HO1424 – What's New in SLE 15

Important Information:

RMT/NTP Server IP Address: 192.168.123.174

SLES 12 VM IP Address:

openSUSE Leap 15 VM IP Address:

SLES 15 VM IP Address: 192.168.123.225

Username: geeko
Password: linux

Username: root
Password: linux

Lab 1 – Build and Migrate

1.1 Build a SLE VM from Installer

1. Ensure the RMT VM is up and running correctly before proceeding.
2. From the lab machine, connect to the RMT VM:

   ```
   $ ssh root@<rmt>
   ```

3. Because of the lab environment, the RMT certificate needs to be regenerated. First stop the running services:

   ```
   # systemctl stop nginxrmt-server
   ```

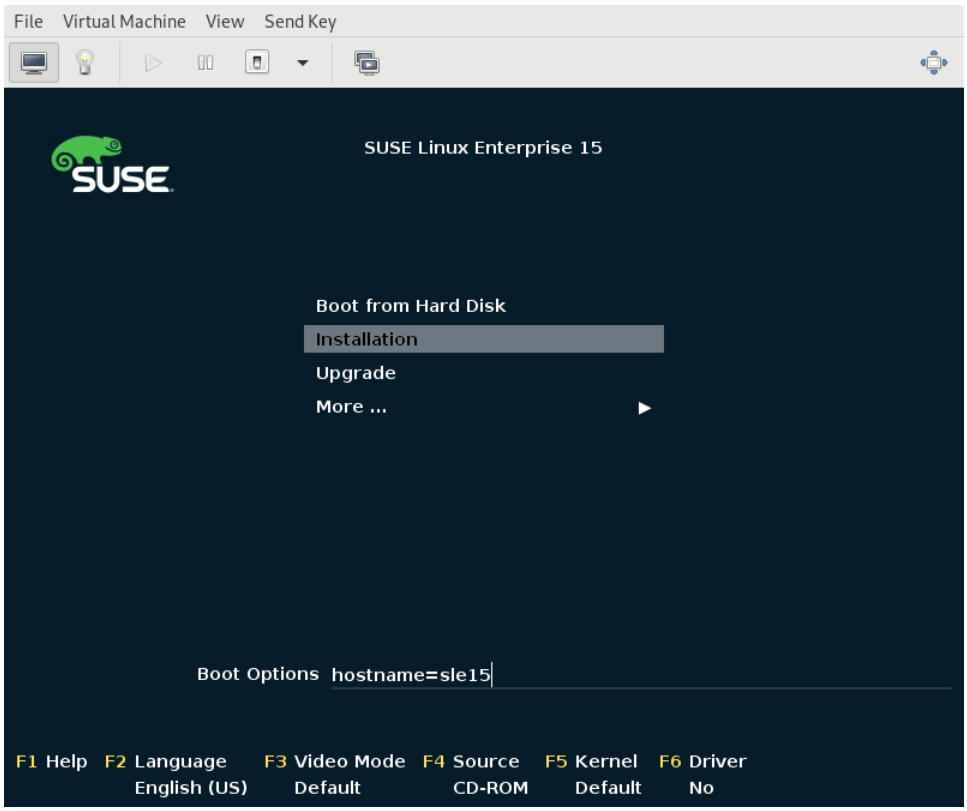
4. Remove previously generated CA and HTTPS certificates:

   ```
   # rm /etc/rmt/ssl/rmt-.*
   ```

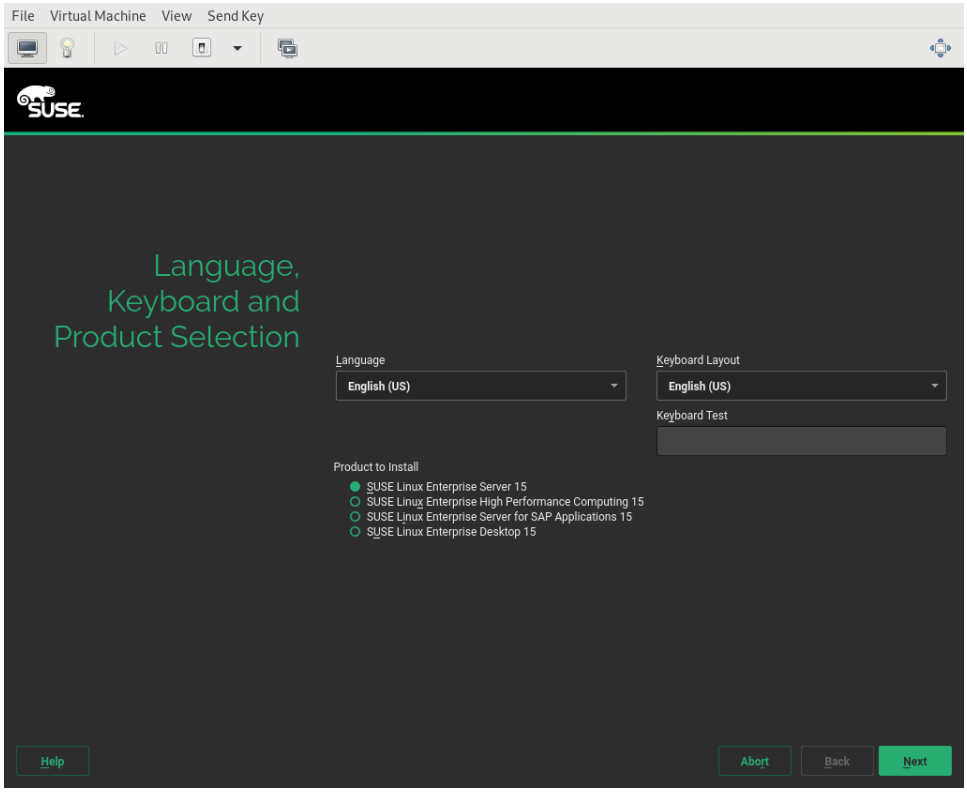
5. Re-run the RMT server configuration, skipping most steps;

   ```
   # yast rmt
   ```

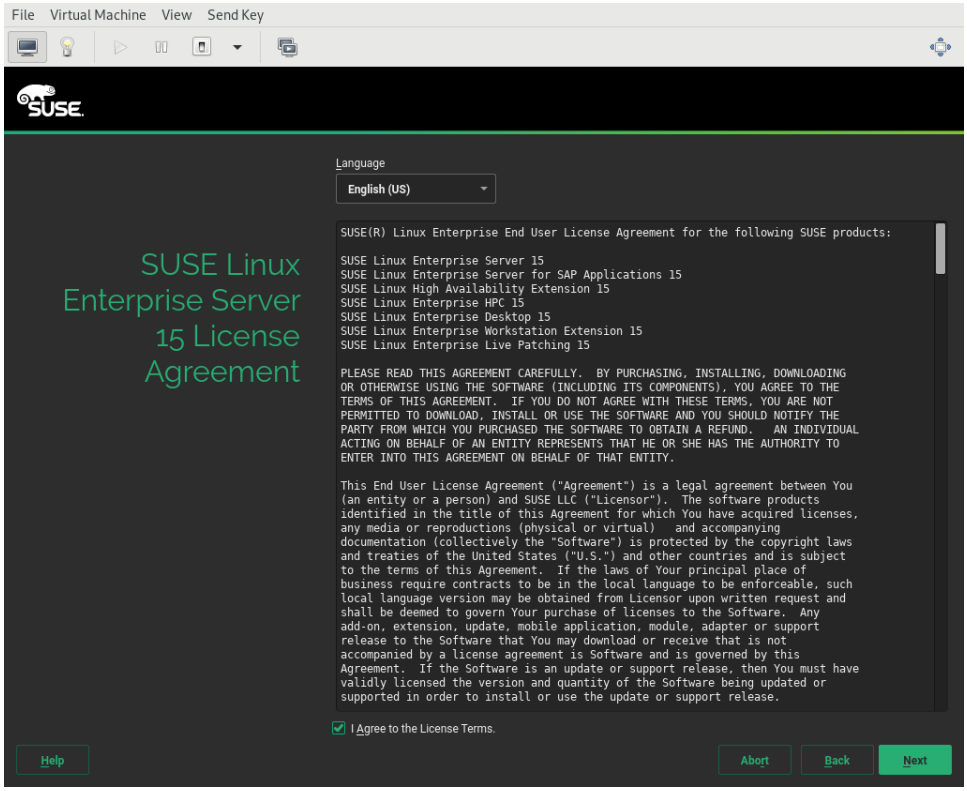
6. Enter nothing in the organization credentials and select “Next”. (We don’t want to actually mirror from SUSE.com for this lab.)
7. At the prompt “Configuration written successfully.” select “OK”.
8. Enter nothing for Database Password and select “Next”.
9. At the prompt “Configuration written successfully.” select “OK”.
10. In SSL Certificate Generation select “Next”.
11. Enter “password” for both Password and Confirm Password and select “OK”.
12. On the firewall configuration select “Next”.
13. Select “Next”.
14. Select “Finish”.
15. In virt-manager, boot the sle15 VM from the DVD.
16. Select Installation, enter “hostname=sle15” in Boot Options, and hit Return.



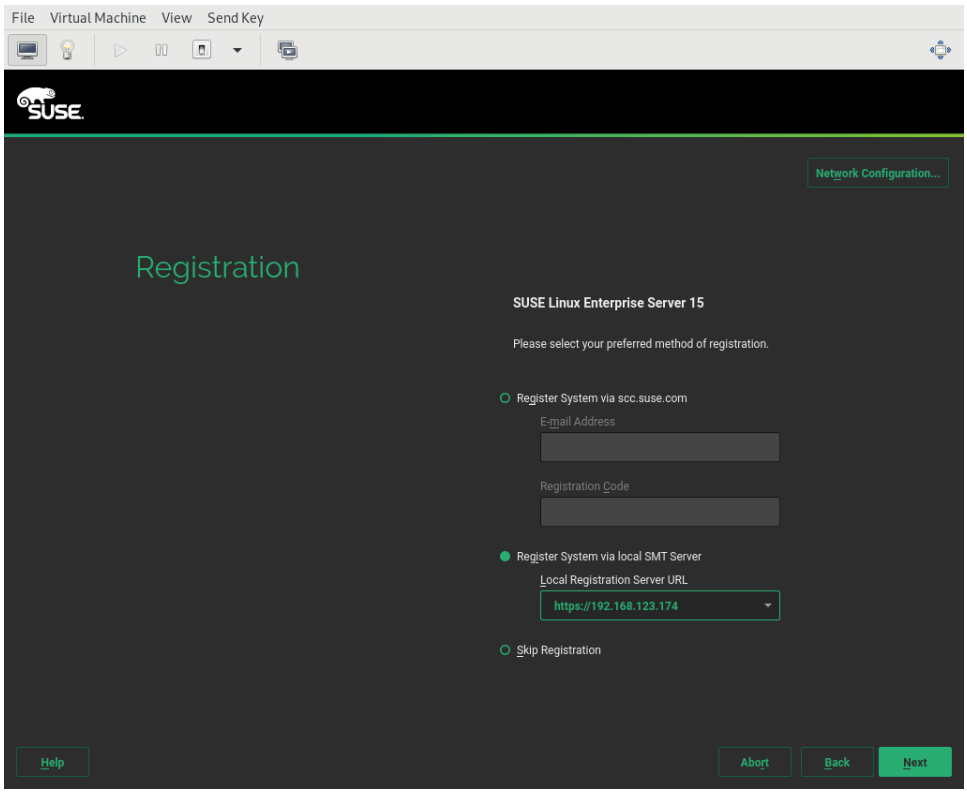
17. Select “SUSE Linux Enterprise Server 15” and click Next in Language, Keyboard and Product Selection.



