NYL AWS

Cloud Community Builders



Author

TAble of  
Contents

About Cloud Community 1

Community Driven Documentation 1

Prerequisites 2

Creating AWS Account 2

Logging into AWS Console 2

AWS courses 3

TIPS 3

Creating Keypairs 4

Generate AWS keys 4

Create PUTTY keys, .ppk 4

Create Public RSA key 5

Create cert.pem from key.pem 5

Creating GitHub account 6

Video guide to setup GitHub Account 6

GitHub login 6

TIPS 6

Creating Terraform account 7

Workspace 8

Workspace Options 8

Create and configure an AWS EC2 instance 8

Desktop Install 8

TurnKey Training Server 9

Turnkey Training Server 9

Desktop Installation 10

Installing AWS CLI 10

Installing GIT 11

Creating EC2 an Image 12

Video 12

Image Builder Process 12

Custom AMI – Recommended 14

Standard AMI 15

Recommend Components 15

Fixing Errors 18

Creating EC2 instance 19

Connecting to EC2 instance 24

AWS console – Browser terminal 25

Initializing Instance 26

Identify connection information 26

Putty - Terminal 27

Graphical Desktop - Mate 28

Installing Visual Studio Code 30

Configure Server 31

Configure AWS CLI 31

Configure CLI 31

Common AWS CLI commands 33

Configure GIT 34

Create repository 34

Install GIT 34

Configure GIT 35

Implement Terraform 36

Internet guides 36

Terraform Installation 36

Common Terraform Commands 38

Training information 39

NYLife Training Curriculum 39

more information 40

GIT Notes 41

AWS 43

Terms Cheat Sheets 43

AWS Mobile app 46

# About Cloud Community

## Community Driven Documentation

Please share the link below for to others, it is restricted to read only access. This will prevent accidental corruption of this document.  
<https://newyorklife-my.sharepoint.com/:w:/p/mjmurphy01/EbIAkpchVMZNjTfdsEUTOeEBE3ClfYgXreaVoI4v33oPhg?e=cL1DD6>

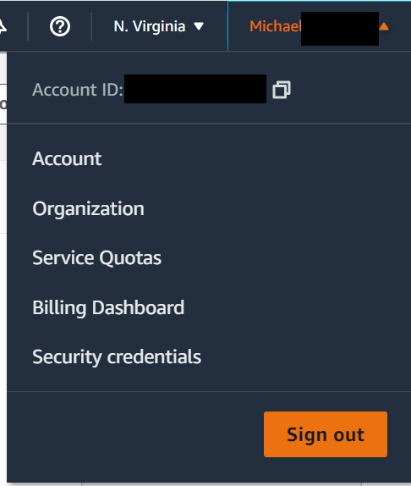
If you wish to edit this document, please use GITUB and access the version at repository: <https://github.com/CloudCommunityBuilders/Documentation>

# Prerequisites

## Creating AWS Account

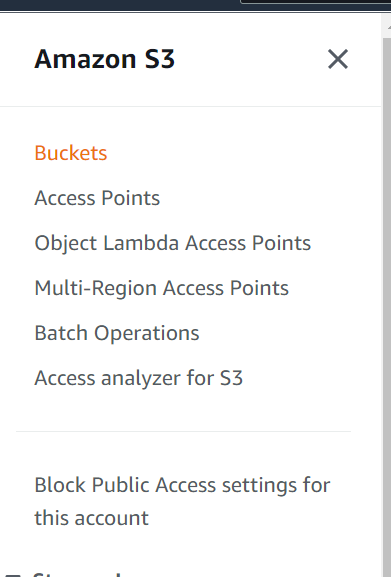
It is strictly forbidden to place NYL assets to your personal AWS account.   
The personal AWS account is only to be used to support personal training courses.  
*You must use a personal email address for the account.*  
Video guide to create an account

<https://youtu.be/bgPuPSPZe2U?t=180>

Logging into AWS Console  
When using a NYL computer for AWS training, you may run into issues logging into your personal AWS account. The NYL devices can get confused between your personal account and a NYL account. Normal console login is here:  
<https://aws.amazon.com/console/>  
  
If you have issues logging into your personal AWS account, try this link:  
[https://signin.aws.amazon.com/signin?client\_id=arn%3Aaws%3Asignin%3A%3A%3Aconsole%2Fcanvas&redirect\_uri=https%3A%2F%2Fus-east-1.console.aws.amazon.com%2Fconsole%2Fhome%3Ffromtb%3Dtrue%26hashArgs%3D%2523%26isauthcode%3Dtrue%26region%3Dus-east-1%26skipRegion%3Dtrue%26state%3DhashArgsFromTB\_us-east-1\_38ce933c8e82e221&page=resolve](https://signin.aws.amazon.com/signin?client_id=arn%3Aaws%3Asignin%3A%3A%3Aconsole%2Fcanvas&redirect_uri=https%3A%2F%2Fus-east-1.console.aws.amazon.com%2Fconsole%2Fhome%3Ffromtb%3Dtrue%26hashArgs%3D%2523%26isauthcode%3Dtrue%26region%3Dus-east-1%26skipRegion%3Dtrue%26state%3DhashArgsFromTB_us-east-1_38ce933c8e82e221&page=resolve)

AWS coursesTo learn more, AWS has courses, as well as NYL <https://explore.skillbuilder.aws/learn/catalog>

### TIPS

1. Recommend using us-east-1, it has all AWS services
2. Under account make sure your information is up to date and accurate. Amazon will ONLY use these credentials to assist you.
3. Recommend enabling MFA to secure your account.  
   One option is to install Google Authenticator on your device and scan the QR code.
4. Recommend clicking under services, then S3. In menu select “block public access settings for this account”. Enable blocking public storage access of your data, this will prevent exposing S3 stored files to the public. [Click here for help](https://explore.skillbuilder.aws/learn/course/146/play/362/amazon-simple-storage-service-amazon-s3-block-public-access).

## Graphical user interface, text, application Description automatically generatedCreating Keypairs

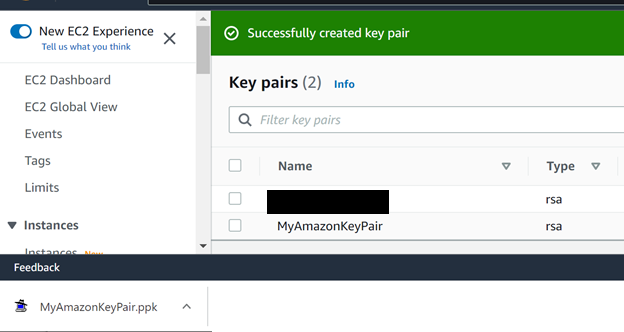
### Generate AWS keys

To connect to your server, you will need to create a keypair. This creates a file you can import into PUTTY or other tools to remotely connect to the server via command line.

