



# Automating Infrastructure Tasks with Ansible on Azure-hosted Ubuntu Server

## Project Description:

In this project, I designed and implemented an automation workflow using **Ansible**, a powerful IT automation tool, to manage and provision an **Ubuntu-based virtual machine (VM)** hosted on **Microsoft Azure**. The control node (Ansible controller) was a **RHEL-based VM** running locally on **VMware**, while the managed node (host) was the remote Ubuntu Azure VM.

The project's objective was to simulate a real-world DevOps task of provisioning infrastructure resources and configuring systems automatically using Ansible playbooks. The initial steps involved setting up secure communication between the controller and the host using an SSH key (PEM file). Once connectivity was verified, I moved on to creating an **Ansible inventory file** and **custom YAML playbooks**.

Key tasks performed via Ansible playbooks include:

- **Installing essential packages:** Used the apt module to install commonly used system utilities such as nginx, curl, and git, ensuring that these packages were available on the remote host for development or operational use.
- **Managing services:** Leveraged the service module to start, restart, or enable services like Nginx, ensuring web servers and system daemons are running as expected.
- **Monitoring disk usage:** Applied the command module to run system-level commands (like df -h) and retrieve disk utilization data, useful for monitoring system health.
- **Creating system users:** Used the user module to create new users on the remote VM with defined properties such as shell access and home directory setup.

The playbooks were modular, well-structured, and designed with **idempotency** in mind, meaning they could be re-run safely without causing unintended changes. They also used privilege escalation (become: yes) to execute administrative-level tasks securely.

Through this hands-on automation project, I simulated a small-scale configuration management workflow, mimicking the operations a DevOps or Site Reliability Engineer might perform in a production or staging environment.



•    Azure VM setting steps

Microsoft Azure

Upgrade

Search resources, services, and docs (G+/J)

Copilot

Home > Compute infrastructure | Virtual machines >

Create a virtual machine

Help me create a low cost VM

Help me create a VM optimized for high availability

Help me choose the right VM size for my workload

Basics    Disks    Networking    Management    Monitoring    Advanced    Tags    Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. [Learn more](#)

This subscription may not be eligible to deploy VMs of certain sizes in certain regions.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

Azure subscription 1

Resource group \* ⓘ

(New) resourcegroup1

Create new

Instance details

Virtual machine name \* ⓘ

ansiblehost

Region \* ⓘ

(US) East US 2

Availability options ⓘ

Availability zone

Zone options ⓘ

☒ Self-selected zone

Choose up to 3 availability zones, one VM per zone

☐ Azure-selected zone (Preview)

Let Azure assign the best zone for your needs

Availability zone \* ⓘ

Zone 1

☒ You can now select multiple zones. Selecting multiple zones will create one VM per zone. [Learn more](#)

Image \* ⓘ

Ubuntu Server 24.04 LTS - x64 Gen2 (free services eligible)

[See all images](#) | [Configure VM generation](#)

VM architecture ⓘ

☐ Arm64

☒ x64

Run with Azure Spot discount ⓘ

☐

You are in the free trial period. Costs associated with this VM can be covered by any remaining credits on your subscription. [Learn more](#)

Size \* ⓘ

Standard\_B1s - 1 vcpu, 1 GiB memory (\$7.59/month) (free services eligible)

[See all sizes](#)



# THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

Help me copy this VM in any region

Connect Start Restart Stop Hibernate Capture Delete Refresh Open in mobile Feedback CLI / PS

## Essentials

JSON View

Resource group (move)	: <a href="#">resourcegroup1</a>	Operating system	: Linux (ubuntu 24.04)
Status	: Running	Size	: Standard B1s (1 vcpu, 1 GiB memory)
Location	: East US 2 (Zone 1)	Public IP address	: <a href="#">74.249.40.120</a>
Subscription (move)	: <a href="#">Azure subscription 1</a>	Virtual network/subnet	: <a href="#">ansiblehost-vnet/default</a>
Subscription ID	: e135e03c-672c-4d80-97e8-dac4633b1197	DNS name	: <a href="#">Not configured</a>
Availability zone	: 1	Health state	: -
Tags (edit)	: <a href="#">Add tags</a>	Time created	: 7/24/2025, 2:44 PM UTC