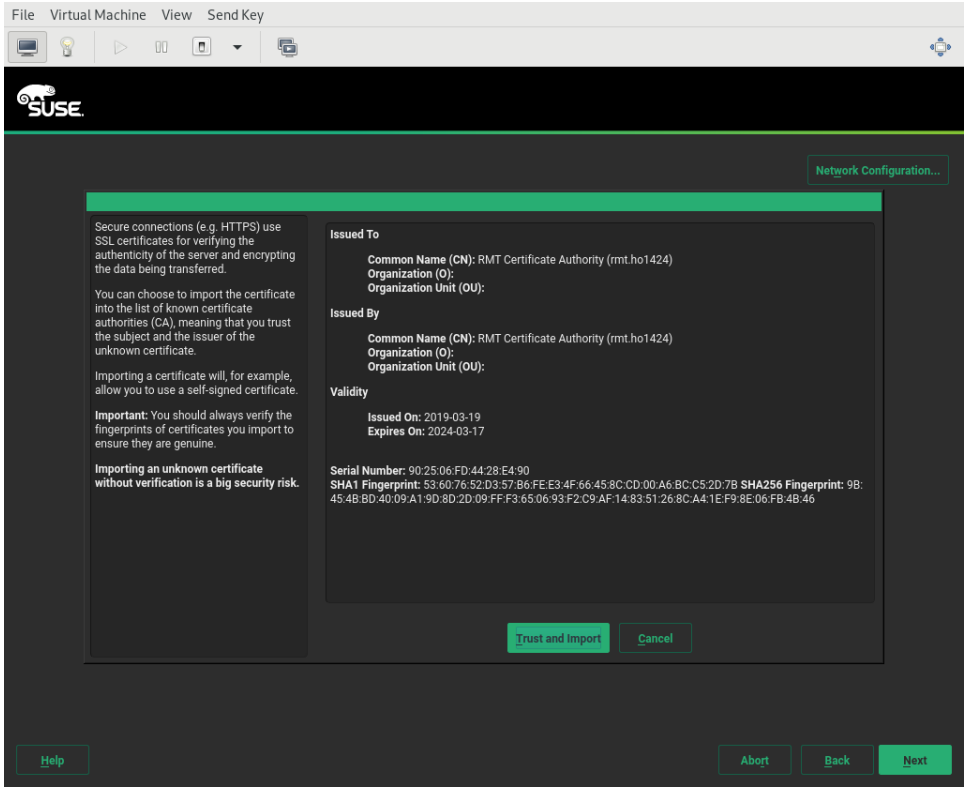
18. Check “I Agree to the License Terms” and click Next in License Agreement.



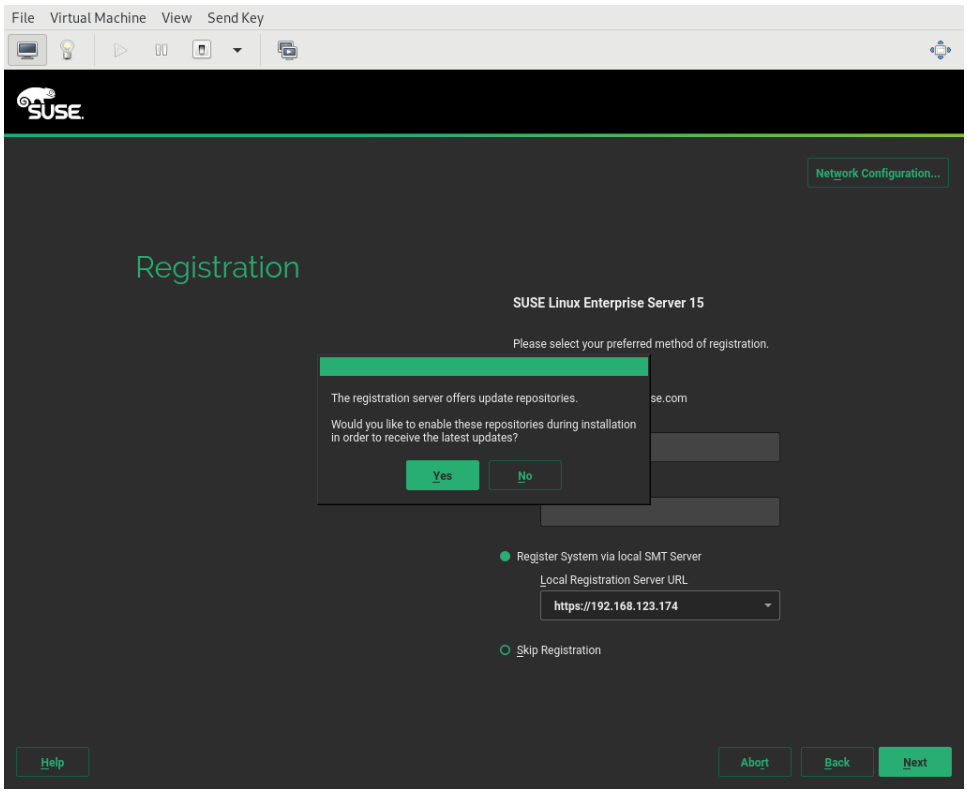
19. Check “Register System via local SMT Server”, enter https://<rmt> in Local Registration Server URL, and click Next on Registration.



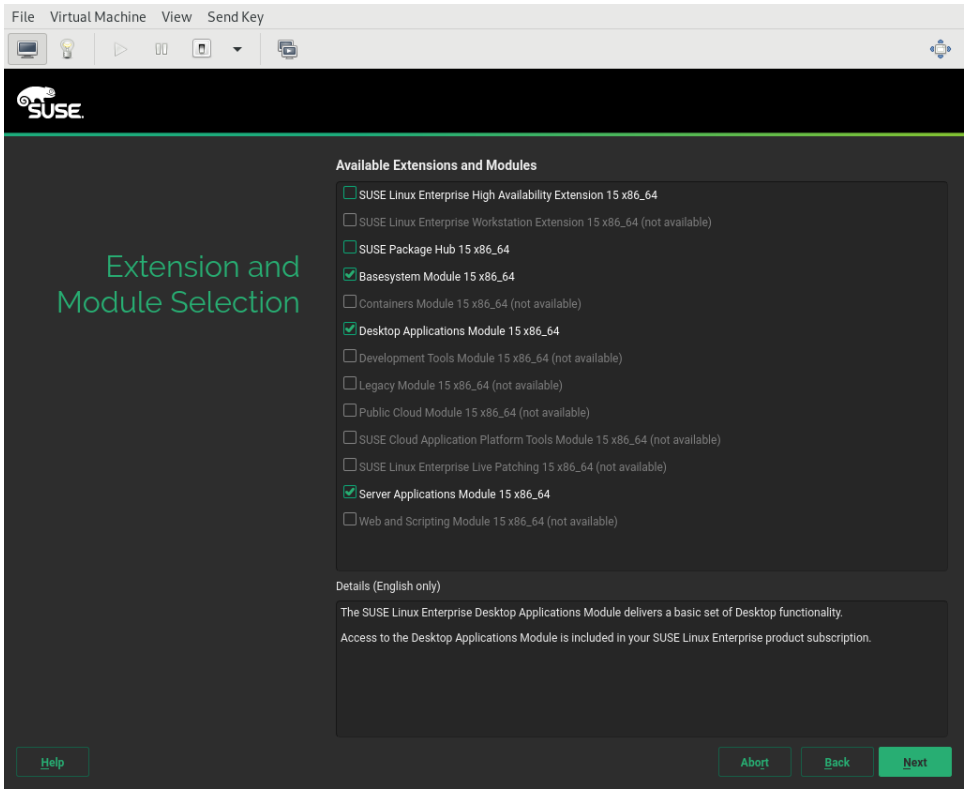
20. Click Trust and Import to accept the self-signed certificate.



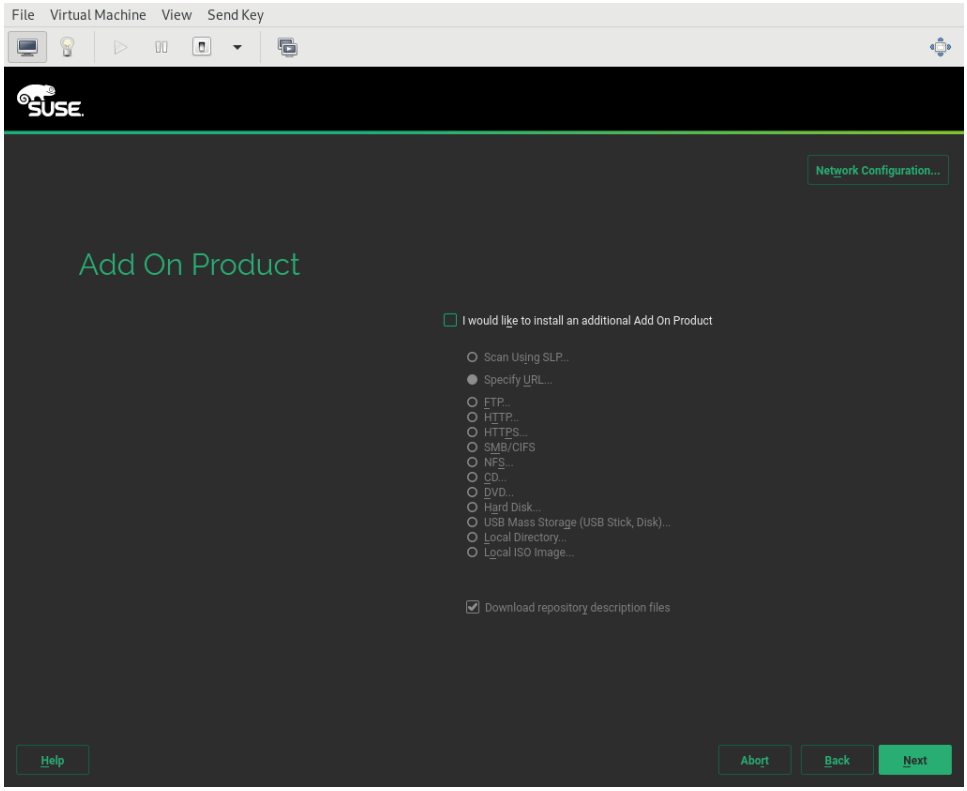
21. Click Yes to enable update repositories.



