I would like to advise following tutorial is for beginners who has no idea setting up an KVM I it made simple just to give you an idea on setting from scratch, from there you guys can grow…sorry if my explanations isn't up to the par but I hope it helps you understand.

KVM (Kernel-based Virtual Machine) is a full virtualization solution for Linux on x86 hardware containing virtualization extensions Intel VT or AMD-V. However, this tutorial is tested on both CentOS and RHEL 64 bit with Intel i7 CPU (with Intel VT) and 64 bit kernels with SELinux running in Disabled mode.

Before heading further deep I would like to share my current network setup with you guys

```
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 10
00
    link/ether 00:0c:29:eb:3e:b5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.100/24 brd 192.168.0.255 scope global noprefixroute ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feeb:3eb5/64 scope link
       valid_lft forever preferred_lft forever
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens33
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPADDR=192.168.0.100
PREFIX=24
GATEWAY=192.168.0.1
DNS1=8.8.8.8
DNS2=8.8.4.4
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=no
NAME=ens33
UUID=bc7efb23-7eb7-42af-8dfd-77bb34372b40
DEVICE=ens33
ONBOOT=yes
[root@localhost ~]# _
```

Lets Start!!!

Verify whether KVM is supported (KVM is a virtual machine software based on x86 virtualization extension technology) (Intel VT or AMD-V)

lsmod | grep kvm

```
[root@localhost ~]# lsmod | grep kvm
kvm_intel             188740  0
kvm                   637289  1 kvm_intel
irqbypass              13503  1 kvm
[root@localhost ~]#
```

vi /etc/sysconfig/selinux (**SELINUX=enforcing** in /etc/sysconfig/selinux to **SELINUX=disabled**) and save changes

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled_
# SELINUXTYPE= can take one of three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Installing KVM environment

Adopt yum Install kvm Foundation Package and Management Tools

yum install -y qemu-kvm libvirt libvirt-python libguestfs-tools virt-install python-virtinst virt-manager bridge-utils (Press Enter once your done)

```
[root@localhost ~]# yum install -y qemu-kvm libvirt libvirt-python libguest-tools virt-install pytho
n-virtinst virt-manager bridge-utils_
```

Packages will start downloading and installing

```
 spice-gtk3                              x86_64    0.35-5.el7_9.1              updates     87 k
 spice-server                            x86_64    0.14.0-9.el7_9.1            updates    405 k
 tcp_wrappers                            x86_64    7.6-77.el7                  base        78 k
 trousers                                x86_64    0.3.14-2.el7                base       289 k
 unbound-libs                            x86_64    1.6.6-5.el7_8               base       406 k
 usbredir                                x86_64    0.7.1-3.el7                 base        47 k
 virt-manager-common                     noarch    1.5.0-7.el7                 base       1.2 M
 vte-profile                             x86_64    0.52.4-1.el7                base       7.1 k
 vte291                                  x86_64    0.52.4-1.el7                base       250 k
 xkeyboard-config                        noarch    2.24-1.el7                  base       834 k
 xml-common                              noarch    0.6.3-39.el7                base        26 k
 xorg-x11-server-utils                   x86_64    7.7-20.el7                  base       178 k
 xorg-x11-xauth                          x86_64    1:1.0.9-1.el7               base        30 k
 xorg-x11-xinit                          x86_64    1.3.4-2.el7                 base        58 k
 yajl                                    x86_64    2.0.4-4.el7                 base        39 k

Transaction Summary
================================================================================================
Install  6 Packages (+238 Dependent packages)

Total download size: 90 M
Installed size: 267 M
Downloading packages:
(1/244): adwaita-icon-theme-3.28.0-1.el7.noarch.rpm                        |  11 MB  00:00:02
(2/244): at-spi2-atk-2.26.2-1.el7.x86_64.rpm                               |  81 kB  00:00:00
(3/244): at-spi2-core-2.28.0-1.el7.x86_64.rpm                              | 158 kB  00:00:00
(4/244): atk-2.28.1-2.el7.x86_64.rpm                                       | 263 kB  00:00:00
(5/244): augeas-libs-1.4.0-10.el7.x86_64.rpm                               | 357 kB  00:00:00
(6/244): autogen-libopts-5.18-5.el7.x86_64.rpm                             |  66 kB  00:00:00
(7/244): avahi-libs-0.6.31-20.el7.x86_64.rpm                               |  62 kB  00:00:00
(8/244): boost-iostreams-1.53.0-28.el7.x86_64.rpm                          |  61 kB  00:00:00
(9/244): boost-random-1.53.0-28.el7.x86_64.rpm                             |  39 kB  00:00:00
(10/244): boost-system-1.53.0-28.el7.x86_64.rpm                            |  40 kB  00:00:00
(11/244): boost-thread-1.53.0-28.el7.x86_64.rpm                            |  58 kB  00:00:00
(12/244): bridge-utils-1.5-9.el7.x86_64.rpm                                |  32 kB  00:00:00
(13/244): bzip2-1.0.6-13.el7.x86_64.rpm                                    |  52 kB  00:00:00
(15/244): cairo-1.15.12-4.el7.x86_6 14% [===-                      ] 2.6 MB/s |  13 MB  00:00:29 ETA
```

Once done restart the host to load the kvm module by issuing **reboot** command

```
[root@localhost ~]# reboot_
```

Once the machine is up issue the following command to verify libvirtd is running

systemctl status libvirtd (verify if its on running state if it isn't issue following command **systemctl start libvirtd**)

```
[root@localhost ~]# systemctl status libvirtd
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2021-04-11 17:48:42 +08; 1min 7s ago
     Docs: man:libvirtd(8)
           https://libvirt.org
 Main PID: 1142 (libvirtd)
    Tasks: 19 (limit: 32768)
   CGroup: /system.slice/libvirtd.service
           ├─1142 /usr/sbin/libvirtd
           ├─1657 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile...
           └─1659 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile...

Apr 11 17:48:43 localhost.localdomain dnsmasq[1657]: started, version 2.76 cachesize 150
Apr 11 17:48:43 localhost.localdomain dnsmasq[1657]: compile time options: IPv6 GNU-getopt DBus...fy
Apr 11 17:48:43 localhost.localdomain dnsmasq-dhcp[1657]: DHCP, IP range 192.168.122.2 -- 192.1...1h
Apr 11 17:48:43 localhost.localdomain dnsmasq-dhcp[1657]: DHCP, sockets bound exclusively to in...r0
Apr 11 17:48:43 localhost.localdomain dnsmasq[1657]: reading /etc/resolv.conf
Apr 11 17:48:43 localhost.localdomain dnsmasq[1657]: using nameserver 8.8.8.8#53
Apr 11 17:48:43 localhost.localdomain dnsmasq[1657]: using nameserver 8.8.4.4#53
Apr 11 17:48:43 localhost.localdomain dnsmasq[1657]: read /etc/hosts - 2 addresses
Apr 11 17:48:43 localhost.localdomain dnsmasq[1657]: read /var/lib/libvirt/dnsmasq/default.addn...es
Apr 11 17:48:43 localhost.localdomain dnsmasq-dhcp[1657]: read /var/lib/libvirt/dnsmasq/default...le
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost ~]# systemctl is-enabled libvirtd
enabled
[root@localhost ~]# _
```

Issue the following command to enable the service always **systemctl is-enabled libvirtd**

```
[root@localhost ~]# systemctl is-enabled libvirtd
enabled
[root@localhost ~]# _
```

Now we are set go!!!

Before that I would like to explain to you guys on the next command before issuing it since im working with headless server I won't be using graphical install…sorry ya guys

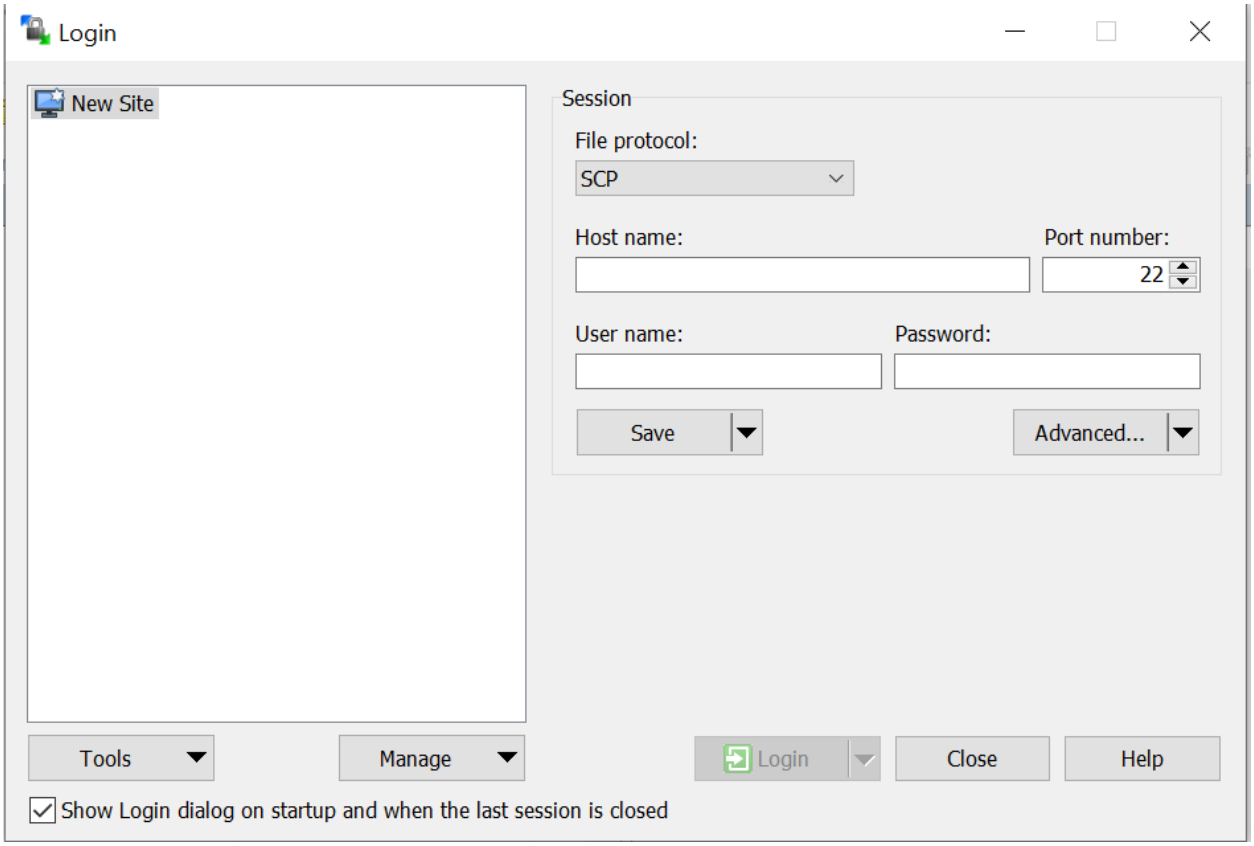
virt-install --virt-type=kvm --name=(**name of the vm**) --vcpus=(**number of cpu**) --memory=(**ram**)

--location=(**location of the iso**) --disk path=(**location where the vm is stored**). qcow2, size=(**number gb allocated for vm**), format=qcow2 --network bridge=br0 --graphics none --extra-args='console=ttyS0' --force

