

# Amazon S3, EC2 Tools

Zoran B. Djordjević

Csci E185 Big Data Analytics

# Signup for S3

The screenshot shows the Amazon S3 website interface. At the top, there is a browser address bar with the URL `http://aws.amazon.com/s3/`. Below the address bar is the Amazon Web Services logo and navigation links for "Sign in to the AWS Management Console" and "Create an AWS Account". A horizontal navigation bar contains links for "AWS", "Products", "Developers", "Community", "Support", and "Account".

On the left side, there is a "Products & Services" dropdown menu. Underneath it, a sidebar lists "Amazon S3" with sub-links for "Amazon S3 Overview", "FAQs", and "Amazon S3 SLA". Below the sidebar is a "Developer Resources" section.

The main content area features the heading "Amazon Simple Storage Service (Amazon S3)". Below the heading, there is a paragraph: "Amazon S3 is storage for the Internet. It is designed to make web-scale computing easier for developers." This is followed by another paragraph: "Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of web sites. The service aims to maximize benefits of scale and to pass those benefits on to developers." To the right of this text is a prominent "Sign Up For Amazon S3" button.

# Pricing

Region: <input type="text" value="US Standard"/>	Storage Pricing	
	Standard Storage	Reduced Redundancy Storage
First 1 TB / month	\$0.125 per GB	\$0.093 per GB
Next 49 TB / month	\$0.110 per GB	\$0.083 per GB
Next 450 TB / month	\$0.095 per GB	\$0.073 per GB
Next 500 TB / month	\$0.090 per GB	\$0.063 per GB
Next 4000 TB / month	\$0.080 per GB	\$0.053 per GB
Over 5000 TB / month	\$0.055 per GB	\$0.037 per GB

Region: <input type="text" value="US Standard"/>	Request Pricing	
PUT, COPY, POST, or LIST Requests	\$0.01 per 1,000 requests	
GET and all other Requests †	\$0.01 per 10,000 requests	
† No charge for delete requests		

Region: <input type="text" value="US Standard"/>	Data Transfer Pricing	
<b>Data Transfer IN</b>		
All data transfer in	\$0.000 per GB	
<b>Data Transfer OUT</b>		
First 1 GB / month	\$0.000 per GB	
Up to 10 TB / month	\$0.120 per GB	
Next 40 TB / month	\$0.090 per GB	
Next 100 TB / month	\$0.070 per GB	
Next 350 TB / month	\$0.050 per GB	
Next 524 TB / month	<a href="#">Contact Us</a>	
Next 4 PB / month	<a href="#">Contact Us</a>	

# Pricing

- Data transfer "in" and "out" refers to transfer into and out of an Amazon S3 Region. There is no Data Transfer charge for data transferred within an Amazon S3 Region via a COPY request.
- Data transferred via a COPY request between Regions is charged at regular rates. There is no Data Transfer charge for data transferred between Amazon EC2 and Amazon S3 within the same Region.
- Data transferred between Amazon EC2 and Amazon S3 across all other Regions (i.e. between the Amazon EC2 Northern California and Amazon S3 US Standard Regions) will be charged at Internet Data Transfer rates on both sides of the transfer.

# Signing up

- Go to <http://aws.amazon.com/S3>
- Click [Sign Up For Amazon S3](#)
- Click [Complete Sign Up](#)

# Functionality of S3

- Write, read, and delete objects containing from 1 byte to 5 gigabytes of data each. The number of objects you can store is unlimited.
- Each object is stored in a bucket and retrieved via a unique, developer-assigned key.
- A bucket can be stored in one of several Regions. You can choose a Region to optimize for latency, minimize costs, or address regulatory requirements.
- Amazon S3 is currently available in the US Standard, EU (Ireland), and US-West (Northern California) Regions.
- The US Standard Region automatically routes requests to facilities in Northern Virginia or the Pacific Northwest using network maps.
- Objects stored in a Region never leave the Region unless you transfer them out. For example, objects stored in the EU (Ireland) Region never leave the EU.

# Essential Concepts

## **Objects**

- Objects are the fundamental entities stored in Amazon S3.
- Objects consist of object data and metadata and can range in size from 1 byte to 5 gigabytes.
- The data portion is opaque to Amazon S3.
- The key is the handle that you assign to an object that allows you retrieve it later.

## **The metadata**

- The metadata is a set of name-value pairs that describe the object.
- The developer can specify custom metadata and standard HTTP metadata, such as Content-Type.

# Essential Concepts

## **Buckets**

- You upload objects into buckets.
- There is no limit to the number of objects that you can store in a bucket.
- The bucket provides a unique namespace for the management of objects contained in the bucket. Each developer can own up to 100 buckets at a time??
- You own each bucket you create. We charge you for storing objects in your buckets and for transferring objects in and out of your buckets.

