Basic
Amazon Web Service (AWS)
Lab 06

CSCIE 63 Big Data Analytics

Use Cases and Solutions, Application Hosting

- AWS, MS Azure, Google, and other offer a large number of products (services) that cater to many IT needs.
- Those services give users, i.e. application vendors, an easy way to use the massive computing power of the Cloud to host their programs as well as commercial software on the Internet.
- The following slide illustrate the multitude of those services offered by Amazon AWS.
- Other vendors offer similar services, typically somewhat fewer in numbers.

Amazon Web Services AWS Services

Compute

- **EC2** — Virtual Servers in the Cloud
- **EC2 Container Service** — Run and Manage Docker Containers
- **Elastic Beanstalk** — Run and Manage Web Apps
- **Lambda** — Run Code in Response to Events

Storage & Content Delivery

- **S3** — Scalable Storage in the Cloud
- **CloudFront** — Global Content Delivery Network
- **Elastic File System** PREVIEW — Fully Managed File System for EC2
- **Glacier** — Archive Storage in the Cloud
- **Storage Gateway** — Integrates On-Premises IT Environments with Cloud Storage

Database

- **RDS** — MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
- **DynamoDB** — Predictable and Scalable NoSQL Data Store
- **ElastiCache** — In-Memory Cache
- **Redshift** — Managed Petabyte-Scale Data Warehouse Service

Networking

- **VPC** — Isolated Cloud Resources
- **Direct Connect** — Dedicated Network Connection to AWS
- **Route 53** — Scalable DNS and Domain Name Registration

Developer Tools

- **CodeCommit** — Store Code in Private Git Repositories
- **CodeDeploy** — Automate Code Deployments
- **CodePipeline** — Release Software using Continuous Delivery

Management Tools

- **CloudWatch** — Monitor Resources and Applications
- **CloudFormation** — Create and Manage Resources with Templates
- **CloudTrail** — Track User Activity and API Usage
- **Config** — Track Resource Inventory and Changes
- **OpsWorks** — Automate Operations with Chef
- **Service Catalog** — Create and Use Standardized Products

Security & Identity

- **Identity & Access Management** — Manage User Access and Encryption Keys
- **Directory Service** — Host and Manage Active Directory
- **Trusted Advisor** — Optimize Performance and Security

Analytics

- **EMR** — Managed Hadoop Framework
- **Data Pipeline** — Orchestration for Data-Driven Workflows
- **Kinesis** — Real-time Processing of Streaming Big Data
- **Machine Learning** — Build Smart Applications Quickly and Easily

Mobile Services

- **Cognito** — User Identity and App Data Synchronization
- **Device Farm** — Test Android, Fire OS, and iOS apps on real devices in the Cloud
- **Mobile Analytics** — Collect, View and Export App Analytics
- **SNS** — Push Notification Service

Application Services

- **API Gateway** — Build, Deploy and Manage APIs
- **AppStream** — Low Latency Application Streaming
- **CloudSearch** — Managed Search Service
- **Elastic Transcoder** — Easy-to-use Scalable Media Transcoding
- **SES** — Email Sending Service
- **SQS** — Message Queue Service
- **SWF** — Workflow Service for Coordinating Application Components

Enterprise Applications

- **WorkSpaces** — Desktops in the Cloud
- **WorkDocs** — Secure Enterprise Storage and Sharing Service
- **WorkMail** PREVIEW — Secure Email and Calendaring Service

EC2, Elastic Cloud Computing

- Perhaps the most important AWS service is EC2, Elastic Cloud Computing service, which lets you rent Amazon AMI-s (Amazon Machine Images) per hour and in unlimited number.
- A typical AMI is XEN image (a Virtual Machine), pre-configured and available on demand from AWS
  - Users can customize AMIs and then store them for reuse
  - Users can allocate storage and mount it to AMI-s as needed
    - The same storage device could be mounted to different AMI's

Simple Storage Service, S3

- Even as storage becomes more plentiful and affordable, businesses are still faced with the task of managing their growing storage infrastructure.
- Amazon Web Services provides a cost-effective solution for storing information in the cloud that eliminates the burden of provisioning and managing hardware.
- Amazon Simple Storage Service (Amazon S3) is a highly scalable, reliable, and inexpensive data storage infrastructure that enables dependable backup solutions.
- Hundreds of thousands of customers use Amazon S3 as their backup location, and other customers have created compelling end-user backup, storage, and disaster recovery solutions using AWS

Amazon Simple Storage Service (S3)

- S3 is the most basic storage service
- Flat storage model consisting of buckets and objects
  - Bucket – has a name and contains objects
    - Bucket name has to be globally unique
  - Objects – has a key, stores 1 byte - 5GB
  - Object key can look like a path
- Cost:
  - $0.125/GB-Month
  - $0.10-0.18/GB of data transferred
  - Data transfers between EC2 and S3 are free of bandwidth charges
- Buckets and objects can be:
  - Public – accessible by anyone
  - Private – accessible to owner, ACL member

Some EC2 Instance types

* 1 EC2 Compute Unit is equivalent to a 1.0-1.2 GHz Opteron or Xeon processor

Type	CPU*	Memory	Storage	Platform
Small (default)	1 EC2 Compute Unit	1.7 GB	160 GB	32-bit
Large	4 EC2 Compute Units	7.5 GB	850 GB	64-bit
Extra large	8 EC2 Compute Units	15 GB	1690 GB	64-bit
High CPU Medium	5 EC2 Compute Units	1.7 GB	350 GB	32-bit
High CPU Extra large	30 EC2 Compute Units	222 GB	1690 GB	64-bit

Dual IPs and Security Groups

- All AMIs come with public IP addresses
  - Security groups allow you to configure which ports to open up for external inbound access
  - Each entry in a Security Group has a protocol, port number and optionally a subnet mask to restrict access by IP address
  - A Security Group must be allocated to an AMI before it is started
    - You can add/delete entries to a security group allocated to one or several AMI-s with out having to restart the AMI. Modifications to the security group affect running AMI-s
- AMIs have one Private IP address that can be used for access by other machines in the AWS Cloud.

Volatile vs. Elastic Block Storage

There are 2 types of storage and to types of machine:

1) Ephemeral (Volatile) storage: when an AMI is terminated all local data is lost. Machines with volatile storage are called Instance Store AMI-s
2) Elastic Block Store and EBS AMI-s
   - Mountable storage volume, like an attachable/detachable drive
   - “SAN on demand”
   - From 1Gb to 1Tb
   - Significantly faster than AMI volatile storage
   - Mounts on a single AMI only
     - Can mount several on a single AMI
   - Cost
     - $0.10/Gb per month (based on size of volume not amount used)
     - $0.10/1 million I/O requests

Elastic IP Addresses

- AMI IP addresses are allocated on startup
  - Most likely will get a unique IP every time
    - Big inconvenience for public sites
- Elastic IP addresses:
  - Statically allocated addresses
  - Associated with your account
  - Attached to an instance (e.g. public facing web server)
    - Can attach an Elastic IP address to another instance if you need to switch AMIs
  - You configure DNS to resolve to the elastic IP address
  - No cost for Elastic IP addresses while in use
    - * $0.01 per non-attached Elastic IP address per complete hour
    - * $0.00 per Elastic IP address remap – first 100 remaps / month
    - * $0.10 per Elastic IP address remap – additional remap / month over 100

Regions and Availability Zones

- In a failover DB server configuration your standby server could run on the same physical machine
- Regions:
  - Geographically dispersed locations
  - Currently there are 10 regions
  - Work in the region where your live. Lower latency
- Availability zone:
  - Part of a region
  - Engineered to be insulated from failure in
  - other zones
- Specify availability zone when launching instances:
  - Same zone for all instances in your cluster for fast data transfer
-

Getting started with Amazon Web Services

- Signing up is easy
  - Credit card required but no charge for signing up
- 4 important pieces of information that you will need post sign up
  - Account Id
  - Access Id
  - Secret key
  - Private key and certificate files
  - You need a phone to confirm your identity.

Information that you need to keep after signing up

- Various combinations of security information generated during sign up are needed for the EC2 API, S3 and other Tools
  - Account Number
    - Currently a 12 digit number
  - Access Id and Secret key
    - Long Character strings
  - Private key and certificate
    - Long character strings that need to be downloaded and stored as files
- This information should be closely guarded.
- Someone could burn real money and create serious bills on your Visa account.

One time setup required prior to working with IBM AMIs

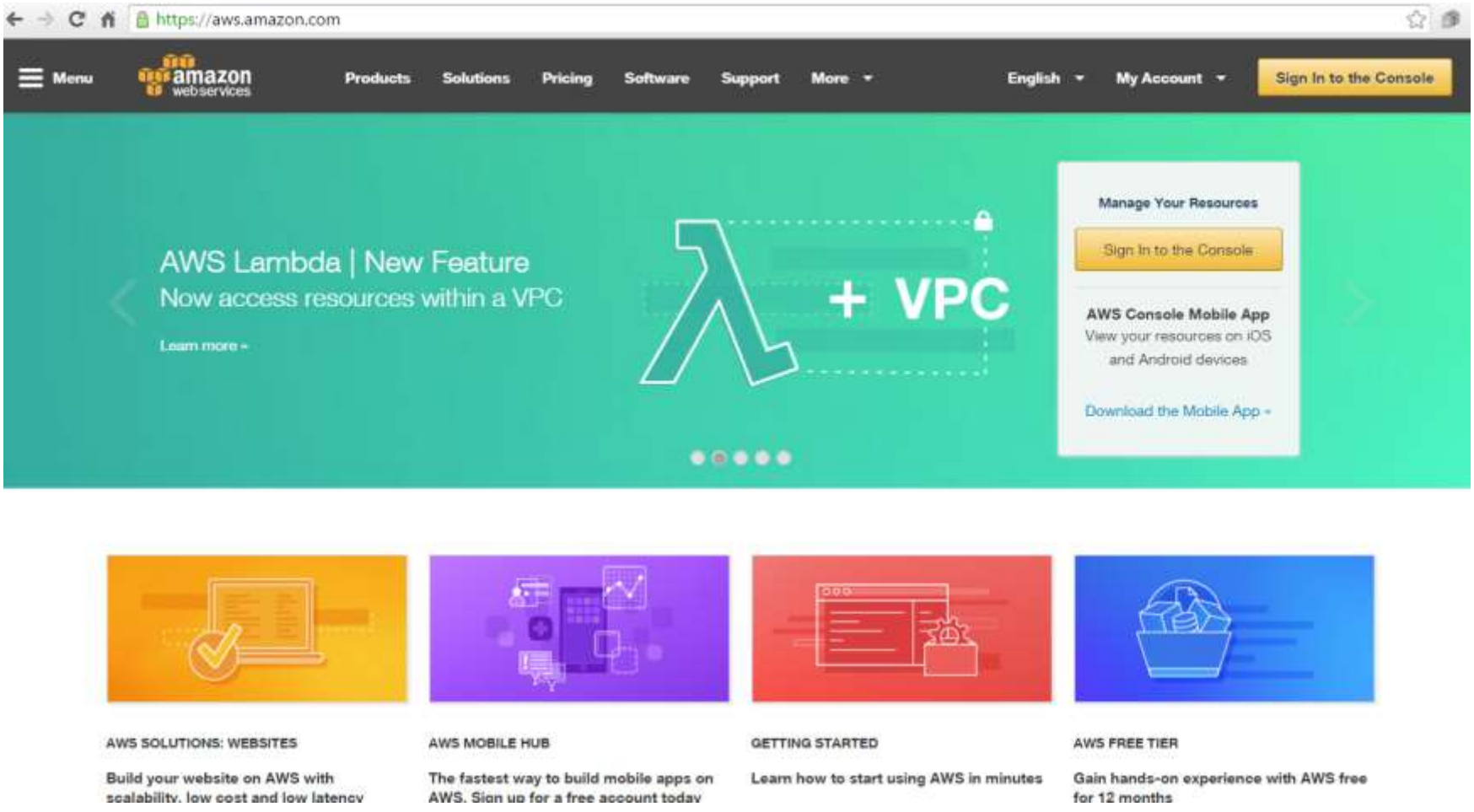
- Generate a key pair
  - Generated the private key and certificate that are saved after signing up
  - Associated with the AMIs that you start and are needed for remote access via SSH to those AMIs

AMI Usage Flow

- Register for AWS (registration is free but Credit Card is required)
- Generate key pair private key and certificate
- Create Security Group(s)
- Activate one or more AMIs
- Configure remote access
  - SSH is preferred method
- Upload any code or data you require
- Customize image
- Allocate permanent storage
  - Volatile storage goes away when the AMI stops running
- Save changes to image
  - Storage costs incurred for saving images and running them

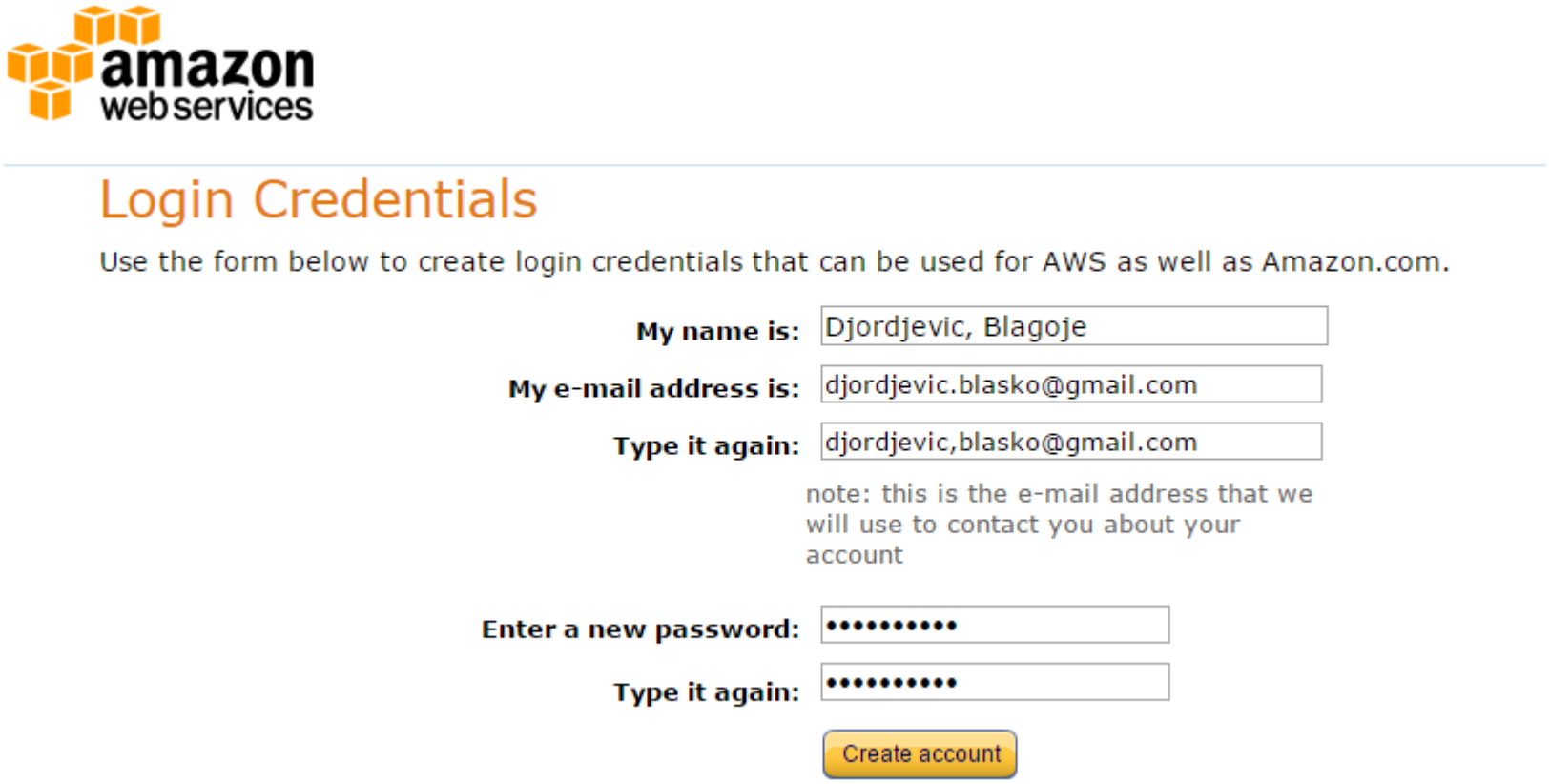
AWS Home, Select

Sign In to the Console



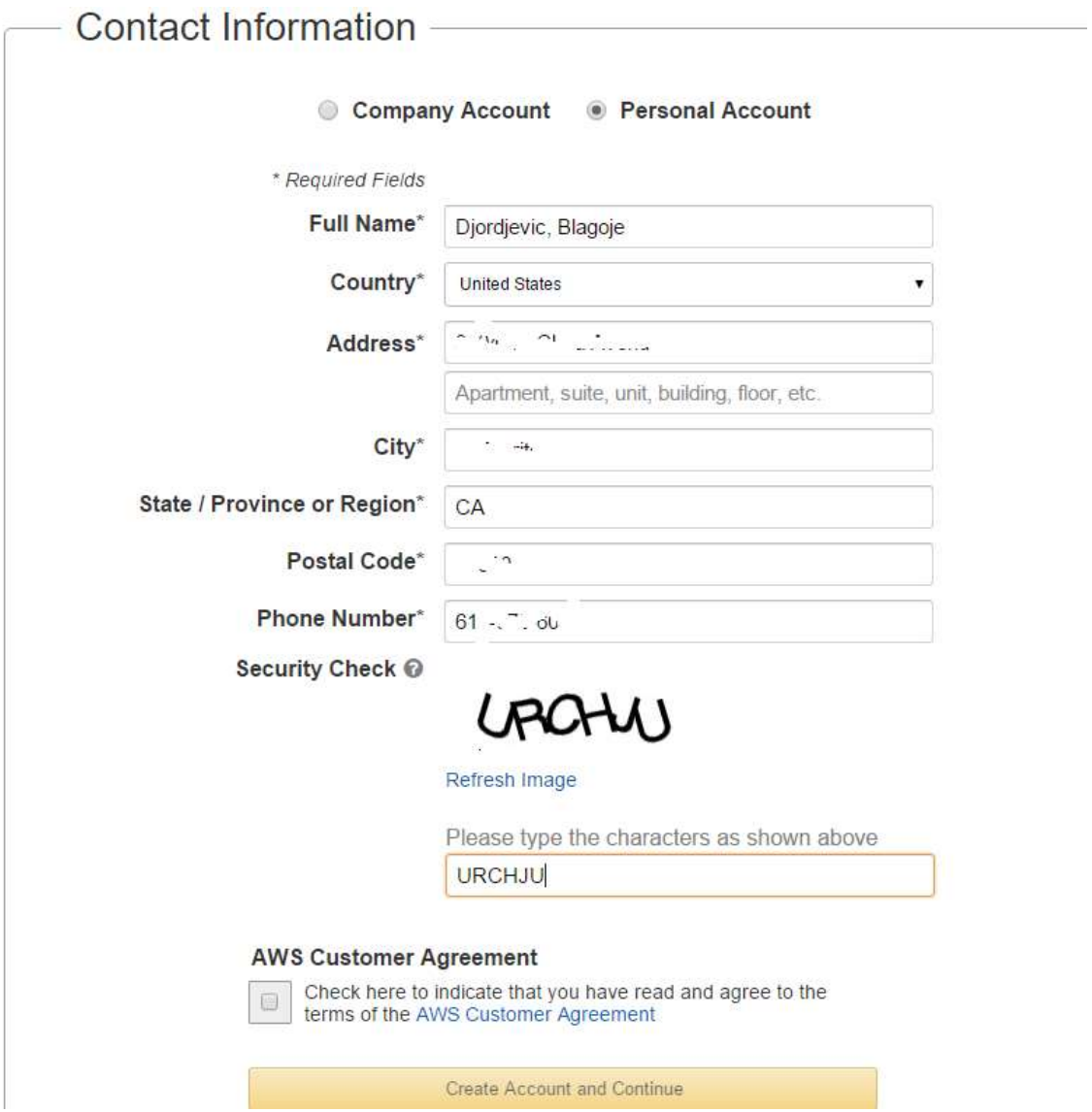
Select “I am a new user”

Provide Name, Create Password

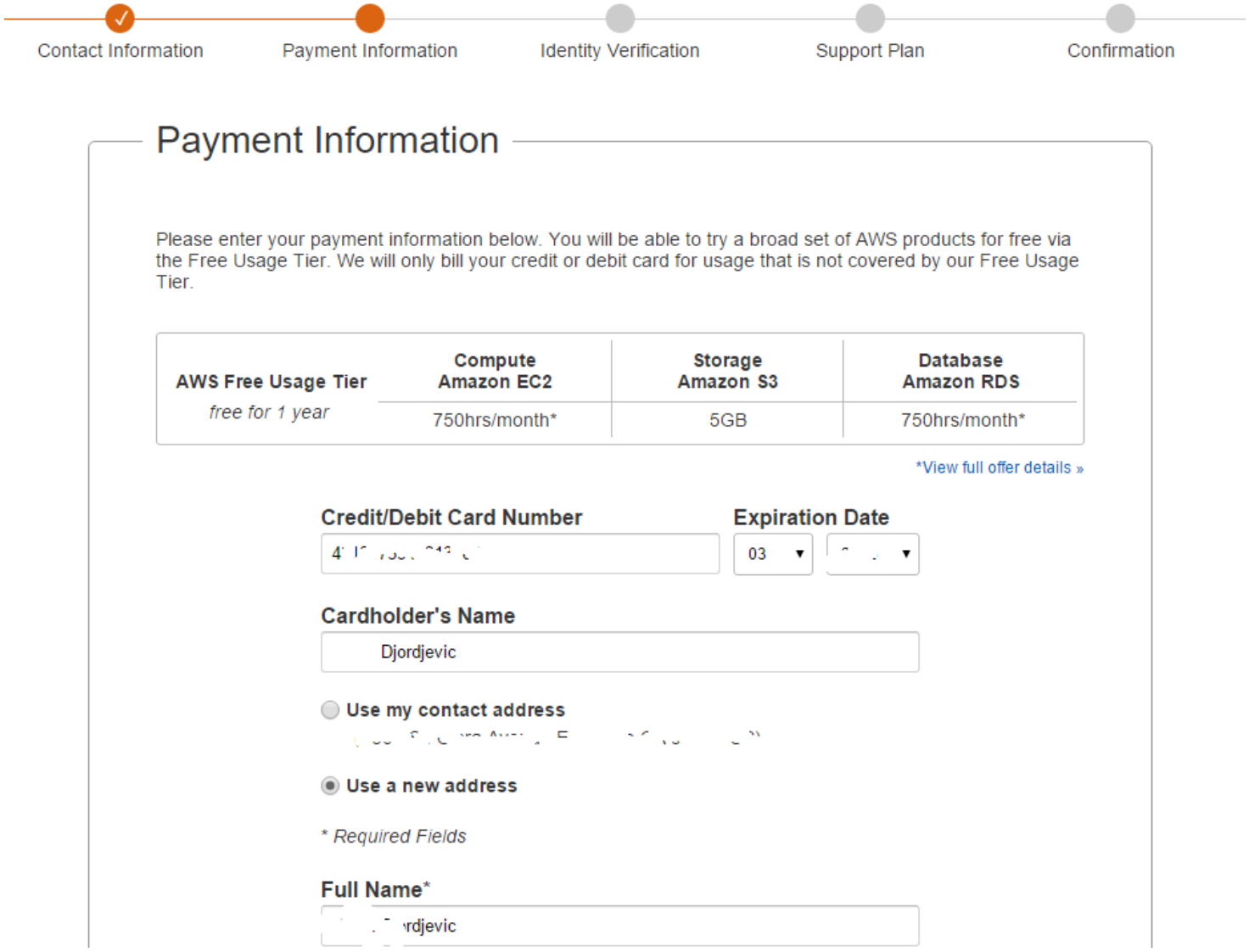


- On the following screens you will be asked for your address and phone, the credit card and phone verification. Be positive, answer the phone. You are in.

