

Assignment 02

Problem 1

1. Choose ‘S3’ in AWS Console. Create a bucket in S3 storage. Set region as ‘US Standard’.

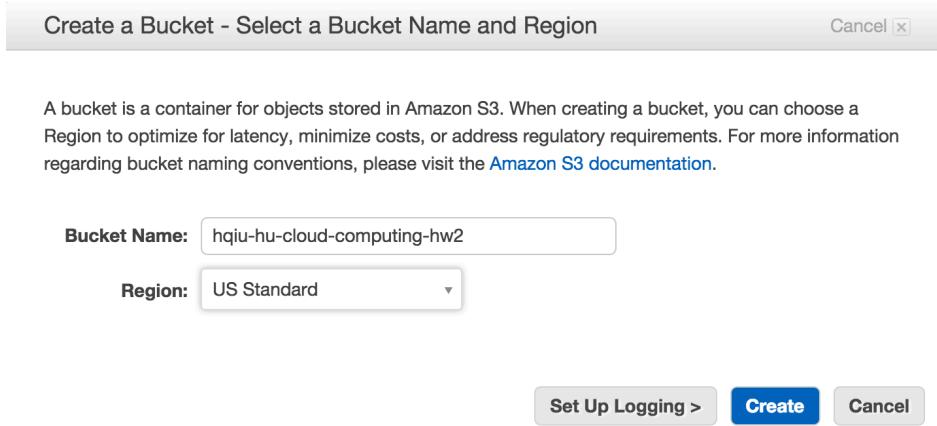
Create a Bucket - Select a Bucket Name and Region

A bucket is a container for objects stored in Amazon S3. When creating a bucket, you can choose a Region to optimize for latency, minimize costs, or address regulatory requirements. For more information regarding bucket naming conventions, please visit the [Amazon S3 documentation](#).

Bucket Name: hqiu-hu-cloud-computing-hw2

Region: US Standard

Set Up Logging > **Create** Cancel



2. Select ‘Create Folder’ and create a double nested folder (hqiu-hu-cloud-computing-hw2/pictures).

Upload Create Folder Actions ▾

All Buckets / hqiu-hu-cloud-computing-hw2

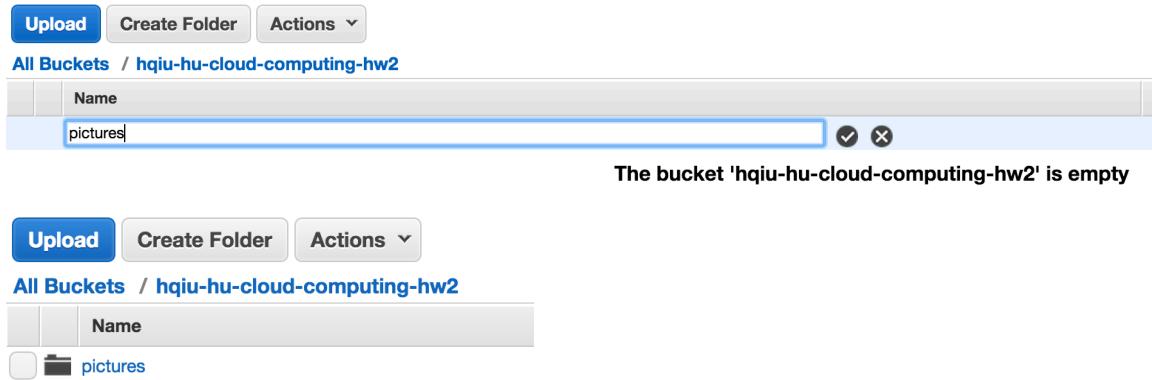
	Name
<input type="checkbox"/>	pictures

The bucket 'hqiu-hu-cloud-computing-hw2' is empty

Upload Create Folder Actions ▾

All Buckets / hqiu-hu-cloud-computing-hw2

	Name
<input type="checkbox"/>	 pictures



3. Upload a photograph. Go to ‘Actions’ and click ‘Upload’. Upload a picture called ‘cat.jpg’. Select the image, there’s a link about this image to enable viewing it online (through browser). In the default setting, the picture cannot be viewed from the browser (can’t be viewed by anyone except myself).

Screenshot of the AWS S3 console showing the 'Actions' dropdown menu over a folder named 'pictures'. The 'Upload' option is highlighted.

The folder 'pictures' is empty.

Upload - Select Files and Folders

Upload to: All Buckets / hqiu-hu-cloud-computing-hw2 / pictures

To upload files (up to 5 TB each) to Amazon S3, click **Add Files**. You can also drag and drop files and folders to the area below. To remove files already selected, click the X to the far right of the file name.

Drag and drop files and folders to upload here.

cat.jpg (51.9 KB)

Add Files Remove Selected Files

Number of files: 1 Total upload size: 51.9 KB

Set Details > Start Upload Cancel

Upload

All Buckets / hqiu-hu-cloud-computing-hw2 / pictures

	Name	Storage Class	Size	Last Modified
<input type="checkbox"/>	cat.jpg	Standard	51.9 KB	Tue Sep 15 19:15:35 GMT-400 2014

