aws CERTIFIED Cloud Practitioner

aws CERTIFIED Developer Associate

aws CERTIFIED SysOps Administrator Associate

aws CERTIFIED Solutions Architect Associate

lab

lab title

Connecting to EC2 Instances V1.00

Course title

BackSpace Academy
AWS Certified Associate

BackSpace

Table of Contents

Contents

About the Lab

These lab notes are to support the instructional videos with AWS in the BackSpace AWS Certified Associate preparation course.

Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the lastest version with any updates or corrections. The videos may not be as current as these lab notes so please follow these lab notes carefully.

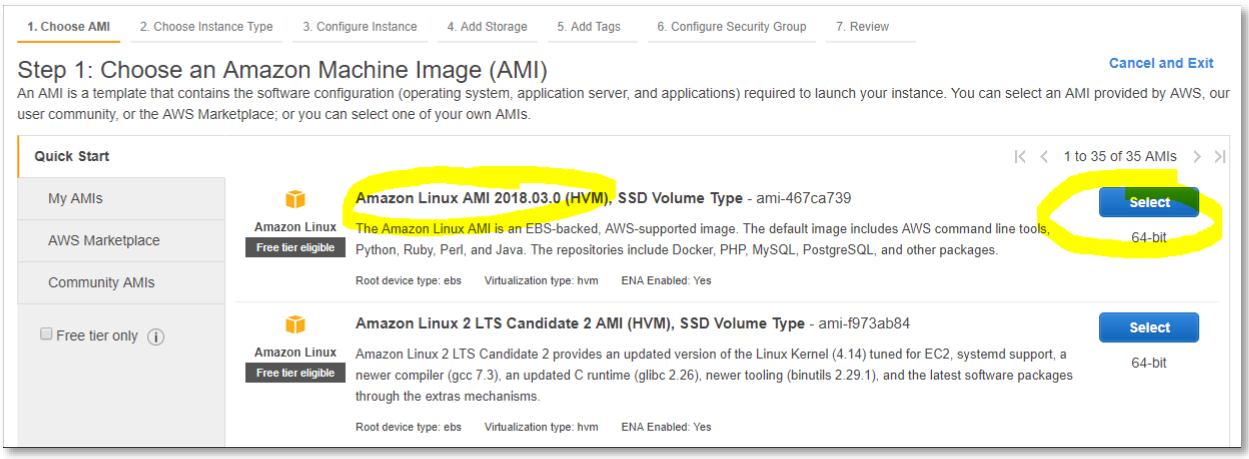
Creating an Amazon EC2 Instance and Key Pair

In this section we will use the EC2 service to create and an EC2 instance and a key pair for connecting to the operating system of the instance.

Select the EC2 Console.

Click “Launch Instance”

Select the Amazon Linux AMI



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-467ca739

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

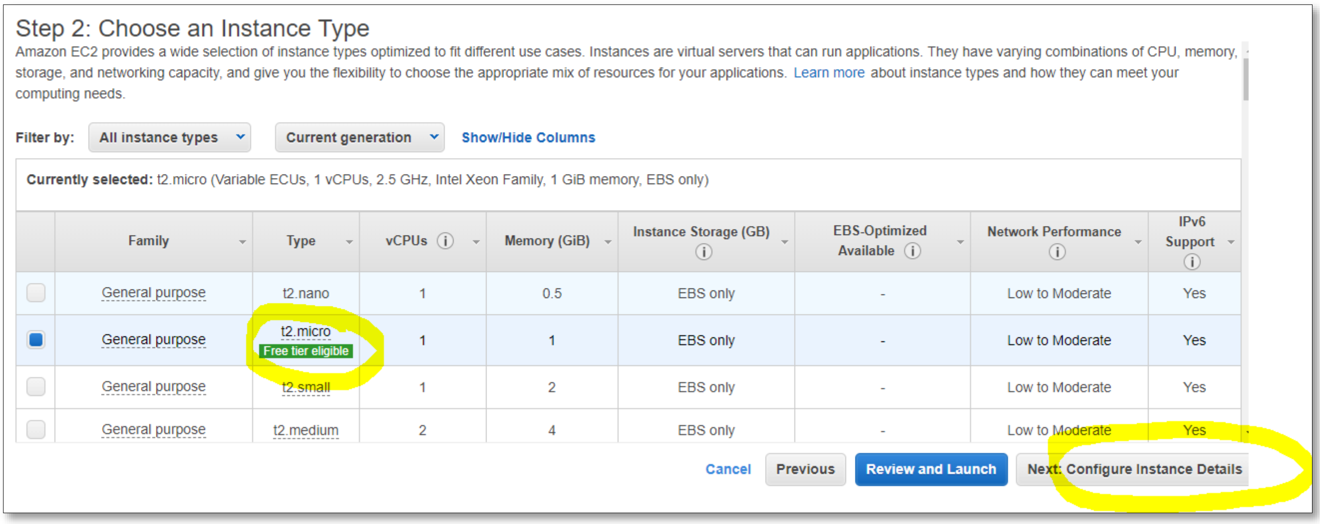
Amazon Linux 2 LTS Candidate 2 AMI (HVM), SSD Volume Type - ami-f973ab84

Amazon Linux 2 LTS Candidate 2 provides an updated version of the Linux Kernel (4.14) tuned for EC2, systemd support, a newer compiler (gcc 7.3), an updated C runtime (glibc 2.26), newer tooling (binutils 2.29.1), and the latest software packages through the extras mechanisms.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select t2 micro instance