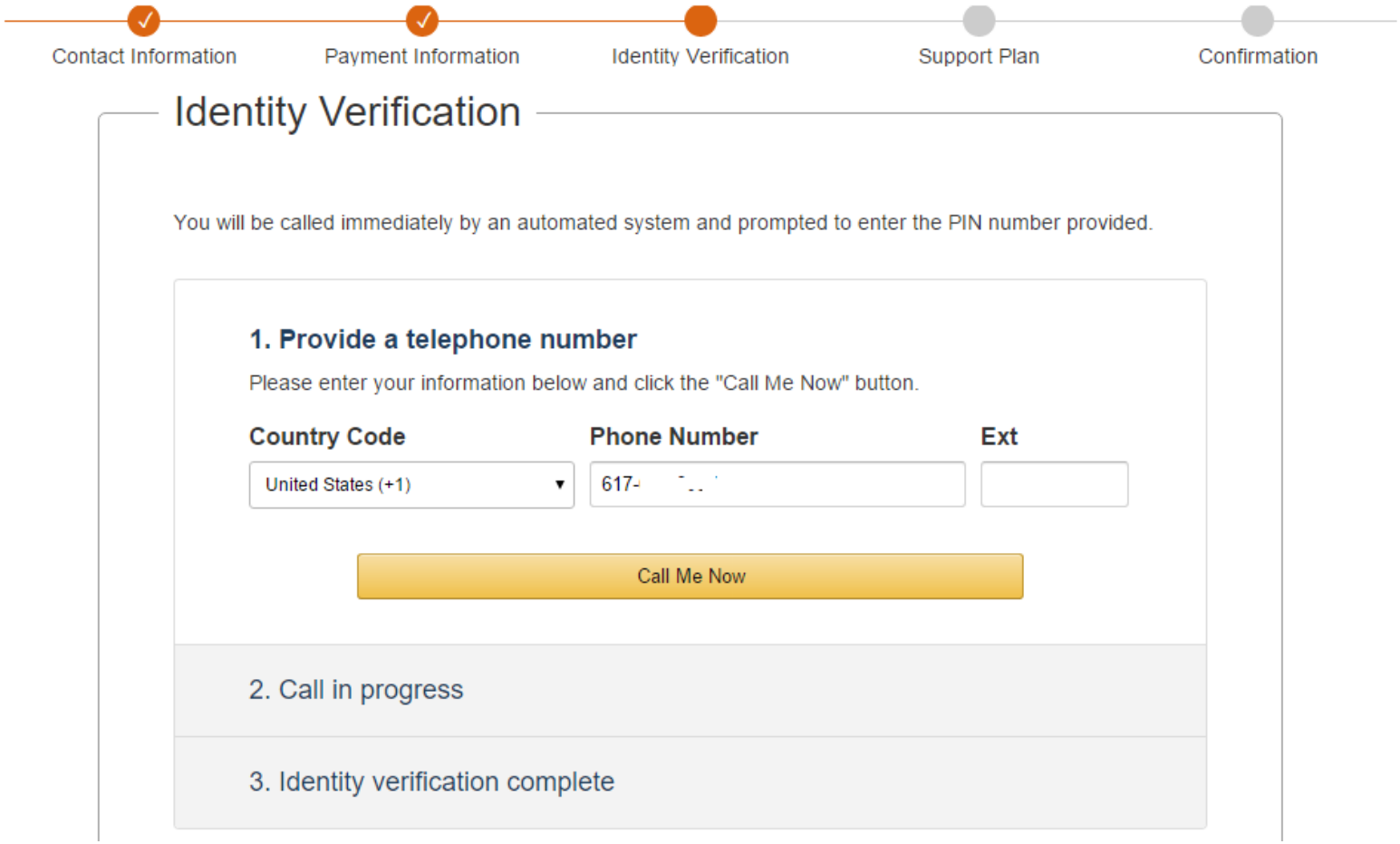
Enter Contact Information, Create Account and …



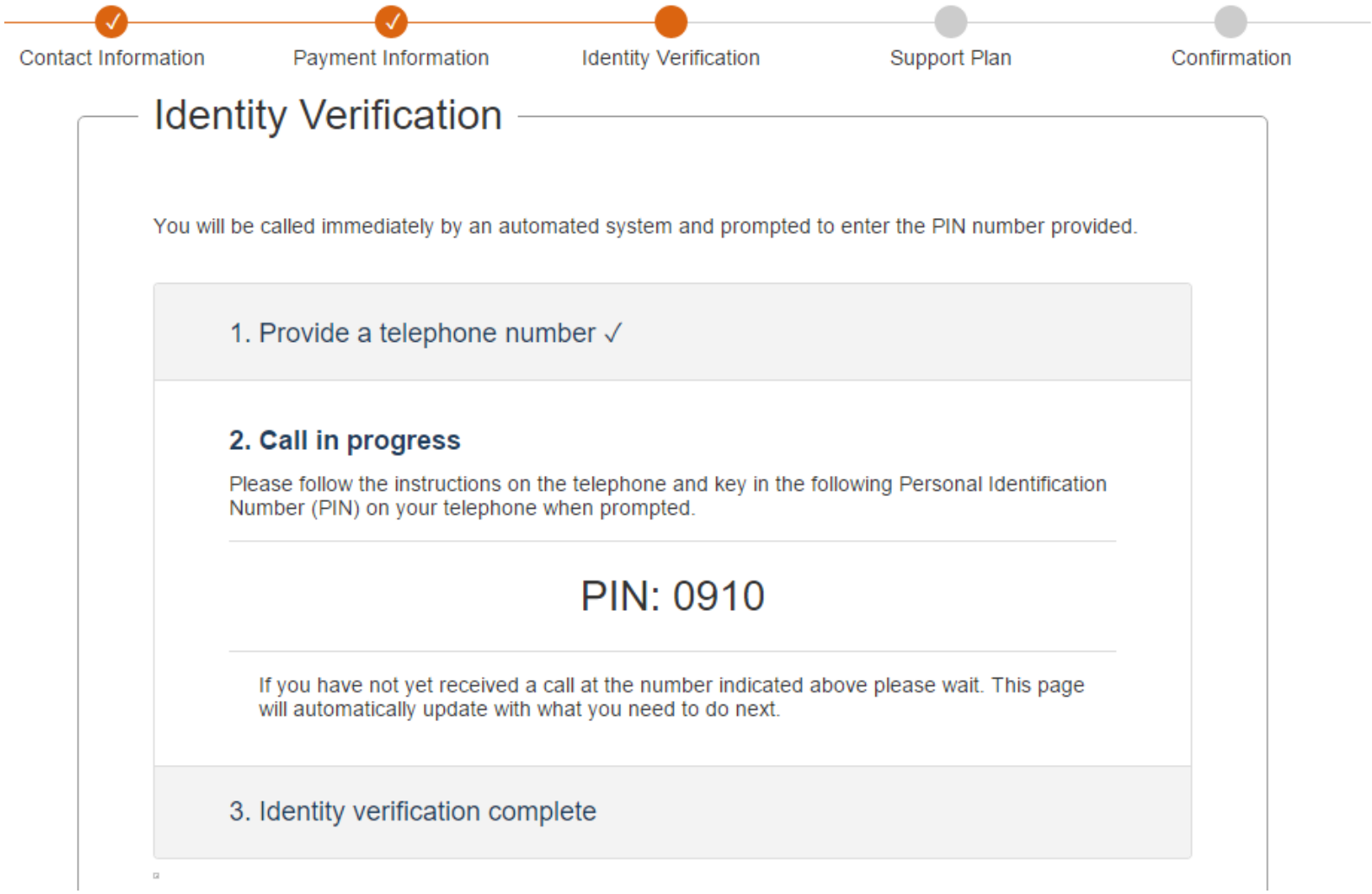
Enter Payment Information



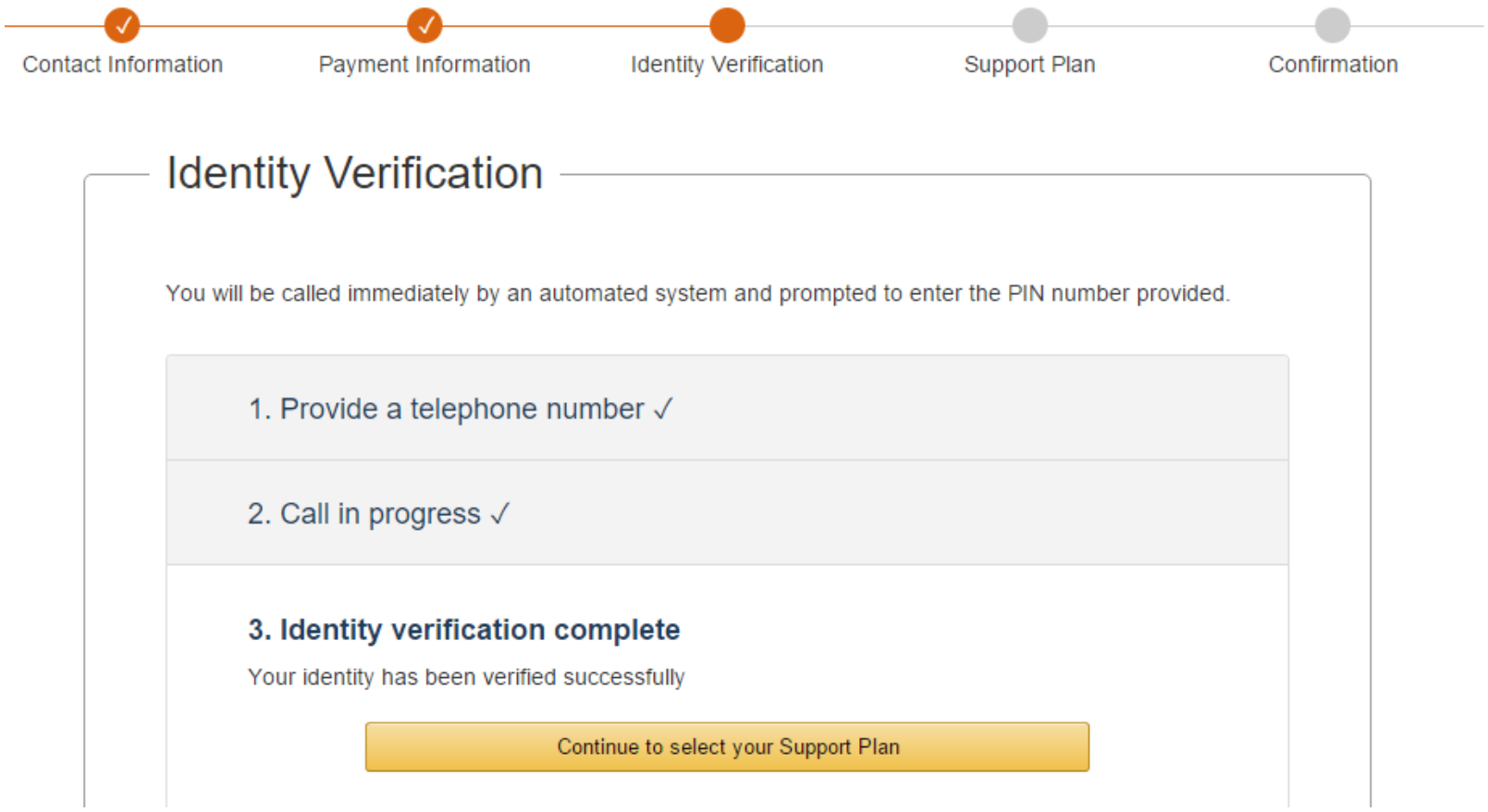
Identity Verification



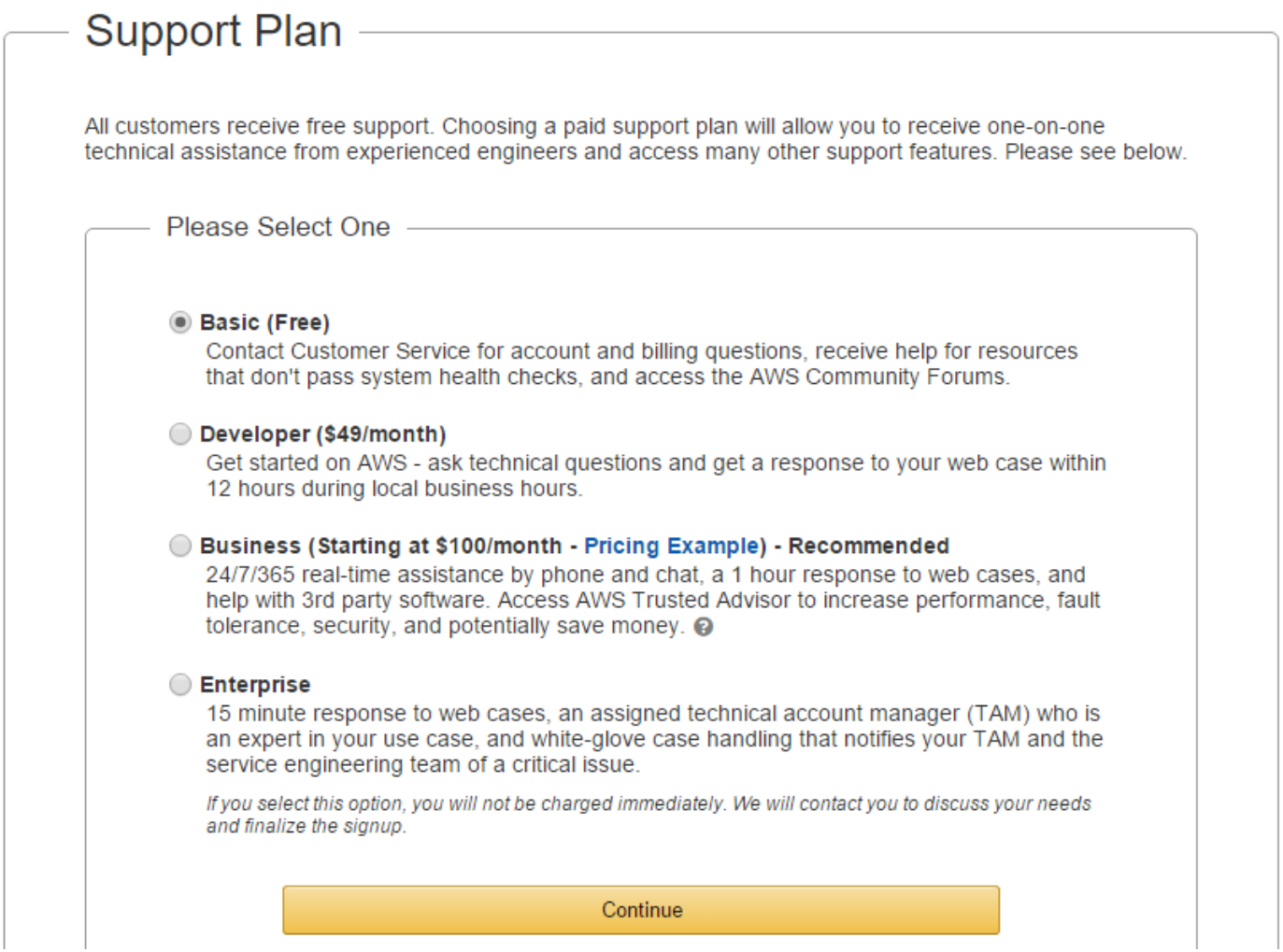
Enter PIN into your Phone Pad



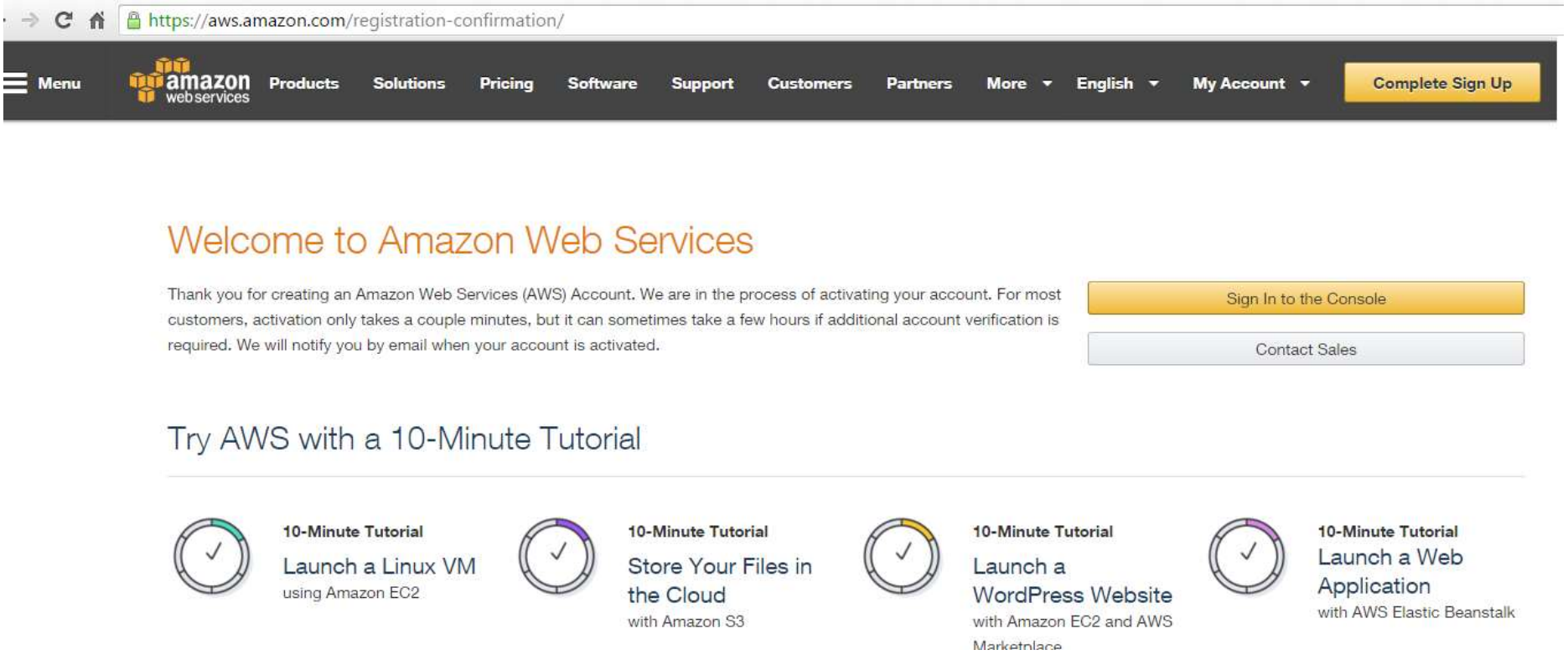
Continue to Select your Support Plan



Select Support Plan



Complete Sign Up



Sign in to the AWS Console

amazon web services

Sign In or Create an AWS Account

What is your e-mail or mobile number?

