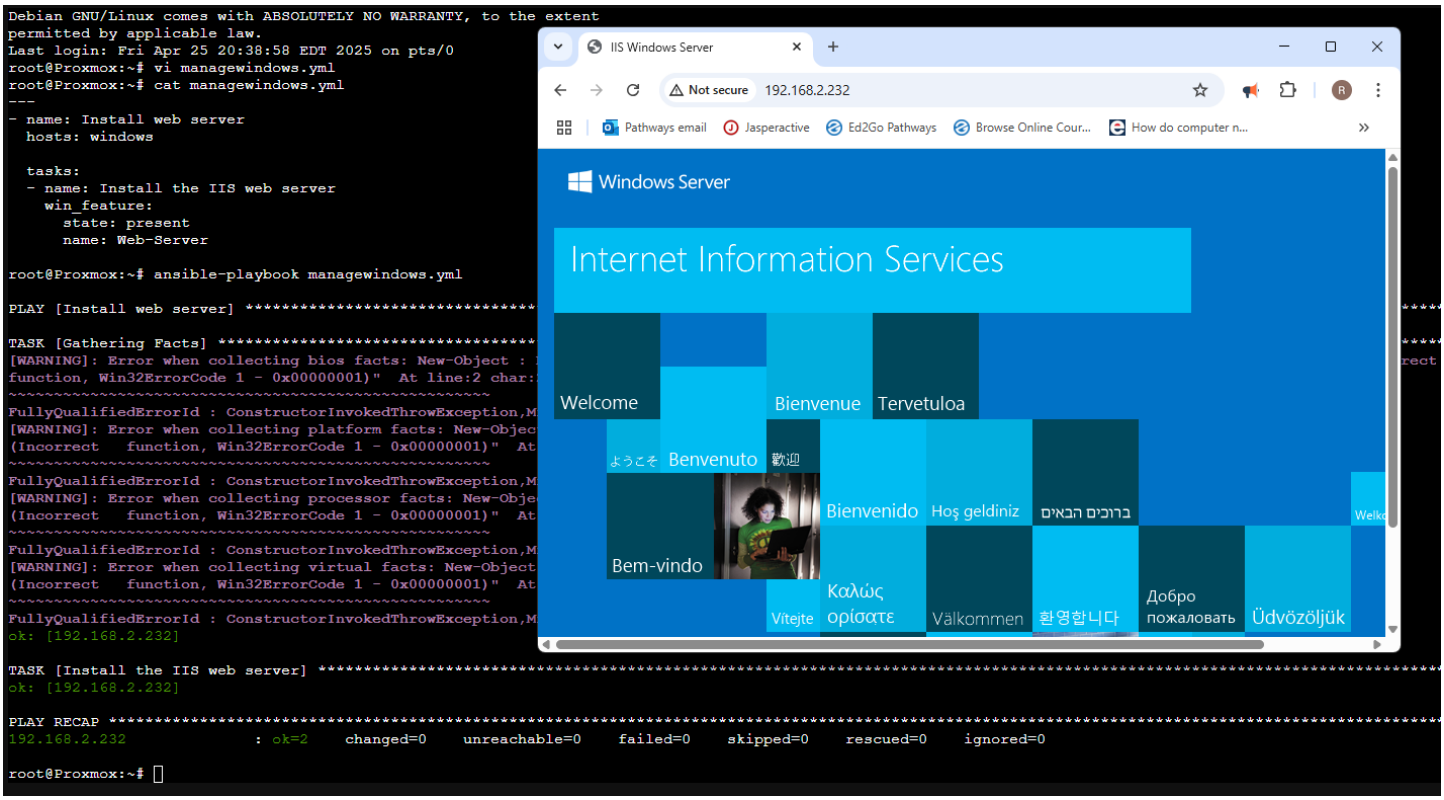
Windows Server Add Ansible User



Windows Server Ansible Win_Ping Success



Windows Server Managewindows.yml and IIS Web Page

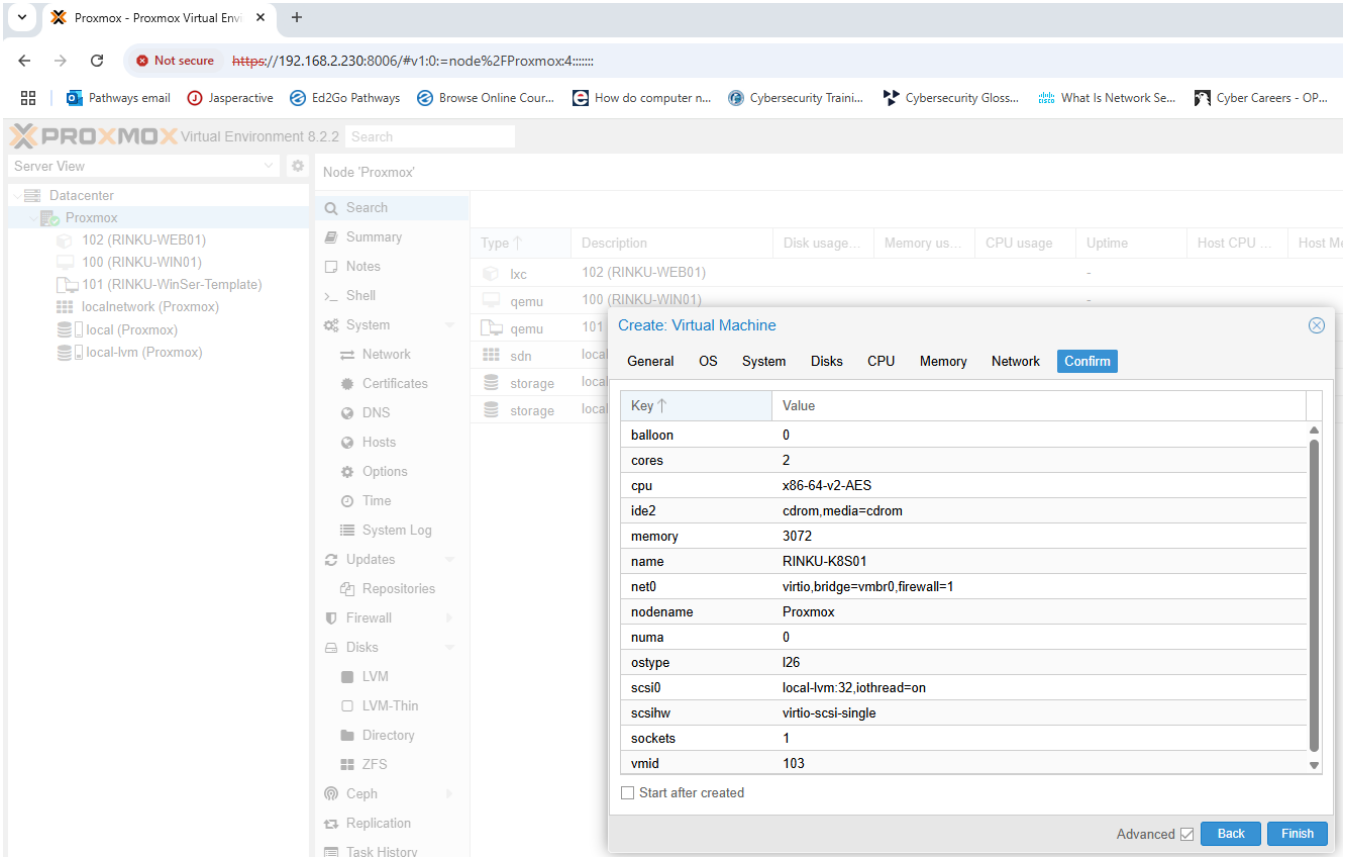


Windows Server Remote Desktop Connection Success

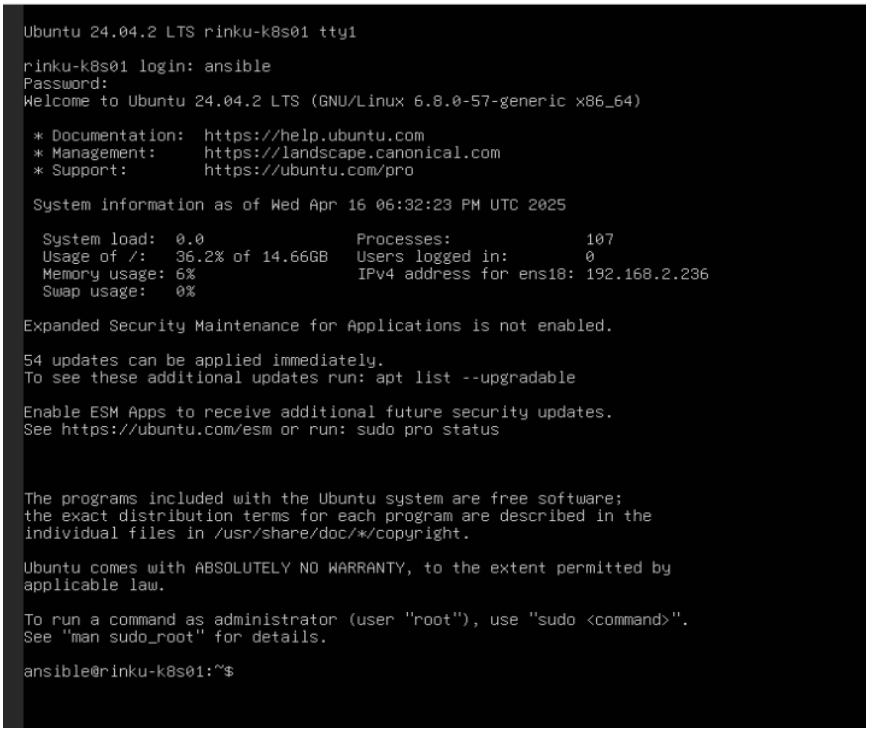
Linux Kubernetes VM

RINKU-K8S01	Ubuntu 24.04 + K3s	192.168.2.236	Ansible	3 GB Memory	2 Core

Ubuntu Server Installation Configuration



Ubuntu Server Installation Success and IP Configuration

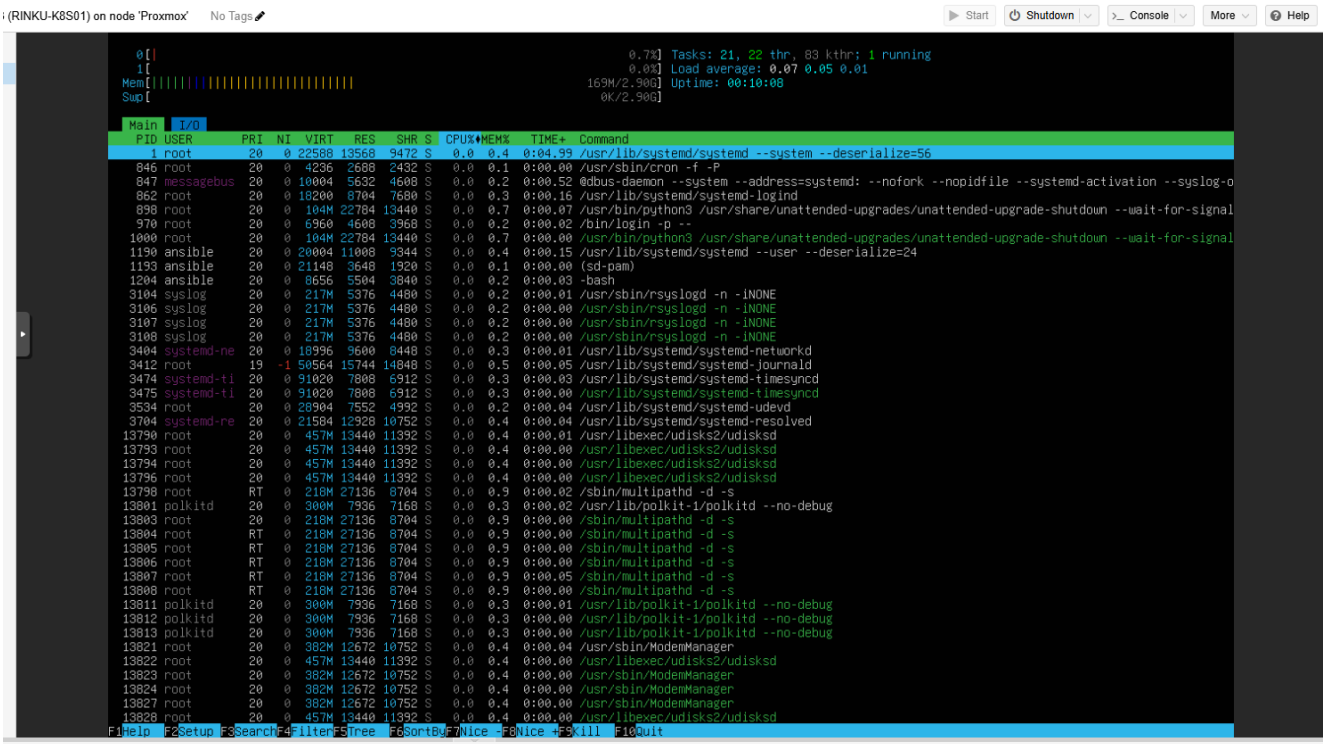


Ubuntu Server Updates and IP Configuration

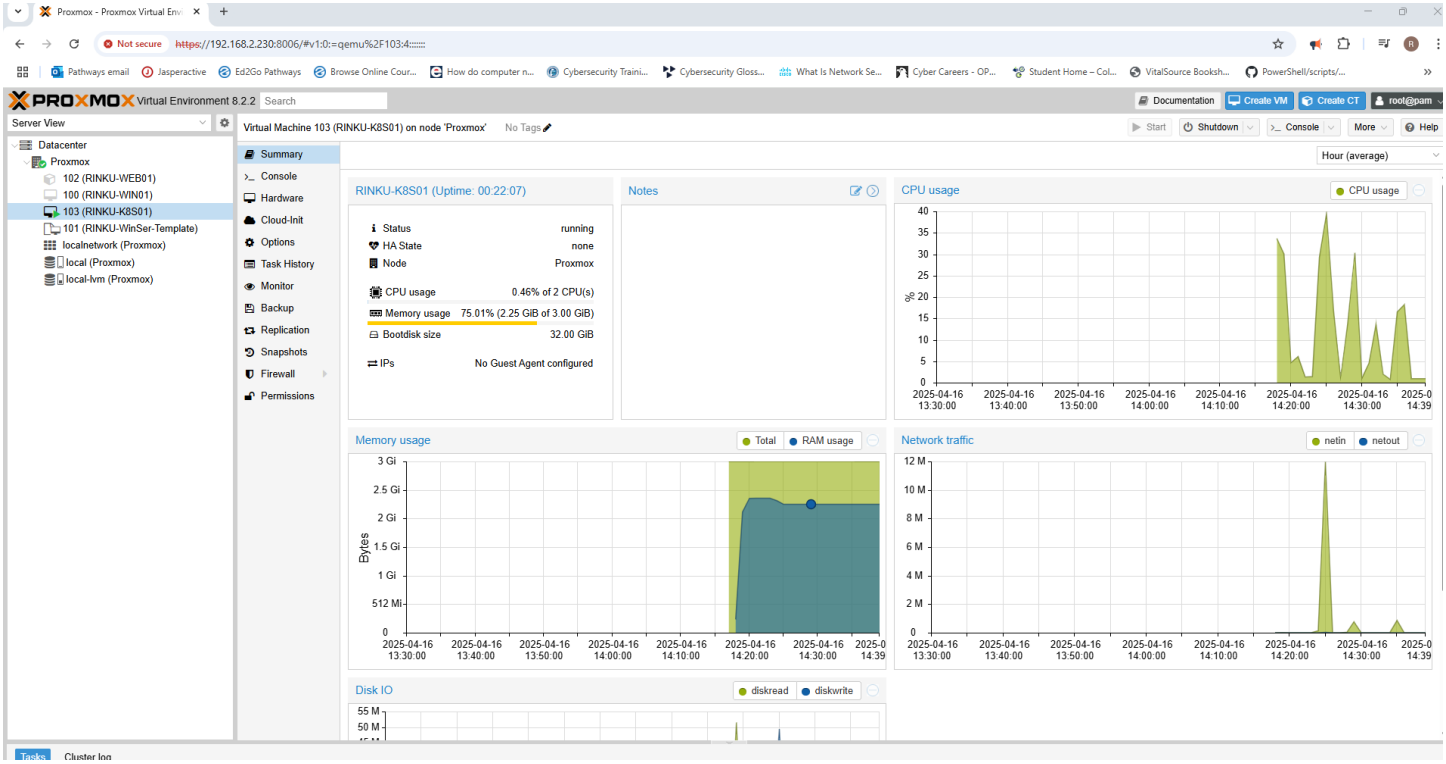
RINKU-K8S01) on node 'Proxmox' No Tags

```
Running kernel seems to be up-to-date.

Restarting services...
 systemctl restart multipathd.service polkit.service udisks2.service

Service restarts being deferred:
 systemctl restart ModemManager.service
 /etc/needrestart/restart.d/dbus.service
 systemctl restart systemd-logind.service
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
 ansible @ session #1: apt[1528], login[970]
 ansible @ user manager service: systemd[1190]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ansible@rinku-k8s01:~$ sudo apt update
Hit:1 http://ca.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://ca.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://ca.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
ansible@rinku-k8s01:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
  ubuntu-drivers-common
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
ansible@rinku-k8s01:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:3c:00:d4 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 192.168.2.236/24 metric 100 brd 192.168.2.255 scope global dynamic ens18
       valid_lft 258945sec preferred_lft 258945sec
    inet6 fe80::be24:11ff:fe3c:d4/64 scope link
       valid_lft forever preferred_lft forever
ansible@rinku-k8s01:~$
```

