



Create your Centos homelab in an hour.

Slides available at
github.com/murphnj/hourlab

Introductions will happen...
while we install our first VM.



Who is this?

- Bob Murphy
 - Linux user since Slackware 3.6
 - Linux Sysadmin
 - Current RHCE student
 - Contact
 - murph@member.fsf.org
 - Fediverse: @murph@fosstodon.org



Why set up a homelab like this?

- Allows easy setup and rebuilding
- Stability in versions of OS
- No network needed for installing new packages.



What is required:

- A downloaded .iso file of the distro that you want to run your lab on (RedHat, Centos, Scientific or a related version)
- A host computer with virtualization software loaded.:
- My example will use Fedora for the host, and KVM for the virtualization software.
- Centos would work basically the same.
- Other computers such as Windows or Mac would be OK, but you will have to change a few of the implementation details to suit your platform.



First, we'll create our local repository VM.

- Start virt-manager, and select a new VM.
- Select "Local install media" and Press "Forward"
- Select "Use ISO image" and browse.
- At the "Choose local Storage Volume" select "Browse local" and navigate to the .iso for your distro.
- Select "Forward"



VM details

- Choose your memory and CPU options, 1G and 1 cpu is fine for the repo, less may even work. Select Forward
- For the Disk image, I'd recommend about 16G, so that you have room for the files. Select forward
- Then name your VM, something that will show you that it is the repo. Press finish



Now, install the repo OS

- Select Install CentOS 7.
- Select your language and continue
- Select installation destination, allow the installer to do Automatic installation, and click done in the upper left hand corner.
- Under Software installation, select "infrastructure server" and be sure to select the "FTP server" checkbox, and click "done"
- "Network and host name" , make sure that ethernet is enabled.



Continue installing

- Click “Begin Installation”
- You have an opportunity to create accounts and passwords for root, and a user account, this is a good opportunity to do so.
- Now we wait (5 minutes or so.)
- When it is finished, select Reboot
- When it reboots, log in with the root account that you set.



What we're doing

- Create new repo server
 - Start and enable FTP service
 - Open firewall port for ftp
 - Copy files from .iso file to server
 - Create repo file to tell OS to use ftp for getting updates and software

Setting up your ftp server

- Start the ftp server : `systemctl start vsftpd`
- Enable it for next time : `systemctl enable vsftpd`
- Check it : `systemctl status vsftpd`
- Set firewall :
`firewall-cmd --add-service=ftp --permanent`
- Enable firewall changes : `firewall-cmd --reload`
- Get your IP address: `ip a`
(We will need it in a few minutes)



Copy files for repo

- Go to "View->details" select ide cdrom1 - click connect - browse (just like the initial iso selection) click OK
- Go to "View->console" to go back to your VM.
- `mkdir /root/temp ; mount /dev/cdrom /temp;`
- `rsync -avhP /temp/ /var/ftp/pub/`
- We wait.



Create the local repo

- `cd /etc/yum.repos.d ; mv * ~`
- `<edit> network.repo`
- ***[network]***
- ***name=network***
- ***baseurl=ftp://192.168.122.<your ip>/pub***
- ***gpgcheck=0***
- save and quit
- `Yum clean all ; yum install ftp ; ftp localhost`



Repo done ; new VM

- Go back to the Virtual machine manager:
- Select a new VM as before, but instead of local CD, we are going to select a network install:
- For URL, we will type:
`ftp://192.168.122.<repo>/pub`
- Memory, cpus, and disk to taste, forward, forward ,
Name the VM, and Finish
- Time for questions.

Make new VM use repo

- We can copy the repo file from our server, to use here.
- `scp root@192.168.122.<repo ip>:/etc/yum.repos.d/network.repo /etc/yum.repos.d/`
- ***or, recreate it as follows:***
- ***[network]***
- ***name=network***
- ***baseurl=ftp://192.168.122.<repo ip>/pub***
- ***Gpgcheck=0***
- `yum clean all ; yum install screen`



Success!

- We now have a complete, self contained Linux network, with local repositories, so that we can install software without a network connection.
- All new VMs will now get new software from our local repo, not the repositories on the internet, so all versions will stay in sync.



Questions

- Slides available at github.com/murphnj/hourlab
- Contact me at:
- murph@member.fsf.org <email>
- fosstodon.org/@murph <fediverse>