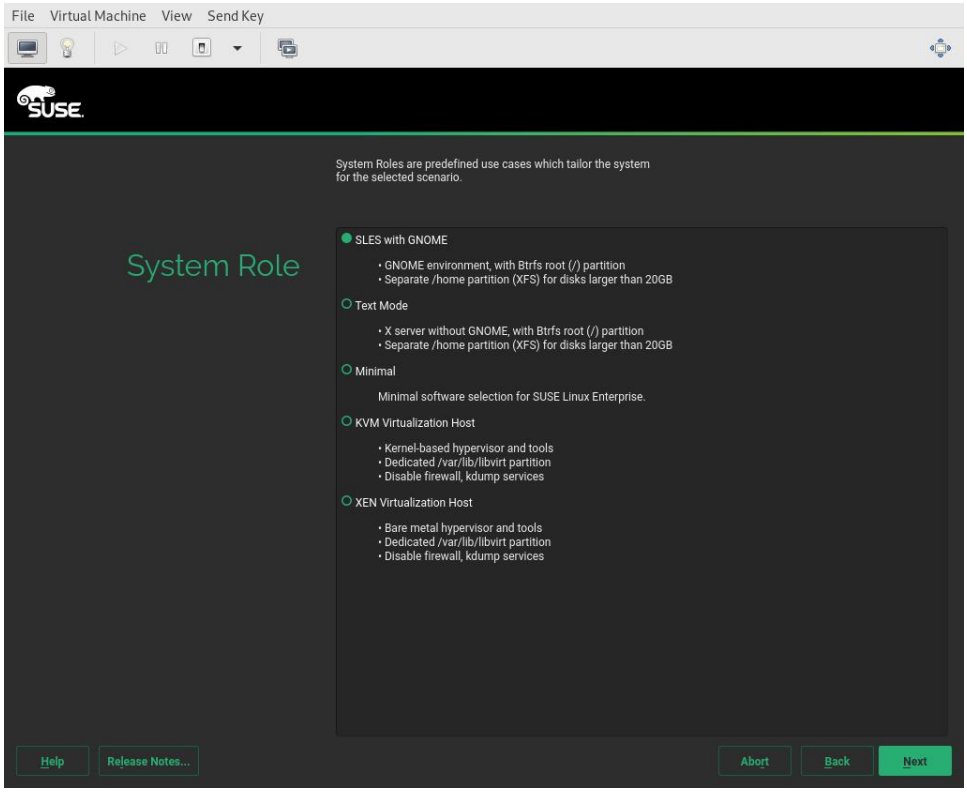
22. Select "Desktop Applications Module 15 x86_64" and click Next in Extension and Module Selection.



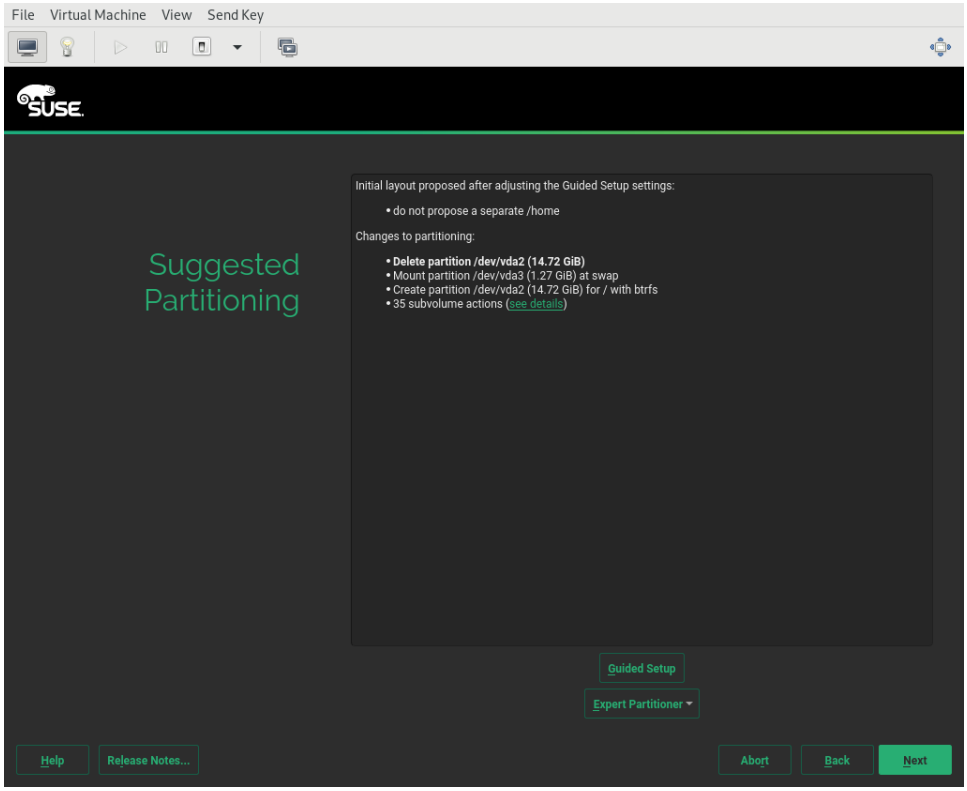
23. Click Next in Add On Product.



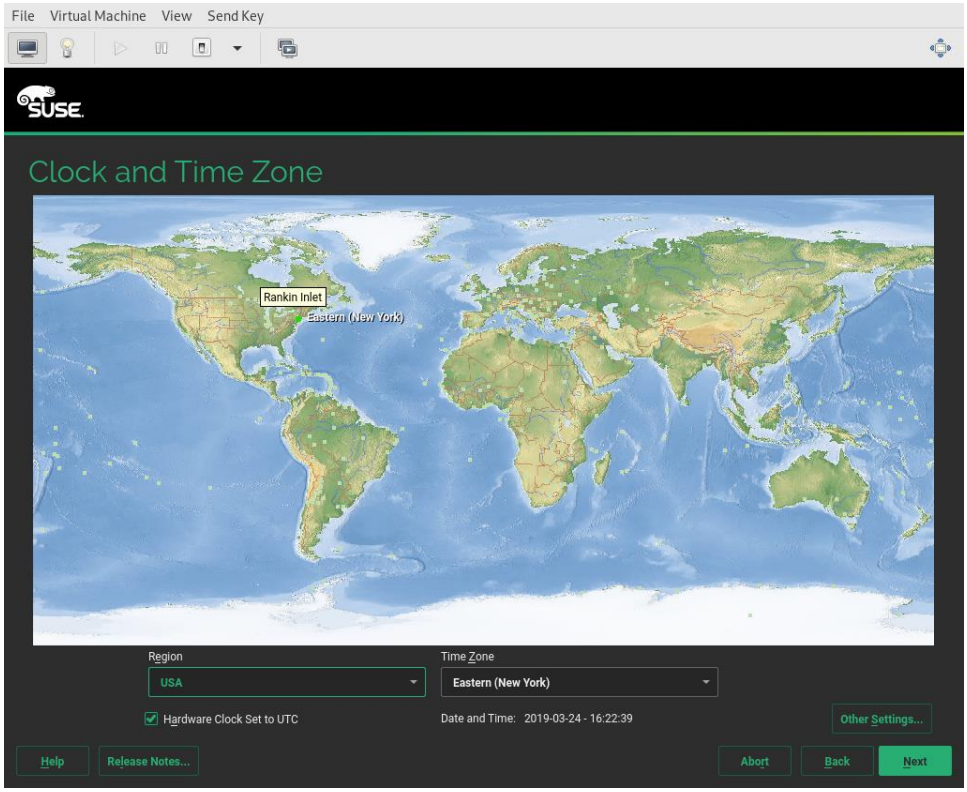
24. Select “SLES with GNOME” and click Next in System Role.



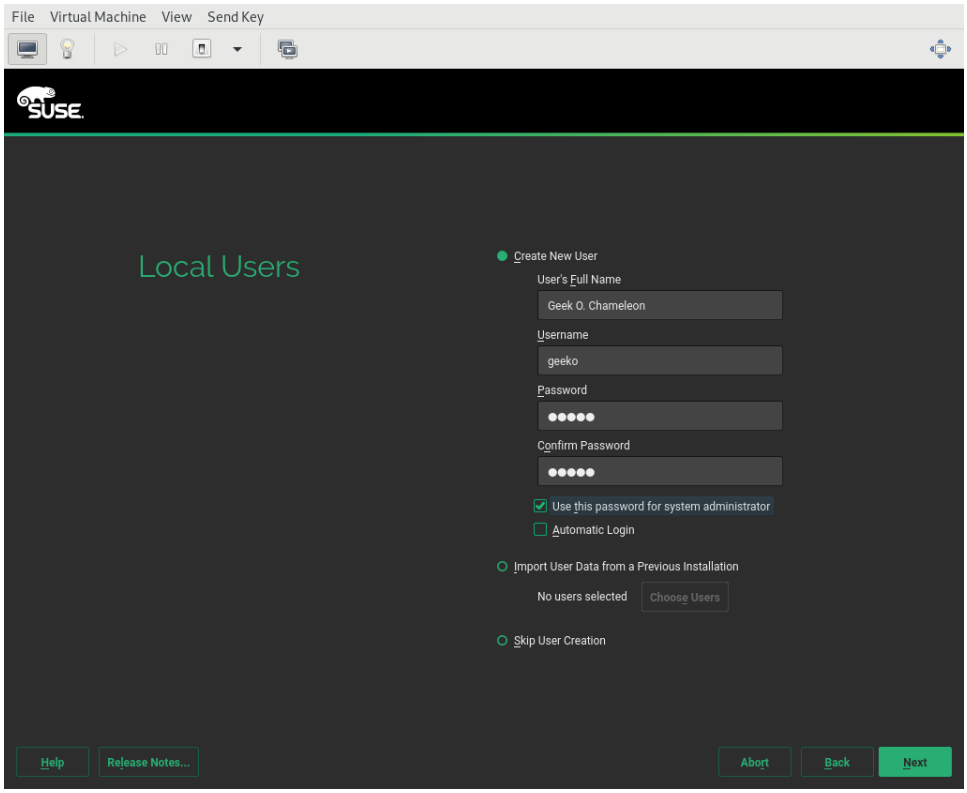
25. Click Next in Suggested Partitioning.