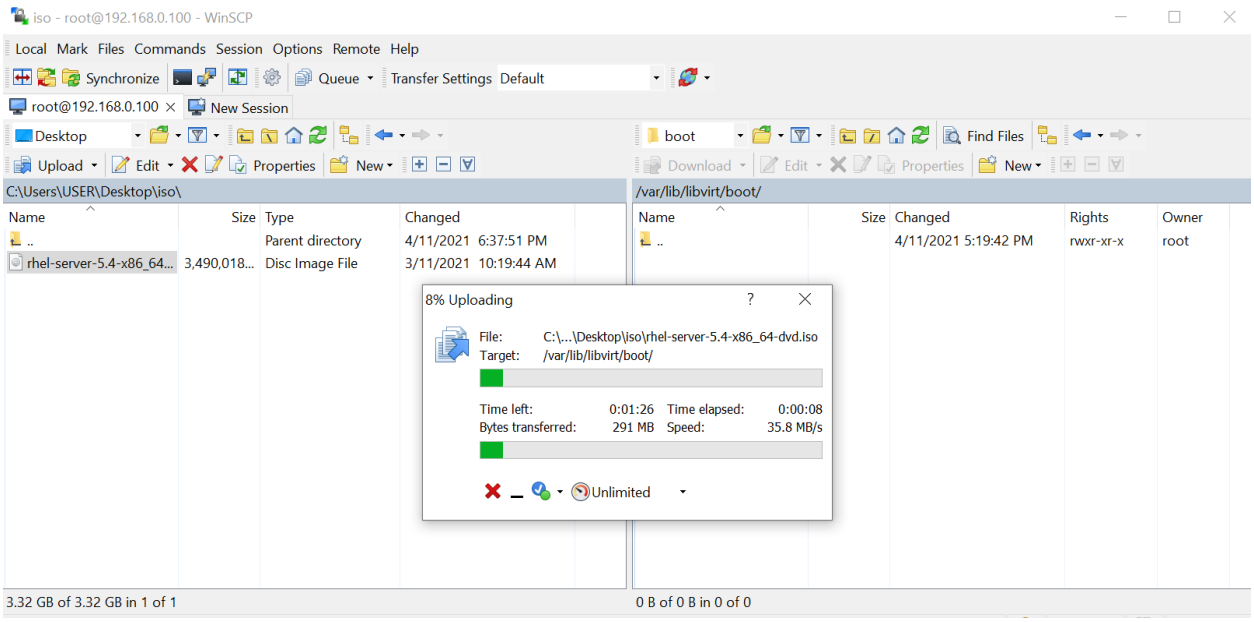
But before that I missed on uploading the iso to do this you'll need to download winscp

Use file protocol **SCP**

In my case hostname would be my server's ip



Copy iso to the following path **/var/lib/libvirt/boot/**



Once you have copied the iso we move to our next step to configure the network since I have single network card I would replicate a bridge the picture below shows current config

```
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 10
00
    link/ether 00:0c:29:eb:3e:b5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.100/24 brd 192.168.0.255 scope global noprefixroute ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feeb:3eb5/64 scope link
       valid_lft forever preferred_lft forever
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens33
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPADDR=192.168.0.100
PREFIX=24
GATEWAY=192.168.0.1
DNS1=8.8.8.8
DNS2=8.8.4.4
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=no
NAME=ens33
UUID=bc7efb23-7eb7-42af-8dfd-77bb34372b40
DEVICE=ens33
ONBOOT=yes
[root@localhost ~]# _
```

Now **vi /etc/sysconfig/network-scripts/ifcfg-ens33** and edit as below save changes

```
TYPE=Ethernet
BOOTPROTO=none
NAME=ens33
DEVICE=ens33
ONBOOT=yes
BRIDGE="br0"
~
~
~
```

Again **vi /etc/sysconfig/network-scripts/ifcfg-br0** there wont be any data but you'll need to fill it up accordingly to your ip configuration save changes

```
TYPE=Bridge
BOOTPROTO=static
NAME=br0
DEVICE=br0
ONBOOT=yes
IPADDR=192.168.0.100
PREFIX=24
GATEWAY=192.168.0.1
DNS1=8.8.8.8
DNS2=8.8.4.4
~
```

Then issue following command **systemctl restart network** for it to take effect on changes and run **ip a** to verify **br0** is up with the configured ip in my case its **192.168.0.100**

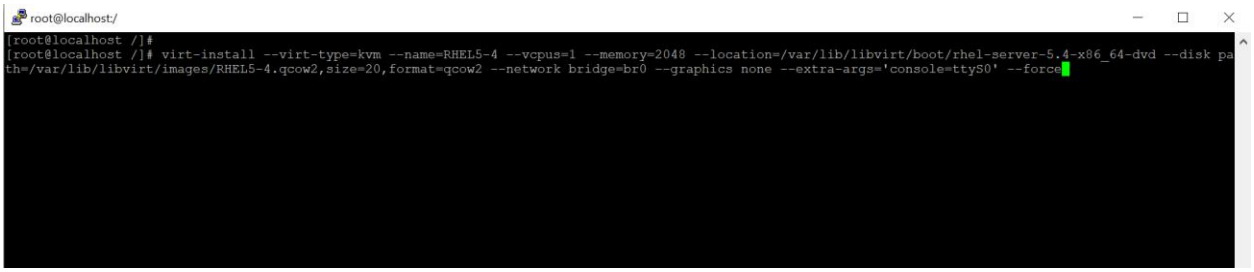
```
[root@localhost ~]# systemctl restart network
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br0 state UP group defa
ult qlen 1000
    link/ether 00:0c:29:eb:3e:b5 brd ff:ff:ff:ff:ff:ff
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen
1000
    link/ether 52:54:00:f2:a6:61 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN group defaul
t qlen 1000
    link/ether 52:54:00:f2:a6:61 brd ff:ff:ff:ff:ff:ff
7: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 00:0c:29:eb:3e:b5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.100/24 brd 192.168.0.255 scope global noprefixroute br0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feeb:3eb5/64 scope link
       valid_lft forever preferred_lft forever
[root@localhost ~]#
```

Now back to track

Issue the following command below to create vm in the KVM

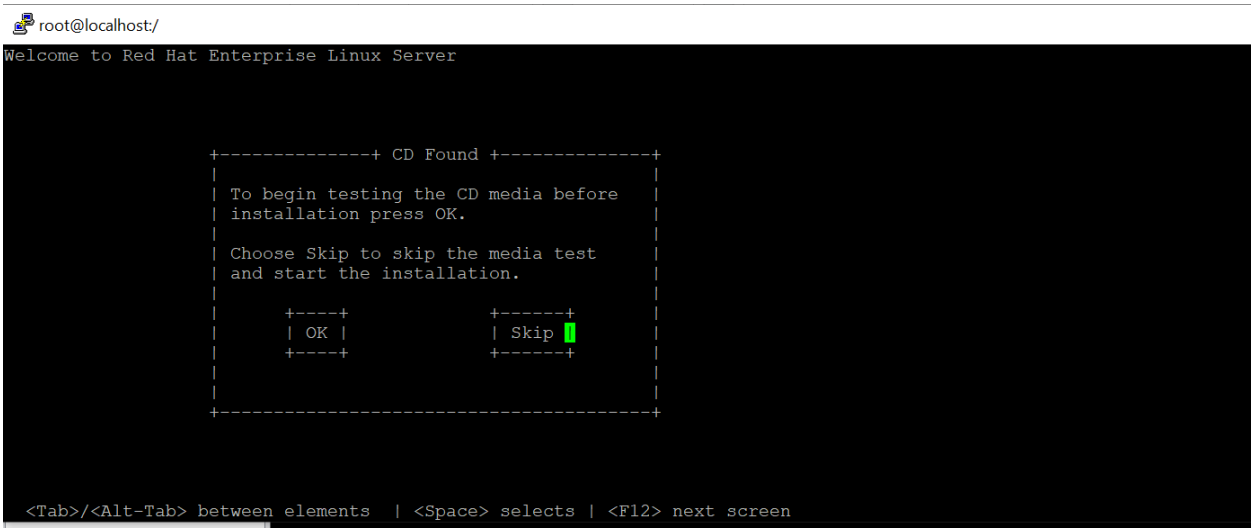
virt-install
--virt-type=kvm --name=RHEL5-4
--vcpus=1--memory=2048
--location=/var/lib/libvirt/boot/rhel-server-5.4-x86_64-dvd.iso
--disk path=/var/lib/libvirt/images/RHEL5-4.qcow2,size=20,format=qcow2
--network bridge=br0
--graphics none
--extra-args='console=ttyS0' –force

you'll need to do it in a line as per the picture below, I suggest you write in a notepad as a single line verify and then copy paste it…Press Enter!!!