# Bucket Namespace

- Every object stored in Amazon S3 is contained in a bucket.
- Buckets partition the namespace of objects stored in Amazon S3 at the top level. Within a bucket, you can use any names for your objects, but bucket names must be unique across all of Amazon S3.
- Buckets are similar to Internet domain names.
- Only one person or organization can own a bucket within Amazon S3.
- The similarities between buckets and domain names is not a coincidence—there is a direct mapping between Amazon S3 buckets and subdomains of `s3.amazonaws.com`.
- Objects stored in Amazon S3 are addressable using the REST API under the domain `s3.amazonaws.com/bucketName`.
- For example, if the object `homepage.html` is stored in the Amazon S3 bucket `mybucket` its Internet address would be
- `http://s3.amazonaws.com/mybucket/homepage.html`

# Essential Concepts

## Keys

- A key is the unique identifier for an object within a bucket.
- The key is the object's name and can not be longer than 1028 bytes.
- Every object has exactly one key. Together, a bucket name and a key uniquely identify an object in Amazon S3.
- You can access every object in Amazon S3 by a combination of the service endpoint, bucket name, and key.
- For example, in `http://s3.amazonaws.com/bucketName/photos/myPicture.jpg`
  - `bucketName` is the bucket name and ,
  - `/photos/myPicture.jpg` is the key.

# Authentication

- Authentication is the process of verifying the identity of a user or service trying to access an Amazon Web Services (AWS) product.
- Every interaction with Amazon S3 is authenticated or anonymous. When you sign up for an AWS account, you are provided with an AWS Access Key ID and a Secret Access Key.
- When you perform a request with Amazon S3, you assemble the request, perform a hash on the request using your Secret Access Key, attach the Signature (hash) to the request, and forward it to Amazon S3.
- Amazon S3 verifies the Signature is a valid hash of the request and, if authenticated, processes the request.

# Naming Buckets and Locations

- Buckets should be named so that you can reference your bucket using the convention
- `s3.amazonaws.com/<bucketname>..`
  - Use 3 to 63 characters.
  - Use only lower case letters (at least one), numbers, '.' and '-'.
  - Don't start or end the bucket name with '.' and don't follow or precede a '.' with a '-'.
  - Keys can be named with any properly encoded UTF-8 character. Literal '+' characters should always be URL encoded.

# Amazon EC2, Your AMI-s

Zoran B. Djordjević

Csci E175 Cloud Computing  
and SaaS

# Two Kinds of AMI-s

- All AMIs are categorized as either
  - *backed by Amazon EBS* or
  - *backed by instance store.*
- **AMI Backed by EBS** means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot.
- **An Instance Store AMI** means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.

# EBS vs. Instance Store AMI

Characteristic	Amazon EBS-Backed	Amazon instance store-backed
Boot Time	Usually less than 1 minute	Usually less than 5 minutes
Size Limit	1 TiB	10 GiB
Root Device Volume	Amazon EBS volume	Instance store volume
Data Persistence	Data on EBS volumes persists after instance termination; you can also attach instance store volumes that don't persist after termination	Data on instance store volumes persists only during the life of the instance;
Upgrading	The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped.	Instance attributes are fixed for the life of an instance
Charges	Instance usage, Amazon EBS volume usage, and Amazon EBS snapshot charges	Instance usage and Amazon S3 charges for AMI storage
AMI Creation/Bundling	Uses a single command/call	Requires installation and use of AMI tools
Stopped State	Can be placed in stopped state where instance is not running, but the instance is persisted in Amazon EBS	Cannot be in stopped state; instances are running or terminated

# Do We have all Credentials

## Sign-In Credentials

To sign in to AWS web sites and applications, AWS requires your Amazon e-mail address and password. Additionally, it supports the AWS Multi-Factor Authentication option. Each sign-in credential is explained below.

## Access Credentials

There are three types of access credentials used to authenticate your requests to AWS services: (a) access keys, (b) X.509 certificates, and (c) key pairs. Each access credential type is explained below.

**Access Keys** | X.509 Certificates | Key Pairs

Use access keys to make secure REST or Query protocol requests to any AWS service API. We create one for you when your account is created — see your access key below.

### Your Access Keys

Created	Access Key ID	Secret Access Key	Status
May 14, 2009	AKIAJXHJT7JMTWJRQ4TQ	Show	Active (Make Inactive)

[Create a new Access Key](#)

For your protection, you should never share your secret access keys with anyone. In addition, industry best practice recommends frequent key rotation.

- Email address/password
- Access Key + Secret Access Key
- X509 Certificate + Private Key
- SSH Key Pairs

# Two More Credentials, Do not forget

- AWS Account ID: 324443219834
- Canonical user ID

## Account Identifiers

AWS uses two types of account identifiers — canonical user ID and AWS account ID. These account identifiers are used to share resources between accounts.

The canonical user ID can be used exclusively for Amazon S3 resources such as buckets or files.

The AWS account ID can be used for all AWS service resources except Amazon S3. These resources include Amazon EC2 AMIs, Amazon EBS snapshots, Amazon SQS queues, etc.

**AWS Account ID:** 3244-4321-9834

**Canonical User ID:** [View canonical user ID](#)

[Learn more about](#)

### Canonical User Id

The canonical user ID is used when granting permissions in Amazon S3. For more information see "Access Control Lists" section of the Amazon S3 Developers Guide.

**ID:** ab4bred55768fdge89337202sadeo84373ffrdes8234mjkl;3ew2q1q223re

# EC2 API Tools & EC2 AMI Tools

- AWS Management Console is fine when we have to deal with a few isolated tasks.
- Massive and speedy work requires scripting or programming tools.
- For use on the client side we have **command line EC2 API Tools.**
- **EC2 API Tools** provide detailed control of instance lifecycle, security and storage
- **EC2 AMI Tools** come installed on all Amazon AMI-s. You should install them on AMI-s you build.
- EC2 AMI Tools serve many tasks, one of the most important is so called “bundling”.

