Amazon EC2

Amazon Elastic Compute Cloud (EC2)

- Amazon Machine Images (**AMIs**) are the basic building blocks of Amazon EC2
- An AMI is a template that contains a software configuration (operating system, application server and applications) that can run on Amazon's computing environment
- AMIs can be used to launch an ***instance***, which is a copy of the AMI running as a virtual server in the cloud.

Getting Started with Amazon EC2

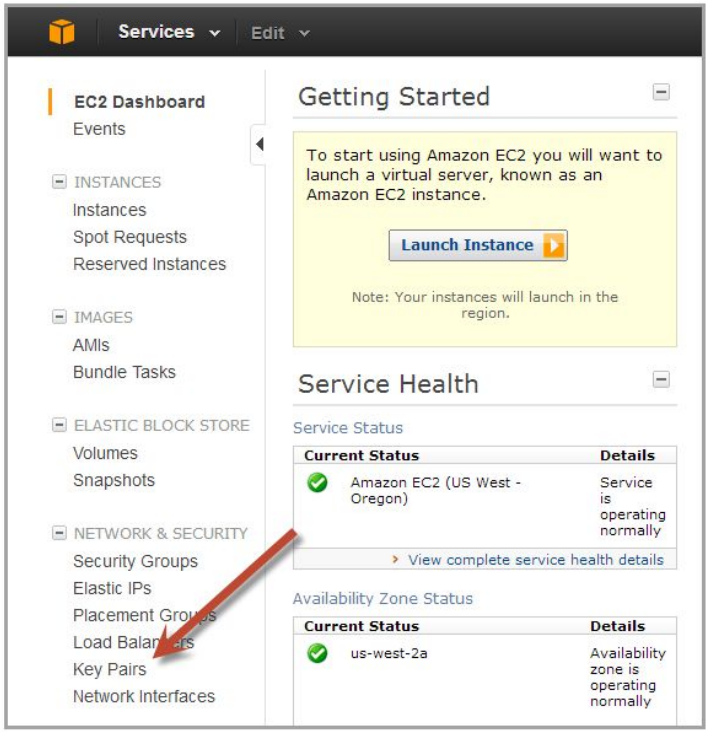
- Step 1: Sign up for Amazon EC2
- Step 2: Create a key pair
- Step 3: Launch an Amazon EC2 instance
- Step 4: Connect to the instance
- Step 5: Customize the instance
- Step 6: Terminate instance and delete the volume created

Creating a key pair

- AWS uses public-key cryptography to encrypt and decrypt login information.
- AWS only stores the public key, and the user stores the private key.
- There are two options for creating a key pair:
  - Have Amazon EC2 generate it for you
  - Generate it yourself using a third-party tool such as OpenSSH, then import the public key to Amazon EC2

Generating a key pair with Amazon EC2

1. Open the Amazon EC2 console at http://console.aws.amazon.com/ec2/
2. On the navigation bar select region for the key pair
3. Click **Key Pairs** in the navigation pane to display the list of key pairs associated with the account



Generating a key pair with EC2 (cont.)

