Fedora Draft
Documentation
Cloud Guide

Fedora in the cloud.

Jared Smith
Eric Christensen
Kaleb Keithley
Robyn Bergeron
Fedora Draft Documentation Cloud Guide
Fedora in the cloud.
Edition 16.4

Author  Jared Smith  jsmith@fedoraproject.org
Author  Eric Christensen  sparks@fedoraproject.org
Author  Kaleb Keithley  kkeithle@redhat.com
Author  Robyn Bergeron  rbergero@fedoraproject.org

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license (“CC-BY-SA”). An explanation of CC-BY-SA is available at http://creativecommons.org/licenses/by-sa/3.0/. The original authors of this document, and Red Hat, designate the Fedora Project as the “Attribution Party” for purposes of CC-BY-SA. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

For guidelines on the permitted uses of the Fedora trademarks, refer to https://fedoraproject.org/wiki/Legal:Trademark_guidelines.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

All other trademarks are the property of their respective owners.

The Cloud Guide contains information on building and managing your own cloud using free and open source solutions.
Preface

1. Document Conventions
This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the Liberation Fonts set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions
Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

Mono-spaced Bold
Used to highlight system input, including shell commands, file names and paths. Also used to highlight keycaps and key combinations. For example:

To see the contents of the file my_next_bestselling_novel in your current working directory, enter the cat my_next_bestselling_novel command at the shell prompt and press Enter to execute the command.

The above includes a file name, a shell command and a keycap, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from keycaps by the hyphen connecting each part of a key combination. For example:

Press Enter to execute the command.
Press Ctrl+Alt+F2 to switch to the first virtual terminal. Press Ctrl+Alt+F1 to return to your X-Windows session.

The first paragraph highlights the particular keycap to press. The second highlights two key combinations (each a set of three keycaps with each set pressed simultaneously).

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in mono-spaced bold. For example:

File-related classes include filesystem for file systems, file for files, and dir for directories. Each class has its own associated set of permissions.

Proportional Bold
This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose System Preferences Mouse from the main menu bar to launch Mouse Preferences. In the Buttons tab, click the Left-handed mouse check box

1 https://fedorahosted.org/liberation-fonts/
and click Close to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a gedit file, choose Applications\Accessories Character Map from the main menu bar. Next, choose Search\Find... from the Character Map menu bar, type the name of the character in the Search field and click Next. The character you sought will be highlighted in the Character Table. Double-click this highlighted character to place it in the Text to copy field and then click the Copy button. Now switch back to your document and choose Edit\Paste from the gedit menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

**Mono-spaced Bold Italic or Proportional Bold Italic**

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type ssh username@domain.name at a shell prompt. If the remote machine is example.com and your username on that machine is john, type ssh john@example.com.

The mount -o remount file-system command remounts the named file system. For example, to remount the /home file system, the command is mount -o remount /home.

To see the version of a currently installed package, use the rpm -q package command. It will return a result as follows: package-version-release.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a DocBook publishing system.