# Command Line Reference Page

<http://docs.amazonwebservices.com/AWSEC2/latest/CommandLineReference/Welcome.html>

← → ↻ 🏠 docs.amazonwebservices.com/AWSEC2/latest/CommandLineReference/Welcome.html



**Amazon Elastic Compute Cloud**  
CLI Reference (API Version 2012-08-15)

Search:

- **Welcome**
- ▶ [API Tools Reference](#)
- ▶ [AMI Tools Reference](#)
- [Document History](#)

[AWS Documentation](#) » [Amazon EC2](#) » [Command Line Reference](#) » [Welcome](#)

[View PDF](#) [Go to the forums](#)

## Welcome

This is the *Amazon Elastic Compute Cloud Command Line Reference*. It provides descriptions, syntax, and *API tools* wrap the Amazon EC2 API actions. You install and run the *AMI tools* on an Amazon EC2 instance backed AMIs.

### Note

This guide also includes the commands for Amazon Virtual Private Cloud (Amazon VPC). For more [Virtual Private Cloud User Guide](#).

How Do I?	Relevant Topics
Download the Amazon EC2 API tools.	<a href="#">Amazon EC2 API Tools</a>
Prepare a computer to use the Amazon EC2 API tools.	<a href="#">Setting Up the Command Line Tools</a>
Get an alphabetical list of the Amazon EC2 API tools.	<a href="#">Commands for API Tools</a>
Get a list of the common options for the API tools.	<a href="#">Common Options for API Tools</a>
Download the Amazon EC2 AMI tools.	<a href="#">Amazon EC2 AMI Tools</a>
Get an alphabetical list of the Amazon EC2 AMI tools.	<a href="#">Commands for AMI Tools</a>
Get a list of the common options for the AMI tools.	<a href="#">Common Options for AMI Tools</a>
Get a list of regions and endpoints for each AWS product.	<a href="#">Regions and Endpoints</a>

# Download EC2 API Tools ZIP

- File `ec2_api_tools.zip` comes from [http://aws.amazon.com/developertools/351?\\_encoding=UTF8&jiveR](http://aws.amazon.com/developertools/351?_encoding=UTF8&jiveR)



## Amazon EC2 API Tools

Developer Tools > Amazon EC2 API Tools



### Community Contributed Software

- Amazon Web Services provides links to these packages as a convenience for our customers, but software not authored by an "@AWS" account has not been reviewed or screened by AWS.
- Please review this software to ensure it meets your needs before using it.

The API 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.

### Details

**Submitted By:** David@AWS

**AWS Products Used:** Amazon EC2

**Languages(s):** Other

**License:** Other

**Created On:** August 23, 2006 9:00 PM GMT

**Last Updated:** August 2, 2012 8:37 PM GMT

### Download

Download the Amazon EC2 API Tools from Amazon S3.

See the related Amazon EC2 technical documentation below for more information on using the command-line tools.

# Installation of EC2 API Tools, Windows

- Unzip `ec2_api_tools.zip` and possibly rename the directory.
- You need Java 1.5 or later.
- Set the following environmental variables:

```
JAVA_HOME=C:\Program Files\Java\jdk1.7.0_07
```

```
EC2_HOME=C:\AWS\ec2-api-tools-1.6.3.0
```

```
EC2_CERT=
```

```
    c:\AWS\hu\cert-BMNQ7GNGXBX5YMFAHCISRWDUUDRXQFPH.pem
```

```
EC2_PRIVATE_KEY=
```

```
    c:\AWS\hu\pk-BMNQ7GNGXBX5YMFAHCISRWDUUDRXQFPH.pem
```

```
PATH=C:\AWS\ec2-api-tools-1.6.3.0\bin;
```

```
    C:\Program Files\Java\jdk1.7.0_07\bin;...
```

```
AWS_ACCESS_KEY= AKJAJYHJT7HMTZJRQ3TQ
```

```
AWS_SECRET_KEY=gUlbtSWIrPBcyqhtiTyMwj122OoRPcJ+F1U2rnKE
```

```
AWS_URL=https://ec2.us-east-1.amazonaws.com
```

# Installation of EC2 API Tools on Unix

- Unzip `ec2_api_tools.zip`
- Set environmental variables in your `.bash_profile` or profile file of whatever shell you are using:

```
EC2_HOME=~/.ec2-api-tools
EC2_CERT=
  ~/.ec2/cert-JYJA2342D231E982332902CQFTARPMTR3.pem
EC2_PRIVATE_KEY=
  ~/.ec2/pk-JYJA2342D231E982332902CQFTARPMTR3.pem
PATH=$PATH:~/.ec2-api-tools/bin
AWS_ACCESS_KEY= AKJAJYHJT7HMTZJRQ3TQ
AWS_SECRET_KEY=gUlbtSWIrPBcyqhtiTyMwj122OoRPcJ+F1U2rnKE
AWS_URL=https://ec2.us-east-1.amazonaws.com
```

- You can do it on the command prompt, as well:

```
$ export EC2_PRIVATE_KEY=~/.ec2/ec2/pk-
  JYJA2342D231E982332902CQFTARPMTR3.pem
$ export EC2_CERT=~/.ec2/cert-
  JYJA2342D231E982332902CQFTARPMTR3.pem
```

- Please note that command line tools use your X509 credentials to identify you to the Cloud.

