

# ACS GT Training 2025 Hands-on Labs - Day 1

- LAB01 - Connection and review
  - Task 1 - Cisco AnyConnect (Secure Client)
  - Task 1 - If Cisco AnyConnect (Secure Client) is not installed
  - Task 2 - Logging into Cyber Range
  - Task 3 - Network addressing and connection review
- LAB02 - Generate and work with the VMs in the vSphere
  - Task 1 - Create Linux VM from template
  - Task 2 - Create Windows VM from template
  - Task 3 - Setup your working environment for SSH
  - Task 4 - Verify that SSH Server is installed in the Linux target VM1
  - Task 5 - Verify that SSH Server is installed in the Windows target VM2
  - Task 6 - Install OpenSSH for Windows (if not installed)
  - Task 7 - Clone the VM1 and VM2 as Templates for later vLM labs
  - Task 8 - Create SSH keypair and copy it to the server
  - Task 9 - Clone Linux VM
  - Task 10 - Clone Windows VM
  - Task 11 - Snapshots
- LAB03 - Ansible
  - Task 1 - Install Ansible
    - Install Ansible
      - Information
      - Ubuntu (applies also to Windows via WSL)
      - MacOS
  - Task 2 - Git as a code repository
    - Install Git client
    - Create your private Git repo in Gitea
    - Pull (clone) the Git repo into your Workstation
    - Modify local files and push changes back to the Gitea

## LAB01 - Connection and review

In the following labs we will be accessing the Land Cyber Training Centre (LCTC) ACS Cyber Range tenant number 80 (main tenant). The services we will be using in those labs are:

1. **vpn.ste13.com**
2. <https://vc.int.ste13.com>
3. <https://vlm.int.ste13.com>
4. <https://git.int.ste13.com>
5. <https://chat.int.ste13.com>
  - a. Training related channel:  
<https://chat.int.ste13.com/group/GT>
6. <https://isa.int.ste13.com>

There are currently 36 Slots allocated within the Training Environments 15

## Task 1 - Cisco AnyConnect (Secure Client) [🔗](#)

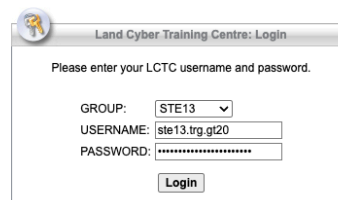
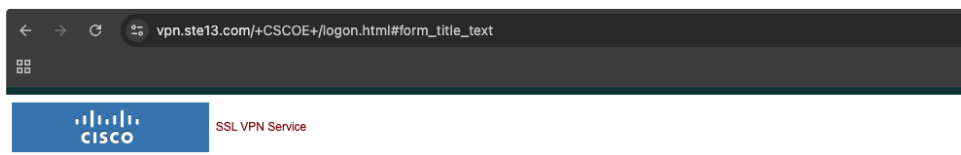
1. If Cisco Secure Client is installed launch the Secure Client software and connect to the VPN server.

 [vpn.ste13.com](https://vpn.ste13.com)

2. Continue to Task 2.

## Task 1 - If Cisco AnyConnect (Secure Client) is not installed [🔗](#)

1. If Cisco Secure Client is not installed you will be required to complete the following steps before proceeding to Task 2.
  - a. Ensure that you have administrator access or to the ability to install new software.
  - b. Open a web browser e.g Chrome and navigate to <https://vpn.ste13.com/>
  - c. Keep default group **STE13**.
  - d. In the **Username** text box, enter your username.
  - e. In the **Password** text box, enter your password and click **Login**.



- f. After a successful Login you will be presented with the correct Cisco Secure Client software to install for the Operating System of your terminal.
- g. **Download** the installation file.
- h. **Execute** the installation file.
- i. **Open** Cisco Secure Client.
- j. Continue to Task 2.

## Task 2 - Logging into Cyber Range [🔗](#)

1. We will log on using tenant specific **Green Team** credentials
  - a. Keep default group **STE13**
  - b. In the **Username** text box, enter your username
  - c. In the **Password** text box, enter your password and click **OK**

## Task 3 - Network addressing and connection review [🔗](#)

1. Verify that you can access GT resources
  - a. Ping [vc.int.ste13.com](https://vc.int.ste13.com) (172.21.1.5)
  - b. Ping <http://vlm.int.ste13.com> (172.21.3.15)

2. Verify that you can access GameNet Services resources
  - a. Ping <http://isa.int.ste13.com> (172.21.11.17)
3. Verify that you can access Training Environment 15 resources (in Slot 01)
  - a. Ping 10.1.144.1
4. Verify that you can login to Chat and send some test message in the channel:  
<https://chat.int.ste13.com/group/GT>

## LAB02 - Generate and work with the VMs in the vSphere

### Task 1 - Create Linux VM from template

The minimum requirements for creating a Linux golden image for vLM Next are:

1. Ubuntu
2. Python3
3. Netplan
4. Account with root privileges

We will generate some virtual machines to play around. Those machines are required for the following Ansible lab. This is not required for the vLM labs.