### 1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```none
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```java
package org.jboss.book.jca.ex1;
import javax.naming.InitialContext;
public class ExClient
```
{  
    public static void main(String args[])  
        throws Exception  
    {  
        InitialContext iniCtx = new InitialContext();  
        Object ref = iniCtx.lookup("EchoBean");  
        EchoHome home = (EchoHome) ref;  
        Echo echo = home.create();  
        System.out.println("Created Echo");  
        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));  
    }  
}  

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

![Note]

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

![Important]

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

![Warning]

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: http://bugzilla.redhat.com/bugzilla/ against the product Fedora.

When submitting a bug report, be sure to mention the manual's identifier: cloud-guide

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.
Cloud Concepts

1.1. An Introduction into Cloud Services using Fedora

1.1.1. What is Cloud Computing
Pardon the pun, but "the cloud" is a nebulous term. When well-meaning people talk about cloud computing, they are often talking about one of many difference scenarios. In this section, we'll explore using Fedora and various aspects of cloud computing.

1.1.2. Types of Cloud Services
There are various types of cloud services:

1.1.3. Infrastructure as a Service

1.1.4. Platform as a Service

1.1.5. Software as a Service
Using Fedora in the Cloud

2.1. Amazon EC2

2.1.1. Overview of EC2
Amazon Web Services (commonly referred to as AWS) is a public cloud, a collection of computing services that allows one to build and run software services in Amazon's data centers. Fedora publishes system images for AWS's virtual machine platform, Amazon Elastic Compute Cloud (or EC2), which allows one to create virtual machines in the Amazon cloud with very little effort. The objective of this section is to familiarize the reader with EC2's terminology and functionality. For more detailed documentation, see the AWS website².

2.1.2. EC2 concepts
What follows are some short explanations of EC2 terminology. For more detailed information, see the EC2 documentation².

2.1.2.1. Images and Instances
A machine image is a snapshot of a system (specifically its root filesystem) that provides the basis for a virtual machine in EC2. When you run a new virtual machine in EC2 you choose a machine image to use as a template. The new virtual machine is then an instance of that machine image that contains its own copy of everything in the image. The instance keeps running until you stop or terminate it, or until it fails. If an instance fails, you can launch a new one from the same image. You can create multiple instances of a single machine image. Each instance will be independent of the others.

You can use a single image or multiple images, depending on your needs. From a single image, you can launch different types of instances. An instance type defines what hardware the instance has, including the amount of memory, disk space, and CPU power.

Amazon, Fedora, other groups, and individuals publish images for public use. You might only need to use images that reputable sources provide, and you can simply customize the resulting instances to suit your needs as you launch them. You can also create your own machine images, but that is beyond the scope of this document.

Machine images in EC2 are sometimes referred to as AMIs.

Machine images have identifiers that begin with ami, such as ami-6ebe4507. Instances have identifiers that begin with the letter i, such as i-12459dbd.

2.1.2.2. Regions and Availability Zones
Amazon hosts datacenters many parts of the world. Those from a particular part of the world make up a region. Regions' names are based on their locations, such as in us-east-1.

Regions are broken up into availability zones, which are designed to isolate failures from one another but still provide faster communication than communication between regions. Distributing a web application amongst several availability zones can help improve its reliability if an availability zone

---

¹ http://aws.amazon.com/
² http://aws.amazon.com/documentation/ec2/
encounters problems. Availability zones' names are based on the regions in which they reside, such as **us-east-1a**.

### 2.1.2.3. Storage

EC2 instances use one or more of three types of storage provided by AWS:

#### 2.1.2.3.1. Simple Storage Service (S3)

Amazon Simple Storage Service (S3) is a web service-based storage system that is accessible inside EC2 and elsewhere on the Internet. As this document will not focus on S3, see the [Amazon S3 documentation](http://aws.amazon.com/s3) for more details.

#### 2.1.2.3.2. Elastic Block Store (EBS)

Amazon Elastic Block Store (EBS) provides instances with persistent, disk-like storage that you can attach to and detach from instances, similar to portable disk drives. By creating EBS **volumes** and attaching them to instances you can store data that you wish to be portable to more than one instance in the event an instance fails or is replaced. Since instances' root filesystem tend to have limited space, volumes also provide a simple way of adding additional disk capacity to instances.

Volumes have identifiers that begin with **vol**, such as **vol-ffe93704**.

You can create a backup **snapshot** of a volume. From the snapshot you can then create a new volume and attach it to another instance. You can create multiple volumes from the same snapshot. Each volume will be independent of the others.