## Connected to azure VM via windows CMD

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

To run a command as administrator (user "root"), use "sudo <command>". See "man sudo\_root" for details.

```
thisara@ansiblehost:~$  
thisara@ansiblehost:~$  
thisara@ansiblehost:~$  
thisara@ansiblehost:~$  
thisara@ansiblehost:~$
```

## Update system packages

```
[thisara@localhost ~]$ sudo dnf update -y  
Updating Subscription Management repositories.  
Docker CE Stable - x86_64 7.6 kB/s | 3.5 kB 00:00  
Hashicorp Stable - x86_64 3.7 kB/s | 1.5 kB 00:00  
Hashicorp Stable - x86_64 1.3 MB/s | 1.8 MB 00:01  
Jenkins 2.7 kB/s | 2.9 kB 00:01  
Jenkins 51 kB/s | 126 kB 00:02  
Kubernetes 1.6 kB/s | 1.7 kB 00:01  
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs) 4.8 kB/s | 4.5 kB 00:00  
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs) 2.0 MB/s | 74 MB 00:38  
Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs) 5.1 kB/s | 4.1 kB 00:00  
Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs) 2.6 MB/s | 96 MB 00:37  
Dependencies resolved.
```

## Manually install EPEL repo RPM

```
[thisara@localhost ~]$  
[thisara@localhost ~]$ sudo dnf install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm  
Updating Subscription Management repositories.  
Last metadata expiration check: 0:14:23 ago on Thu 24 Jul 2025 08:03:02 AM PDT.  
epel-release-latest-8.noarch.rpm 15 kB/s | 25 kB 00:01  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository	Size
Installing: epel-release	noarch	8-22.el8	@commandline	25 k



## • Update metadata

```
[thisara@localhost ~]$ sudo dnf update -y
Updating Subscription Management repositories.
Extra Packages for Enterprise Linux 8 - x86_64
Last metadata expiration check: 0:00:10 ago on Thu 24 Jul 2025 08:18:51 AM PDT.
Dependencies resolved.
1.2 MB/s | 14 MB    00:12

=====
Package                                Arch      Version                                Repository                                Size
=====
Installing:
kernel                                x86_64    4.18.0-553.63.1.el8_10                rhel-8-for-x86_64-baseos-rpms            10 M
kernel-core                           x86_64    4.18.0-553.63.1.el8_10                rhel-8-for-x86_64-baseos-rpms            44 M
kernel-devel                           x86_64    4.18.0-553.63.1.el8_10                rhel-8-for-x86_64-baseos-rpms            24 M
kernel-modules                         x86_64    4.18.0-553.63.1.el8_10                rhel-8-for-x86_64-baseos-rpms            36 M
```

## • Install ansible

```
[thisara@localhost ~]$ sudo dnf install ansible -y
[sudo] password for thisara:
Updating Subscription Management repositories.

This system is registered with an entitlement server, but is not receiving updates. You can use subscription-manager to assign s
ubscriptions.

Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
Last metadata expiration check: 0:00:33 ago on Thu 24 Jul 2025 09:57:44 AM PDT.
Dependencies resolved.
1.3 MB/s | 74 MB    00:58

=====
Package                                Architecture  Version                                Repository                                Size
=====
Installing:
ansible                                noarch        9.2.0-1.el8                           epel                                        46 M
```

## • Ansible version

```
[thisara@localhost ~]$ ansible --version
ansible [core 2.16.3]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/thisara/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.12/site-packages/ansible
  ansible collection location = /home/thisara/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.11 (main, Jun 19 2025, 11:41:33) [GCC 8.5.0 20210514 (Red Hat 8.5.0-27)] (/usr/bin/python3.12)
  jinja version = 3.1.2
  libyaml = True
```



- **Create a ansible inventory file**

```
[thisara@localhost ~]$ mkdir -p ~/ansible
[thisara@localhost ~]$
[thisara@localhost ~]$
[thisara@localhost ~]$
[thisara@localhost ~]$
[thisara@localhost ~]$
[thisara@localhost ~]$ cd ~/ansible
[thisara@localhost ansible]$
[thisara@localhost ansible]$
[thisara@localhost ansible]$ mkdir -p ~/ansible
[thisara@localhost ansible]$ cd ~/ansible
[thisara@localhost ansible]$
[thisara@localhost ansible]$ echo -e "[local]\nlocalhost ansible_connection=local" > inventory
[thisara@localhost ansible]$
```

- **Test Ansible with a Ping Command**