Log into **vCenter** server [vc.int.ste13.com](https://vc.int.ste13.com) using your GT (Training account) user account and password.

1. First, we need to create VM customization specification, which will configure the operating system in the way we want
  - a. Click on “**vSphere Client**” and go to “**VM Customization Specifications**” view from the Shortcuts menu
  - b. Select “**+ New...**”
  - c. Type “<UserID>-Linux-trg15-cmn-net01” to “**Name**” field
  - d. Leave the vCenter Server as the default vc.int.ste13.com
  - e. Select **Linux** as Target guest OS
  - f. Leave the default “**Generate a new security identity (SID)**” ticked
  - g. Click “**Next**”
  - h. Select “**Use the virtual machine name**” for computer name
  - i. Within the Domain name enter “**training.crp**” and click “**Next**”
  - j. Select your preferred time zone for time zone and click “**Next**”
  - k. On Customization script page click “**Next**”
  - l. Select “**Manually select custom settings**” in Network page
    - select Edit on NIC1
    - Use the following IP settings:
      - Settings = **Prompt the user for an IPv4 address when the specification is used**
      - Subnet mask = **255.255.255.0**
      - Default gateway = **10.180.144.1**
    - Click OK and “**Next**”
  - m. Type Primary DNS = **1.1.1.1**
  - n. Type “**training.crp**” to “Enter **DNS search Path**” and click “**Add**”
  - o. Click “**Next**” to continue
  - p. Review the information and click “**Finish**”
2. Click on “**vSphere Client**” and go to “**VMs and Templates**” view from the Shortcuts menu
3. Go to “**VMs and Templates**” view and select folder “**STE13-TEMPLATES**” within the “**STE13**” datacenter.

4. Find the VM named “**ste13\_template\_ubuntu\_server\_2204**”. Easiest is to use the Search bar.
5. Right-click on it and select “**Clone → Clone to Virtual Machine...**”
6. Type “**<User ID>-VM1**” as name for new VM
7. Select datacenter “**STE13**” and folder “**STE13-DEVELOPMENT**” as inventory location and click “**Next**”
8. Select Compute Resource “**STE13-Cluster-3/STE13-DEVELOPMENT-C3**” as target Compute Resource and click “**Next**”
9. Select “**acs-game-05**” as destination storage and click “**Next**”
  - a. Make sure “**Same format as source**” is select as virtual disk format (this means thin provisioning in our case as template is using thin disk)
10. Select “**Customize the operating system**” for guest customization
11. Select “**Customize this virtual machine's hardware**”
12. Select “**Power on this virtual machine after creation**” and click “**Next**”
13. Select “**<UserID>-Linux-trg15-cmn-net01**” for “**Customize Guest OS**” spec and click **Next**.
  - a. Enter IP address **10.180.144.1XX (where XX is your assigned ID)** for the VM
    - i. For example Student with ID 01 will use IP address 10.180.144.101 and Student with ID 12 will use IP address 10.180.144.112
  - b. Notice that other relevant information is already pre-filled
    - If not, use information entered for customization:
      - Subnet mask = **255.255.255.0**
      - Default gateway = **10.180.144.1**
  - c. Click “**Next**”
14. On **Customize hardware** page connect select Network adapter 1 to **ste13-training15-cmn-net01** network and click “**Next**”
15. Review the information and click “**Finish**”
16. Wait until cloning has been completed.
17. Navigate to the folder **STE13-DEVELOPMENT** and select VMs from the right hand window pane and locate the newly created VM. A green triangle denotes that the VM is powered on.
18. Select the “**<User ID>-VM1**” from inventory tree and select “**Summary**” tab on the right of the screen.
19. Wait until “**VMware Tools**” shows “**Running (Guest managed)**”
20. Open VM console (or use SSH) and log in using username “**root**” and password “**Cool2Pass**”.
21. Verify that your VM is working by listing files inside the VM. For example `ls -la` . You should see the list of files in `/root` folder.
22. Verify that your VM network is working by pinging the DNS server `1.1.1.1` `ping 1.1.1.1`
23. You can close the console window
24. Leave vSphere Client opened

## Task 2 - Create Windows VM from template

The minimum requirements for creating a Windows golden image for vLM Next are (as of March 2024):

1. OpenSSH
2. Default shell for SSH = PowerShell
3. Account with Administrative permissions

Log into **vCenter** server [vc.int.ste13.com](https://vc.int.ste13.com) using your GT (Training account) user account and password.