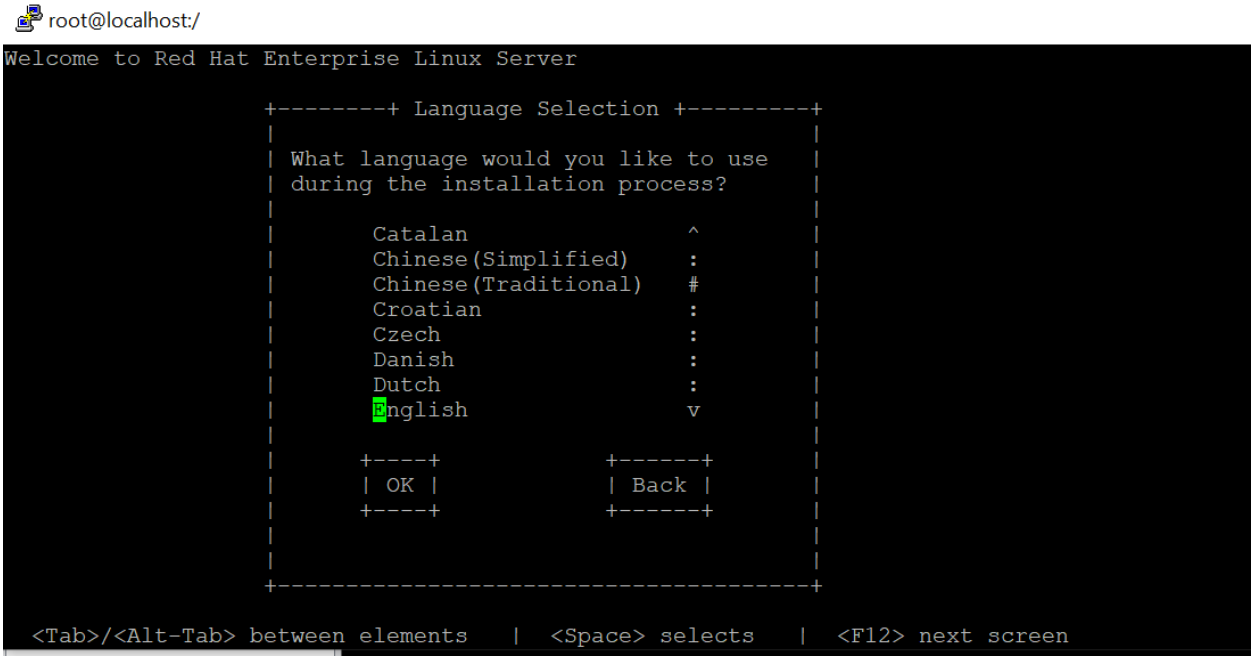
Viola it has started

It’ll prompt following installation

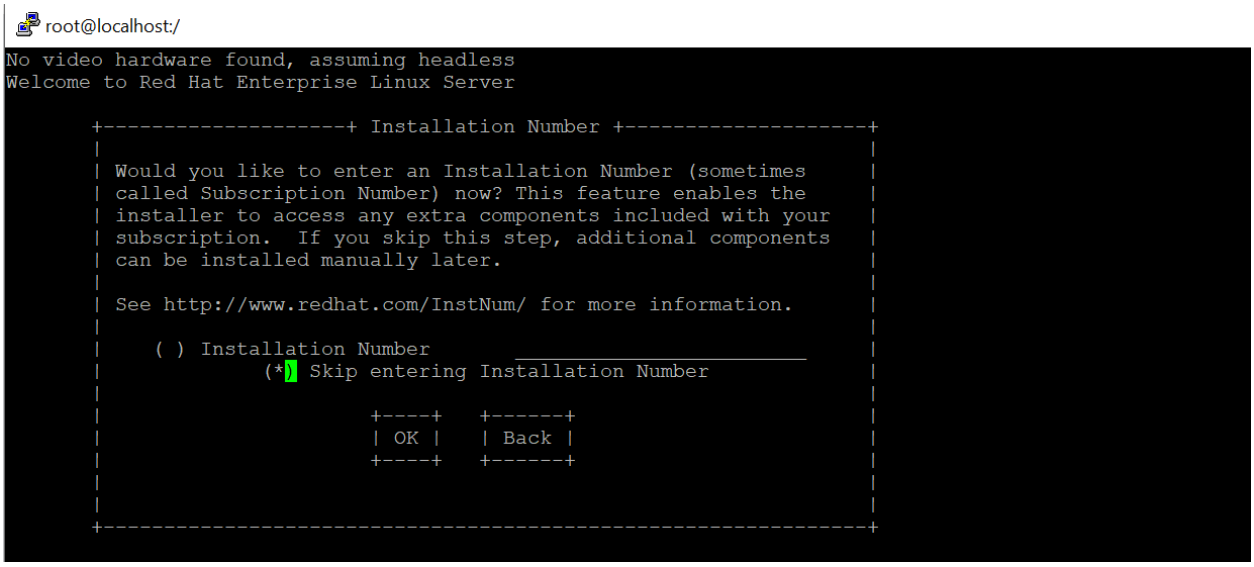


Installing you vm select ok

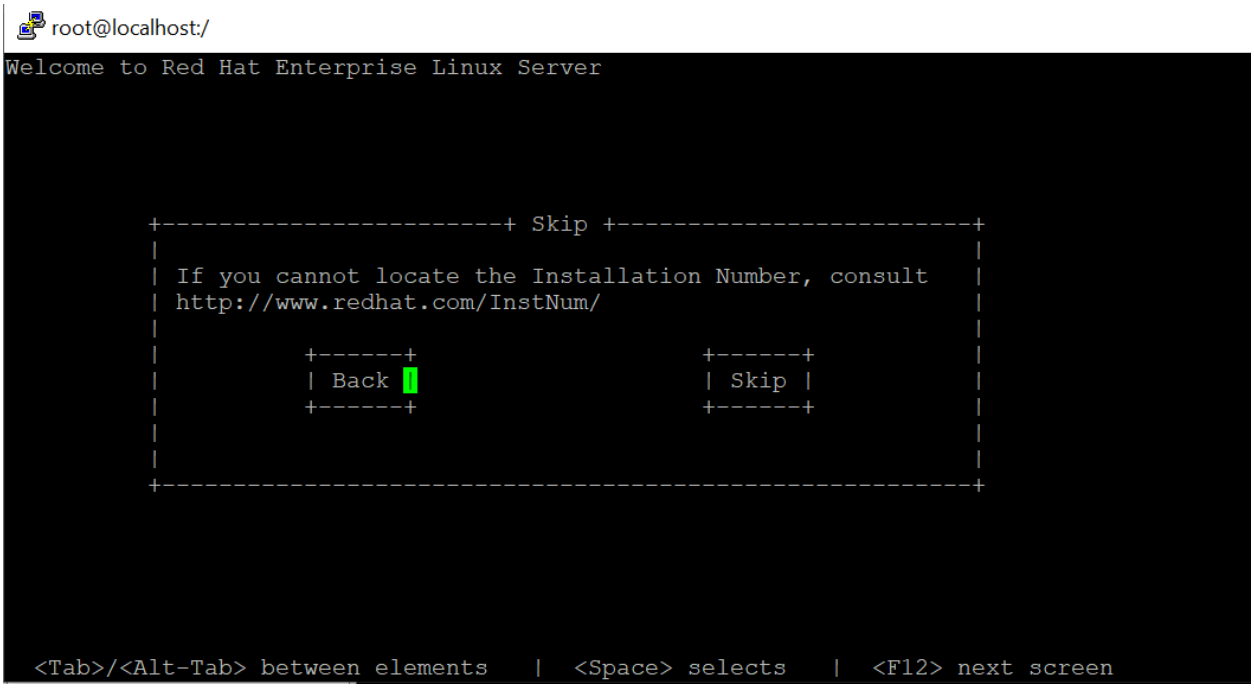
Press enter



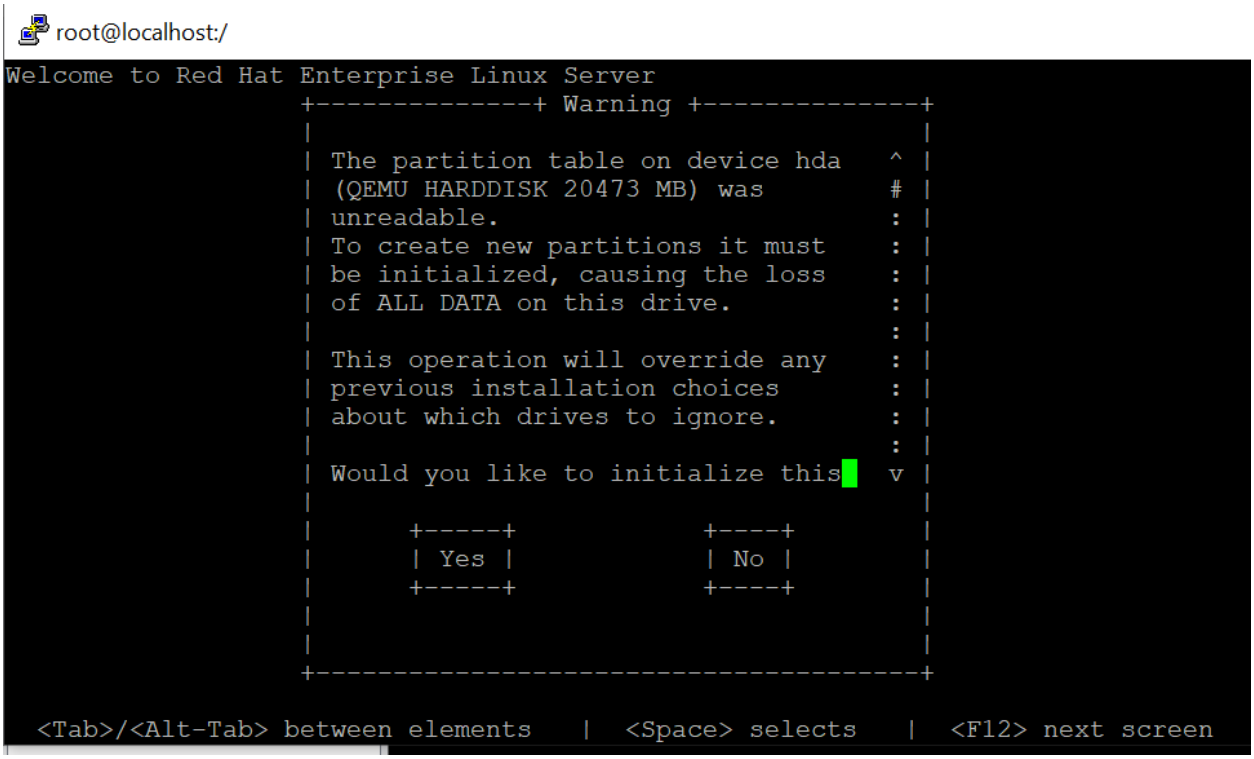
Select skip and tab to ok and press enter



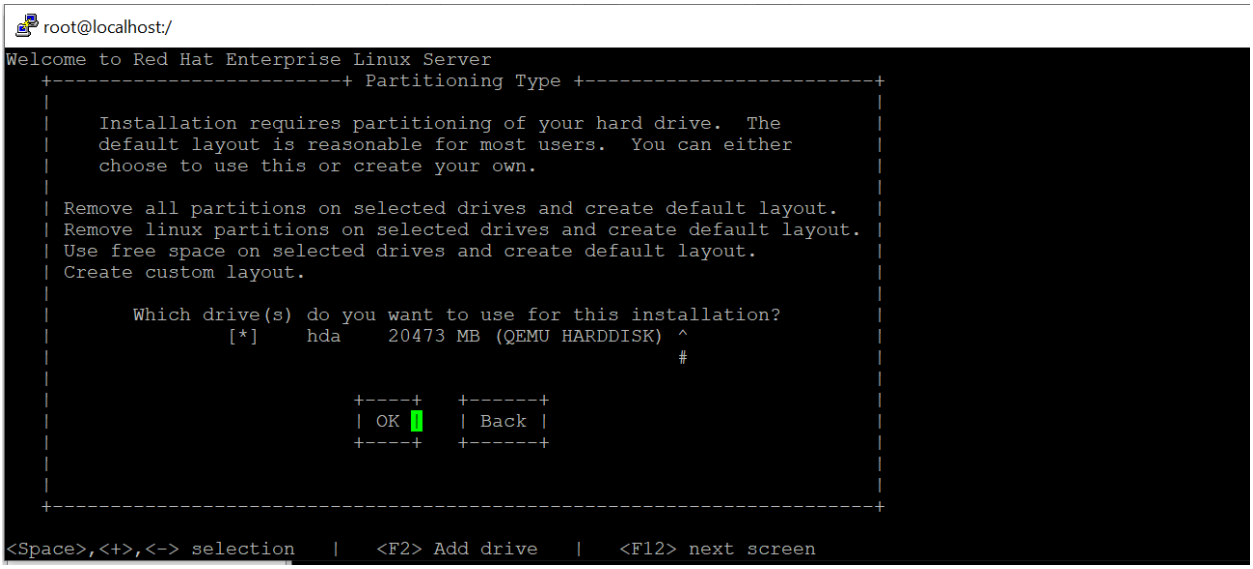
Skip this option as I do not have any subscription



