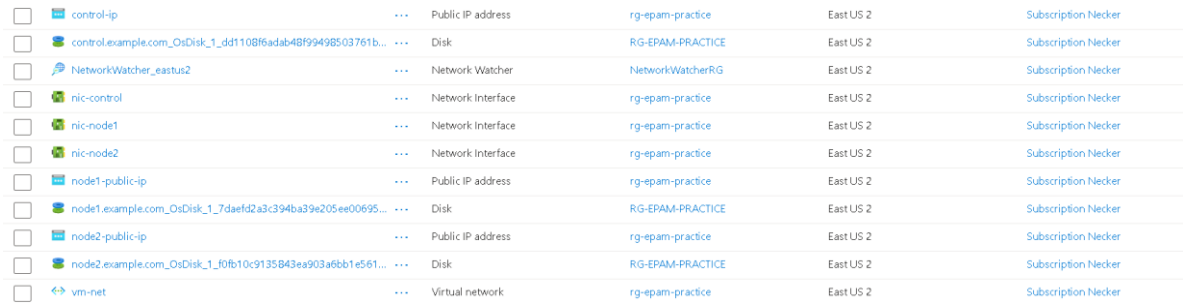
Ansible Practical Task

Azure Infrastructure:



control-ip	Public IP address	rg-epam-practice	East US 2	Subscription Necker
control.example.com_OsDisk_1_dd110Bf6adab48f99498503761b...	Disk	RG-EPAM-PRACTICE	East US 2	Subscription Necker
NetworkWatcher_eastus2	Network Watcher	NetworkWatcherRG	East US 2	Subscription Necker
nic-control	Network Interface	rg-epam-practice	East US 2	Subscription Necker
nic-node1	Network Interface	rg-epam-practice	East US 2	Subscription Necker
nic-node2	Network Interface	rg-epam-practice	East US 2	Subscription Necker
node1-public-ip	Public IP address	rg-epam-practice	East US 2	Subscription Necker
node1.example.com_OsDisk_1_7daefd2a3c394ba39e205ee00695...	Disk	RG-EPAM-PRACTICE	East US 2	Subscription Necker
node2-public-ip	Public IP address	rg-epam-practice	East US 2	Subscription Necker
node2.example.com_OsDisk_1_f0fb10c9135843ea903a6bb1e561...	Disk	RG-EPAM-PRACTICE	East US 2	Subscription Necker
vm-net	Virtual network	rg-epam-practice	East US 2	Subscription Necker

Task 1:

Setup Infra:

```
adminuser@control:~/ansible/Ansible$ ansible-playbook -i inventory.ini setup-infra.yml

PLAY [Configure infrastructure] ************************************************************************************

TASK [Ensure all nodes are in /etc/hosts] **************************************************************************
ok: [control.example.com] => (item=node1.example.com)
ok: [node2.example.com] => (item=node1.example.com)
ok: [node1.example.com] => (item=node1.example.com)
ok: [control.example.com] => (item=node2.example.com)
ok: [node2.example.com] => (item=node2.example.com)
ok: [node1.example.com] => (item=node2.example.com)
ok: [control.example.com] => (item=control.example.com)
ok: [node1.example.com] => (item=control.example.com)
ok: [node2.example.com] => (item=control.example.com)

PLAY [Configure control node ssh] **********************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [control.example.com]

TASK [Create ~/.ssh directory] *************************************************************************************
ok: [control.example.com]

TASK [Update SSH config for node* hosts] ***************************************************************************
changed: [control.example.com]

PLAY RECAP *********************************************************************************************************
control.example.com        : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
node1.example.com          : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
node2.example.com          : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Step 1: Ansible installation:

```
adminuser@control:~$ ansible --version
ansible 2.10.8
  config file = None
  configured module search path = ['/home/adminuser/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Feb  4 2025, 14:57:36) [GCC 11.4.0]
```

Step 2: Test Response using ping

```
adminuser@control:~/ansible/Ansible$ ansible -i inventory.ini nodes -m ping -u adminuser
node2.example.com | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
node1.example.com | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
adminuser@control:~/ansible/Ansible$
```

```
adminuser@control:~/ansible/Ansible$ ansible -i inventory.ini nodes -m command -a "uname -a"
node1.example.com | CHANGED | rc=0 >>
Linux node1 6.8.0-1029-azure #34~22.04.1-Ubuntu SMP Thu May  1 02:51:54 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
node2.example.com | CHANGED | rc=0 >>
Linux node2 6.8.0-1029-azure #34~22.04.1-Ubuntu SMP Thu May  1 02:51:54 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
```

