

By *enomaly*

Published: 2006-11-24 14:14

Running Windows on the Amazon Elastic Compute Cloud with Qemu and Linux

Level: Introductory

November 05, 2006

Khazret Sapenov, Reuven Cohen (Enomaly) > <http://www.enomaly.net>1 **Abstract**

This tutorial explains how to set up and run Windows on the Amazon Elastic Compute Cloud Beta with the help of Qemu and Linux. This HOWTO can be adapted to install any guest operating systems supported by QEMU. We have chosen Windows Server 2003 because they offer a free evaluation copy, but QEMU can be used for emulating other operating systems - including, Solaris, *BSD, Unix, etc.

2 Prerequisites and required components

- You need to have an account with [Amazon Elastic Compute Cloud](http://aws.amazon.com/ec2) (<http://aws.amazon.com/ec2>).
- Basic Linux/Unix administration skills
- Linux/Unix distribution (Fedora Core 6, Ubuntu, Debian, *BSD) as a base for your experiments (You can also do it in Windows, using VMware appliances).
- Processor emulator [QEMU](http://fabrice.bellard.free.fr/qemu/) (<http://fabrice.bellard.free.fr/qemu/>) for x86 (currently Amazon doesn't support 64-bit version of QEMU).
- Windows Installation in ISO format
- Batteries not included, Some assembly required. :)

This HOWTO can be adapted to install any guest operating systems supported by QEMU. We have chosen Windows Server 2003 because they offer a free evaluation copy, but QEMU can be used for emulating other operating systems - including, Solaris, *BSD, Unix, etc.

["Installation/QemuEmulator".?](#) In many ways, Windows is not the best choice, as it is slower than either Windows 9x or Windows 2000 under emulation. But for many of us, it will be what we have lying around. There is nothing in this HOWTO that is Windows specific. There may well be specific issues for non-Windows versions operating systems.

If your using Windows desktop, you may want to view the [Qemu in Windows tutorial](#).

3 Licensing

Your Windows license only allows one install of Windows at a time. Installing Windows on a virtual machine while simultaneously having the same copy on another partition or machine will break that license.

* [Windows Server 2003 R2 Trial Software Download](http://www.microsoft.com/technet/downloads/winsrvr/default.msp) (<http://www.microsoft.com/technet/downloads/winsrvr/default.msp>)

4 Setting Up Local Environment

4.1 Fedora Core 6

We have chosen Fedora Core 6, but this tutorial should help you in Fedora Core 5 and 4 as well.

4.1.1 installing qemu

The easiest way to install qemu was:

```
yum install qemu
```

Though you can compile it from source as well, it'll definitely give more options. Read instructions how to do it at qemu site.

4.1.2 creating windows image

To install Windows you will need to create a virtual drive to store it on:

```
qemu-img create -f qcow winxp.img 4G
```

This creates a 4 gigabyte virtual drive called winxp.img in the location where you ran the command. Now let's install Windows:

```
qemu -hda winxp.img -cdrom yourDistro.iso -m 256 -boot d
```

This tells QEMU to use the virtual disk as the harddisk (`-hda winxp.img`), which drive to use as the CDROM (in this case it's your image file) and to allocate 256Mb of memory for the virtual pc (`-m 256`). Once QEMU boots up, you will be in the Windows install. This is identical to any other installation of Windows. Follow the standard Windows installation and after a few reboots of QEMU and about 40 minutes, you will have a copy of Windows installed on your virtual drive.

You will likely need to interact with the install. Just click on the QEMU window and interact away. When you need your mouse, press `CTRL+ALT` and your mouse is free. You can also toggle fullscreen with `ALT+CTRL+F`. For full list of options just run `qemu` without any options in terminal.

When you have finished playing with the installation, you will eventually want to shut it down - which you do via the start menu as for any other Windows installation. To "turn off your computer", just close the QEMU window. To test your newly installed Windows, run `qemu` again:

```
qemu -hda winxp.img -m 256 -boot c
```

This should load usual Windows startup screen.