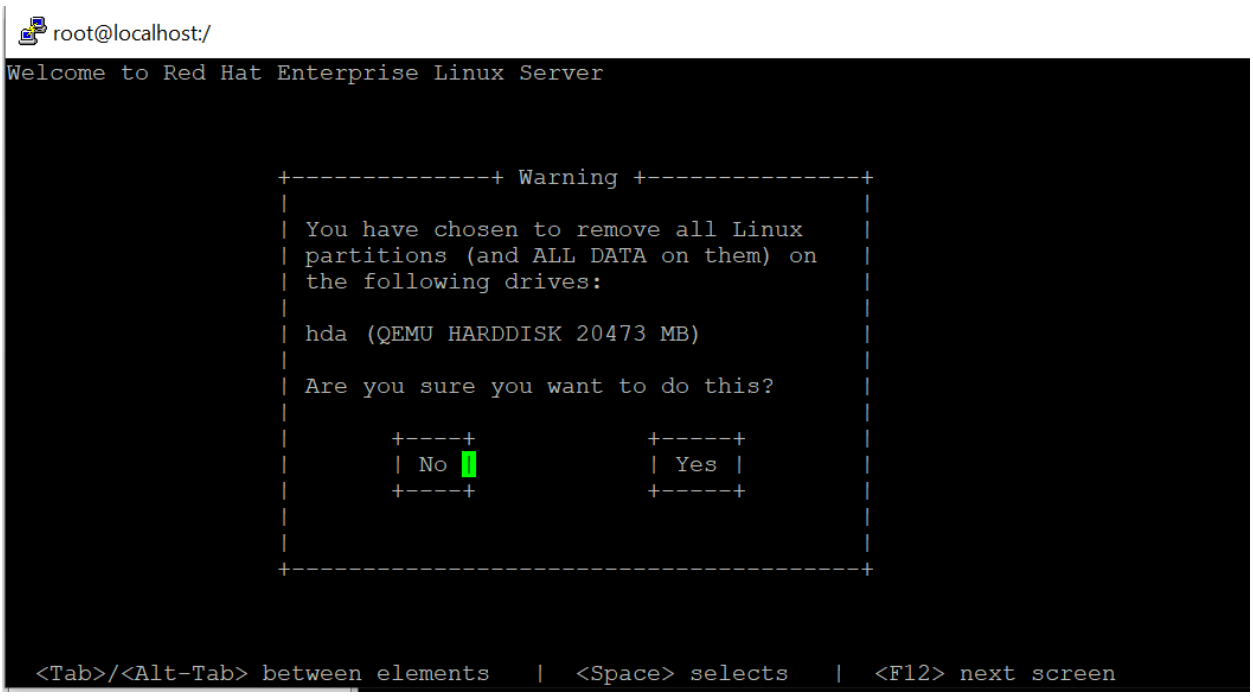
In the part below please select yes as might look back at my command (--disk path=/var/lib/libvirt/images/RHEL5-4.qcow2**,size=20**,format=qcow2) yup that's my 20GB



Confirmation on the disk press ok

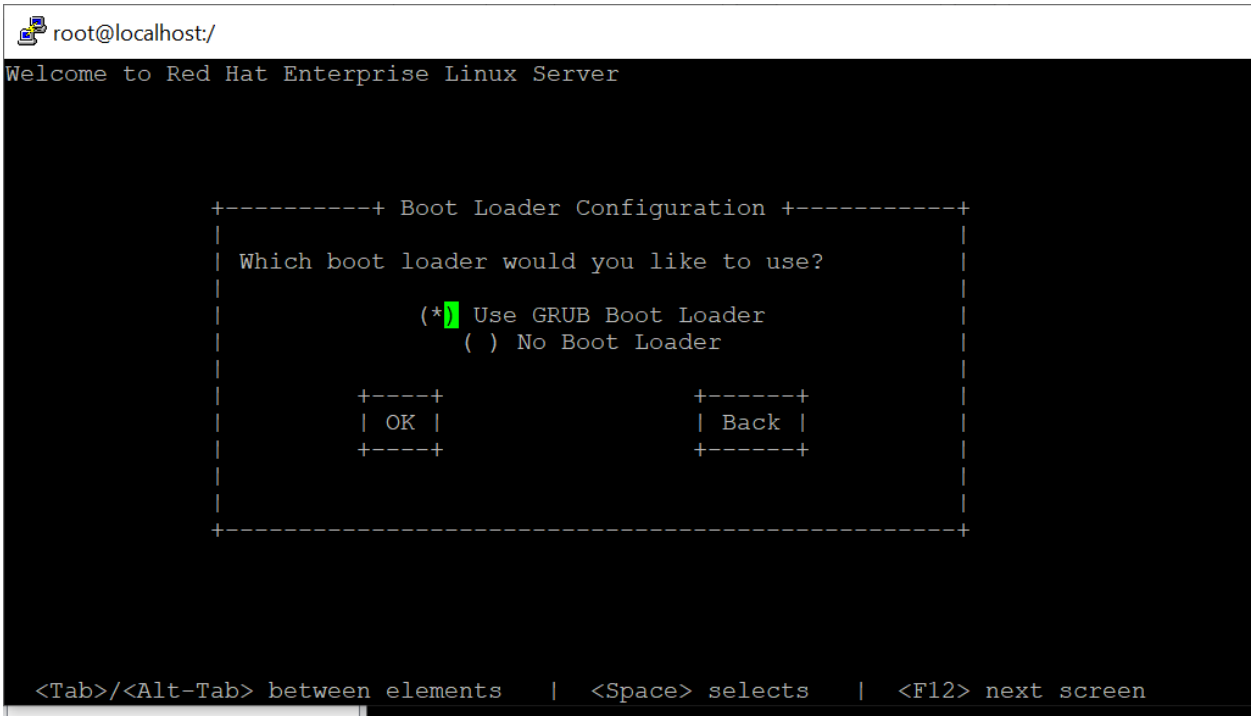


Ok for this as well

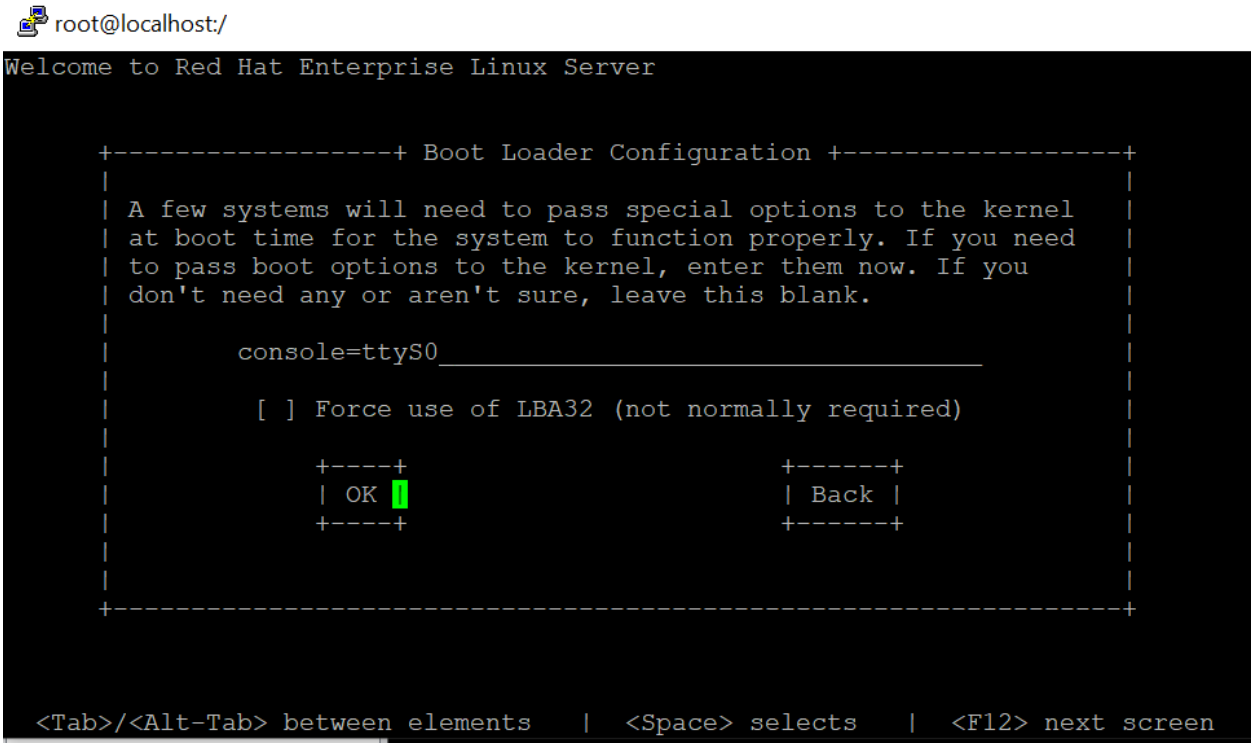


Since im using an older version you might need to create a partition and press ok