Ubuntu Server Htop



Ubuntu Server Baseline



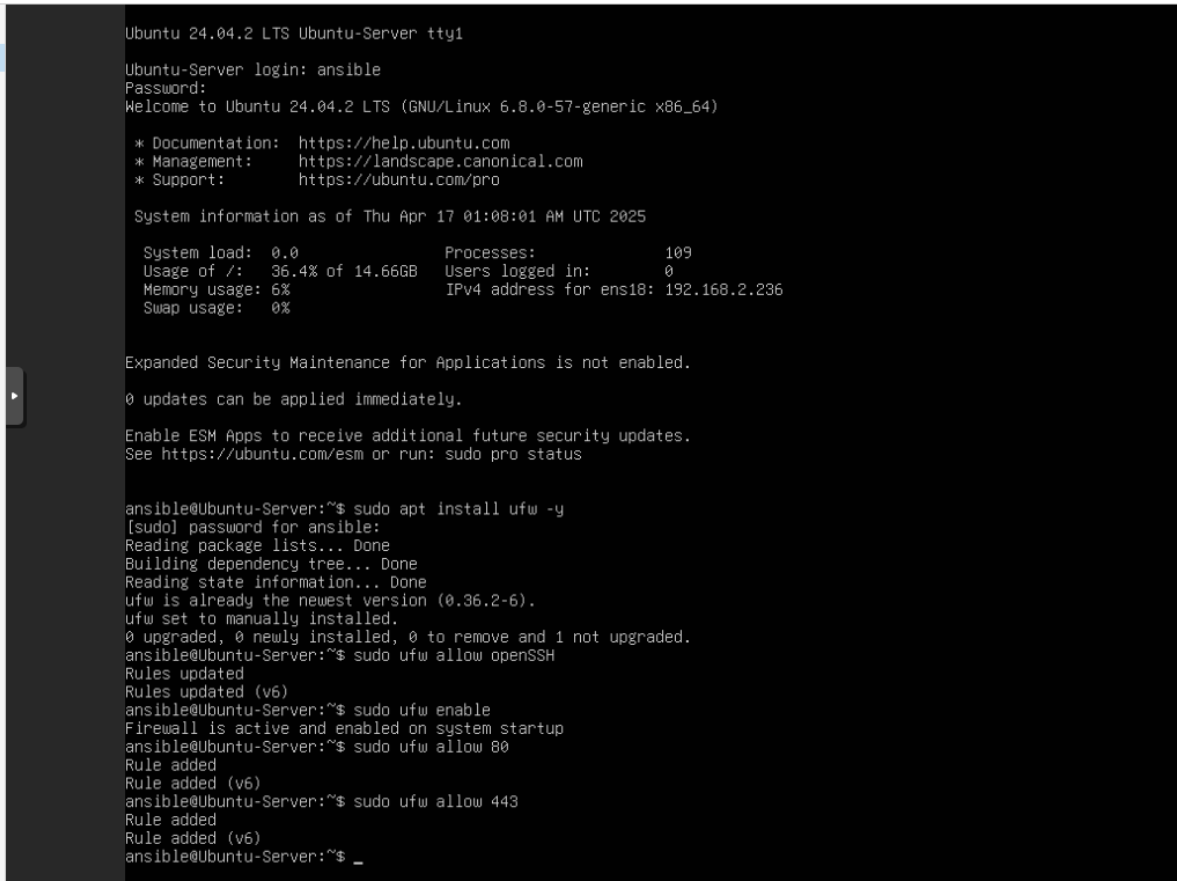
Ubuntu Server Add Ansible User

RINKU-K8S01) on node 'Proxmox' No Tags

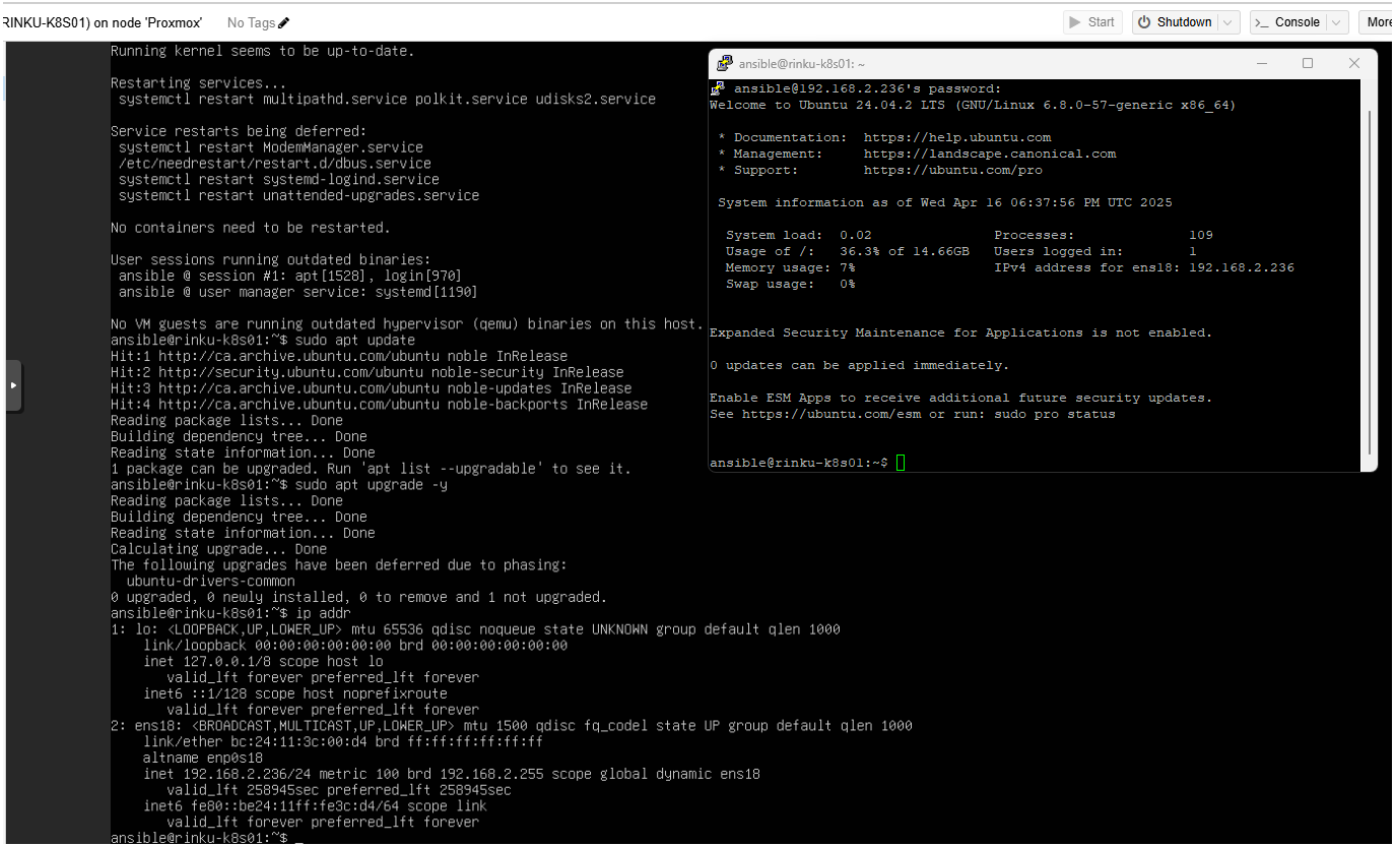
```
Ubuntu 24.04.2 LTS RINKU-K8S01 tty1

RINKU-K8S01 login: ansible
Password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-57-generic x86_64)

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

 System information as of Thu Apr 17 01:22:41 AM UTC 2025

  System load:  0.06              Processes:             111
  Usage of /:   36.5% of 14.66GB  Users logged in:       0
  Memory usage: 6%                IPv4 address for ens18: 192.168.2.236
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


ansible@RINKU-K8S01:~$ sudo usermod -aG sudo ansible
[sudo] password for ansible:
ansible@RINKU-K8S01:~$ grep sudo /etc/group
sudo:x:27:ansible
ansible@RINKU-K8S01:~$ grep "^%sudo" /etc/sudoers
grep: /etc/sudoers: Permission denied
ansible@RINKU-K8S01:~$ sudo grep "^%sudo" /etc/sudoers
%sudo   ALL=(ALL:ALL) ALL
ansible@RINKU-K8S01:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:73:a3:04 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 192.168.2.236/24 metric 100 brd 192.168.2.255 scope global dynamic ens18
       valid_lft 256890sec preferred_lft 256890sec
    inet6 fe80::be24:11ff:fe73:a304/64 scope link
       valid_lft forever preferred_lft forever
ansible@RINKU-K8S01:~$
```

Ubuntu Server Firewall

RINKU-K8S01) on node 'Proxmox' No Tags



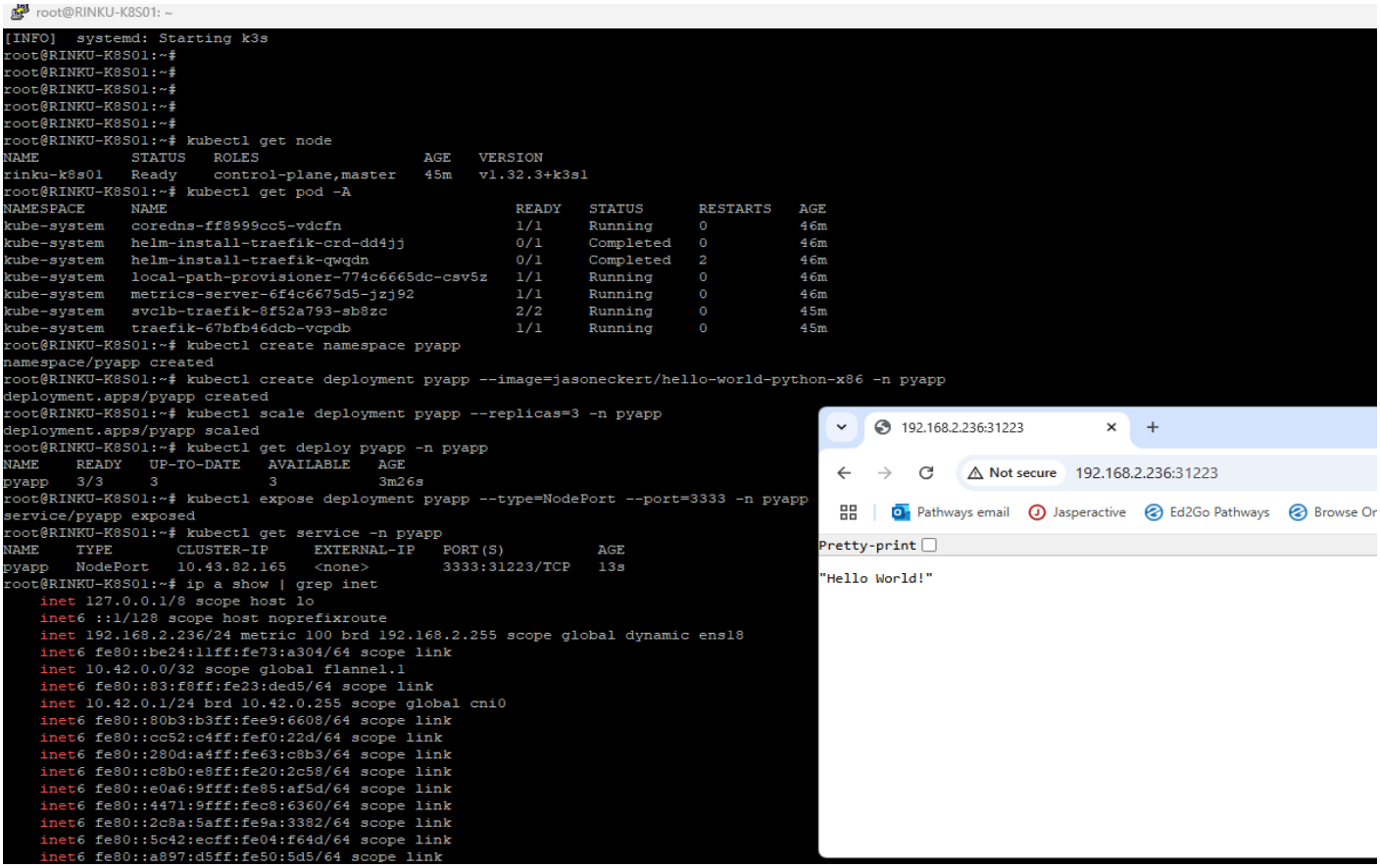
Ubuntu Server Putty SSH Ansible Login



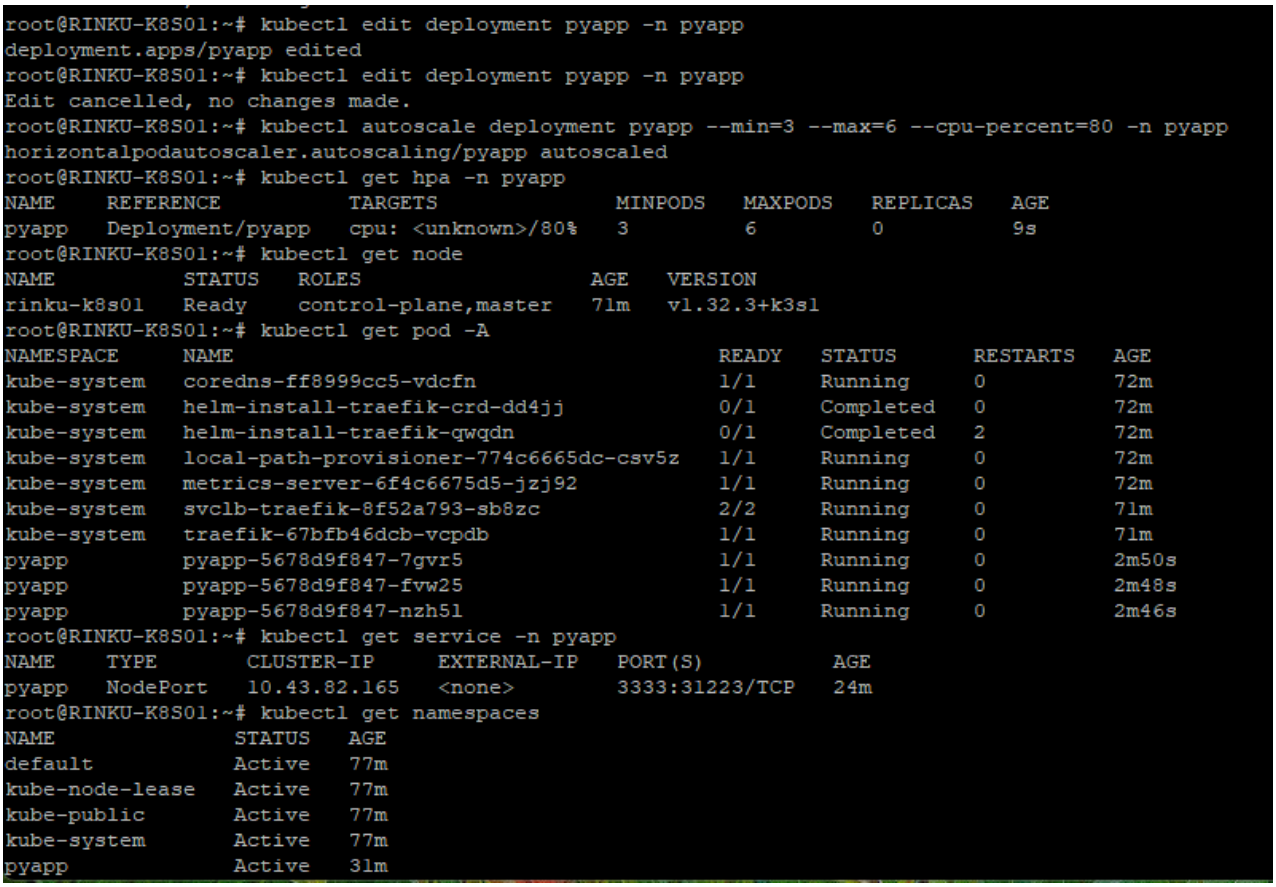
The following steps have been used to deploys a Python web application on Kubernetes using k3s, a lightweight distribution, ensuring the app is scalable and externally accessible. It covers installing Kubernetes, creating an isolated namespace, deploying the app, scaling replicas for availability, exposing the service via NodePort, and configuring autoscaling based on CPU usage. The setup is efficient and provides flexibility for resource management and further customization.