Youtube video on the process:

<https://www.youtube.com/watch?v=TkSuLg-8TD8>

In the search bar type “key Management Service”, and click “create a key”.  
NOTE: There is a $1 charge per month per keypair, first [20,000 uses are free per month](https://aws.amazon.com/kms/pricing/).

Type in your keypair name, like “MyAmazonKeyPair”, select RSA and choose .Pem / OpenSSH.

Once created download the pem and store in a safe place. You cannot redownload this file so do not lose!

### Create PUTTY keys, .ppk

This is only required for users of PUTTY.   
You can watch this video or follow directions here.

<https://www.youtube.com/watch?v=bi7ow5NGC-U>

Place your key.pem file (your name will be likely different) created in chapter [Generate AWS keys](#_Generate_AWS_keys).

Load PuTTyGen program, in the program press ‘load’ button and load the PEM key. Click button Save Private key and give it a name. This will generate the PPK file required in chapter [Putty - Terminal](#_Putty_-_Terminal).

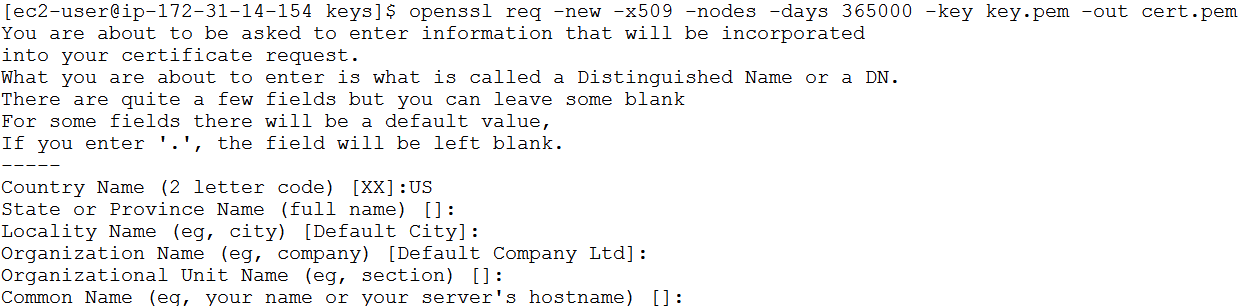
### Create Public RSA key

This is required to connect your GITHUB account to your AWS instance if your repository is marked private.

From command line change permissions from your private key, then type this commend to create an RSA public one:

chmod 600 key.pem  
ssh-keygen -y -f ./key.pem >publickey.pub

### Create cert.pem from key.pem

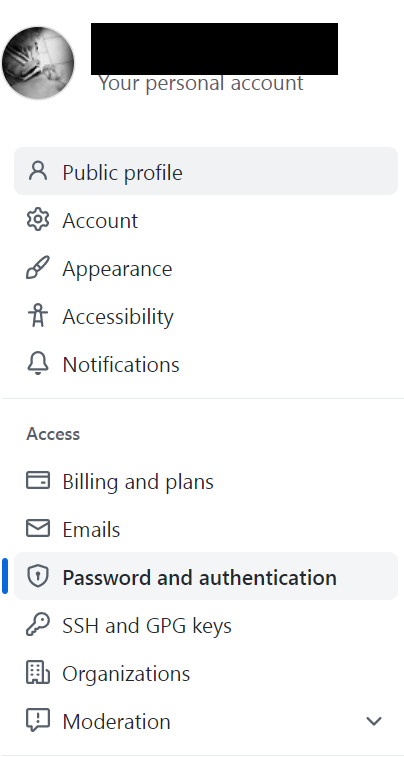
*For NYL Training there is no need to generate cert.pem, however this is put here to help if you need these keys.*  
Once you have your server running, and connect via terminal, you can create cert.pem from your key.pem.   
After connecting to your server via terminal program, you will be in your user directory.

Type “vi key.pem” and press enter, then hit the letter a. (this is to add text to the file)  
Open the pem file from [Generate AWS keys](#_Generate_AWS_keys) , and copy the contents. Now return to the terminal and right click to paste the text. Then hit the ESCape key and type :wq!

From the command line type the following commend to generate cert.pem from the key.pem.  
openssl req -new -x509 -nodes -days 365000 -key key.pem -out cert.pem

You can vi each file to view the contents and press ESCape and typing :q! to verify contents.  
With the files created you can now place them in the directory to allow remote desktop connections using MATE.

## Creating GitHub account

GitHub is a web hosted service that allows you manage source code. This will be used in conjunction with Terraform, learning infrastructure as code. Please note, it is strictly forbidden to place any NYL assets to your personal GitHub account. This is to be used only for training material.  
*You must use a personal email address for the account.*

### Video guide to setup GitHub Account

<https://www.youtube.com/watch?v=iv8rSLsi1xo>

GitHub login  
<https://github.com/>

### TIPS

1. Under settings add additional information.  
   For your personal account do NOT reference New York Life as part of your personal account.
2. Recommend enabling MFA to secure your account.  
   One option is to install Google Authenticator on your device and scan the QR code.
3. GitHub has an app for your phone, making some of the work easier.

## Creating Terraform account

Goto <https://app.terraform.io/> and open an account.  
Recommend using the same email address as used for GITHub.  
Recommend enabling 2FA.  
  
Create Tokes

Navigate settings, then “Tokens” and name your tokens, for example, NYLTrainingServer.

Be sure to copy the token! It will not be accessible again.

# Workspace

## Workspace Options

To get the most out of your time taking training courses, it is recommended you can use the tools. A training workspace requires a personal GITHub account, Terraform account, AWS account, and AWS V2 CLI installed on a desktop.

### Create and configure an AWS EC2 instance

You can create your own AWS Training server on our own AWS Account!  
This is the recommended path, as this will give you immediate hands-on experience.  
Ensure you complete all [Prerequisites](#_Prequisites) , then resume this guide starting at [creating EC2 an Image](#_creating_EC2_an).

Desktop Install  
You may deploy AWS CLI and Git tools onto your PERSONAL desktop. NYL does not permit installing these tools onto your corporate PC for personal account use. See chapter [Desktop Installation](#_Desktop_Installation).

### TurnKey Training Server

Load a turn-key training server on your personal AWS account.  
This server is ***Not certified by NYL*** and is NOT supported by NYL Cloud team.This image is not permitted on NYL assets, and is a personally (at this time) maintained by [Michael\_J\_Murphy@newyorklife.com](mailto:Michael_J_Murphy@newyorklife.com).  
Ensure you complete all [Prerequisites](#_Prequisites) , then complete chapter [Turnkey Training Server](#_Turnkey_Training_Server).  
Then resume this guide starting with [Connecting to EC2 instance](#_Connecting_to_EC2).