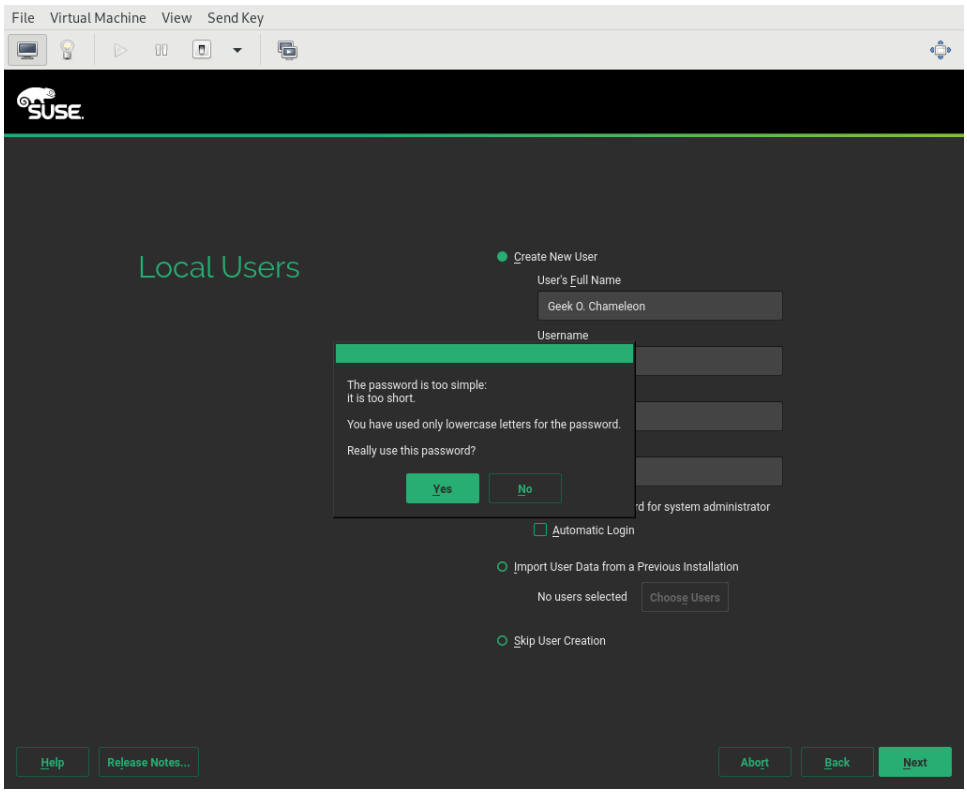
26. Click Next in Clock and Time Zone.



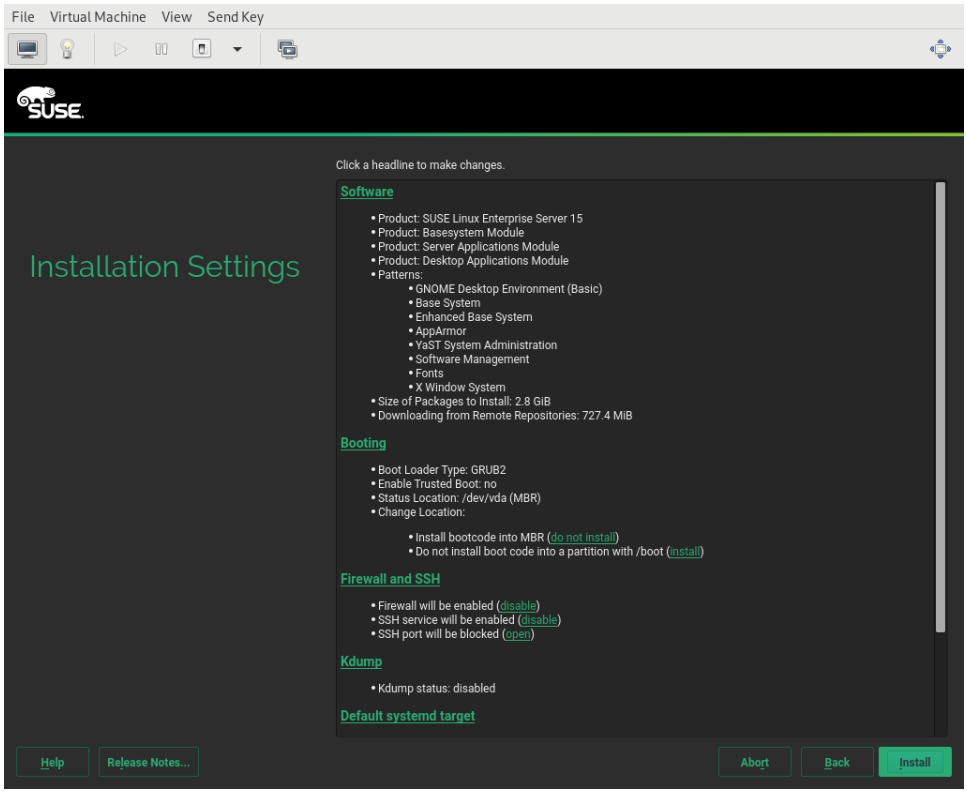
27. Enter “Geek O. Chameleon” for User’s Full Name, “geeko” for Username, “linux” for Password and Confirm Password, check “Use this password for system administrator”, and click Next in Local Users.



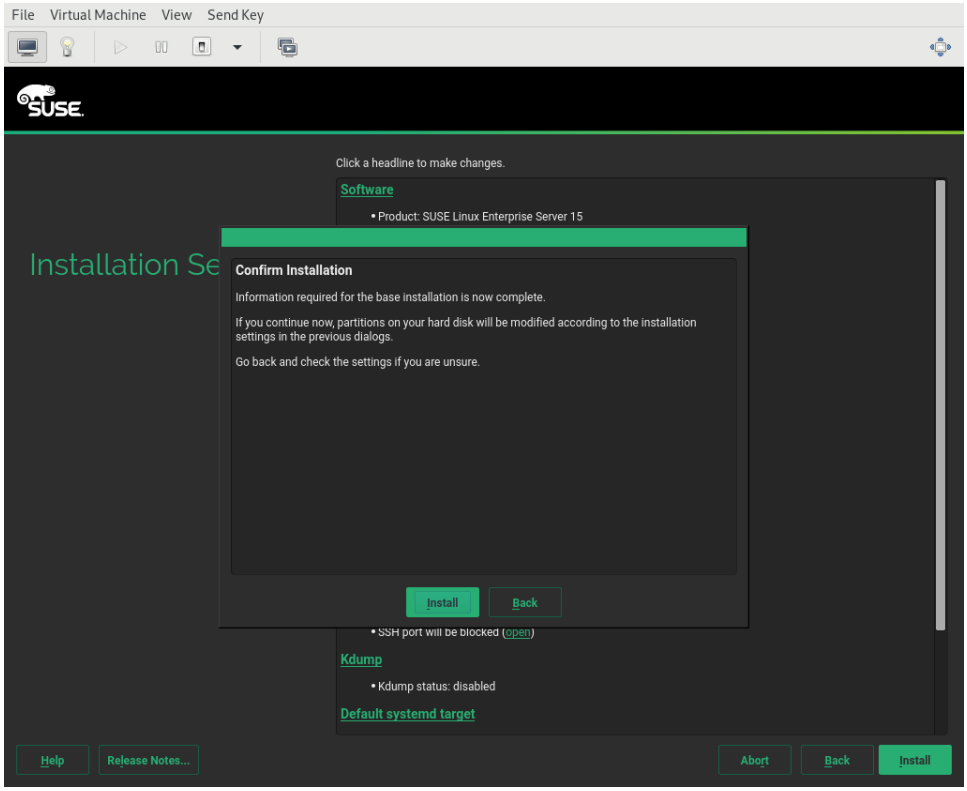
28. Click Yes to accept the too-simple password.



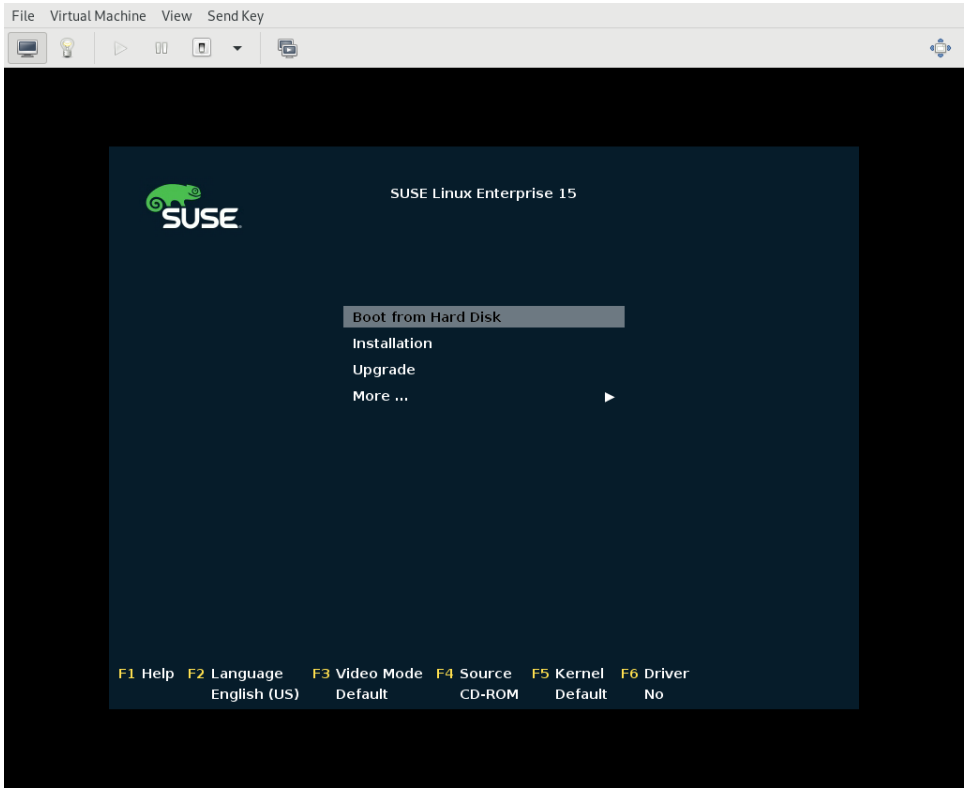
29. Click Install in Installation Settings. (Don't change anything in the Firewall and SSH section as it will affect the next lab)