Step	Description	Command
Step 1	Install k3s	curl -sfL https://get.k3s.io | sh -
Step 2	Verify Kubernetes Node	kubectl get node
Step 3	Check All Pods Across Namespaces	kubectl get pod -A
Step 4	Create Namespace for the Application	kubectl create namespace pyapp
Step 5	Deploy the Python Web App	kubectl create deployment pyapp --image=jasoneckert/hello-world-python-x86 -n pyapp
Step 6	Scale the Deployment to 3 Replicas	kubectl scale deployment pyapp --replicas=3 -n pyapp
Step 7	Expose the Deployment as a NodePort Service	kubectl expose deployment pyapp --type=NodePort --port=3333 -n pyapp
Step 8	Get Service Details to Check the Assigned NodePort	kubectl get service -n pyapp
Step 9	Edit Deployment Configuration (Optional)	kubectl edit deployment pyapp -n pyapp
Step 10	Enable Autoscaling with Minimum and Maximum Replicas	kubectl autoscale deployment pyapp --min=3 --max=6 --cpu-percent=80 -n pyapp
Step 11	Verify Horizontal Pod Autoscaler	kubectl get hpa -n pyapp

Ubuntu Server Deploy Pyapp and IP configuration



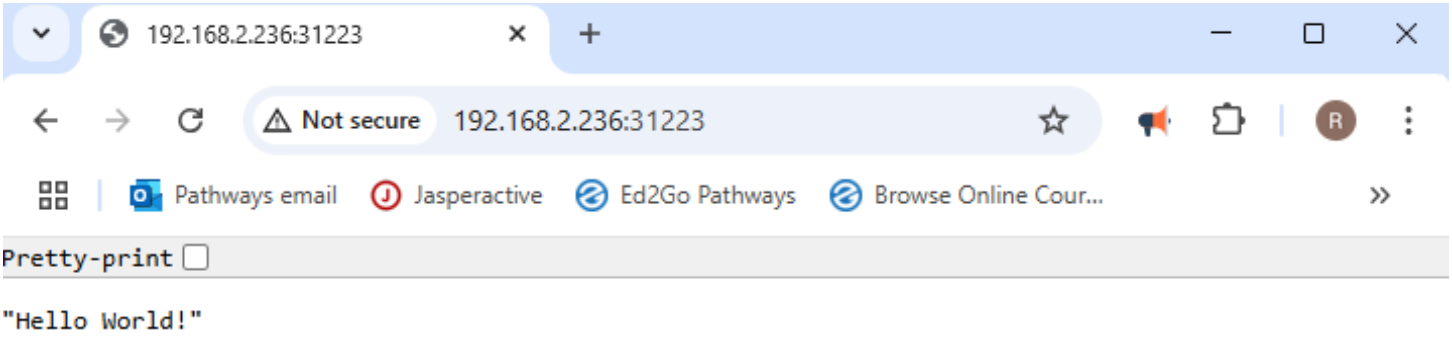
Ubuntu Server AutoScale Deployment Pyapp, Get Node, Get HPA, Get Pod, Get Service, Get Namespaces



Ubuntu Server Deploy Pyapp Configuration

```
root@RINKU-K8S01: ~
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "2"
  creationTimestamp: "2025-04-26T02:05:37Z"
  generation: 3
  labels:
    app: pyapp
  name: pyapp
  namespace: pyapp
  resourceVersion: "2198"
  uid: f5cc48b6-b37a-4297-8b81-43cdb69352de
spec:
  progressDeadlineSeconds: 600
  replicas: 3
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: pyapp
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: pyapp
    spec:
      containers:
      - image: jasoneckert/hello-world-python-x86
        imagePullPolicy: Always
        name: hello-world-python-x86
        resources:
          requests:
            cpu: 100m
            memory: 128Mi
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
- /tmp/kubectl-edit-1275464791.yaml 1/70 1%
```

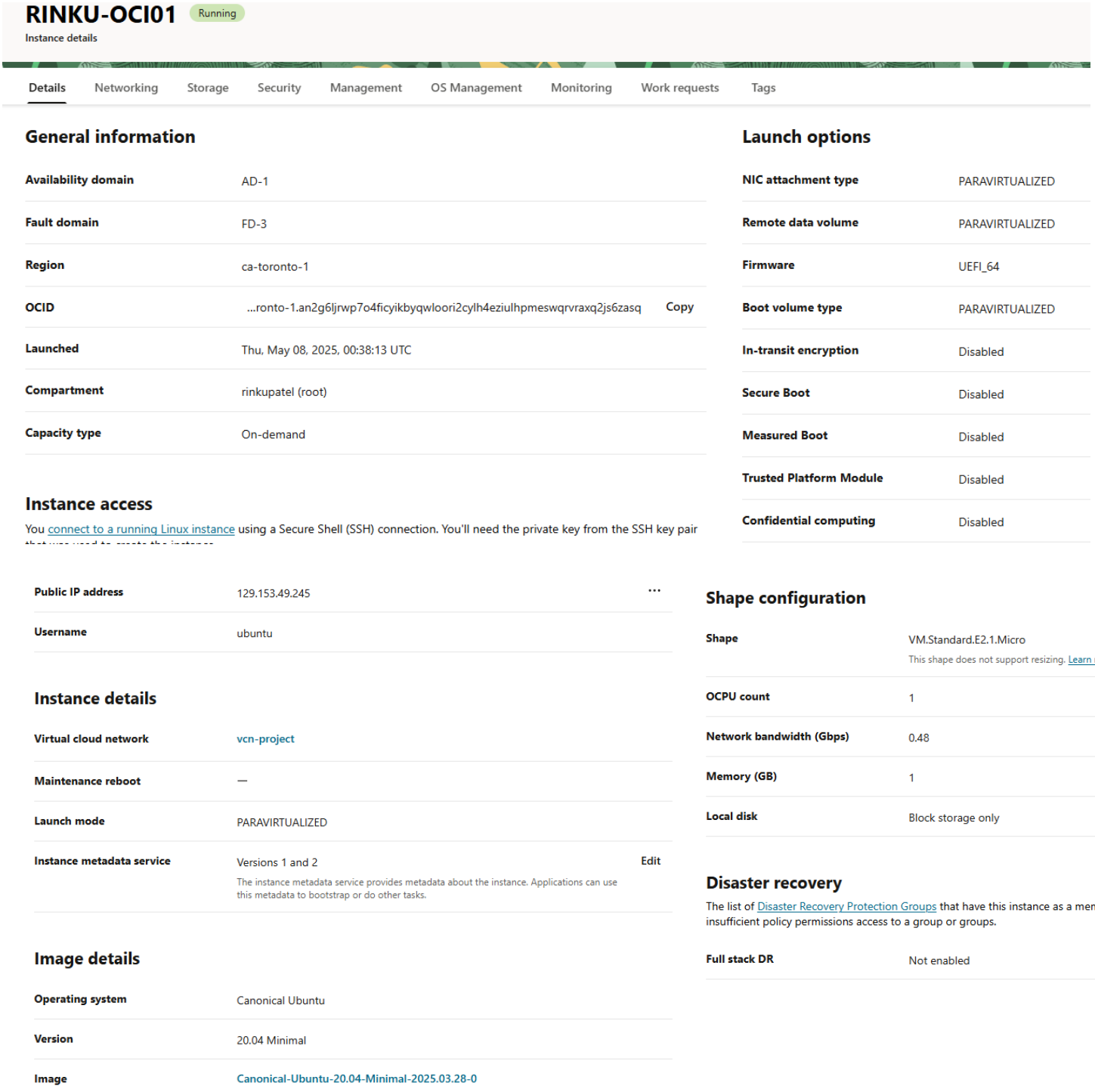
Ubuntu Server Web Page



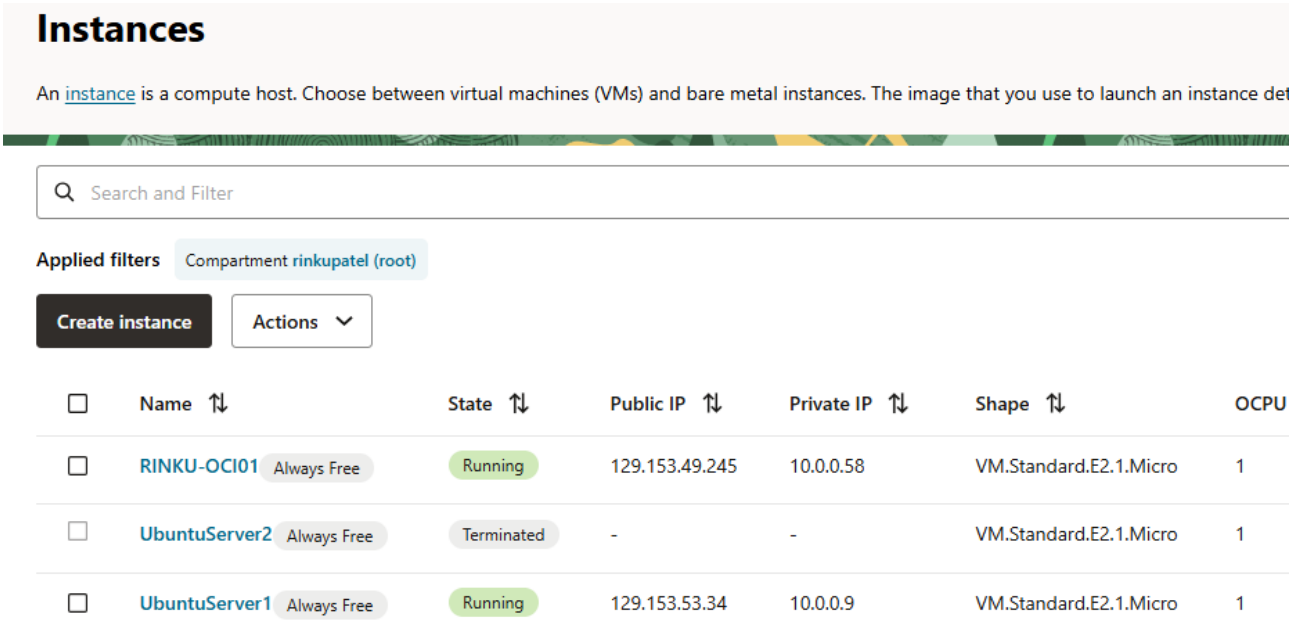
Oracle Cloud VM

RINKU-OCI01	Oracle Cloud Linux VM/ Canonical Ubuntu 20.04 Minimal	Public IP 129.153.49.245 Private IP 10.0.0.58	Ubuntu	1 GB Memory	1Core