1. First, we need to create VM customization specification, which will configure the operating system in the way we want
  - a. Click on “**vSphere Client**” and go to “**VM Customization Specifications**” view from the Shortcuts menu
  - b. Select “**New...**”
    - i. Type “**<UserID>-Windows-trg15-cmn-net01**” to “**Name**” field

- c. Leave the vCenter Server as the default [vc.int.ste13.com](http://vc.int.ste13.com)
  - d. Keep **Windows** selected as Target guest OS and click “**Next**”
  - e. Leave the default “**Generate a new security identity (SID)**” ticked
  - f. Set Owner and Organization names as **<Username>** and **Training**
  - g. Select “**Next**”
  - h. Select “**Use the virtual machine name**” for computer name. Select “**Next**”
  - i. Select “**Next**”
  - j. For Windows license leave values as default. Select “**Next**”
  - k. Set Administrator password **Cool2Pass**
    - i. Set **Automatically logon as Administrator** count as 1
  - l. Select your preferred time zone for time zone and click “**Next**”
  - m. On **Commands to run once** page click “**Next**”
  - n. On the Network configuration page select “**Manually select custom settings**” and:
    - select Edit on NIC1
    - Use the following IP settings:
      - Settings = **Prompt the user for an IPv4 address when the specification is used**
      - Subnet mask = **255.255.255.0**
      - Default gateway = **10.180.144.1**
    - Switch to DNS tab and set Preferred DNS server as **1.1.1.1**. Click “**Ok**”
    - Select “**Next**”
  - o. Keep workgroup as WORKGROUP and click “**Next**” to continue
  - p. Review the information and click “**Finish**”
2. Click on “**vSphere Client**” and go to “**VMs and Templates**” view from the Shortcuts menu
  3. Go to “**VMs and Templates**” view and select folder “**STE13-TEMPLATES**” within the “**STE13**” datacenter
  4. Search for the template “**template\_windows\_server\_2022**”. Easiest is to use the Search bar.
  5. Right-click on it and select “**New VM from This Template...**”
  6. Type “**<User ID>-VM2**” as name for new VM
  7. Select datacenter “**STE13**” and folder “**STE13-DEVELOPMENT**” as inventory location and click “**Next**”
  8. Select Compute Resource “**STE13-DEVELOPMENT-C3**” as target Compute Resource and click “**Next**”
  9. Select “**acs-game-05**” as destination storage and click “**Next**”
    - a. Make sure “**Same format as source**” is selected as virtual disk format (this means thin provisioning in our case as template is using thin disk)
  10. Select “**Customize the operating system**” for guest customization
  11. Select “**Customize this virtual machine's hardware**”
  12. Select “**Power on this virtual machine after creation**” and click “**Next**”
  13. Select “**<UserID>-Windows-trg15-cmn-net01**” for “**Customize Guest OS**” spec.
    - a. Enter IP address **10.180.144.2XX (where XX is your assigned ID)** for the VM
      - i. For example Student with ID 01 will use IP address 10.180.144.201 and Student with ID 12 will use IP address 10.180.144.212
    - b. Notice that other relevant information is already pre-filled
      - If not, use information mentioned previously
  14. On Customize hardware page connect Network adapter 1 to **ste13-training15-cmn-net01** network and click “**Next**”
  15. Review the information and click “**Finish**”
  16. Wait until cloning has been completed.
  17. Navigate to the folder **STE13-DEVELOPMENT** and select VMs from the right hand window pane and locate the newly created VM. A green triangle denotes that the VM is powered on.

18. Select “<User ID>-VM2” from inventory tree and select “Summary” tab on the right of the screen.
19. Wait until “VMware Tools” shows “Running (Guest managed)” and IP address is 10.180.144.2XX (where XX is your assigned ID). This will take several minutes.
20. Open VM console (or use RDP) and log in using username “Administrator” and password “Cool2Pass”
21. Verify that your VM network is working by pinging 1.1.1.1 ( `ping 1.1.1.1` )

### Task 3 - Setup your working environment for SSH [🔗](#)

Before we move on to the next steps, It is important If you are using Windows on your laptop, to make sure that Windows Subsystem for Linux is installed in your workstation

1. To test if installed, run from command line:

```
wsl --status
```

2. If getting empty response, wsl is not installed. Install the wsl with the following command from elevated PowerShell prompt:

```
wsl --install
```

3. To enter to the wsl prompt on Windows open command line and execute command:

```
wsl
```

4. Verify that SSH Client is installed. Test if SSH command works inside the wsl prompt:

```
ssh
```

The command should display ssh command usage as an output.

### Task 4 - Verify that SSH Server is installed in the Linux target VM1 [🔗](#)

The next commands are expected to run from the wsl command line in Windows, BASH on Ubuntu or Terminal on MacOS

1. Try to connect to your VM1 over SSH

```
ssh root@10.180.144.1XX
```

2. If SSH server is working you will get notification about unknown SSH fingerprint. Answer “yes” for this prompt

- a. Log in by entering your root password (default password is Cool2Pass)

- b. If connecting to the servers times out, the OpenSSH is not installed

- i. Open vSphere Console into your VM VM1

- ii. On the BASH prompt enter command:

```
sudo apt install openssh-server -y
```

- iii. Try again connecting to your VMs over SSH (repeat step 1)

- c. Check if Python3 is installed (Python3 is preinstalled in Ubuntu 22.04)

```
python3 --version
```