Snapshots have identifiers that begin with **snap**, such as **snap-773491a0**.

#### 2.1.2.3.3. Instance Storage

Some instance types have **instance storage**, scratch space that persists only as long as an instance runs. Instance storage is destroyed when an instance stops, terminates, or fails. For this reason, it is also referred to as ephemeral storage.

When EC2 was first introduced, all machine images were **backed by instance storage**, meaning that their instances' root filesystems were stored in instance storage. Machine images can now also be **backed by EBS**, meaning that their instances' root filesystem instead reside on EBS volumes.

### 2.1.2.4. Security Groups

A **security group** defines firewall rules for your EC2 instances. These rules specify which incoming network traffic should be delivered to an instance (e.g., accept web traffic on port **80** or SSH traffic on port **22**). All other traffic is ignored. By default, a security group has no rules, which causes it to block all incoming traffic. You can modify the rules for a security group at any time.

Every instance runs inside of a security group. You can create your own security groups, or you can use the **default** security group that EC2 provides for you. When you run a new instance it will run in the **default** security group unless you choose a different one.

### 2.1.3. Signing up for an EC2 account

To use AWS you need to create an online account. You can do this by following the steps below:

---

3 [http://aws.amazon.com/s3](http://aws.amazon.com/s3)
Amazon AWS is not free

AWS is designed as a pay-as-you-go online service. Much of EC2 is free for new users; the rest is available for per-hour or per-month fees that are detailed on the EC2 Website. As such, Amazon requests a credit card number to keep on file with your new account.

1. **Launch Web Browser**
   Launch your web browser and navigate to the Amazon EC2 website. The Amazon EC2 website is located at [http://aws.amazon.com/ec2/](http://aws.amazon.com/ec2/). Click the **Sign Up Now** button to commence registration.

2. **Sign In or Create an AWS Account**
   The first screen of the registration process allows existing users to login by providing their email address and password. New users only need provide their email address to commence registration. To proceed:
   - Enter your email address in the **My e-mail address is:** field. This will be your Amazon EC2 username;
   - Ensure that the **I am a new user.** radio button is selected.

   Once these values have been provided click **Sign in using our secure server** to proceed with registration.

3. **Login Credentials**
   The **Login Credentials** screen requests entry of the basic authentication tokens that will be associated with your Amazon EC2 account. The basic authentication tokens are your email address, and a password. The email address is effectively the username for the account.
   a. Enter your full name in the **My name is:** field.
   b. Enter your email address in the **My e-mail address is:** field.
   c. Enter your email address again in the **Type it again:** field, located immediately after the **My e-mail address is:** field, to confirm.
   d. Enter your desired password in the **Enter a new password:** field.
   e. Enter your desired password again in the **Type it again:** field, located immediately after the **Enter a new password:** field.

   Click **Continue** to proceed to the next step in the registration process.

---

4 [http://aws.amazon.com/ec2/#pricing](http://aws.amazon.com/ec2/#pricing)
Chapter 2. Using Fedora in the Cloud

Important — Password Security

It is important that a secure password is used for accessing Amazon Web Services. Amazon recommend that you do not write down your password. Use a password that is at least 8 characters long. It should contain a mix of uppercase, lowercase, and numeric characters. It should also contain at least one special character.

4. Contact Information

The Contact Information screen requests entry name, address, and a phone number. Fields with a * next to them are required. Registration will not be allowed to proceed if they are not populated.

Enter the required information, then click the link to review the Amazon Web Services Customer Agreement. The Amazon Web Services Customer Agreement is also available by directly accessing http://aws.amazon.com/agreement/. It is necessary to review and indicate that you agree with the terms defined in the agreement prior to completing Amazon Web Services registration.

Check the box labeled Check here to indicate that you have read and agree to the terms of the Amazon Web Services Customer Agreement. Click the Create Account and Continue button to proceed with registration.

2.1.4. Connecting to EC2 using euca2ools

One can interact with EC2 through either a web-based management console or via euca2ools, a suite of command line tools designed for services like EC2. This tutorial will focus on using EC2 with euca2ools at the command line.

To using the command line tools you first need to obtain access keys for your account. You can find them by going to the AWS management console on the web, clicking your name on the top, followed by Security Credentials, and scrolling down to the section titled Access Credentials. Make note of the Access Key ID and the Secret Access Key that appears beside it. Both of them should be long sets of alphanumeric characters. Create a file called .iamrc in your home directory that contains those keys in this format:

```
AWSAccessKeyId=your_access_key_id
AWSSecretKey=your_secret_key
```

Since euca2ools is designed to work with all AWS-compatible clouds, not just AWS itself, it needs to know which cloud to contact. Create a file called .eucarc in your home directory with the following content to point it toward AWS:

```
AWS_CREDENTIAL_FILE=~/.iamrc
EC2_URL=https://ec2.amazonaws.com/
S3_URL=https://s3.amazonaws.com/
EUARE_URL=https://iam.amazonaws.com/
```

5 https://console.aws.amazon.com/ec2/
source "$AWS_CREDENTIAL_FILE"
export EC2_ACCESS_KEY=AWSAccessKeyId
export EC2_SECRET_KEY=AWSSecretKey
export AWS_ACCESS_KEY=AWSAccessKeyId
export AWS_SECRET_ACCESS_KEY=AWSSecretKey

Finally, add these settings to your shell's environment by running: $ source ~/.eucarc

2.1.4.1. Installing the euca2ools package
Installing the euca2ools package is simple. To do so with yum, run the following command as the root user:

# yum install euca2ools

2.1.4.2. Choosing your EC2 region
Next, choose an EC2 region to use. Things to consider when choosing a region include its geographic location, the pricing for instances in that region, and whether the image you wish to use is available in that region. You can get a list of regions by running the euca-describe-regions, which results in a list such as this:

<table>
<thead>
<tr>
<th>REGION</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>eu-west-1</td>
<td>ec2.eu-west-1.amazonaws.com</td>
</tr>
<tr>
<td>us-east-1</td>
<td>ec2.us-east-1.amazonaws.com</td>
</tr>
<tr>
<td>ap-northeast-1</td>
<td>ec2.ap-northeast-1.amazonaws.com</td>
</tr>
<tr>
<td>us-west-1</td>
<td>ec2.us-west-1.amazonaws.com</td>
</tr>
<tr>
<td>ap-southeast-1</td>
<td>ec2.ap-southeast-1.amazonaws.com</td>
</tr>
</tbody>
</table>

When you choose an EC2 region you can make euca2ools start using it by editing the line that contains EC2_URL in your .eucarc file. For example:

export EC2_URL=https://ec2.us-east-1.amazonaws.com/

After adjusting the region in the .eucarc file, reset the settings in your shell's environment by running this command: $ source ~/.eucarc

2.1.4.3. Creating a key pair
The primary way of logging into Fedora instances running in EC2 is via SSH. Since Fedora instances have no passwords, you need a SSH key pair to log in to them. The private half of this key pair is stored on your computer, while the public half is stored in EC2 so instances can download them as they start. This allows you to securely log into your instances without a password.

You can have multiple key pairs. Each key pair has its own name. Key pairs are specific to each EC2 region.

Choose a name for a new key pair and then use the euca-add-keypair command to create it and write the private key to a file. Be sure to choose a name that is easy to remember. To create your new key pair, run the following command, replacing mykey with a memorable name:

$ euca-add-keypair mykey > mykey.pem
Key pairs are irreplaceable

EC2 does not store the private halves of key pairs. The time you run `euca-add-keypair` is the only chance you will have to save a copy of the private key. There is no way to recover a lost private key from EC2.

You can use `euca-describe-keypairs` to display a list of your keypairs.

```
$ euca-describe-keypairs
```

You can use `euca-delete-keypairs` to delete a keypair. Deleting a keypair does not remove it from instances that are already running; it merely prevents new instances from using it.

```
$ euca-delete-keypair mykey1
```

### 2.1.4.4. Setting up a security group

Each security group has its own set of firewall rules. While this tutorial uses the default security group that EC2 provides for you, you can also create your own security groups.

The `euca-authorize` command lets you tell EC2 to allow traffic from ranges of IP addresses and ports into a security group. To allow access to SSH (TCP port 22) running on instances in the default security group, run the following command replacing 1.2.3.4 with the IP address from which you will connect to your EC2 instances.

```
$ euca-authorize default -p 22 -s 1.2.3.4/32
```

If you do not specify a range of IP addresses then the port(s) you choose will be open to the entire Internet. For example, the following command allows SSH access from any machine, not just your computer:

```
$ euca-authorize default -p 22
```

To allow pings and other ICMP traffic you can run:

```
$ euca-authorize default -P icmp
```

The opposite of `euca-authorize` is `euca-revoke`. You can use `euca-describe-groups` to obtain a list of security groups and the firewall permissions you have applied to them.

### 2.1.5. Running EC2 Instances

Now that our preparation work is done, we can start running EC2 instances.
2.1.5.1. Choosing an image
The Cloud SIG maintains an index of machine images published by Fedora. While all of the images for a given release behave the same, they differ by architecture, EC2 region, and where the root filesystem is stored (that is, instance store or EBS). Choose the image that is most appropriate for you and note its ID, which begins with ami.

2.1.5.2. Choosing an instance type
Amazon offers several instance types, which are detailed on the EC2 web site. As of the time of writing, the smallest and cheapest instance types are m1.small and t1.micro, though each of those carries a restriction: m1.small instances must use the i386 architecture. t1.micro instances have no instance storage and therefore must boot from EBS. If the image you choose fits neither of these criteria or if you simply need more resources than they can provide then you need to use a larger and more expensive instance type.

2.1.5.3. Running an instance
You can start a new instance of the image and instance type you chose with euca-run-instances. To be able to log into the new instance, you must also specify the name of the key pair you created earlier. For example, to run a t1.micro instance of the image ami-7f5a063a with a key pair named mykey, run the following command:

```
$ euca-run-instances ami-7f5a063a -t t1.micro -k mykey
RESERVATION  r-4d5ea00a  0123456789ab  default
INSTANCE     i-910fbbd6  ami-7f5a063a  pending  0  mykey  t1.micro  2011-10-11T00:00:00.000Z
us-east-1c  aki-9ba0f1de
```

The output of euca-run-instances contains the ID of the instance you just started. In the example above, the instance's ID is i-910fbbd6. You will need this ID to use tools that need to refer to the instance.

The instance starts in the pending state. When it finished booting it changes to the running state. When you terminate it it changes to the shutting-down and finally terminated states.

2.1.5.4. Logging into an instance
As the instance starts it obtains an IP address from EC2 and changes to the running state. You can check on your instances by running euca-describe-instances, optionally with the ID of the instance in question. When the instance is ready (or nearly ready) to use, euca-describe-instances will display the address you can use to log into it:

```
$ euca-describe-instances
RESERVATION  r-4d5ea00a  0123456789ab  default
INSTANCE     i-910fbbd6  ami-7f5a063a  ec2-204-236-168-22.us-east-1.compute.amazonaws.com
  ip-10-170-15-23.us-east-1.compute.internal  running  0  mykey  t1.micro
  2011-10-11T00:00:00.000Z  us-east-1c  aki-9ba0f1de
```

The public address of the instance in this example is ec2-204-236-168-22.us-east-1.compute.amazonaws.com. Other useful bits of information from this command include the availability zone in which the instance is running (us-east-1c in this case) and the time that the instance started.

---

6 http://fedoraproject.org/wiki/Cloud_SIG/EC2_Images
7 http://aws.amazon.com/ec2/instance-types/
Once the instance is running you can log into it with `ssh`. On Fedora's images you should log in as the user `ec2-user`:

```
$ ssh -i mykey.pem ec2-user@ec2-204-236-168-22.us-east-1.compute.amazonaws.com
[ec2-user@i-910fbbd6 ~]$ cat /etc/fedora-release
Fedora release 16 (Verne)
```

You can now use the instance as you would use any other computer running Fedora.

### 2.1.5.5. Terminating an instance

When you finish using an instance you should terminate it with the `euca-terminate-instances` command to free up resources and reduce your costs:

```
$ euca-terminate-instances i-910fbbd6
```

### 2.1.6. Using Elastic Block Storage (EBS)

#### 2.1.6.1. Managing volumes

EBS volumes act like removable disks that you can attach to instances, except you can create and destroy them at will. Each volume is specific to an availability zone. The following section will show you how to use and manage EBS volumes.

##### 2.1.6.1.1. Creating volumes

You can create a volume of nearly any size, in 1 GiB increments. As of the time of writing, the maximum size of a volume is 1 TiB. To create a new, empty volume, choose a size (in GiB) and the availability zone in which to create it and supply those values to `euca-create-volume`:

```
$ euca-create-volume -s 10 -z us-east-1c
VOLUME  vol-23ca3542  10  creating  2011-10-11T00:00:00.000Z
```

The command's output contains the ID of the newly-created volume. In the example above, the volume's ID is `vol-23ca3542`. You will need this ID to use tools that need to refer to the volume.

##### 2.1.6.1.2. Describing volumes

The `euca-describe-volumes` will provide a list of all volumes available to you in the entire region in addition to where they are attached:

```
$ euca-describe-volumes
VOLUME  vol-23ca3542  10  us-east-1c  available  2011-10-11T00:00:00.000Z
```

##### 2.1.6.1.3. Using volumes

For an instance to make use of a volume you must first attach the volume to the instance. You also need to supply a `device name` that the volume should appear as from inside the instance. The device name you choose must be `/dev/sdX`, where X is a letter. It will appear inside the instance as either `/dev/sdX` or `/dev/xvdX`.