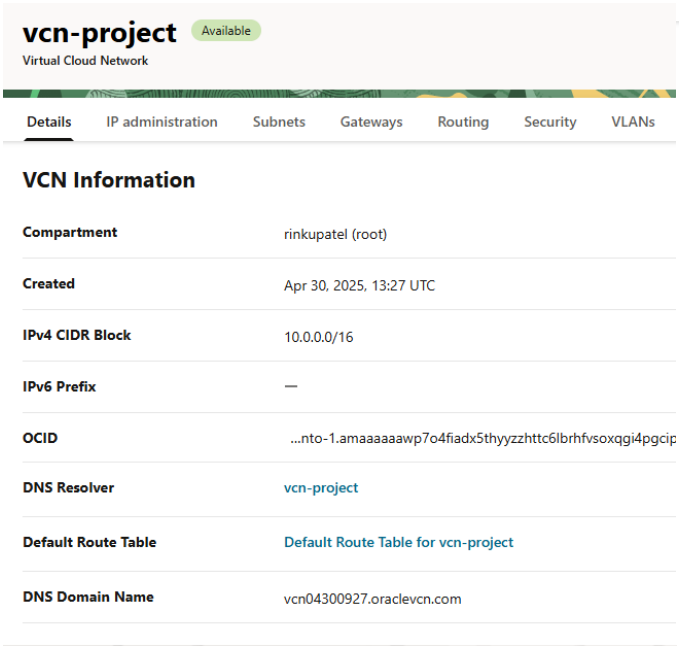
Ubuntu Cloud VM Configurations

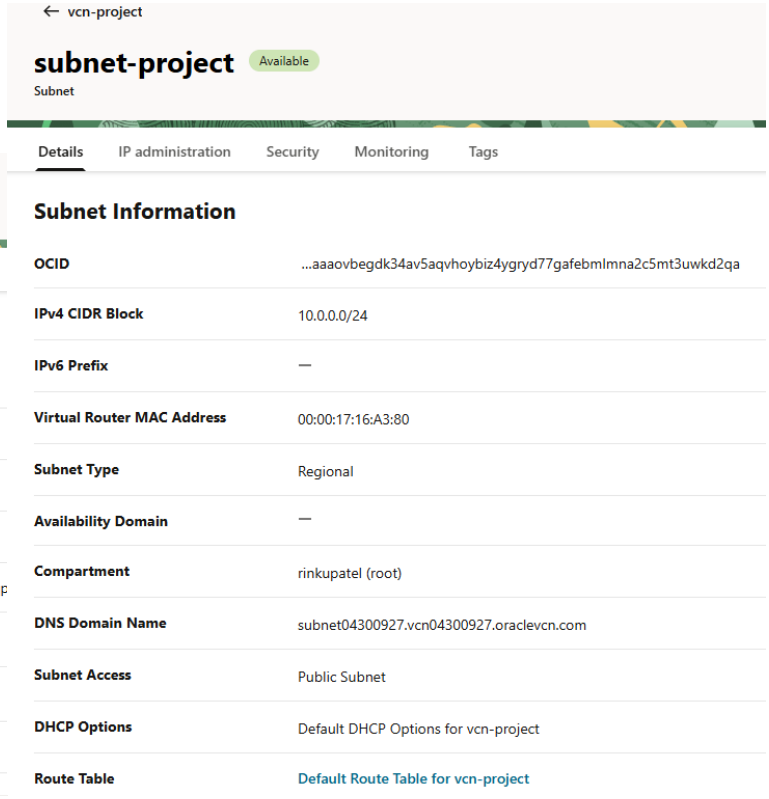


Ubuntu Cloud RINKU-OCI01 Instance

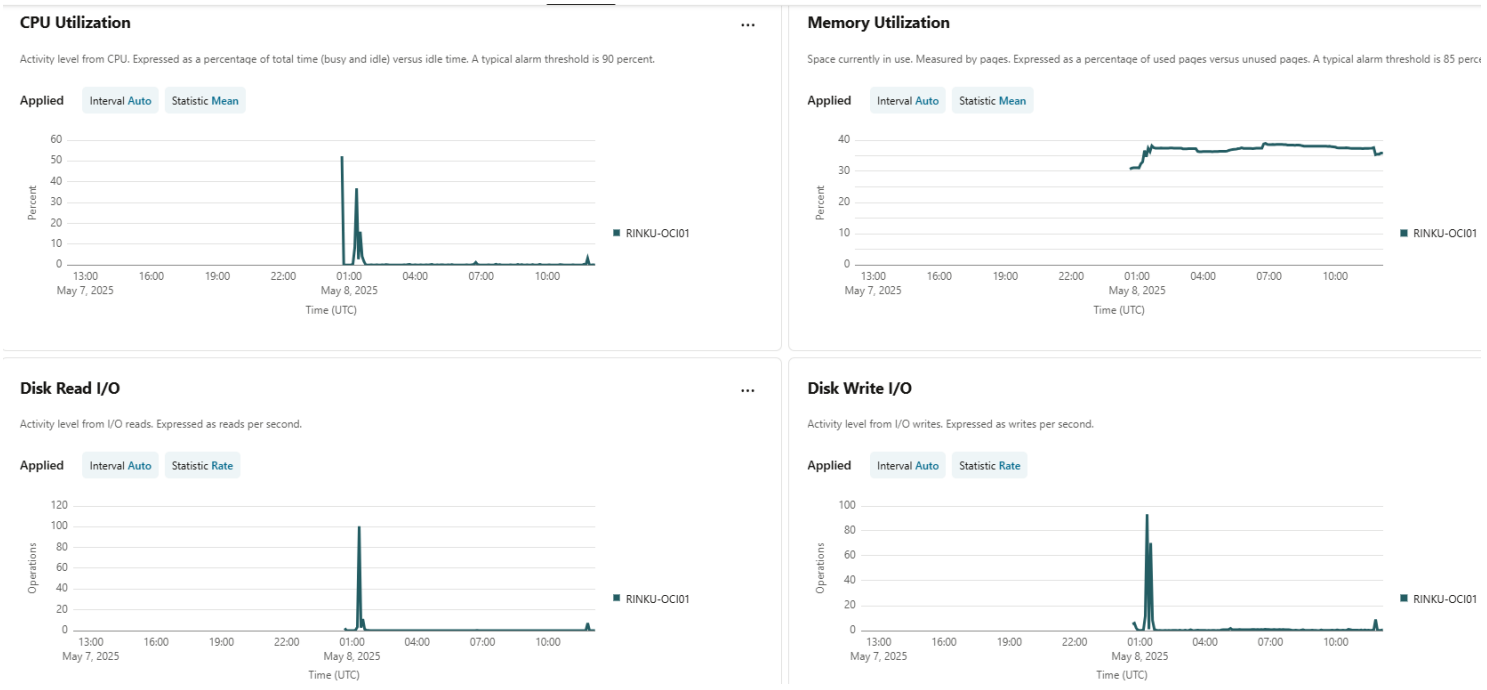


Ubuntu Cloud VCN Configuration and Ubuntu Cloud Subnet Configuration

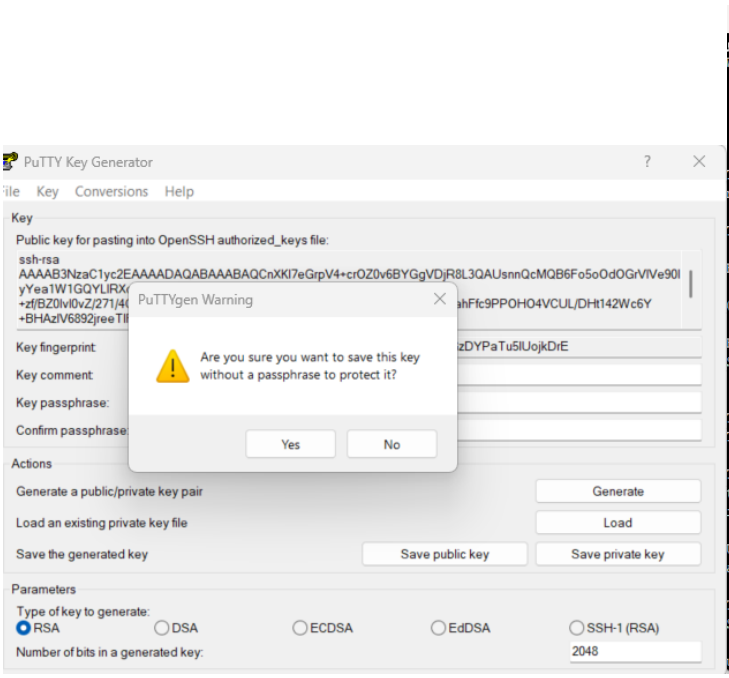


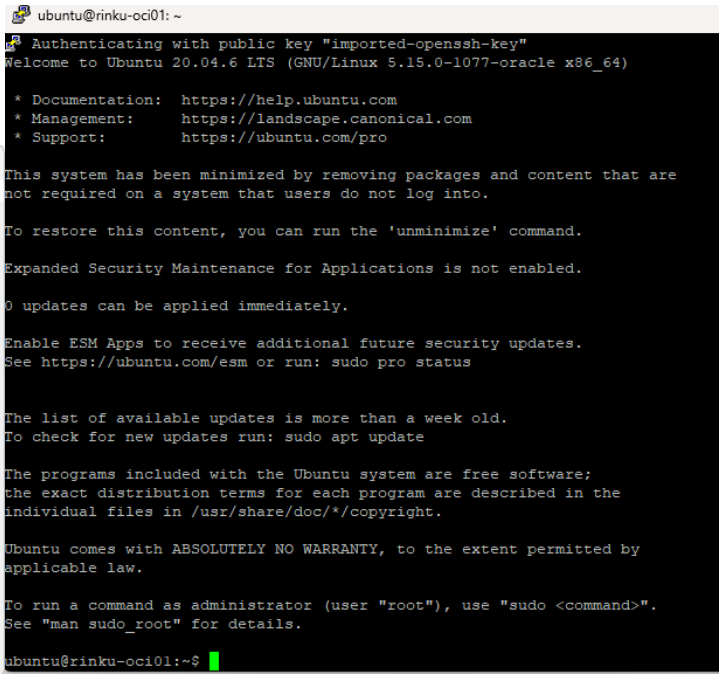


Ubuntu Cloud Baselines

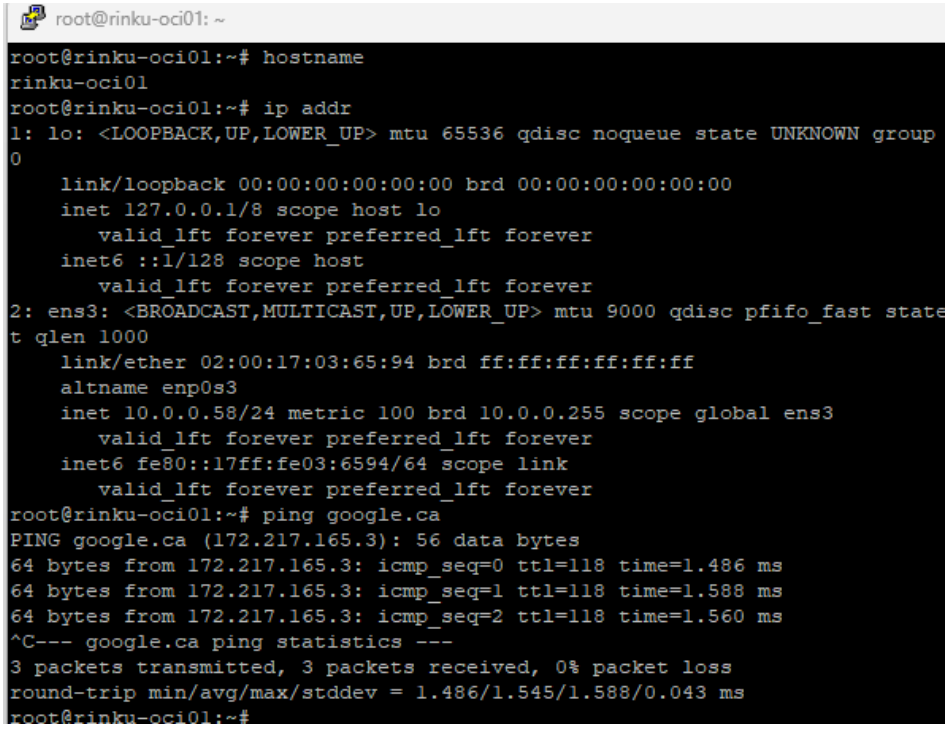


Ubuntu Cloud Puttygen Key Conversion and Login with SSH with Putty

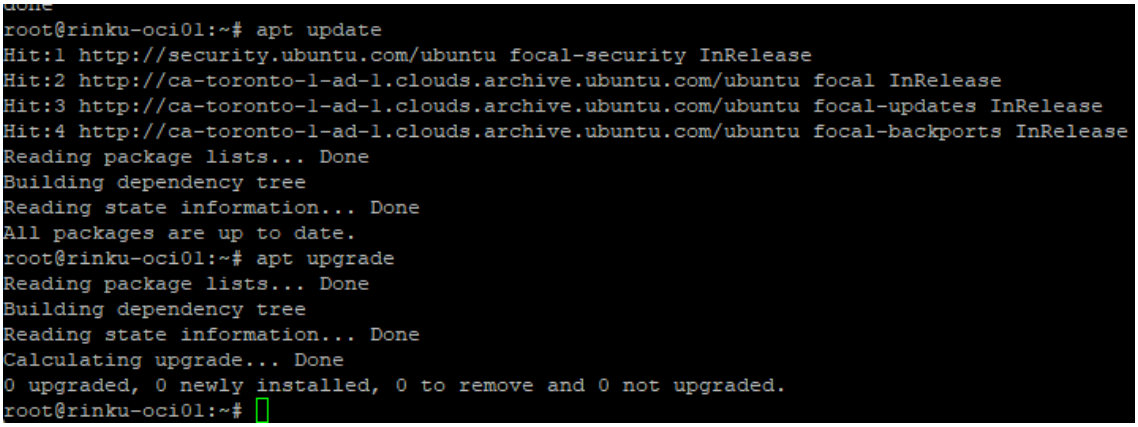




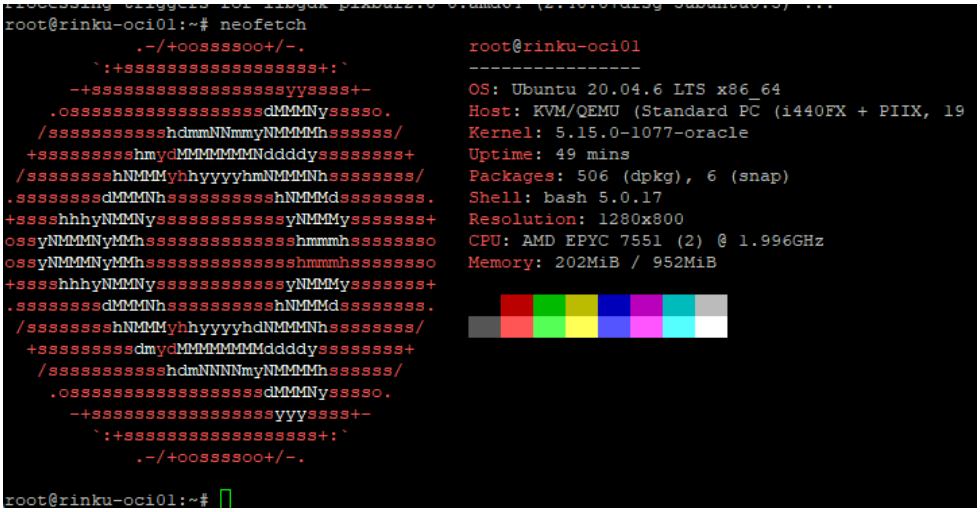
Ubuntu Cloud Hostname, IP configuration, Ping google.ca



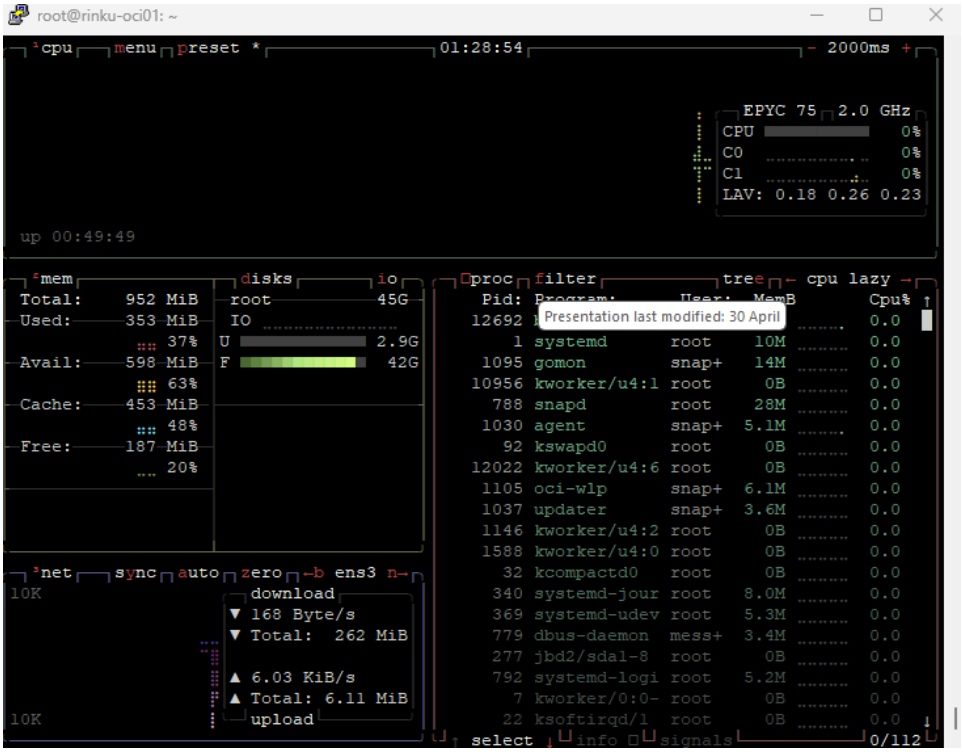
Ubuntu Cloud Updates



Ubuntu Cloud Neofetch



Ubuntu Cloud Btop



Ubuntu Cloud Firewall Iptables-Rules

```
root@rinku-oci01:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain InstanceServices (0 references)