Note:

[Important ! At this step you need to enable remote access to this machine, to be able to administer it once it's at Amazon.](#)

4.1.3 Bundling and Uploading our AMI to Amazon EC2

The Amazon EC2 AMI Tools are command-line utilities to help bundle an Amazon Machine Image (AMI), create an AMI from an existing machine or installed volume, and upload a bundled AMI to Amazon S3. Now we are ready to bundle our system and upload the snapshot to Amazon Elastic Cloud. To do this, we will need auxiliary tools (I assume you have sufficient privileges to install programs in your system):

```
yum install ruby
```

```
wget http://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.noarch.rpm  
  
rpm -Uvh ec2-ami-tools.noarch.rpm
```

At this point, the machine image should be ready for bundling:

```
ec2-bundle-vol -d /mnt -k ~root/pk-HKZYKTAIG2ECMXIYIBH3HXV4ZBZQ55CLO.pem -u 495219933132 -s 1536
```

Note: all keys should contain your personal private key and X.509 certificate.

This command may take several minutes to complete. Once the bundling process has been completed, the AMI, its associated manifestfile and image parts can be found in the /mnt directory. Confirm this with the following command:

```
ls -l /mnt/image.*
```

Uploading the AMI to Amazon S3.

All AMIs are loaded from Amazon S3 storage. The newly bundled AMI needs to be uploaded to an existing account on Amazon S3 such as the one you created in the section called [œSigning up for Amazon S3](#).

S3 stores data objects in buckets, which are similar in concept to directories. You'll need to specify a bucket name in the command below as . Buckets have globally unique names and are owned by unique users. If you have used S3 before, you can use any of your existing buckets or just give *ec2-upload-bundle* any name that makes sense to you. The *ec2-upload-bundle* utility will upload the bundled AMI to a specified bucket. If the specified bucket does not exist it will create it. If the specified bucket belongs to another user *ec2-upload-bundle* will fail, and you will have to try a different name.

For this step you'll need your AWS Access Key ID () and AWS Secret Access Key (). The steps to find these were covered in the section called [œSigning up for Amazon S3](#).

The upload process can be quite lengthy, but you will get continuous feedback until the upload has completed:

```
# ec2-upload-bundle -b -m /mnt/image.manifest -a -s
```

```
Encrypting bundle manifest...
Completed encryption.
Uploading encrypted manifest...
Uploaded encrypted manifest to https://s3.amazonaws.com//image.manifest.
Uploading bundled AMI parts to https://s3.amazonaws.com//image...
Uploaded 00 to https://s3.amazonaws.com//00.
Uploaded 01 to https://s3.amazonaws.com//01.
Uploaded 02 to https://s3.amazonaws.com//02.
Uploaded 03 to https://s3.amazonaws.com//03.
...
Uploaded 23 to https://s3.amazonaws.com//23.
Uploaded 24 to https://s3.amazonaws.com//24.
Upload Bundle complete.
```

5 Setting Up Amazon Environment

Amazon EC2 Command-Line Tools The command-line tools serve as the client interface to the Amazon EC2 web service. Use these tools to register and launch instances, manipulate security groups, and more.

Registering the AMI

Your image must be registered with Amazon EC2, so we can locate it and run instances based on it. Execute the following command to register the AMI you've just uploaded to Amazon S3:

```
PROMPT> ec2-register /image.manifest
```

```
IMAGE ami-61a54008
```

As output you will get an AMI identifier, the value next to the `IMAGE` tag (`ami-61a54008` in the example) that can be used to run instances. Running

Instances

You can now run an instance of the modified AMI by specifying the image identifier you received when you registered the image. Execute the following command, substituting the image identifier with that received in the section called «Registering the AMI».

```
PROMPT> ec2-run-instances ami-61a54008

INSTANCE i-10a64379 ami-61a54008 pending
```

This will start a single instance based on your newly created AMI and provide you with an instance identifier, the value immediately to the right of the INSTANCE tag, that can be used to monitor the status of the running instance, as you did in the section called «Running an Instance», to confirm the instance is up and running.