### Task 5 - Verify that SSH Server is installed in the Windows target VM2 [🔗](#)

The next commands are expected to run from the wsl command line in Windows, BASH on Ubuntu or Terminal on MacOS

1. Try to connect to your VM2 over SSH

```
ssh administrator@10.180.144.2XX
```

2. If SSH server is working you will get notification about unknown SSH fingerprint. Answer “yes” for this prompt

3. Log in by entering your administrator password (default password is Cool2Pass)

4. Check if you can run PowerShell commands directly in the session:

```
Get-ComputerInfo
```

5. If connecting to the servers times out, the OpenSSH is not installed follow the steps in Task 6 below

### Task 6 - Install OpenSSH for Windows (if not installed) [🔗](#)

To use Ansible in Windows, install the OpenSSH Server and set SSH command line to powershell.

1. Use Microsoft Remote Desktop client to log into the VM2 over IP address: **10.180.144.2XX**
2. Open **Settings**, select **Apps**, then select **Optional Features** on the Apps tab
3. **OpenSSH Client** is already installed. If not, at the top of the page, select **Add a feature**, then:
  - Find **OpenSSH Client**, then select **Install**
  - Find **OpenSSH Server**, then select **Install**
4. Open the **Services** desktop app
  - Select **Start**
  - Type *services.msc* in the search box, and select the **Service** app or press ENTER
5. In the Services list look for **OpenSSH SSH Server** and double-click on it
6. On the **General** tab, from the **Startup type** drop-down menu, select **Automatic** and then select **Ok**
7. To start the service, right-click on it and select **Start**
8. In order for Ansible to be able to run powershell commands directly in SSH session, we need to change the SSH default shell. Run the following **Powershell** command as Administrator:

```
1 New-ItemProperty -Path "HKLM:\SOFTWARE\OpenSSH" -Name DefaultShell -Value
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -PropertyType String -Force
```

9.

- i** Once installed, you can connect to OpenSSH Server from your local machine with the OpenSSH client installed. From PowerShell, CMD or Terminal, run the following command.

```
ssh username@servername
```

For example: `ssh administrator@10.180.144.2XX` (where XX is your assigned ID)

Try to run an example powershell command, try to run `Get-ComputerInfo` directly in SSH session

## Task 7 - Clone the VM1 and VM2 as Templates for later vLM labs [🔗](#)

If your VMs are ready in this state, we will clone these as templates for using them in the later labs

1. Select "**<User ID>-VM1**" from inventory tree, right-click on it and select "**Clone → Clone to Template...**"
2. Type "**<User ID>-VM1-Template**" as name for new VM
3. Select datacenter "**STE13**" and folder "**STE13-DEVELOPMENT**" as inventory location and click "**Next**"
4. Select Resource Pool "**STE13-DEVELOPMENT-C3**" as target Compute Resource and click "**Next**"
5. Select "**acs-game-05**" as destination storage and click "**Next**"
6. Make sure "**Same format as source**" is select as virtual disk format (this means thin provisioning in our case as template is using thin disk)
7. Review the information and click "**Finish**"
8. Note that you were not asked any further questions regarding OS customization or anything else, because the Template is exact copy of the original VM with only difference that is has different name and you cannot power it on
9. Wait until cloning has been completed
10. Now do the same tasks for VM2
11. Select "**<User ID>-VM2**" from inventory tree, right-click on it and select "**Clone → Clone to Template...**"
12. Type "**<User ID>-VM2-Template**" as name for new VM
13. Select datacenter "**STE13**" and folder "**STE13-DEVELOPMENT**" as inventory location and click "**Next**"
14. Select Resource Pool "**STE13-DEVELOPMENT-C3**" as target Compute Resource and click "**Next**"
15. Select "**acs-game-05**" as destination storage and click "**Next**"

16. Make sure “**Same format as source**” is select as virtual disk format (this means thin provisioning in our case as template is using thin disk)
17. Review the information and click “**Finish**”
18. Wait until cloning has been completed

## Task 8 – Create SSH keypair and copy it to the server [🔗](#)

SSH keys are way better alternative to passwords when using SSH. To use keys for SSH, we need to have a private key and public key, which is called a keypair. The private key will always be local in your Workstation, you need to protect it and you will never copy it out from your workstation. Public key in the other hand needs to be copied to the server into which we want to connect over SSH. We will create an SSH keypair in this task and copy the public part of the keypair into our servers.