E-mail or mobile number:

djordjevic.blasko@gmail.com

I am a new user.

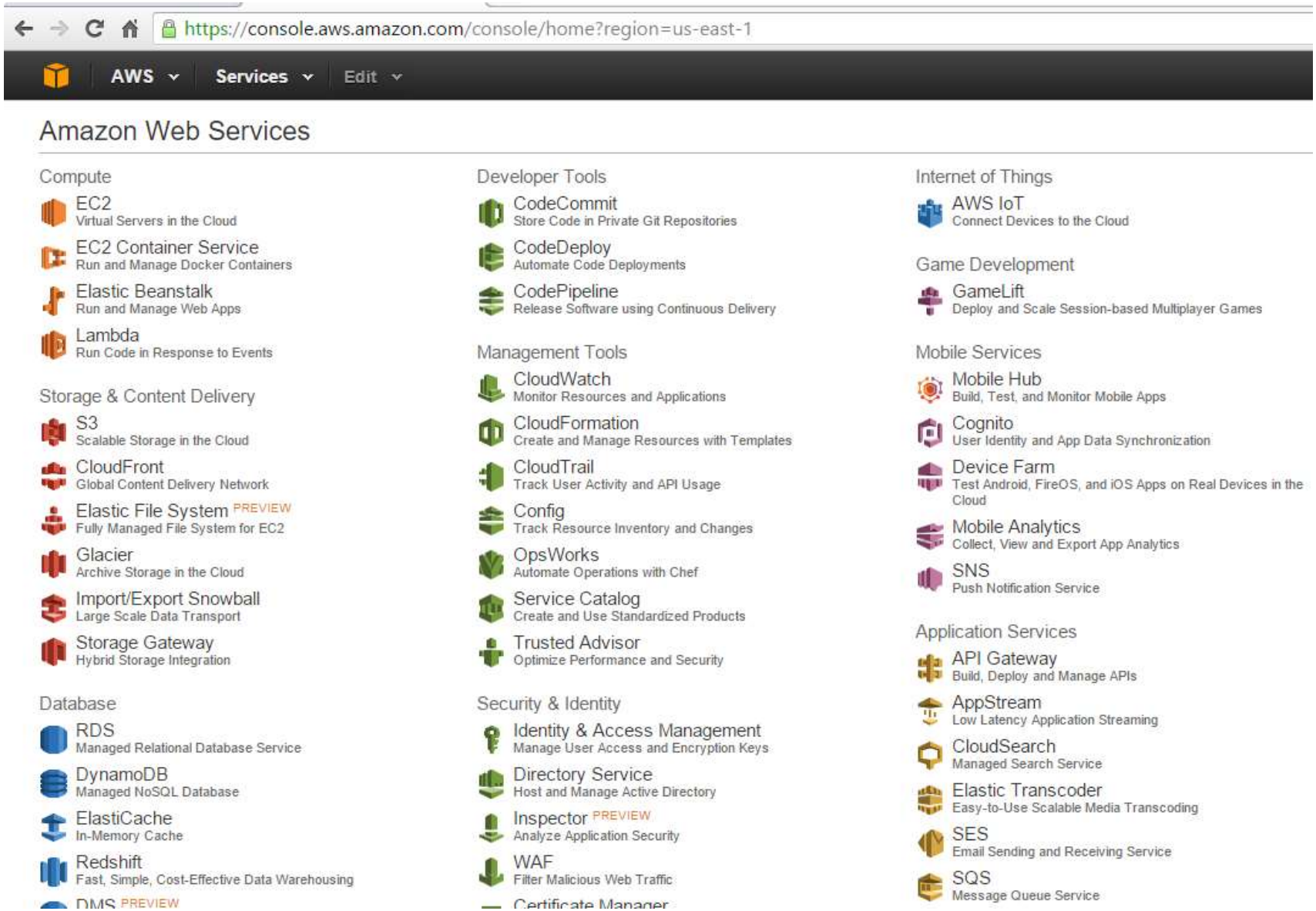
I am a returning user and my password is:

••••••••••

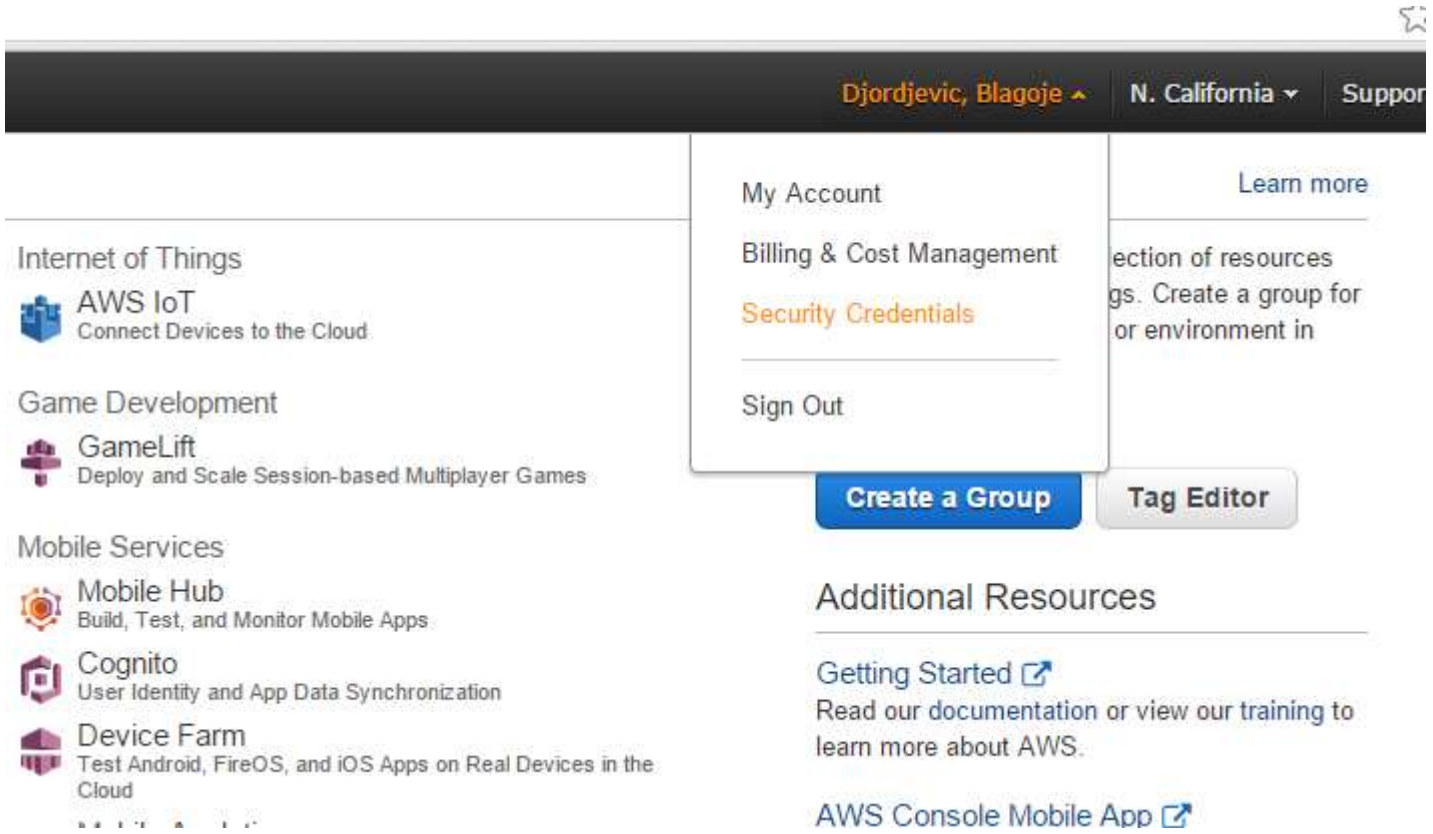
Sign in using our secure server

Forgot your password?

You will see a page with all AWS Services



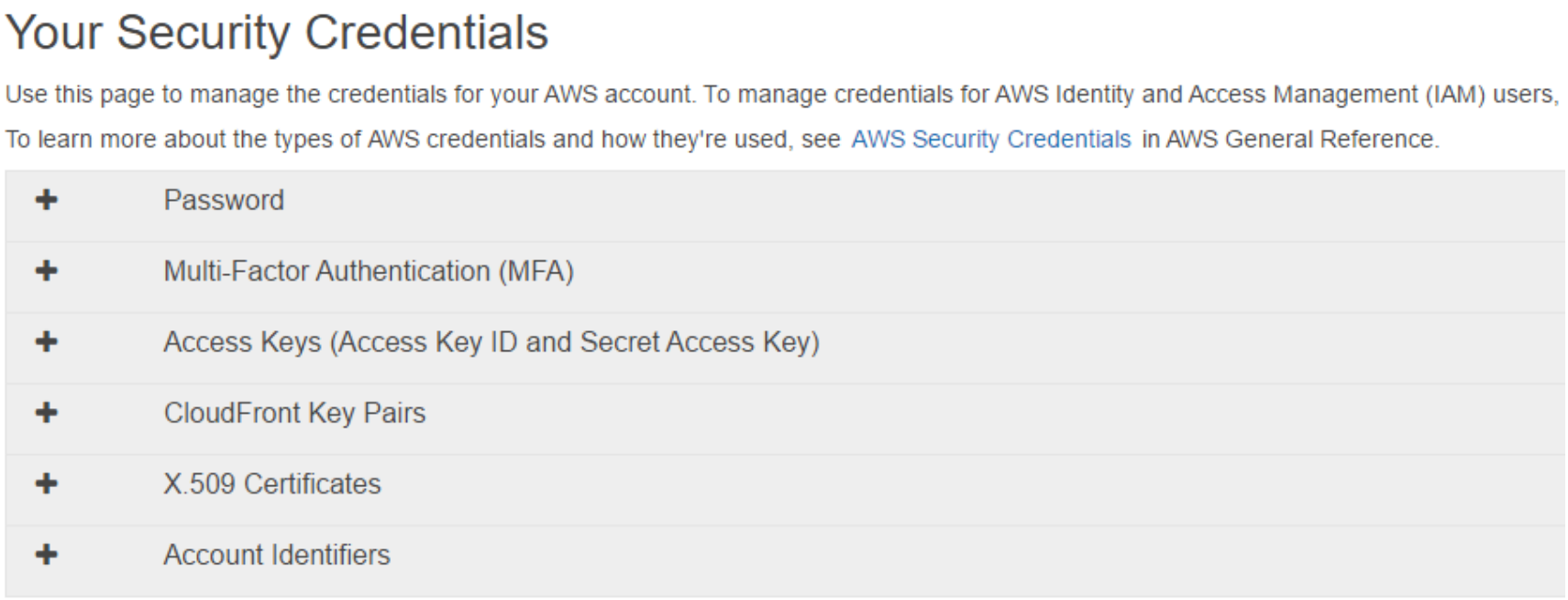
After Signing in, go to Security Credentials



On the next screen again select Continue to Security Credentials

Security Credentials

- You already have password. Other credentials you need to create.
- Multi-Factor Authentication and CloudFront Key Pairs can be ignored for now.
- Select Access Keys:



Hit Create New Access Key

Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

+ Password
+ Multi-Factor Authentication (MFA)
− Access Keys (Access Key ID and Secret Access Key)

You use access keys to sign programmatic requests to AWS services. To learn how to sign requests using your access keys, see the signing documentation. For your prote
keys securely and do not share them. In addition, AWS recommends that you rotate your access keys every 90 days.

Note: You can have a maximum of two access keys (active or inactive) at a time.

Created	Deleted	Access Key ID	Last Used	Last Used Region	Last Used Service	Status

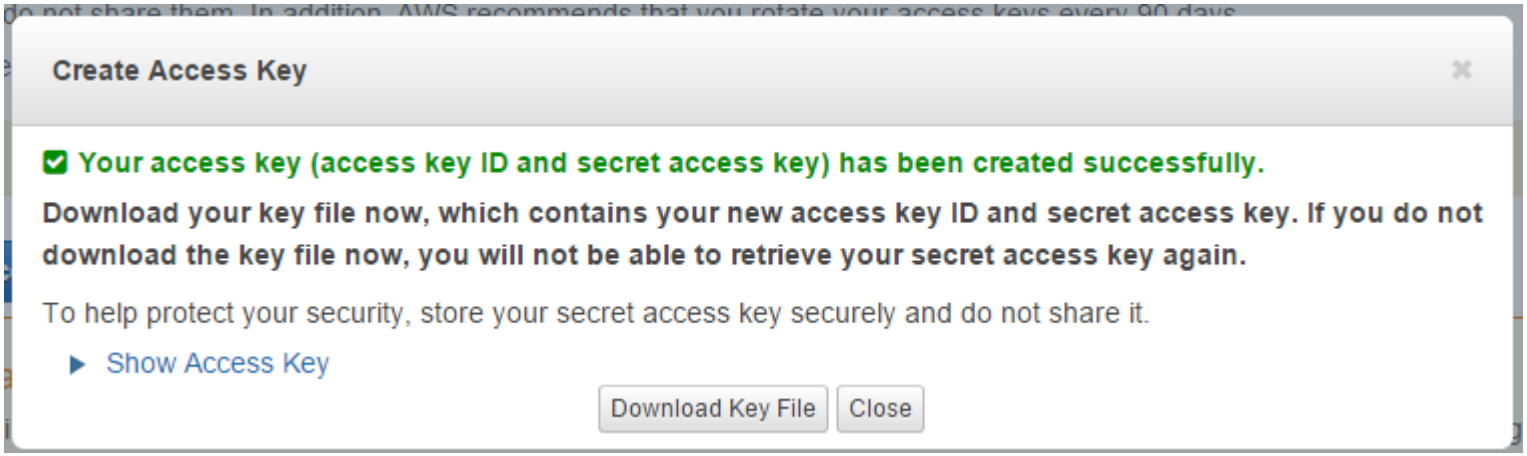
Create New Access Key

Important Change - Managing Your AWS Secret Access Keys

As described in a previous announcement, you cannot retrieve the existing secret access keys for your AWS root account, though you can still create a new root
time. As a best practice, we recommend creating an IAM user that has access keys rather than relying on root access keys.

+ CloudFront Key Pairs
+ X.509 Certificates
+ Account Identifiers

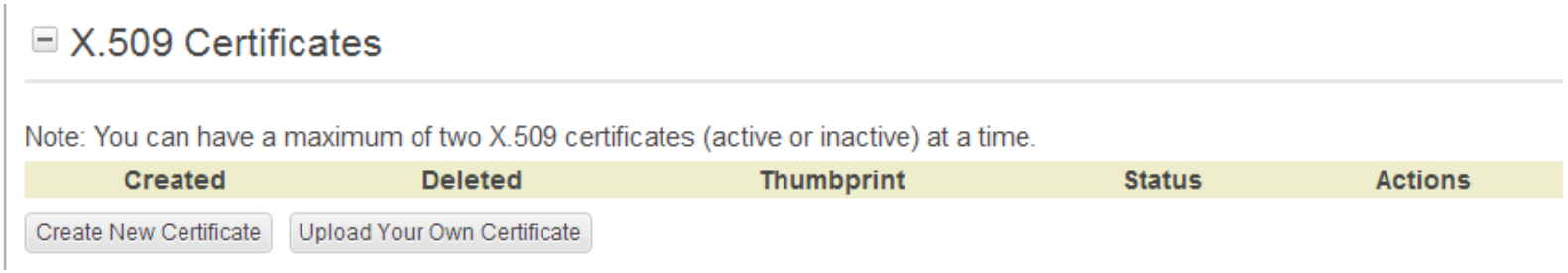
Download Access Key File



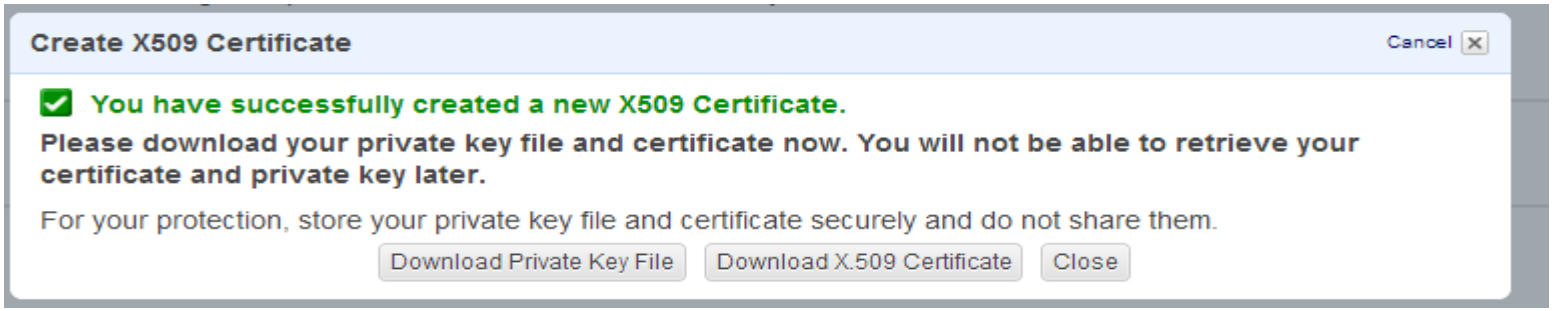
- File named `rootkey.csv` was downloaded to your hard drive.
- Save that file it contains to important credentials:

```
AWSAccessKey=ETWEHJGKHDOBEKBRA
AWSSecretKey=hl1BVgjWddgdfrsjdfopsefkxdlP60VIp3DkX3
```

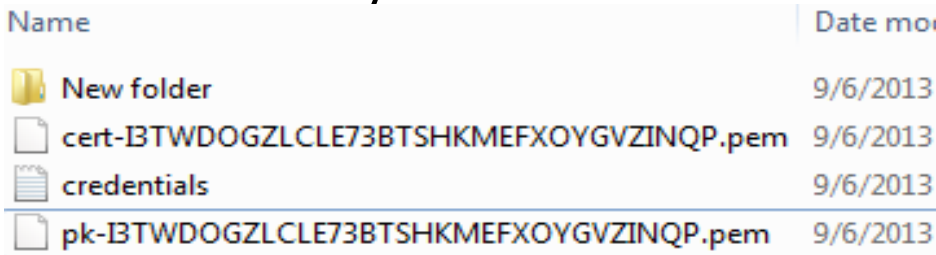
Generate X509 Certificate and Private Key



- Private key can be downloaded only once.
- Hit “Create New Certificate”

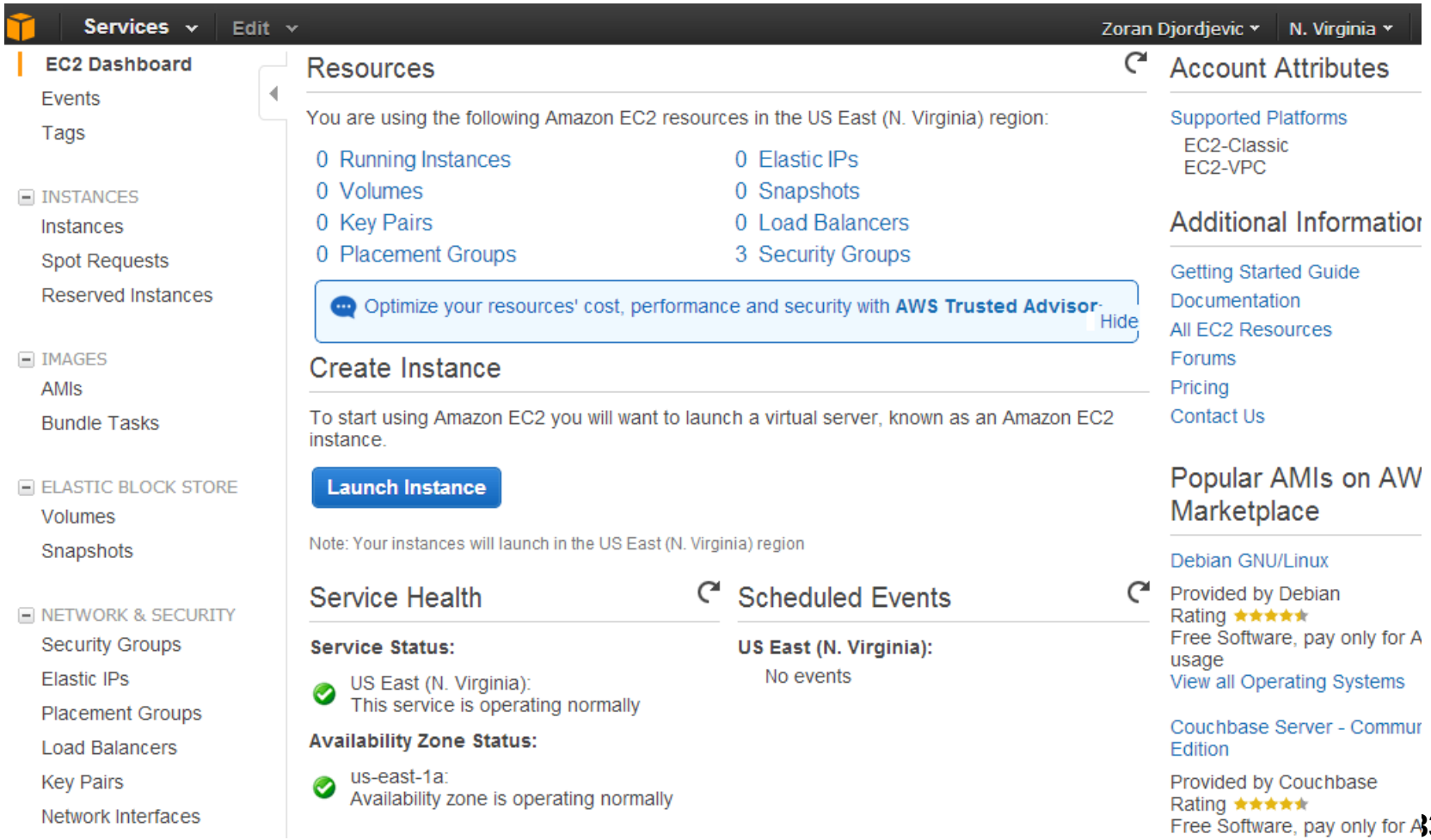


- Download and save both the Private Key and X.509 Certificate as separate files. Store them in a standard directory
- Certificate is your public key, you can show it to anyone.
- Private key is very secret.
- Never show private key to anyone.