```
adminuser@control:~/ansible/Ansible$ ansible -i inventory.ini nodes -m command -a "uptime"
node1.example.com | CHANGED | rc=0 >>
 01:42:30 up  1:22,  1 user,  load average: 0.00, 0.00, 0.00
node2.example.com | CHANGED | rc=0 >>
 01:42:30 up  1:22,  1 user,  load average: 0.08, 0.02, 0.01
```

```
adminuser@control:~/ansible/Ansible$ ansible -i inventory.ini nodes -m apt -a "name=htop state=present" --become
node1.example.com | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1749714832,
    "cache_updated": false,
    "changed": false
}
node2.example.com | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1749714832,
    "cache_updated": false,
    "changed": false
}
```

Step 3. Add managed nodes:

```
[control]
control.example.com ansible_host=20.186.153.18 ansible_user=adminuser

[nodes]
node1.example.com ansible_host=20.186.152.254 ansible_user=adminuser
node2.example.com ansible_host=20.36.187.36 ansible_user=adminuser

[all:vars]
ansible_ssh_common_args='-o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null'
ansible_ssh_private_key_file=~/ansible/Ansible/vm_ssh_key

[managed_nodes]
node1.example.com
node2.example.com
```

Step 4: Verify with ad hoc commands

```
adminuser@control:~/ansible/Ansible$ ansible -i inventory.ini managed_nodes -m setup -a "filter=ansible_hostname"
node2.example.com | SUCCESS => {
    "ansible_facts": {
        "ansible_hostname": "node2",
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false
}
node1.example.com | SUCCESS => {
    "ansible_facts": {
        "ansible_hostname": "node1",
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false
}
```

```
adminuser@control:~/ansible/Ansible$ ansible -i inventory.ini managed_nodes -m setup -a "filter=ansible_distribution*"
node1.example.com | SUCCESS => {
    "ansible_facts": {
        "ansible_distribution": "Ubuntu",
        "ansible_distribution_file_parsed": true,
        "ansible_distribution_file_path": "/etc/os-release",
        "ansible_distribution_file_variety": "Debian",
        "ansible_distribution_major_version": "22",
        "ansible_distribution_release": "jammy",
        "ansible_distribution_version": "22.04",
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false
}
node2.example.com | SUCCESS => {
    "ansible_facts": {
        "ansible_distribution": "Ubuntu",
        "ansible_distribution_file_parsed": true,
        "ansible_distribution_file_path": "/etc/os-release",
        "ansible_distribution_file_variety": "Debian",
        "ansible_distribution_major_version": "22",
        "ansible_distribution_release": "jammy",
        "ansible_distribution_version": "22.04",
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false
}
```

Step 5: Create a playbook for network interfaces

```
adminuser@control:~/ansible/Ansible$ ansible-playbook -i inventory.ini network_interfaces.yml

PLAY [Gather and display network interfaces] ***********************************************

TASK [Gathering Facts] *********************************************************************
ok: [control.example.com]
ok: [node1.example.com]
ok: [node2.example.com]

TASK [Display network interfaces] **********************************************************
ok: [control.example.com] => {
    "msg": "Host control.example.com has these interfaces: ['lo', 'eth0']"
}
ok: [node1.example.com] => {
    "msg": "Host node1.example.com has these interfaces: ['lo', 'eth0']"
}
ok: [node2.example.com] => {
    "msg": "Host node2.example.com has these interfaces: ['lo', 'eth0']"
}

TASK [Display detailed IP information] *****************************************************
ok: [control.example.com] => {
    "ansible_facts": {
        "all_ipv4_addresses": [
            "192.168.0.4"
        ],
        "all_ipv6_addresses": [
            "fe80::7e1e:52ff:fec6:358d"
        ],
        "ansible_local": {},
        "apparmor": {
            "status": "enabled"
        },
        "architecture": "x86_64",
        "bios_date": "03/08/2024",
        "bios_vendor": "Microsoft Corporation",
```

Task 2: Ansible role; use variables, handlers, and conditions; deploy your first software; and configure an OS with Ansible.

Step 1: Execute the playbook