You now have your very own image starting up and getting ready. You can monitor its status until it's ready and then connect to it with your web browser to confirm the changes you had made are actually live. Congratulations!

You have successfully built and deployed your very own AMI, and run instances based on it. This custom AMI is private to your account. You can build as many custom AMIs as you need and use them to launch as many instances as you need.

Authorizing Access to Your Instances

Before you can reach your instance over the internet, you have to authorize traffic to port that is responsible for remote administration in our case it is port 3389:

```
ec2-authorize default -p 3389
```

6 Starting Windows at Amazon

Log in to your instance and setup system to startup windows automatically. I did it this way:

```
nano /root/.bashrc
```

add alias to save time:

```
alias win2003='nohup qemu -hda /dist/win2003srv.img -m 500 -boot c -redir tcp:3389::3389 -nographic &'
```

Note: you might have different choice for your hda disk.

Last thing is to actually start it, log off and log in back. Type our alias at last:

```
#win2003
```

Now it should be started automatically and stays running even if close your session.

7 Connecting to your windows

If you are using Windows XP go to '*Start > All programs > Accessories > Communication > Remote Desktop Connection*'. Use access info that you set up during installation of windows.

At this point you should be playing inside windows.

8 Conclusion

Some of the benefits of QEMU running within an AMI;

- Able to track usage variables. This would be ideal for web hosting and other various utility models.
- Significantly lower per/hour costs by running several additional Virtual Machines per AMI. This would require a Qemu abstraction / accelerator.

You could foresee up to 15+ addition concurrent operation systems.

- Able to more effectively simulate varied operating/hardware environments for easy migration from legacy systems.
- Backups to S3 - Using the Qemu-img tool, instant snapshots can be saved to S3. Used with a versioning system, this could provide for unlimited roll backs.

- VMware compatible - Qemu-img supports vmdk conversion as well as several other formats enabling easy migration from existing virtualized server environments.

One of the main hurdles that needs to be overcome is in regards to the performance and overall speed of the emulated systems. To overcome this, a Qemu abstraction / accelerator will need to be developed. This could be developed by us or any ambitious AMI users.

Another option for getting windows running, might be using a combo of para-virtualization (Xen) with a system abstraction layer tailored for optimal Windows performance. I should note, the new version of Xen supports Windows, but only with specific BIOS & CPUs such as Intel VT.

A separate para-virtualized abstraction layer would allow you to compile an AMI image rather than a Qemu image. This would need to be developed by someone with internal access to EC2 and would require very new hardware.