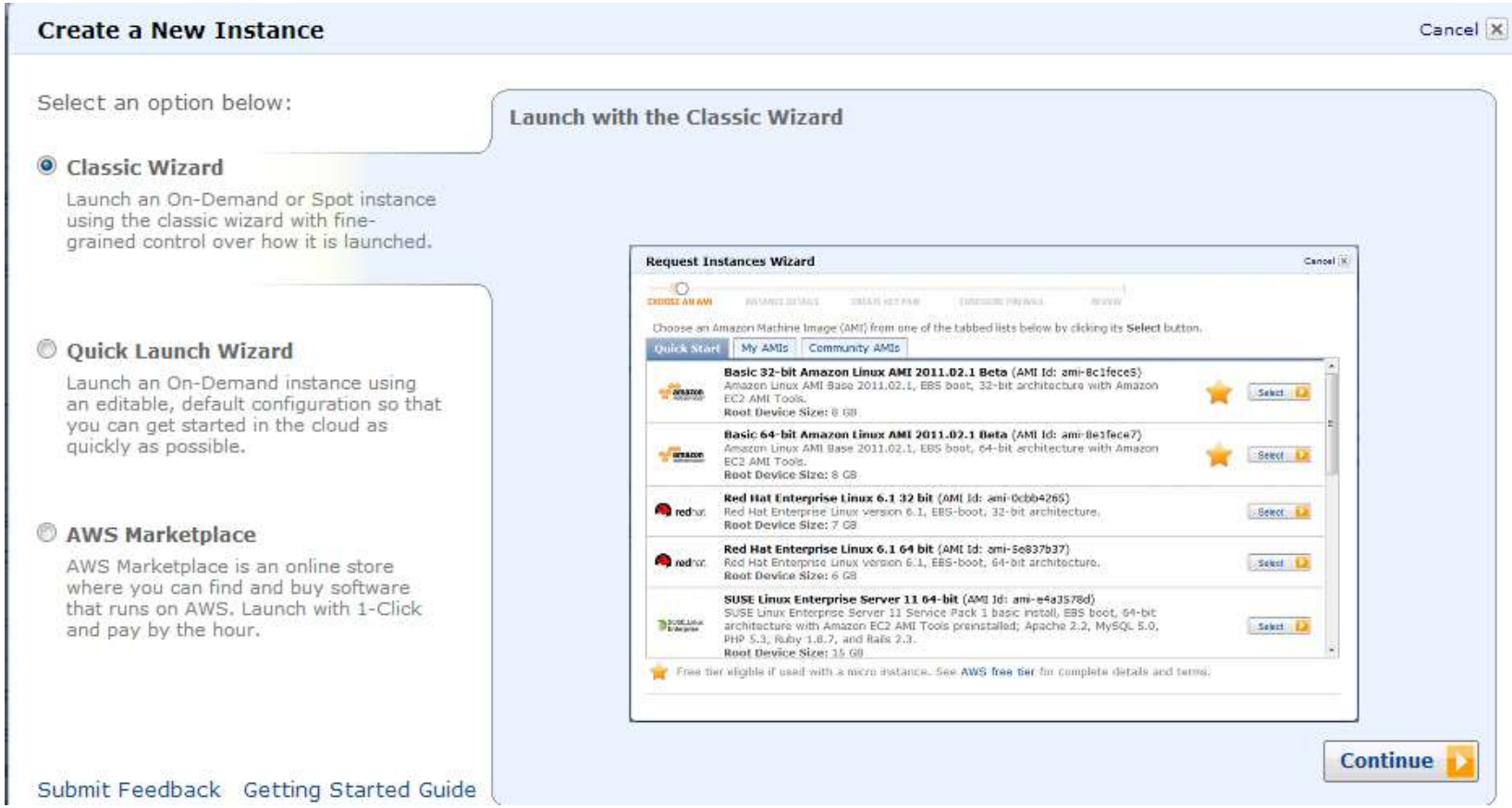


Launch Instance

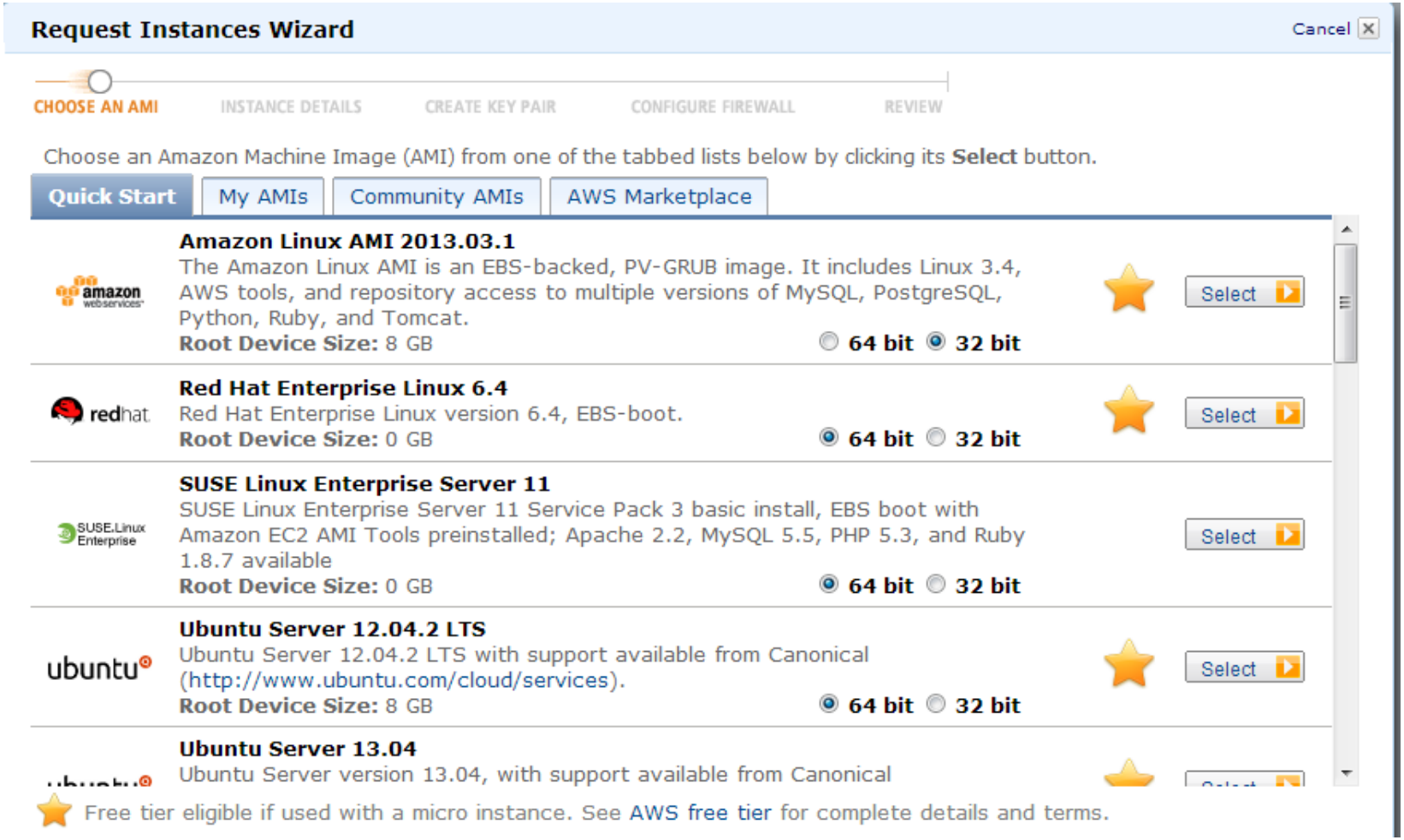
- Sign in to AWS Management Console. Select EC2 Service.
- Select Launch Instance. Notice “N. Virginia”. This my region. If you are in Europe or Asia you should select a different region.



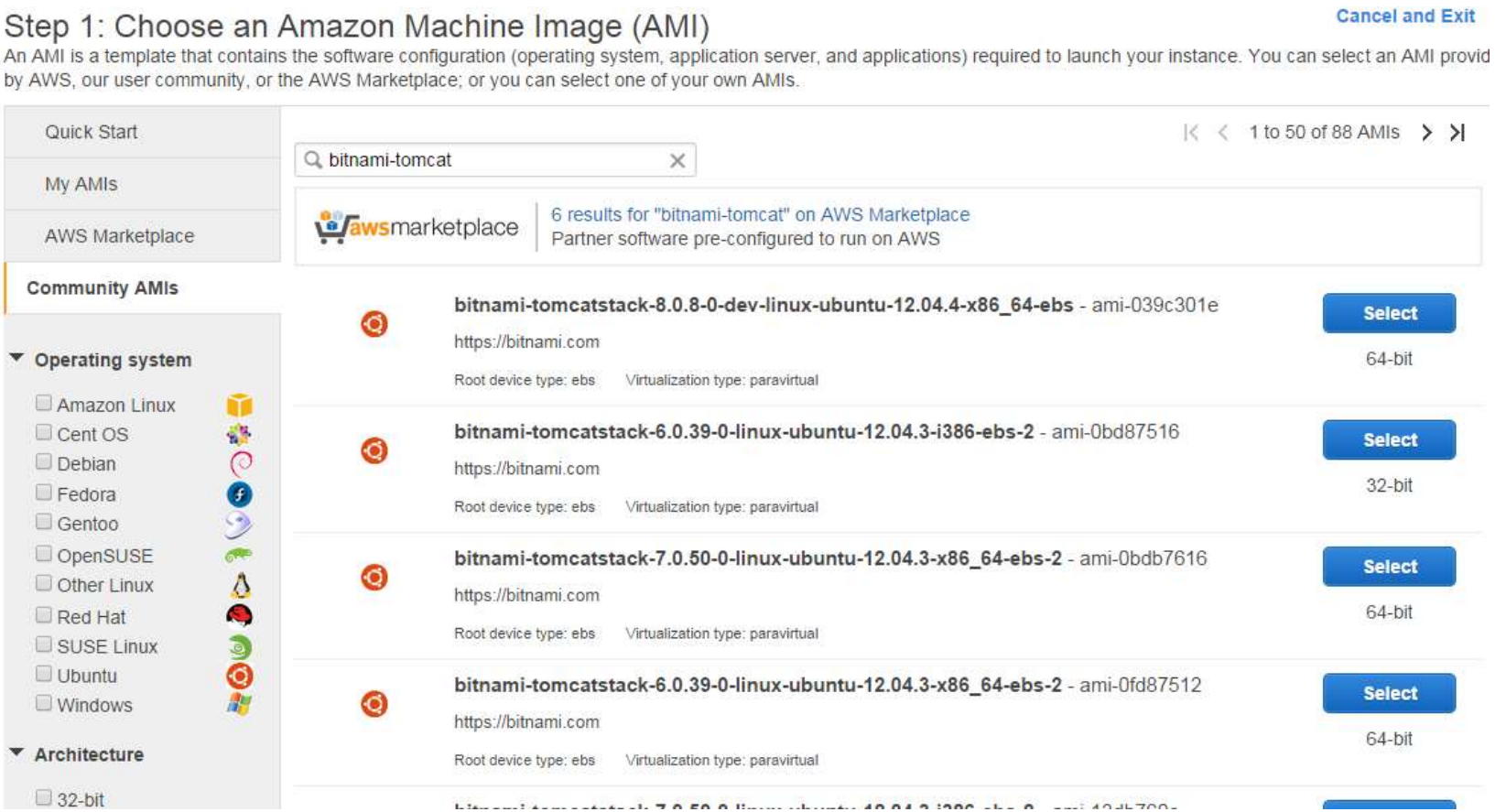
Create New Instance, Continue



Select “Community AMIs” tab

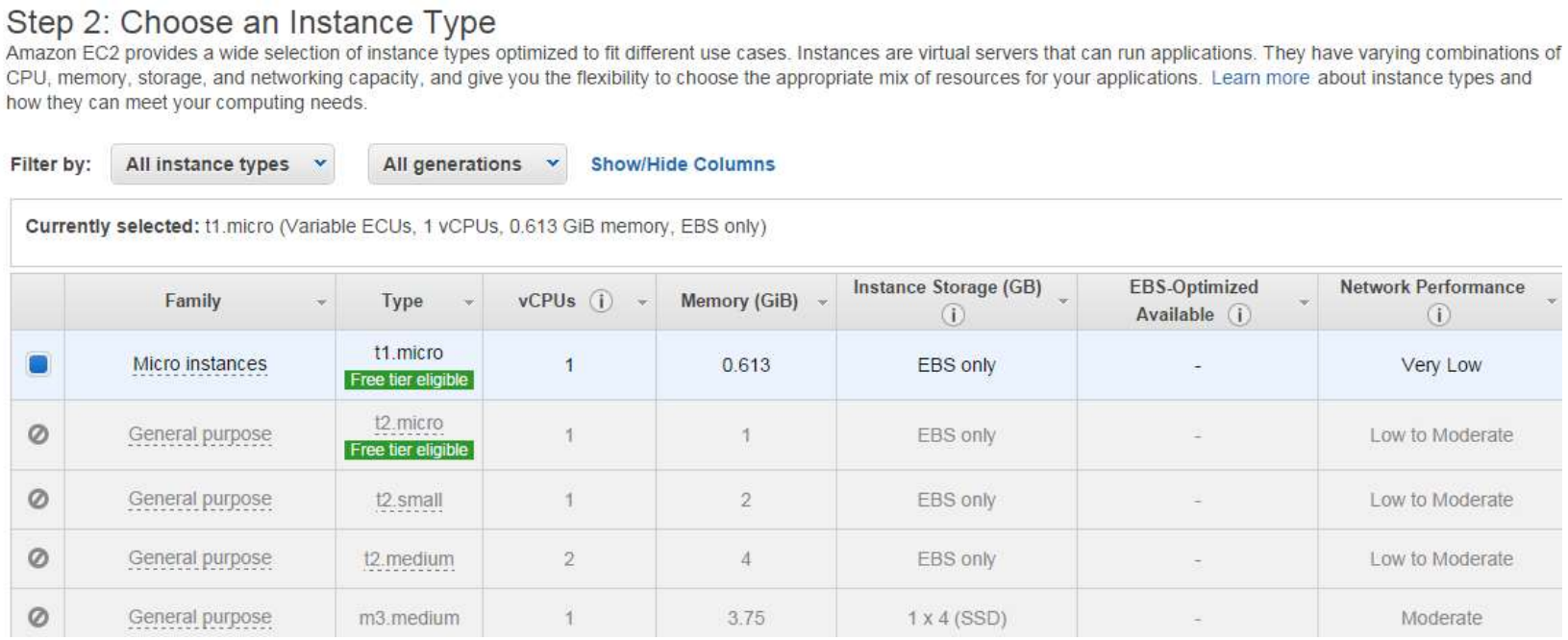


Enter `bitnami-tomcat` in Search Criteria



- Do not choose an AMI with `aws-marketplace` prefix. Those need an additional authorization and some additional payments.. Select for example ami-008db468

Choose Micro Instance Type, Leave No Preference for Availability Zone



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types | All generations | Show/Hide Columns

Currently selected: t1.micro (Variable ECUs, 1 vCPUs, 0.613 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
●	Micro instances	t1.micro Free tier eligible	1	0.613	EBS only	-	Very Low
⊘	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
⊘	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
⊘	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
⊘	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate

- On the following, Configure Instance Details Screen, take the defaults and hit Continue.
- On the Add Storage screen, leave defaults, Continue.
- On the Tag instance screen provide some name in the Value field, e.g. My Tomcat server. Continue.

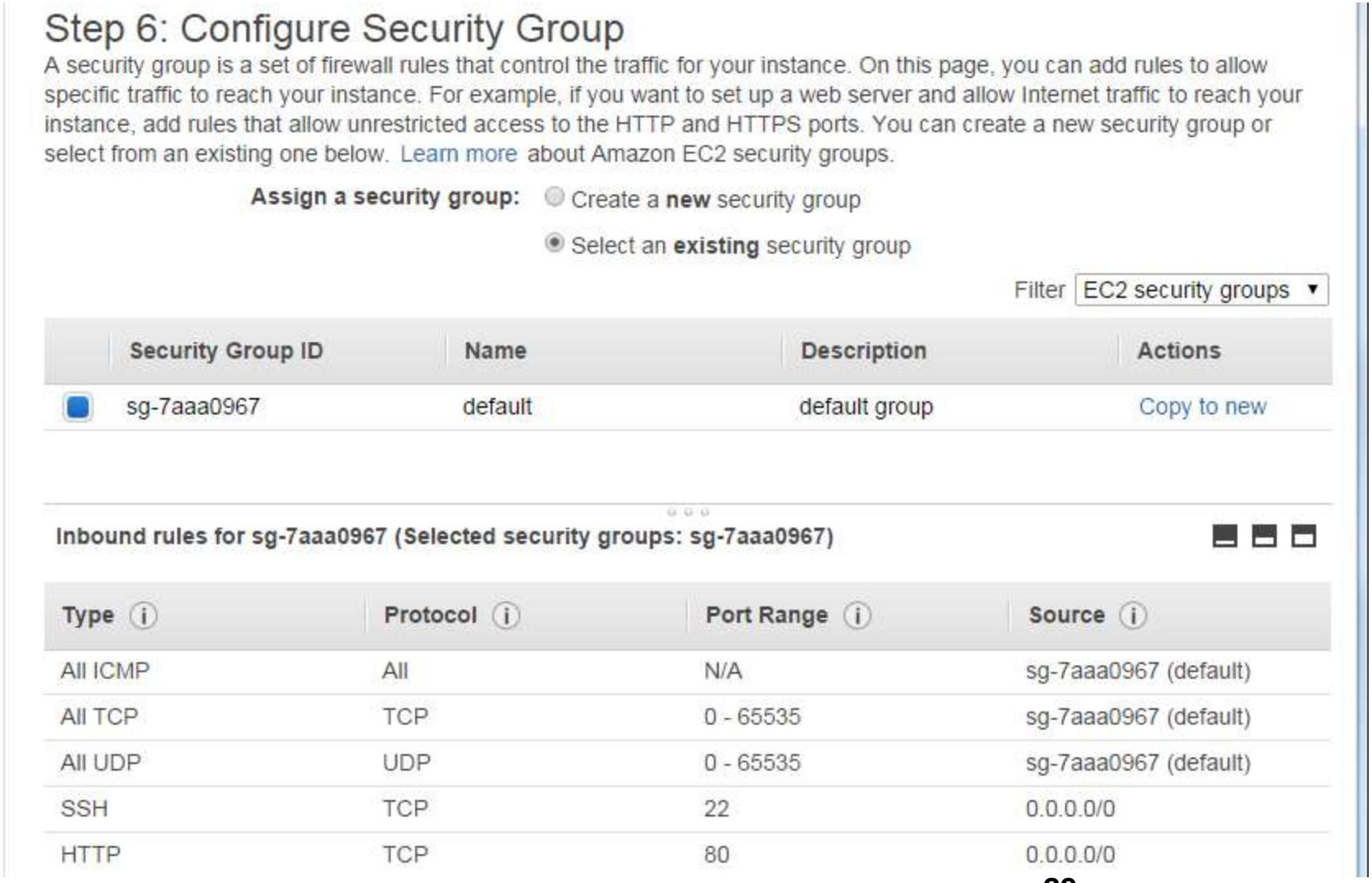
Create New Key Pair

- The next screen is introducing “Key Pair”. We do not have one, so let us select “Create a new Key Pair”.

- Save the file, `ec2hu.pem`, in my case, to a safe location.
- Next time you launch an instance, can use the same Key Pair.

Select Security Group

- For now, please select the default security group. Continue.