Click “Next: Configure Instance Details”



Step 2: Choose an Instance Type

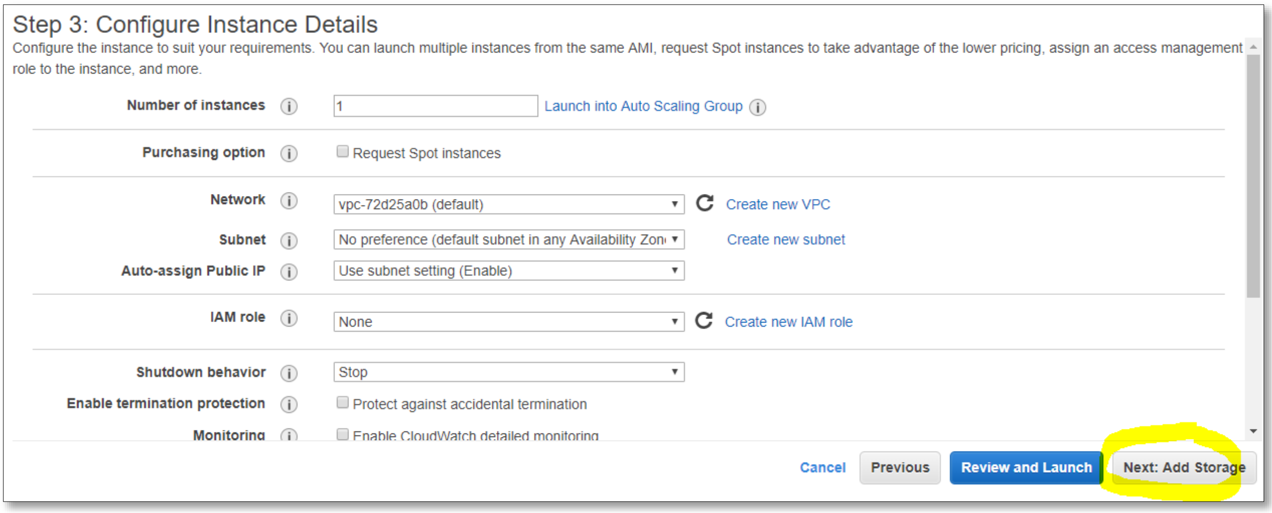
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources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

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes

Leave default settings

Click “Next: Add Storage”

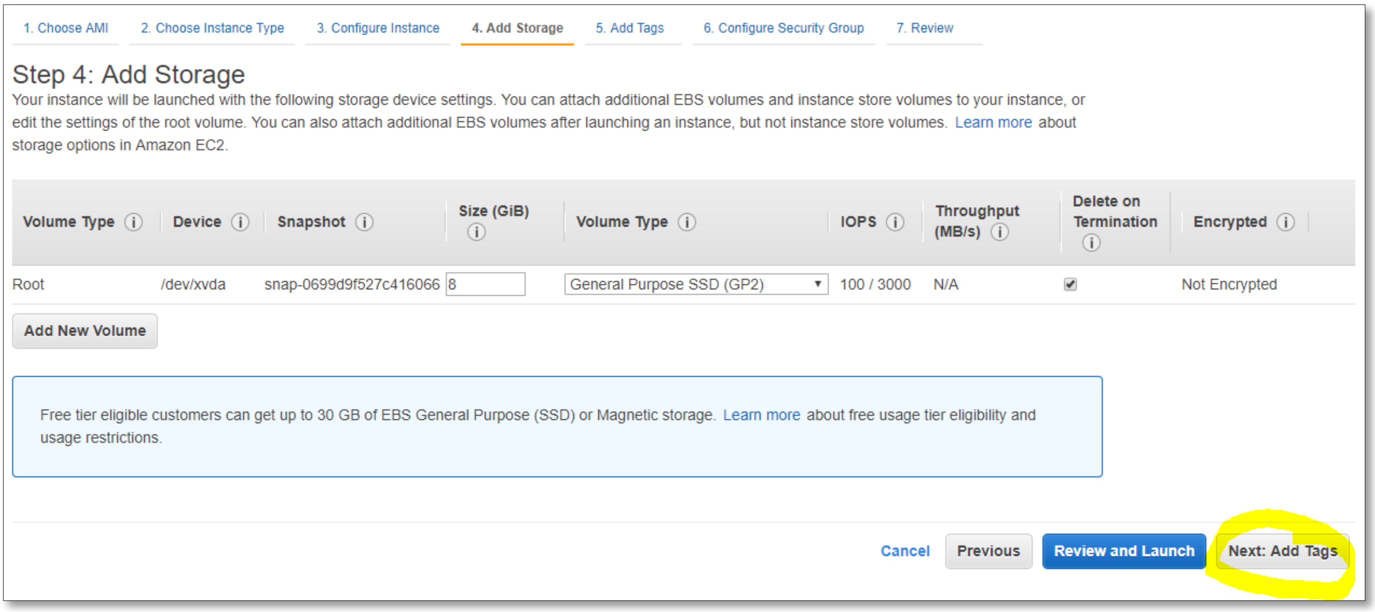


Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1
Purchasing option	□ Request Spot instances
Network	vpc-72d25a0b (default)
Subnet	No preference (default subnet in any Availability Zon
Auto-assign Public IP	Use subnet setting (Enable)
IAM role	None
Shutdown behavior	Stop
Enable termination protection	□ Protect against accidental termination
Monitoring	□ Enable CloudWatch detailed monitoring

