

Windows Instances first Setup On Radiant

Just a quick write-up of the things I used when setting up an initial instance of Windows 10 for testing on Radiant. This process could be done numerous different ways, simply the route I followed.

For example, I need to look into slipstreaming some of the necessary drivers into a custom windows ISO, which would be more space efficient than the . qcow2 approach I follow below.

Image Creation:

Other approaches can be used such as Virtual Box to create this image, but I primarily followed this guide with slight modifications: [Create Windows 10 Image in OpenStack](#),

- Used the windows 10 ISO from the UIUC Webstore but did use the provided VirtIO drivers.
- Adjusted the volume disk size based on what the specific version of Windows needs rather than the default 20GB
- Stop right before the "Prepare the Windows guest OS for Cloud Use" section as CloudBaseInit is not necessary if using the normal Campus /NCSA AD system. If want to connect, will likely want to connect, and then run sysprep in the image, and shutdown it down.

The newly created image will be located at `/var/lib/libvirt/images`

Additionally, theoretically after the first image was setup, can copy the volumes / instances as needed with those configurations.

Radiant instructions:

After loading the Radiant web interface

1. Click on the **images** tab, then the **create image** button
 - a. Add name and description as appropriate
 - b. Select the new Image as the Image source
 - c. Select the format as QCOW2
 - d. Allocate minimum disk and RAM size as wanted
 - e. Mark visibility as public and not protected (to follow convention of other images)
 - f. Edit Metadata tab as applicable (default should be okay though)
 - g. Click **Create Image** and just wait for it to finish uploading (may take a while)
2. Click on the **Volumes** tab, then create Volume
 - a. Add name and description as appropriate
 - b. Select Image as Volume source, then select the newly uploaded image
 - c. Adjust size if needed
 - d. Click **Create Volume** and wait for it to finish (may take a while).

3. Click on the **Instances** tab, then **Launch Image**

Details - set name as appropriate

Source - choose volume and then add the one that was just created to the available section. Additionally I left as do not delete volume on instance delete

Flavor - choose a node that has enough CPU, RAM and disk for necessary need so it in the available section.

Network - add bbji-net to give it only access to NCSA internal network (thus users need to be on VPN to RDP in)

Security Groups - add at least RDP_for_Windows. Maybe additionally remote HTTP/HTTPS or others as needed. (if additional groups or ports needed, see note below).

Key Pair - Add SSH key pair if wanted.

Configuration - Any custom script needed, and mark Configuration Drive if needed

Everything else change if needed, then click **Launch Image** and wait for it to spawn / be built

Next click the drop down next to the action bar for the instance, and select **Add Floating IP**, then select the floating IP, and the associated port for that instance, and click **Associate**.

4. Click on the instance itself in the stances tab and in the console portion, there is a copy of the screen that can be used to make sure RDP is enabled and any other changes that need done. Technically after RDP is enabled, can just use a normal RDP client as it will be a much smoother experience.

And that should be a functional Window Instance on Radiant, which can be configured as needed.

Other Notes:

If other ports need opened or anything, go into the **Security Groups** tab under **Networking**, then click **Create new Security Group** or modify an existing one and add the necessary protocols and ports. Then go back to instances, and under the actions drop down, click **edit instance**, go into **Security Groups** tab and add the new group.

Alternatively, this documentation [OpenStack Windows Documentation](#) also exists from OpenStack to create a similar image, specifically discussing for Windows Server 2012, and mostly via CLI.