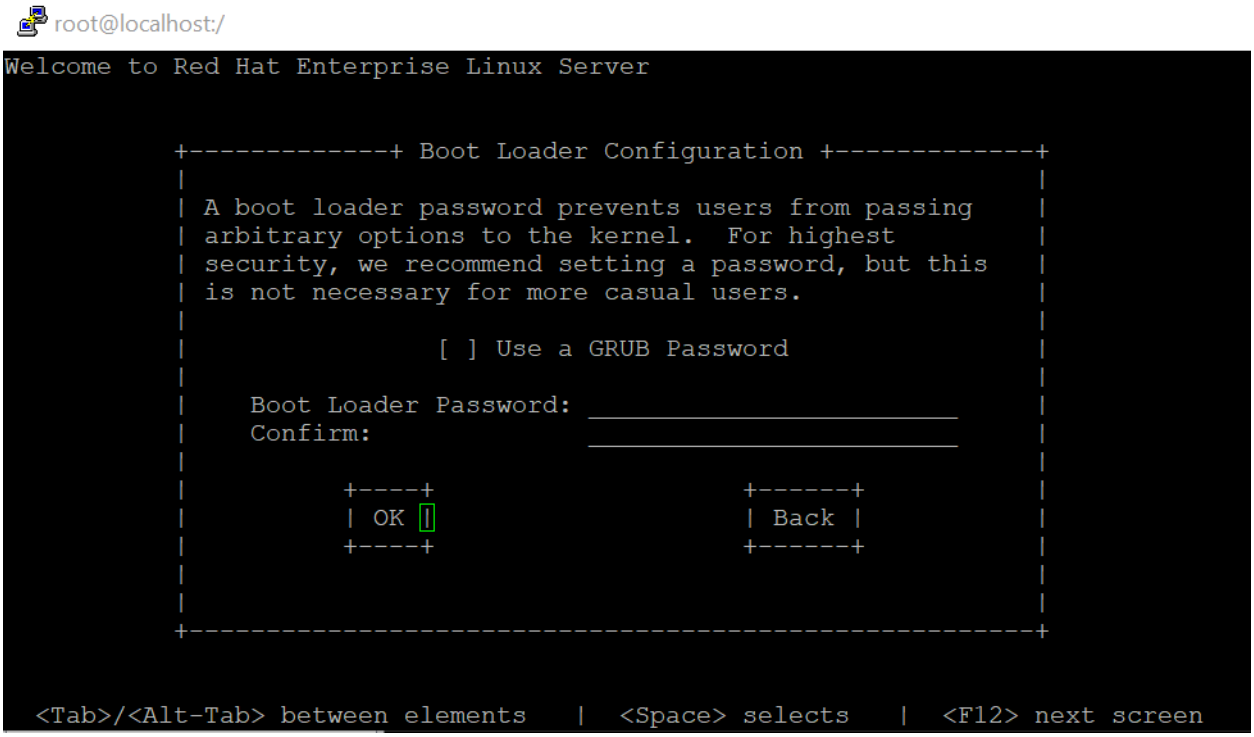
Select Grub Bootloader and ok



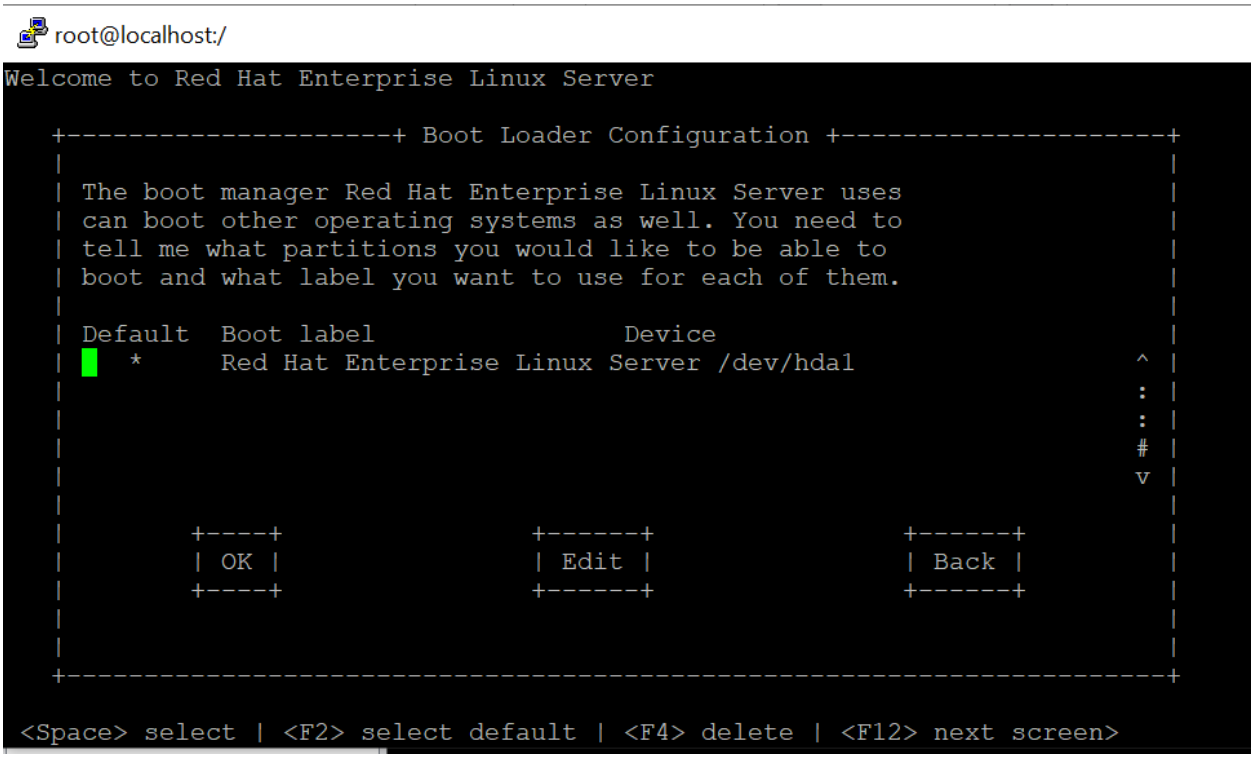
Ok for this



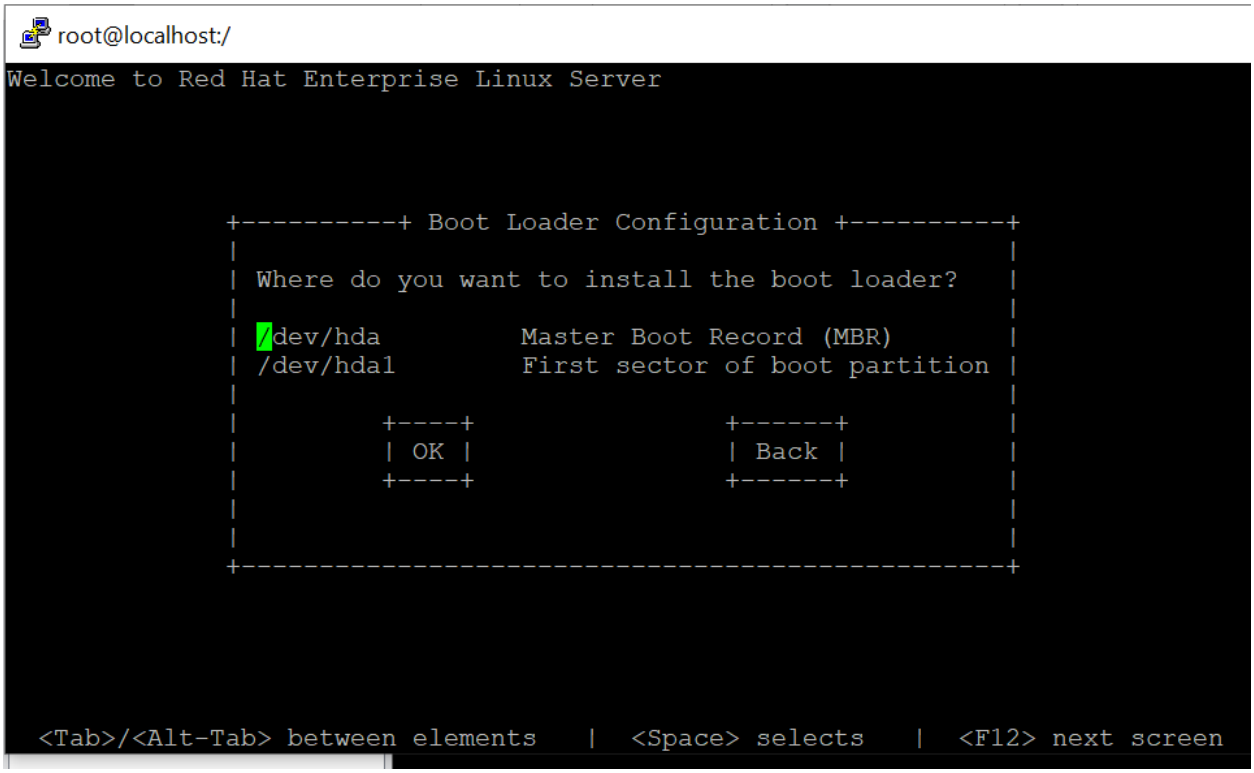
Im not going to setup password for this but you if you want too but make sure you remember