Click “Next: Add Tags”



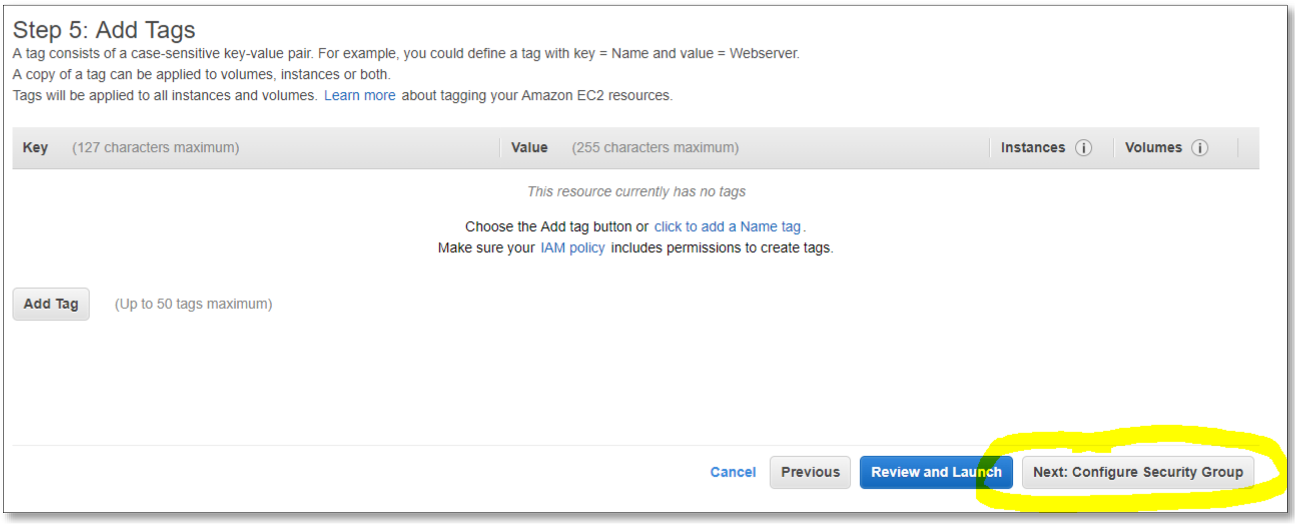
Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0699d9f527c416066	8	General Purpose SSD (GP2)	100 / 3000	N/A	☑	Not Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Click “Next: Configure Security Group”



Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

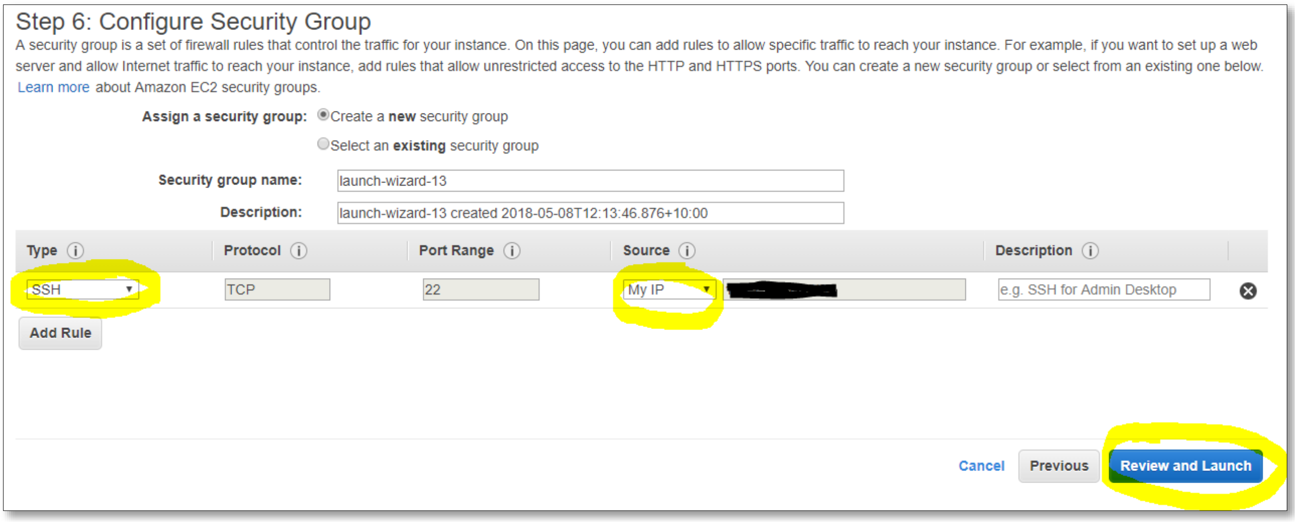
Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes

This resource currently has no tags

Choose the Add tag button or click to add a Name tag.
Make sure your IAM policy includes permissions to create tags.

Select “My IP” to allow SSH access only from your IP address (note if you are using a dynamic IP address leave this as “Anywhere”)

Click “Review and Launch”



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.