## Turnkey Training Server

*NOTE: Not yet done 4/2022*

## Desktop Installation

Github, terraform, and AWS CLI are all deployable on a windows desktop. At NYL you are not permitted to install these onto your corporate machine for training. You may install these onto a personal desktop instead of deploying an AWS EC2 instance for training. This guide is NOT complete! Please enhance as you learn!

### Installing AWS CLI

WS CLI is the command line interface that allows you to connect to your AWS infrastructure.  
This will ONLY work with windows 10 or greater.  
  
**Install AWS CLI**Video to guide install AWS cli.  
<https://www.youtube.com/watch?v=11aYe_VWMg0>  
  
Connecting to your AWS account<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-prereqs.html>

Troubleshooting:  
<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>  
 **AWS CLI Download**

<https://awscli.amazonaws.com/AWSCLIV2.msi>

<https://awscli.amazonaws.com/>

### Installing GIT

Follow the guide on GitHub to install on your server/workspace!  
<https://github.com/git-guides/install-git>

To download on your local desktop, go to:

<https://desktop.github.com/>

# Creating EC2 an Image

AWS EC2 Image builder creates a repeatable process to create a server image ready for deployment. In the event you need to make a change to the build, it is as simple as modifying the Image builder recipe and rerunning.

## Video

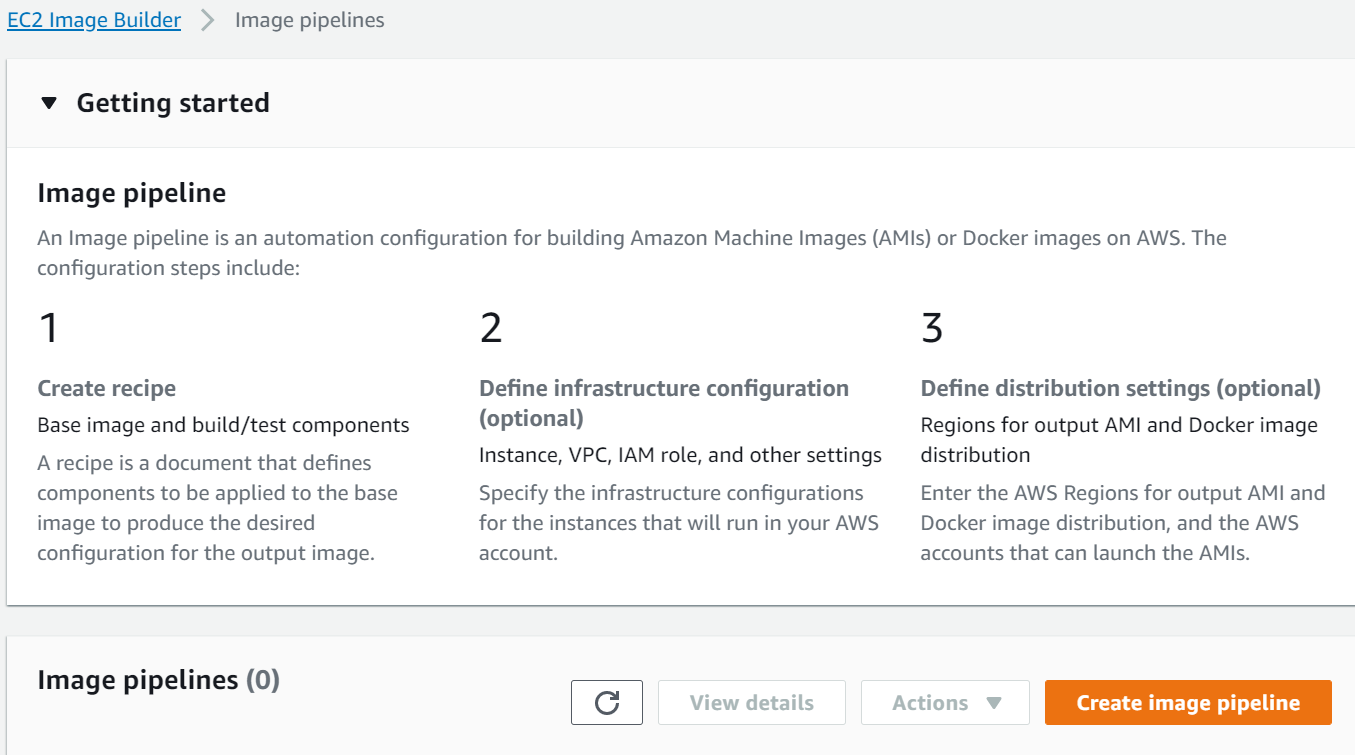
This video doesn’t cover exact recommendations, but is a great over-view. Install these [Recommend Components](#_Recommend_Components).

<https://www.youtube.com/watch?v=btAL8GWX9QU>

## Image Builder Process

Log into your AWS account, in the top search bar type “EC2”, and select EC2 Image Builder. This is a workflow to assist in creating your EC2 instance.

Select “Create Image Pipeline” on far right. Name pipeline something like CloudTrainingServer. Choose schedule to be manual.

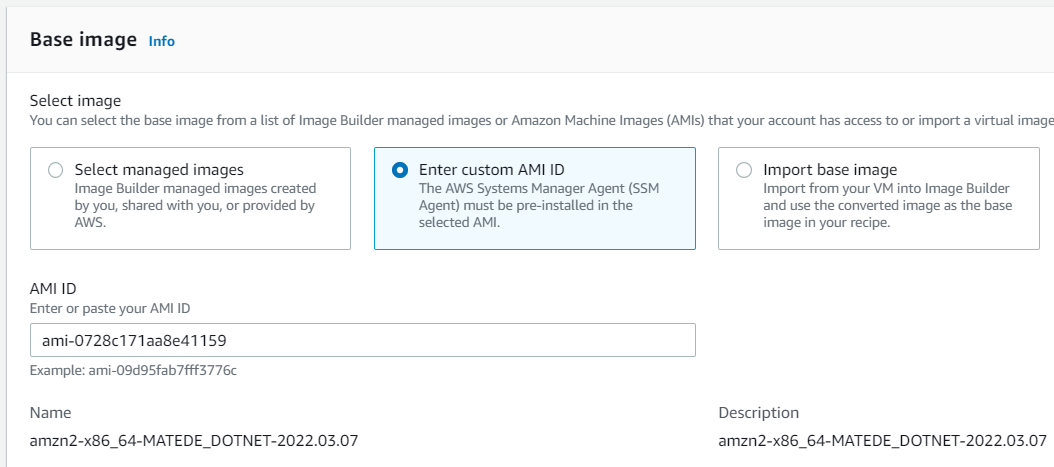


Select “create new recipe” and choose “AMI”.