# Partial List of EC2 API Command Line Tools

## AMIs

- ec2-create-image
- ec2-deregister
- ec2-describe-image-attribute
- ec2-describe-images
- ec2-migrate-bundle
- ec2-migrate-image
- ec2-modify-image-attribute

## Availability Zones and Regions

- ec2-describe-availability-zones
- ec2-describe-regions

## Elastic Block Store

- ec2-attach-volume
- ec2-create-snapshot
- ec2-create-volume
- ec2-delete-snapshot
- ec2-delete-volume
- ec2-describe-snapshot-attribute
- ec2-describe-snapshots
- ec2-describe-volumes
- ec2-detach-volume
- ec2-modify-snapshot-attribute
- ec2-reset-snapshot-attribute

## Elastic IP Addresses

- ec2-allocate-address
- ec2-associate-address
- ec2-describe-addresses
- ec2-disassociate-address
- ec2-release-address

## General

- ec2-get-console-output

## Images

- ec2-register
- ec2-reset-image-attribute

## Instances

- ec2-describe-instance-attribute
- ec2-describe-instances
- ec2-modify-instance-attribute
- ec2-reboot-instances
- ec2-reset-instance-attribute
- ec2-run-instances
- ec2-start-instances
- ec2-stop-instances
- ec2-terminate-instances

# EC2 API Command Line Tools

## Key Pairs

- ec2-add-keypair
- ec2-delete-keypair
- ec2-describe-keypairs
- ec2-fingerprint-key

## Monitoring

- ec2-monitor-instances
- ec2-unmonitor-instances
- Reserved Instances
- ec2-describe-reserved-instances
- ec2-describe-reserved-instances-offerings
- ec2-purchase-reserved-instance-offering

## Security Groups

- ec2-add-group
- ec2-authorize
- ec2-delete-group
- ec2-describe-group
- ec2-revoke

## Spot Instances

- ec2-cancel-spot-instance-requests
- ec2-create-spot-datafeed-subscription
- ec2-delete-spot-datafeed-subscription
- ec2-describe-spot-datafeed-subscription
- ec2-describe-spot-instance-requests
- ec2-describe-spot-price-history
- ec2-request-spot-instances

## Windows

- ec2-bundle-instance
- ec2-cancel-bundle-task
- ec2-describe-bundle-tasks
- ec2-get-password
- ec2-get-password-data

- Please note, these are Windows or Linux

# Where to find EC2 AMI Tools

- <http://aws.amazon.com/developertools/368>



## Amazon EC2 AMI Tools

Developer Tools > Amazon EC2 AMI Tools



### Community Contributed Software

- Amazon Web Services provides links to these packages as a convenience for our customers, but software not authored by an "@AWS" account has not been reviewed or screened by AWS.
- Please review this software to ensure it meets your needs before using it.

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.

### Details

**Submitted By:** [Dave@AWS](#)

**AWS Products Used:** Other

**Languages(s):** Other

**License:** Other

**Created On:** August 25, 2006 7:56 PM GMT

**Last Updated:** December 15, 2011 2:56 AM GMT

Download the Amazon EC2 AMI Tools from Amazon S3:

- [Download RPM](#)
- [Download ZIP File](#)