1. Check if you have any existing SSH keys

```
ls -la ~/.ssh
```

2. If the **.ssh** folder does not exist or If you only see **known\_hosts** file, then there are no SSH keys in your workstation
3. If you see any other files there, specially beginning with **id\_\*** then you have some SSH keys already generated there. Make sure you are not overwriting those keys. We will generate additional key for the next Ansible labs
4. Lets use ssh-keygen utility to generate an SSH keypair. We will use type **ed25519** which is more secure and also has smaller private key size.

```
ssh-keygen -t ed25519 -C "Cybexer lab"
```

5. Accept the default name and do not create a passphrase, by pressing enter twice
6. Check the contents of the public and private keys

```
cat ~/.ssh/id_ed25519.pub
```

```
cat ~/.ssh/id_ed25519
```

7. Lets copy the public key into our servers. For this we can use ssh-copy-id utility.
8. For **Linux** use

```
ssh-copy-id -i ~/.ssh/id_ed25519.pub root@10.180.144.1XX
```

Enter the root password to proceed

9. Lets test Linux SSH connection using SSH keys. You need to use your **private key**. You should not get any password prompt and the Bash prompt of VM1 should be given

```
ssh root@10.180.144.1XX -i id_ed25519
```

10. With Windows we need to do some additional work.

- a. Login to Windows VM2 over **RDP** (10.180.144.2XX)

- b. Open **Notepad**

- c. Copy and paste the contents of **YOUR public** key. For example:

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIAYbLfNW4/jw2Pv0eVc84z56G61iCHxipwA9jBiCKk1g Cybexer lab
```

- d. Select File → Save As

- e. Select folder (you need to type it in the address bar, as its hidden folder and not visible when browsing the location)

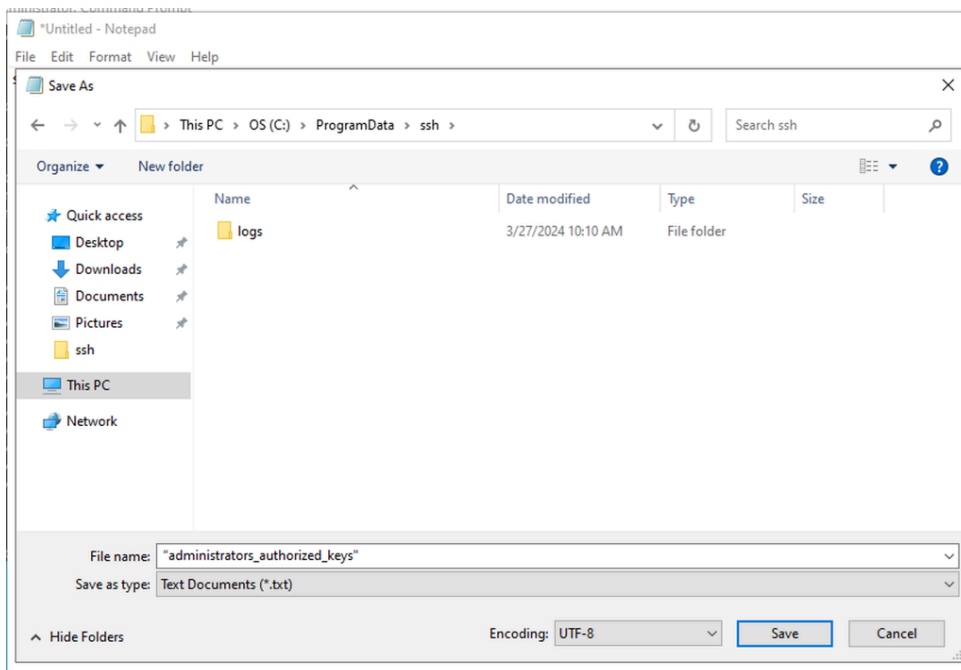
```
C:\ProgramData\ssh
```

Press Enter to go into the folder above

- f. As File name enter with double quotes: `"administrators_authorized_keys"` (this will generate the file name without .txt extension which is needed for us)

If you get error saying that name is not valid, you probably copied the string from the lab guide. Delete the double quotes and add them again.





g. Click **Save** and close the Notepad

h. Open **Command Prompt**

i. Navigate to `cd C:\ProgramData\ssh`

j. Update the permissions on the authorized keys file

```
icacls.exe "C:\ProgramData\ssh\administrators_authorized_keys" /inheritance:r /grant "Administrators:F" /grant "SYSTEM:F"
```

k. Close the Command Prompt and close the **RDP** session

l. Now test if you can SSH into the VM2 without password and using only **private key**