None Properties Transfers

Transfers

Automatically clear finished transfers

Done

Upload: Uploading cat.jpg to hqiu-hu-cloud-computing-hw2

https://s3.amazonaws.com/hqiu-hu-cloud-computing-hw2/pictures/cat.jpg

Apps Algo MachineLearning 1point3acres Programming OpenCourses temp

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>44C9EF7C0737B978</RequestId>
  <HostId>
    exsgpkaoFSuU6Wat8VmIXF4UDdeDuzA/MyMTNnsekTSBl2fEEMgIew8JtDRVpROdH5m7atGZNeU=
  </HostId>
</Error>
```

4. Change the properties of the image. Select the image and in the right hand corner of the S3 console page select ‘Properties’, go to ‘Permissions’. Add a new policy and enable it to be ‘open/download/view’ by ‘Everyone’. Then the image will be publicly accessible through the link below:

<https://s3.amazonaws.com/hqiu-hu-cloud-computing-hw2/pictures/cat.jpg>

Check through the browser.

Object: cat.jpg

Bucket: hqiu-hu-cloud-computing-hw2
 Folder: pictures
 Name: cat.jpg
 Link: <https://s3.amazonaws.com/hqiu-hu-cloud-computing-hw2/pictures/cat.jpg>
 Size: 53167

Last Modified: Tue Sep 15 19:26:09 GMT-400 2015
 Owner: glycine76
 ETAG: 0x89322050e74dbff81c02b334f049
 Expiry Date: None
 Expiration Rule: N/A

Details

Permissions

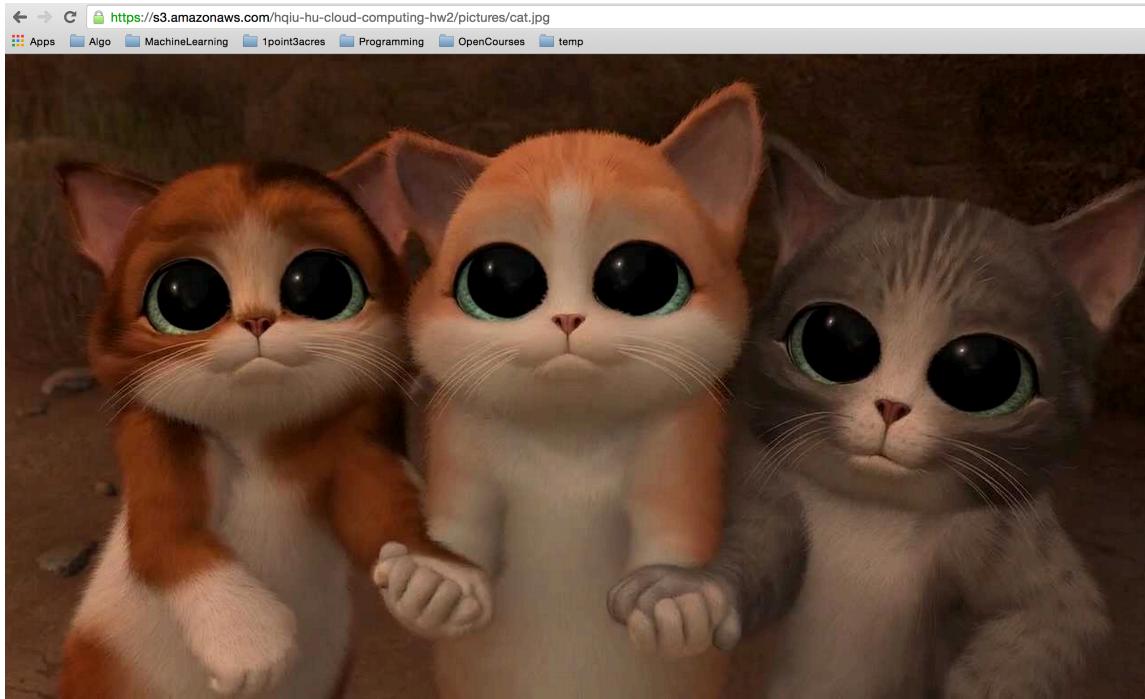
You can control access to the bucket and its contents using access policies. [Learn more](#).

Grantee: glycine76	<input checked="" type="checkbox"/> Open/Download	<input checked="" type="checkbox"/> View Permissions	<input checked="" type="checkbox"/> Edit Permissions	X
Grantee: Everyone	<input checked="" type="checkbox"/> Open/Download	<input checked="" type="checkbox"/> View Permissions	<input type="checkbox"/> Edit Permissions	X

Add more permissions

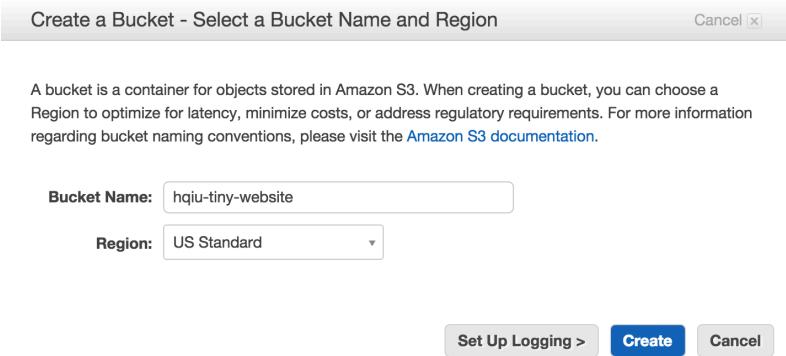
Save Cancel

Now we can view it through the browser!



Problem 2

1. Create a bucket in S3 storage for hosting a tiny web site. Set the region to ‘US Standard’.



2. Select this bucket and select 'Properties' on the right hand corner of the S3 console page. Select 'Static Website Hosting' and 'Enable website hosting'. Add the 'index.html' into 'Index Document' and 'error.html' into 'Error Document'. Upload these two files and other related files into this bucket. The end point of this website is:
<http://hqiutiny-website.s3-website-us-east-1.amazonaws.com/>
In the default setting, this website is not publicly accessible.

Bucket: hqiutiny-website X

Bucket: hqiutiny-website
Region: US Standard
Creation Date: Tue Sep 15 19:58:29 GMT-400 2015
Owner: glycine76

▶ **Permissions**

▼ **Static Website Hosting**

You can [host your static website](#) entirely on Amazon S3. Once you enable your bucket for static website hosting, all your content is accessible to web browsers via the Amazon S3 website endpoint for your bucket.

Endpoint: hqiutiny-website.s3-website-us-east-1.amazonaws.com

Each bucket serves a website namespace (e.g. "www.example.com"). Requests for your host name (e.g. "example.com" or "www.example.com") can be routed to the contents in your bucket. You can also redirect requests to another host name (e.g. redirect "example.com" to "www.example.com"). See our [walkthrough](#) for how to set up an Amazon S3 static website with your host name.

Do not enable website hosting

Enable website hosting

Index Document: index.html

Error Document: error.html

▶ **Edit Redirection Rules:** You can set custom rules to automatically redirect web page requests for specific content.

Upload Create Folder Actions ▾

All Buckets / hqiu-tiny-website

	Name	Storage Class	Size	Last Modified
<input type="checkbox"/>	error.html	Standard	474 bytes	Tue Sep 15 20:01:47 GMT-400 2
<input checked="" type="checkbox"/>	index.html	Standard	761 bytes	Tue Sep 15 20:13:25 GMT-400 2
<input type="checkbox"/>	minion.jpg	Standard	20 KB	Tue Sep 15 20:01:48 GMT-400 2
<input type="checkbox"/>	minions.jpg	Standard	217.3 KB	Tue Sep 15 20:01:48 GMT-400 2



403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: 1A30BF8FBDA33A4
- HostId: 584E+DNAJ5KGKVGHCSzHu/nwl2cxvlJakip8xh/tpNCHSFj7rNOmG/qe9V2vhj1l

An Error Occurred While Attempting to Retrieve a Custom Error Document

- Code: AccessDenied
- Message: Access Denied

3. Select the ‘Properties’ and select ‘Permissions’. Add another policy, which gives the ‘List’ and ‘View Permission’ to ‘Everyone’. Also, add this policy to each of the files in this bucket (index.html, error.html, etc.). Then this page can be publicly accessible through:

<http://hqiu-tiny-website.s3-website-us-east-1.amazonaws.com>

Bucket: hqiu-tiny-website X

Bucket: hqiu-tiny-website
Region: US Standard
Creation Date: Tue Sep 15 19:58:29 GMT-400 2015
Owner: glycine76

▼ Permissions

You can control access to the bucket and its contents using access policies. [Learn more.](#)

Grantee: glycine76	<input checked="" type="checkbox"/> List	<input checked="" type="checkbox"/> Upload/Delete	<input checked="" type="checkbox"/> View Permissions	<input checked="" type="checkbox"/>	Edit Permissions
Grantee: Everyone	<input checked="" type="checkbox"/> List	<input type="checkbox"/> Upload/Delete	<input checked="" type="checkbox"/> View Permissions	<input type="checkbox"/>	Edit Permissions

[Add more permissions](#) [Add bucket policy](#) [Add CORS Configuration](#)

[Save](#) [Cancel](#)

Screenshot of the AWS S3 console showing the 'hqiutiny-website' bucket. The 'index.html' file is selected. The object details pane shows the following information:

Name	Storage Class	Size	Last Modified
error.html	Standard	474 bytes	Tue Sep 15 20:01:47 GMT-400 2015
index.html	Standard	761 bytes	Tue Sep 15 20:13:25 GMT-400 2015
minion.jpg	Standard	20 KB	Tue Sep 15 20:01:48 GMT-400 2015
minions.jpg	Standard	217.3 KB	Tue Sep 15 20:01:48 GMT-400 2015

Object: index.html

Bucket: hqiutiny-website
Name: index.html
Link: <https://s3.amazonaws.com/hqiutiny-website/index.html>
Size: 761
Last Modified: Tue Sep 15 20:13:25 GMT-400 2015
Owner: glycine76
ETag: 07087ff63d1bba139cb57c06fd231fd
Expiry Date: None
Expiration Rule: N/A

Details

Permissions

You can control access to the bucket and its contents using access policies. [Learn more](#).

Grantee: glycine76 Open/Download View Permissions Edit Permissions X

Grantee: Everyone Open/Download View Permissions Edit Permissions X

Add more permissions

Save Cancel

Now we can view it through the browser!

Screenshot of a web browser displaying the website at hqiutiny-website.s3-website-us-east-1.amazonaws.com. The page title is "Sample 'Hello, World' Application". On the left, there is a small image of a Minion character.

This is the home page for the HelloWorld Web application.

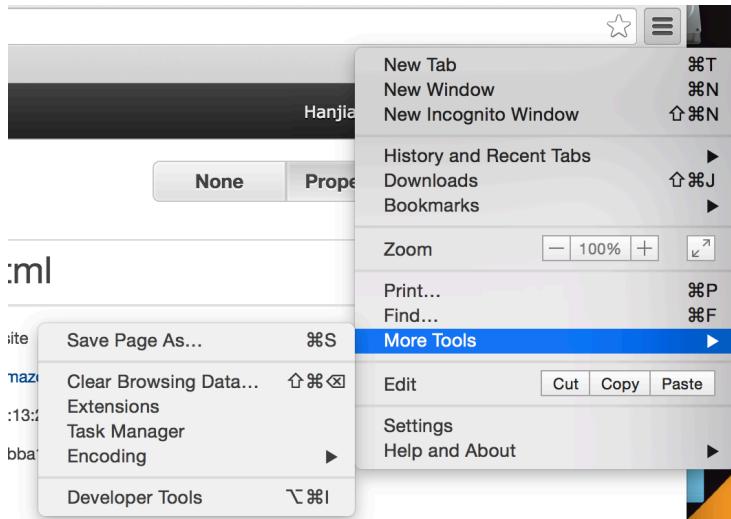
To prove that they work, you can execute either of the following links:

- To a [Google homepage](#).
- To a [Facebook homepage](#).