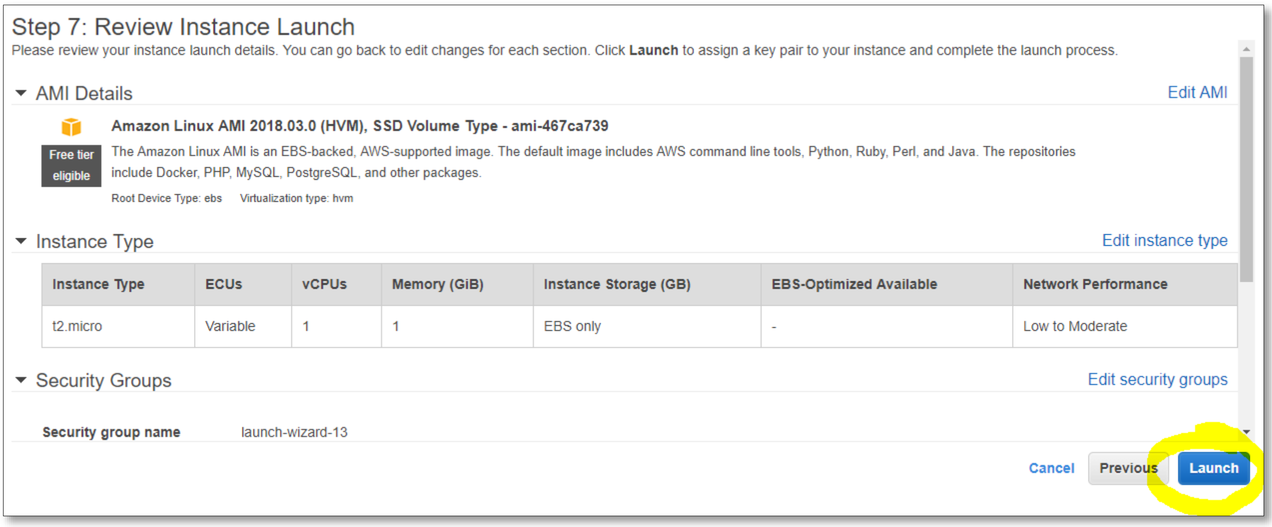
Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: launch-wizard-13

Description: launch-wizard-13 created 2018-05-08T12:13:46.876+10:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	My IP [redacted]	e.g. SSH for Admin Desktop

Click “Launch”



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details — Edit AMI

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-467ca739

Free tier eligible

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root Device Type: ebs Virtualization type: hvm

Instance Type — Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups — Edit security groups

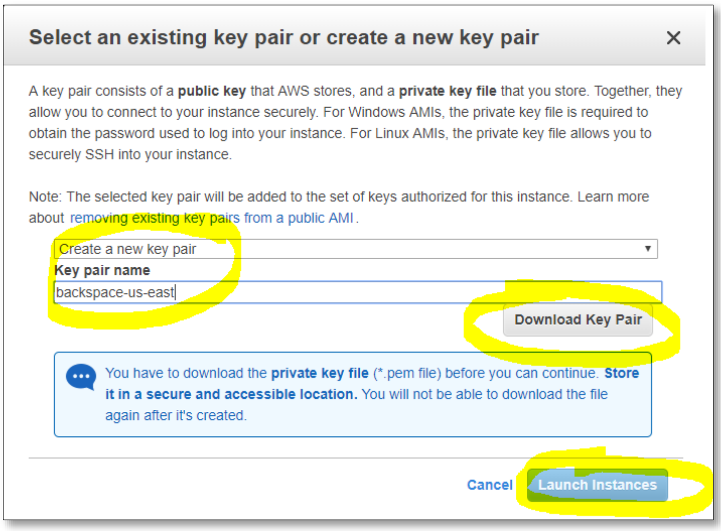
Security group name launch-wizard-13

Select “Create a new key pair”

Give the key pair a name.

Download the key pair (pem file) somewhere safe.

Click “Launch Instance”



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair

Key pair name

backspace-us-east

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

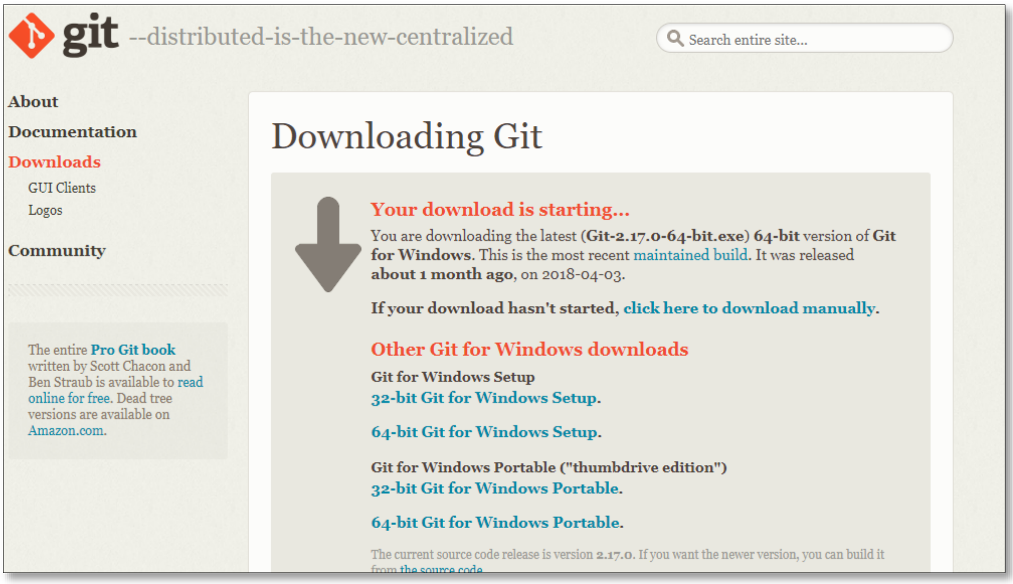
Installing a Cygwin Client for Windows

This section is only required for Windows operating systems. Linux/Unix systems such as Mac will work out of the box with SSH. In this section we will install Git for Windows. This will install and integrate with the Cygwin environment.