```
$ euca-attach-volume -i i-910fbbd6 -d /dev/sdf vol-23ca3542
```
Volumes are zone-specific

Each volume exists only within one availability zone. A volume in a given zone can therefore only be attached to instances that are running in the same zone.

Once you have attached a volume to the instance it will appear as a disk in the instance's /dev directory, ready to be formatted and used.

```
[ec2-user@i-910fbbd6 ~]$ mkfs.ext4 /dev/xvdf
[ec2-user@i-910fbbd6 ~]$ mount /dev/xvdf /mnt
```

When you finish using a volume you can unmount it from within the instance and then detach it:

```
[ec2-user@i-910fbbd6 ~]$ umount /dev/xvdf
[ec2-user@i-910fbbd6 ~]$ logout
$ euca-detach-volume vol-23ca3542
```

2.1.6.1.4. Deleting volumes

When you finish using an EBS volume you can delete it to free up resources and reduce your costs:

```
$ euca-delete-volume vol-23ca3542
```

2.1.6.2. Using snapshots

Volume snapshots provide an easy way to save a backup copy of an entire volume. Unlike a volume, a snapshot is available to all availability zones within a region, which makes snapshots the simplest way to copy a volume between availability zones.

2.1.6.2.1. Creating a snapshot

You can create a snapshot by providing the name of the volume you wish to take a snapshot of to `euca-create-snapshot`:

```
$ euca-create-snapshot vol-23ca3542
SNAPSHOT  snap-00acc96e  vol-23ca3542  pending  2011-10-11T00:00:00.000Z
```

The command's output contains the ID of the newly-created snapshot. In the example above, the snapshot's ID is `snap-00acc96e`. You will need this ID to use tools that need to refer to the snapshot.

Volumes should not change while creating snapshots

Snapshots take time to complete. While a snapshot is in progress, ensure that the contents of the volume do not change to avoid data corruption. You can monitor a snapshot's progress with `euca-describe-snapshots`. 
2.1.6.2.2. Describing snapshots

The `euca-describe-snapshots` will provide a list of all snapshots available to you in the region:

```
$ euca-describe-snapshots
SNAPSHOT   snap-00acc96e   vol-042d3a6a   completed  2011-10-12T05:56:29.000Z  100%
```

Dealing with too much output

By default, `euca-describe-snapshots` will list all snapshots that you can access, including those that you do not own. To narrow down the command's output you can supply a list of snapshots to the command or use any of its numerous methods of filtering output.

2.1.6.2.3. Creating volumes from snapshots

To copy the contents of a snapshot to a new volume, run `euca-create-volume` and specify a snapshot instead of a size:

```
$ euca-create-volume --snapshot snap-00acc96e -z us-east-1c
```

You can create multiple volumes from the same snapshot. Each volume will be independent of the others.

2.1.6.2.4. Deleting snapshots

To delete a snapshot, use `euca-delete-snapshot`. Any volumes created from that snapshot will be unaffected.

```
$ euca-delete-snapshot snap-00acc96e
```

2.2. BoxGrinder

`BoxGrinder` creates appliances (virtual machines) from simple plain text appliance definition files for various virtual platforms. There are effectively three types of transactions that BoxGrinder performs. The first is to create an operating system image (or in BoxGrinder terminology, a build). The second operation is to convert the image to the target hypervisor. For example, this takes the raw disk image (created in the earlier step) and converts it to an EC2 AMI, a XenServer VHD, or qemu QCOW2 file. The final operation is to push the freshly converted image to the destination hypervisor or cloud platform.

This section is intended to give you a brief introduction to BoxGrinder. More details and tutorials are available online.²

2.2.1. Installing BoxGrinder

Installation of BoxGrinder is quite easy using `yum`.