target     prot opt source               destination
root@rinku-oci01:~# netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
root@rinku-oci01:~#
```

Ubuntu Cloud Ingress Rules

Ingress Rules

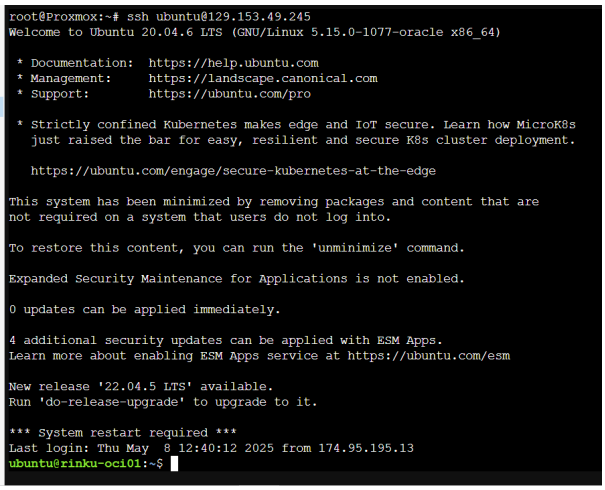
Search and Filter

Add Ingress Rules | Actions

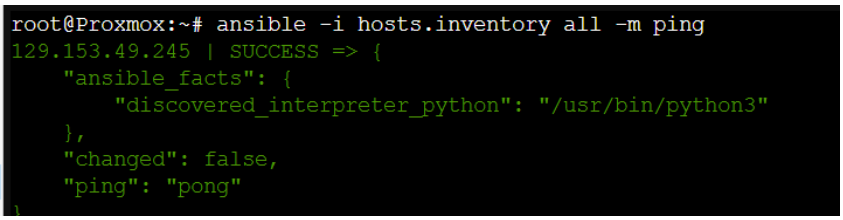
Stateless	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows	Description
No	0.0.0.0/0	TCP	All	22		TCP traffic for ports: 22 SSH Remote Login Protocol	
No	0.0.0.0/0	ICMP			3, 4	ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set	
No	10.0.0.0/16	ICMP			3	ICMP traffic for: 3 Destination Unreachable	
No	0.0.0.0/0	TCP	All	80		TCP traffic for ports: 80	Allow inbound HTTP
No	0.0.0.0/0	TCP	All	443		TCP traffic for ports: 443 HTTPS	Allow inbound HTTPS