Please note there are a number of options available for Unix emulation client on Windows but Git for Windows provides the most reliable integrated solution.

Go to https://git-scm.com/download/win

Download and install Git for Windows



The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Downloading Git

Your download is starting...

You are downloading the latest (Git-2.17.0-64-bit.exe) 64-bit version of Git for Windows. This is the most recent maintained build. It was released about 1 month ago, on 2018-04-03.

If your download hasn't started, click here to download manually.

Other Git for Windows downloads

Git for Windows Setup
32-bit Git for Windows Setup.

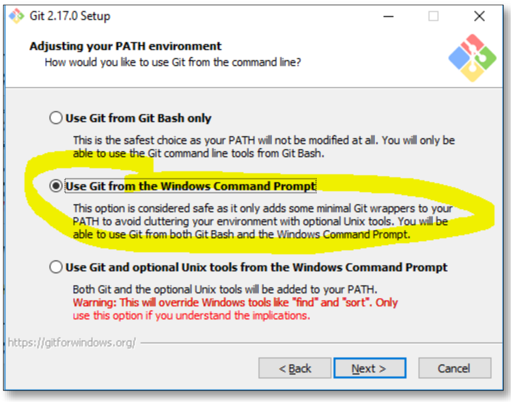
64-bit Git for Windows Setup.

Git for Windows Portable ("thumbdrive edition")
32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

The current source code release is version 2.17.0. If you want the newer version, you can build it from the source code.

Select “Use GIT from the Windows Command Prompt”, click Next



Adjusting your PATH environment

How would you like to use Git from the command line?

○ **Use Git from Git Bash only**

This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

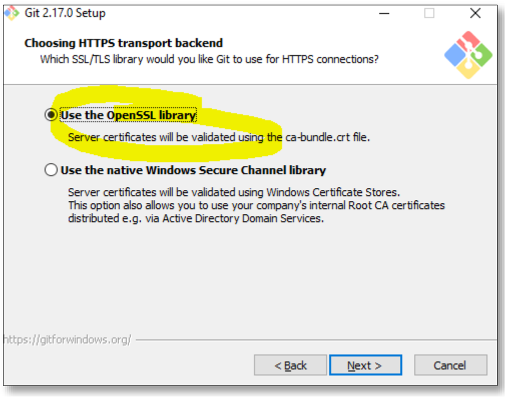
● **Use Git from the Windows Command Prompt**

This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt.

○ **Use Git and optional Unix tools from the Windows Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

Select OpenSSH, click Next



Choosing HTTPS transport backend

Which SSL/TLS library would you like Git to use for HTTPS connections?

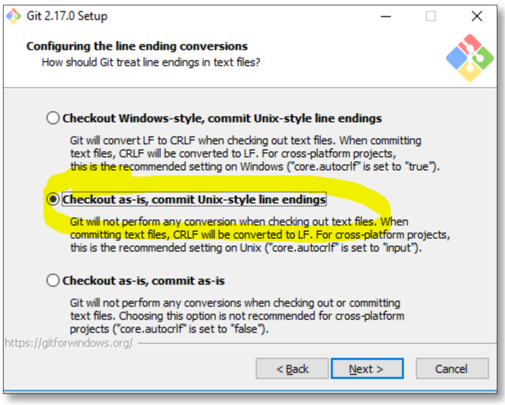
● **Use the OpenSSL library**

Server certificates will be validated using the ca-bundle.crt file.

○ **Use the native Windows Secure Channel library**

Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

Select Checkout as-is, commit Unix-style line changes style, click Next



Configuring the line ending conversions

How should Git treat line endings in text files?

○ **Checkout Windows-style, commit Unix-style line endings**

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

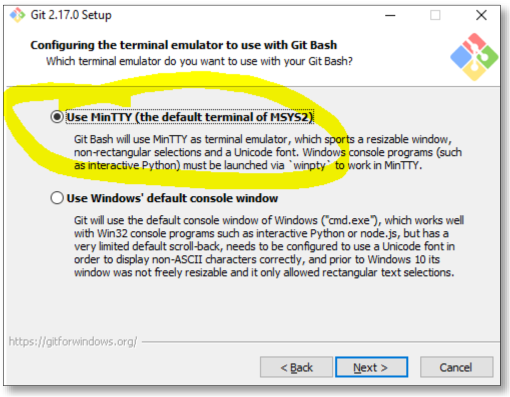
● **Checkout as-is, commit Unix-style line endings**

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

○ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

Select Use MinTTY, click Next



Configuring the terminal emulator to use with Git Bash

Which terminal emulator do you want to use with your Git Bash?

● **Use MinTTY (the default terminal of MSYS2)**

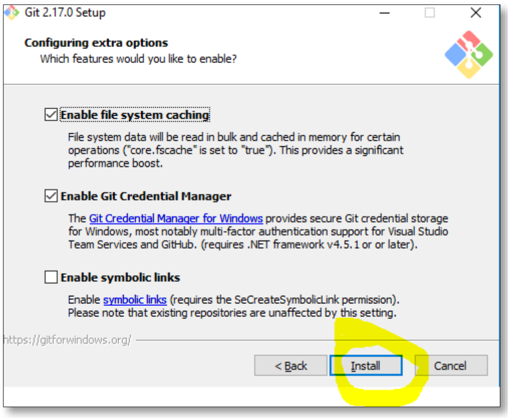
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.