### Custom AMI – Recommended

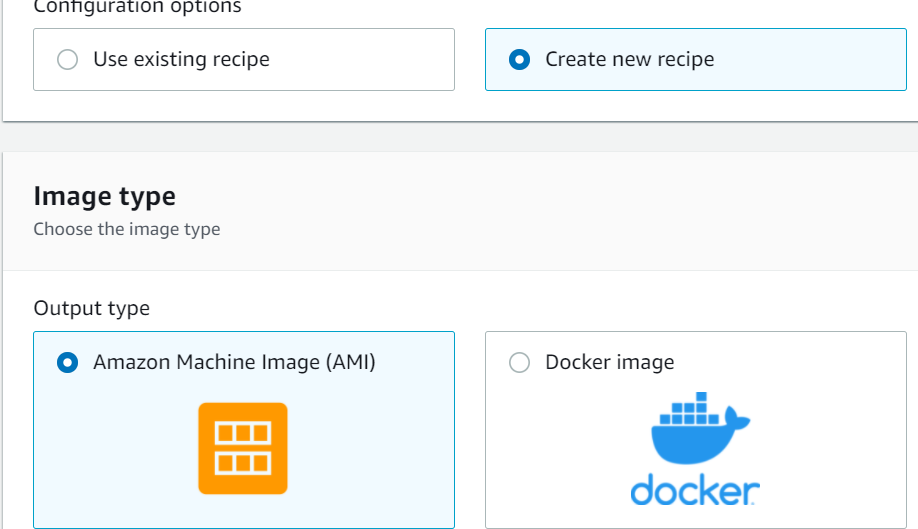
Name the configuration, for example “LinuxNetWorkspace”. (This guide covers linux .NET deploy)

Select custom AMI ID, and enter in ami-0728c171aa8e41159, then go to section [Recommend Components](#_Recommend_Components) .   
NOTE Newer AMI ID will be created, this ID is correct as of 4/20/22.   
To find this image and latest AMI ID, follow [this video](https://youtu.be/btAL8GWX9QU?t=63), or go to “EC2 > Launch Instances > Browse more AMI’s” and Filter on Linux.



### Standard AMI

Name the configuration, for example “LinuxWorkspace”. (This guide covers linux deploy)  
  
Select “Quickstart” and select “Amazon Linux 2 Kernal 5 x86” (or a different of your choosing).



### Recommend Components

Install components onto your image to ‘play’ and learn more about AWS services!

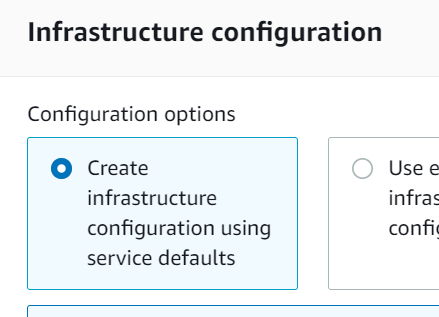
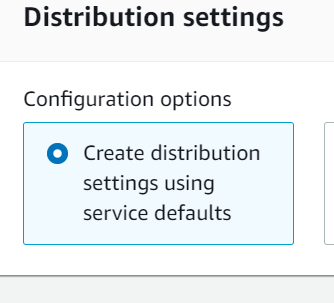
The following are recommended but NOT required for training server. Services can be installed later.  
Recommend the first item in the “order of install” be the Linux Kernel mainline.

|  |  |  |
| --- | --- | --- |
| **Required** | **Package Name** | **Description** |
| Y | aws-cli-version-2-linux | AWS CLI (required for training) |
| N | Amazon-Cloudwatch-Agent-Linux | Allows for greater detailed AWS console monitoring |
| N | amazon-corretto-11 | Pure opensource Java by AWS |
| N | aws-codedeploy-agent-linux | Able to use AWS code deploy for CICD pipeline |
| N | nodejs-12-lts-linux | To run NodeJS code |
| N | python-3-linux | Python |
| N | mate-de-linux | NOTE ONLY install \*IF\* not using the .NET MATE AMI Image from section [Custom AMI – Recommended](#_Custom_AMI_–) |

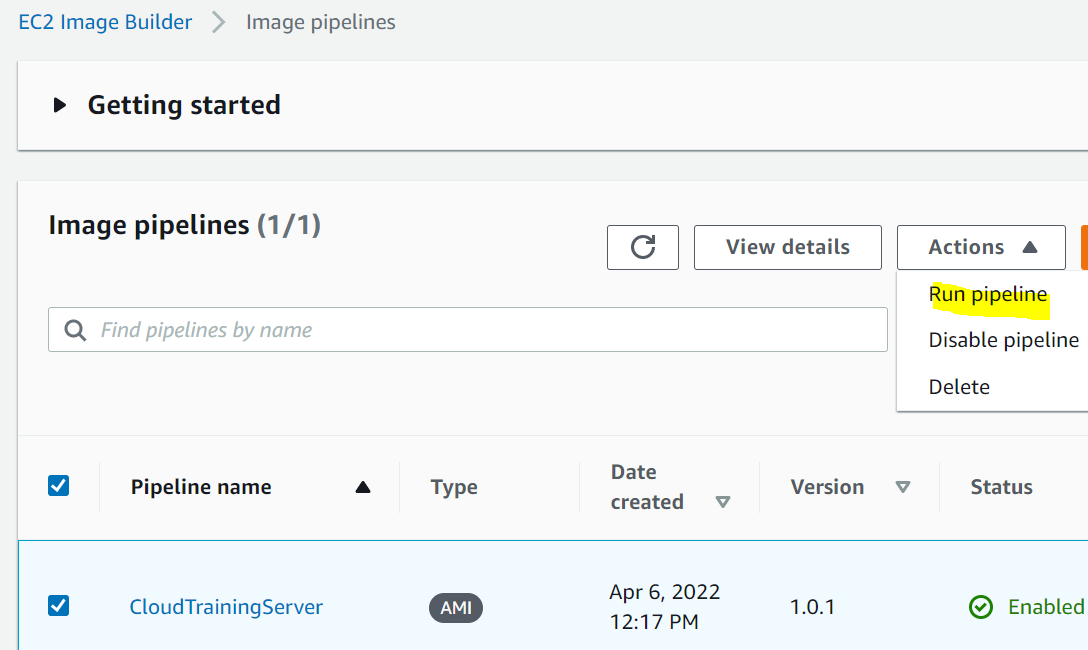
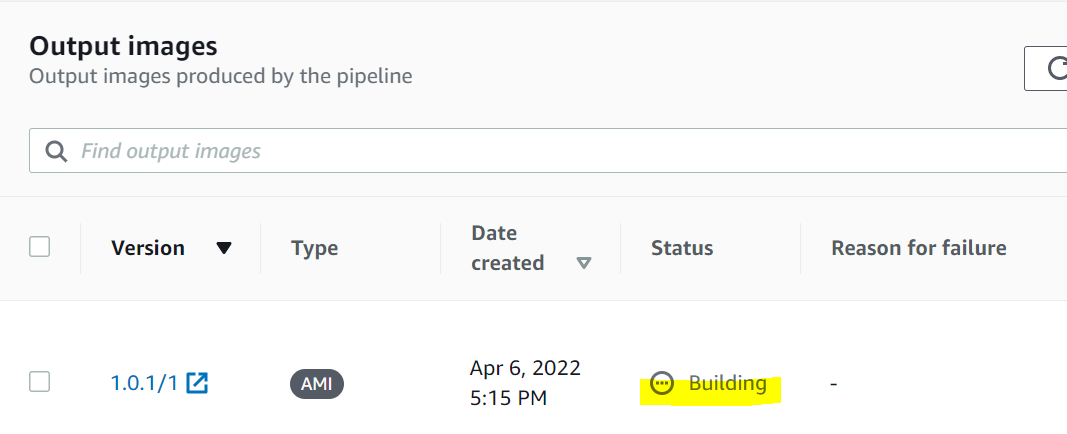
Install components onto your image to ‘play’ and learn more about AWS services!

