

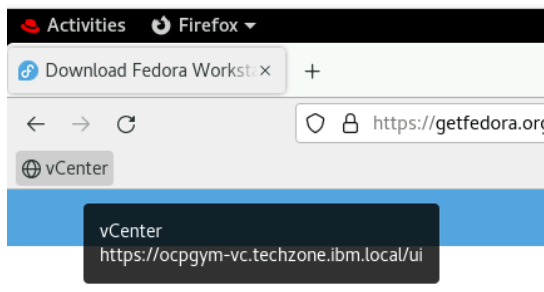
This second paper helps those that don't have a powerful laptop computer to install OpenShift Local configured for MAS Local as described [here](#) and assumes you have reserved an OCP Gymnasium environment, and possibly, also installed a SNO with MAS in there (what the first paper helps with: <https://github.com/evilADevil/mas-techzone/blob/main/MAS%20SNO%20on%20TechZone.pdf>).

We are going to use a Fedora Workstation to install OCP Local and then MAS Manage.

Get a remote console to the Bastion VM, open the Firefox browser and navigate to the Fedora download page:

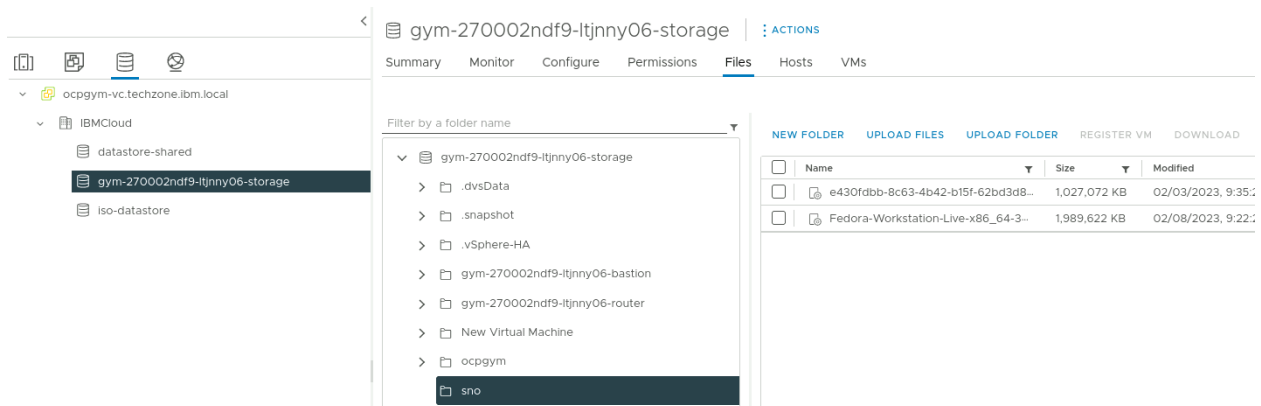
<https://getfedora.org/en/workstation/download/>

Download the x86_64 Linux ISO. Once the download has finished, use the "vCenter" prepopulated bookmark in Firefox to open the vSphere vCenter client window

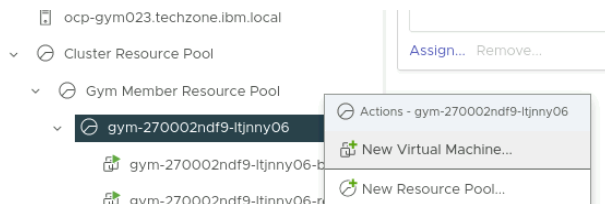


Log into vCenter using the credentials available in your reservation page.

Upload the downloaded Fedora ISO into the same "sno" folder where you uploaded the SNO iso:







Use the vCenter to create a new VM that will install and run Fedora.



Use the same steps used for the SNO VM. Call the VM "Fedora".

Virtual machine name: Fedora

Select a location for the virtual machine.

- ✓  ocpgym-vc.techzone.ibm.local
- ✓  IBMCloud
- ✓  ocp-gym
 -  gym-270002ndf9-ltjnnY06

The VM configuration must be modified with CPU to 24, Memory to 128 GB and Hard Disk 1 500 GB as shown in the following picture:

Customize hardware

Configure the virtual machine hardware

Virtual Hardware

VM Options

ADD NEW DEVICE

> CPU *	24		
> Memory *	128	GB	
> New Hard disk *	500	GB	
> New SCSI controller *	VMware Paravirtual		
> New Network *	<input checked="" type="checkbox"/> Connect		

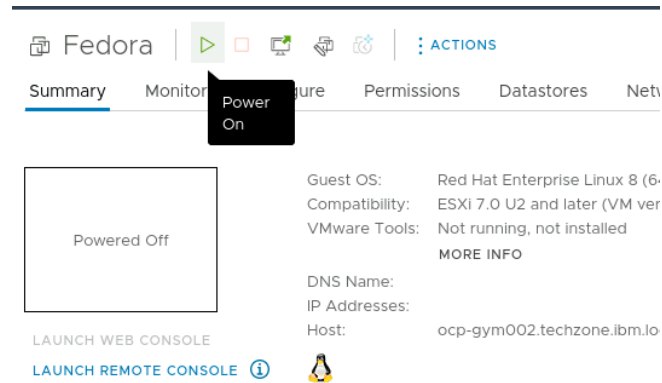
Then open the CPU section and make sure to check the “Expose hardware assisted virtualization to guest OS” and “Enabled” for I/O MMU.