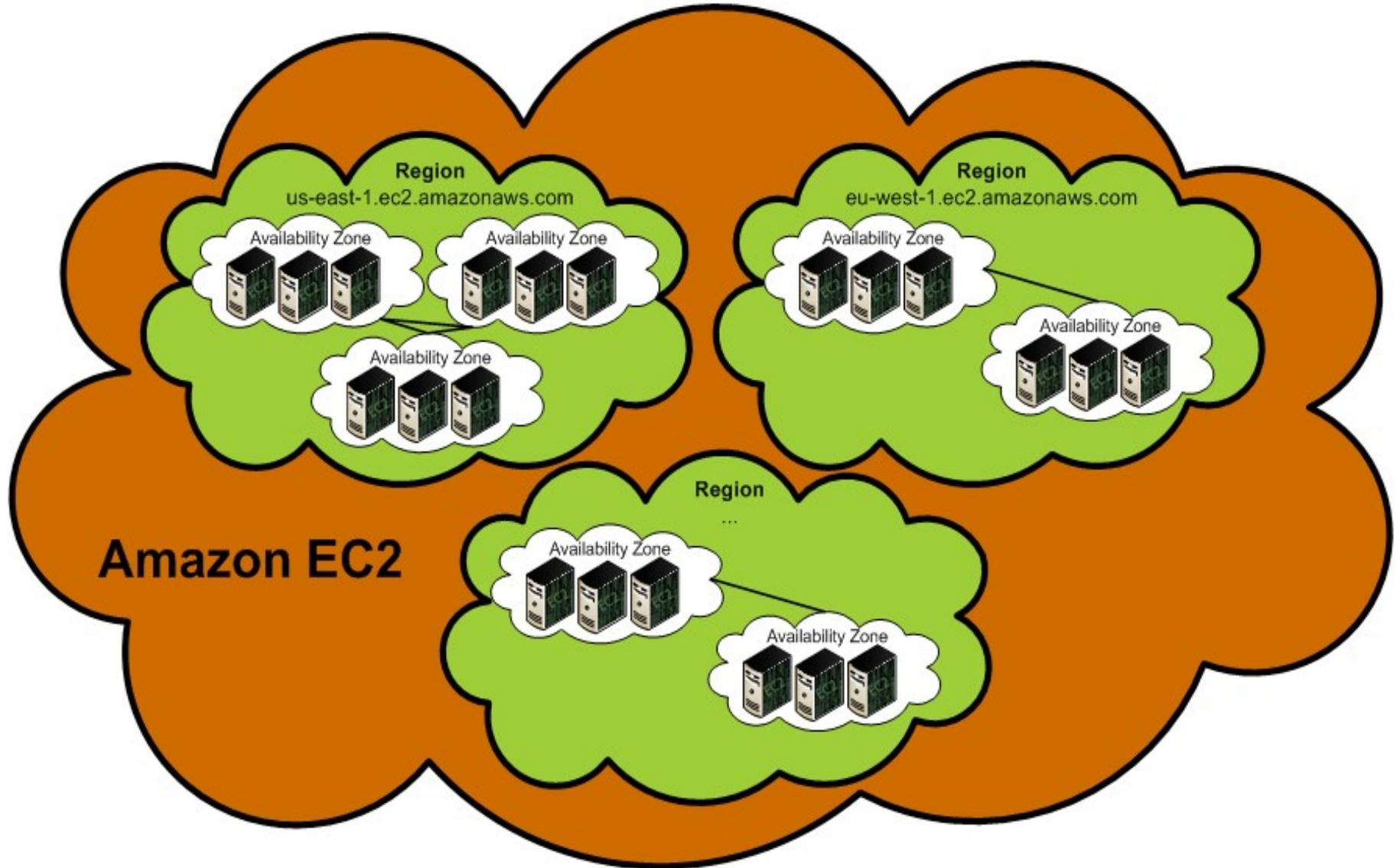
# AMI API Tools

- ec2-bundle-image
- ec2-bundle-vol
- ec2-delete-bundle
- ec2-download-bundle
- ec2-migrate-bundle
- ec2-migrate-manifest
- ec2-unbundle
- ec2-upload-bundle
- These are used for bundling of Linux, Unix, instances on the instances backed AMI-s themselves.
- Windows' instances are bundled with a set of commands in the EC2 API Set.
- Please note, these are Windows or Linux commands.
- Windows users do not need Cygwin to run ec2?

# Regions and Availability Zones

- Amazon EC2 allows us to place instances in multiple locations.
- Amazon EC2 locations are composed of Regions and Availability Zones and Regions.
- Regions are located in separate geographic areas (e.g., US and EU, South Asia, Japan). Currently, there are 7 regions.
- Availability Zones are distinct locations within a Region that are engineered to be isolated from failures in other Availability Zones. They provide inexpensive, low latency network connectivity to other Availability Zones in the same Region.
- By launching instances in separate Regions, you design your application to be closer to specific customers or to meet legal or other requirements.
- By launching instances in several Availability Zones, you can protect your applications from the failure of a single location.
- The following graphic shows a representation of Amazon EC2. Each Region is completely independent. Availability Zones are isolated, but connected through low-latency links to other zones.

# Regions and Availability Zones



# Using Command Line Tools

- For example, find existing regions:

```
C:\ec2> ec2-describe-regions
REGION    eu-west-1          ec2.eu-west-1.amazonaws.com
REGION    sa-east-1          ec2.sa-east-1.amazonaws.com
REGION    us-east-1          ec2.us-east-1.amazonaws.com
REGION    ap-northeast-1    ec2.ap-northeast-1.amazonaws.com
REGION    us-west-2          ec2.us-west-2.amazonaws.com
REGION    us-west-1          ec2.us-west-1.amazonaws.com
REGION    ap-southeast-1    ec2.ap-southeast-1.amazonaws.com
```

- Find availability zones

```
C:\ec2>ec2-describe-availability-zones
AVAILABILITYZONE    us-east-1a          available    us-east-1
AVAILABILITYZONE    us-east-1b          available    us-east-1
AVAILABILITYZONE    us-east-1c          available    us-east-1
AVAILABILITYZONE    us-east-1d          available    us-east-1
```

- Some times it is easier to fetch information on the command prompt than to go through the browser.

# Find Help for a Specific Command

## --help

```
C:\AWS\hu>ec2-describe-availability-zones --help
```

### SYNOPSIS

```
ec2daz ([ec2-describe-availability-zones])  
ec2daz [GENERAL OPTIONS] [ZONE [ZONE [...]]]
```

### GENERAL NOTES

Any command option/parameter may be passed a value of '-' to indicate that values for that option should be read from stdin.

### DESCRIPTION

List the availability zones you have access to.  
The ZONE parameter specifies the zones(s) to be described.  
If unspecified, all your zones will be returned.