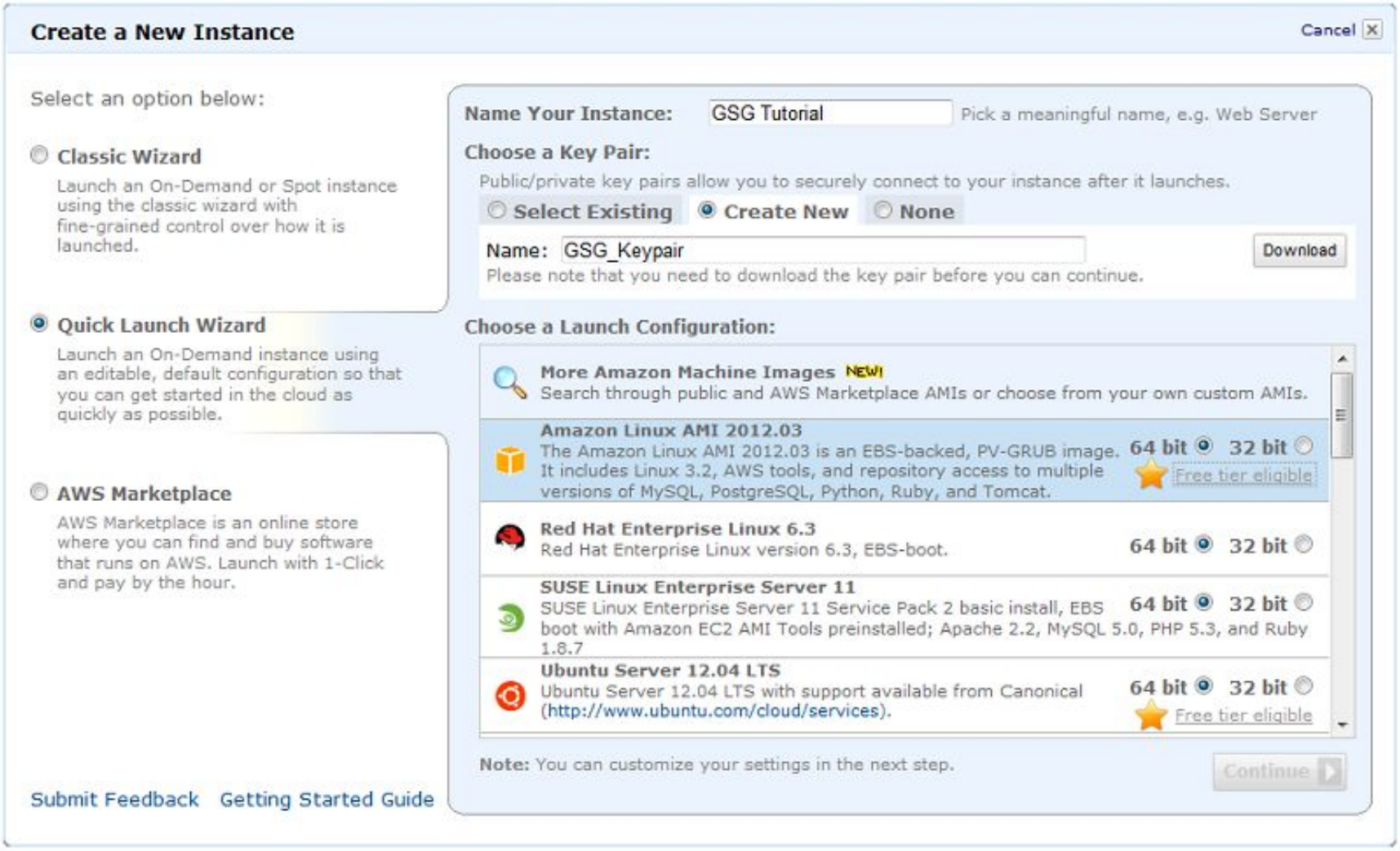
4. Click **Create Key Pair**
5. Enter a name for the key pair in the **Key Pair Name** field of the dialog box and click **Create**
6. The private key file, with .pem extension, will automatically be downloaded by the browser.

Launching an Amazon EC2 instance

1. Sign in to AWS Management Console and open the Amazon EC2 console at http://console.aws.amazon.com/ec2/
2. From the navigation bar select the region for the instance

Launching an Amazon EC2 instance (cont.)