```
yum install rubygem-boxgrinder-build
```

² The BoxGrinder website is located at http://boxgrinder.org/, and tutorials are available at http://boxgrinder.org/tutorials
2.2.2. Using BoxGrinder to Create a Fedora Image

Images built by BoxGrinder are generally done via an appliance definition file, though BoxGrinder also supports kickstart files. Below is a very simple appliance file that installs Fedora 15.

Building the image now that you have a definition file is quite easy.

2.2.3. Using BoxGrinder to Create a Fedora Instance on Amazon EC2

2.2.4. BoxGrinder Plugins and Platforms

2.2.4.1. OS plugins

2.2.4.2. Platform plugins

2.2.4.3. Delivery platforms
Using Tools in Fedora to Manage the Cloud

3.1. Deltacloud

3.1.1. Overview of Deltacloud

3.1.2. Typical Use Cases for Deltacloud

3.1.3. Getting Started with Deltacloud

3.2. Aeolus

Aeolus is an umbrella for a number of cloud related projects. The projects all come together to create the Conductor, a Web UI and tools for managing cloud instances across heterogeneous clouds. For instance, a user can build and launch instances into EC2, RHEV-M, and Rackspace all from the same UI.

Aeolus is built up from a number of different pieces:

Audrey

Conductor

Configure

ImageFactory

ImageWarehouse

Oz

3.3. Matahari

Matahari helps you manage systems deployed in a cloud.
Building your own cloud infrastructure using Fedora

4.1. Infrastructure as a Service (IaaS)

4.1.1. Creating your own cloud with Openstack

OpenStack, licensed under the Apache 2.0 license, provides technologies for creating open source clouds.

OpenStack consists of several different pieces:

Nova
Nova is the management stack and user interface to OpenStack.

Swift
Swift is an object storage mechanism for OpenStack

Glance
Glance is the image creation portion of OpenStack

Let's explore the various pieces of OpenStack and how they interact.

4.1.1.1. Openstack Nova

Nova is the primary operator in the Infrastructure as a Service, or IaaS, system which is used to host and manage clouds.

Additional information on Nova can be found at http://nova.openstack.org

4.1.1.2. Openstack Swift

Swift is an object-oriented storage solution.

Additional information on Swift can be found at http://swift.openstack.org.

4.1.1.3. Openstack Glance

Glance is a system used to manage virtual machine images. The service provides discovery, registration, and retrieval support. VM images can be stored using regular filesystems or in object-oriented systems like Swift.

Additional information on Glance can be found at http://glance.openstack.org.

4.1.2. CloudStack

CloudStack is an open source cloud computing platform for creating and maintaining IaaS clouds. Supporting several hypervisors and network configurations, CloudStack makes building a IaaS cloud with ease.
4.1.2.1. Installing CloudStack
CloudStack is not currently available in the Fedora or EPEL repositories. Binaries and source can be found at http://cloudstack.org/download.html. Downloading the appropriate binary and installing it via the GUI or CLI is quite easy.

Open a terminal and go to the directory where your binary file is located...

4.1.3. Sheepdog
Sheepdog is a distributed object-based storage system for QEMU/KVM. Providing high-availability, block-level storage, Sheepdog can manage hundreds of nodes automatically balancing load when failures and additions occur.

For additional information please visit http://www.osrg.net/sheepdog.

4.1.3.1. Prerequisites
Before installing

4.1.4. HekaFS
HekaFS (formerly CloudFS) is cloud filesystem built on top of Gluster. It provides a set of translators to make Gluster more suitable as a cloud file system. HekaFS is a feature for the Fedora 16 release, and the Fedora Cloud special interest group (SIG) is actively involved in development and testing.

The Fedora HekaFS feature page has information about the status of HekaFS as a feature in a future release of Fedora.

See Jeff Darcy's HekaFS blog to learn more about the philosophy of HekaFS and find out what's happening with HekaFS development.

4.1.4.1. How to use HekaFS
General HekaFS discussions are at: https://fedorahosted.org/mailman/listinfo/cloudfs-general.

Red Hat Cloud OpenShift uses HekaFS. The user community for that product is found at: http://www.redhat.com/openshift/forums.

4.1.4.2. How to participate in HekaFS
Let's start with some important links.