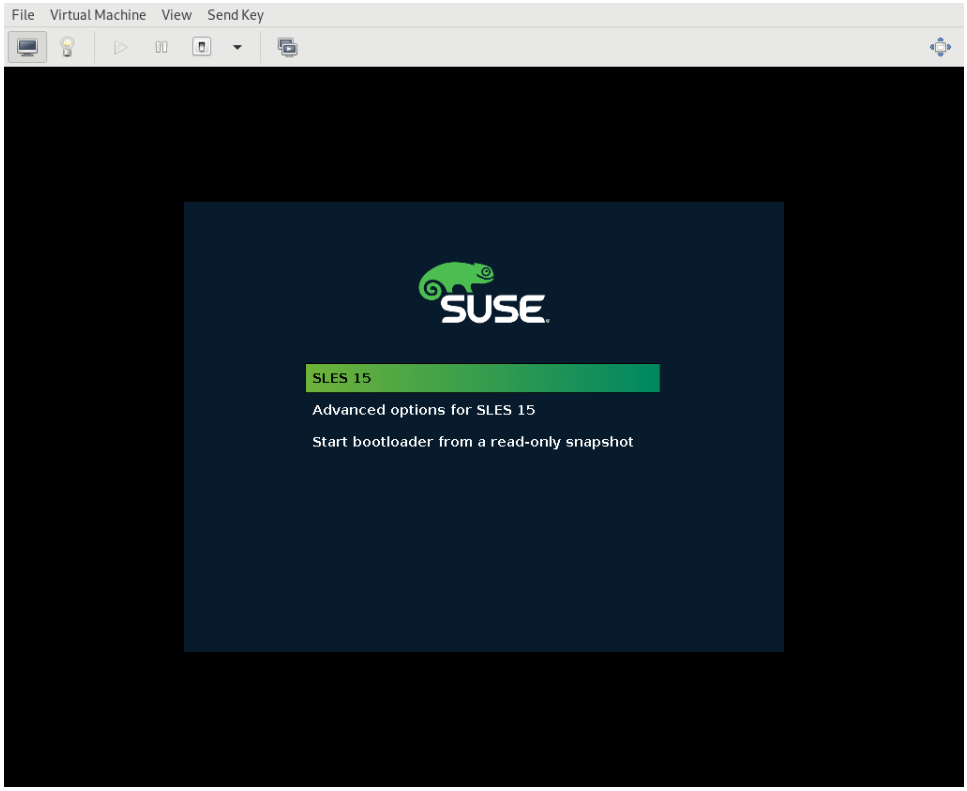
30. Click Install to confirm installation, and wait.



31. When the server reboots up from the DVD, select Boot from Hard Disk.



32. Select SLES 15.



1.2 Migrate openSUSE Leap to SLES 15

https://www.suse.com/documentation/sles-15/book_sle_upgrade/data/sec_upgrade-online_opensuse_to_sle.html

1. In virt-manager, start the opensuse15.0 VM.
2. Open a terminal on your host system and connect to the openSUSE Leap VM:

```
$ ssh root@<leap>
```

3. Check the distro and version (optional)

```
# cat /etc/os-release
```

4. Install SUSEConnect:

```
# zypper in SUSEConnect
```

5. Remove packages that produce file conflict during the migration:

```
# rpm -e --nodeps yast2-branding-openSUSE
```

6. Copy your RMT's root CA:

```
# curl --insecure -o /etc/pki/trust/anchors/rmt.crt
https://<rmt>/rmt.crt
```

7. Install the new certificate:

```
# update-ca-certificates
```

7. Register with RMT server:

```
# SUSEConnect –u https://<rmt>/ -p SLES/15.0/x86_64
```

8. List and disable all openSUSE repositories on your system:

```
# zypper lr
# zypper rr REPO_IDS
```

Note: REPO_IDS can be stated as a "sequence expression"

```
Example: {7..16}
```

9. Add modules needed for installation:

```
# SUSEConnect --list-extensions
# SUSEConnect –p sle-module-basesystem/15.0/x86_64
```

Optional Step: will update gnome and other desktop based applications.

```
# SUSEConnect –p sle-module-desktop-applications/15.0/x86_64
```

10. Migrate installed packages to the SUSE Linux Enterprise Server repositories.

```
# zypper dup --force-resolution
```

Note: If you get a question about proceeding with conflicting files say yes. This is generally just relates to branding.

11. Remove orphaned packages:

```
# zypper rm $(zypper --no-refresh packages --orphaned | gawk
'{print $5}' | tail -n +5)
```

1.3 Build an Autoyast file from SLE 15 VM

Using SLES tools to create a clone of an existing system and then editing the installation script. Use the virt-manager console window for the following steps as ssh is still firewalled off.

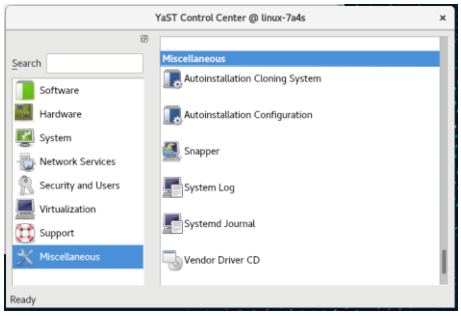
1. Install the YaST AutoYast module:

```
# zypper in autoyast2
```

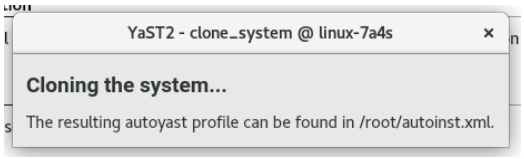
```
linux-7a4s:/home/cliddle # zypper in autoyast2
Refreshing service 'Basesystem_Module_15_x86_64'.
Refreshing service 'Desktop_Applications_Module_15_x86_64'.
Refreshing service 'Development_Tools_Module_15_x86_64'.
Refreshing service 'Legacy_Module_15_x86_64'.
Refreshing service 'SUSE_Linux_Enterprise_Server_15_x86_64'.
Refreshing service 'SUSE_Package_Hub_15_x86_64'.
Refreshing service 'Server_Applications_Module_15_x86_64'.
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following 2 NEW packages are going to be installed:
  autoyast2 yast2-schema

2 new packages to install.
Overall download size: 479.9 KiB. Already cached: 0 B. After the operation,
additional 2.0 MiB will be used.
Continue? [y/n/...? shows all options] (y): y
Retrieving package yast2-schema-4.0.2-1.81.x86_64                (1/2),  76.1 KiB (971.8 KiB unpacked)
Retrieving: yast2-schema-4.0.2-1.81.x86_64.rpm ..................................................[done]
Retrieving package autoyast2-4.0.67-3.14.5.noarch                (2/2), 403.8 KiB (  1.1 MiB unpacked)
Retrieving: autoyast2-4.0.67-3.14.5.noarch.rpm ..................................................[done]
Checking for file conflicts: ....................................................................[done]
(1/2) Installing: yast2-schema-4.0.2-1.81.x86_64 ................................................[done]
(2/2) Installing: autoyast2-4.0.67-3.14.5.noarch ................................................[done]
Additional rpm output:
Updating /etc/sysconfig/autoinstall ...


linux-7a4s:/home/cliddle #
```

2. Click activities in the top left or hit the “windows key” to access the program search menu and type “yast”:

3. Select the “Miscellaneous” group:



4. Select “Autoinstallation Cloning System”:

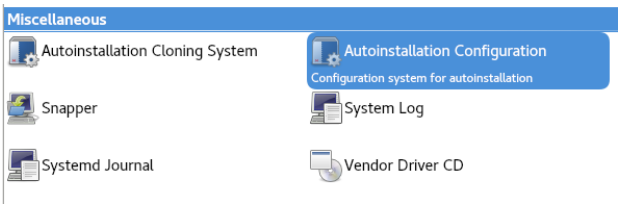


5. In a terminal window review the resulting Autoinstall.xml file in /root:

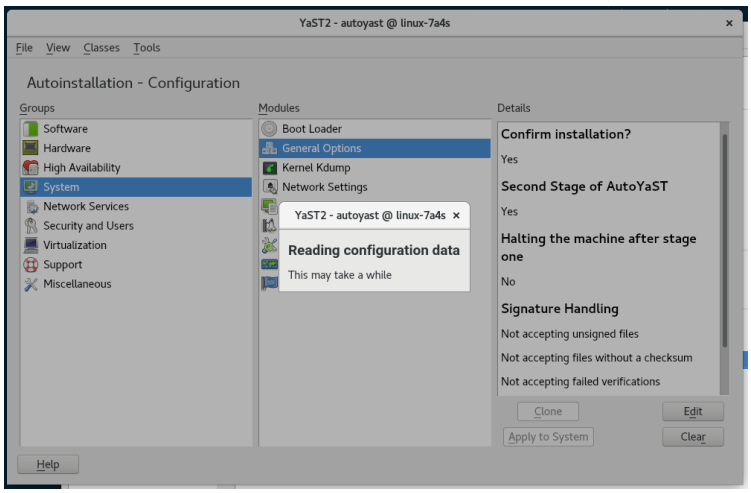
```
# cat /root/autoinst.xml
```

Notice how difficult it is to understand and edit the auto install options in the resulting XML file.

6. Use the YaST editor for Autoyast files. Select “Autoinstallation Configuration” option in YaST:



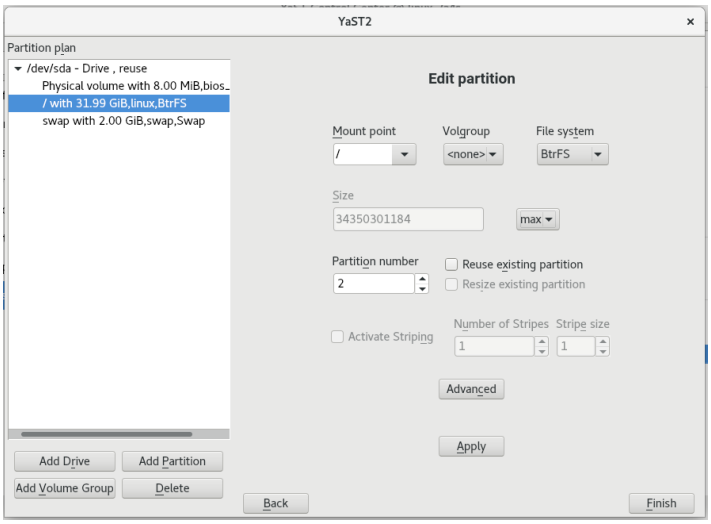
7. Open the autoyast file from the cloning operation earlier. Select File – Open - "/root/autoinstall.xml":



Once the file has been read into the system you can edit the properties of the cloned Autoyast xml file.