```
[thisara@localhost ansible]$ ansible -i inventory local -m ping
localhost | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
[thisara@localhost ansible]$
```

- **Target server(host) ip configs**

```
thisara@ansiblehost:~$ ip a
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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 7c:1e:52:ad:e4:50 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.4/24 metric 100 brd 10.0.0.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::7e1e:52ff:fead:e450/64 scope link
        valid_lft forever preferred_lft forever
thisara@ansiblehost:~$ |
```



- **Test ssh connection using the private key (in my controller)**

```
[thisara@localhost Downloads]$ ssh -i ~/.ssh/ansiblehost_key.pem thisara@74.249.40.120
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.11.0-1018-azure 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.
```

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

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

```
Last login: Thu Jul 24 17:11:31 2025 from 112.135.76.166
```

```
thisara@ansiblehost:~$
```

```
thisara@ansiblehost:~$
```

- **Add host vm's details to ansible controller's inventory**

```
thisara@localhost:~/ansible
File Edit View Search Terminal Help
GNU nano 2.9.8 inventory

[azureubuntu]
74.249.40.120 ansible_user=thisara ansible_ssh_private_key_file=~/.ssh/ansiblehost_key.pem
```



- **Create playbook for install Nginx**

```
thisara@localhost:~/ansible
File Edit View Search Terminal Help
GNU nano 2.9.8 /home/thisara/ansible/install-nginx.yml

- name: Install Nginx on Azure Ubuntu VM
  hosts: azureubuntu
  become: yes
  tasks:
    - name: Update apt cache
      apt:
        update_cache: yes

    - name: Install nginx
      apt:
        name: nginx
        state: present

    - name: Ensure nginx is running and enabled
      service:
        name: nginx
        state: started
        enabled: yes
```

- **Run playbook**

```
[thisara@localhost ansible]$ ansible-playbook -i ~/ansible/inventory ~/ansible/install-nginx.yml

PLAY [Install Nginx on Azure Ubuntu VM] *****

TASK [Gathering Facts] *****
ok: [74.249.40.120]

TASK [Update apt cache] *****
changed: [74.249.40.120]

TASK [Install nginx] *****
changed: [74.249.40.120]

TASK [Ensure nginx is running and enabled] *****
ok: [74.249.40.120]

PLAY RECAP *****
74.249.40.120 : ok=4 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[thisara@localhost ansible]$
```

- **Host vm output**

```
thisara@ansiblehost:~$
thisara@ansiblehost:~$ nginx -v
nginx version: nginx/1.24.0 (Ubuntu)
thisara@ansiblehost:~$ |
```



- Create playbooks for Install multiple packages (like nginx, git, curl)

```
thisara@localhost:~/ansible
File Edit View Search Terminal Help
GNU nano 2.9.8 install-packages.yml

- name: Install multiple packages on Azure VM
  hosts: azureubuntu
  become: yes
  tasks:
    - name: Update apt cache
      apt:
        update_cache: yes

    - name: Install nginx, git, curl
      apt:
        name:
          - nginx
          - git
          - curl
        state: present

[thisara@localhost ansible]$ ansible-playbook -i ~/ansible/inventory ~/ansible/install-packages.yml

PLAY [Install multiple packages on Azure VM] *****

TASK [Gathering Facts] *****
ok: [74.249.40.120]

TASK [Update apt cache] *****
changed: [74.249.40.120]

TASK [Install nginx, git, curl] *****
ok: [74.249.40.120]

PLAY RECAP *****
74.249.40.120 : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