CPU		24		
Cores per Socket	1		Sockets: 24	
CPU Hot Plug	<input type="checkbox"/> Enable CPU Hot Add			
Reservation	0		MHz	
Limit	Unlimited		MHz	
Shares	Normal	24000		
Hardware virtualization	<input checked="" type="checkbox"/> Expose hardware assisted virtualization to the guest OS			
Performance Counters	<input type="checkbox"/> Enable virtualized CPU performance counters			
I/O MMU	<input checked="" type="checkbox"/> Enabled			
Memory	128		GB	

Make sure to select the Fedora install ISO in the “New CD/DVD Drive” and make sure the “Connect...” checkmark is checked.

Click on the Next button at the bottom and then the Finish button to complete the creation step.

Power ON the Fedora VM using the green “play” icon near the VM name:



Click on the “LAUNCH WEB CONSOLE” link to connect to the VM console and complete the Fedora installation on the Hard Drive. When the installation completes, reboot the VM and define a user with a password.

Make sure you install all the latest Fedora patches, because I’ve noticed that if you are not at the latest level of Fedora, you may have problems with starting OCP Local.

Open a terminal from the Gnome desktop and enable and start the ssh daemon using the following commands:

```
sudo systemctl enable sshd
sudo systemctl start sshd
```

At this point you can ssh into the Fedora VM from your computer using the user you have defined and its IP address that you see in the vCenter page.

```
ssh alex@192.168.252.106
```

Next step is to install the OCP CLI. Issue the following commands to download OCP Local, unpack it and install it:

```
mkdir -p ~/bin
export PATH=$PATH:$HOME/bin
echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
curl -O https://mirror.openshift.com/pub/openshift-
v4/x86_64/clients/ocp/stable/openshift-client-linux.tar.gz
tar xf openshift-client-linux.tar.gz -C ~/bin
```

Next step is to install OCP Local. These are the commands:

```
curl -LO https://developers.redhat.com/content-
gateway/rest/mirror/pub/openshift-v4/clients/crc/2.12.0/crc-linux-
amd64.tar.xz
tar xf crc-linux-amd64.tar.xz
cp crc-linux-2.12.0-amd64/crc bin
```

```
crc config set consent-telemetry no
crc config set cpus 14
crc config set memory 30720
crc config set disk-size 200
crc setup
```

Next step is to download the MAS Local helper files. We need git for that. This is the set of commands to use:

```
sudo dnf -y install git-all
git clone https://github.com/evilADevil/mas-local
chmod 755 mas-local/masinst.sh
```

At this point follow the instructions in the paragraph at this link to configure your installation. I would suggest you to use UDS running in the SNO you installed previously.

<https://github.com/evilADevil/mas-local#what-you-would-need-to-run-mas-manage>

Remember that you can scp files into the Fedora VM. Therefore you can issue commands like from your computer:

```
scp license.dat alex@192.168.252.106:/home/alex/mas-local
```

Now everything should be set up to start OCP Local and install MAS. We'll proceed using the Fedora VM remote console, therefore you should exit from the ssh session.

In the remote console, open a terminal and start OpenShift Local using the following command:

```
crc start
```

You will be asked for a pull secret that you can copy from this web page:

```
https://console.redhat.com/openshift/create/local
```

OCP Local may take some attempts to start. Keep in mind that this is a virtualized environment contained in another virtualized environment built just for the purpose to show how OCP Local works and can be used to demo MAS Manage. In a normal computer all these problems do not happen.

I install and use the Virtual Machine Manager to see what it is going on in the crc virtual machine. Sometimes you need to crc stop, crc delete and restart from the beginning to make it start properly. After the first successful start, it will come up without problems every time, which means the initial problems are due to the virtual environment initialization.

When the crc starts, you will see a message like this that will provide you with info on how to log into crc and launch the console:

```
Started the OpenShift cluster.
```

```
The server is accessible via web console at:
https://console-openshift-console.apps-crc.testing
```

```
Log in as administrator:
Username: kubeadmin
Password: sxAIq-i2XFV-B7KwN-JxWJw
```

Log in as user:

Username: developer

Password: developer

Use the 'oc' command line interface:

```
$ eval $(crc oc-env)
```

```
$ oc login -u developer https://api.crc.testing:6443
```

Now that you have the OpenShift Local running, you can open the OpenShift web console to follow the MAS installation we will start shortly.

In the terminal windows, issues commands like these to login as kubeadmin.

```
eval $(crc oc-env)
```

```
oc login -u kubeadmin -p sxAIq-i2XFV-B7KwN-JxWJw https://api.crc.testing:6443
```

Then start the MAS install by run masinst.sh

```
./masinst.sh
```

In case you want so see a recording of a full install on a Windows computer, you'll find it here:

<https://youtu.be/LrbPGrxxAvo>