8. Modify Disk Partitions. Select Hardware –> Partisi –> Edit.

Notice the partitioning settings are directly from the existing machine. Change it to make it more generic and select Max size instead of a specific amount:



Click Apply and Back.

9. Set Time zone

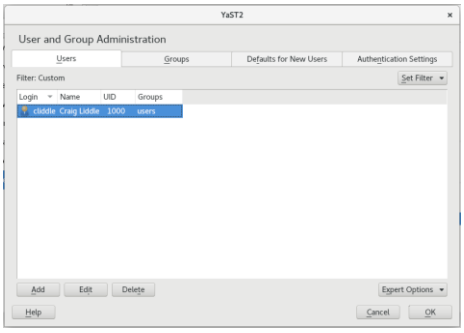
Select System – Date and Time – Edit (Button on the right)

Notice this looks just like it does during the install process. Select a different time zone than you had done in the install:

Click Next

10. Add Users and Groups

    Select Security and Users- User and Group Management – Edit

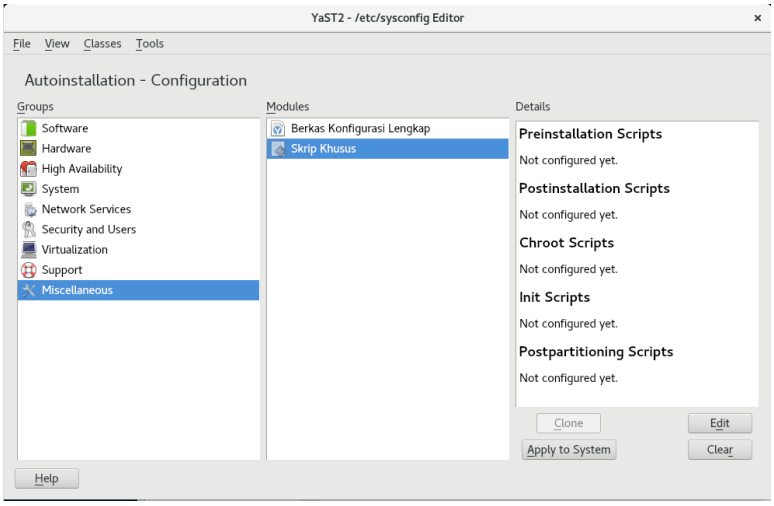


    Now you can add the appropriate users and groups for any new installs.

11. Set Pre and Post installation Scripts

    Select Miscellaneous –> Skrip Khusus

    From this interface you can insert scripts into the autoyast provisioning process



12. Save modified autoyast file

    Select File –> Save As –> File name

    You now have an autoyast file to apply to an existing machine through this interface or a file you can use to deploy as many machines as you like via Autoyast install.

For more information:

https://www.suse.com/documentation/sles-15/book_autoyast/data/overviewandconcept.html

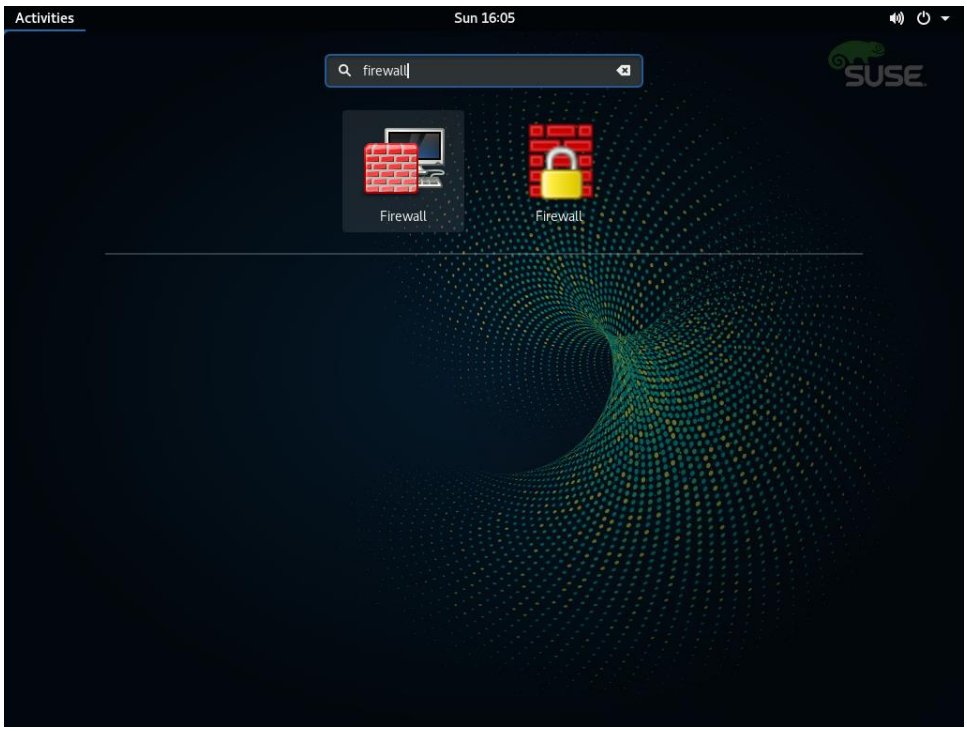
Lab 2 – Firewalld

2.1 Open SSH port for trusted zone

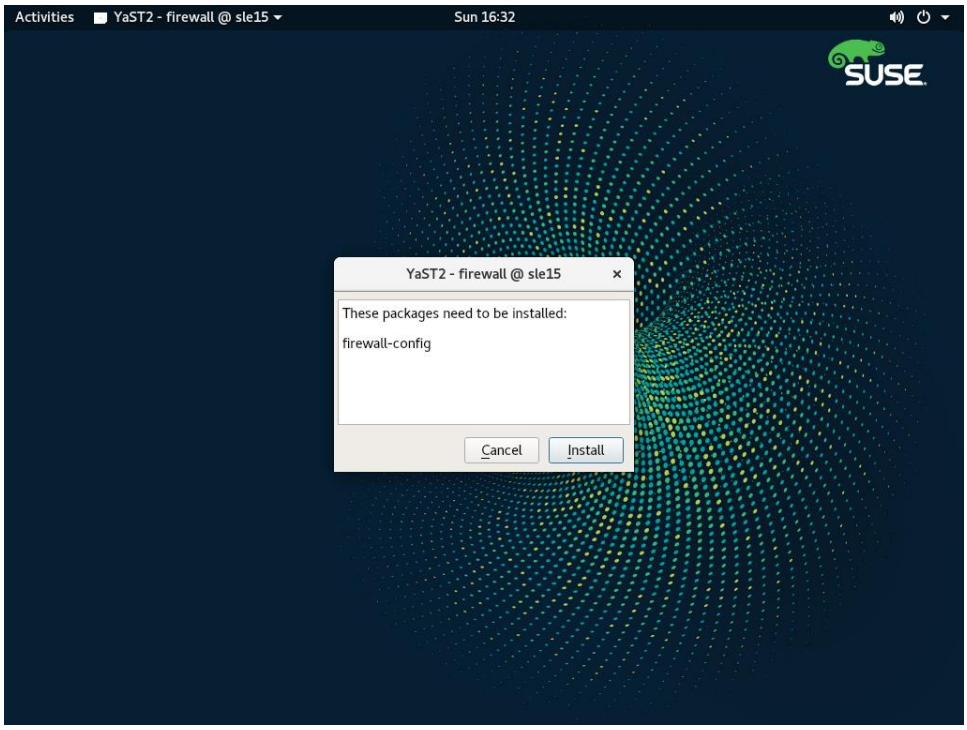
1. Attempt to ssh to the SLE 15 VM from laptop (this should fail):

```
$ ssh root@<sle15>
```