○ **Use Windows' default console window**

Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

https://gitforwindows.org/

Click “Install”



Configuring extra options

Which features would you like to enable?

☑ **Enable file system caching**

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

☑ **Enable Git Credential Manager**

The Git Credential Manager for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or or later).

☐ **Enable symbolic links**

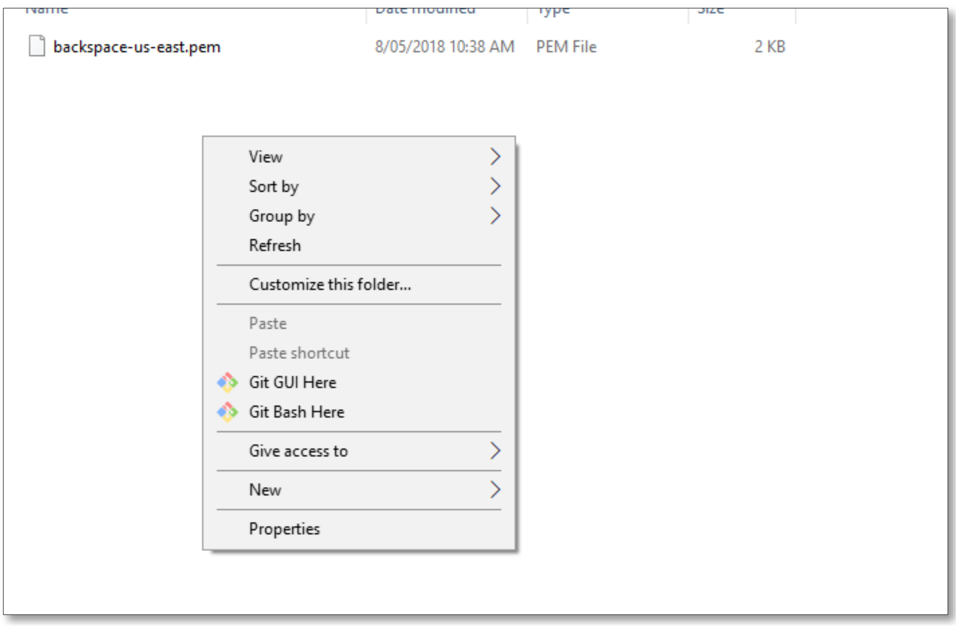
Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

https://gitforwindows.org/

Open File Explorer

Right click inside a folder

If the installation was successful, then you should see an option “Git Bash Here”.



Name	Date modified	Type	Size
backspace-us-east.pem	8/05/2018 10:38 AM	PEM File	2 KB

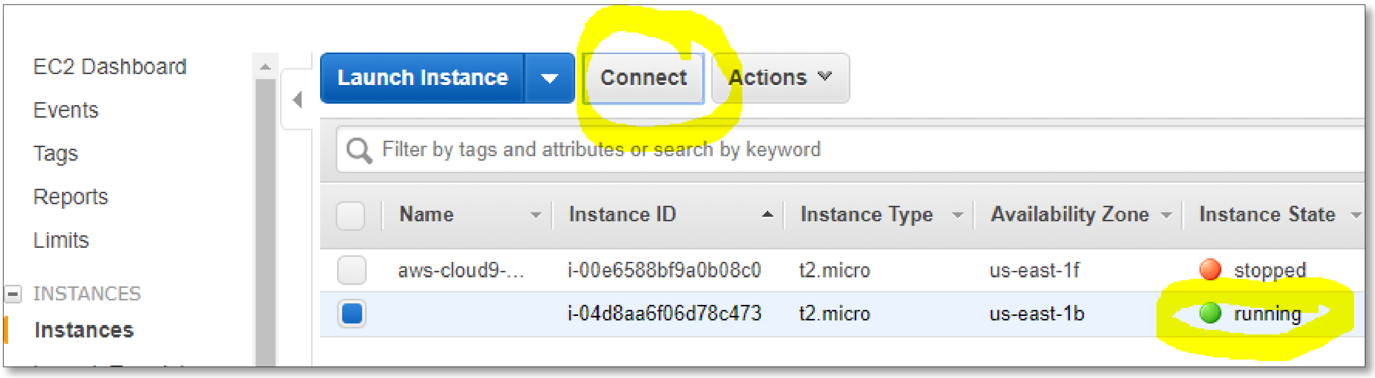
Connecting to an EC2 Instance using an SSH Client

In this section we will use Git Bash to connect in to our EC2 instance. Mac and Linux users can use the terminal to connect.