3. From the Amazon EC2 console dashboard, click **Launch Instance**

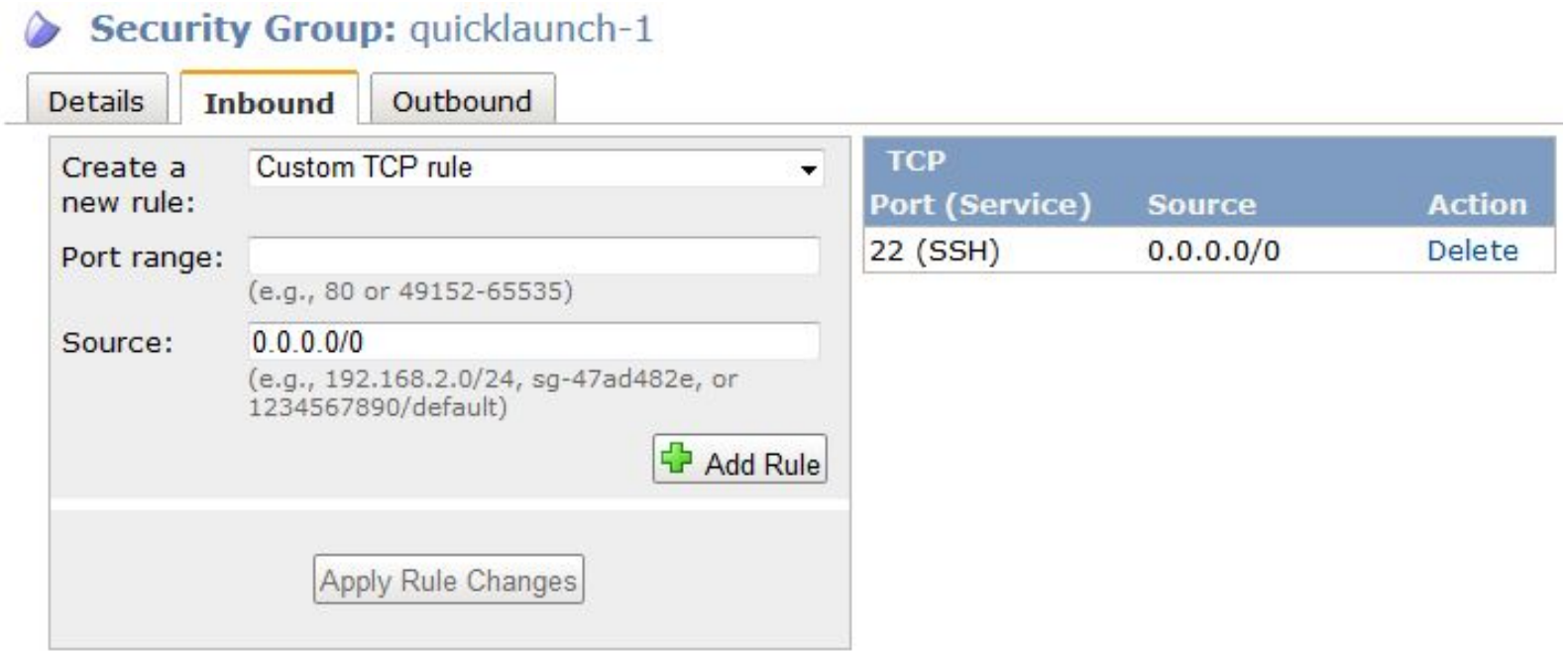


Launching an Amazon EC2 instance (cont.)

4. On the **Create a New Instance** page, click **Quick Launch Wizard**
5. In **Name Your Instance**, enter a name for the instance
6. In **Choose a Key Pair**, choose an existing key pair, or create a new one
7. In Choose a Launch Configuration, a list of basic machine configurations are displayed, from which an instance can be launched
8. Click continue to view and customize the settings for the instance

Launching an Amazon EC2 instance (cont.)

9. Select a security group for the instance. A **Security Group** defines the firewall rules specifying the incoming network traffic delivered to the instance. Security groups can be defined on the Amazon EC2 console, in **Security Groups** under **Network and Security**



Launching an Amazon EC2 instance (cont.)

10. Review settings and click **Launch** to launch the instance
11. Close the confirmation page to return to EC2 console
12. Click **Instances** in the navigation pane to view the status of the instance. The status is **pending** while the instance is launching

	Name	Instance	AMI ID	Root Device	Type	State	Public DNS
	GSG Tutorial	i-e1ab569a	ami-aecd60c7	ebs	t1.micro	pending	

After the instance is launched, its status changes to **running**

	Name	Instance	AMI ID	Root Device	Type	State	Public DNS
	GSG Tutorial	i-e1ab569a	ami-aecd60c7	ebs	t1.micro	running	ec2-50-19-54-72.compute-1.amazonaws.com

Connecting to an Amazon EC2 instance

- There are several ways to connect to an EC2 instance once it's launched.

- **Remote Desktop Connection** is the standard way to connect to Windows instances.

- An **SSH client** (standalone or web-based) is used to connect to Linux instances.

Connecting to Linux/UNIX Instances from Linux/UNIX with SSH

Prerequisites:

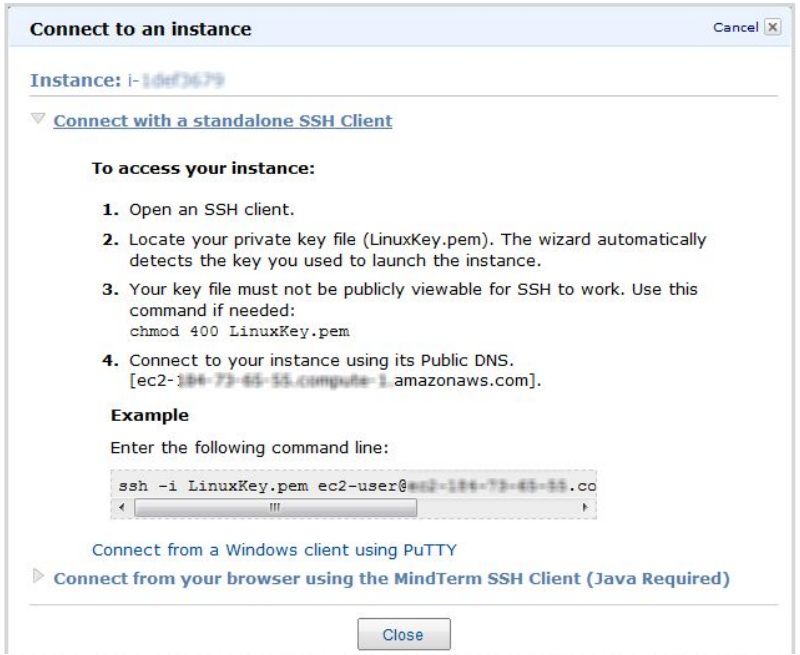
- Most Linux/UNIX computers include an SSH client by default, if not it can be downloaded from openssh.org
- Enable SSH traffic on the instance (using security groups)
- Get the path the private key used when launching the instance

1. In a command line shell, change directory to the path of the private key file
2. Use the **chmod** command to make sure the private key file isn't publicly viewable

```
chmod 400 My_Keypair.pem
```

Connecting to Linux/UNIX Instances(cont.)

3. Right click on the instance to connect to on the AWS console, and click **Connect**.
4. Click **Connect using a standalone SSH client.**
5. Enter the example command provided in the Amazon EC2 console at the command line shell




```
ssh -i <your key a name>.pem ec2-user@ec2-184-72-204-112.compute-1.amazonaws.com
```

Transfering files to Linux/UNIX instances from Linux/UNIX with SCP

Prerequisites:

- Enable SSH traffic on the instance
- Install an SCP client (included by default mostly)
- Get the ID of the Amazon EC2 instance, public DNS of the instance, and the path to the private key

If the key file is My_Keypair.pem, the file to transfer is samplefile.txt, and the instance's DNS name is ec2-184-72-204-112.compute-1.amazonaws.com, the command below copies the file to the ec2-user home

```
scp -i My_Keypair.pem samplefile.txt ec2-user@ec2-184-72-204-112.compute-1.amazonaws.com:~
```

Terminating Instances

- If the instance launched is not in the free usage tier, as soon as the instance starts to boot, the user is billed for each hour the instance keeps running.
- A terminated instance cannot be restarted.
- To terminate an instance:
  1. Open the Amazon EC2 console
  2. In the navigation pane, click **Instances**
  3. Right-click the instance, then click **Terminate**
  4. Click **Yes, Terminate** when prompted for confirmation

Google App Engine Quick Start

adapted from
https://developers.google.com/appengine/docs/whatisgoogleappengine

Google App Engine (GAE)

- GAE lets users run web applications on Google's infrastructure
- GAE data storage options are:
  - Datastore: a NoSQL schemaless object datastore
  - Google Cloud SQL: Relational SQL database service
  - Google Cloud Storage: Storage service for objects and files
- All applications on GAE can use up to 1 GB of storage and enough CPU and bandwidth to support an efficient application serving around 5 million page views a month for free.
- Three runtime environments are supported: **Java**, **Python** and **Go**.

Developing Java Applications on GAE

- The easiest way to develop Java applications for GAE is to use the Eclipse development environment with the Google plugin for Eclipse.
- App Engine Java applications use the Java Servlet standard for interacting with the web server environment.
- An application's files, including compiled classes, JARs, static files and configuration files, are arranged in a directory structure using the WAR standard layout for Java web applications.

Running a Java Project

- The App Engine SDK includes a web server application to test applications. The server simulates the complete GAE environment.
- The project can be run using the "**Debug As > Web Application**" option of Eclipse or using Ant.
- After running the server, the application can be tested by visiting the server's URL in a Web browser.

Uploading an Application to GAE

- Applications are created and managed using the Administration Console at https://appengine.google.com.
- Once an application ID is registered for an application, the application can be uploaded to GAE using the Eclipse plugin or a command-line tool in the SDK.
- After uploading, the application can be accessed from a Web browser. If a free appspot.com account was used for registration, the URL for the application will be http://app_id.appspot.com/, where app_id the application id assigned during registration.