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ○ Create a **new** security group

◉ Select an **existing** security group

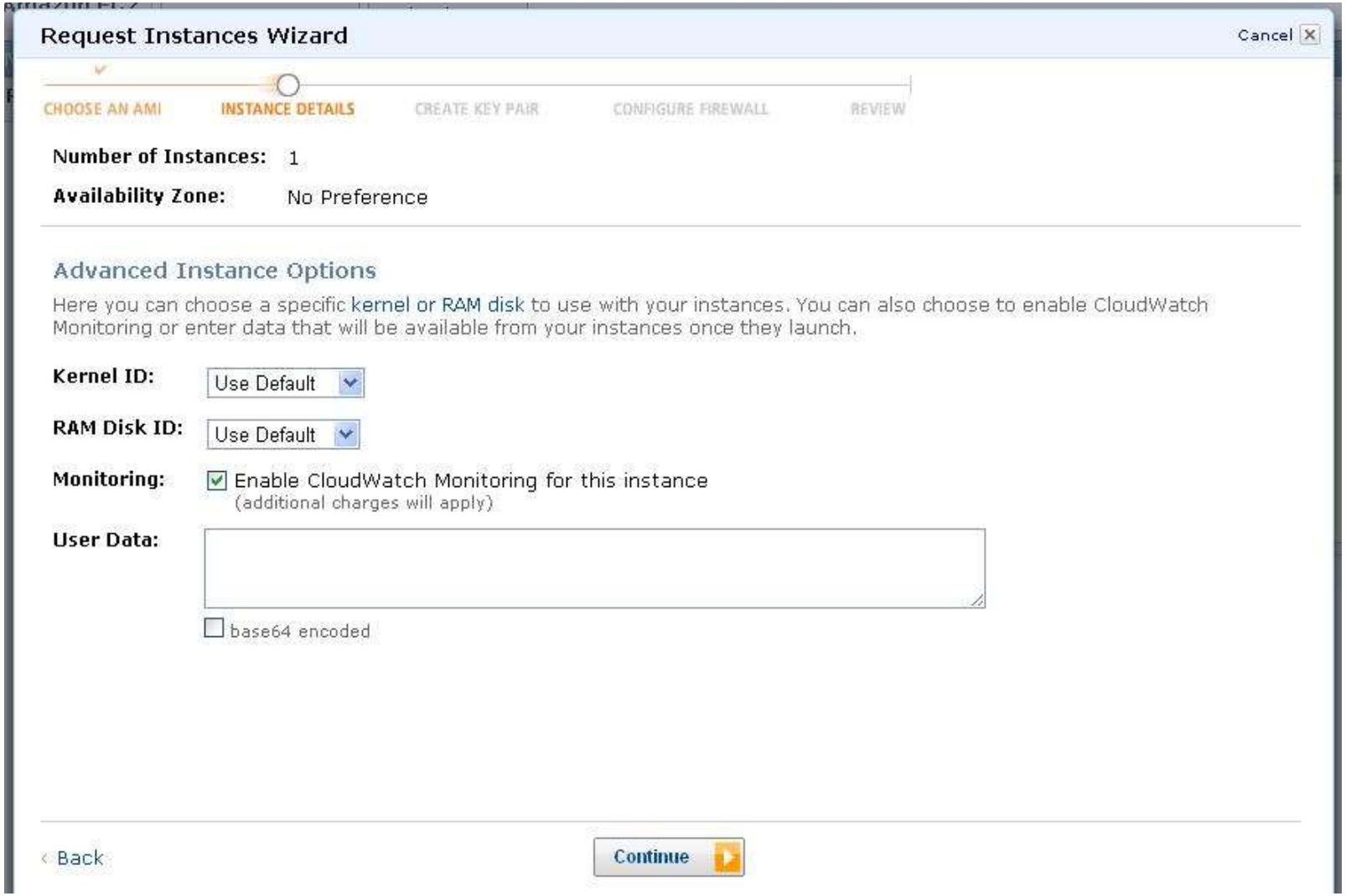
Filter: EC2 security groups

	Security Group ID	Name	Description	Actions
■	sg-7aaa0967	default	default group	Copy to new

Inbound rules for sg-7aaa0967 (Selected security groups: sg-7aaa0967)

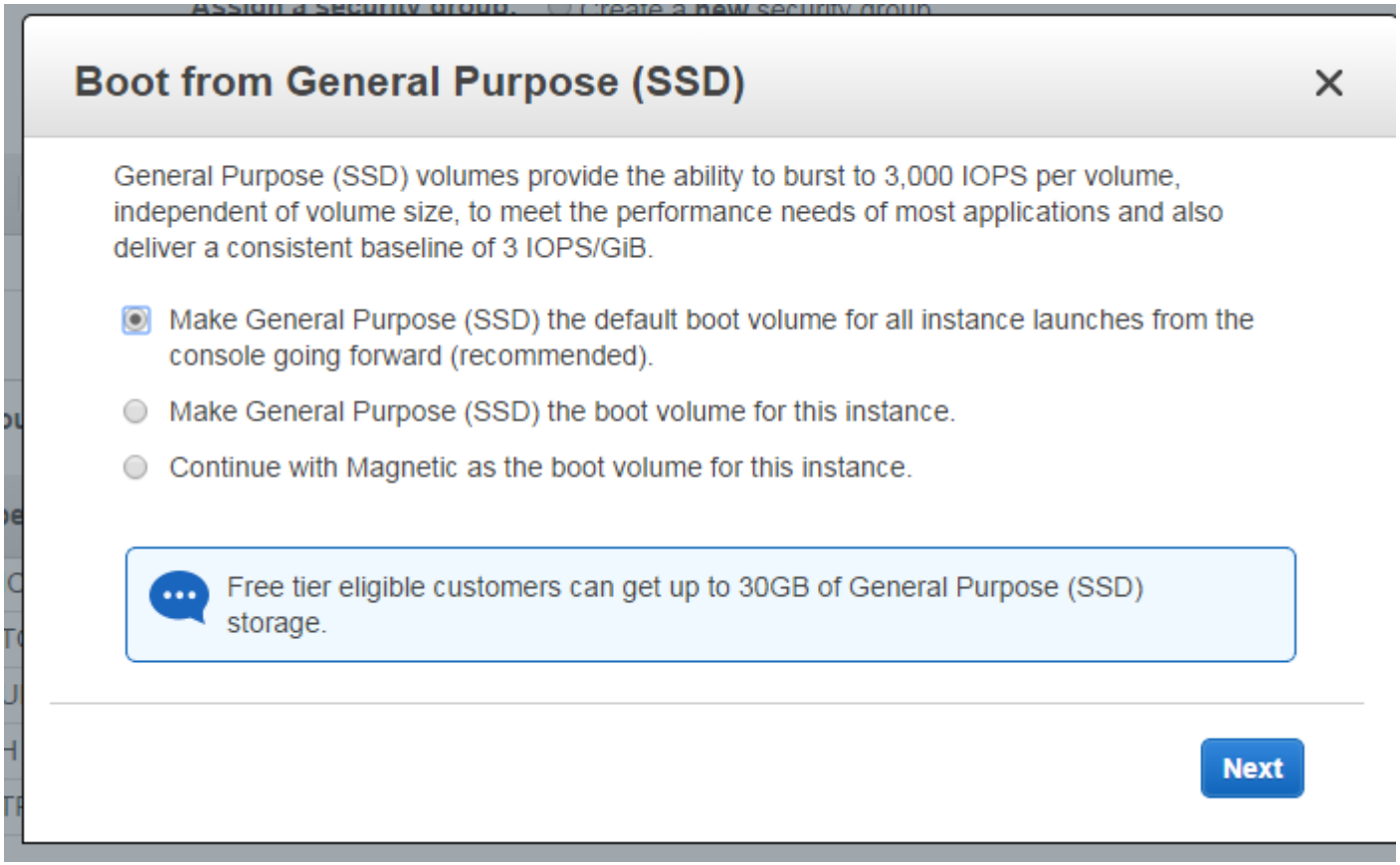
Type	Protocol	Port Range	Source
All ICMP	All	N/A	sg-7aaa0967 (default)
All TCP	TCP	0 - 65535	sg-7aaa0967 (default)
All UDP	UDP	0 - 65535	sg-7aaa0967 (default)
SSH	TCP	22	0.0.0.0/0
HTTP	TCP	80	0.0.0.0/0

Enable Cloud Watch

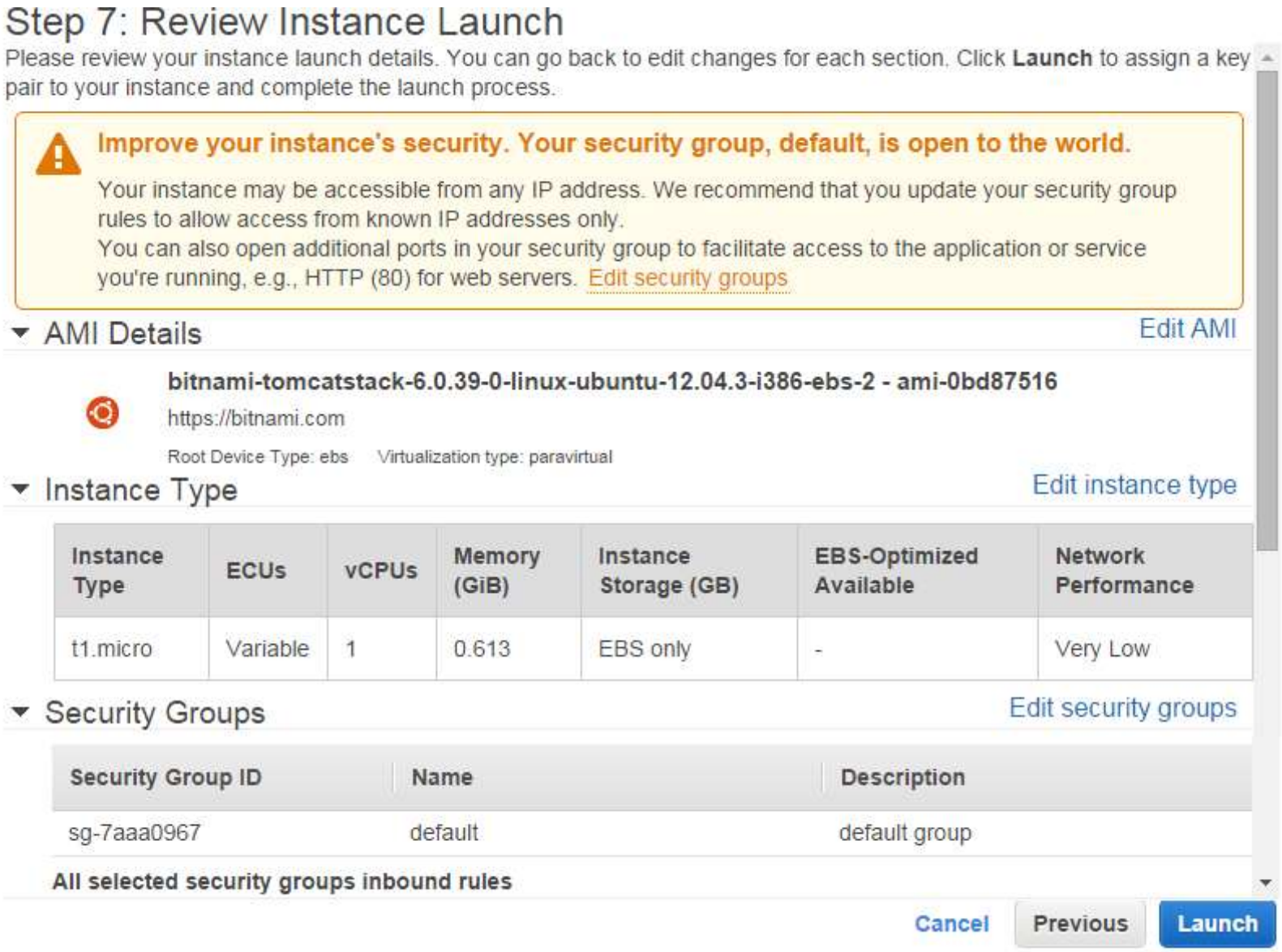


Boot from General Purpose (SSD) Volume

- If not too expensive always choose Solid State Drive (SSD)

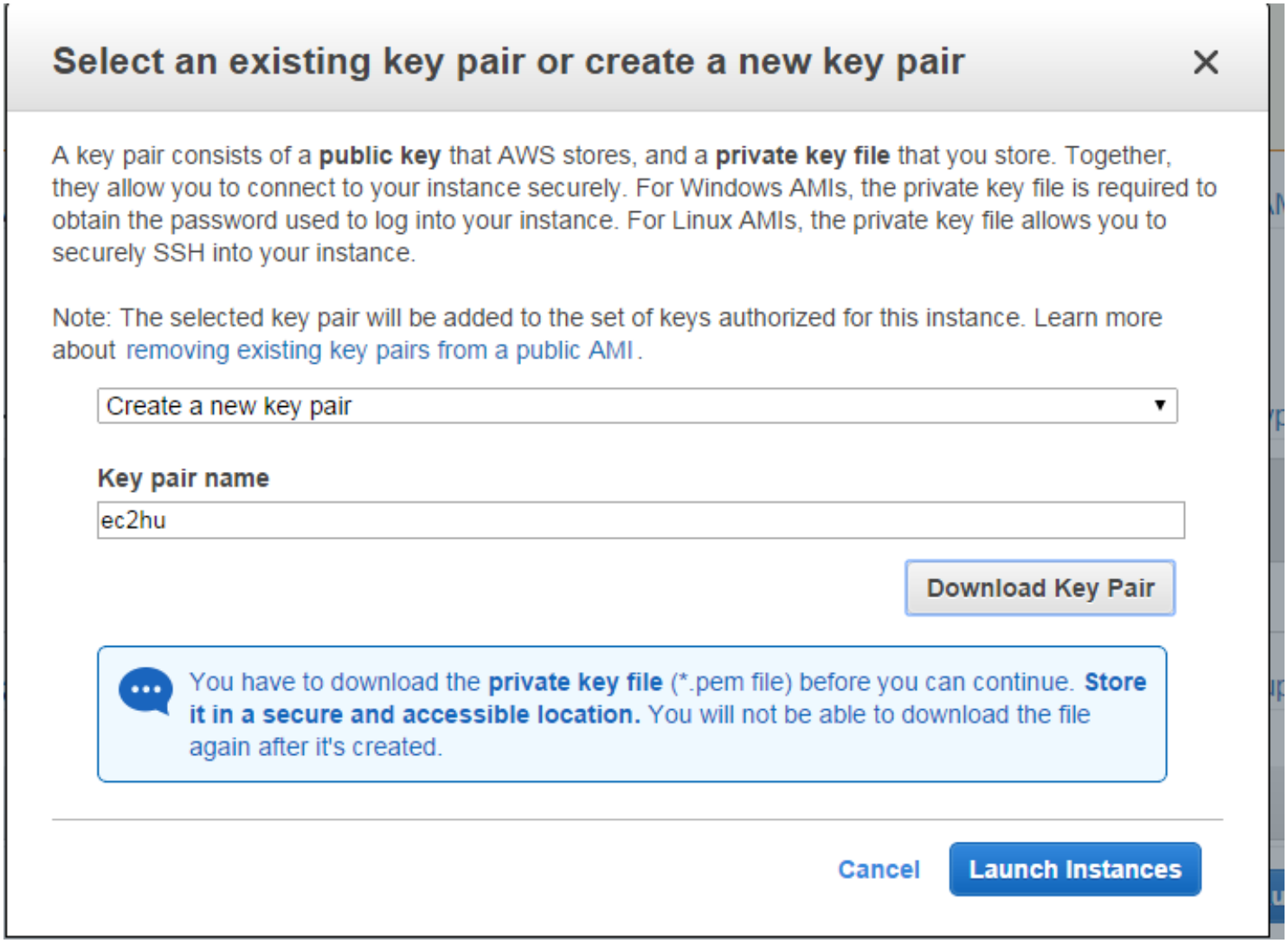


Review Instance Description, Launch

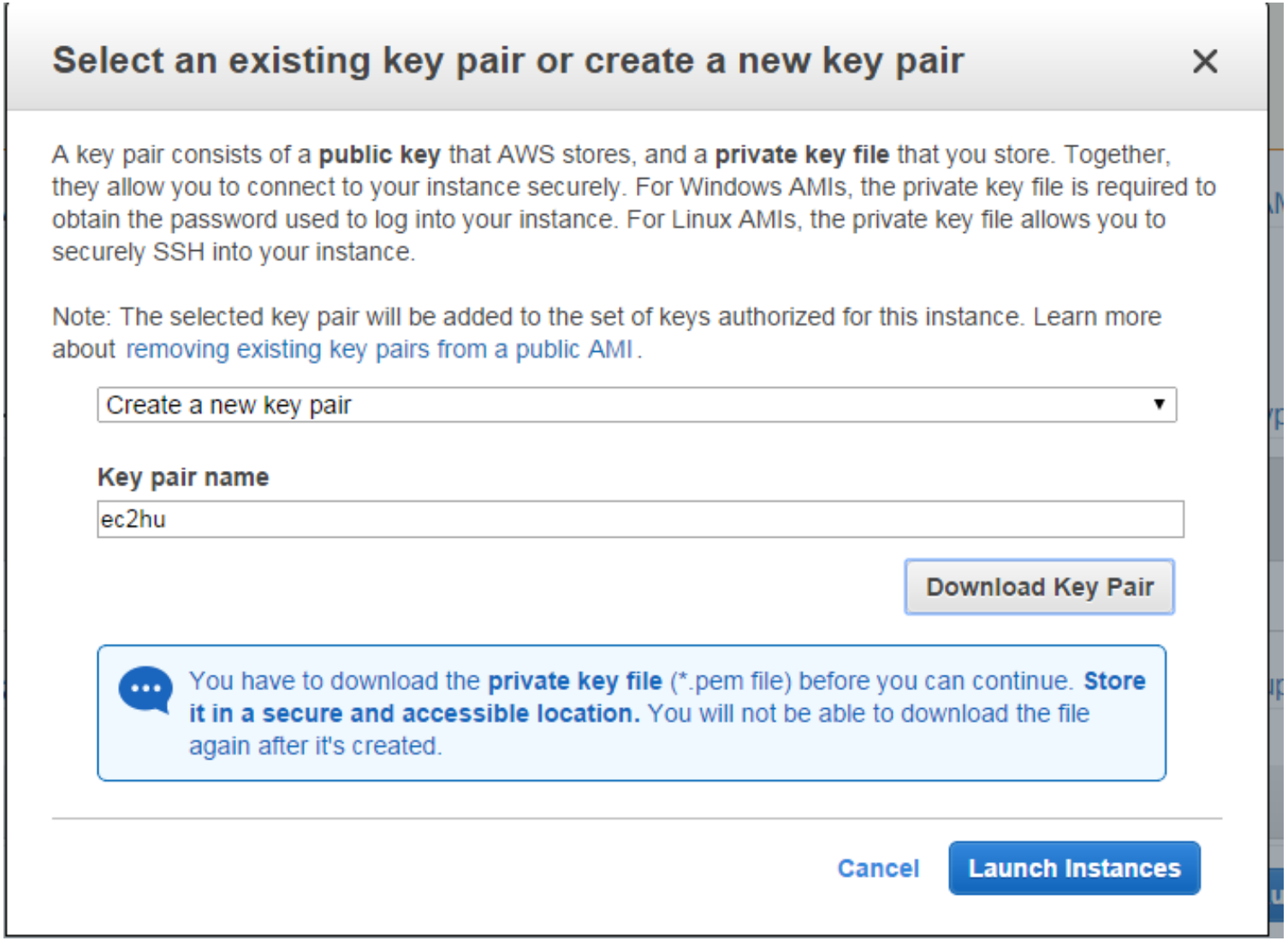


Review Instance Description, Launch

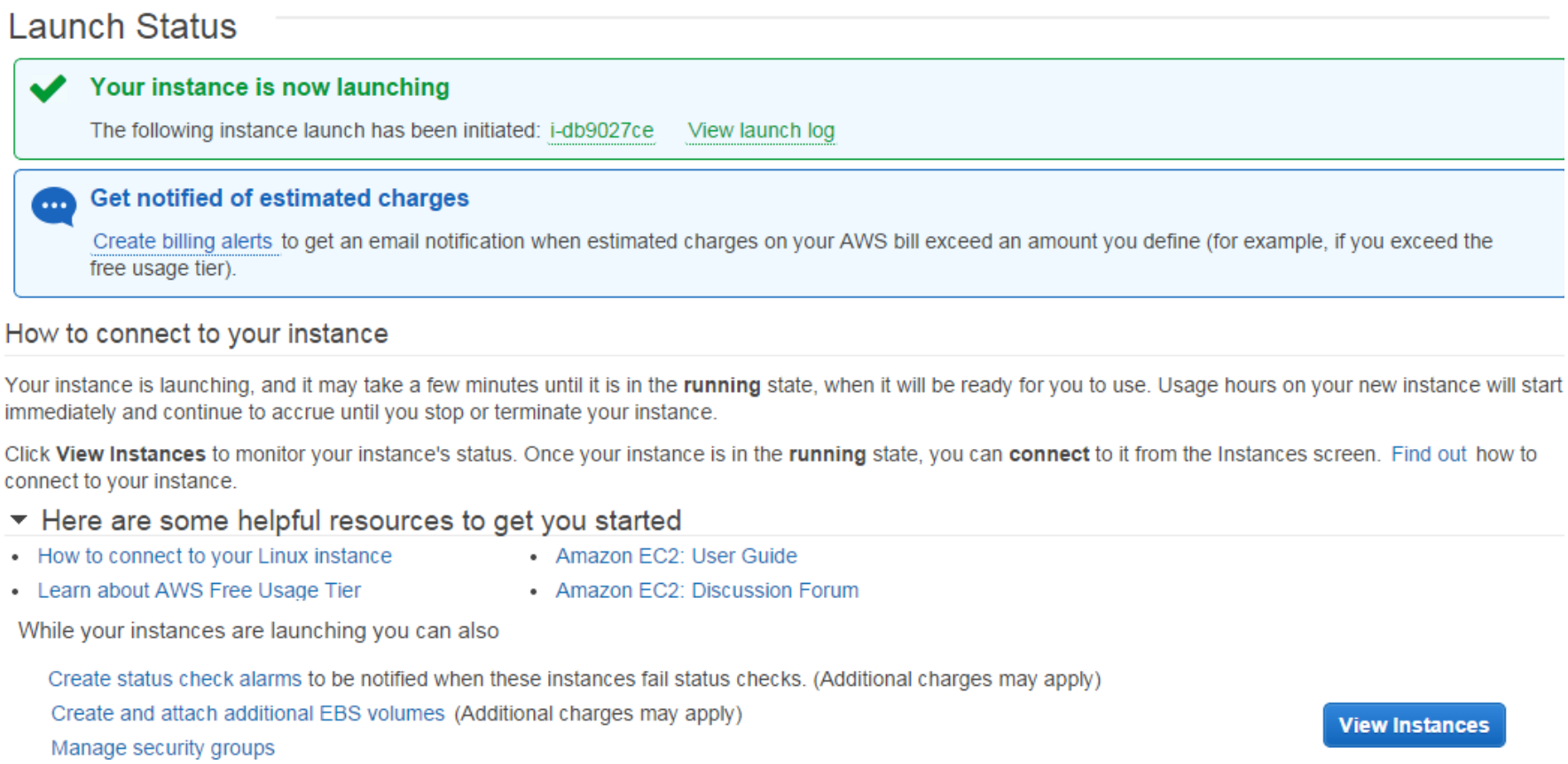
Select an existing key pair or create a new