The following are recommended but NOT required for training server. Services can be installed later.  
Recommend the first item in the “order of install” be the Linux Kernel mainline.

|  |  |  |
| --- | --- | --- |
| **Required** | **Test Component** | **Description** |
| N | chrony-time-configuration-test | Verifies Server time synchronization is correctly configured. |
|  |  |  |

Recommend giving 8GB of disk space. Free Tier is up to 30GB per account:  
Note there is a charge for disk storage!  
  
Click next, and choose “create infrastructure configuration using service defaults”. And “Create Distribution settings using service defaults”.

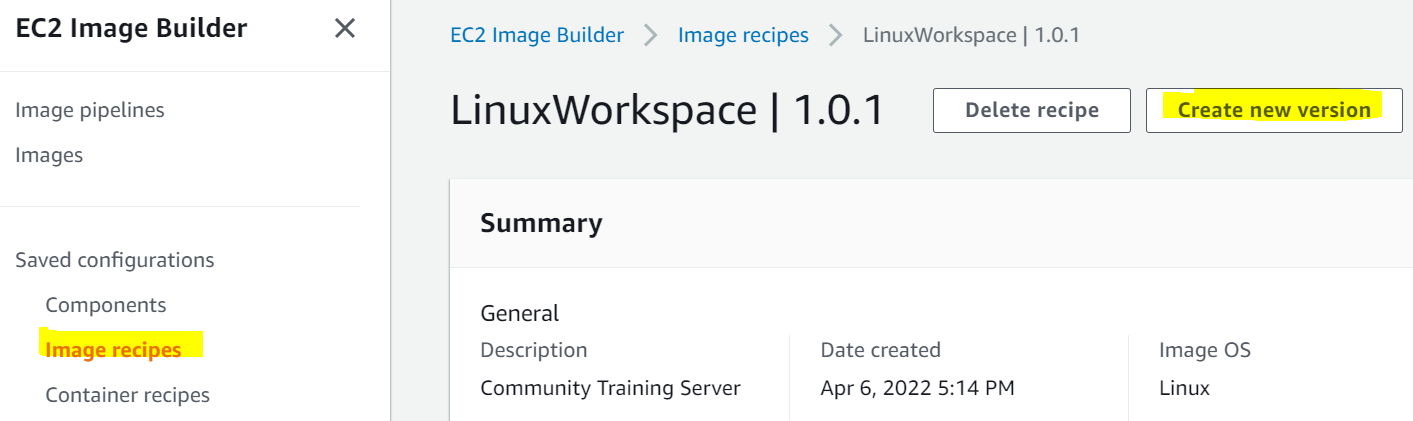
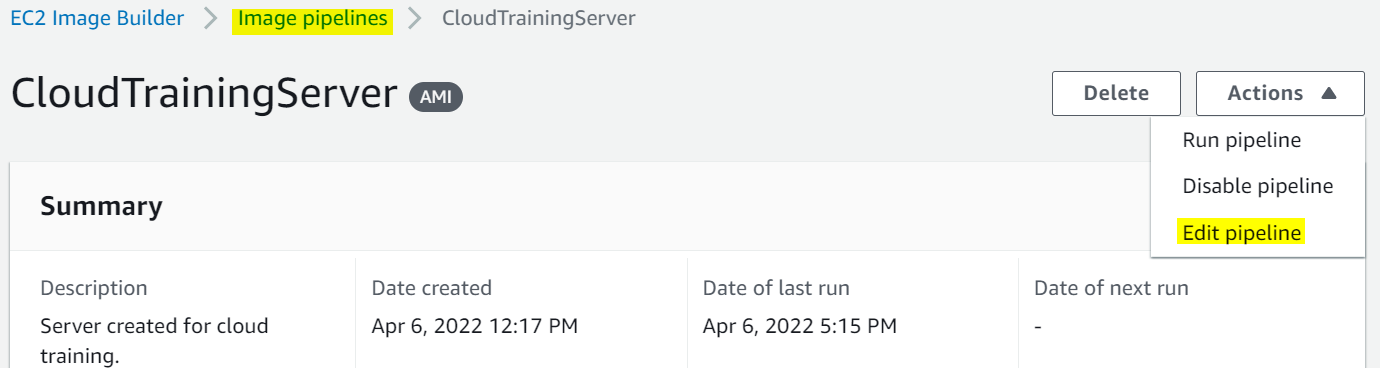
Review and click “Create Pipeline!”

The Pipeline is a “Recipe” that can be run to recreate a server” To create your first   
server select the pipeline “CloutTrainingServer” and under actions select “Run pipeline”.

To view status, click on your pipeline name, in screenshot below “CloudTrainingServer”. You can view the output image status. Building may take upwards of an hour to complete. (5:15pm start)

### Fixing Errors

If you have an error during creation process, you can edit the Image recipe by creating a new version and edit. Then edit the pipeline and change to your new image recipe and save. Repeat until errors are corrected.



## Creating EC2 instance

*Graphical user interface, text, application

Description automatically generated*In the search bar, type “EC2”, then clock instances on far left, and on far right launch instances.

*Graphical user interface, text, application, email

Description automatically generated*

Click on left “My AMI” and select your AMI.

Choose your AMI type, recommend free tier eligible server, then click configure instance details.