Go to the EC2 console

Select the instance you created previously

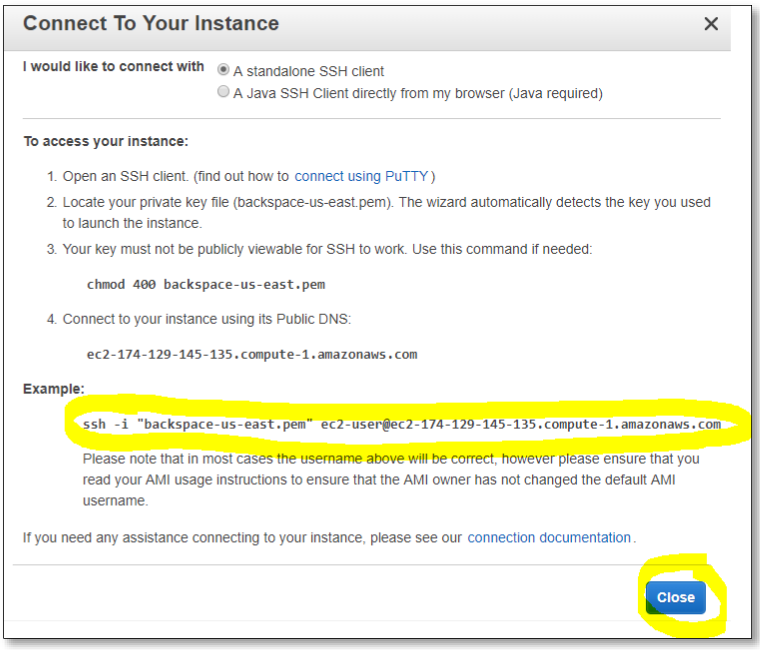
Click “Connect”



	Name	Instance ID	Instance Type	Availability Zone	Instance State
□	aws-cloud9-...	i-00e6588bf9a0b08c0	t2.micro	us-east-1f	● stopped
■		i-04d8aa6f06d78c473	t2.micro	us-east-1b	● running

Copy the connection command

Click “Close”



Connect To Your Instance

I would like to connect with ● A standalone SSH client ○ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (backspace-us-east.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 backspace-us-east.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-174-129-145-135.compute-1.amazonaws.com
```

Example:

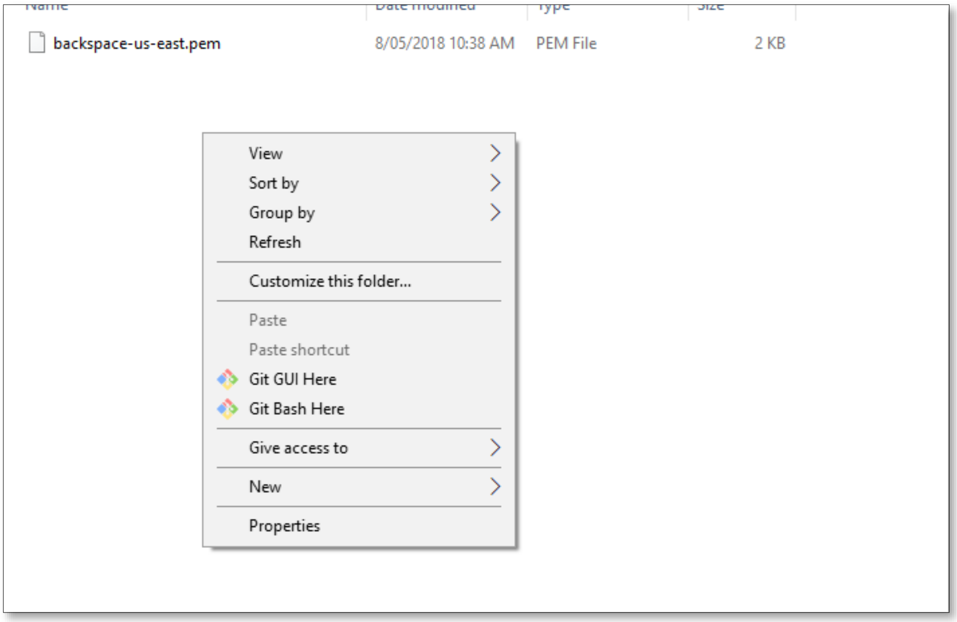
```
ssh -i "backspace-us-east.pem" ec2-user@ec2-174-129-145-135.compute-1.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation.

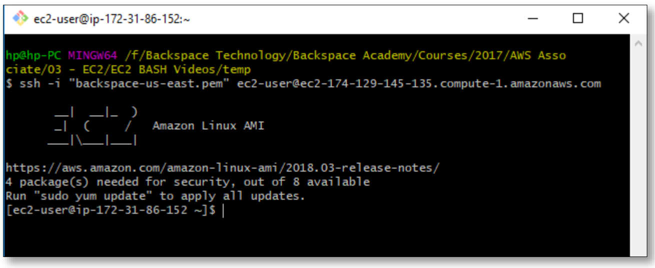
Navigate to the location of your key pair (pem) file using File Explorer.

Right click and select “Git Bash here”



backspace-us-east.pem 8/05/2018 10:38 AM PEM File 2 KB

Paste in the connection command and press enter



ec2-user@ip-172-31-86-152:~

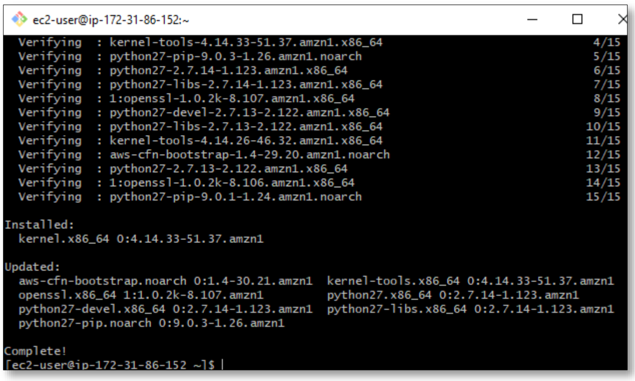
```
hp@hp-PC MINGW64 /f/Backspace Technology/Backspace Academy/Courses/2017/AWS Asso
ciate/03 - EC2/EC2 BASH Videos/temp
$ ssh -i "backspace-us-east.pem" ec2-user@ec2-174-129-145-135.compute-1.amazonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
4 package(s) needed for security, out of 8 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-86-152 ~]$ |
```

You will now be connected to the Linux operating system of your EC2 instance.