### GENERAL OPTIONS

-O, --aws-access-key KEY

AWS Access Key ID. Defaults to the value of the AWS\_ACCESS\_KEY environment variable (if set).

-K, --private-key KEY

[DEPRECATED] Specify KEY as the private key to use. Defaults to the value of the EC2\_PRIVATE\_KEY environment variable (if set). Overrides the default.

--region REGION

Specify REGION as the web service region to use.  
This option will override the URL specified by the "-U URL" option and EC2\_URL environment variable.

# Review instances associated with you

- Find which instances you have around

```
C:\ec2> ec2-describe-instances
```

```
RESERVATION      r-864932ee      951414139794      lamp-basic
INSTANCE       i-8551a7ee     ami-26df324f     ec2-184-73-
36-42.compute-1.amazonaws.com      ip-10-243-122-
181.ec2.internal running ec2zjordj2HU0221  0
  m1.small        2010-02-22T15:28:53+0000      us-
east-1d          aki-94c527fd      ari-96c527ff
monitoring-disabled 184.73.36.42      10.243.122.181
  ebs

BLOCKDEVICE      /dev/sda1      vol-0bed2562      2010-02-
22T01:45:42.000Z

RESERVATION      r-e8483380      951414139794      default
INSTANCE       i-e95ea882     ami-0f54b966
  stopped ec2zjordj2HU0221      0      m1.small
  2010-02-22T01:49:16+
0000      us-east-1d      aki-92ba58fb      ari-94ba58fd
  monitoring-disabled      ebs

BLOCKDEVICE      /dev/sda1      vol-13ed257a      2010-02-
22T01:49:28.000Z
```

# Instance Information in AWS Console

- This is certainly easier to visualize.
- Unfortunately, you cannot GREP through it, though.

	Instance	AMI ID	Root Device Type	Type	Status	Lifecycle	Public DNS
<input type="checkbox"/>	 i-8551a7ee	ami-26df324f	ebs	m1.small	 running	normal	ec2-184-73-36-42.compute-1
<input checked="" type="checkbox"/>	 i-e95ea882	ami-0f54b966	ebs	m1.small	 stopped	normal	

EC2 Instance: i-e95ea882

Description Monitoring

<b>AMI ID:</b>	ami-0f54b966	<b>Zone:</b>	us-east-1d
<b>Security Groups:</b>	default	<b>Type:</b>	m1.small
<b>Status:</b>	stopped	<b>Owner:</b>	951414139794
<b>Reservation:</b>	r-e8483380	<b>Ramdisk ID:</b>	ari-94ba58fd
<b>Platform:</b>	-	<b>Key Pair Name:</b>	ec2zdjordj2HU0221
<b>Kernel ID:</b>	aki-92ba58fb	<b>Monitoring:</b>	disabled
<b>AMI Launch Index:</b>	0	<b>Elastic IP:</b>	-
<b>Root Device:</b>	/dev/sda1	<b>Root Device Type:</b>	ebs
<b>Block Devices:</b>	/dev/sda1=vol-13ed257a:attached:2010-02-22T01:49:28.000Z:true		
<b>Lifecycle:</b>	normal		
<b>Public DNS:</b>			

# Stop, terminate, start an AMI

```
C:\ec2> ec2-stop-instances i-8551a7ee
```

```
INSTANCE          i-8551a7ee          running  stopping
```

```
C:\ec2> ec2-start-instances i-e95ea882
```

```
INSTANCE          i-e95ea882          stopped  pending
```

```
C:\ec2> ec2-terminate-instances i-8551a7ee
```

```
INSTANCE          i-8551a7ee          stopping  
shutting-down
```

# Selecting an AMI

```
C:\ec2> ec2-describe-images -o self -o amazon
```

```
IMAGE    ami-27e7004e    /aws-console-quickstart-
amis/perl/1.2/perlquickstart.manifest.xml    amazon    av
IMAGE    ami-60da3d09    /aws-console-quickstart-amis/phpquickstart-
v1.4.manifest.xml    amazon    available
IMAGE    ami-5ee70037    /aws-console-quickstart-
amis/ruby/1.1/rubyquickstart.manifest.xml    amazon    av
IMAGE    ami-45e7002c    /aws-console-quickstart-
amis/tomcat/1.1/tomcatquickstart.manifest.xml    amazon    av
IMAGE    aki-d4ca2dbd    aki-linux-2.6.18.92-92.el5xen-xfs/vmlinuz-
2.6.18.92-92.el5xen.i386.aki.manifest.xml
IMAGE    aki-46e7002f    aki-linux.2.6.21.7-2.fc8xen-
xfs/vmlinuz.manifest.xml    amazon    available    public
IMAGE    ami-84db39ed    amazon/fedora-8-i386-v1.14-std    amazon    available
public    i386    ma
BLOCKDEVICEMAPPING    /dev/sda1    snap-b9aa35d0    15
IMAGE    ami-86db39ef    amazon/fedora-8-x86_64-v1.14-std    amazon
available    public    x86_64
BLOCKDEVICEMAPPING    /dev/sda1    snap-bbaa35d2    15
```