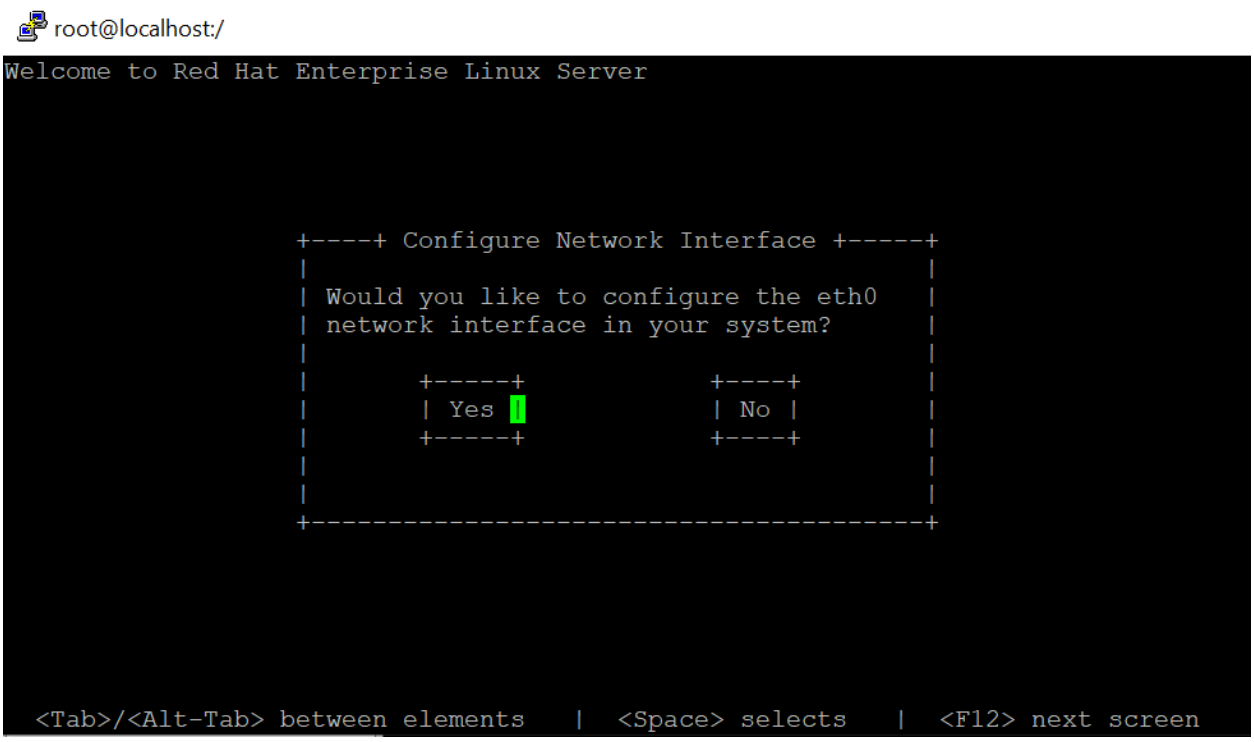
Select ok for this



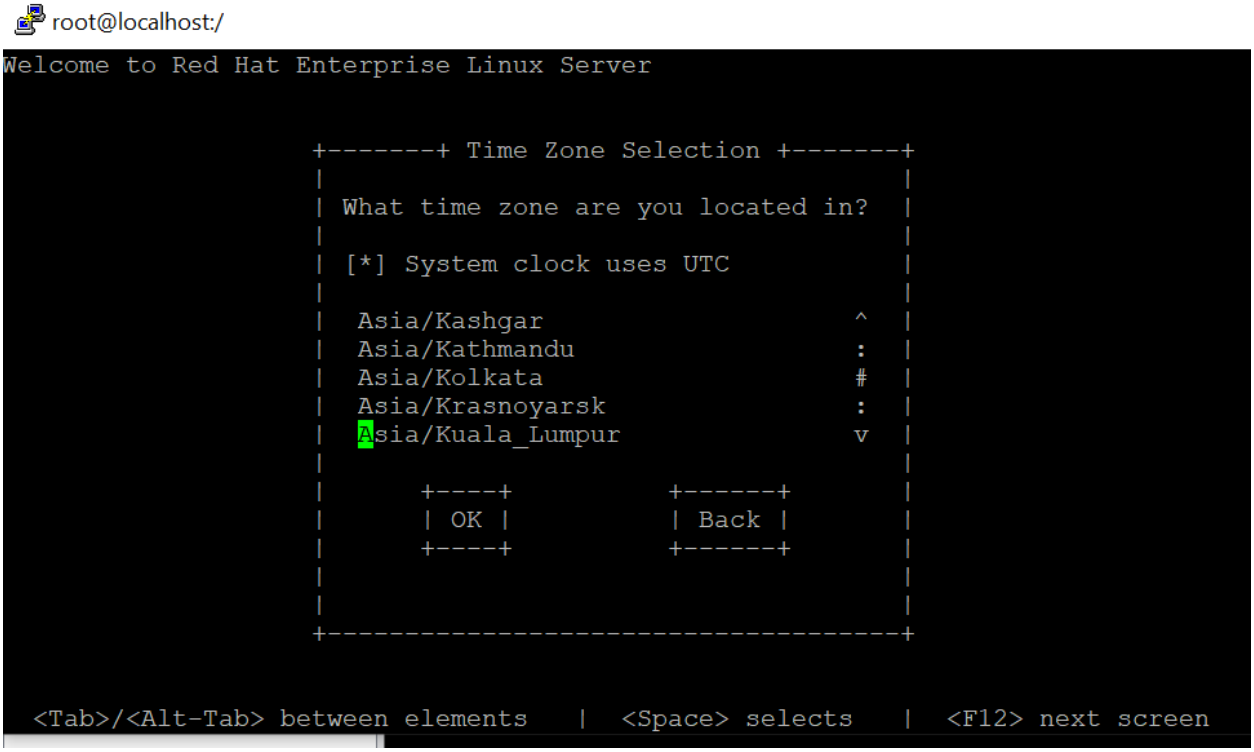
Select ok for this as well



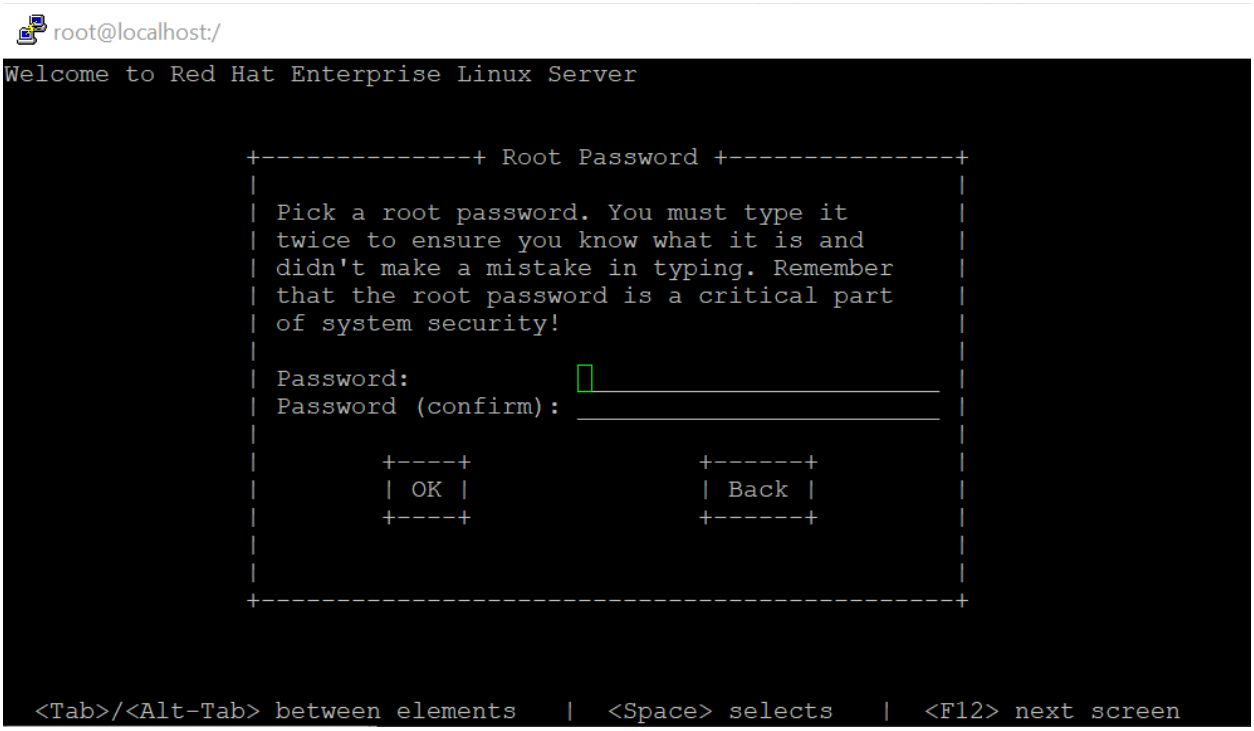
You can configure network now if you want or you could do it manually later as well



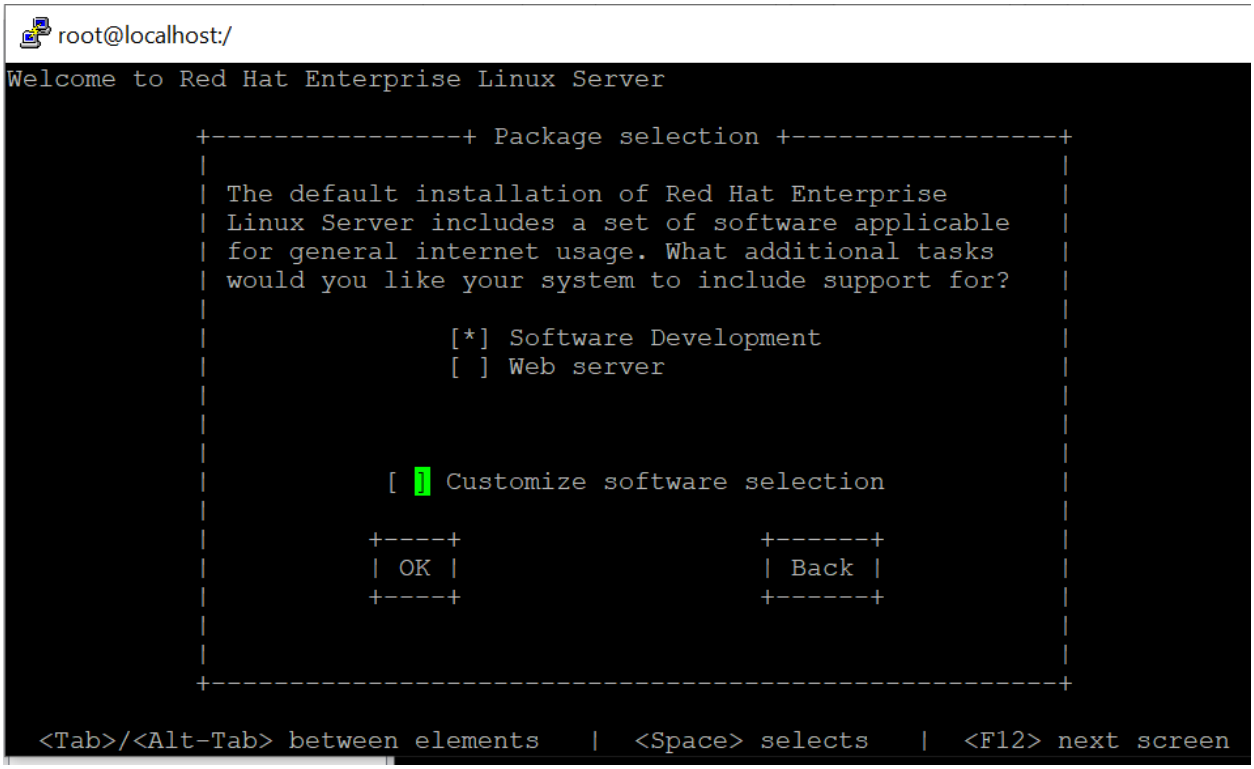
Select your Time Zone in my case im in Malaysia Kuala Lumpur



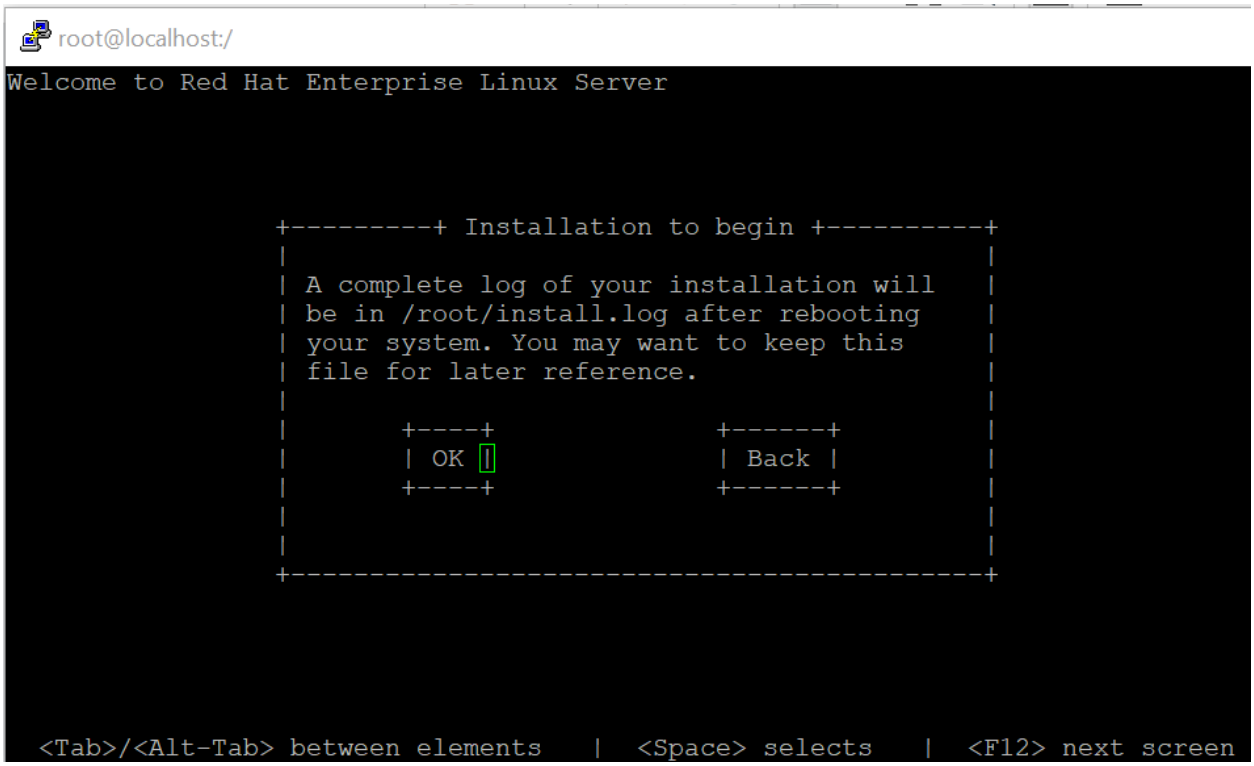
This is your server password make sure you remember when you set it