Try a sudo yum update command



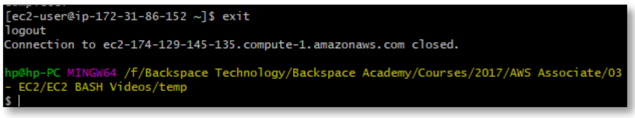
```
ec2-user@ip-172-31-86-152:~
  Verifying  : kernel-tools-4.14.33-51.37.amzn1.x86_64                          4/15
  Verifying  : python27-pip-9.0.3-1.26.amzn1.noarch                              5/15
  Verifying  : python27-2.7.14-1.123.amzn1.x86_64                                6/15
  Verifying  : python27-libs-2.7.14-1.123.amzn1.x86_64                           7/15
  Verifying  : 1:openssl-1.0.2k-8.107.amzn1.x86_64                               8/15
  Verifying  : python27-devel-2.7.13-2.122.amzn1.x86_64                          9/15
  Verifying  : python27-libs-2.7.13-2.122.amzn1.x86_64                          10/15
  Verifying  : kernel-tools-4.14.26-46.32.amzn1.x86_64                          11/15
  Verifying  : aws-cfn-bootstrap-1.4-29.20.amzn1.noarch                         12/15
  Verifying  : python27-2.7.13-2.122.amzn1.x86_64                               13/15
  Verifying  : 1:openssl-1.0.2k-8.106.amzn1.x86_64                              14/15
  Verifying  : python27-pip-9.0.1-1.24.amzn1.noarch                             15/15

Installed:
  kernel.x86_64 0:4.14.33-51.37.amzn1

Updated:
  aws-cfn-bootstrap.noarch 0:1.4-30.21.amzn1  kernel-tools.x86_64 0:4.14.33-51.37.amzn1
  openssl.x86_64 1:1.0.2k-8.107.amzn1         python27.x86_64 0:2.7.14-1.123.amzn1
  python27-devel.x86_64 0:2.7.14-1.123.amzn1  python27-libs.x86_64 0:2.7.14-1.123.amzn1
  python27-pip.noarch 0:9.0.3-1.26.amzn1

Complete!
[ec2-user@ip-172-31-86-152 ~]$ |
```

To exit the connection type exit



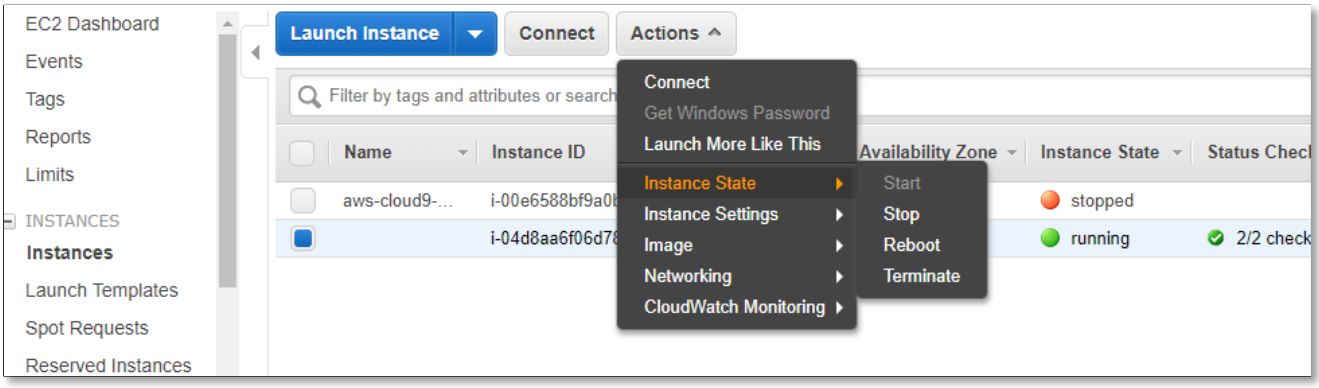
```
[ec2-user@ip-172-31-86-152 ~]$ exit
logout
Connection to ec2-174-129-145-135.compute-1.amazonaws.com closed.

hp@hp-PC MINGW64 /f/Backspace Technology/Backspace Academy/Courses/2017/AWS Associate/03
- EC2/EC2 BASH Videos/temp
$ |
```

Clean Up:

To ensure you don't get billed for the instance make sure you go back to the EC2 console and terminate the instance.

You can leave the key pair for use later. Just make sure you save it somewhere safe.


EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Launch Instance
Connect
Actions
Filter by tags and attributes or search
Name
Instance ID
Availability Zone
Instance State
Status Checl
aws-cloud9-...
i-00e6588bf9a0l
stopped
i-04d8aa6f06d78
running
2/2 check
Connect
Get Windows Password
Launch More Like This
Instance State
Instance Settings
Image
Networking
CloudWatch Monitoring
Start
Stop
Reboot
Terminate