Problem 3

1. I use the Google Chrome Network Monitoring Tool here. Select ‘More Tools’ on the top right corner of the browser and select ‘Developer Tools’. Once in display, this tool will start monitoring. I will start this tool just before uploading the file into the S3 bucket. Otherwise it will record other traffic going through.



2. Go to my S3 bucket. Try to upload an image called ‘mickey.jpg’ into the bucket ‘hqiu-hu-cloud-computing-hw2/pictures’. Before I select ‘Start Upload’, open the ‘Developer Tools’. The right part displays the monitoring.

A screenshot of the AWS S3 console. On the left, there's a navigation bar with 'AWS', 'Services', 'Edit', and a dropdown for 'HanJiao Qiu'. Below that is a toolbar with 'load', 'Create Folder', 'Actions', and buttons for 'None', 'Properties', and 'Transfers'. A modal window titled 'Upload - Select Files and Folders' is open, showing a file named 'mickey.jpg (7.1 KB)' and buttons for 'Add Files' and 'Remove Selected Files'. It also shows 'Number of files: 1 Total upload size: 7.1 KB'. On the right side of the screen, the developer tools Network tab is open, showing a timeline of network requests. The timeline has columns for 'Name', 'St...', 'Ty...', 'Initiator', 'Size', 'Ti...', and 'Timeline - Start Time'. There are several entries in the timeline, with the last one being a large PUT request for the uploaded file.

3. Select ‘Start Upload’. We can see all the traffic going through on the right side window. The following snapshot shows a sequence of requests when upload the image into the S3 bucket. The action ‘CreateBlob’ sends a ‘HTTPS PUT’ request

and gets ‘200 OK’ as the response. This is the step to upload the file. In the Request Headers, the ‘Content-Length’ is 7309 bytes and the ‘Content-Type’ is image/jpeg. This exactly is the image that I’ve uploaded. The ‘mickey.jpg’ itself is 7.1 kb. With some headers, the ‘Content-Length’ is about 7300 bytes.

Name	Status	Type	Initiator	Size	Timeline - Start Time
V4Sign	200	xhr	Console	1....	1ms
&k0=Expires...	200	gif	Other	33....	1ms
CreateBlob	200	xhr	Console	63....	1ms
DeliverHttp	200	xhr	Console	1....	3ms
Tattle	200	xhr	AB5F5...	52....	9ms

5 / 6 requests | 4.0 KB / 4.3 KB transferred

Headers Preview Response Cookies Timing

Remote Address: 72.21.215.92:443
Request URL: https://s3-console-us-standard.console.aws.amazon.com/CreateBlo
b
Request Method: PUT
Status Code: 200 OK

Response Headers view source

Connection: keep-alive
Content-Length: 0
server: S2ConsoleWebServer/1.0
Set-Cookie: s3.console.ws.security.aws=mtp#0:A64FF27D-A713-47FA-8B54-B724FADD!
62A.0:1442358627936; path=/; expires=Wed, 1 Jan 2031 16:47:08 +0000; domain=.c
nsole.aws.amazon.com; secure
x-s2-cws-hid: 20120601:1:KqVKxMNqjU3ETeMhP7t8fw==:I1HDy/ZDLGhCWOnNuv83JAeFur
HD0r7Dp/CMJfoaZ1R8naRcwC8aqGigudJWhchA2Xsq1l2xJtbkL91CKqQ==
x-s2-cws-rid: jMAUgLGUKa5u5jYUvsW0j7tKdpyzvYl1DvIXpeHPusabhUuhGjDd8yjjIOHj0CF
E
x-s2-ks-blob-digest: Md5Hex:668427e04e92e625f11eac5e20ea3d27
x-s3-console-remote-file-digest: Md5Hex:668427e04e92e625f11eac5e20ea3d27

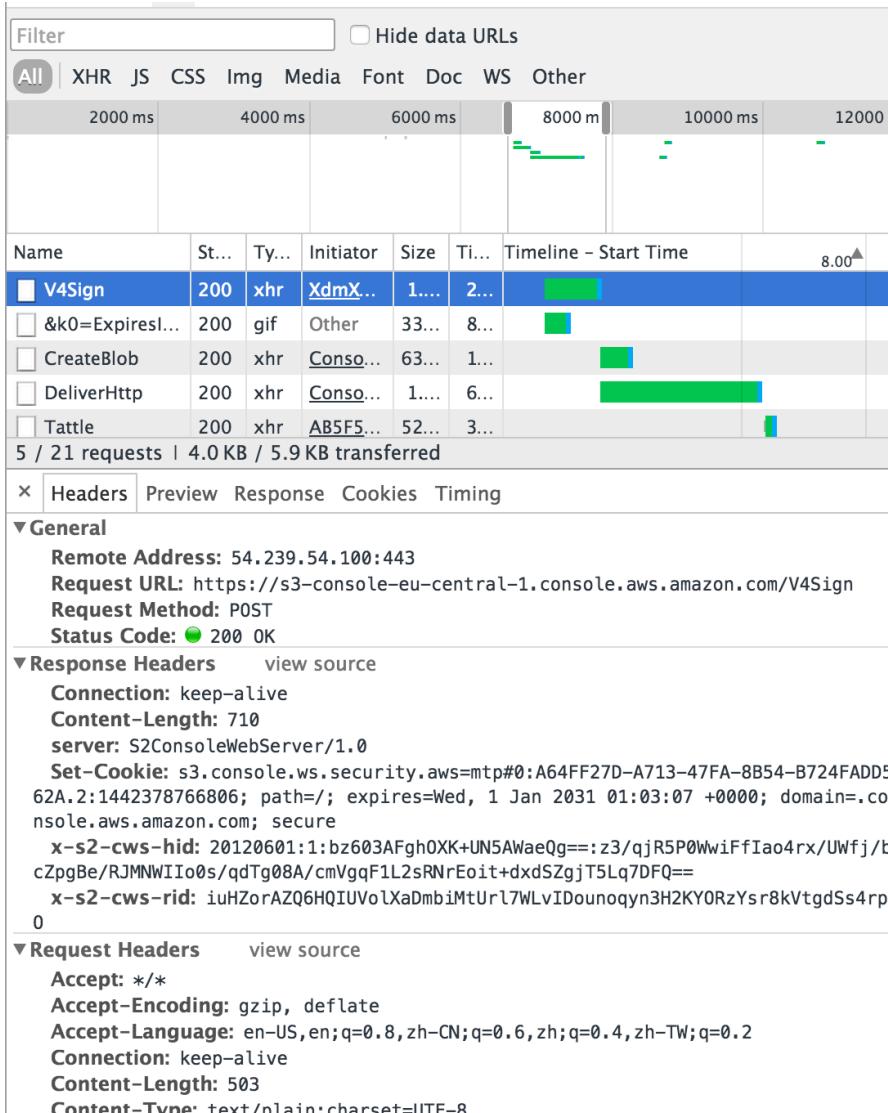
Request Headers view source

Accept: */*
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,zh-CN;q=0.6,zh;q=0.4,zh-TW;q=0.2
Connection: keep-alive
Content-Length: 7309
Content-Type: image/jpeg

All Buckets / hqiu-hu-cloud-computing-hw2 / pictures

	Name	Storage Class	Size	Last Modified
<input type="checkbox"/>	cat.jpg	Standard	51.9 KB	Tue Sep
<input type="checkbox"/>	mickey.jpg	Standard	7.1 KB	Tue Sep

4. There are some other requests such as ‘PUT’ and “POST”. These are the steps to communicate/negotiate with the server to prepare for the upload and exchange some credential information. Since it uses port 443, it uses ‘HTTPS’ for secure data transfer.



5. I also upload another image called ‘minnie.jpg’, which is about 23.1 kb. It gives similar results as the previous step. In the ‘HTTPS PUT’ request, the ‘Content-Length’ is 23722 bytes, which is consistent with the image size. Here are the links to the images:

<https://s3.amazonaws.com/hqiuhu-cloud-computing-hw2/pictures/mickey.jpg>

<https://s3.amazonaws.com/hqiuhu-cloud-computing-hw2/pictures/minnie.jpg>

Filter Hide data URLs

All | XHR JS CSS Img Media Font Doc WS Other

2000 ms 4000 ms 6000 ms 8000 ms 10000 ms 12000 ms

Name	St...	Ty...	Initiator	Size	Ti...	Timeline – Start Time	
V4Sign	200	xhr	XdmX...	1....	2...	<div style="width: 50%;"> </div>	8.00
&k0=Expiresl...	200	gif	Other	33...	8...	<div style="width: 10%;"> </div>	
CreateBlob	200	xhr	Conso...	63...	1...	<div style="width: 100%; background-color: #0072bc;"> </div>	
DeliverHttp	200	xhr	Conso...	1....	6...	<div style="width: 80%;"> </div>	
Tattle	200	xhr	AB5F5...	52...	3...	<div style="width: 10%;"> </div>	

5 / 21 requests | 4.0 KB / 5.9 KB transferred

Headers Preview Response Cookies Timing

▼ General

Remote Address: 176.32.102.239:443
Request URL: https://s3-console-us-standard.console.aws.amazon.com/CreateBlo

b
Request Method: PUT
Status Code: 200 OK

▼ Response Headers [view source](#)

Connection: keep-alive
Content-Length: 0
server: S2ConsoleWebServer/1.0
Set-Cookie: s3.console.ws.security.aws=mtp#0:A64FF27D-A713-47FA-8B54-B724FADD562A.2:1442378766806; path=/; expires=Wed, 1 Jan 2031 16:59:20 +0000; domain=.co
nsole.aws.amazon.com; secure
x-s2-cws-hid: 20120601:1:eUM3sB2MDc2ewuKVCq5oPQ==:7wMsIsjh0k5kWvUTY+j6H3TeaWW
BfZYQv1Ei5Qn20C9zSJP/94G0UAg9GqLMiEpWmSv/XvjkZJSKnGE1MGXm4==
x-s2-cws-rid: r4K4V6wYUGkuLqkxszP4zxnNaiUcmUQFDk6IgHSzXqaQZujlIWMAbwwhyqZ1NCf
h
x-s2-ks-blob-digest: Md5Hex:7496db29132e40c29b7541e5a355bddb
x-s3-console-remote-file-digest: Md5Hex:7496db29132e40c29b7541e5a355bddb

▼ Request Headers [view source](#)

Accept: */*
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,zh-CN;q=0.6,zh;q=0.4,zh-TW;q=0.2
Connection: keep-alive
Content-Length: 23722

Content-Type: image/jpeg
Cookie: x-wl-uid=1AGlwF1t4TFgWB81n8YEJ3PxnpVlr5p2owCVJgnCgpQCzt4Nblc9tRFGjjPJF
MvytdAKAS3bdaRjgNk+4AVV8kXgR2Ct3bX+RUjAjV0Bq9YfAsu7tbfSKUgbL6J1ejdy+6HRhxmsTPb
I=; aws_lang=en; aws-target-static-id=1440715993952-559167; skin=noskin; x-mai
n="IwPNcUqKDpKjUxnJ2RG9kLozY3623ak@"; at-main=5|E3qoGgJF0vnJeh0zlmwu10LmIY51TCv
0hHsw710HvMXuzstQxm66W/6pKxYBmaRKLBDg08mjL2sPo2dIyE8QMvnAQ2F3z0XkkiqR/zjMzXsZR

Upload **Create Folder** **Actions ▾**

All Buckets / hqiu-hu-cloud-computing-hw2 / pictures

	Name	Storage Class	Size	Last Modified
<input type="checkbox"/>	cat.jpg	Standard	51.9 KB	Tue Sep 15 19:26:09 GMT-400 2
<input type="checkbox"/>	mickey.jpg	Standard	7.1 KB	Tue Sep 15 22:06:05 GMT-400 2
<input checked="" type="checkbox"/>	minnie.jpg	Standard	23.1 KB	Wed Sep 16 00:48:11 GMT-400

Steps to install AWS CLI Tools

1. Install the AWS CLI using the bundled installer on OS X. Do the prerequisites, check the Python installation.