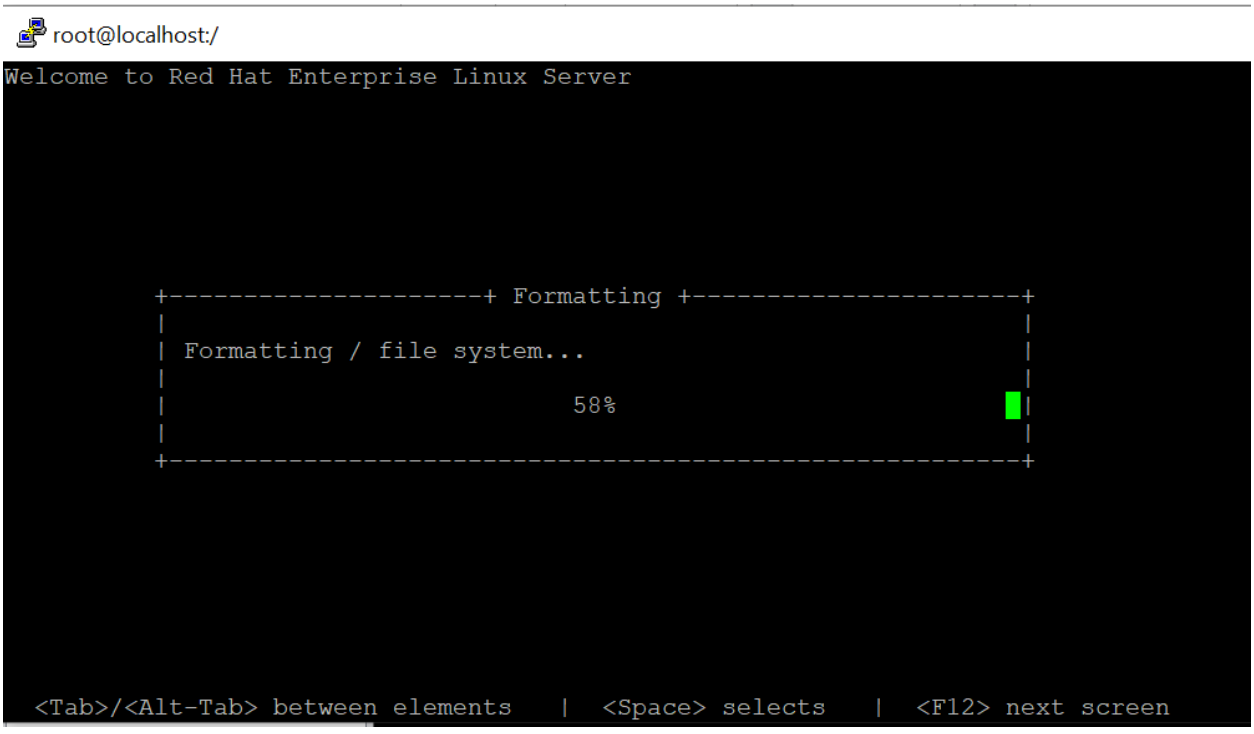
And the next screen it'll be package selection menu selection is up to you once done press ok