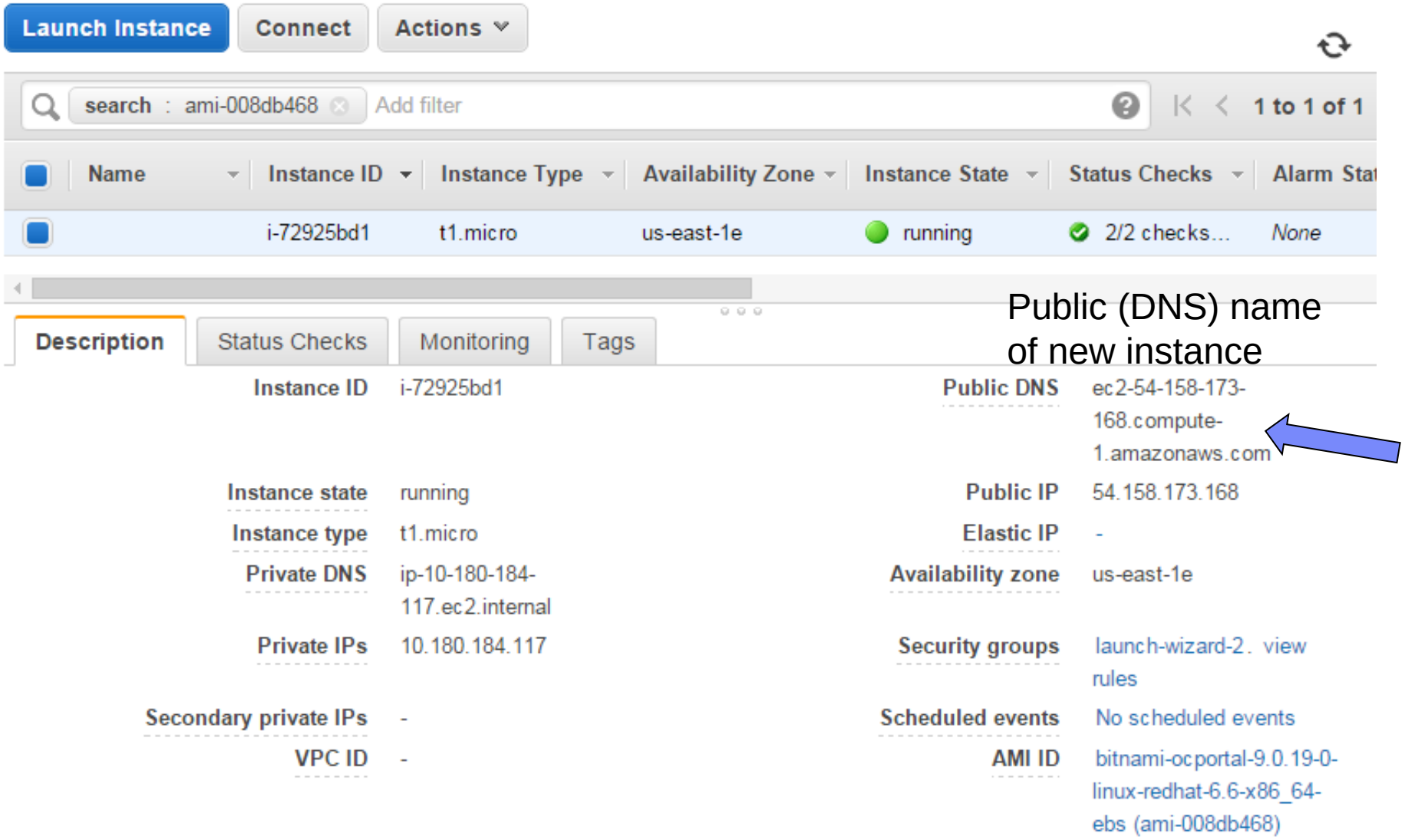
If asked confirm that you do have the key pair



Instance is Launching



My Instances



Public (DNS) name of new instance

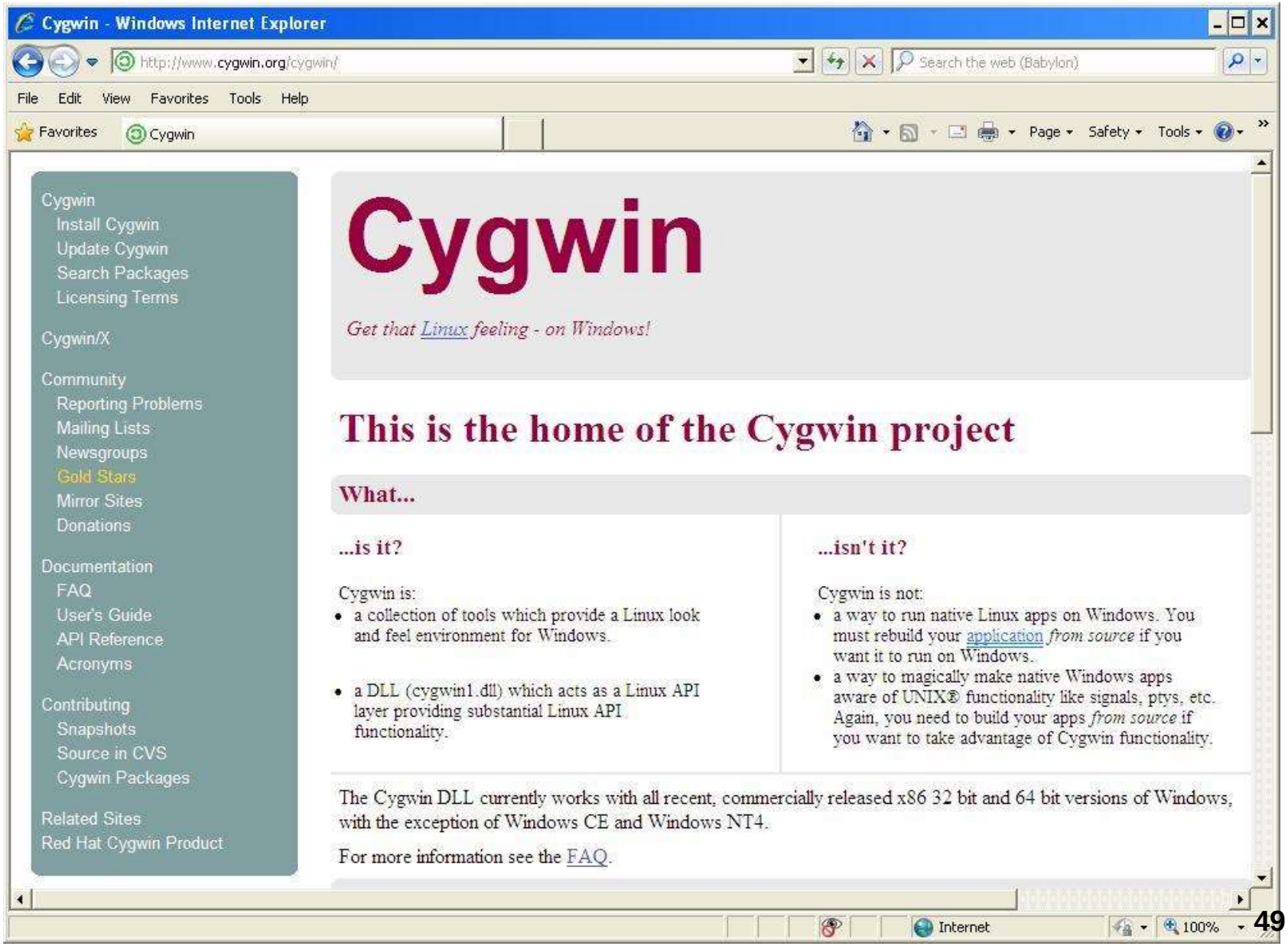
- Please note: security group is "`default`". Instance has a unique name: `i-72925bd1`, AMI ID is `ami-008db468`.

Tools to Connect With

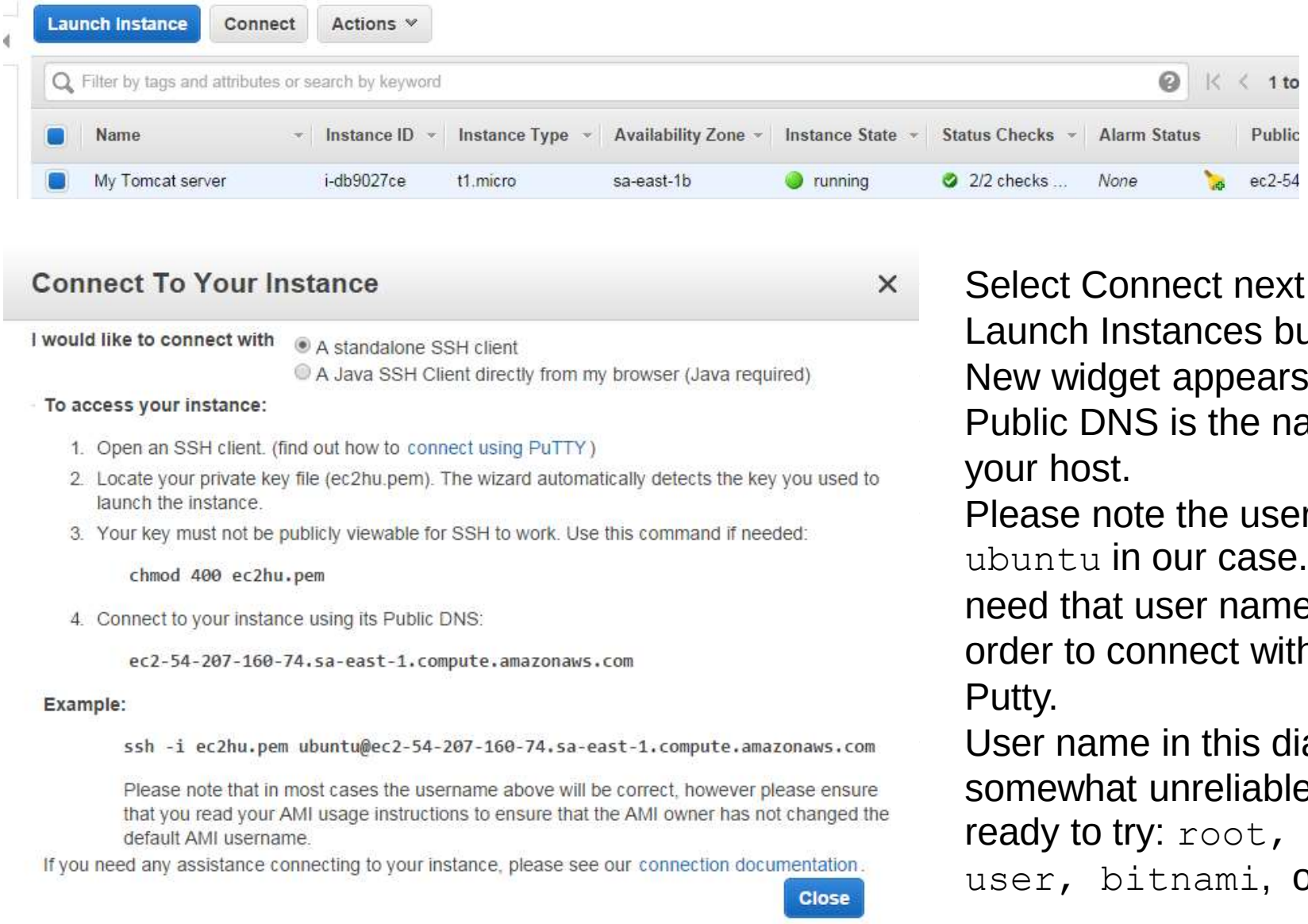
- To connect to an instance you need a SSH (Secure Shell) tool.
- `Putty` is one of Windows tools emulating SSH protocol. When you are downloading Putty, from the same site, please, download `WinSCP` as well. `WinSCP` provides secure copy functionality on Windows.

- `CygWin` on Windows provides a complete set of Linux (Unix) utilities, including SSH.
- Linux and Unix boxes, including Apple machines, by rule have a `ssh` tool.
- OpenSSH project provides a free implementation of the full suite of SSH tools and is part of CygWin download.
- OpenSSH project is under Net group of projects. When downloading Cygwin, make sure you check (select) OpenSSH.

Download CygWIN



Select Connect on Your Instance Screen



Select Connect next to Launch Instances button.
New widget appears.
Public DNS is the name of your host.
Please note the user name, `ubuntu` in our case. We need that user name in order to connect with ssh or Putty.
User name in this dialog is somewhat unreliable. Be ready to try: `root`, `ec2-user`, `bitnami`, others.

Connect with SSH

- In Cygwin's command line shell, change directory to the location of the key pair file that you created when you launched the instance.
- Use the `chmod` command to make sure your key pair file isn't publicly viewable.
- For example, my file is named `ec2hu.pem`. I do this first

```
$chmod 400 ec2hu.pem
```

- Connect to your instance using the instance's public DNS name (which you should have recorded or copied earlier). For example, if the key file is `ec2hu.pem`, user name `ubuntu` and the instance's DNS name:

```
ec2-54-158-173-168.compute.amazonaws.com
```

we use the following command (all on one line)

```
$ssh -i ec2hu.pem
ubuntu@ec2-54-158-173-168.compute-1.amazonaws.com
```

- Note: `ubuntu` is the user name or the Linux account. That user usually has `sudo` privileges, i.e. it could execute commands as if `root`.

Secure Copy with SCP

- If you want to transfer a new file to home directory of your EC2 machine, the following command will do that.

```
scp -i ec2hu.pem newfile.txt ubuntu@ec2-54-158-173-168.compute-1.amazonaws.com:/home/ubuntu
```

Be Positive, Say ‘yes’

```
$ ssh -i ec2hu.pem ubuntu@ec2-54-158-173-168.compute-
1.amazonaws.com
The authenticity of host 'ec2-54-158-173-168.compute-
1.amazonaws.com (54.242.17.195)' can't be established.
ECDSA key fingerprint is
f8:bc:f7:b2:52:8e:a6:ac:a2:9f:92:7d:dd:76:3c:f9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-54-158-173-168.compute-
1.amazonaws.com,54.242.17.195' (ECDSA) to the list of known hosts.
Permission denied (publickey).
```

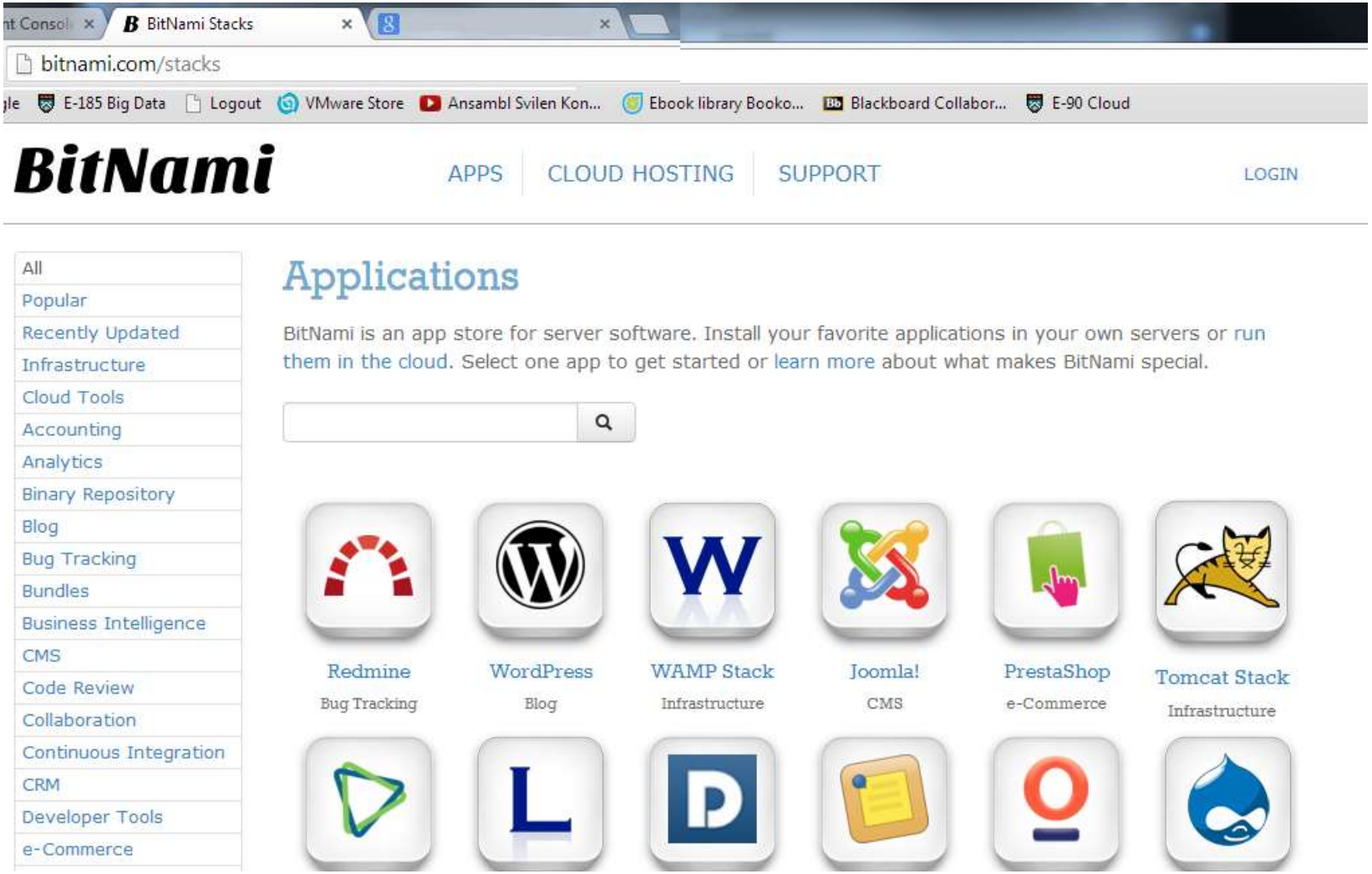
- Remote host (AWS box) might reject your login.
- The issue might be in the user name. Try `bitnami or ec2-user`

Be Positive, Say ‘yes’

```
$ ssh -i ec2hu.pem bitnami@ec2-54-158-173-168.compute-
1.amazonaws.com
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.2.0-49-virtual i686)

       __ _ _   _  _                 _
      | _ |_) |_| \| |__ _ _ __  (_)
      | _ \ |  _| .` / _` | '  \| |
      |___/_|\__|_|\_\__,_|_|_|_|_|

  *** Welcome to the BitNami Tomcat 7.0.42-0 ***
  *** BitNami Wiki:   http://wiki.bitnami.com/ ***
  *** BitNami Forums: http://answers.bitnami.com/ ***
To run a command as administrator (user "root"), use "sudo
<command>".
See "man sudo_root" for details.

bitnami@ip-10-196-125-57:~$
```

- This time, we are in, as user `bitnami`.
- The note about `sudo` command is very important. In most cases, if you do not login into the instance as `root`, you login as a user with `sudo` (super user doer??) privilege.

Bitnami.com/stacks



Are Apache and Tomcat present?