Ubuntu Cloud Apache2

Ubuntu Cloud SSH Login Proxmox

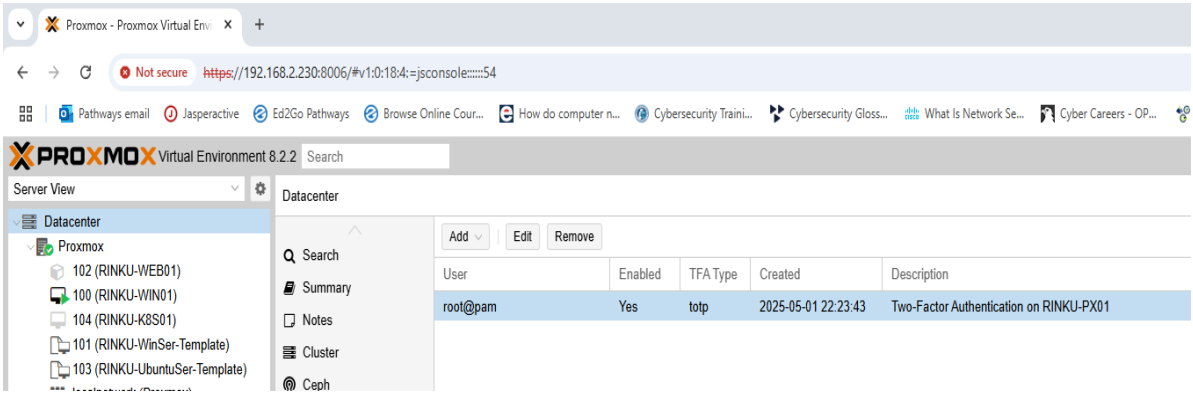


Ubuntu Cloud Ping Success



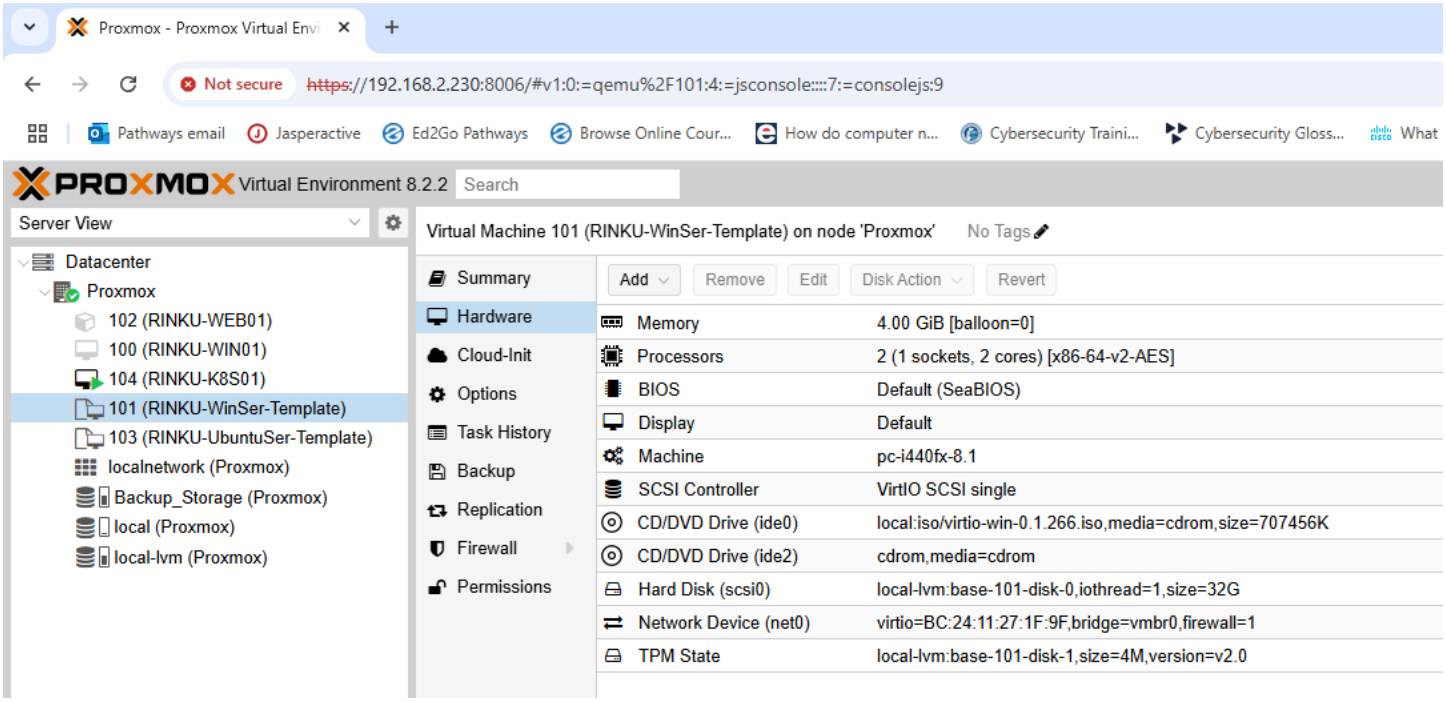
Security & Authentication

Two-Factor Authentication (2FA) on RINKU-PX01

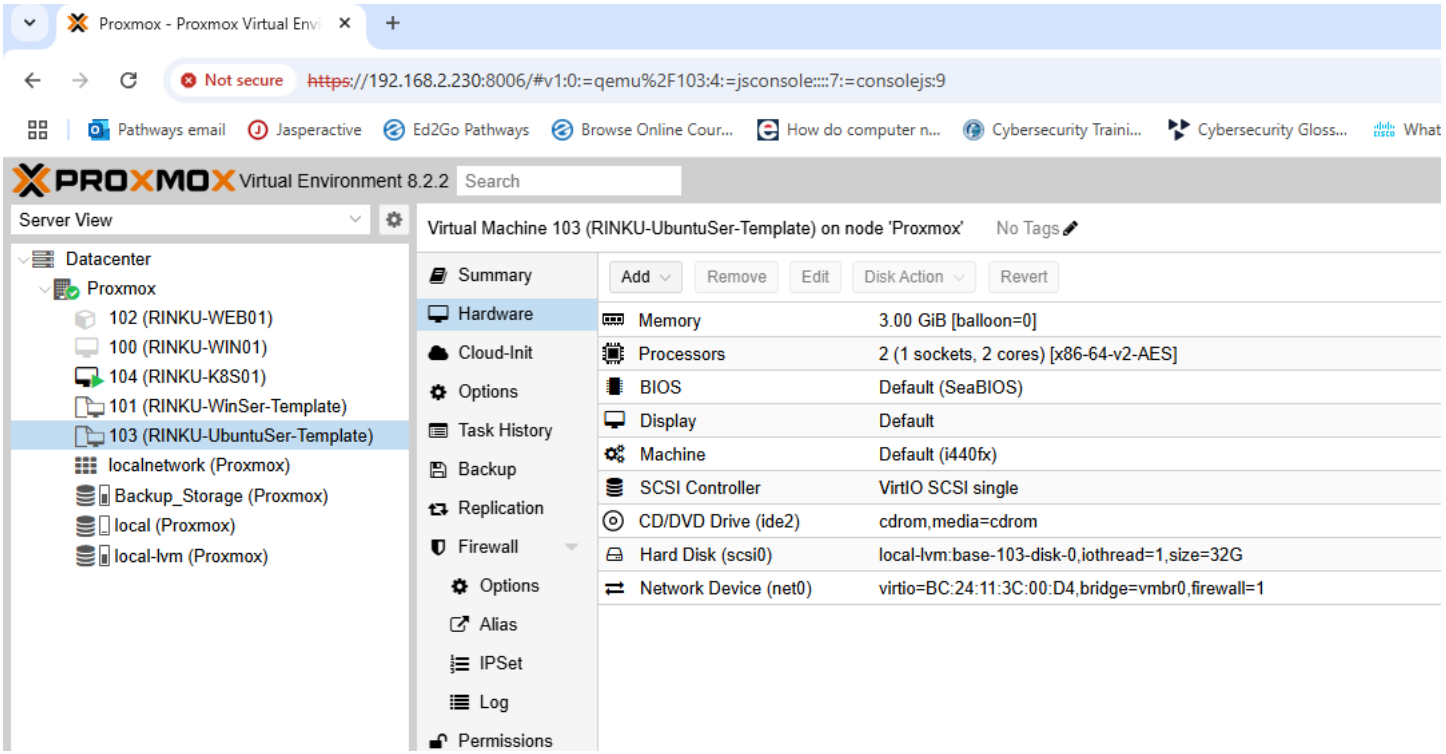


System Templates

Windows Server Template



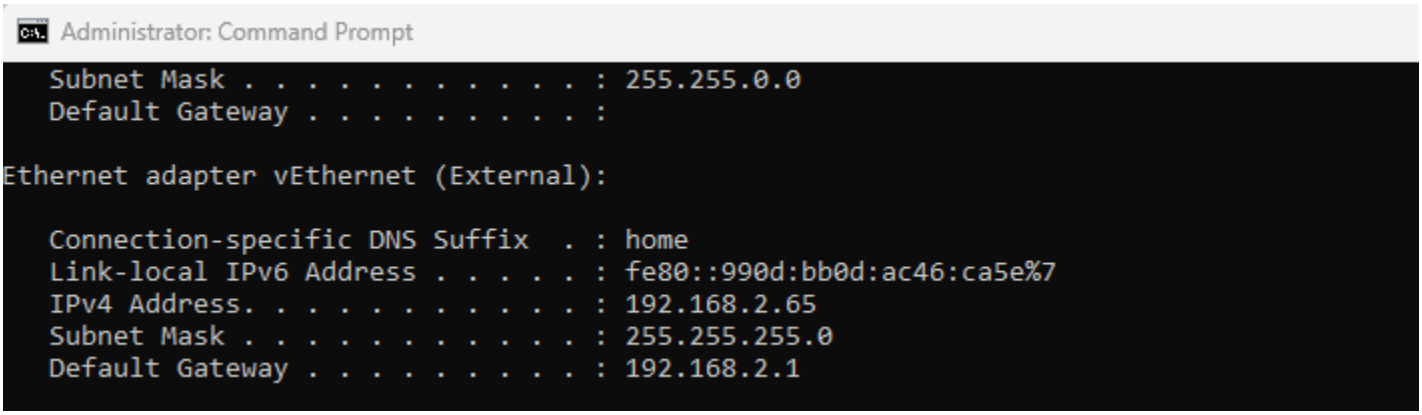
Ubuntu Server Template



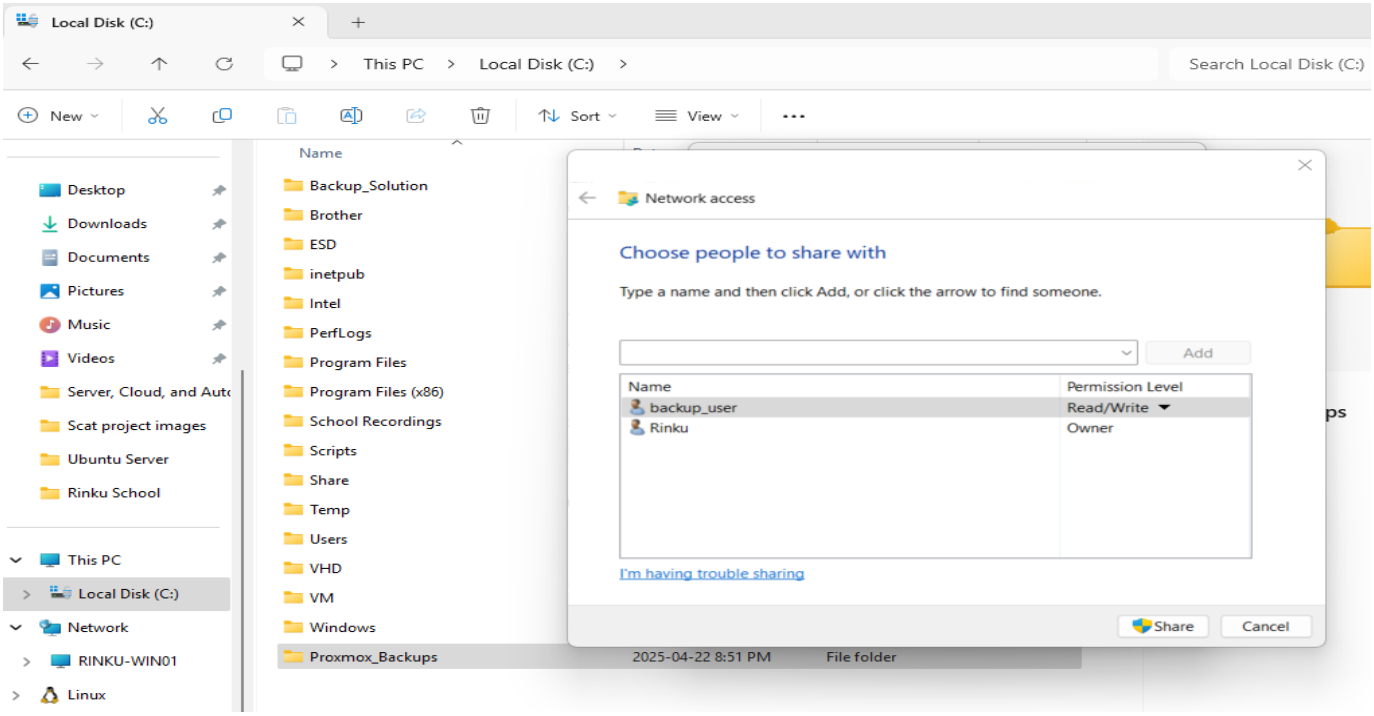
Proxmox Backup Process (SMB/CIFS Setup)