```
hqiu@bos-mpdei>> python --version
Python 2.7.10
```

2. Install the AWS CLI using the bundled installer. The install script runs under the system default version of Python (here is Python 2.7.10).

```
hqiu@bos-mpdei>> cd AWS_CLI/
hqiu@bos-mpdei>> ls
hqiu@bos-mpdei>> curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
% Total    % Received % Xferd  Average Speed   Time   Time   Current
          Dload  Upload   Total Spent   Left  Speed
100 6109k  100 6109k    0     0  7601k      0 --:--:-- --:--:-- --:--:-- 16.3M
hqiu@bos-mpdei>> ls
awscli-bundle.zip
hqiu@bos-mpdei>> unzip awscli-bundle.zip
Archive: awscli-bundle.zip
  inflating: awscli-bundle/install
  inflating: awscli-bundle/packages/six-1.9.0.tar.gz
  inflating: awscli-bundle/packages/simplejson-3.3.0.tar.gz
  inflating: awscli-bundle/packages/python-dateutil-2.4.2.tar.gz
  inflating: awscli-bundle/packages/ordereddict-1.1.tar.gz
  inflating: awscli-bundle/packages/pyasn1-0.1.8.tar.gz
  inflating: awscli-bundle/packages/colorama-0.3.3.tar.gz
  inflating: awscli-bundle/packages/botocore-1.2.1.tar.gz
  inflating: awscli-bundle/packages/docutils-0.12.tar.gz
  inflating: awscli-bundle/packages/rsa-3.1.4.tar.gz
  inflating: awscli-bundle/packages/argparse-1.2.1.tar.gz
  inflating: awscli-bundle/packages/jmespath-0.7.1.tar.gz
  inflating: awscli-bundle/packages/virtualenv-13.0.3.tar.gz
  inflating: awscli-bundle/packages/awscli-1.8.3.tar.gz

hqiu@bos-mpdei>> ls
awscli-bundle      awscli-bundle.zip
hqiu@bos-mpdei>> sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
Password:
Running cmd: /Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS/Python virtualenv.py --python /Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS/Python /usr/local/aws
Running cmd: /usr/local/aws/bin/pip install --no-index --find-links file:///Users/hqiu/Documents/HarvardExtension/Cloud_Computing/AWS_CLI/awscli-bundle/packages awscli-1.8.3.tar.gz
You can now run: /usr/local/bin/aws --version
hqiu@bos-mpdei>> █

hqiu@bos-mpdei>> aws --version
aws-cli/1.8.3 Python/2.7.10 Darwin/14.4.0
```

3. Test the AWS CLI installation.

```

hqiu@bos-mpdei>> aws list
usage: aws [options] <command> <subcommand> [parameters]
aws: error: argument command: Invalid choice, valid choices are:

autoscaling           | cloudformation
cloudfront            | cloudhsm
cloudsearch            | cloudsearchdomain
cloudtrail             | cloudwatch
codecommit              | codepipeline
cognito-identity      | cognito-sync
datapipeline           | devicefarm
directconnect          | ds
dynamodb               | dynamodbstreams
ec2                     | ecs
efs                      | elasticache
elasticbeanstalk        | elastictranscoder
elb                      | emr
glacier                  | iam
importexport            | kinesis
kms                      | lambda
logs                      | machinelearning
opsworks                 | rds
redshift                  | route53
route53domains           | sdb
ses                      | sns
sns                      | ssm
storagegateway           | sts
support                  | swf
workspaces                | s3api
s3                         | configure
deploy                   | configservice
help

```

4. Configuration. Set environment variables and credentials in my '.bash_profile'.

```

# AWS CREDENTIALS
export EC2_CERT=/Users/hqiu/Documents/HarvardExtension/Cloud_Computing/AWS/cert-53WR3UN6I
2U4EPMCGV7G06APA0XKERJP.pem
export EC2_PRIVATE_KEY=/Users/hqiu/Documents/HarvardExtension/Cloud_Computing/AWS/pk-53WR
3UN6I2U4EPMCGV7G06APA0XKERJP.pem
export PATH=$PATH:/usr/local/aws/bin
export AWS_ACCESS_KEY_ID=AKIAIABZEJE32N2IXMZA
export AWS_SECRET_ACCESS_KEY=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
export AWS_DEFAULT_REGION=us-east-1

~
~

"~/./.bash_profile" 26L, 907C written

```

5. Test the AWS CLI configuration.

```

hqiu@bos-mpdei>> aws ec2 describe-regions --output table
+-----+-----+
|       DescribeRegions      |
+-----+-----+
|       Regions              |
+-----+-----+
|   Endpoint    | RegionName |
+-----+-----+
| ec2.eu-west-1.amazonaws.com | eu-west-1 |
| ec2.ap-southeast-1.amazonaws.com | ap-southeast-1 |
| ec2.ap-southeast-2.amazonaws.com | ap-southeast-2 |
| ec2.eu-central-1.amazonaws.com | eu-central-1 |
| ec2.ap-northeast-1.amazonaws.com | ap-northeast-1 |
| ec2.us-east-1.amazonaws.com | us-east-1 |
| ec2.sa-east-1.amazonaws.com | sa-east-1 |
| ec2.us-west-1.amazonaws.com | us-west-1 |
| ec2.us-west-2.amazonaws.com | us-west-2 |
+-----+-----+
hqiu@bos-mpdei>> █

```

Steps to install AWS EC2 CLI Tools

1. Download and install the AWS EC2 CLI Tools.

```

hqiu@bos-mpdei>> cd AWS_EC2_CLI/
hqiu@bos-mpdei>> curl -O http://s3.amazonaws.com/ec2-downloads/ec2-api-tools.zip
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload Total   Spent   Left Speed
100 16.7M  100 16.7M    0     0  20.1M      0 ---:--- ---:--- ---:--- 44.5M
hqiu@bos-mpdei>> ls -lart
total 34312
drwxr-xr-x  7 hqiu  600      238 Sep 15 10:57 ..
-rw-r--r--  1 hqiu  600  17566389 Sep 15 10:57 ec2-api-tools.zip
drwxr-xr-x  3 hqiu  600      102 Sep 15 10:57 .
hqiu@bos-mpdei>> █

hqiu@bos-mpdei>> sudo mkdir /usr/local/ec2
Password:
hqiu@bos-mpdei>> sudo unzip ec2-api-tools.zip -d /usr/local/ec2
Archive: ec2-api-tools.zip
  creating: /usr/local/ec2/ec2-api-tools-1.7.5.1/
  inflating: /usr/local/ec2/ec2-api-tools-1.7.5.1/THIRDPARTYLICENSE.TXT
  creating: /usr/local/ec2/ec2-api-tools-1.7.5.1/bin/
  inflating: /usr/local/ec2/ec2-api-tools-1.7.5.1/bin/ec2-accept-vpc-peering-connection
  inflating: /usr/local/ec2/ec2-api-tools-1.7.5.1/bin/ec2-accept-vpc-peering-connection.cmd
  inflating: /usr/local/ec2/ec2-api-tools-1.7.5.1/bin/ec2-activate-license
  inflating: /usr/local/ec2/ec2-api-tools-1.7.5.1/bin/ec2-activate-license.cmd
  inflating: /usr/local/ec2/ec2-api-tools-1.7.5.1/bin/ec2-add-group
  inflating: /usr/local/ec2/ec2-api-tools-1.7.5.1/bin/ec2-add-group.cmd
  inflating: /usr/local/ec2/ec2-api-tools-1.7.5.1/bin/ec2-add-keypair
  inflating: /usr/local/ec2/ec2-api-tools-1.7.5.1/bin/ec2-add-keypair.cmd
  inflating: /usr/local/ec2/ec2-api-tools-1.7.5.1/bin/ec2-allocate-address
  inflating: /usr/local/ec2/ec2-api-tools-1.7.5.1/bin/ec2-allocate-address.cmd
  inflating: /usr/local/ec2/ec2-api-tools-1.7.5.1/bin/ec2-assign-private-ip-addresses
  inflating: /usr/local/ec2/ec2-api-tools-1.7.5.1/bin/ec2-assign-private-ip-addresses.cmd

```

2. Tell the tools where Java lives. Set the 'JAVA_HOME' variable.

```

hqiu@bos-mpdei>> which java
/usr/bin/java
hqiu@bos-mpdei>> /usr/libexec/java_home
/Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home
hqiu@bos-mpdei>> export JAVA_HOME=$(/usr/libexec/java_home)
hqiu@bos-mpdei>> echo $JAVA_HOME
/Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home

```

Verify the ‘JAVA_HOME’ setting.

```
hqiu@bos-mpdei>> $JAVA_HOME/bin/java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)
```

Set the ‘JAVA_HOME’ in ‘~/.bash_profile’. Open a new tab, verify the setting.

```
vim
if [ -f ~/.bashrc ]; then
    source ~/.bashrc
fi

export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home
export M2_HOME=/Applications/apache-maven-3.2.5
export PATH=$PATH:$M2_HOME/bin
```

3. Set the ‘EC2_HOME’ and ‘PATH’ environment variables. Tell the CLI Tools where they live.

```
hqiu@bos-mpdei>> export EC2_HOME=/usr/local/ec2/ec2-api-tools-1.7.5.1
hqiu@bos-mpdei>> echo $EC2_HOME
/usr/local/ec2/ec2-api-tools-1.7.5.1

hqiu@bos-mpdei>> echo $PATH
/Users/hqiu/anaconda/bin:/usr/local/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/opt/X11/bin:/usr
/texbin:/Applications/apache-maven-3.2.5/bin
hqiu@bos-mpdei>> export PATH=$PATH:$EC2_HOME/bin
hqiu@bos-mpdei>> echo $PATH
/Users/hqiu/anaconda/bin:/usr/local/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/opt/X11/bin:/usr
/texbin:/Applications/apache-maven-3.2.5/bin:/usr/local/ec2/ec2-api-tools-1.7.5.1/bin
```

4. The credential information has already been setup in previous steps. Verify the tool setup.

```
hqiu@bos-mpdei>> ec2-describe-regions
REGION eu-west-1      ec2.eu-west-1.amazonaws.com
REGION ap-southeast-1 ec2.ap-southeast-1.amazonaws.com
REGION ap-southeast-2 ec2.ap-southeast-2.amazonaws.com
REGION eu-central-1   ec2.eu-central-1.amazonaws.com
REGION ap-northeast-1 ec2.ap-northeast-1.amazonaws.com
REGION us-east-1      ec2.us-east-1.amazonaws.com
REGION sa-east-1      ec2.sa-east-1.amazonaws.com
REGION us-west-1      ec2.us-west-1.amazonaws.com
REGION us-west-2      ec2.us-west-2.amazonaws.com
```

Problem 4

1. Here I will use the general ‘AWS CLI’ tool to do the following problems. Although we’ve already stored the environment variables and the credential information into ‘.bash_profile’, I still try the quick configuration command here. Just for getting familiar with different commands. Check the configuration.