## Syntax:

```
ec2-describe-images [ami_id ...] [-a] [-o owner ...]
                    [-x user_id]
```

# Could narrow the search

```
C:\ec2> ec2-describe-images -o self -o amazon |
    egrep -i ami-84db39ed                -- On Linux
    findstr /i ami-84db39ed              -- On Windows
IMAGE    ami-84db39ed    amazon/fedora-8-i386-v1.14-std    amazon
         available    public    i386    machine    aki-94c527fd    ari-96c527ff    ebs
```

```
C:\ec2> ec2-describe-images -o self -o amazon |
    egrep -i ibm

IMAGE    ari-868174ef    ec2-paid-ibm-images-db2/initrd-2.6.16.60-0.29-
         xenpae.i38
6.manifest.xml    amazon    available    public    i386    ramdisk
                 instance-store    paravirtual    xen
IMAGE    ari-88e60be1    ec2-paid-ibm-images-db2/initrd-2.6.16.60-0.29-
         xenpae.i38
6.manifest.xml    amazon    available    public    [devpay: A481895D]
         i386
ramdisk    instance-store    paravirtual    xen
IMAGE    ari-888174e1    ec2-paid-ibm-images-db2/initrd-2.6.16.60-0.30-
         xen.x86_64
.manifest.xml    amazon    available    public    x86_64    ramdisk
                 instance-store    paravirtual    xen
paravirtual    xen
```

# Create, Delete a SSH Key Pair

```
C:\>ec2-add-keypair ec2zdjordj0222HU > ec2zdjordj0222HU.pem
```

- Will have to remove the first line, though. Use `sed` or something.

```
C:\ vi ec2zdjordj0222HU.pem
KEYPAIR ec2zdjordj0222HU          4c:d1:f4:4a:d0:ff:c3:d4:ec:7f:31:c2:58:dd:68:
-----BEGIN RSA PRIVATE KEY-----
MIIEogI BAAKCAQEAgjkUPLRLdXWsbMs1L5yjuBn+f2S77FggIUGhwueS5bTjipzk108tz0rztjtFR
qDYKJfDr1JBeBkTfgLf015fxVKmhdSgvJcpOQpCR0XCI86DcUGDCALtgXRnzKs7A0DmvmZ6XkPFU
U95YX6bhBexXj+OWWA6RAETIzTp/UlKn0gQS2TI705g6YAL18vj3lCZq4WytbmygxfSZPCUfcKHO
uZKt415BecxgJ5dKRR1CuUA52PMuGUMUxUwN1PPEgiUr+ICao0F6nbRqSN/nK2C0/uU5nhXfNPFr
IHgj5Ub7CYk32JS2YyM7tY0YZDtABIMNF/7s+Uv/TuLFrI1wjwxPGwIDAQABAoI BAA98mQBRE2O9
6C19h4us0uzeK6paon5vFWibUdoGeeDX1mRZp/BcL017Y2iuhHMJHuWuah2DcZjBGo+lJqzxx0Um
CbtUYCd+9DHuxtDDrUTH9/P/bF7jUaEQ2BWK2BZSKtLZMCqI+r7cb0ZKRih857M0FQOU41fjJPCi
M4ZUDQyhUq0oqUEuNieijYQ0eL+rE0cZkx+DvF+cFpinv6aq8sB9yMe7Xa7htpZ2gx5UzR/cAJhs
8mJHcndJEZM36vUbe9w0YHDWtPSquFt0g0r9uj5i1q3UFd0+OZBo+UqUXyOkDPDCdNUwS+8JRKJb
gUEkj/CNU49qtxfgcECuI Br4UdECgyEA5C219qBhH7NWy1H1twghnvUD2NbXsw/GYzDeDAUd9HOG
txY0lc5nX09w8DkkE4S1KdJIz9vd1UyfsTF8scRb52MnaPZTZGfy3ufTgcXzp4PkeMooW1CcX8N9
4BwaCW32gW5z15g56p8ltqzNTs1I5aAZjwbr0YyJQUq3phfE6pUCgyEAn53dInMvYILhAfypgbLK
5PcDoM69x0YIqyT1ZEnn2QB7vAo7okJGUIIaqx0fRdxcdRZNX7W2d5h89mT9qdxYS2Znk9GQE6YW
DTUPUcSsys06UffygR68aTSs47ha1PkGs9YrOUD7HWwEFBnCwPjWiwZBUdhsbj8N0og+jz7s6lu8C
gYAKu+cByhunJR00NNzURofB3Y7XxYUShdzct7EyaY82L7UXgdDBd1s/cPgkNwAmw6WsBC96gbr
ftb4U/akuJMemJKDCotzvrwbJla51SEYMUkMxtiiPFC5ARo5TtA1E5zWknPyNZ4phH1HP3vheKL
AtQQ+WdgjOndqXEOpBgEXQKBgDPsP45AWA4WJSnnTiDBz5yx+f0qo7Lufcdmg2jnXcNKK/W0gnej
BYqU+CXnwE+SgoWf ggX7b5LyCNyCN740zogg1Lk9X5f0tEGxsNAev14tDnZAZggRnwWxZMCUhy1s
Yd4yPDEAawtghoj0wRzqx7qFv+P3p3yZwUpN4iImLAYDAoGAcDeQB02L9Q11cty3kf4BaT8WohF3
/YgUK7j4kAs3UMGGRR1mUuGQUY34PKu2bLPaUD9qUcJlAFEusHy5bE4MsZMX58voCAGMc74C8Tog
Krk181GdKLJgGGoyJHdUCwUsyoItc7F6nibCZxG6QNhUMYNjqEz3Cpz0Pi4wTI33+Zc=
-----END RSA PRIVATE KEY-----
~
~
"ec2zdjordj0222HU.pem" 24 lines, 1758 characters
```

```
C:\>ec2-delete-keypair ec2zdjordj0222HU
```

```
C:\ec2>rm ec2zdjordj0222HU.pem -- Linux
```

```
C:\ec2>del ec2zdjordje0222HU.pem -- Windows
```