- All installed successfully (target host)

```
thisara@ansiblehost:~$ nginx -v
nginx version: nginx/1.24.0 (Ubuntu)
thisara@ansiblehost:~$ git --version
git version 2.43.0
thisara@ansiblehost:~$ curl --version
curl 8.5.0 (x86_64-pc-linux-gnu) libcurl/8.5.0 OpenSSL/3.0.13 zlib/1.3 brotli/1.1.0 zstd/1.5.5 libidn2/2.3.7 libpsl/0.21.2 (+libidn2/2.3.7) libssh/0.10.6/openssl/zlib nghttp2/1.59.0 librtmp/2.3 OpenLDAP/2.6.7
Release-Date: 2023-12-06, security patched: 8.5.0-2ubuntu10.6
Protocols: dict file ftp ftps gopher gophers http https imap imaps ldap ldaps mqtt pop3 pop3s rtmp rtsp scp sftp smb smb
s smtp smtps telnet tftp
Features: alt-svc AsynchDNS brotli GSS-API HSTS HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM PSL SPNEGO SSL t
hreadsafe TLS-SRP UnixSockets zstd
thisara@ansiblehost:~$
```





- **Playbook for start, stop, or restart a service**

```
thisara@localhost:~/ansible
File Edit View Search Terminal Help
GNU nano 2.9.8 restart-nginx.yml

- name: Restart nginx service on Azure VM
  hosts: azureubuntu
  become: yes
  tasks:
    - name: Restart nginx
      service:
        name: nginx
        state: restarted
```

```
[thisara@localhost ansible]$ sudo nano restart-nginx.yml
[sudo] password for thisara:
[thisara@localhost ansible]$
[thisara@localhost ansible]$
[thisara@localhost ansible]$
[thisara@localhost ansible]$ ansible-playbook -i ~/ansible/inventory ~/ansible/restart-nginx.yml

PLAY [Restart nginx service on Azure VM] *****

TASK [Gathering Facts] *****
ok: [74.249.40.120]

TASK [Restart nginx] *****
changed: [74.249.40.120]

PLAY RECAP *****
74.249.40.120 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```



- **Playbook for check disk usage**

```
thisara@localhost:~/ansible
File Edit View Search Terminal Help
GNU nano 2.9.8 check-disk.yml Modified

- name: Check disk usage on Azure VM
  hosts: azureubuntu
  gather_facts: no
  tasks:
    - name: Run df -h command
      command: df -h
      register: disk_usage

    - name: Show disk usage
      debug:
        var: disk_usage.stdout_lines
```

```
[thisara@localhost ansible]$ sudo nano check-disk.yml
[thisara@localhost ansible]$ ansible-playbook -i ~/ansible/inventory ~/ansible/check-disk.yml

PLAY [Check disk usage on Azure VM] *****

TASK [Run df -h command] *****
changed: [74.249.40.120]

TASK [Show disk usage] *****
ok: [74.249.40.120] => {
  "disk_usage.stdout_lines": [
    "Filesystem      Size  Used Avail Use% Mounted on",
    "/dev/root        29G   1.9G   27G   7% /",
    "tmpfs            424M    0   424M    0% /dev/shm",
    "tmpfs            170M 1008K  169M    1% /run",
    "tmpfs            5.0M    0   5.0M    0% /run/lock",
    "efivarfs         128K   37K   87K   30% /sys/firmware/efi/efivars",
    "/dev/sda16        881M   60M  760M    8% /boot",
    "/dev/sda15        105M   6.2M   99M    6% /boot/efi",
    "/dev/sdb1         3.9G   28K   3.7G    1% /mnt",
    "tmpfs            85M   12K   85M    1% /run/user/1000"
  ]
}

PLAY RECAP *****
74.249.40.120      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[thisara@localhost ansible]$
```



## What I Learned:

- Gained practical experience in setting up and using Ansible as an automation engine to manage Linux infrastructure.
- Understood how to write YAML-based playbooks that are clear, readable, and maintainable for configuration management.
- Learned to use various Ansible modules (apt, service, command, user, and debug) to perform system-level tasks efficiently.
- Established secure communication between controller and host machines using SSH key-based authentication, a critical step in remote DevOps workflows.
- Practiced infrastructure-as-code (IaC) principles, ensuring repeatability, automation, and consistency of system configurations.
- Enhanced my ability to troubleshoot and manage remote cloud infrastructure, particularly in Azure Linux environments.
- Developed skills in organizing Ansible projects with clean file structures and inventory management for multi-node scalability in the future.
- Learned how automation reduces manual error, improves scalability, and accelerates system provisioning and maintenance.