*Table

Description automatically generated*

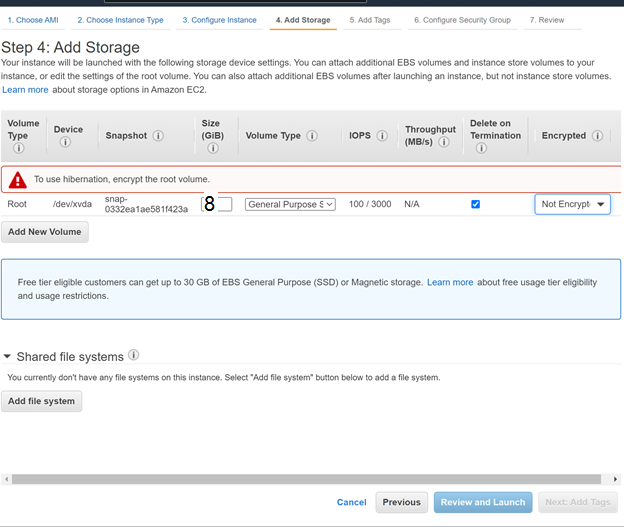
You can Optionally select a “spot instance”. This is a much cheaper version of the server if you end up using it past the free tier usage. A spot instance can go down, at anytime, in the event the region extra capacity is purchased by a spike in demand. Since this is used for training, I selected spot with a maximum payment of 0.01 per hour. *Graphical user interface, table

Description automatically generated*If the costs exceed a penny an hour, my server will go down.

*Graphical user interface, text, application, email

Description automatically generated*

As you learn more about AWS common settings that you will set is Capacity reservation, IAM Role, and Monitoring. For now default is fine.

**

Recommend 8 GB of space, the free tier is 30GB. For the training server about half the space will be used by OS and utilities, leaving plenty of space for DevOps work.

*Graphical user interface, application

Description automatically generatedGraphical user interface, text, application, Word

Description automatically generated*Select create new security group and choose defaults. NOTE: the SOURCE IP on the far right can be limited to your IP address, select “MY IP”. If you try to connect from any other IP you will be denied. If you have problems connecting in the future, be sure to check if you IP address has changed. You can modify the IP allowed after deployed and while the server is still running

.

Review and Launch the instance!

*Graphical user interface, text, application, email

Description automatically generated*

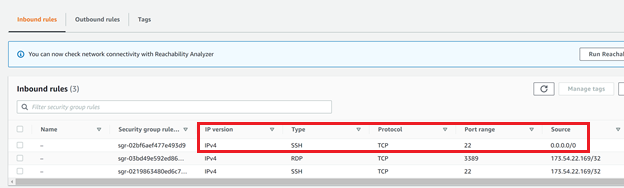
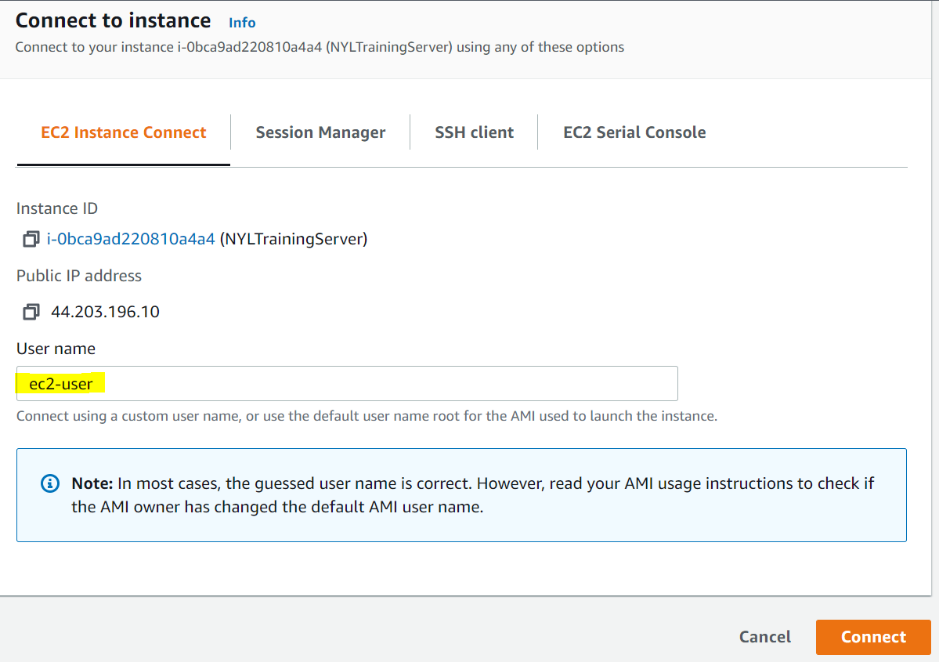
During Launch you will be asked to select a keypair, choose the one already created in previous step!

If it fails review errors and retry!.  
  
Be sure to edit this community document to capture how to handle errors.

# Connecting to EC2 instance

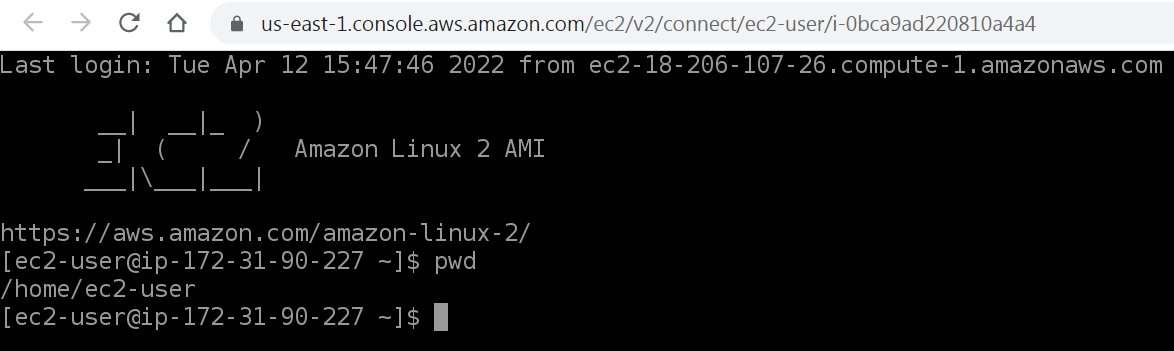
You have a variety of options to use your Linux instance! To gain command line access, you can use your web browser and connect using AWS console, a PUTTY connection (PC program with remote terminal),or using a remote desktop connection to a graphical interface! Choose one or connect using all three.

## AWS console – Browser terminal

AWS has a built-in text/command line access via the console.  
To connect, you must have port 22 available to the ‘world’ and not restrict by IP. To change:  
EC2 > [Instance ID] > Security > [Security Group ID] > “Edit inbound rules”.  
Although the port is open to the world, to connect the person must have your AWS Keypair.

To connect via console:  
EC2 > [Instance ID] > Connect > EC2 Instance Connect

Change user name from “root” to default user “ec2-user”. (or a different user if you change the default user)



Your screen should look something like this:  
Type “pwd” to see it inaction!