2. Connect to the VM in virt-manager and log in as geeko.
3. Select “Activities” and type “firewall”:

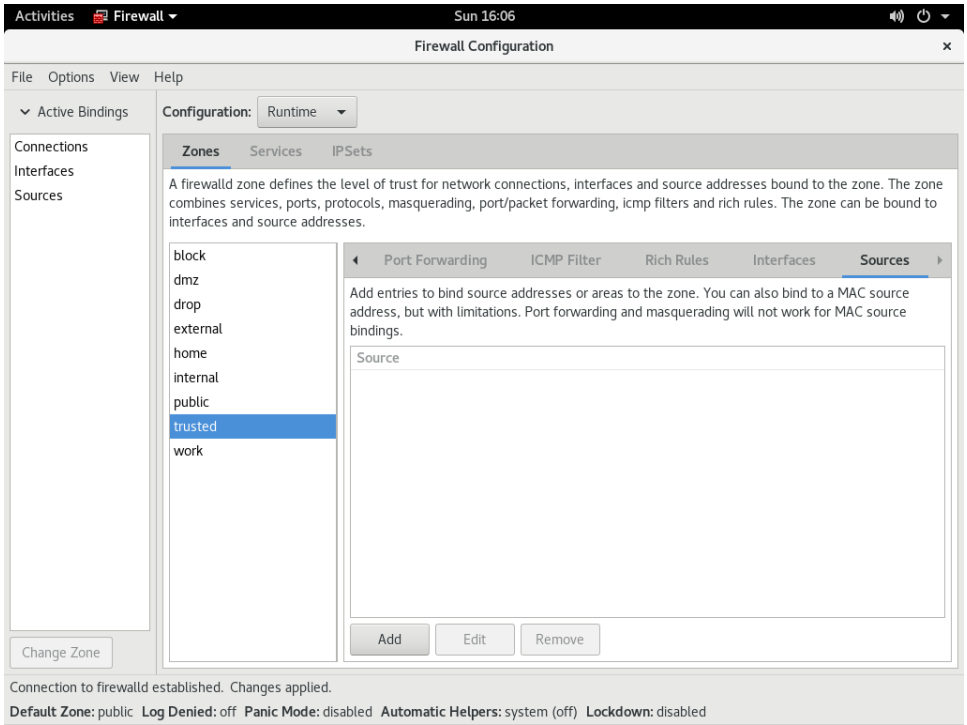


4. Click the Firewall icon and enter the root password.
5. Click Install to install the firewall-config package.

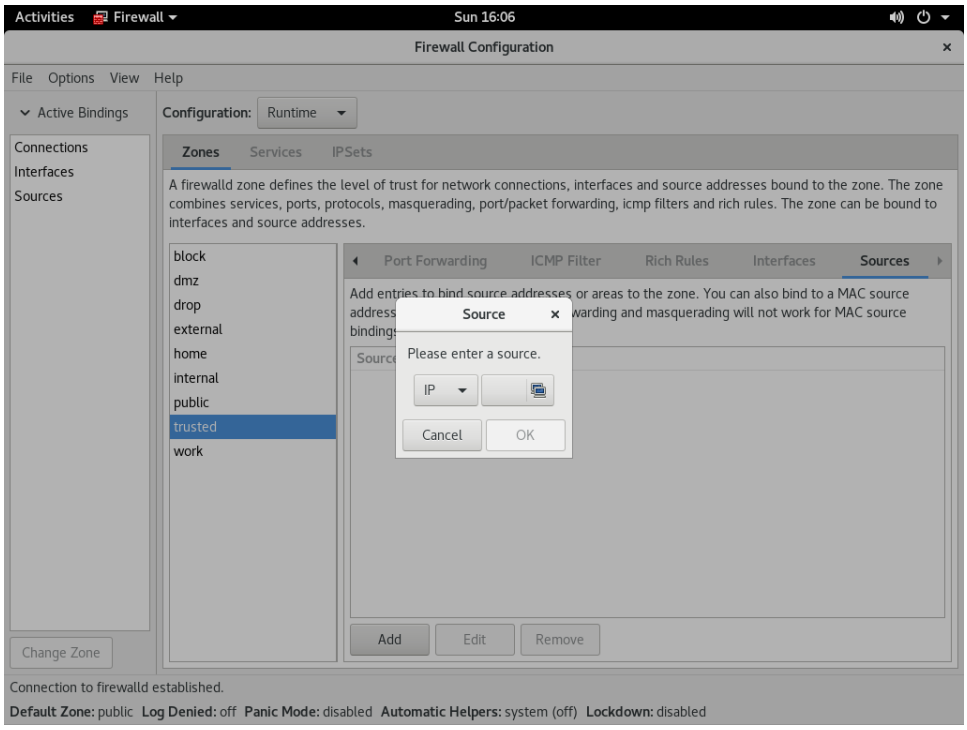


Note: The changes below only affect “runtime” and don’t persist through a reboot. You can click the drop down to change that, but you still need to change the runtime settings for immediate change OR restart the firewalld service under the options menu.

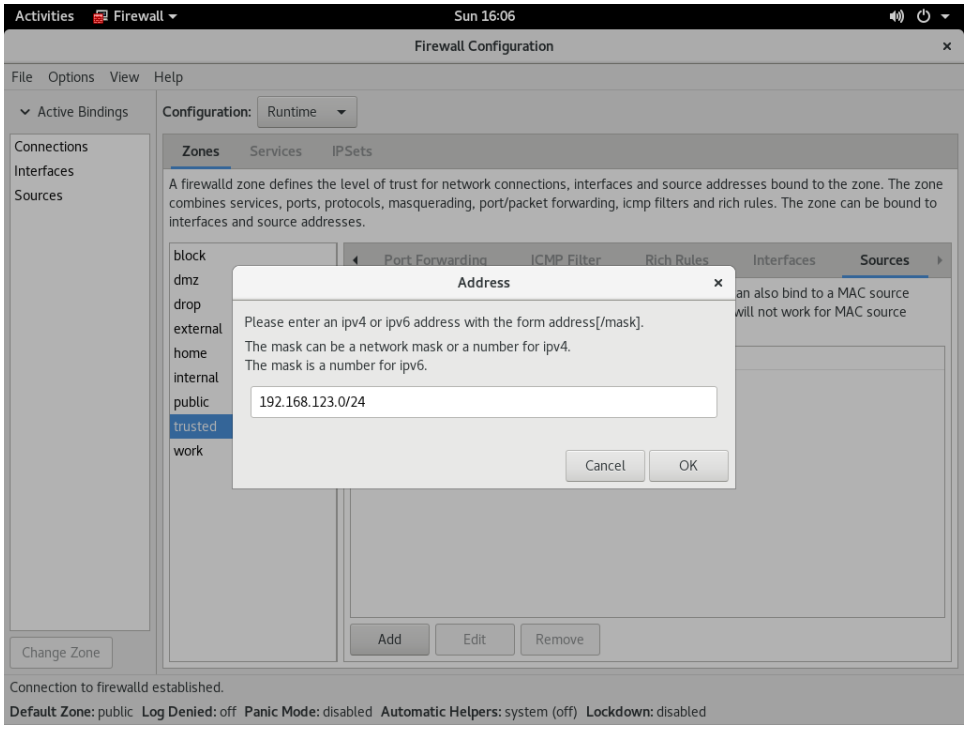
6. Click on Zones -> trusted -> Sources



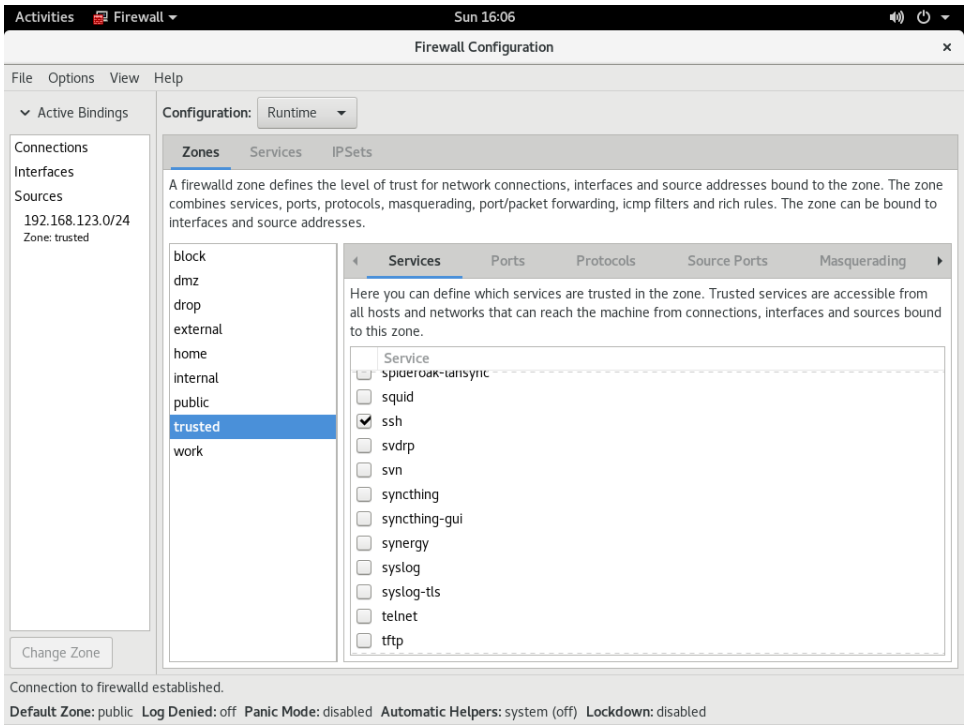
7. Click Add



8. Click on address field
9. Enter “192.168.123.0/24” and click OK



10. Click OK
11. Select Zones -> trusted -> Services
12. Check “ssh”



13. Attempt to ssh to the SLE 15 VM from laptop

```
$ ssh root@<sle15>
```

14. Make the change permanent by selecting Configuration -> Permanent and repeating steps 6-12.

2.2 Use the command line

1. View all zones:

```
$ firewall-cmd --get-zones
```

2. View the current configuration for the default zone

```
$ firewall-cmd --list-all
```

3. View the current configuration for the trusted zone

```
$ firewall-cmd --zone=trusted --list-all
```

4. View the current configuration for the public zone

```
$ firewall-cmd --zone=public --list-all
```

5. Open HTTP port to public internet:

```
$ firewall-cmd --zone=public --add-service http --add-service
https
```

6. Make the changes permanent:

```
$ firewall-cmd --permanent --zone=public --add-service http --
add-service https
$ firewall-cmd --reload
```

Lab 3 – Salt, nginx, and chrony

3.0 Setup a hosts file (optional)

This shouldn't be strictly necessary, but it is good practice to base salt on hostnames rather than IP addresses. Since we don't have a DNS setup here it would need to be done through /etc/hosts. At a minimum we need to include our rmt server and the sle15 VM we just installed. You may also pull in other VMs from previous labs if you want to have more to test with. We will leave that up to you.

1. Edit the hosts file

   ```
   $ sudo vi /etc/hosts
   ```

2. Include lines similar to the following, this can vary depending on your IP address and hostname

   ```
   192.168.123.174 rmt.HO1424 rmt

   192.168.123.226 sle15.HO1424 sle15
   ```

3. Copy the file to all machines you plan to test with, example below:

   ```
   $ scp /etc/hosts root@<sle15>:/etc/hosts
   ```

3.1 Register the new system to the Salt master

1. Connect to all VMs you plan to include if not already

   ```
   $ ssh root@<rmt>
   $ ssh root@<sle15>
   ...
   ```

2. Check what is installed already

   $ zypper se salt

   You will see that the salt master is already installed on the rmt server, but nothing on the others.

3. Install salt minions

   ```
   $ zypper in salt-minion
   ```

   Complete for each system in the lab

4. Edit /etc/salt/minion to point to the Salt master

   Change the following line:
   `#master: salt`
   To:
   `master: <rmt>`

   Complete for each system in the lab

5. Restart the Salt minion:

```
$ systemctl restart salt-minion
```

Complete for each system in the lab

6. Enable the minions so they persist through reboot

   $ systemctl enable salt-minion

   Complete for each system in the lab

7. Verify the minions are requesting access

   ```
   $ Salt-key
   ```

8. Accept the Salt minion on the **rmt server**

   ```
   $ Salt-key -A
   ```

9. Ping the Salt minion

   ```
   $ salt '*' test.ping
   ```

   Note: It may take a minute or so before this command will work as there is some initialition taking place after the keys are accepted.

10. Create salt fileserver

    ```
    $ vi /etc/salt/master
    ```

    Search for the line in the file that includes '/srv/salt'. In vi you can type a '/' to open the search pane and type the folder in there and hit enter. By default this should take you to line 661. Uncomment the following section:

    ```
    file_roots:
      base:
        - /srv/salt/
    ```

    Ensure there are no leading spaces in the lines. The first line should have no spaces, the next has 2, then 4.