```

hqiu@bos-mpdei>> aws configure
AWS Access Key ID [None]: AKIAIABZEJE32N2IXMZA
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json

hqiu@bos-mpdei>> cat ~/.aws/config
[default]
output = json
region = us-east-1
hqiu@bos-mpdei>> cat ~/.aws/credentials
[default]
aws_access_key_id = AKIAIABZEJE32N2IXMZA
aws_secret_access_key = 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

```

2. Create a new key pair using AWS CLI tools. It allows us to connect to the instance. Change the permission of the key file.

```

hqiu@bos-mpdei>> aws ec2 create-key-pair --key-name ec2hqiу --query 'KeyMaterial' --out
put text > ~/Documents/HarvardExtension/Cloud_Computing/AWS/ec2hqiу.pem
hqiu@bos-mpdei>> cd ~/Documents/HarvardExtension/Cloud_Computing/AWS
hqiu@bos-mpdei>> ls -lart ec2hqiу.pem
-rw-r--r-- 1 hqiу 600 1675 Sep 15 15:43 ec2hqiу.pem
hqiu@bos-mpdei>> sudo chmod 400 ec2hqiу.pem
Password:

```

3. Create a new security group and add rules that allow incoming traffic over port 22 for SSH, port 80 for HTTP and port 443 for HTTPS.

```

hqiu@bos-mpdei>> aws ec2 create-security-group --group-name launch-hqiу --description "security group for development environment in EC2"
{
    "GroupId": "sg-adfff3ca"
}
hqiu@bos-mpdei>> aws ec2 authorize-security-group-ingress --group-name launch-hqiу --pr
otocol tcp --port 22 --cidr 0.0.0.0/0
hqiu@bos-mpdei>> aws ec2 authorize-security-group-ingress --group-name launch-hqiу --pr
otocol tcp --port 80 --cidr 0.0.0.0/0
hqiu@bos-mpdei>> aws ec2 authorize-security-group-ingress --group-name launch-hqiу --pr
otocol tcp --port 443 --cidr 0.0.0.0/0

```

```

hqiu@bos-mpdei>> aws ec2 describe-security-groups --group-name launch-hqiu
{
    "SecurityGroups": [
        {
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "UserIdGroupPairs": [],
                    "PrefixListIds": []
                }
            ],
            "Description": "security group for development environment in EC2",
            "IpPermissions": [
                {
                    "PrefixListIds": [],
                    "FromPort": 443,
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "ToPort": 443,
                    "IpProtocol": "tcp",
                    "UserIdGroupPairs": []
                },
                {
                    "PrefixListIds": [],
                    "FromPort": 80,
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "ToPort": 80,
                    "IpProtocol": "tcp",
                    "UserIdGroupPairs": []
                },
                {
                    "PrefixListIds": [],
                    "FromPort": 22,

```

4. Create a new Linux instance of previously selected LAMP AMI from homework
1. My region is ‘US-East’. To launch and connect to the instance, use ‘run-instances’. To check the health and status, use ‘describe-instances’.

Describe the information of the AMI-008db468:

```

hqiu@bos-mpdei> aws ec2 describe-images --image-ids ami-008db468
{
  "Images": [
    {
      "VirtualizationType": "paravirtual",
      "Name": "bitnami-ocportal-9.0.19-0-linux-redhat-6.6-x86_64-ebs",
      "Hypervisor": "xen",
      "ImageId": "ami-008db468",
      "RootDeviceType": "ebs",
      "State": "available",
      "BlockDeviceMappings": [
        {
          "DeviceName": "/dev/sda1",
          "Ebs": {
            "DeleteOnTermination": true,
            "SnapshotId": "snap-5a823026",
            "VolumeSize": 10,
            "VolumeType": "standard",
            "Encrypted": false
          }
        },
        {
          "DeviceName": "/dev/sdb",
          "VirtualName": "ephemeral0"
        },
        {
          "DeviceName": "/dev/sdc",
          "VirtualName": "ephemeral1"
        },
        {
          "DeviceName": "/dev/sdd",
          "VirtualName": "ephemeral2"
        },
        {
          "DeviceName": "/dev/sde",
          "VirtualName": "ephemeral3"
        }
      ],
      "Architecture": "x86_64",
    }
  ]
}

```

Launch and connect to the instance:

```

hqiu@bos-mpdei> aws ec2 run-instances --image-id ami-008db468 --count 1 --instance-type t1.micro --key-name ec2hqiui --security-groups launch-hqiui --query 'Instances[0].InstanceId'
"i-3bf572ee"

```

Query the status: it's running and is the EBS root type.

```

hqiu@bos-mpdei> aws ec2 describe-instances --instance-ids i-3bf572ee --query 'Reservations[0].Instances[0].PublicIpAddress'
"54.209.52.7"
hqiu@bos-mpdei> aws ec2 describe-instances --instance-ids i-3bf572ee --query 'Reservations[0].Instances[0].State.Name'
"running"
hqiu@bos-mpdei> aws ec2 describe-instances --instance-ids i-3bf572ee --query 'Reservations[0].Instances[0].RootDeviceType'
"ebs"

```

```
hqiu@bos-mpdei> aws ec2 describe-instances --output table
```

DescribeInstances	
Reservations	
OwnerId	217134905396
ReservationId	r-0ef321f3
Instances	
AmiLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
Hypervisor	xen
ImageId	ami-008db468
InstanceId	i-3bf572ee
InstanceType	t1.micro
KernelId	aki-919dcaf8
KeyName	ec2hqi
LaunchTime	2015-09-15T19:52:34.000Z
PrivateDnsName	ip-172-31-1-169.ec2.internal
PrivateIpAddress	172.31.1.169
PublicDnsName	ec2-54-209-52-7.compute-1.amazonaws.com
PublicIpAddress	54.209.52.7
RootDeviceName	/dev/sda1
RootDeviceType	ebs
SourceDestCheck	True
StateTransitionReason	
SubnetId	subnet-66024a11
VirtualizationType	paravirtual
VpcId	vpc-dfb48aba
Groups	
GroupId	sg-adfff3ca
GroupName	launch-hqi
PrivateIpAddresses	
Primary	True
PrivateDnsName	ip-172-31-1-169.ec2.internal
PrivateIpAddress	172.31.1.169
Association	
IpOwnerId	system.zim.mapped
PublicDnsName	ec2-54-209-52-7.compute-1.amazonaws.com
PublicIp	54.209.52.7
Placement	
AvailabilityZone	us-east-1b
GroupName	
Tenancy	default
SecurityGroups	
GroupId	sg-adfff3ca
GroupName	launch-hqi
State	
Code	16
Name	running

Check through the AWS Console, it's running and using AMI-008db468.

The screenshot shows the AWS EC2 Instances page. At the top, there are buttons for "Launch Instance", "Connect", and "Actions". Below the header is a search bar and a navigation bar with filters: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS, and Public IP. A single instance row is displayed: i-3bf572ee, t1.micro, us-east-1b, running, 2/2 checks..., None, ec2-54-209-52-7.compute-1.amazonaws.com, 54.209.52.7.

The screenshot shows the AWS EC2 Instance Details page for instance i-3bf572ee. The instance is running. The details include:

- Description:** i-3bf572ee
- Public DNS:** ec2-54-209-52-7.compute-1.amazonaws.com
- Instance ID:** i-3bf572ee
- Instance state:** running
- Instance type:** t1.micro
- Private DNS:** ip-172-31-1-169.ec2.internal
- Private IPs:** 172.31.1.169
- Secondary private IPs:**
- VPC ID:** vpc-dfb48aba
- Subnet ID:** subnet-66024a11
- Network interfaces:** eth0
- Source/dest. check:** True
- EBS-optimized:** False
- Root device type:** ebs
- Public DNS:** ec2-54-209-52-7.compute-1.amazonaws.com
- Public IP:** 54.209.52.7
- Elastic IP:** -
- Availability zone:** us-east-1b
- Security groups:** launch-hqiu, view rules
- Scheduled events:** No scheduled events
- AMI ID:** bitnami-ocportal-9.0.19-0-linux-redhat-6.6-x86_64-ebs (ami-008db468)
- Platform:** -
- IAM role:** -
- Key pair name:** ec2hqiu
- Owner:** 217134905396
- Launch time:** September 15, 2015 at 3:52:34 PM UTC-4 (less than one hour)
- Termination protection:** False

- Try to stop, start and terminate the instance using AWS CLI commands. Check with the AWS Console.

Change from ‘running’ to ‘stopping’, then to ‘stopped’.

```
hqiu@bos-mpdei>> aws ec2 stop-instances --instance-ids i-3bf572ee
{
    "StoppingInstances": [
        {
            "InstanceId": "i-3bf572ee",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

The screenshots show the AWS EC2 Instances page at three different times:

- The first screenshot shows the instance in the "running" state.
- The second screenshot shows the instance in the "stopping" state.
- The third screenshot shows the instance in the "stopped" state.

Each screenshot includes a search bar, a navigation bar with filters, and a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS, and Public IP.

Change from ‘Stopped’ to ‘pending’, then to ‘running’.

```
hqiu@bos-mpdei>> aws ec2 start-instances --instance-ids i-3bf572ee
{
    "StartingInstances": [
        {
            "InstanceId": "i-3bf572ee",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```

Filter by tags and attributes or search by keyword								
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
		i-3bf572ee	t1.micro	us-east-1b	pending	Initializing	None	 ec2-54-209-188-112.co...

Filter by tags and attributes or search by keyword								
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
		i-3bf572ee	t1.micro	us-east-1b	running	2/2 checks ...	None	 ec2-54-209-188-112.co... 54.209.188

Change from ‘running’ to ‘shutting-down’, then to ‘terminated’.

