

- Partners
- Support
- Community
- Ubuntu.com

- Login to edit

 

# UEC/Images

## Intro

The term 'Ubuntu Cloud Guest' refers to the Official Ubuntu images that are available at <http://cloud-images.ubuntu.com>. These images are built by Canonical. They are then registered on EC2, and compressed tarfiles are made also available for download. For using those images on a public cloud such as Amazon EC2, you simply choose an image and launch it. To use those images on a private cloud, or to run the image on a local hypervisor (such as KVM) you would need to download those images and either publish them to your private cloud, or launch them directly on a hypervisor. The following sections explain in more details how to perform each of those actions

## Ubuntu Cloud Guest on Amazon EC2 Cloud

In order to start using official Ubuntu images on the public cloud (ex. Amazon EC2), you

- Choose an Ubuntu release you want to use (Lucid, Maverick, Natty...etc)
- Get the latest AMI ID corresponding to the release you want to use
- Launch your chosen image on EC2

### Choosing an Ubuntu Release

Choosing the Ubuntu release depends on a lot of factors such as whether or not you would like to launch an LTS image for long term support, or whether you need specific versions of libraries or applications. If in doubt, choose the latest LTS release (currently Precise 12.04)

### Get Ubuntu AMI-ID

As Ubuntu Cloud Guest images are uploaded and registered on the Amazon EC2 cloud, they are referred to as AMI (Amazon Machine Images). Each AMI is a machine template from which you can instantiate new servers. Each AMI has its own unique ID. In order to launch an instance on the EC2 cloud, you first need to locate its ID. The AMI Locator web application can help you quickly locate an AMI ID. Here's how to use it

- Visit AMI Locator
- Type a few characters in the search box signifying what you're searching for. For example if you would like to find out the AMI-ID for the latest release of Maverick Meerkat to run on a 64 bit ebs instance in the us-east region, you would search for "mav 64 us-east ebs" or a subset thereof.

### AMI Locator

Show  entries

Search:

Zone	Name	Arch	EBS	Release	AMI-ID	AKI-ID
us-east-1	maverick	amd64	ebs	20101225	ami-cef405a7	aki-427d952b

Any Any Any Any Any Any

### Launch Ubuntu AMI

- For a video tutorial on starting your first Ubuntu Cloud Guest on the Amazon EC2 cloud using the web management console (recommended for newcomers), check out this video tutorial <http://www.youtube.com/user/ubuntucloud#p/u/4/rYJLIIfVuSMY>
- For using the ec2 command line tools (recommended for advanced users and script writers), check out the EC2 Starters Guide

## Ubuntu Cloud Guest on UEC Private Cloud

If you are running your own private UEC cloud, you may wish to publish new Ubuntu Cloud Guest images (as they are made available) to your cloud infrastructure.

### Contents

1. Intro
2. Ubuntu Cloud Guest on Amazon EC2 Cloud
  1. Choosing an Ubuntu Release
  2. Get Ubuntu AMI-ID
  3. Launch Ubuntu AMI
3. Ubuntu Cloud Guest on UEC Private Cloud
  1. Downloading Daily and Released Images
4. Ubuntu Cloud Guest images on Local Hypervisor (Maverick)
5. Ubuntu Cloud Guest images on Local Hypervisor 11.04 (Natty) or 11.10 (Oneiric)
6. Ubuntu Cloud Guest images on 12.04 LTS (Precise) and beyond using NoCloud
7. ARM Images
  1. Conversion to RAW Disk formats
  2. QEMU invocation
  3. Modify the image
  4. Booting in QEMU
8. Customizing Ubuntu Cloud Guest images Boot Sequence with Cloud-Init
9. Building Your Own Ubuntu Cloud Guest images
10. Machine Consumable Ubuntu Cloud Guest images Availability Data

## Downloading Daily and Released Images

Ubuntu Cloud Guest images are created in two flavors. "Daily" is a daily composed image that provides a fresh but perhaps less tested image. and "Release" that provides regularly updated and well tested images

"Daily" builds are made available for testing and continuous integration. The daily builds are not tested at all. However, the content of the images has already been through the SRU process, so in general the daily builds for stable releases are stable. Quality of daily builds for the development release will vary.

Daily builds are available under <http://cloud-images.ubuntu.com/server/<codename>> .

- 10.04 LTS Lucid <http://cloud-images.ubuntu.com/server/lucid/>
- 10.10 Maverick <http://cloud-images.ubuntu.com/server/maverick/>
- 11.04 Natty <http://cloud-images.ubuntu.com/server/natty/>
- 11.10 Oneiric <http://cloud-images.ubuntu.com/server/oneiric/>

Released builds are available under <http://cloud-images.ubuntu.com/server/releases/<codename>> .