```
adminuser@control:~/ansible/Ansible$ ansible-playbook -i inventory.ini common_playbook.yml

PLAY [Apply common configuration to all nodes] *********************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [node2.example.com]
ok: [node1.example.com]

TASK [common : Include package installation tasks] *****************************************************************
included: /home/adminuser/ansible/Ansible/roles/common/tasks/packages.yml for node1.example.com, node2.example.com

TASK [common : Install required packages] **************************************************************************
ok: [node2.example.com]
ok: [node1.example.com]

TASK [common : Include SELinux tasks] ******************************************************************************
included: /home/adminuser/ansible/Ansible/roles/common/tasks/selinux.yml for node1.example.com, node2.example.com

TASK [common : Check if SELinux is installed] **********************************************************************
ok: [node1.example.com]
ok: [node2.example.com]

TASK [common : Check SELinux status (when installed)] **************************************************************
skipping: [node1.example.com]
skipping: [node2.example.com]

TASK [common : Disable SELinux] ************************************************************************************
skipping: [node1.example.com]
skipping: [node2.example.com]

TASK [common : Reboot if SELinux was disabled] *********************************************************************
skipping: [node1.example.com]
skipping: [node2.example.com]

PLAY RECAP *********************************************************************************************************
node1.example.com          : ok=5    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
node2.example.com          : ok=5    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
```

Verification

Step 1: Check installed packages:

```
adminuser@control:~/ansible/Ansible$ ansible -i inventory.ini managed_nodes -m command -a "dpkg -l curl lsof mc nano ta
vim zip"
node1.example.com | CHANGED | rc=0 >>
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name           Version                      Architecture Description
+++-==============-============================-============-=================================================
ii  curl           7.81.0-1ubuntu1.20           amd64        command line tool for transferring data with URL syntax
ii  lsof           4.93.2+dfsg-1.1build2        amd64        utility to list open files
ii  mc             3:4.8.27-1                   amd64        Midnight Commander - a powerful file manager
ii  nano           6.2-1ubuntu0.1               amd64        small, friendly text editor inspired by Pico
ii  tar            1.34+dfsg-1ubuntu0.1.22.04.2 amd64        GNU version of the tar archiving utility
ii  unzip          6.0-26ubuntu3.2              amd64        De-archiver for .zip files
ii  vim            2:8.2.3995-1ubuntu2.24       amd64        Vi IMproved - enhanced vi editor
ii  zip            3.0-12build2                 amd64        Archiver for .zip files
node2.example.com | CHANGED | rc=0 >>
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name           Version                      Architecture Description
+++-==============-============================-============-=================================================
ii  curl           7.81.0-1ubuntu1.20           amd64        command line tool for transferring data with URL syntax
ii  lsof           4.93.2+dfsg-1.1build2        amd64        utility to list open files
ii  mc             3:4.8.27-1                   amd64        Midnight Commander - a powerful file manager
ii  nano           6.2-1ubuntu0.1               amd64        small, friendly text editor inspired by Pico
ii  tar            1.34+dfsg-1ubuntu0.1.22.04.2 amd64        GNU version of the tar archiving utility
ii  unzip          6.0-26ubuntu3.2              amd64        De-archiver for .zip files
ii  vim            2:8.2.3995-1ubuntu2.24       amd64        Vi IMproved - enhanced vi editor
ii  zip            3.0-12build2                 amd64        Archiver for .zip files
```

Step 2. Check SELinux status:

```
adminuser@control:~/ansible/Ansible$ ansible -i inventory.ini managed_nodes -m setup -a 'filter=ansible_selinux'
node2.example.com | SUCCESS => {
    "ansible_facts": {
        "ansible_selinux": {
            "status": "Missing selinux Python library"
        },
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false
}
node1.example.com | SUCCESS => {
    "ansible_facts": {
        "ansible_selinux": {
            "status": "Missing selinux Python library"
        },
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false
}
```

Step 3: Check if nodes were rebooted (look at uptime):

```
adminuser@control:~/ansible/Ansible$ ansible -i inventory.ini managed_nodes -m command -a "uptime"
node1.example.com | CHANGED | rc=0 >>
 03:02:11 up  2:42,  1 user,  load average: 0.00, 0.00, 0.00
node2.example.com | CHANGED | rc=0 >>
 03:02:11 up  2:42,  1 user,  load average: 0.00, 0.01, 0.00
```