- We can verify whether Apache server and Tomcat server are up and running by doing Unix `grep` for `httpd` (name of Apache process) and by grepping for `tomcat`.
- On our instance, both tests are positive:

bitnami@ip-10-196-125-57:~$ ps -ef | grep httpd

```
root      1212     1  0 02:31 ?        00:00:00
/opt/bitnami/apache2/bin/httpd -f /opt/bitnami/apache2/conf/httpd.conf

daemon    1244  1212  0 02:31 ?        00:00:00
/opt/bitnami/apache2/bin/httpd -f /opt/bitnami/apache2/conf/httpd.conf

daemon    1245  1212  0 02:31 ?        00:00:00
/opt/bitnami/apache2/bin/httpd -f /opt/bitnami/apache2/conf/httpd.conf
```

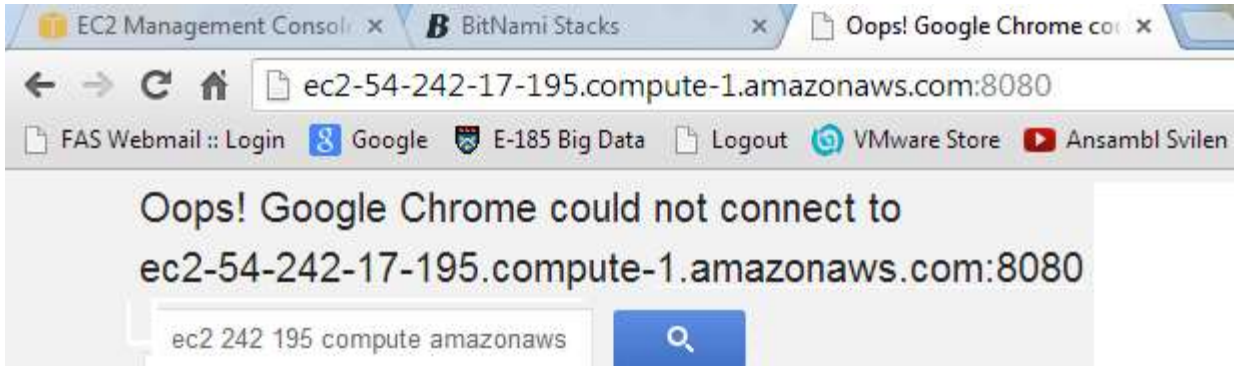
bitnami@ip-10-196-125-57:~$ ps -ef | grep tomcat

```
root      1192     1  0 02:31 ?        00:00:00 jsvc.exec -java-home
/opt/bitnami/java -user tomcat -pidfile /opt/bitnami/apache-
tomcat/temp/catalina.pid -wait 10 -outfile /opt/bitnami/apache-
tomcat/logs/catalina-daemon.out -errfile &1 -classpath
/opt/bitnami/apache-tomcat/bin/bootstrap.jar:/opt/bitnami/apache-
tomcat/bin/commons-daemon.jar:/opt/bitnami/apache-tomcat/bin/tomcat-
juli.jar -Djava.util.logging.config.file=/opt/bitnami/apache-
tomcat/conf/logging.properties -XX:MaxPermSize=512m -Xms256m -Xmx512m
```

Test the URLs

- Test at DNS name and port 80 is successful. Apache is visible.

- Test at DNS name and port 8080 is not successful. Tomcat is not visible at its default port.



Examine on which ports Linux box listens

- We can use `netstat -an` command to see where, on which ports is Linux box listening for incoming calls.

```
bitnami@ip-10-196-125-57:~$ netstat -an | grep LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:21            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::443                  :::*                    LISTEN
tcp6       0      0 :::8009                 :::*                    LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
unix  2  [ ACC ] STREAM     LISTENING     7203     /var/run/acpid.socket
```

- We see that port 80 is active. That must be the Apache

sudo vi /opt/bitnami/apache-tomcat/conf/server.xml

- We open the file with vi. The relevant portion of `server.xml` file turns out to be commented out:

```
<!-- A "Connector" represents an endpoint by which requests are received
         and responses are returned. Documentation at :
         Java HTTP Connector: /docs/config/http.html(blocking & non-blocking)
         Java AJP  Connector: /docs/config/ajp.html
         APR (HTTP/AJP) Connector: /docs/apr.html
         Define a non-SSL HTTP/1.1 Connector on port 8080
    -->
    <!--
    <Connector port="8080" URIEncoding="UTF-8" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    -->
    <!-- A "Connector" using the shared thread pool-->
     <!--
     <Connector executor="tomcatThreadPool"
                port="8080" protocol="HTTP/1.1"
                connectionTimeout="20000"
                redirectPort="8443" />
     -->
```

- We will remove comments around connector at 8080 and restart the Tomcat. Save changes in vi by doing: `Esc(ape), :wq! .`

New server.xml

```
    Define a non-SSL HTTP/1.1 Connector on port 8080
-->
<Connector port="8080" URIEncoding="UTF-8" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<Connector executor="tomcatThreadPool"
           port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
<!-- Define a SSL HTTP/1.1 Connector on port 8443
```

- For Tomcat to accept changes we either run shutdown.sh and startup.sh scripts in Tomcat's bin directory

```
..:/opt/bitnami/apache-tomcat/bin$ sudo ./shutdown.sh
..:/opt/bitnami/apache-tomcat/bin$ sudo ./startup.sh
```

- Or run `ctlscript.sh` in `/opt/bitnami` directory, as:

```
..:/opt/bitnami$ sudo ./ctlscript.sh stop tomcat
/opt/bitnami/apache-tomcat/scripts/ctl.sh : tomcat stopped
..:/opt/bitnami$ sudo ./ctlscript.sh start tomcat
/opt/bitnami/apache-tomcat/scripts/ctl.sh : tomcat started
bitnami@ip-10-196-125-57:/opt/bitnami$
```

- If we now run `netstat -an | grep LISTEN` again, we will see that something is listening on port `8080`.

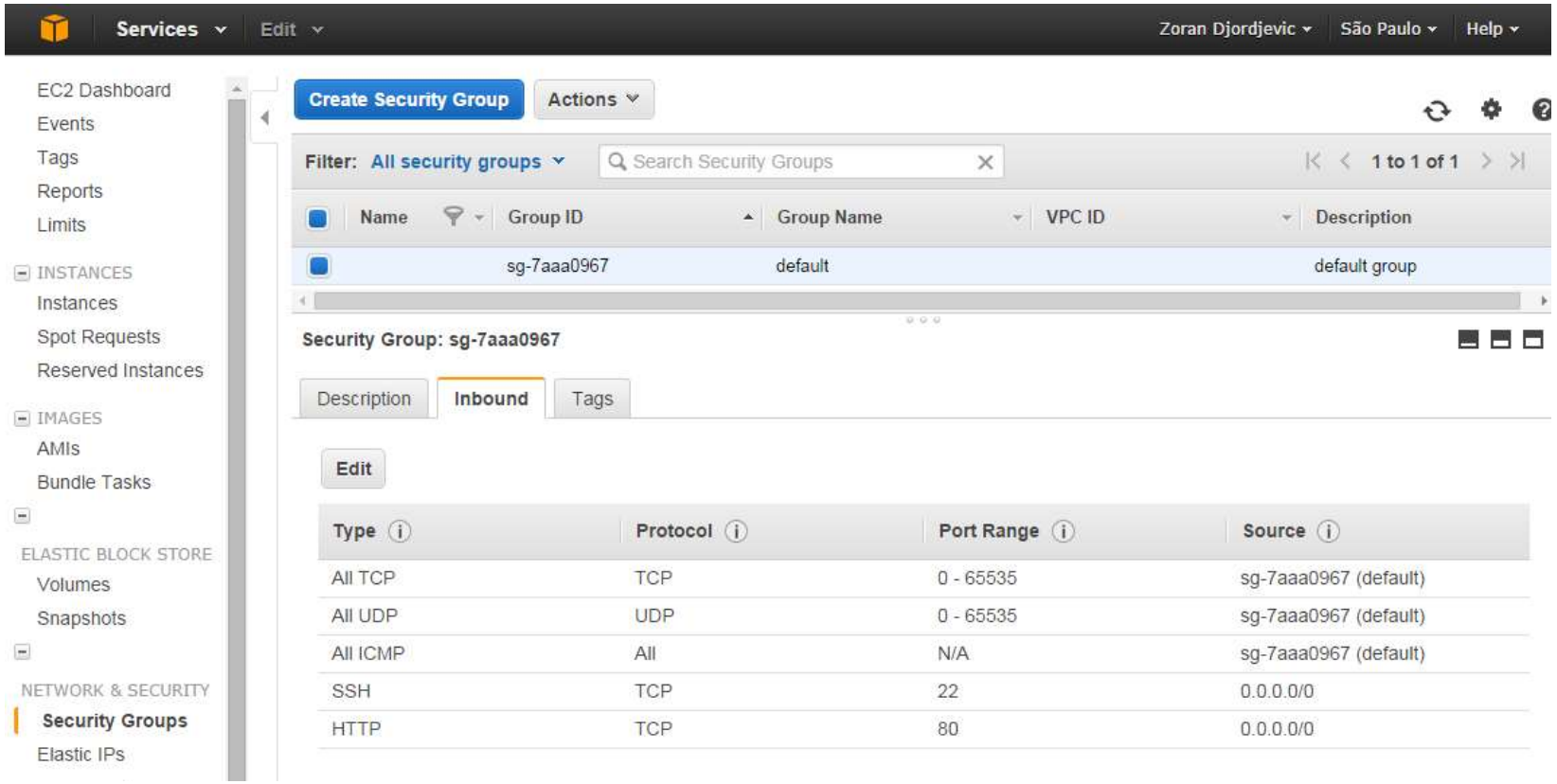
Test the URL:8080

- If we now repeat the test at:

http://ec2-54-158-173-168.compute-1.amazonaws.com:8080/

- We will again fail to see Tomcat's page. Something else is blocking our access to port 8080.
- We need to look at the security group, in our case "default" group. Security groups controls inbound traffic on AWS instances. If we double click on Security Groups and then check mark the default group, we will see that port 80 is open , but port 8080 is not.
- We add 8080 in the Port range filed, click + Add Rule and then, Apply Rule Change. The result with port 8080 added to the firewall is presented on the next slide.

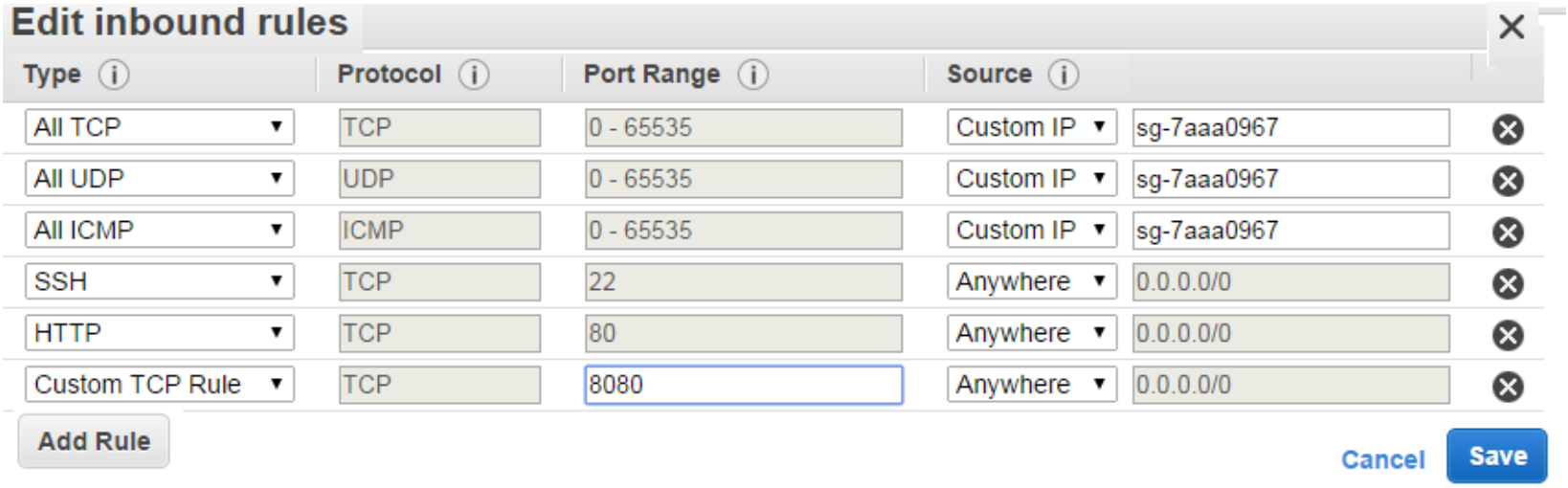
Change Security Group, Add Port 8080



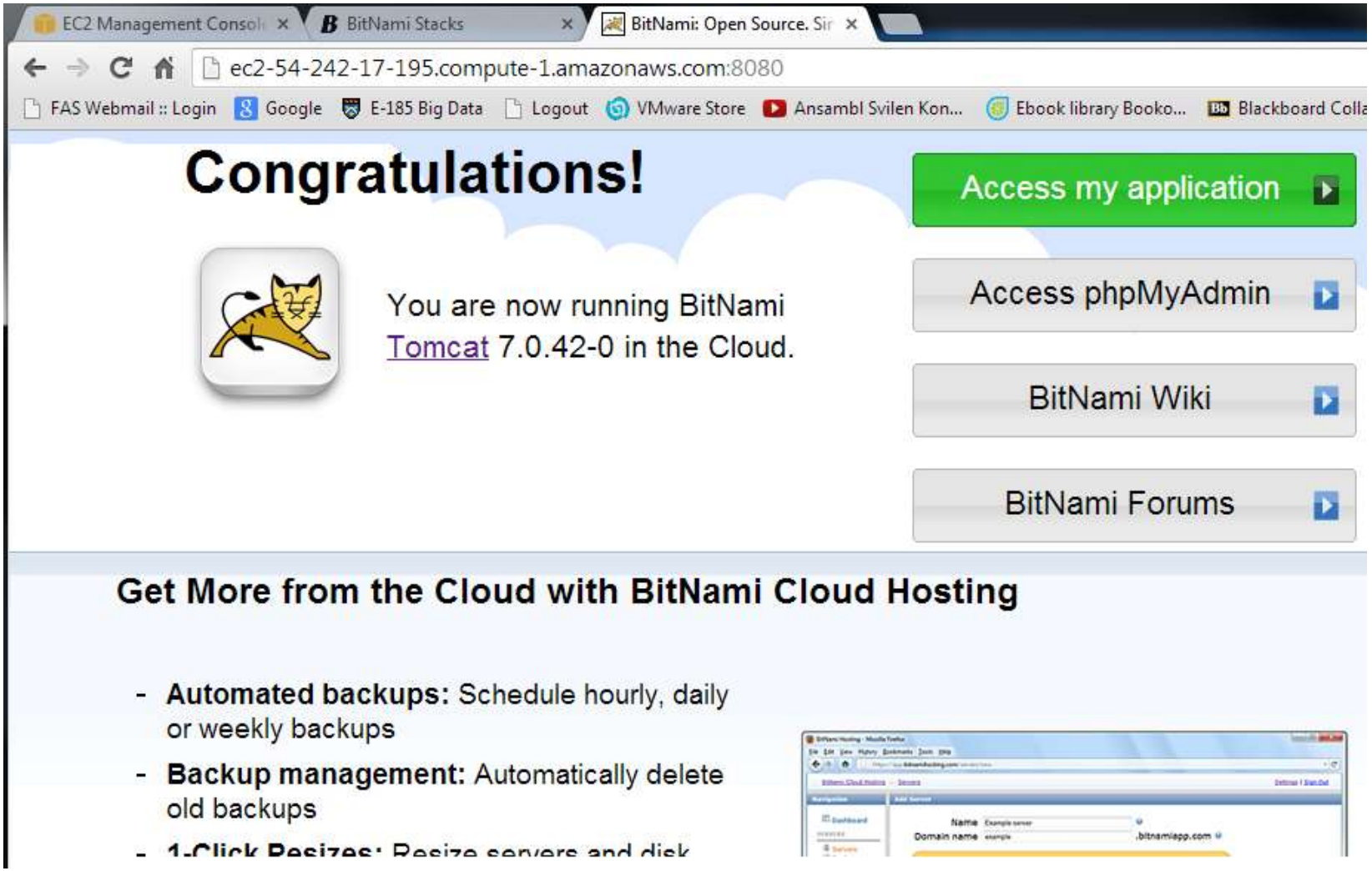
- Select Edit above the rules matrix.

Add Custom TCP Rule for port 8080

- Select Add Rule and add a custom TCP rule for port 8080
- Save Rule Change.
- Apply Rule Change actually propagates the change an makes modifications in `iptables` that control the firewall on your instance.



Test Again



- Tomcat is there at port 8080 and we can declare a Victory.
- Tomcat is easy to deal with and we can start building apps in the Cloud.

Amazon S3

csci 63 Big Data Analytics
LAB 06

Brief Overview of S3

What is Amazon S3

- Amazon S3 is a simple storage service that offers software developers a highly-scalable, reliable, and low-latency data storage infrastructure at a very low cost.
- S3 can store any amount of data. Data can be stored in S3 anywhere from the Internet. AWS Storage facilities are grouped in Regions. You can choose any region that optimizes the latency, cost or address local regulations.

What could be stored in S3

- Users of S3 commonly store images, video, and other content for their websites on Amazon S3. You can host an entire static website from an Amazon S3 buckets.

How much could be stored

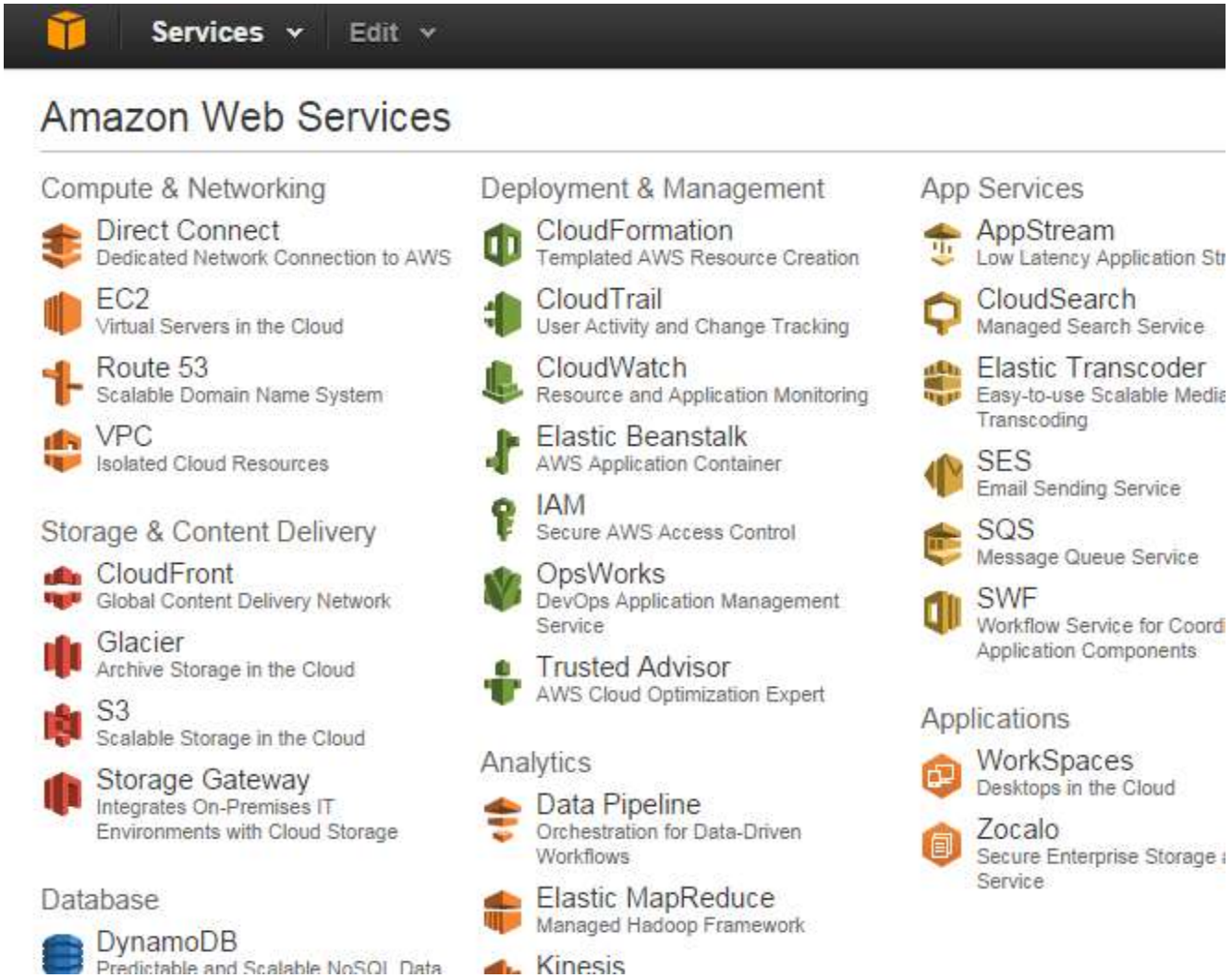
- The total volume of data and number of objects one can store are unlimited. Individual Amazon S3 objects can range in size from 1 byte to 5 terabytes. The largest object uploaded in a single PUT is 5 gigabytes.

Data Durability, Reliability, Versioning

- Amazon S3 provides a highly durable storage infrastructure.
- S3 redundantly stores data in multiple facilities and on multiple devices within each facility. Data is stored across multiple facilities before SUCCESS is returned. S3 calculates checksums on all network traffic to detect corruption of data packets when storing or retrieving data. S3 performs regular, systematic data integrity checks and is built to be automatically self-healing.
- 99.999999999% durability and 99.99% availability of objects over a given year. Designed to sustain the concurrent loss of data in two facilities.
- S3 provides further protection via Versioning. You can use Versioning to preserve, retrieve, and restore every version of every object stored in your S3 bucket. This allows easy recovery from both unintended user actions and application failures. By default, requests retrieve the most recent version. Older versions of an object can be retrieved by specifying a version in the request. Storage rates apply for every version stored.

Start using S3

- Go to aws.amazon.com
- Login into AWS Management Console
- On the Dashboard listing all AWS services select **S3**.