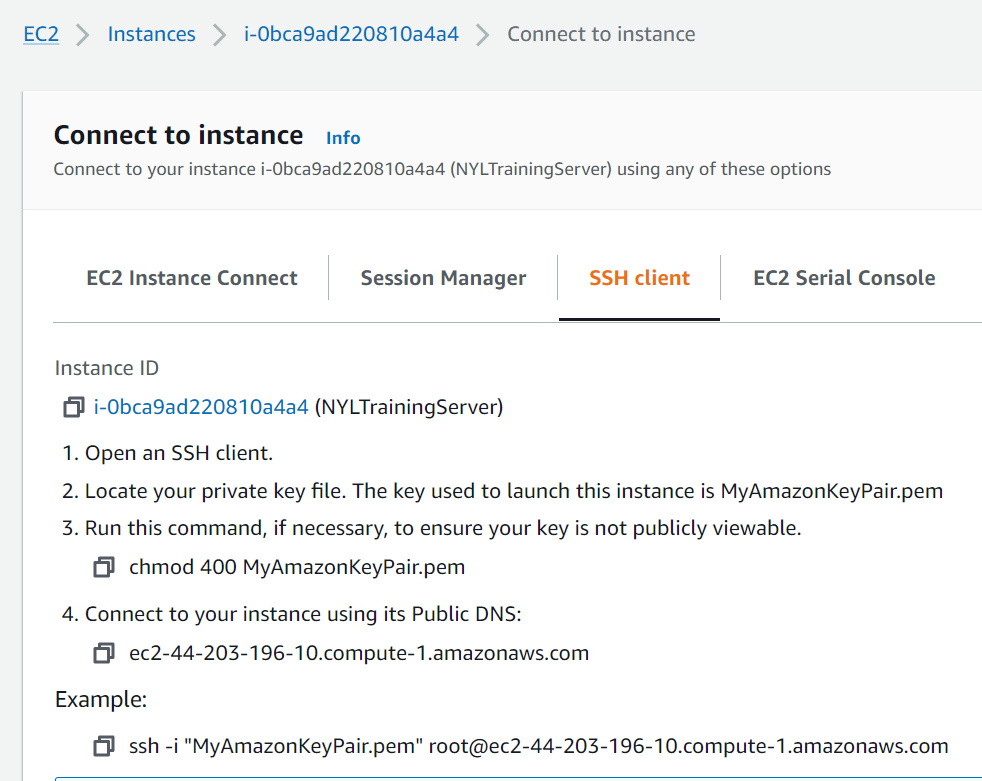
## Initializing Instance

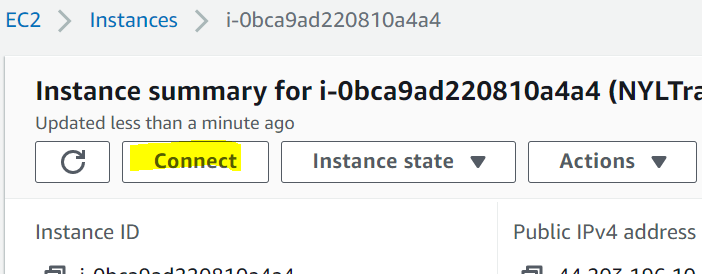
Text

Description automatically generatedOnce you connect with PUTTY or AWS console, you can optionally set a password for ec2-user. This is required before you can connect to your instance with a graphical desktop. For other connections, you do not need to set one.  
To change the password, from command line type   
“sudo passwd ec2-user” and set your password!   
Note: for Training server, set to NYLTra1n1ng.  
  
As best practie, update the server on initial connection by typing:  
sudo yum update -y

## Identify connection information

To connect to your Linux instance, you can use windows PowerShell on your laptop, or other utilities such as PUTTY. To get details on your instance, go to your instance and click on the ID.

Then click on connect

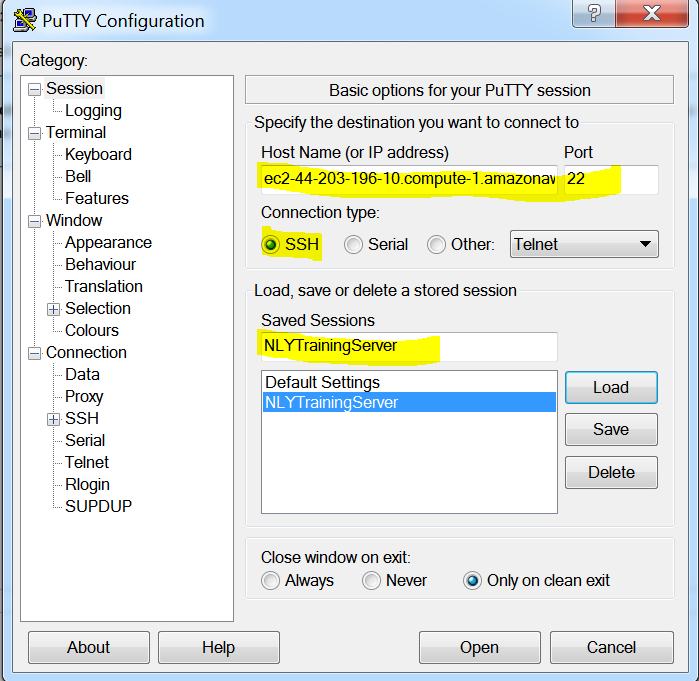
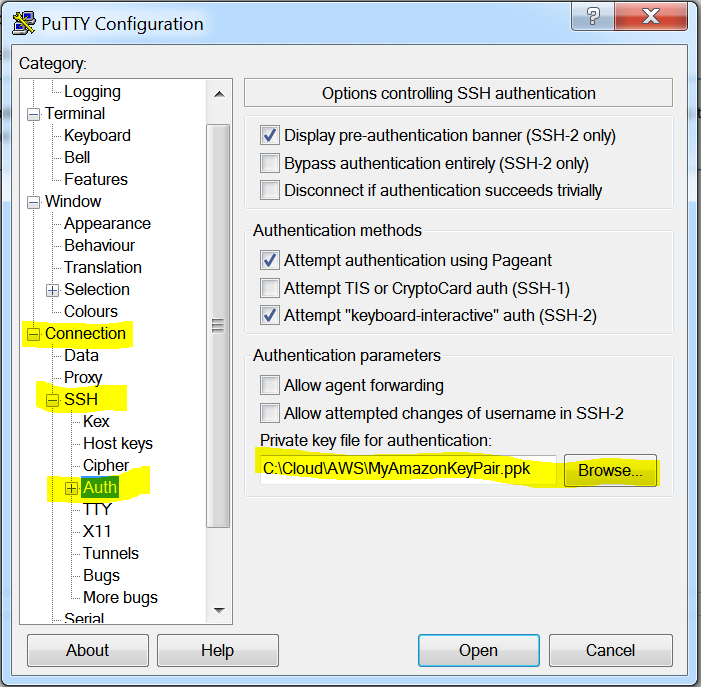


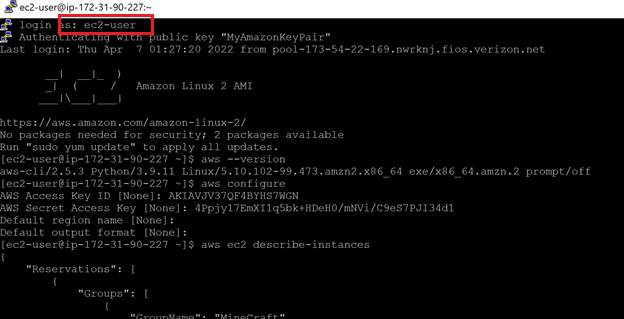
Then select SSH Client (or other options) to connect!