```
ssh -i ~/.ssh/id_ed25519 administrator@10.180.144.2XX
```

m. You should have SSH access without any prompt or password

## Task 9 - Clone Linux VM [🔗](#)

In this task we will clone our VMs into new additional VMs.

1. Select "**<User ID>-VM1**" from inventory tree, right-click on it and select "**Clone → Clone to Virtual Machine ...**"
2. Type "**<User ID>-VM10**" as name for new VM
3. Select datacenter "**STE13**" and folder "**STE13-DEVELOPMENT**" as inventory location and click "**Next**"
4. Select Resource Pool "**STE13-DEVELOPMENT-C3**" as target Compute Resource and click "**Next**"
5. Select "**acs-game-05**" as destination storage and click "**Next**"
6. Make sure "**Same format as source**" is select as virtual disk format (this means thin provisioning in our case as template is using thin disk)
7. Select "**Customize the operating system**" for guest customization
8. Select "**Power on this virtual machine after creation**" and click "**Next**"
9. Select "**<UserID>-Linux-trg15-cmn-net01**" for "**Customize Guest OS**" spec.
10. Enter IP address **10.180.144.1XX +30 (where XX is your assigned ID)** for the VM (For example for Student01 the IP will be 10.180.144.131 and Student12 will be 10.180.144.142)
11. Notice that other relevant information is already pre-filled, click "**Next**"
12. Review the information and click "**Finish**"

13. Note that you were not asked any further questions regarding OS customization
14. Wait until cloning has been completed and your VM powers on.
15. Select “<User ID>-VM10” from inventory tree and select “Summary” tab on the right of the screen
16. Wait until “VMware Tools” shows “Running (Guest managed)”
17. Open VM console or use SSH and log in using username “root” and using your private key (accept the host fingerprint)  

```
ssh -i ~/.ssh/id_ed25519 root@10.102.144.1XX+30
```
18. Exit from SSH by typing  

```
exit
```

## Task 10 - Clone Windows VM [🔗](#)

In this task we will clone our VMs into new additional VMs.

1. Select “<User ID>-VM2” from inventory tree, right-click on it and select “Clone → Clone to Virtual Machine ...”
2. Type “<User ID>-VM20” as name for new VM
3. Select datacenter “STE13” and folder “STE13-DEVELOPMENT” as inventory location and click “Next”
4. Select Resource Pool “STE13-DEVELOPMENT-C3” as target Compute Resource and click “Next”
5. Select “acs-game-05” as destination storage and click “Next”
6. Make sure “Same format as source” is select as virtual disk format (this means thin provisioning in our case as template is using thin disk)
7. Select “Customize the operating system” for guest customization
8. Select “Power on this virtual machine after creation” and click “Next”
9. Select “<UserID>-Windows-trg15-cmn-net01” for “Customize Guest OS” spec.
10. Enter IP address 10.180.144.2XX + 30 (where XX is your assigned ID) for the VM (For example for Student01 the IP will be 10.180.144.231 and Student12 will be 10.180.144.242)
11. Notice that other relevant information is already pre-filled, click “Next”
12. Review the information and click “Finish”
13. Note that you were not asked any further questions regarding OS customization
14. Wait until cloning has been completed and your VM gets powered on
15. Select “<User ID>-VM20” from inventory tree and select “Summary” tab on the right of the screen
16. Wait until “VMware Tools” shows “Running (Guest managed)”
17. Open VM console or use SSH and log in using username “administrator” and using your private key (accept the host fingerprint)  

```
ssh -i ~/.ssh/id_ed25519 administrator@10.180.144.2XX+30
```
18. Exit from SSH by typing  

```
exit
```

## Task 11 - Snapshots [🔗](#)

1. Select “<User ID>-VM1” from inventory tree and open the VM console or use SSH
2. Log into your VM using username “root” (password is still Cool2Pass)
3. Check current directory using command `pwd`. You should be in folder `/root`
4. List files in current directory using command `ls -la`
5. Create new file using the following command `echo "before snapshot" > test.txt`
6. List files in current directory again using command “ls”. Note that new file “test.txt” was created
7. List the file content using command `cat test.txt`
8. Leave console window opened and select “<User ID>-VM1” from the inventory tree again
9. Right-click on VM1 and select “Snapshots → Take snapshot...” from dropdown menu

10. Type „**snapshot1**“ for snapshot name
11. Select “**Include the virtual machine’s memory**”
12. Click “**CREATE**”
13. Wait until snapshot creation task completes. It takes some time to create memory snapshot
14. Open **VM console** again
15. Make sure your previously created file still exists using command `ls`
16. Delete the file using command `rm test.txt`
17. Make sure the file is missing using command `ls`
18. Leave console window opened and select “**<User ID>-VM1**” from inventory tree again
19. Right-click on VM and select “**Snapshots → Revert to Latest Snapshot**” from dropdown menu. Confirm by clicking “**Revert**”
20. Wait until revert task completes; it takes few seconds
21. Open **VM console** again (it might disconnect, so you have to close the previous console window connect again)
22. Check the existing files using command `ls`. Is the deleted file back?
23. Check the file content using command `cat test.txt`. Does it match the previous content?
24. Close the console window and select “**<User ID>-VM1**” from inventory tree, right-click on it and select “**Snapshots → Manage Snapshots**” from dropdown menu
25. Select your snapshot (“**snapshot1**”) from list and click “**Delete**”, when prompted click the red “**Delete**” button again.

## LAB03 - Ansible [🔗](#)

The aim for this lab is to demonstrate the basics of an Ansible workflow. We will use the VMs created in the previous labs and we will install a web server and MySQL server. We will configure Windows server, add users and install IIS. In later labs we will use the similar playbooks with vLM.

Start by installing Ansible directly into your machine

### Task 1 - Install Ansible [🔗](#)

#### Install Ansible [🔗](#)

##### Information [🔗](#)

This guide also applies to Windows. If you are using Windows, make sure that Windows Subsystem for Linux is installed.

To test if installed, run from command line:

```
wsl --status
```

If getting empty response, wsl is not installed. Install the wsl with the following command from elevated PowerShell prompt:

```
wsl --install
```

#### Ubuntu (applies also to Windows via WSL) [🔗](#)

To install Ansible on an Ubuntu system, follow these steps:

1. Update the System
  - a. First, update your system's package index. **Open a terminal** and run

```
sudo apt update
```

2. Install Software Properties Common. This package provides an abstraction of the used apt repositories. Run:

```
sudo apt install software-properties-common
```

3. Add Ansible PPA

Personal Package Archives (PPA) are repositories hosted on Launchpad. You can add the official Ansible PPA by running:

```
sudo add-apt-repository ppa:ansible/ansible
```

#### 4. Update the System Again

Update the package index again to include the newly added PPA.

```
sudo apt update
```

#### 5. Install Ansible. Now, install Ansible using apt.

```
sudo apt install ansible -y
```

#### 6. Verify Installation. Check the installed version to verify that Ansible is installed correctly.

```
ansible --version
```

### MacOS [🔗](#)

On MacOS, the easiest way to install Ansible is using Homebrew, a package manager for MacOS.

#### 1. **Install Homebrew:** If you don't have Homebrew installed, **open a terminal** and run:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

#### 2. **Install Ansible:** Once Homebrew is installed, you can install Ansible by running:

```
brew install ansible
```

#### 3. **Verify Installation:** Check the installed version to ensure Ansible is correctly installed.

```
ansible --version
```

---

## Task 2 – Git as a code repository [🔗](#)

### Install Git client [🔗](#)

In order to use Git, we need to install Git client into our Workstation

#### 1. In the wsl command prompt, Bash shell in Ubuntu or Terminal in MacOS run command:

```
which git
```

#### 2. If you get output for the location of Git binary, then you have Git installed and you can try to run the basic Git command:

```
git
```

#### 3. If you do not receive a reply from the command in step 1, you will need to install Git

##### a. In wsl or Bash type:

```
sudo apt update
```

```
sudo apt install git -y
```

##### b. In MacOS use:

```
brew install git
```

#### 4. Test if basic Git command gives any output:

```
git
```

### Create your private Git repo in Gitea [🔗](#)

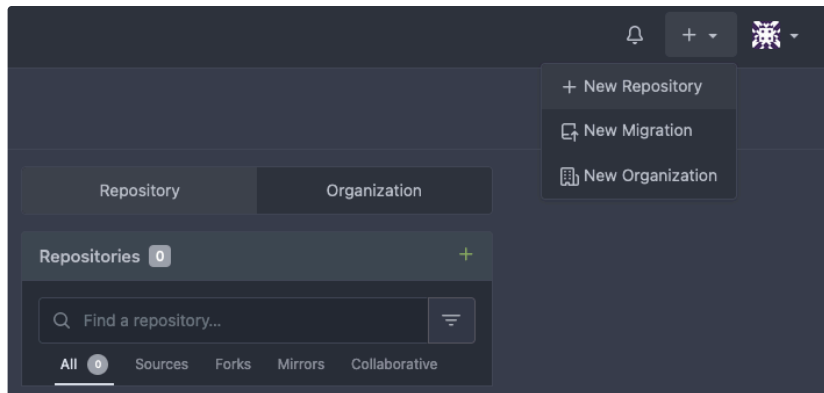
We will create a personal Git repository in our tenant Git service.

#### 1. Use web browser and connect to

<https://git.int.ste13.com>

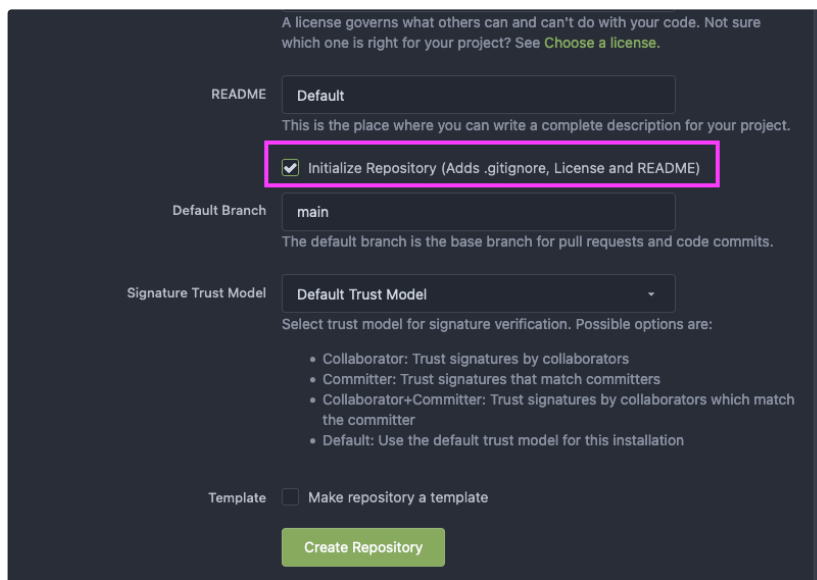
#### 2. Log in with your Cyber Range credentials

#### 3. On the right side of the screen click plus sign (“+”) and select “+ New Repository”



4. Repository name: “**crp-training**”

5. Select “**Initialize Repository**” and click “**Create Repository**”



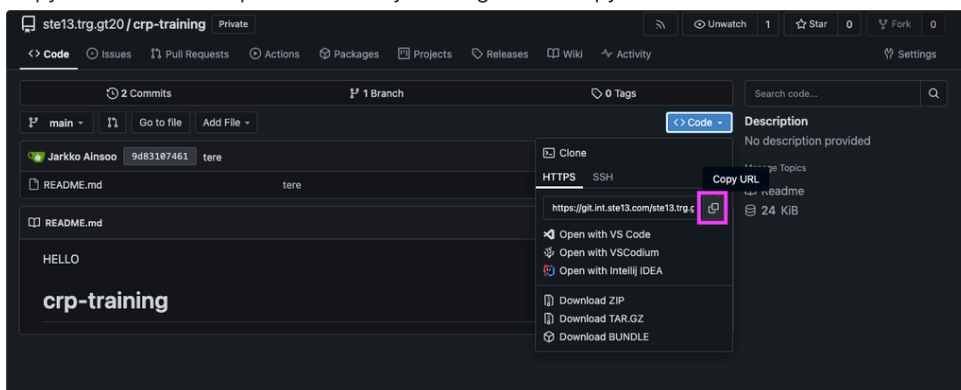
6. You have now generated your private Git repository which we will use during the Ansible training

## Pull (clone) the Git repo into your Workstation [🔗](#)

In order to modify files in our Git repository we need to have a local copy of the repository in our Workstation. In order to pull the repository for the first time, we need to clone it.

1. Copy the repository address from Gitea

- In Gitea UI find your repository (should be already open after the previous Lab)
- Copy the address of repo as HTTPS by clicking on the Copy URL button



2. Open **wsl** in your Workstation **command prompt** and run the git clone command (replace the username in the URL below with your username)