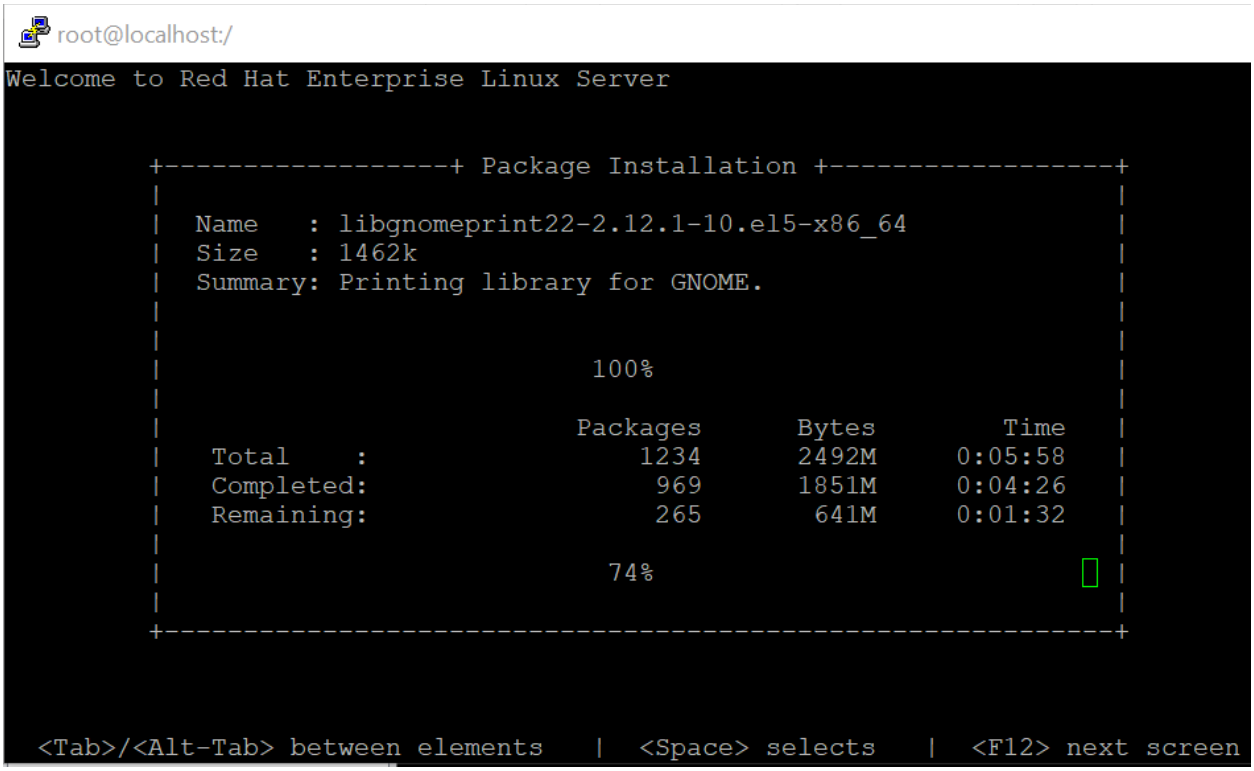
Press ok



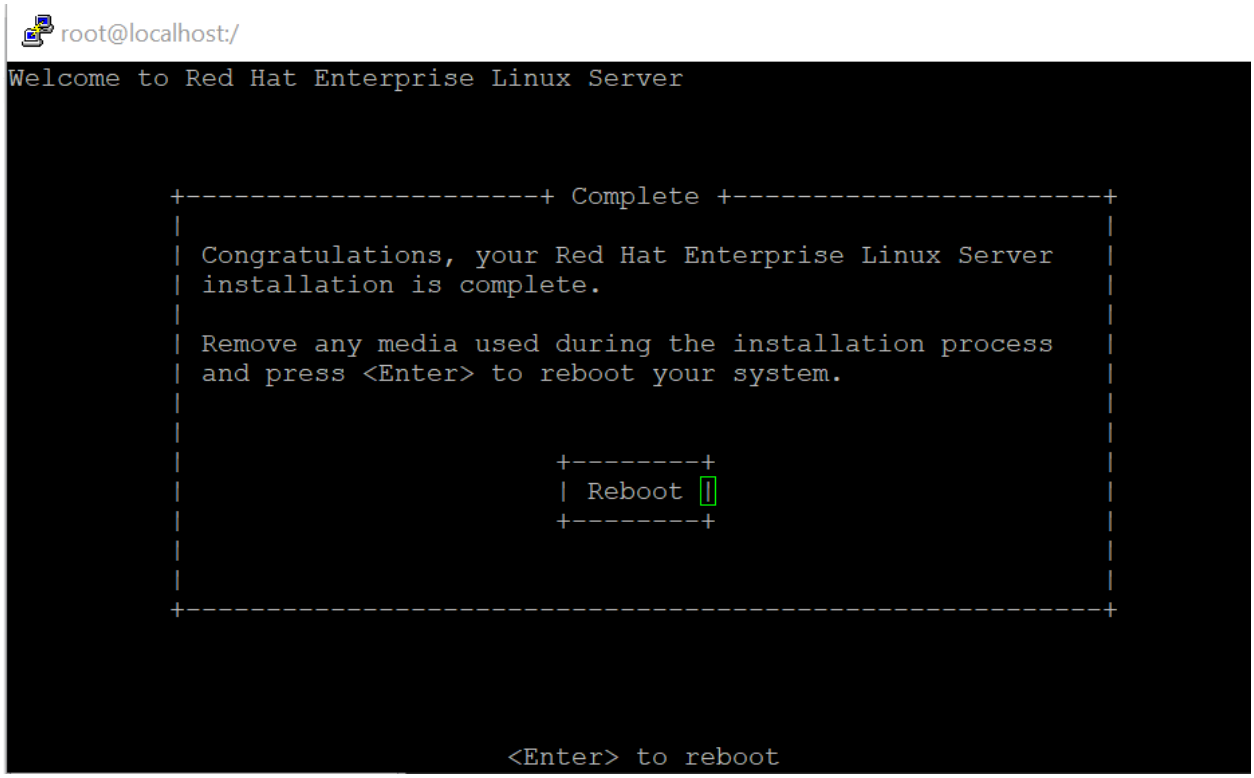
And it'll start installation



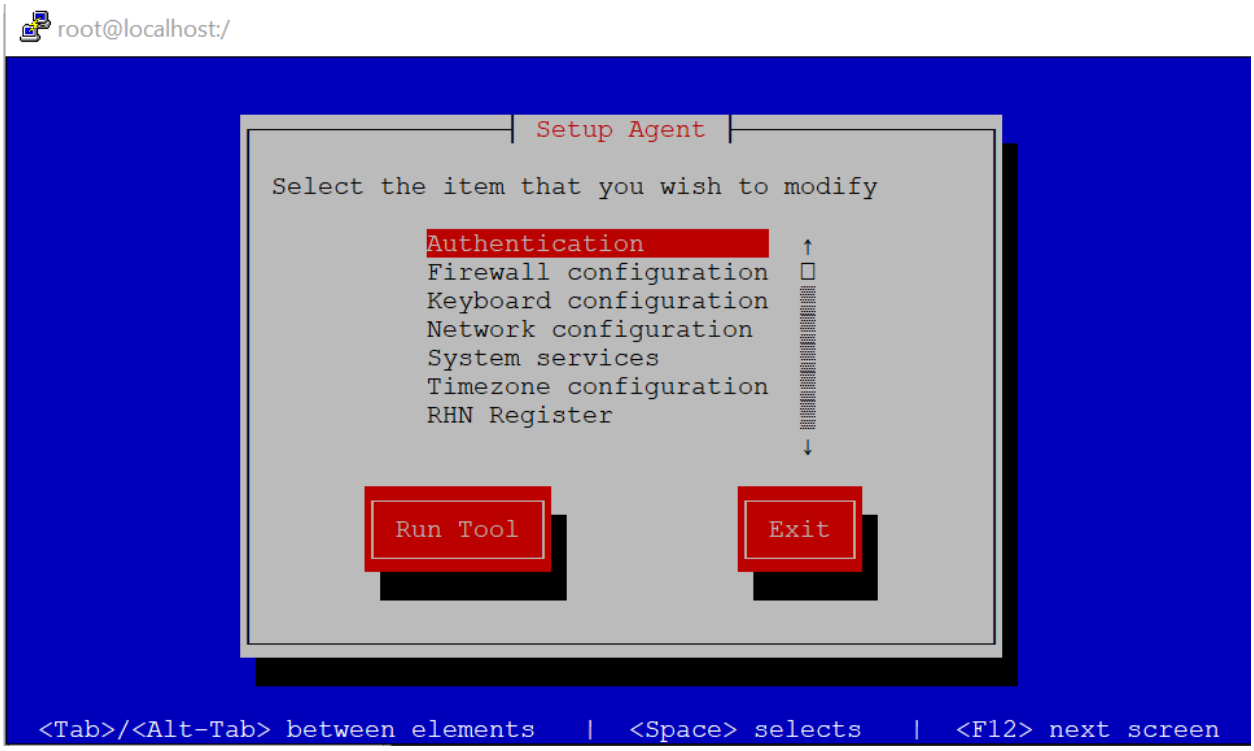
And we wait for completion



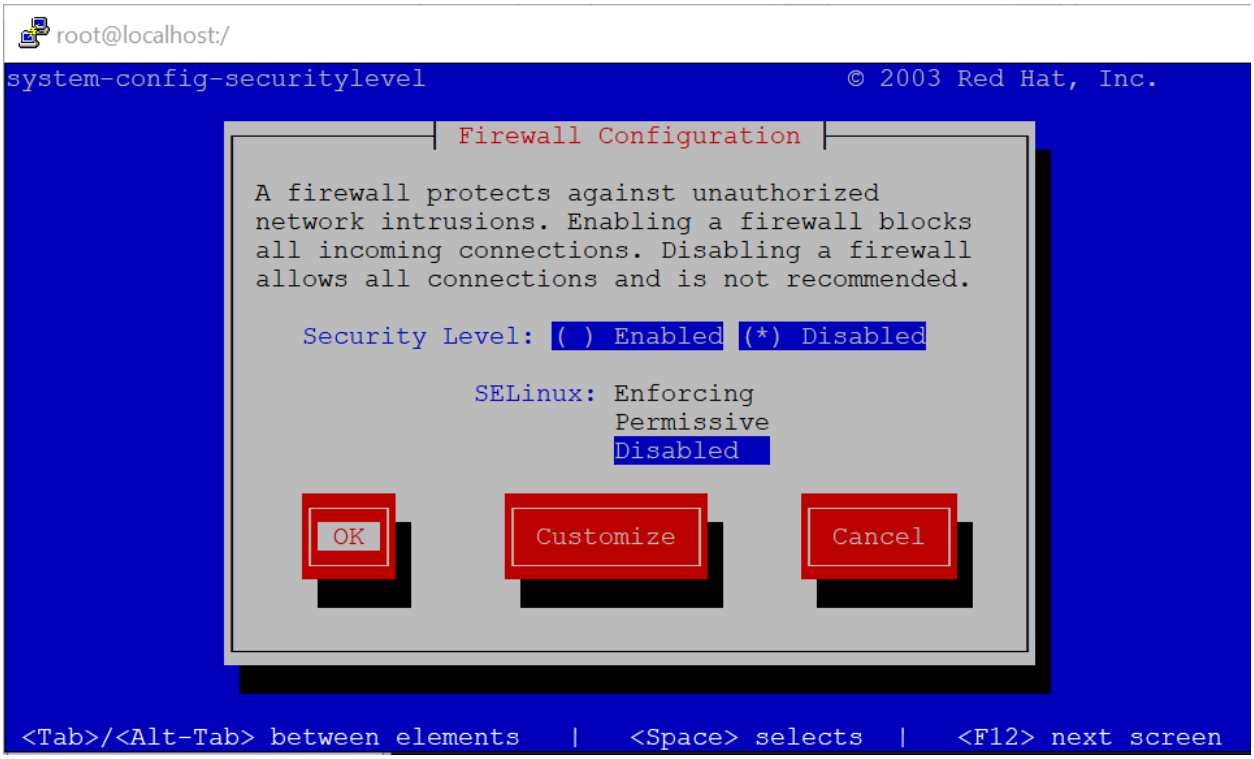
Finally enter to reboot



Since im installing RHEL older version installations methods might differ

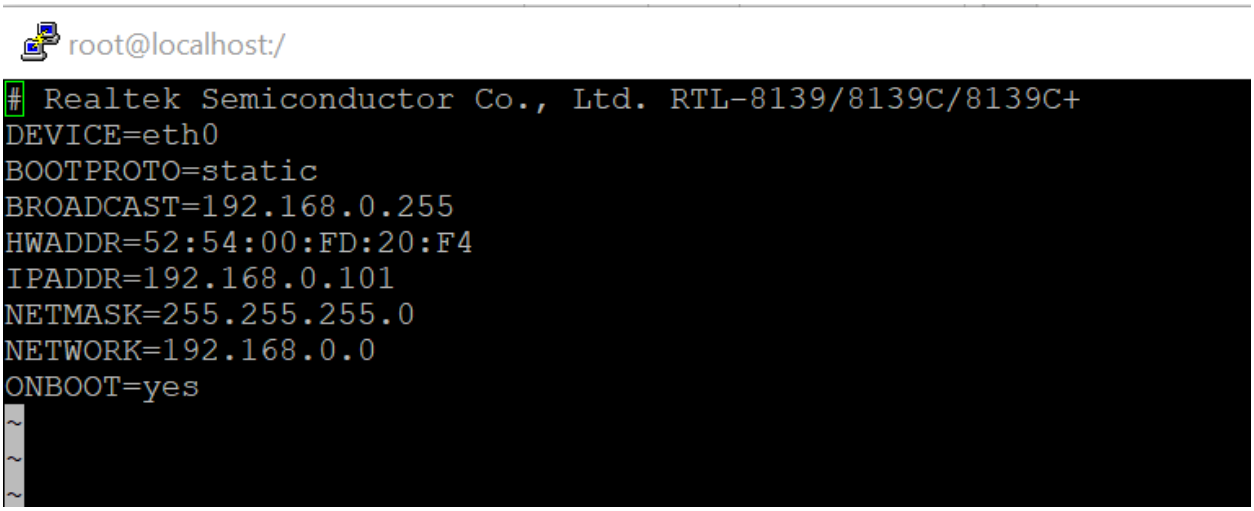


I just need you guys to follow this for the vm and then exit



And your set to go ignore the error login root and password

Login and **vi /etc/sysconfig/network-scripts/ifcfg-eth0** add the following details as below



Try pinging from KVM Server to VM

```
[root@localhost ~]# cat /etc/centos-release
CentOS Linux release 7.9.2009 (Core)
[root@localhost ~]# ping 192.168.0.101
PING 192.168.0.101 (192.168.0.101) 56(84) bytes of data.
64 bytes from 192.168.0.101: icmp_seq=1 ttl=64 time=2.92 ms
64 bytes from 192.168.0.101: icmp_seq=2 ttl=64 time=0.736 ms
^C
--- 192.168.0.101 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.736/1.829/2.922/1.093 ms
[root@localhost ~]#
```

And ping from VM to KVM Server

root@localhost:/

```
[root@rhel5-4 ~]# cat /etc/redhat-release
Red Hat Enterprise Linux Server release 5.4 (Tikanga)
[root@rhel5-4 ~]# ping 192.168.0.100
PING 192.168.0.100 (192.168.0.100) 56(84) bytes of data.
64 bytes from 192.168.0.100: icmp_seq=1 ttl=64 time=1.05 ms
64 bytes from 192.168.0.100: icmp_seq=2 ttl=64 time=2.55 ms
64 bytes from 192.168.0.100: icmp_seq=3 ttl=64 time=1.11 ms
64 bytes from 192.168.0.100: icmp_seq=4 ttl=64 time=1.17 ms

--- 192.168.0.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 1.054/1.474/2.554/0.625 ms
[root@rhel5-4 ~]#
```

Now that is done you have successfully setup a KVM server and VM

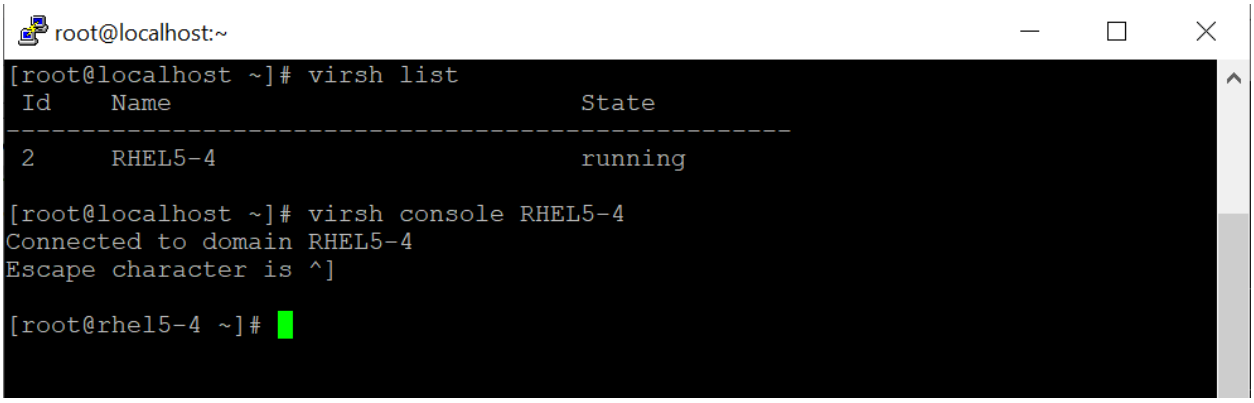
To verify list of VM you have in your KVM Server issue the following commands as below **`virsh list`** or **virsh list --all**

```
[root@localhost ~]# virsh list
 Id    Name                           State
----------------------------------------------------
 2     RHEL5-4                        running

[root@localhost ~]# virsh list --all
 Id    Name                           State
----------------------------------------------------
 2     RHEL5-4                        running

[root@localhost ~]#
```

Connect to a VM issue the following command **virsh console vm-name** in my case its **RHEL5-4**



In case if there's an issue connecting try killing the session and reconnect back by issuing **ps aux | grep console** and kill the session by process number **kill -9 4617**

```
[root@localhost ~]# ps aux | grep console
root      4016  0.0  1.3 655304 79152 pts/0    Sl+  21:25   0:01 /usr/bin/python2 /usr/share/virt-m
anager/virt-install --virt-type=kvm --name=RHEL5-4 --vcpus=1 --memory=2048 --location=/var/lib/libvi
rt/boot/rhel-server-5.4-x86_64-dvd.iso --disk path=/var/lib/libvirt/images/RHEL5-4.qcow2,size=20,for
mat=qcow2 --network bridge=br0 --graphics none --extra-args=console=ttyS0 --force
root      4617  0.0  0.1 347384  6900 pts/0    Sl+  22:10   0:00 virsh --connect qemu:///system con
sole RHEL5-4
root      5211  0.0  0.0 112812   968 tty1     S+   23:04   0:00 grep --color=auto console
[root@localhost ~]# kill -9 4617
[root@localhost ~]# virsh list --all
 Id    Name                           State
----------------------------------------------------
 2     RHEL5-4                        running

[root@localhost ~]# ps aux | grep console
root      5224  0.0  0.0 112808   964 tty1     S+   23:05   0:00 grep --color=auto console
[root@localhost ~]#
```

If you get the steps right you could ssh directly into the VM itself