- 10.04 LTS Lucid <http://cloud-images.ubuntu.com/server/releases/lucid/>
- 10.10 Maverick <http://cloud-images.ubuntu.com/server/releases/maverick/>
- 11.04 Natty <http://cloud-images.ubuntu.com/server/releases/natty/>
- 11.10 Oneiric <http://cloud-images.ubuntu.com/server/releases/oneiric/>

Once you have downloaded a suitable image, you most likely want to publish it to your private cloud, a process called Bundling Images

## Ubuntu Cloud Guest images on Local Hypervisor (Maverick)

Images after 10.10 (maverick) can now be run outside of the cloud environment without any difficult setup, and even provide an easy way to provide metadata to the instance! If you would like to run a Ubuntu Cloud Guest image on your local KVM hypervisor, checkout the previous section explaining Downloading Daily and Released Images then untar the image and boot it. Here is an example to make things clearer

- The Image file name will vary. For this example, it is "maverick-server-uec-amd64.tar.gz".

```
$ tarball=maverick-server-uec-amd64.tar.gz
```

- Extract the tar file and set some variables names for its contents. Here is an example

```
$ contents=${tarball}.contents
$ tar -Sxvzf ${tarball} | tee "${contents}"
maverick-server-uec-amd64.img
maverick-server-uec-amd64-vmlinuz-virtual
maverick-server-uec-amd64-loader
maverick-server-uec-amd64-floppy
README.files
$ base=$(sed -n 's/.img$//p' "${contents}")
$ kernel=$(echo ${base}-vmlinuz-*)
$ floppy=${base}-floppy
$ img=${base}.img
```

- Create a copy on write disk image, based on the disk image we downloaded and boot that image in local kvm

```
# make a qcow image just to keep the original pristine
$ qemu-img create -f qcow2 -b ${img} disk.img
$ kvm -fda ${floppy} -drive if=virtio,file=disk.img -boot a
```

- Get the password for 'ubuntu' and log in

The default selection in the grub menu will result in a randomly generated password for the 'ubuntu' user on first boot. The password is written two places, the console and the serial device. On the console, you will see lines like:

```
[ 1.820549] EXT4-fs (sda): re-mounted. Opts: (null)
::unccloud-init: setting ubuntu pass = qLnYZTSy
::unccloud-init: changing ubuntu user's password!
::unccloud-init: enabled console on ttyS0
```

The password is also written to the kvm serial console. You can see the kvm serial console by hitting **ctrl-alt-3**. On first boot, you will see lines like

```
===
ubuntu_pass = PhAKp1kx
===
```

- Want to get that first boot again? Easy, just delete the cow disk, create a new one and boot it

```
$ rm -rf disk.img && qemu-img create -f qcow2 -b ${img} disk.img
$ kvm -fda ${floppy} -drive if=virtio,file=disk.img -boot a
```

You can customize the way Ubuntu Cloud Guest images boot under a local KVM instance by passing certain kernel command line variables. This approach

enables you to pass custom cloud-init data-sources, set the instance's password, hostname, instance-id ...etc. If that interests you, you can read more about passing KVM Kernel Options

## Ubuntu Cloud Guest images on Local Hypervisor 11.04 (Natty) or 11.10 (Oneiric)

Booting from local disk is greatly simplified compared to the above in Natty (11.04) and going forward. The process is documented in cloud-init's source.

## Ubuntu Cloud Guest images on 12.04 LTS (Precise) and beyond using NoCloud

In 12.04 and later, you can use the "NoCloud" datasource. The cloud-utils package included in 12.10 and later contains a tool **cloud-localds** that helps in creating this data.

There is a blog post and a gist that describes how to use this. It is copied below.

```
## Install a necessary packages
$ sudo apt-get install kvm cloud-utils genisoimage

## URL to most recent cloud image of 12.04
$ img_url="http://cloud-images.ubuntu.com/server/releases/12.04/release"
$ img_url="${img_url}/ubuntu-12.04-server-cloudimg-amd64-disk1.img"
## download the image
$ wget $img_url -O disk.img.dist

## Create a file with some user-data in it
$ cat > my-user-data <<EOF
#cloud-config
password: passw0rd
chpasswd: { expire: False }
ssh_pwauth: True
EOF

## Convert the compressed qcow file downloaded to a uncompressed qcow2
$ qemu-img convert -O qcow2 disk.img.dist disk.img.orig

## create the disk with NoCloud data on it.
$ cloud-localds my-seed.img my-user-data

## Create a delta disk to keep our .orig file pristine
$ qemu-img create -f qcow2 -b disk.img.orig disk.img

## Boot a kvm
$ kvm -net nic -net user -hda disk.img -hdb my-seed.img -m 512
```

## ARM Images

Starting with Oneiric, two different ARM images are being built. Both images have the OMAP kernel installed and are provided in QCOW2 format. The root file-system on both images are identical.

How to identify the images

```
oneiric-server-cloudimg-armel-disk1.img: non-bootable
oneiric-server-cloudimg-armel_omap-disk1.img: bootable
```

The non-bootable image contains a single partitioned file system.

The bootable image contains two partitions, with the first being a OMAP3/4 bootloader and the second containing the root file system.

### Conversion to RAW Disk formats

To use/mount these images you will need to convert the image into a raw format. To do so:

```
qemu-img convert -O raw <DISK>.img <DISK>.raw
```

### QEMU invocation

The initial releases of the bootable QCOW2 images will need modification in order to be usable within QEMU, this is due to the installation of Cloud-init. Unlike traditional x86 hardware, Linux on ARM is booted with u-boot. If these images are booted with a meta-data service compatible with Cloud-init, they will work like a normal cloud image.