```
git clone https://git.int.ste13.com/<username>/crp-training.git
```

3. Enter your valid Cyber Range credentials

4. If the pull was successful, you will see a new folder generated named “**crp-training**” into the location where you ran the command. Check using

```
ls
```

```
cd crp-training
```

5. You are now in possession of a local copy of the repository that you created on your Workstation. There is currently only one file - “[README.md](#)” in the repository which was created when the repo was initialised.

## Modify local files and push changes back to the Gitea [🔗](#)

We will now modify the [README.md](#) file locally in our Workstation and then push the changes back to main repository (origin)

First we need to setup our Git client in order to allow updating the origin

1. We need to add user and email addresses to the config

```
git config --global user.name "First Last"
```

```
git config --global user.email "somebody@somewhere.net"
```

Now we are ready to work with the Git

1. Make sure you are in the root folder of the repository. If you type ls you need to see only the [README.md](#) file

```
1 user@jab:~/crp-training$ ls
2 README.md
```

2. Let's run a basic and useful Git command. Check the status of your Git repo

```
git status
```

3. Modify the [README.md](#) file

a. nano README.me

b. Write something below the first line, like “**Hello World**”

c. Exit Nano with **CTRL + X**

d. Type **Y** in the Save dialog

e. Press enter for accepting the same file name

4. Run `git status` command again to see that you now have one modified file. The file is not staged for committing in the repo

5. Stage the [README.me](#) file to be included in the next git commit

```
git add README.md
```

6. Run `git status` command again to see that file is now added to the commit list

7. Set up the [README.md](#) file to be included in a commit. This will commit all the changes you have added to any files in your repository

```
git commit -m "Updated readme file, initial commit"
```

8. Run `git status` command again to see that now your local branch is ahead of 1 commit when comparing with the origin

9. Let's send the commit to Gitea

```
git push origin main
```

10. You will see the following output

```
1 user@jaub:~/crp-training$ git push origin main
2 Username for 'https://git.int.ste13.com': <username>
3 Password for 'https://ste13.trg.gt<id>@git.int.ste13.com':
4 Enumerating objects: 5, done.
5 Counting objects: 100% (5/5), done.
```

```
6 Writing objects: 100% (3/3), 286 bytes | 286.00 KiB/s, done.  
7 Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
8 remote: . Processing 1 references  
9 remote: Processed 1 references in total  
10 To https://git.int.stel3.com/stel3.trg.gt<id>/crp-training.git  
11    34374ec..31fa062  main -> main
```

11. Your changes are now in the Gitea server. Refresh your browser in Gitea UI and you will see the changes in the [README.md](#) file