Write playbooks to satisfy the following requirements. Use a host group named **poc\_hosts** for all tasks below.

# Cohort 1 Tasks:

**\*\* POC Developer Tasks (low): \*\***

1. Connectivity test:

* use a module to verify connectivity to the defined target
* use the ping module to accomplish this

1. Creating directories:

* create two directories in your user’s home directory
* the directories should be named “PoC1” and “PoC2”
* use mode 0775
* Use owner and group as user1

**\*\* POC Developer Tasks (medium)\*\***

1. Web server:

* install and enable the correct web server package for the target OS
* As the main index page, use the following for content when a visitor hits this page:
  + inventory hostname of this server
  + IP address displayed for any visitor.

**\*\* PoC Developer Tasks (complex) \*\***

1. Provision a vCenter VM with the following details: (where indicated, simply use the variable name and not a real value):

* name: ansible-poc-vm
* vCenter hostname is {{ vcenter\_hostname }}
* vCenter username is {{ vcenter\_username }}
* vCenter password is {{ vcenter\_password }}
* do not validate certs
* Datacenter is {{ vcenter\_datacenter }}
* Cluster is {{ vcenter\_cluster }}
* template to clone from: {{ vcenter\_template }}
* Folder to store {{ vcenter\_folder }}
* memory: same as the current host (may require a task to determine first)
* create a disk of 10 gb
* make the state as powered on
* tag the VM with the following tags: tag1 -> poc, tag2 -> ansible

1. Also, take a snapshot of the VM just provisioned:
   * put it in the {{ vcenter\_folder }}
   * name it as ansible-poc-snapshot

# 

# 

# Cohort 2 Tasks:

**\*\* POC Developer Tasks (low): \*\***

1. Create files:
   1. create a file with content “poc1 text” in a file named poc1.txt
   2. create a file with content “poc2 text” in a file named poc2.txt
2. Copy Files
3. copy the file poc1.txt into a new folder “PoC3”
4. copy the file poc2.txt into a new folder “PoC3”

**\*\* POC Developer Tasks (medium)\*\***

1. Gathering system information:
2. gather system information about the target host
3. save it to a file “system\_facts.json” in JSON format in the directory “PoC1”

also save these system facts to another file “system\_facts.yaml” in YAML format in the directory “PoC2”

**\*\* PoC Developer Tasks (complex) \*\***

1. Provision an AWS EC2 VM with the following details (where indicated, simply use the variable name and not a real value):

* Name it as ansible-poc-aws
* Access key is {{ aws\_access }}
* Secret key is {{ aws\_secret }}
* Ssh key is {{ ssh\_key\_name }}
* Instance type should be {{ aws\_instance\_type }}
* Security group is {{ aws\_security\_group }}
* Image to use {{ aws\_image\_id }}
* AWS Region is {{ aws\_region }}
* VPC Subnet Id is {{ aws\_vpc }}
* Enable 2 cores and 2 threads per core
* Public ip should be enabled
* state should be present
* tag the VM with the following tags: tag1 -> poc, tag2 -> ansible

1. Also, take a snapshot of the VM just provisioned

* Volume Id: {{ aws\_volume\_id }}
* AWS Region: {{ aws\_region }}
* Description: ansible-poc-snapshot