```
hqiu@bos-mpdei>> aws ec2 terminate-instances --instance-ids i-3bf572ee
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-3bf572ee",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

Filter by tags and attributes or search by keyword								
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public IP
		i-3bf572ee	t1.micro	us-east-1b	terminated		None	

Problem 5

1. Create a Windows instance with 64-bit, root device type EBS, instance type t1.micro. First go to ‘AWS console->EC2->Launch Instance’ to choose an AMI. Choose from the ‘Community AMIs’. Do not launch the instance now, just record

the AMI ID and launch through the AWS CLI Tool. I first choose a Windows server 2008, ami-3bd17a50. Then I chose a Windows Server 2012, ami-cd9339a6 later. Both of them worked well.

AMI Name	Description	Select	Architecture
Windows_Server-2012-R2_RTM-English-64Bit-Base-2015.08.12 - ami-cd9339a6	Microsoft Windows Server 2012 R2 RTM 64-bit Locale English AMI provided by Amazon	Select	64-bit
Windows_Server-2012-R2_RTM-English-64Bit-SQL_2014_RTM_Express-2015.08.12 - ami-8359f1e8	Microsoft Windows Server 2012 R2 RTM 64-bit Locale English with SQL Express 2014 AMI provided by Amazon	Select	64-bit
Windows_Server-2012-R2_RTM-English-64Bit-SQL_2014_RTM_Web-2015.08.12 - ami-a75ef6cc	Microsoft Windows Server 2012 R2 RTM 64-bit Locale English with SQL Web 2014 AMI provided by Amazon	Select	64-bit
Windows_Server-2012-R2_RTM-English-64Bit-SQL_2014_RTM_Standard-2015.08.12 - ami-4d55fd26	Microsoft Windows Server 2012 R2 RTM 64-bit Locale English with SQL Standard 2014 AMI provided by Amazon	Select	64-bit
Windows_Server-2012-RTM-English-64Bit-Base-2015.08.12 - ami-417bcf2a	Microsoft Windows Server 2012 RTM 64-bit Locale English Base AMI provided by Amazon	Select	64-bit
Windows_Server-2012-RTM-English-64Bit-SQL_2012_SP1_Express-2015.08.12 - ami-6773c60c	Microsoft Windows Server 2012 RTM 64-bit Locale English with SQL Express 2012 AMI provided by Amazon	Select	64-bit

- Add a policy for RDP 3389, it's not default in my security group. Otherwise we can't do the remote connection via port 3389.

```
hqiu@bos-mpdei:> aws ec2 authorize-security-group-ingress --group-name default --protocol tcp --port 3389 --cidr 0.0.0.0/0
```

- Launch the instance. The Instance ID is i-200988f5.

```
hqiu@bos-mpdei:> aws ec2 run-instances --image-id ami-3bd17a50 --count 1 --instance-type t1.micro --key-name ec2hu --security-groups default
{
    "OwnerId": "217134905396",
    "ReservationId": "r-e4449919",
    "Groups": [],
    "Instances": [
        {
            "Monitoring": {
                "State": "disabled"
            },
            "PublicDnsName": "",
            "Platform": "windows",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "EbsOptimized": false,
            "LaunchTime": "2015-09-16T02:32:54.000Z",
            "PrivateIpAddress": "172.31.11.38",
            "ProductCodes": [],
            "VpcId": "vpc-dfb48aba",
            "StateTransitionReason": "",
            "InstanceId": "i-200988f5",
            "ImageId": "ami-3bd17a50",
            "PrivateDnsName": "ip-172-31-11-38.ec2.internal",
            "KeyName": "ec2hu",
            "SecurityGroups": [
                {
                    "GroupName": "default",
                    "GroupId": "sg-817073e5"
                }
            ],
            "RootDeviceType": "ebs",
            "RootDeviceName": "/dev/xvda",
            "BlockDeviceMappings": []
        }
    ]
}
```

4. Check the health and status of the instance. The ‘PublicIPAddress’ is 54.175.95.67.

```
hqiu@bos-mpdei>> aws ec2 describe-instances --instance-ids i-200988f5 --output table
```

DescribeInstances	
Reservations	
OwnerId	217134905396
ReservationId	r-e4449919
Instances	
AmiLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
Hypervisor	xen
ImageId	ami-3bd17a50
InstanceId	i-200988f5
InstanceType	t1.micro
KeyName	ec2hu
LaunchTime	2015-09-16T02:32:54.000Z
Platform	windows
PrivateDnsName	ip-172-31-11-38.ec2.internal
PrivateIpAddress	172.31.11.38
PublicDnsName	ec2-54-175-95-67.compute-1.amazonaws.com
PublicIpAddress	54.175.95.67
RootDeviceName	/dev/sda1
RootDeviceType	ebs
SourceDestCheck	True
StateTransitionReason	
SubnetId	subnet-66024a11
VirtualizationType	hvm
VpcId	vpc-dfb48aba
SecurityGroups	
GroupId	sg-817073e5
GroupName	default
State	
Code	16
Name	running

5. Request remote password using AWS CLI tools.

```
hqiu@bos-mpdei>> aws ec2 get-password-data --priv-launch-key ec2hu.pem --instance-id i-200988f5
{
    "InstanceId": "i-200988f5",
    "Timestamp": "2015-09-16T02:36:43.000Z",
    "PasswordData": "$LQG(wfwXp$"
}
```

6. Check the status in the AWS console.

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
	i-200988f5	t1.micro	us-east-1b	running	2/2 checks ...	None	ec2-54-175-95-67.compute-1.amazonaws.com...	54.175.95.6

Instance: i-200988f5 Public DNS: ec2-54-175-95-67.compute-1.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID	i-200988f5	Public DNS	ec2-54-175-95-67.compute-1.amazonaws.com
Instance state	running	Public IP	54.175.95.67
Instance type	t1.micro	Elastic IP	-
Private DNS	ip-172-31-11-38.ec2.internal	Availability zone	us-east-1b
Private IPs	172.31.11.38	Security groups	default . view rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-dfb48aba	AMI ID	Windows_Server-2008-R2_SP1-English-64Bit-Base-2015.08.12 (ami-3bd17a50)
Subnet ID	subnet-66024a11	Platform	windows
Network interfaces	eth0	IAM role	-
Source/dest. check	True	Key pair name	ec2hu
EBS-optimized	False	Owner	217134905396
Root device type	ebs	Launch time	September 15, 2015 at 10:32:54 PM UTC-4 (less than one hour)
		Termination protection	False

Security groups default . view rules

Security Groups associated with i-3834b5ed

Ports	Protocol	Source	default
3389	tcp	0.0.0.0/0	✓
All	All	sg-817073e5	✓

Key pair name ec2hu
Owner 217134905396

The password is the same with what we get using ‘aws ec2 get-password-data’.

Connect To Your Instance



You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

[Download Remote Desktop File](#)

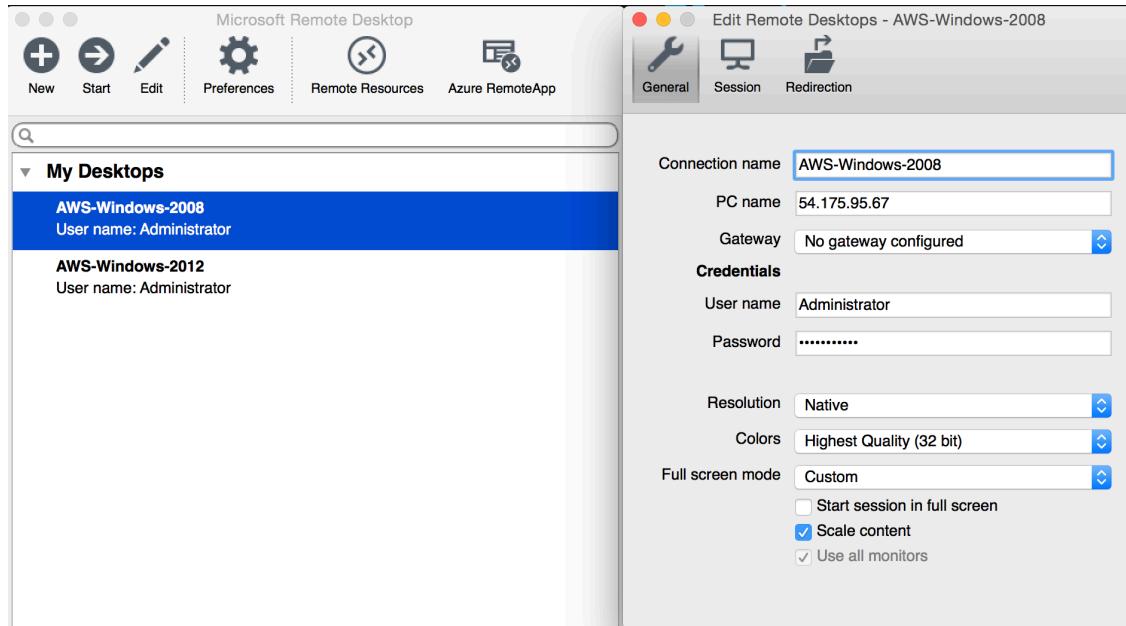
When prompted, connect to your instance using the following details:

Public IP 54.175.95.67
User name Administrator
Password \$LQG(wfwXp\$

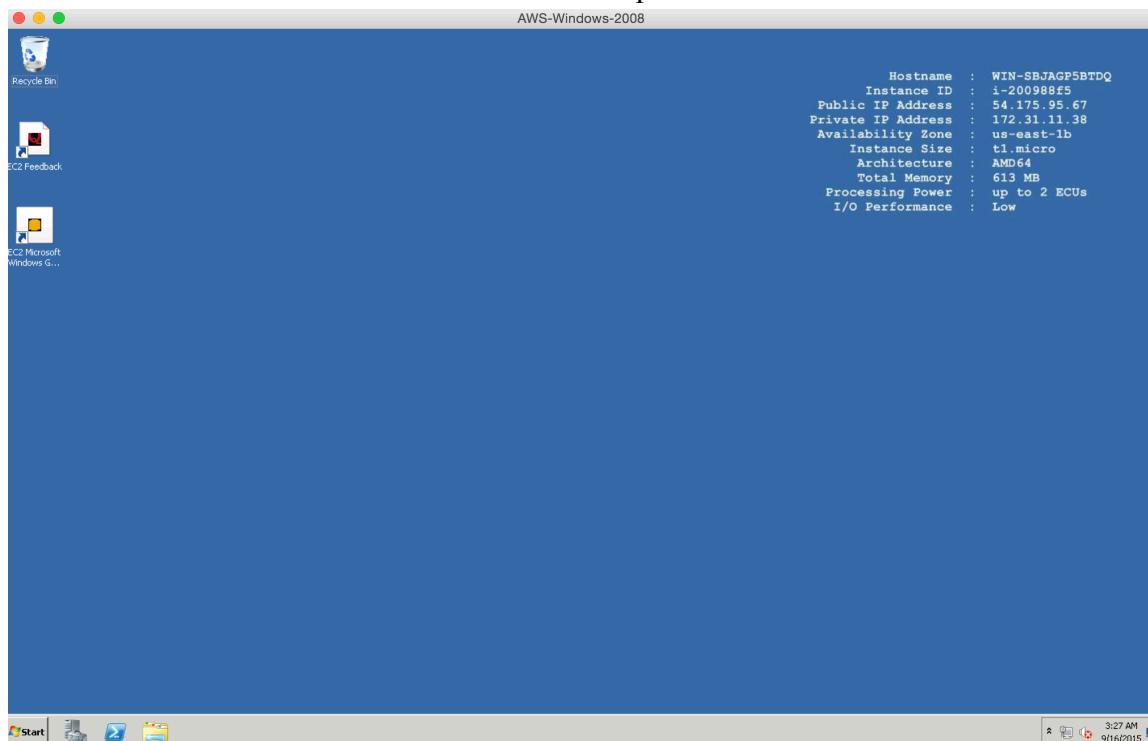
If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

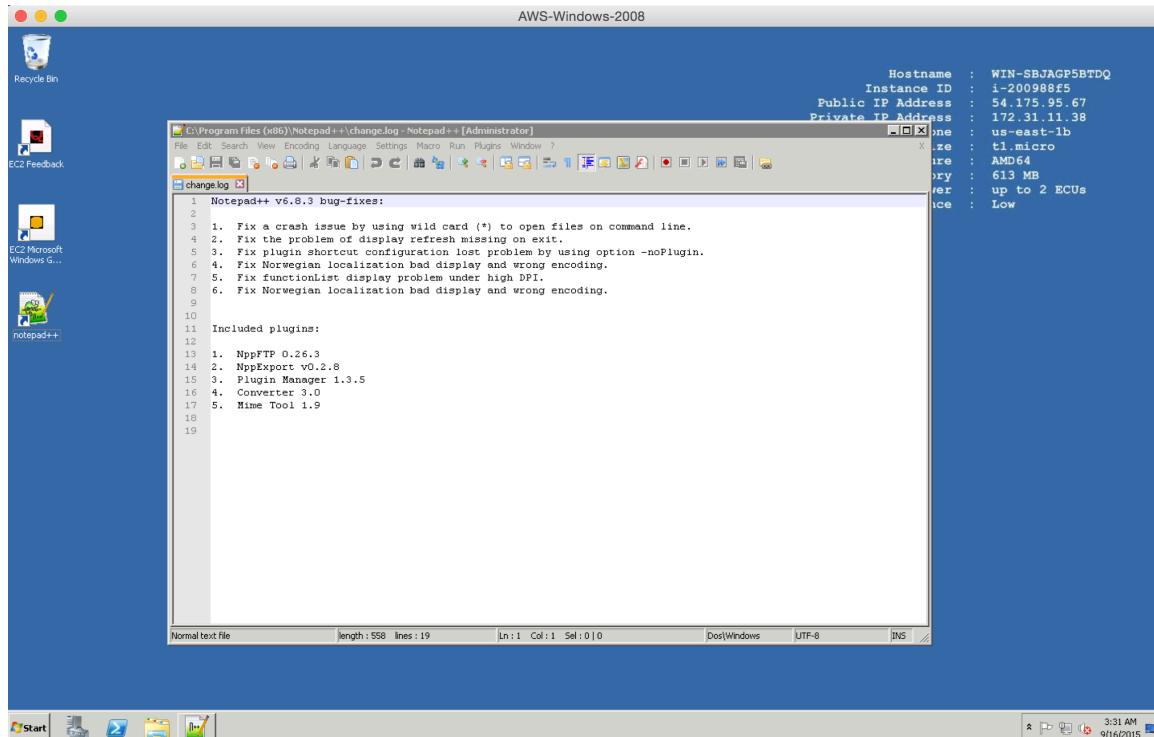
If you need any assistance connecting to your instance, please see our [connection documentation](#).

7. Use Remote Desktop Protocol and port 3389 to connect to the instance. Use the ‘Microsoft Remote Desktop’ downloaded from Apple Store. The PC name is the public IP address 54.175.95.67. Username is Administrator. Password is what we get previously. Select ‘Start’ to connect.



8. Once we get in, we can see the Public IP address and Instance ID from the Desktop. Change the password. Download text editor called ‘notepad++’ onto the instance and create a shortcut onto the desktop.





9. Select another Windows Server 2012, ami-cd9339a6 and do the same thing.

Launch the instance, get Instance ID i-3834b5ed.

```

hqiu@bos-mpdei>> aws ec2 run-instances --image-id ami-cd9339a6 --count 1 --instance-type t1.micro --key-name ec2hu --security-groups default
{
    "OwnerId": "217134905396",
    "ReservationId": "r-306cb1cd",
    "Groups": [],
    "Instances": [
        {
            "Monitoring": {
                "State": "disabled"
            },
            "PublicDnsName": "",
            "Platform": "windows",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "EbsOptimized": false,
            "LaunchTime": "2015-09-16T03:03:27.000Z",
            "PrivateIpAddress": "172.31.4.131",
            "ProductCodes": [],
            "VpcId": "vpc-dfb48aba",
            "StateTransitionReason": "",
            "InstanceId": "i-3834b5ed",
            "ImageId": "ami-cd9339a6",
            "PrivateDnsName": "ip-172-31-4-131.ec2.internal",
            "KeyName": "ec2hu",
            "SecurityGroups": [
                {
                    "GroupName": "default",
                    "GroupId": "sg-817073e5"
                }
            ],

```

Check the health and status:

```
hqiu@bos-mpdei> aws ec2 describe-instances --instance-ids i-3834b5ed --output table
```

DescribeInstances	
Reservations	
OwnerId	217134905396
ReservationId	r-306cb1cd
Instances	
AmiLaunchIndex	0
Architecture	x86_64
ClientToken	
EbsOptimized	False
Hypervisor	xen
ImageId	ami-cd9339a6
InstanceId	i-3834b5ed
InstanceType	t1.micro
KeyName	ec2hu
LaunchTime	2015-09-16T03:03:27.000Z
Platform	windows
PrivateDnsName	ip-172-31-4-131.ec2.internal
PrivateIpAddress	172.31.4.131
PublicDnsName	ec2-54-172-148-64.compute-1.amazonaws.com
PublicIpAddress	54.172.148.64
RootDeviceName	/dev/sda1
RootDeviceType	ebs
SourceDestCheck	True
StateTransitionReason	
SubnetId	subnet-66024a11
VirtualizationType	hvm
VpcId	vpc-dfb48aba
State	
Code	16
Name	running

Get remote password:

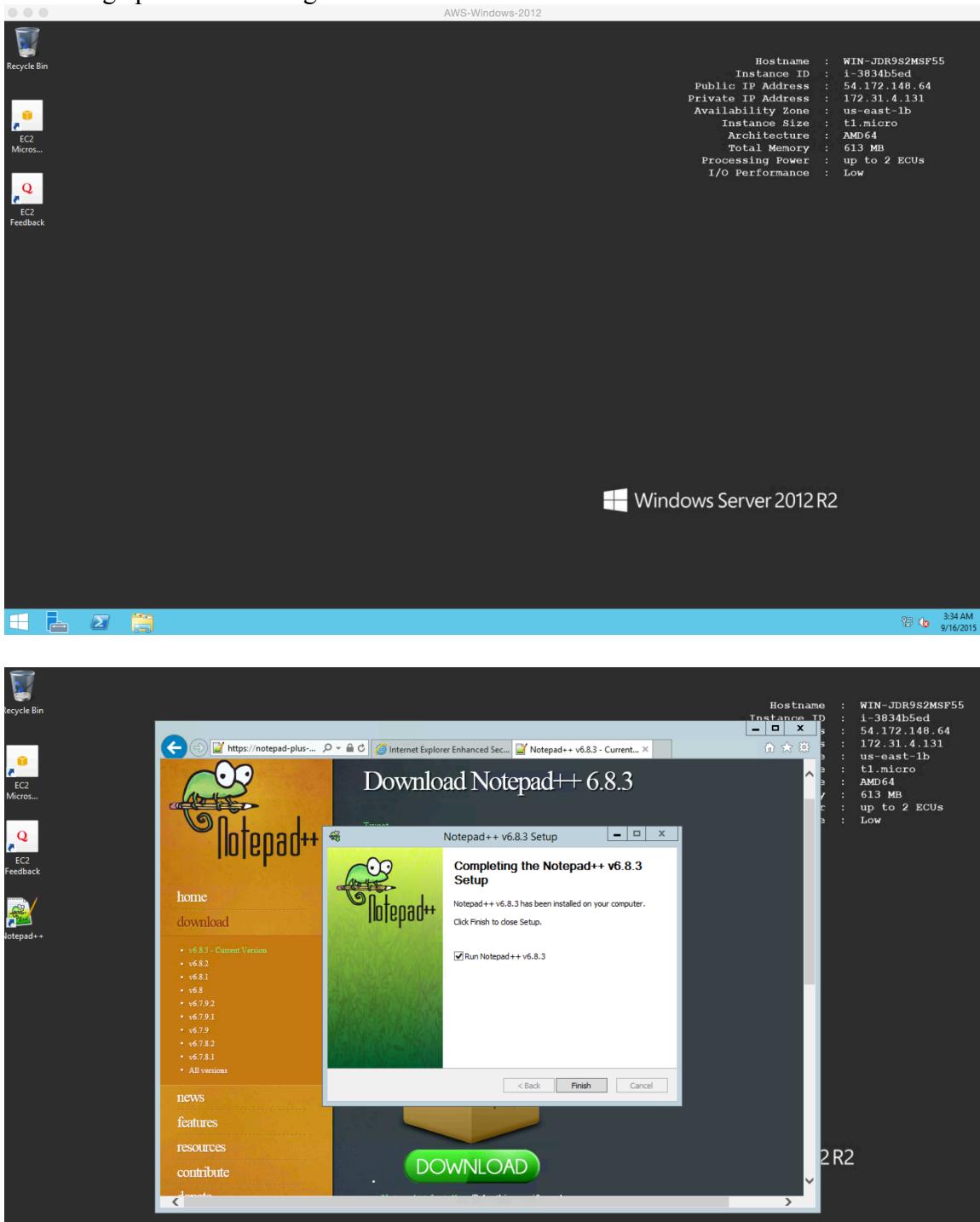
```
hqiu@bos-mpdei> aws ec2 get-password-data --priv-launch-key ec2hu.pem --instance-id i-3834b5ed
{
    "InstanceId": "i-3834b5ed",
    "Timestamp": "2015-09-16T03:06:40.000Z",
    "PasswordData": "eb!jFLFewX"
}
```