Backup_Storage	SMB/CIFS Backup Storage	192.168.2.65	backup_user	Proxmox_Backups

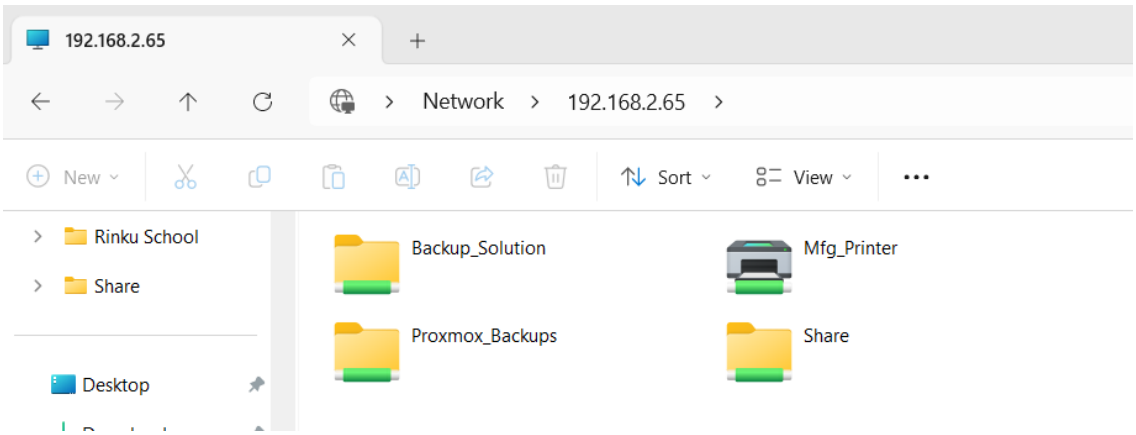
Backup IP Configuration



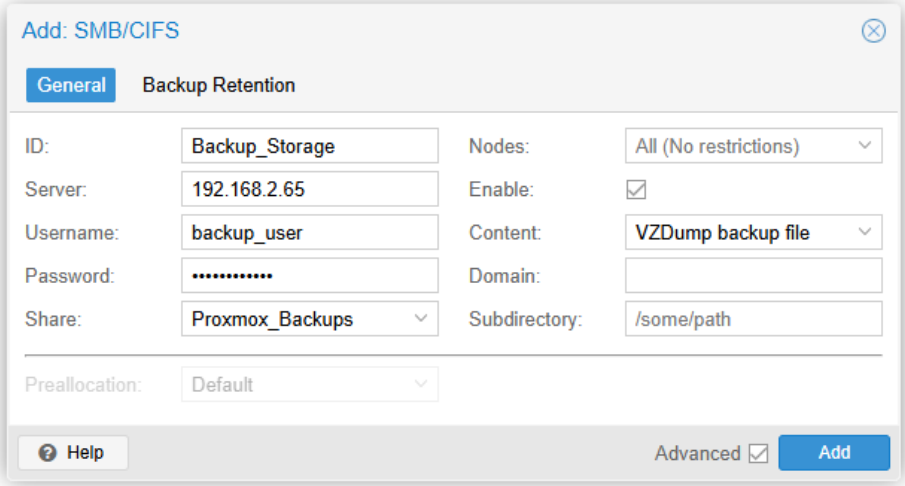
Proxmox_Backups Folder and Backup_User Configuration



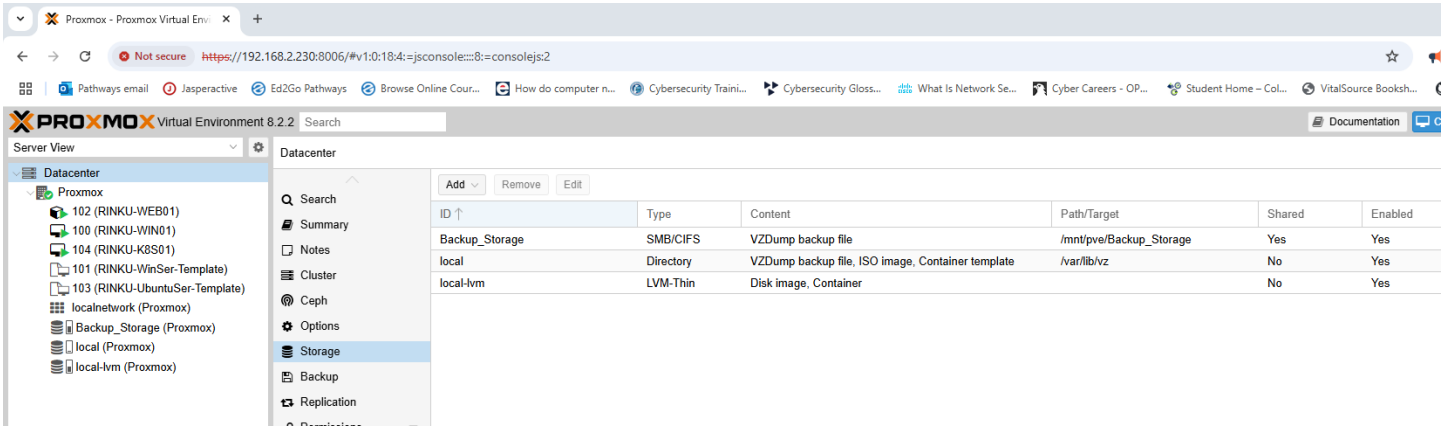
Proxmox_Backups Folder



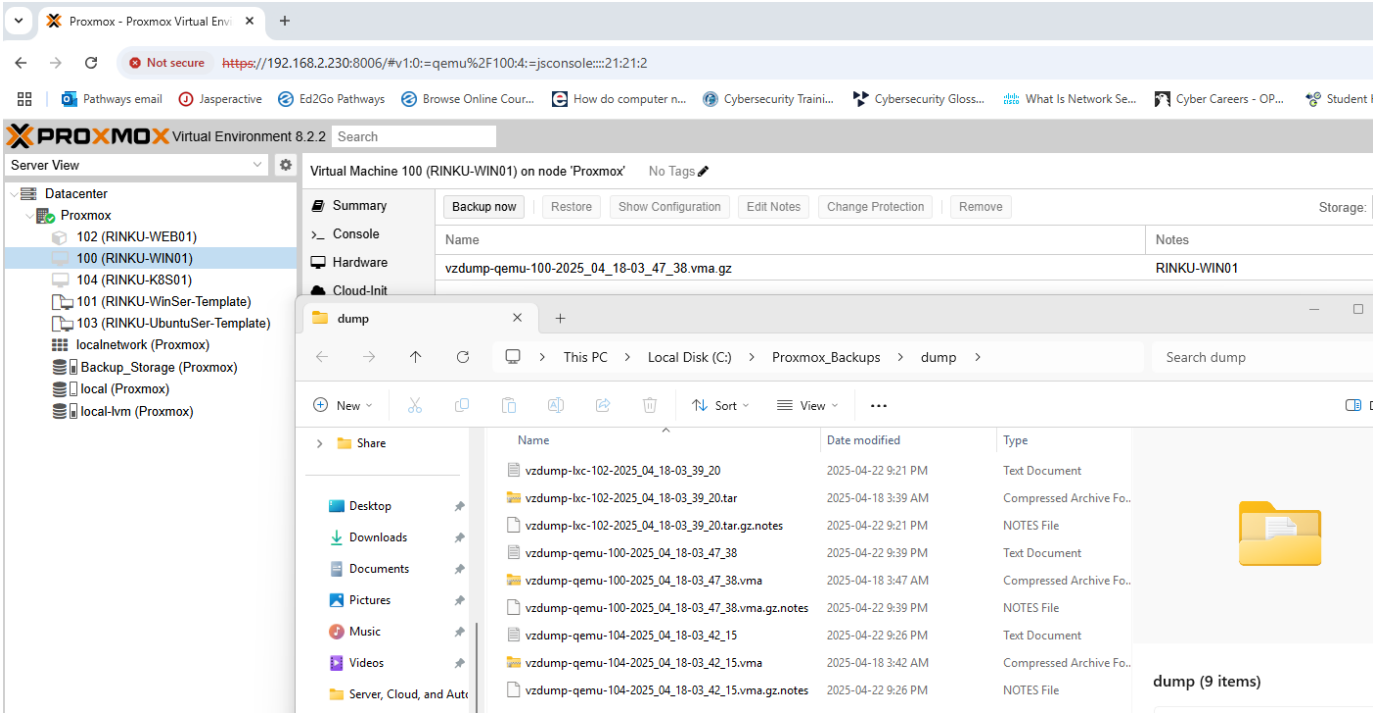
Proxmox Backup_Storage Configuration



Proxmox Backup_Storage Added



Proxmox_Backups Folder



Conclusion

This hybrid cloud project successfully integrates Proxmox, Kubernetes, and Oracle Cloud Infrastructure (OCI) to create a scalable, efficient, and highly resilient environment for hosting and managing applications. By combining on-premises virtualization with cloud-based solutions, the project ensures flexibility, automation, and improved performance. The implementation of Ansible for configuration management streamlines system deployment, while Kubernetes (k3s) enhances container orchestration and scalability. The integration of Proxmox Backup further strengthens data security and recovery strategies.

Key Findings & Learnings

- **Hybrid Cloud Efficiency**: The combination of private and public cloud components balances cost, security, and scalability.
- **Automation & Management**: Using Ansible simplifies infrastructure provisioning and configuration.
- **Containerization with Kubernetes**: K3s provides a lightweight, effective solution for deploying and managing applications.
- **Security Enhancements**: Enabling Two-Factor Authentication (2FA) on Proxmox adds an extra layer of protection for system access.

Future Enhancements

- **Advanced Monitoring**: Implement Prometheus & Grafana for real-time performance insights.
- **Infrastructure as Code (IaC)**: Expand Ansible playbooks for fully automated deployments.
- **Cloud Optimization**: Further refine OCI configurations for improved resource allocation.
- **Networking Enhancements**: Introduce Ingress and Load Balancing for optimized traffic handling.

Through strategic integration of virtualization, automation, and containerization, this hybrid cloud infrastructure is both future-ready and adaptable, ensuring a robust foundation for scaling and extending functionality.