### Modify the image

- Install uboot-tools and qemu-kvm packages
- Convert the image to raw format

```
qemu-img convert -O raw <DISK>.img <DISK>.raw
```
- Add the raw disk image via kpartx:

```
kpartx -a -v <DISK>.raw
```

Output will look something like:

```
add map loop0p1 (254:0): 0 144584 linear /dev/loop0 1
add map loop0p2 (254:1): 0 4192965 linear /dev/loop0 144585
```

In this example, loop0p1 is the boot partition for the image, and the device mapper path is /dev/mapper/loop0p1.

- Mount the first partition:

```
mount /dev/mapper/loop0p1 /mnt
```

- Create a new boot.cmd file in /tmp. Populate it with the following

```
setenv dvimode 1280x720MR-16@60
setenv vram 12MB
setenv bootcmd 'fatload mmc 0:1 0x80300000 uImage; fatload mmc 0:1 0x81600000 uInitrd; bootm 0x80300000 0x81600000'
setenv bootargs console=ttyO2 omapfb.mode=dvi init=/usr/lib/cloud-init/uncloud-init root=LABEL=cloudimg-rootfs rw fixrtc ubuntu
boot
```

- Compile the boot.cmd to boot.scr

```
mkimage -A arm -O linux -T script -C none -a 0 -e 0 -n "Ubuntu" -d /tmp/boot.cmd /mnt/boot.scr
```

- Umount /mnt:

```
umount /mnt
```

- Remove the loop mapping:

```
kpartx -d /dev/loop0; losetup -d /dev/loop0
```

- Boot or burn the image

## Booting in QEMU

To boot these images into QEMU, run: `qemu-system-arm -M beaglexm -drive file=<DISK>.img,if=sd,cache=writeback -nographic`

## Customizing Ubuntu Cloud Guest images Boot Sequence with Cloud-Init

cloud-init is the Ubuntu package that handles early initialization of a cloud instance. It is installed in the Ubuntu Cloud Guest images and also in the official Ubuntu images available on EC2. Images contain CloudInit as well as uncloud-init

uncloud-init, is a behind the scenes tool that helps configure Ubuntu Cloud Guest images to run in a local hypervisor environment. uncloud-init helps set a password for user "ubuntu", enables password authentication in sshd configuration and spawns a login tty on the serial console

## Building Your Own Ubuntu Cloud Guest images

A reliable build process in place. Current images are built on a 10.04 LTS host from a 0.11 branch of vmbuilder. The build scripts are available under the automated-ec2-builds, and published to cloud-images and EC2 using ec2-publishing-scripts.

Read more about creating your own image

## Machine Consumable Ubuntu Cloud Guest images Availability Data

In order to provide information about what builds are available for download or running on ec2, a 'query' interface is exposed at <http://cloud-images.ubuntu.com/query> . This will allow users of the service to download images or find out the latest ec2 AMIs programmatically.

The data is laid out as follows:

- There are 2 files in top level director 'daily.latest.txt' and 'released.latest.txt'. Each of these files contains tab delimited data, with 4 fields per record. daily.latest.txt has information about the daily builds, released.latest.tt about released builds:

```
<suite> <build_name> <label>      <serial>
hardy   server       release    20100128
```

- For each record in the top level files another set of files will exist: \

File	Description
<suite>/<build_name>/daily-dl.txt	downloadable images data for all daily builds currently available
<suite>/<build_name>/daily-dl.current.txt	downloadable images data for the most recent daily build

<suite>/<build\_name>/released-dl.txt      downloadable images data for all released builds

<suite>/<build\_name>/released-dl.current.txt      downloadable images data for the most recent released build

<suite>/<build\_name>/daily.txt      registered ec2 AMI data for all daily builds ever

<suite>/<build\_name>/daily.current.txt      registered ec2 AMI data for the most recent daily build

<suite>/<build\_name>/released.txt      registered ec2 AMI data for all released builds

<suite>/<build\_name>/released.current.txt      registered ec2 AMI data for the most recent released build

- The downloadable image data files contain 7 tab delimited fields:

```
<suite> <build_name> <label> <serial> <arch> <download_path> <suggested_name>
maverick server      daily   20100826 i386   server/maverick/20100826/maverick-server-uec-i386.tar.gz  ubuntu-maverick-daily-i
```

Download path is relative to "http://cloud-images.ubuntu.com/". Suggested name is the basename that this build would be published as in EC2.

- The registered ec2 AMI data contain 10 tab delimited fields:

```
<suite> <build_name> <label> <serial> <root-store> <arch> <region> <ami> <aki> <ari>
maverick server      alpha3 20100803.2 instance-store i386   us-west-1 ami-ebdc8dae aki-99a0f1dc <ari>
```

For many images, there may be no ari registered, so the field may be blank.

UEC/Images (last edited 2013-05-02 01:01:51 by smoser @ d14-69-66-169.try.wideopenwest.com[69.14.169.66]:smoser)