11. Install git

    We need to get files from github, which means we need to install the git client. This will require us to register the RMT server to two more repositories, Desktop Applications and Development Tools.

    ```
    # SUSEConnect -p sle-module-desktop-applications/15/x86_64
    # SUSEConnect -p sle-module-development-tools/15/x86_64
    # zypper in git
    ```

12. Download salt files from github and copy them into place

    ```
    # git clone https://github.com/nbornstein/HO1424.git
    # cp -r HO1424/salt/* /srv/salt/
    ```

3.2 Configure nginx web server using Salt

1. Review the nginx.sls state file

```
$ cat /srv/salt/webserver/nginx.sls
```

This installs the package, turns on the service and enables it so it turns on during boot. The watch command will check for differences in the conf file when the state is run and recopy it if it isn't the same. This can be done as a daily task. We then also insert a index file for the web server and define the managed conf file.

```
pkg_nginx:
  pkg.installed:
    - name: nginx

srv_nginx:
  service.running:
    - name: nginx
    - enable: True
    - watch:
      - pkg: nginx
      - file: /etc/nginx/nginx.conf

push_index:
  file.managed:
    - name: /srv/www/htdocs/index.html
    - source: salt://webserver/index.html

cfg_nginx:
  file.managed:
    - name: /etc/nginx/nginx.conf
    - source: salt://webserver/nginx.conf
```

2. Apply salt state to start nginx and load index.html (use the rmt terminal session)

```
$ salt 'sle15*' state.apply webserver.nginx
```

Note: The wildcard is just so we don't need to type the FQDN, the definition of "webserver.nginx" is based on the files we moved into the /srv/salt/ directory which we previous set at the root of the salt fileserver.

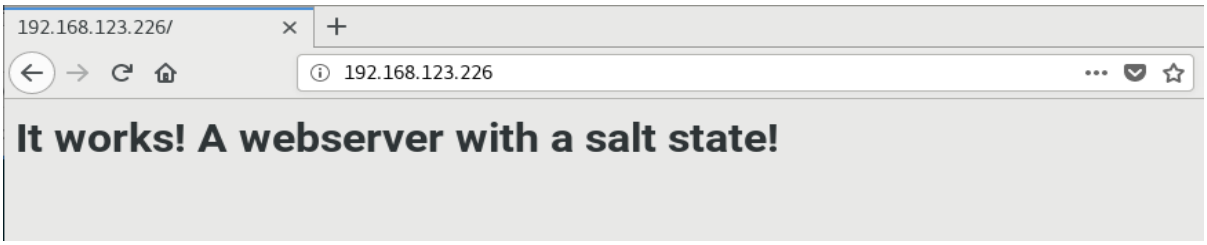
3. Verify the web server is running

```
$ systemctl status nginx
```

Note: The status may not update immediately and take a minute.

4. Open in Firefox

On the host system open a browser and go to http://<sle15>



3.3 Configure NTP server using chrony

1. Review the ntp state file

```
$ cat /srv/salt/chrony/ntp.sls
```

```
This will install chrony (if not already), turn it on, enable
it, and then manage the config file.
```

```
pkg_ntp:
  pkg.installed:
    - name: chrony

srv_ntp:
  service.running:
    - name: chronyd
    - enable: True
    - watch:
      - pkg: chrony
      - file: /etc/chrony.conf

cfg_ntp:
  file.managed:
    - name: /etc/chrony.conf
    - source: salt://chrony/chrony.conf
    - user: root
    - group: chrony
    - mode: 640
```

2. Review the top part of the chrony.conf file

```
$ vi /srv/salt/chrony/chrony.conf
```

If the IP address in the “pool” setting at the top of the file doesn’t match the IP of your rmt server, then edit the file to fix it.

3. Apply salt state to add NTP server setting

```
$ salt 'sle15*' state.apply chrony.ntp
```

4. Verify the setting was changed

```
$ vi /etc/chrony.conf
```

```
$ chronyc sources
```

Note: chronyc is a command line tool for interacting with chrony settings and is added here as just an optional step you can take as an alternative to see that the NTP setting is updated. If you get a message that it cannot talk to the daemon then the chrony service isn't started.

3.4 Understanding highstate

1. Review the top.sls file

```
$ cat /srv/salt/top.sls
```

This file is very basic and just outlines what states to apply to each system by hostname. This can get much more complex however and be organized by other settings aside from hostname. You can also define groups of rules that are outside 'base' such as 'dev' or 'prod'

```
base:
  '*':
    - chrony.ntp
  'sle15*':
    - webserver.nginx
```

2. Revert all states we have done so far

```
$ salt 'sle15*' state.apply webserver.nonginx
```

```
$ salt 'sle15*' state.apply chrony.nontp
```

3. Apply the highstate to all systems

```
$ salt '*' state.highstate
```

Lab 4 – Repository Management Tool

4.1 Install RMT Server

1. Connect to the SLE 15 VM via SSH:

```
$ ssh root@<sle15>
```

2. Install the RMT server:

```
# zypper in rmt-server
```

3. Ignore the update notifications from MariaDB by hitting Return at the prompt:

```
Update notifications were received from the following
packages:
mariadb-10.2.22-3.14.1.x86_64 (/var/adm/update-
messages/mariadb-10.2.22-3.14.1-something)
View the notifications now? [y/n] (n):
```

4.2 View RMT Status

1. Connect to the RMT VM via SSH:

```
$ ssh root@<rmt>
```

2. View the RMT timers

```
# systemctl status rmt-server-sync.timer rmt-server-
mirror.timer
```

3. View the RMT services:

```
# systemctl status rmt-server rmt-server-sync rmt-server-
mirror
```

4.3 View products and repositories

1. View mirrored products:

```
# rmt-cli products list
```

2. View all available products:

```
# rmt-cli products list --all
```

3. View mirrored repositories:

```
# rmt-cli repos list
```

4. View all available repositories:

```
# rmt-cli repos list --all
```

Lab 5 – Software Management

5.1 Use zypper to search for a package

1. Connect to the SLE 15 VM via SSH:

```
$ ssh root@<sle15>
```

2. Search for gcc-objc in currently assigned repositories:

```
# zypper se gcc-objc
```

3. Search for gcc-objc across all available repositories:

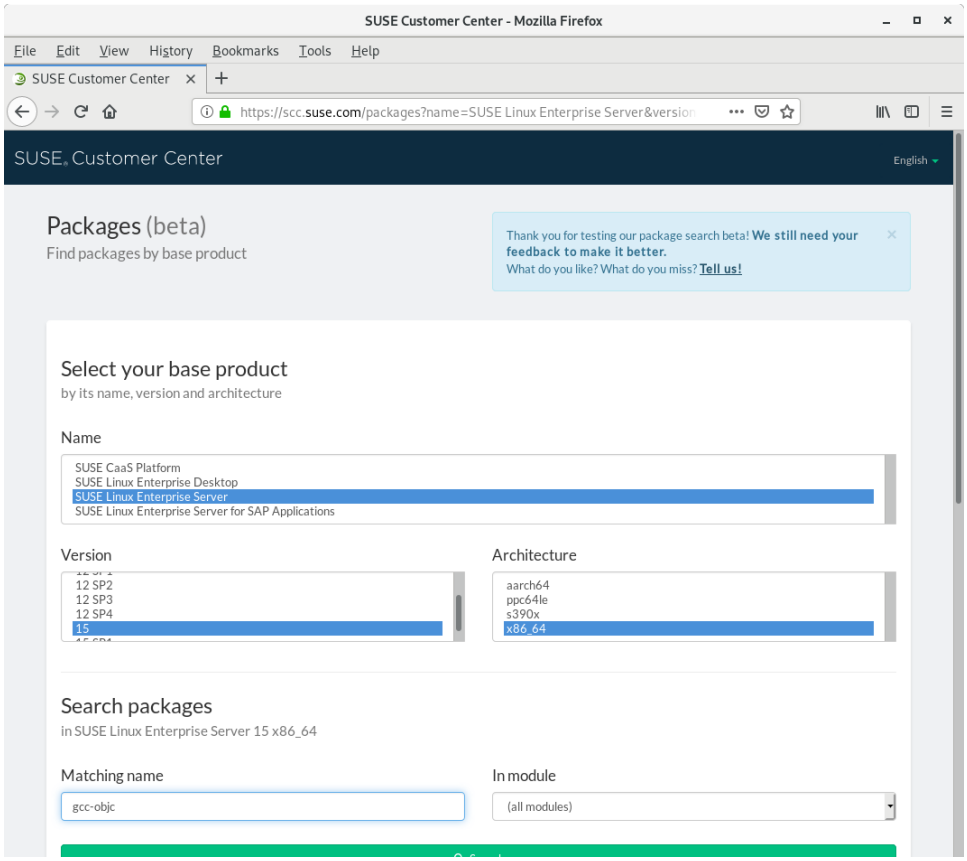
```
# zypper search-packages gcc-objc
```

5.2 Use SUSE Customer Center to search for a package

1. Browse to the following URL in Firefox:

```
https://scc.suse.com/packages
```

2. Fill out the fields as shown and click “Search”:



Lab 6 – Extra Credit

6.1 Migrate SLES 12 SP4 to SLES 15

https://www.suse.com/documentation/sles-15/singlehtml/book_sle_upgrade/book_sle_upgrade.html#cha.upgrade-offline

1. In virt-manager, boot the sles12sp4 VM from the DVD.
2. Select Boot from Hard Disk and hit Return
3. Select “SLES 12-SP4"
4. SSH into the VM

```
$ ssh root@<sles12sp4>
```

5. Remove any existing registrations

```
# SUSEConnect --cleanup
```

6. Reboot the VM from DVD
7. Select Upgrade, enter “`media_upgrade=1`” in Boot Options, and hit Return
8. Click Next in “Language and Keyboard Selection”
9. Click Next in “Select for Update”
10. Check “I Agree to the License Terms” and click Next in the License Agreement
11. Click Next in Previously Used Repositories
12. Click OK in Media Base Upgrade
13. Select DVD... in Add On Product and click Next
14. In VM console, select View -> Details
15. Select SATA CDROM1
16. In Source path select “`SLE-15-Packages-x86_64-GM-DVD1.iso`”
17. Click Apply
18. In VM console, select View -> Console
19. Click Continue
20. Select Basesystem-Module 15-0 and SLES15 15-0 in Extension and Module Selection
21. Click Next
22. Click Next in Add-On Product Installation
23. Click Update in Installation Settings
24. Click Start Update in Confirm Update
25. Change the DVD as requested following steps 14-17