The upstream project overview (slides, blog posts) is at http://hekafs.org/cloudfs-overview/

The source repository is at: http://git.fedorahosted.org/git/?p=CloudFS.git

The developer mailing list is at: https://fedorahosted.org/mailman/listinfo/cloudfs-devel

The user mailing list (pretty empty so far) is at https://fedorahosted.org/mailman/listinfo/cloudfs-general

HekaFS is still under active development, so there are still many ways to contribute. Here are some possibilities.

Use it, report bugs, suggest features. The packaging and documentation are still works in progress, so this might require some interaction with one of the developers, but we don't bite. Feel free to contact jdarcy@fedoraproject.org, jdarcy@redhat.com, or jeff@pl.atyp.us - all the same person - and I'd be delighted to help you through it.
Use the upstream (http://gluster.org), report bugs, suggest features. For the most part, HekaFS is GlusterFS with a couple of extra bits, so many of the issues you're likely to experience are actually GlusterFS issues. HekaFS is not just its own code, though. It's also a sort of unofficial "CloudFS SIG" within the GlusterFS community. If you want GlusterFS fixes or enhancements because of your interest in HekaFS, we can drive those as part of HekaFS.

Say hi in Freenode IRC. #cloudfs is pretty dead (let's fix that), but #gluster and #fedora-cloud are pretty lively.

Help with documentation. There are several bits of documentation in the source tree, but a little loving from a genuine wordsmith would go a long way.

Improve the user interface. There is a web-based management UI with pretty good functionality - here are some screenshots and stuff - but the user experience is pretty old-school. If you're a real web programmer who knows all the modern AJAX/JavaScript ways of doing stuff, or a graphic designer, I'm sure you could make something a lot better.

Improve the core code. Some parts of HekaFS are practically done, but others are still in active development and there's still plenty of room for more developers. In particular, if you're a security/cryptography expert, more review of those pieces would be most welcome. People might look at HekaFS as a more secure alternatives to Dropbox or Jungledisk, both of which have had problems in this area recently, so this is a great area to focus on. Despite being a file system this is not kernel code, by the way. It should be relatively free of those additional complications and comprehensible to non-kernel programmers.

Get the word out. The number of participants is directly proportional to the number of people who've heard about it, and some people who might want to contribute might not even know we're here. Blog, tweet, whatever, let people know that there's an open-source project to create a scalable, secure, multi-tenant filesystem.

4.2. Platform as a Service (PaaS)

4.2.1. What is OpenShift?
OpenShift is a PaaS management solution

4.3. Software as a Service (SaaS)
Using Fedora to Create Cloud Images

5.1. Creating cloud images using Fedora
Storage in the Cloud

6.1. Ceph

For additional information please visit http://ceph.newdream.net.

6.2. Amazon S3

6.2.1. Overview of S3
The Amazon EC2 provides a place for anyone with an account to create and operate their own virtualized computing environment. As one of the largest Infrastructure-as-a-Service (IaaS) providers, Amazon provides an almost unlimited
Appendix A. Revision History

Revision 16.4-1 Thu 3 November 2011
Updated EC2 chapter.

Eric Christensen
sparks@fedoraproject.org

Revision 16.3-1 Fri September 16 2011
Updated CloudStack section.
Modified chapter layout.
Added HekaFS chapter information from Fedora wiki.
Added information from Fedora Cloud mini-hackfest.

Eric Christensen
sparks@fedoraproject.org

Revision 16.2-1 Mon Jun 20 2011
Created sections for Ceph.
Provided basic information on several sections.

Eric Christensen
sparks@fedoraproject.org

Revision 16.1-1 Sat Jun 11 2011
Created sections for Aeolus, BoxGrinder, HekaFS, Cloudstack, and Sheepdog.

Eric Christensen
sparks@fedoraproject.org

Revision 0.1-1 Wed Oct 13 2010
Initial creation of book by publican
Created sections for Introduction, Images, Amazon EC2, and Openstack.

Eric Christensen
sparks@fedoraproject.org

Initial creation of book by publican
Created sections for Introduction, Images, Amazon EC2, and Openstack.
Index

F

feedback
  contact information for this manual, vii