# Launch a new Instance of Instance Store AMI

- Let us find an Instance Store AMI image using AWS Console. We found `ami-74d20b69` in `sa-east-1a` availability zone.
- You have to point your `EC2_URL` to the zone, SET
- `EC2_URL=https://ec2.sa-east-1.amazonaws.com`
- Launch an instance:

```
C:\ec2> ec2-run-instances ami-74d20b69 -k ec2_hu  
--availability-zone sa-east-1a
```

```
RESERVATION      r-e67566fb          951414139794      default  
INSTANCE         i-df5cbac3         ami-74d20b69  
                  pending 0          m1.small          2012-09-  
14T20:53:15+0000      sa-east-1a aki-ca8f51d7  
monitoring-disabled instance-store  paravirtual xen  
sg-7aaa0967          default false
```

Note: **i-df5cbac3** This is the instance ID.

- You will use the instance ID to refer to that instance, as long as the instance is not terminated.

**Syntax:** `ec2-run-instances` *ami\_id* [-n *instance\_count*] [-g *group* [-g *group ...*]] [-k *keyname*] [-d *user\_data* | -f *user\_data\_file*] [ --addressing *addressing\_type*] [ --type *instance\_type*] [ --availability-zone *zone*] [ --kernel *kernel\_id*] [ --ramdisk *ramdisk\_id*] [ --block-device-  
mapping *block device mapping*]

# Find out instance status

```
C:\ec2> ec2-describe-instances i-df5cbac3
```

```
RESERVATION      r-e67566fb      951414139794      default
INSTANCE         i-df5cbac3      ami-74d20b69      ec2-177-71-225-
198.sa-east-1.com falsepute.amazonaws.com      ip-10-248-3-
107.sa-east-1.compute.internal running 0 m1.small 2012-09-
14T20:53:15+0000 sa-east-1a      aki-ca8f51d7
monitoring-disabled      177.71.225.198 10.248.3.107
instance-store paravirtual      xen sg-7aaa0967
default
```

- Since we forgot to assign a particular security group to the new instance, it is assigned the DEFAULT security group. DEFAULT sec group might not be what we need.
- We should have used an existing security group, `lamp-basic`, for example.

```
C:\ec2> ec2-run-instances ami-2cb05345 -k ec2_hu  
--availability-zone sa-east-1a -group lamp-basic
```

# Add Authorization to the Security Group

- **Our instance has security group DEFAULT. So we will change ports of the DEFAULT security group**

```
C:\ec2> ec2-authorize default -p 80
```

```
GROUP default PERMISSION default ALLOWS tcp 80
      80 FROM CIDR 0.0.0.0/0 ingress
```

Alternatively:

```
Client.InvalidPermission.Duplicate: The permission
'0.0.0.0/0' has already been granted to the specified
group
```

```
C:\ec2> ec2-authorize default -p 22 -s 140.247.251.141/32
```

```
GROUP default
PERMISSION default ALLOWS tcp 22 22 FROM CIDR
                          140.247.251.141/32
```

- Permission for access to port 22 is added. On you Cygwin or Linux prompt you could invoke the ssh command, now:

```
$ ssh -i ec2_hu.pem
      root@ec2-184-73-12-147.compute-1.amazonaws.com
```

- Will work.

# Copy X509 Credentials to the Instance

- At the later stage, in the process called bundling, we will need our X509 credentials in `/mnt` directory of the new instance.
- Let us **secure copy** those files there. On Cygwin or Linux prompt on a single line we type:

```
$ scp -i ec2_hu.pem
pk-JYJA2NNBBIKPNFA5CA6PCQFTARPMH5D.pem
cert-JYJA2NNBBIKPNFA5CA6PCQFTARPMH5D.pem
ec2-user@ec2-184-73-12-147.compute-1.amazonaws.com: /mnt
pk-JYJA2NNBBIKPNFA5CA6PCQFTARPMH5D.pem      100%  926      0.9KB/s   00:00
cert-JYJA2NNBBIKPNFA5CA6PCQFTARPMH5D.pem    100%  916      0.9KB/s   00:00
```

```
Zoran@LENOVOZ /cygdrive/c/ec2
```

```
$ ssh -i ec2_hu.pem
root@ec2-184-73-12-147.compute-1.amazonaws.com
```

```
[root@ip-10-250-7-143 ~]# cd /mnt
[root@ip-10-250-7-143 mnt]# ls
cert-JYJA2NNBBIKPNFA5CA6PCQFTARPMH5D.pem
lost+found
pk-JYJA2NNBBIKPNFA5CA6PCQFTARPMH5D.pem
```