Storage Pricing

- Price of storing data at Amazon (S3) depends slightly on the region. USA prices are provided below.
- If one could allow reduce reliability (redundancy) of storage, price is some 20% lower.
- Glacier storage is considerably cheaper, however, getting data from Glacier storage might take hours

	Standard Storage	Reduced Redundancy Storage	Glacier Storage
First 1 TB / month	$0.0300 per GB	$0.0240 per GB	$0.0100 per GB
Next 49 TB / month	$0.0295 per GB	$0.0236 per GB	$0.0100 per GB
Next 450 TB / month	$0.0290 per GB	$0.0232 per GB	$0.0100 per GB
Next 500 TB / month	$0.0285 per GB	$0.0228 per GB	$0.0100 per GB
Next 4000 TB / month	$0.0280 per GB	$0.0224 per GB	$0.0100 per GB
Over 5000 TB / month	$0.0275 per GB	$0.0220 per GB	$0.0100 per GB

Data Transfer Pricing

- Data transfer "in" and "out" refers to transfer into and out of an Amazon S3 Region.

Data Transfer IN To Amazon S3	
All data transfer in	$0.000 per GB
Data Transfer OUT From Amazon S3 To	
Amazon EC2 in the Northern Virginia Region	$0.000 per GB
Another AWS Region or Amazon CloudFront	$0.020 per GB
Data Transfer OUT From Amazon S3 To Internet	
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

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 terabytes.
- 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. AWS charges 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

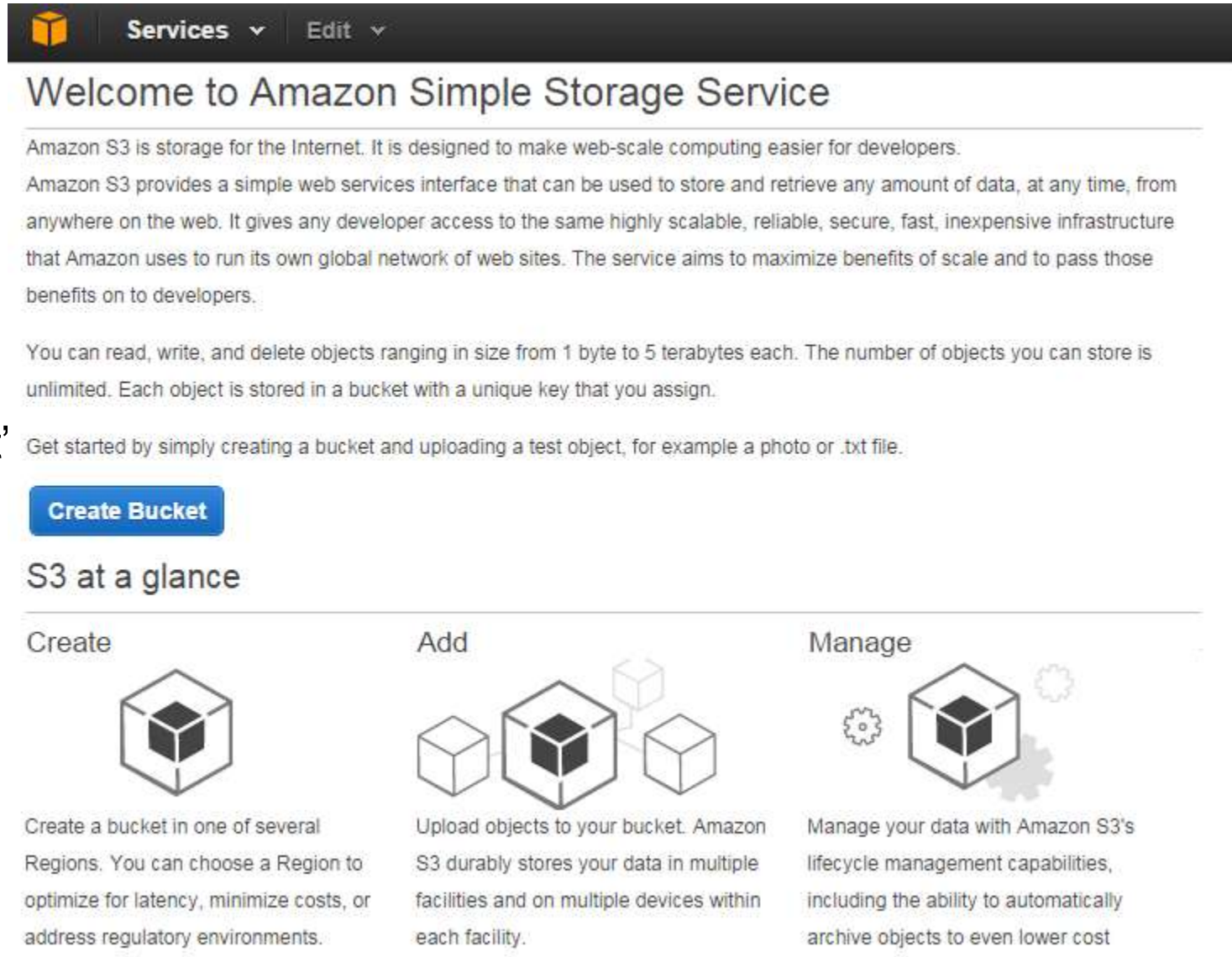
- `http://s3.amazonaws.com/mybucket/homepage.html`

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.

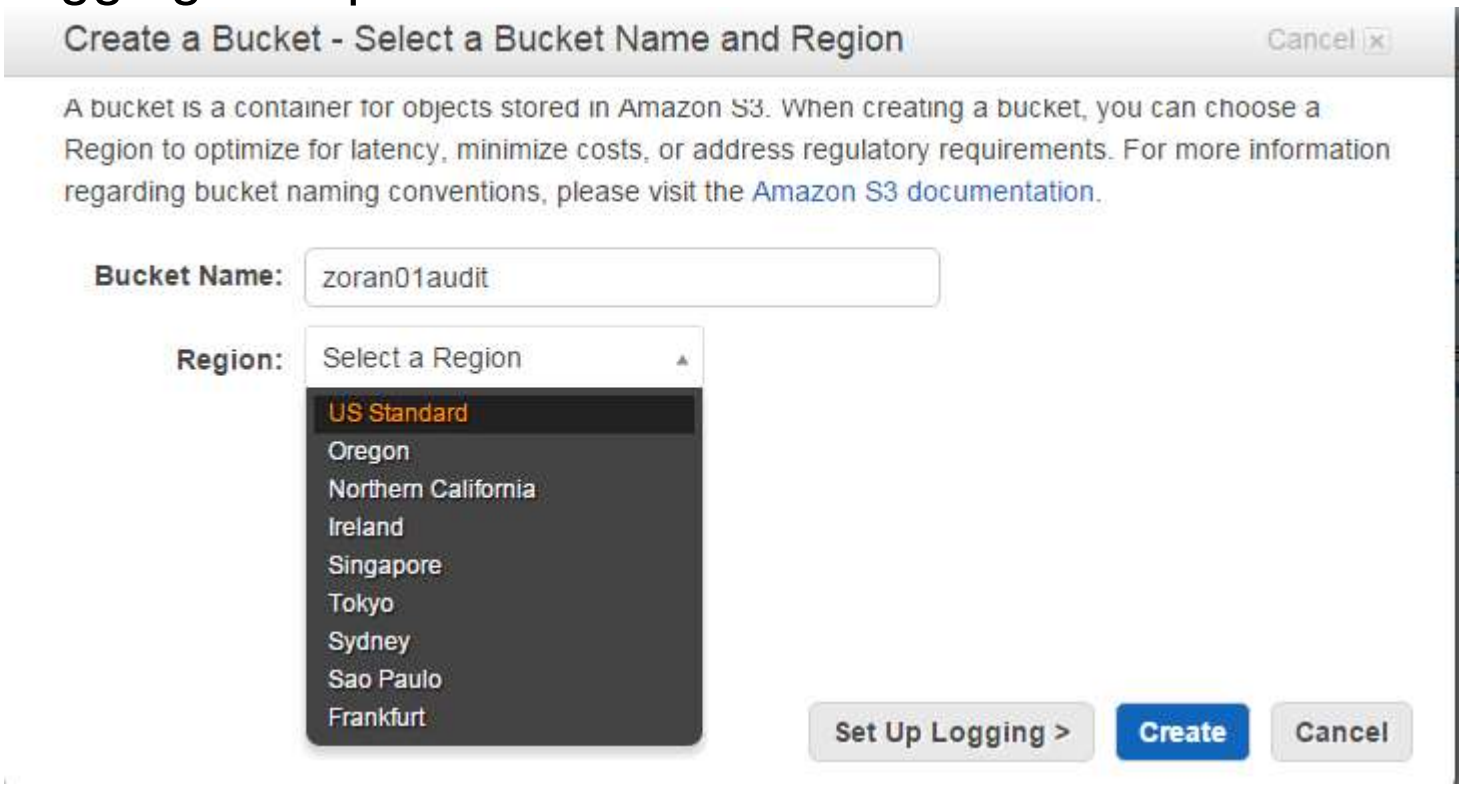
Select S3 Service, Create Bucket

- On AWS Management Console select S3 service.
- On the Welcome screen, select “Create Bucket’

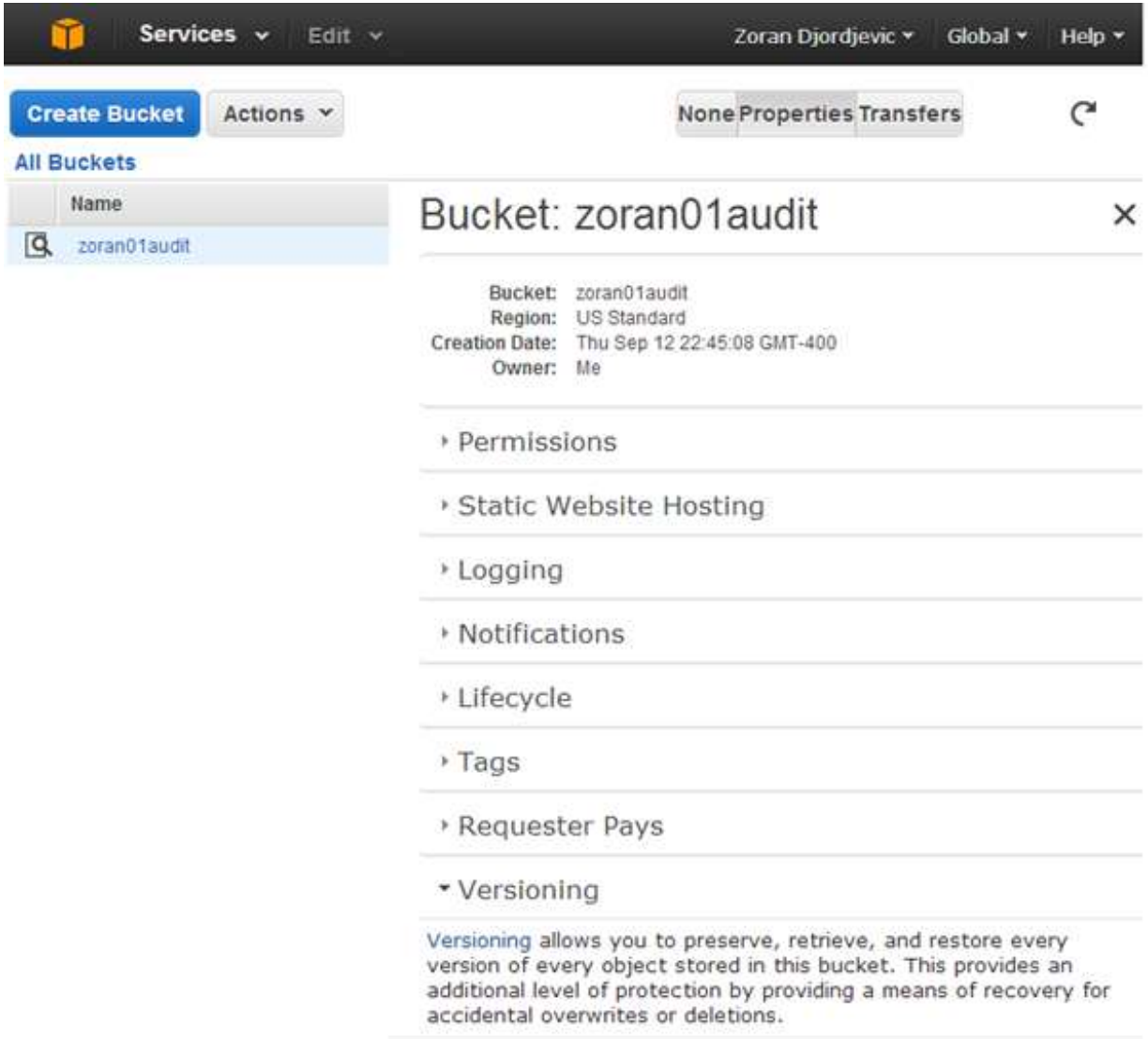


Create Bucket and Region

- Bucket name must be unique across entire Amazon S3.
- Select one of 10 available regions.
- In order to enable logging, we need an existing bucket. Let us call that bucket xx..01audit. That bucket does not have logging set up.

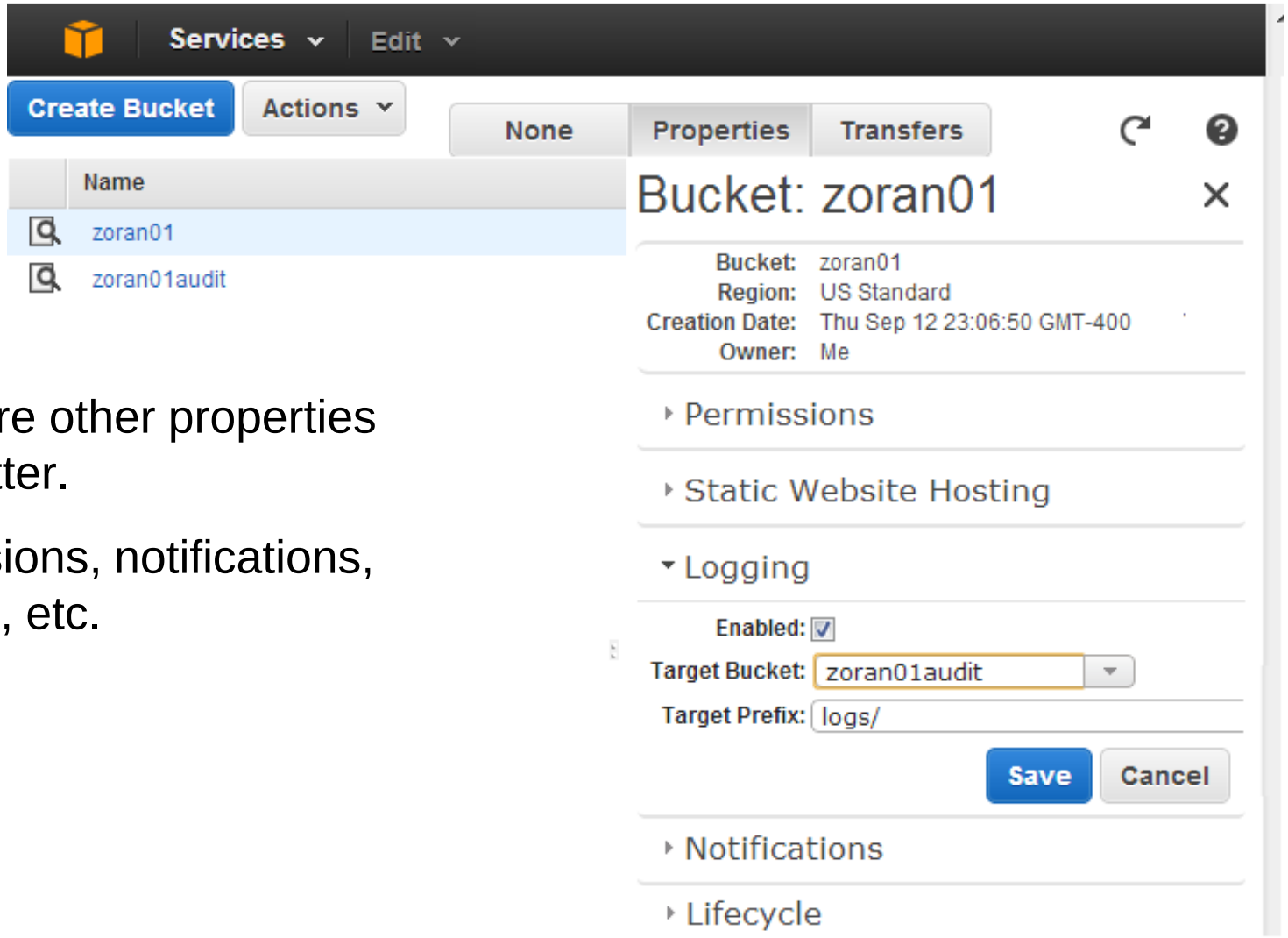


First Bucket is Created



- To create the second bucket we just hit Create Bucket button on this page.
- If we create new bucket with name zoran01, it will appear above the first bucket.

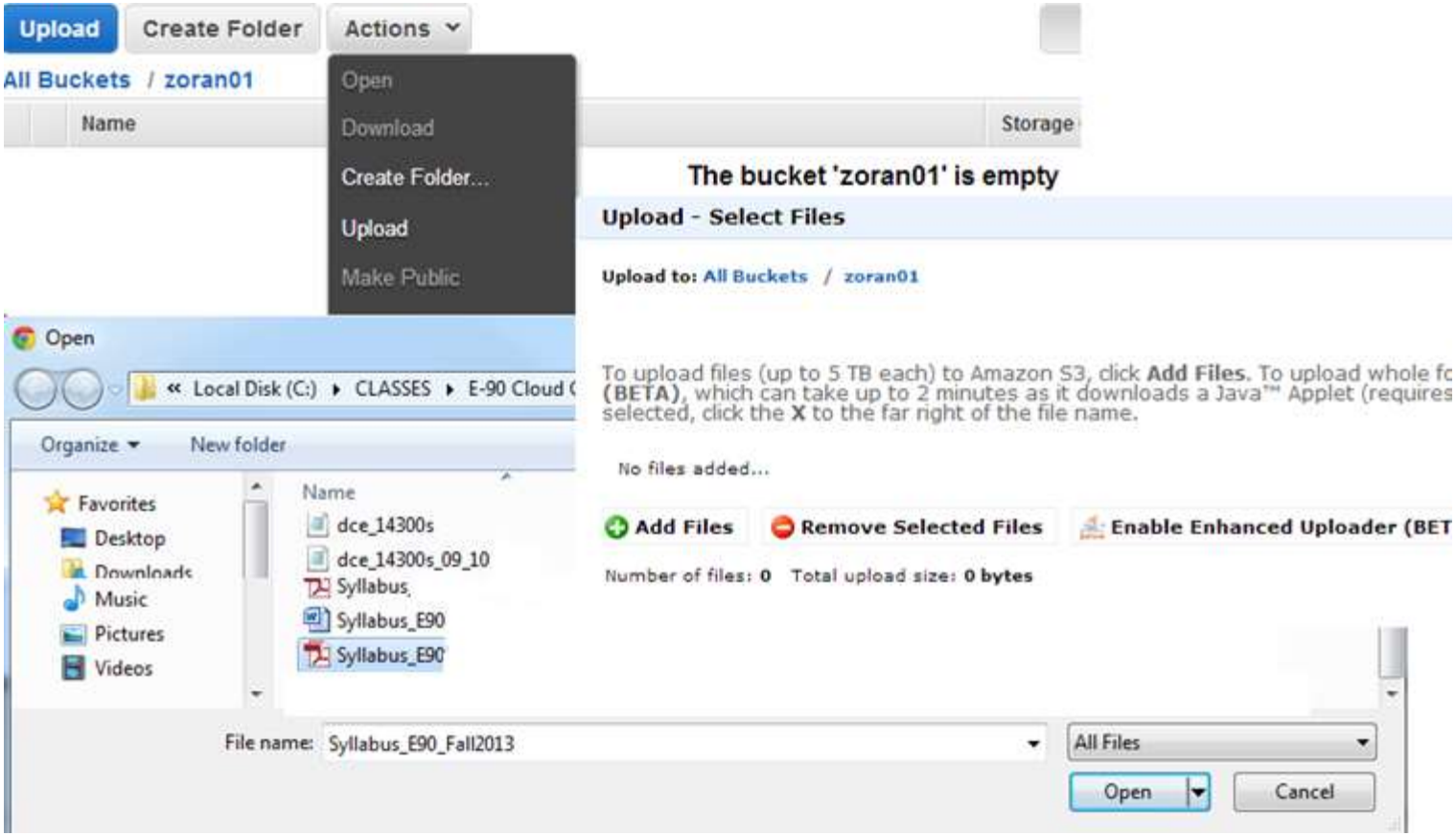
Modify Properties, Add Logging



- There are other properties that matter.
- Permissions, notifications, lifecycle, etc.

Upload a File, Any HTML or PDF file

- From Actions menu select Upload, on the new widget, select Add Files
- Your file(s), if not too big is(are) up in the bucket right away.



Try to Locate the file from a Browser

- Our file should be reachable at:

http://s3.amazonaws.com/zoran01/Syllabus_E90.pdf

- Open another browser (Mozilla) and try. We get back:

```
<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>2B289BE6F7282D14</RequestId>
  <HostId>v62x3+gRzISTfCZuK1Q2XqMLpblrgDa31YGseWgq0VCC2uk5bBVCWP4T
  </HostId>
</Error>
```

The issue is in permissions, most probably.

- To change permissions on the specific object in your bucket, highlight the object, select Properties in the top menu, then expand Permissions, and select Add more permissions.