Task 1:

Configure the following:

- Create a regular user **automation** with the password of **devops**. Use this user for all challenge tasks and playbooks, unless you are working on the task #2 that requires creating the **automation** user on inventory hosts. You have root access to all servers.
- All playbooks and other Ansible configuration that you create for this challenge should be stored in `/home/automation/plays`.

Create a configuration file `/home/automation/plays/ansible.cfg` to meet the following requirements:

- The roles path should include `/home/automation/plays/roles`, as well as any other path that may be required for the course of the sample exam.
- The inventory file path is `/home/automation/plays/inventory`.
- Privilege escalation is **disabled** by default.
- Ansible should be able to manage **10 hosts** at a single time.
- Ansible should connect to all managed nodes using the **automation** user.

Create an inventory file `/home/automation/plays/inventory` with the following:

- ansible2.local is a member of the **proxy** host group.
- ansible3.local is a member of the **webservers** host group.
- ansible5.local is a member of the **database** host group.

Task 2:

Generate an SSH keypair on the control node. You can perform this step manually.

Write a script `/home/automation/plays/adhoc` that uses Ansible ad-hoc commands to achieve the following:

- User **automation** is created on all inventory hosts (not the control node).
- SSH key (that you generated) is copied to all inventory hosts for the **automation** user and stored in `/home/automation/.ssh/authorized_keys`.
- The **automation** user is allowed to elevate privileges on all inventory hosts without having to provide a password.

After running the adhoc script on the control node as the **automation** user, you should be able to SSH into all inventory hosts using the **automation** user without password, as well as a run all privileged commands.

Task 3:

Create a playbook `/home/automation/plays/sshd.yml` that runs on all inventory hosts and configures SSHD daemon as follows:

- banner is set to `/etc/motd`
- X11Forwarding is disabled
- MaxAuthTries is set to 3

Task 4:

Create Ansible vault file `/home/automation/plays/secret.yml`. Encryption/decryption password is **devops**.
Add the following variables to the vault:

- **user_password** with value of **devops**
- **database_password** with value of **devops**

Store Ansible vault password in the file `/home/automation/plays/vault_key`.

Task 5:

You have been provided with the list of users below.

Use `/home/automation/plays/vars/user_list.yml` file to save this content.

```
---

users:

  - username: thabo

    uid: 1201

  - username: vincent

    uid: 1202

  - username: sandy

    uid: 2201

  - username: patrick

    uid: 2202
```

Create a playbook `/home/automation/plays/users.yml` that uses the vault file `/home/automation/plays/secret.yml` to achieve the following:

- Users whose user ID starts with 1 should be created on servers in the **webservers** host group. User password should be used from the **user_password** variable.
- Users whose user ID starts with 2 should be created on servers in the **database** host group. User password should be used from the **user_password** variable.
- All users should be members of a supplementary group **wheel**.
- Shell should be set to `/bin/bash` for all users.
- Account passwords should use the SHA512 hash format.
- Each user should have an SSH key uploaded (use the SSH key that you created previously, see task #2).

After running the playbook, users should be able to SSH into their respective servers without passwords.

Task 6:

Create a role called **sample-mysql** and store it in `/home/automation/plays/roles`. The role should satisfy the following requirements:

- A primary partition number 1 of size 800MB on device `/dev/sdb` is created.
- An LVM volume group called `vg_database` is created that uses the primary partition created above.
- An LVM logical volume called `lv_mysql` is created of size 512MB in the volume group `vg_database`.
- An XFS filesystem on the logical volume `lv_mysql` is created.
- Logical volume `lv_mysql` is permanently mounted on `/mnt/mysql_backups`.
- **mysql-community-server** package is installed.
- Firewall is configured to allow all incoming traffic on MySQL port TCP 3306.
- MySQL root user password should be set from the variable **database_password** (see task #4).
- MySQL server should be started and enabled on boot.
- MySQL server configuration file is generated from the `my.cnf.j2` Jinja2 template with the following content:

```
[mysqld]

bind_address = {{ ansible_default_ipv4.address }}

skip_name_resolve

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock



symbolic-links=0

sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES



[mysqld_safe]

log-error=/var/log/mysqld.log

pid-file=/var/run/mysqld/mysqld.pid
```

Create a playbook `/home/automation/plays/mysql.yml` that uses the role and runs on hosts in the **database** host group.

Task 7:

Create a playbook `/home/automation/plays/selinux.yml` that runs on hosts in the **webservers** host group and does the following:

- Uses the selinux **RHEL system role**.
- Enables **httpd_can_network_connect** SELinux boolean.
- The change must survive system reboot.