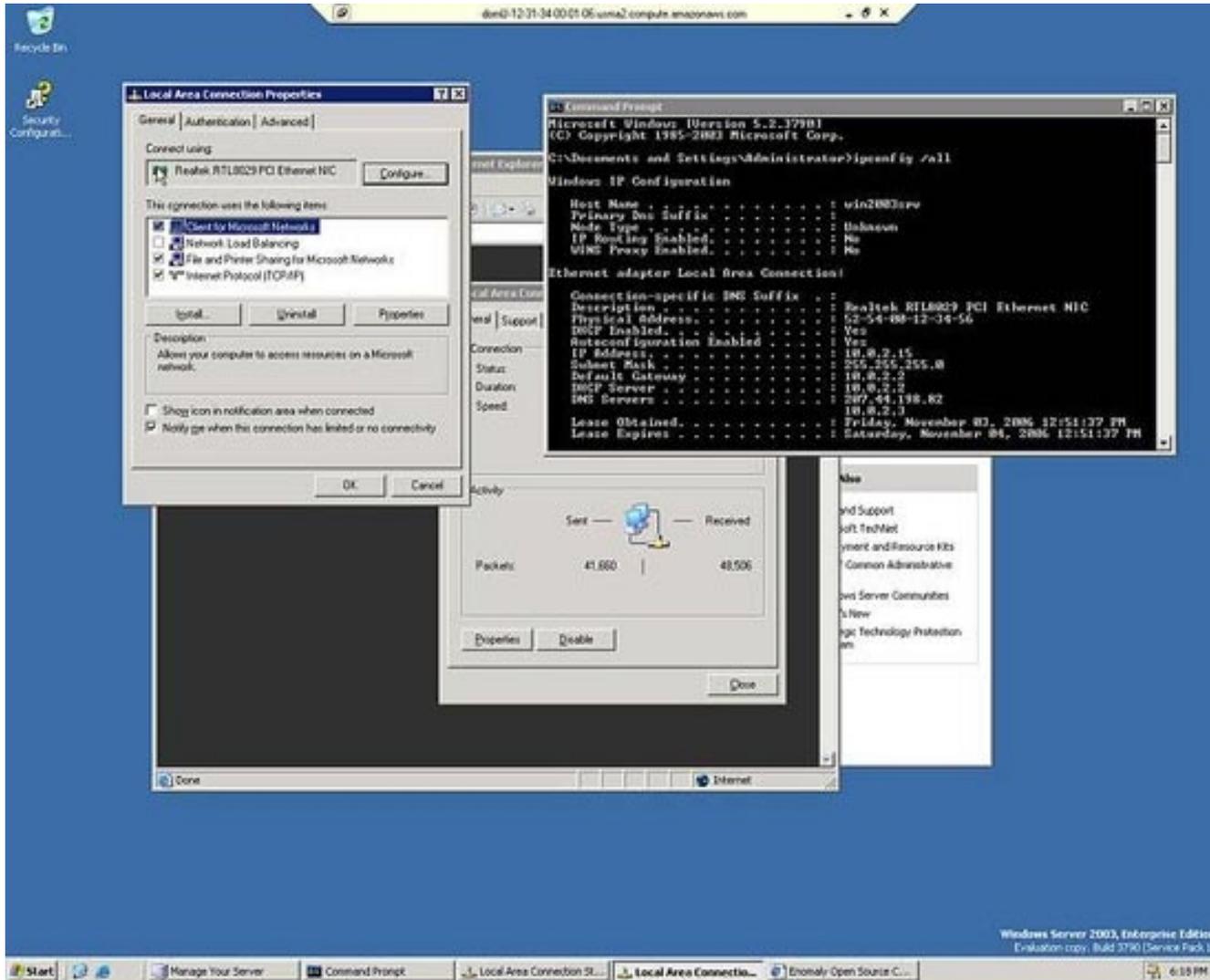
9 Other Cool Stuff

Forget system restore and backup under XP. QEMU uses a concept called 'overlay images'. Once you have done your initial install, you can create an overlay image and run QEMU off this. The overlay image is very much smaller than the original image and will only contain changes made since the original install. If you trash your XP installation, you can simply delete the overlay image and create a new one from the original image. This is an ideal use in conjunction to Amazon's S3 storage services.

To create an overlay image, use the qemu-image program which is installed with QEMU. To create an image winxp.ovl from winxp.img, issue the following:

```
qemu-img create -b winxp.img -f qcow winxp.ovl
```

10 ScreenShot



11 Resources

- [QEMU Homepage](http://fabrice.bellard.free.fr/qemu/) (<http://fabrice.bellard.free.fr/qemu/>)

- [Ubuntu emulated in Windows with QEMU](http://homepage.sunrise.ch/mysunrise/ekeller00/EricKellerUbuntuPage.html) (<http://homepage.sunrise.ch/mysunrise/ekeller00/EricKellerUbuntuPage.html>)
- [Forum thread about the script used in this article](http://www.ubuntuforums.org/showthread.php?t=187413) (<http://www.ubuntuforums.org/showthread.php?t=187413>)
- [Original Tutorial](https://help.ubuntu.com/community/WindowsXPUnderQemuHowTo) (<https://help.ubuntu.com/community/WindowsXPUnderQemuHowTo>)
- [Qemu in Windows tutorial](#)

12 About authors

- Reuven Cohen - Chief Technologist
- Khaz Sapenov - Chief Architect