Connect through Microsoft Remote Desktop:

The screenshot shows the Microsoft Remote Desktop application interface. On the left, there's a sidebar titled 'My Desktops' with two entries: 'AWS-Windows-2008' and 'AWS-Windows-2012'. The 'AWS-Windows-2012' entry is selected and highlighted in blue. On the right, the main window is titled 'Edit Remote Desktops - AWS-Windows-2012'. It contains several configuration fields:

- Connection name: AWS-Windows-2012
- PC name: 54.172.148.64
- Gateway: No gateway configured
- Credentials:
 - User name: Administrator
 - Password: [REDACTED]
- Resolution: Native
- Colors: True Color (24 bit)
- Full screen mode: Custom
 - Start session in full screen (unchecked)
 - Scale content (checked)
 - Use all monitors (checked)

Change password once get in and download a software onto the instance.



10. Acquire an Elastic IP address. We can associate this static IP address to any instance. In this way, if an instance fails, we can migrate this Elastic IP address to another good instance. Since I'm a VPC-only account, I will be allocated with a EC2-VPC address.

```
hqiu@bos-mpdei>> aws ec2 allocate-address
{
    "PublicIp": "52.5.51.59",
    "Domain": "vpc",
    "AllocationId": "eipalloc-1f35367a"
}
```

11. Associate this Elastic IP address to instance i-200988f5. When we query the instance PublicIpAddress again, it will give the Elastic IP address 52.5.51.59.

```
hqiu@bos-mpdei>> aws ec2 associate-address --instance-id i-200988f5 --allocation-id eipalloc-1f35367a
{
    "AssociationId": "eipassoc-b61892d0"
}
hqiu@bos-mpdei>> aws ec2 describe-instances --instance-ids i-200988f5 --query 'Reservations[0].Instances[0].PublicIpAddress'
"52.5.51.59"
```

12. Check the AWS console, we can see the public IP for the first instance changes.

	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
<input checked="" type="checkbox"/>	i-200988f5	i-200988f5	t1.micro	us-east-1b	green running	green 2/2 checks ...	none	ec2-52-5-51-59.compute...	52.5.51.59
<input type="checkbox"/>	i-3834b5ed	i-3834b5ed	t1.micro	us-east-1b	green running	green 2/2 checks ...	none	ec2-54-172-148-64.co...	54.172.148.

Instance: i-200988f5 Elastic IP: 52.5.51.59	
Description	Status Checks
Monitoring	Tags
Instance ID	i-200988f5
Instance state	running
Instance type	t1.micro
Private DNS	ip-172-31-11-38.ec2.internal
Private IPs	172.31.11.38
Secondary private IPs	
VPC ID	vpc-dfb48aba
Subnet ID	subnet-66024a11
Network interfaces	eth0
Source/dest. check	True
Public DNS	ec2-52-5-51-59.compute-1.amazonaws.com
Public IP	52.5.51.59
Elastic IP	52.5.51.59
Availability zone	us-east-1b
Security groups	default, view rules
Scheduled events	No scheduled events
AMI ID	Windows_Server-2008-R2_SP1-English-64Bit-Base-2015.08.12 (ami-3bd17a50)
Platform	windows
IAM role	-
Key pair name	ec2hu

13. Disassociate the Elastic IP address to the instance. Check the public IP address again, it changes back to a normal dynamic allocated public IP address.

```
hqiu@bos-mpdei>> aws ec2 disassociate-address --association-id eipassoc-b61892d0
hqiu@bos-mpdei>> aws ec2 describe-instances --instance-ids i-200988f5 --query 'Reservations[0].Instances[0].PublicIpAddress'
"54.174.9.182"
```

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
<input checked="" type="checkbox"/>	i-200988f5	t1.micro	us-east-1b	running	2/2 checks ...	None	ec2-54-174-9-182.com...	54.174.9.18
<input type="checkbox"/>	i-3834b5ed	t1.micro	us-east-1b	running	2/2 checks ...	None	ec2-54-172-148-64.co...	54.172.148.

Instance: i-200988f5 Public DNS: ec2-54-174-9-182.compute-1.amazonaws.com

Description	Status Checks	Monitoring	Tags
Instance ID: i-200988f5 Instance state: running Instance type: t1.micro Private DNS: ip-172-31-11-38.ec2.internal Private IPs: 172.31.11.38 Secondary private IPs: VPC ID: vpc-dfb48aba Subnet ID: subnet-66024a11 Network interfaces: eth0 Source/dest. check: True	Public DNS: ec2-54-174-9-182.compute-1.amazonaws.com Public IP: 54.174.9.182 Elastic IP: - Availability zone: us-east-1b Security groups: default, view rules Scheduled events: No scheduled events AMI ID: Windows_Server-2008-R2_SP1-English-64Bit-Base-2015.08.12 (ami-3bd17a50) Platform: windows IAM role: - Key pair name: ec2hu Owner: 217134905396		

14. Release the acquired Elastic IP address.

```
hqiu@bos-mpdei>> aws ec2 release-address --allocation-id eipalloc-1f35367a
```

Allocate New Address Actions ▾

Filter by attributes or search by keyword

You do not have any elastic IPs in this region.

Click on the "Allocate New Address" button to allocate your first elastic IP.

Allocate New Address

15. Stop and terminate the two Windows instances.

```
hqiu@bos-mpdei>> aws ec2 stop-instances --instance-ids i-200988f5
{
  "StoppingInstances": [
    {
      "InstanceId": "i-200988f5",
      "CurrentState": {
        "Code": 64,
        "Name": "stopping"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
hqiu@bos-mpdei>> aws ec2 stop-instances --instance-ids i-3834b5ed
{
  "StoppingInstances": [
    {
      "InstanceId": "i-3834b5ed",
      "CurrentState": {
        "Code": 64,
        "Name": "stopping"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

Filter by tags and attributes or search by keyword								
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-200988f5	t1.micro	us-east-1b		stopping	None		ec2-54-174-9-182.com... 54.174.9.18
	i-3834b5ed	t1.micro	us-east-1b		stopping	None		ec2-54-172-148-64.co... 54.172.148

Filter by tags and attributes or search by keyword								
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-200988f5	t1.micro	us-east-1b		stopped	None		
	i-3834b5ed	t1.micro	us-east-1b		stopped	None		

```
hqiu@bos-mpdei> aws ec2 terminate-instances --instance-ids i-3834b5ed
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-3834b5ed",
            "CurrentState": {
                "Code": 48,
                "Name": "terminated"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```

```
hqiu@bos-mpdei> aws ec2 terminate-instances --instance-ids i-200988f5
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-200988f5",
            "CurrentState": {
                "Code": 48,
                "Name": "terminated"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```

Filter by tags and attributes or search by keyword								
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-200988f5	t1.micro	us-east-1b		shutting-down	None		
	i-3834b5ed	t1.micro	us-east-1b		terminated	None		

Filter by tags and attributes or search by keyword								
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-200988f5	t1.micro	us-east-1b		terminated	None		
	i-3834b5ed	t1.micro	us-east-1b		terminated	None		

Problem 6

- Check the cost and usage from ‘Billing & Cost Management’. Since I use the free tier, it doesn’t cost me anything. But I use 2 hours EC2-Linux, 3 hours EC2-Windows and some S3 Puts and Gets for this problem set.

Month-to-Date Top Services by Spend	Amount
EC2	\$0.00
DataTransfer	\$0.00
S3	\$0.00
SNS	\$0.00
Other Services	\$0.00
Tax	\$0.00
Total	\$0.00

All Free Tier services by usage



Service	Month-to-date actual usage	Month-end forecasted usage	Free Tier usage limit	Status
S3 - Puts	154.00 Requests	7.70%	288.75 Requests	14.44%
S3 - Gets	504.00 Requests	2.52%	945.00 Requests	4.73%
EBS - I/Os	27,899.00 IOs	1.39%	52,310.63 IOs	2.62%
EBS - Volumes	0.17 GB-Mo	0.56%	0.31 GB-Mo	1.04%
EC2 - Linux	4.00 Hrs	0.53%	7.50 Hrs	1.00%
EC2 - Windows	3.00 Hrs	0.40%	5.63 Hrs	0.75%
KMS - Requests	4.00 Requests	0.02%	7.50 Requests	0.04%
S3 - Storage	0.00 GB-Mo	0.02%	0.00 GB-Mo	0.04%