## Putty - Terminal

There are many resources available to help in using PUTTY. This guide is here for your convenience.  
*NOTE: NYL VPN will prevent PUTTY connections. To connect do not use the corporate network.*

A screenshot of a computer

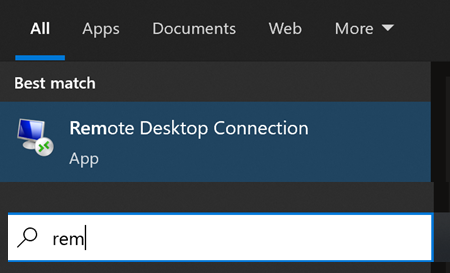
Description automatically generatedAfter [Putty is downloaded](https://www.putty.org/) and installed fill in information shown on the right. Then click “session” and SAVE!  
Once saved click the open button. You will be prompted to trust the SSH fingerprint, click accept.

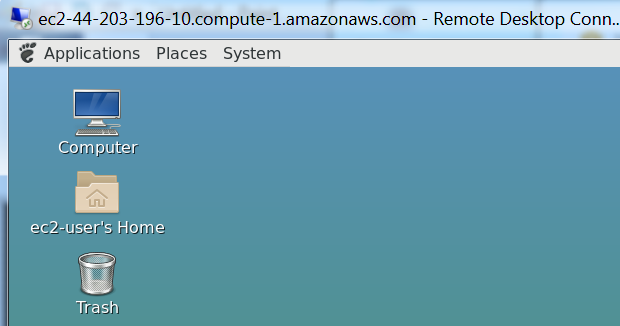
*You will be prompted for the user name, enter user “ec2-user” and press enter. You will NOT be prompted for a password since you have the keypair, and you are in!*

Graphical Desktop - Mate   
  
Connecting to Graphical Desktop

Video on configuring a server and connecting to mate:  
<https://www.youtube.com/watch?v=btAL8GWX9QU&t=187s>  
  
When you created your EC2 instance, if you installed “mate-de-linux” you can connect to your instance using a graphical desktop! Before you do, be sure to set a password for ec2-user as discussed in previous section.  
*NOTE: NYL VPN will prevent remote desktop connections. To connect do not use the corporate network.*

To be able to connect, you must have port 22 available. To change:  
EC2 > [Instance ID] > Security > [Security Group ID] > “Edit inbound rules”.  
  
It is recommended to NOT open to the world and use your IP address.

**Graphical user interface, application

Description automatically generated  
In Windows, on the search bar type “rem” and choose Remote Desktop Connection.   
Enter the connection information and user “ec2-user” (the server DNS) and click connect.  
If you have issues check the “always ask for credentials” and be sure to enter the password set in section [Initializing Instance](#_Initializing_Instance).  
  
If a desktop login is presented, enter user name and ec2-user, and enter our password.  
  
And you are in! You can now interact with your server using a graphical interface!

  
Installing text editor

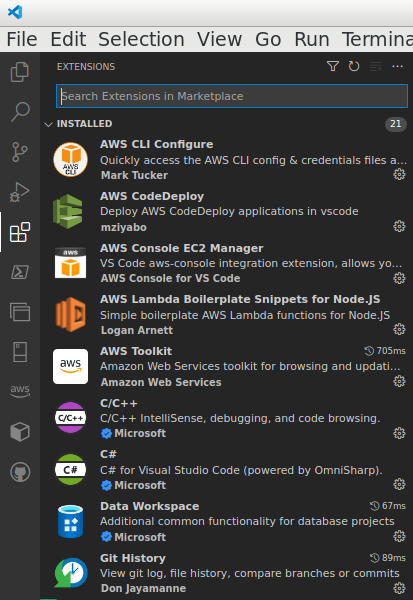
To edit files using the UI desktop, a common lightweight editor is gedit.

From the UI desktop, open a command session and type.

sudo yum install gedit

gedit

An editor should appear, it will also be automatically registered as your default editor when using the file manager to edit a file.

Installing Visual Studio Code

Visual studio Code is 100% open source and one of the most common coding editors. There are extensions for every language, and for working with AWS!

Server Resources  
The program requires more disk and requires more resources than a t2.micro.  
Set your server to t2.medium or t2.large for better performance. For intense compiling, larger servers may be needed. Recommend 12-16GB of disk to allow for workspace.

Larger servers cost more! Recommend you start/stop the server in the AWS console to minimize costs.

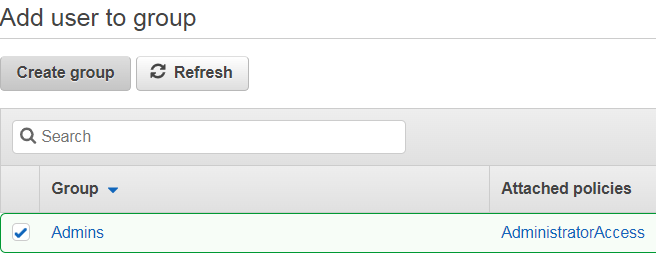
Installation  
You can follow this video or directions below.  
<https://youtu.be/btAL8GWX9QU?t=405>  
  
On the target desktop, download the 64 bit .tar.gz at:  
<https://code.visualstudio.com/download>  
  
If using the image described in [Custom AMI – Recommended](#_Custom_AMI_–), it will place the download under /home/ec2-user .  
On your Linux desktop go to Applications > System Tools > MATE Terminal, then type:  
cd Downloads  
ls -ltr  
tar -xvf code-\*  
cd VS\*/bin  
./code

# Configure Server

## Configure AWS CLI

### Configure CLI

First, on your AWS console type “IAM” to access the Identity and Access Management Services.  
Go to Users, and click button “Add users”

Name the user , like “NYLTrainingServer”, select Access Key, and click next.   
  
Click “create group” and lets create a group to assign this user to. Name the group “Admins” and c hoose “AdministratorAccess” to give this group full access permissions.  
Click next and when the credentials are created, copy the access ID and secret. NOTE: You can never get the secret again. Optionally you can download as a CSV.  
Follow these steps to setup AWS CLI.  
  
$ **aws configure**  
AWS Access Key ID [None]: ***AKIAIOSFODNN7EXAMPLE***  
AWS Secret Access Key [None]: ***wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY***  
Default region name [None]: ***us-east-1***  
Default output format [None]: ***json***

For more info:  
<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-quickstart.html>

You can try a command by typing:  
aws ec2 describe-instances

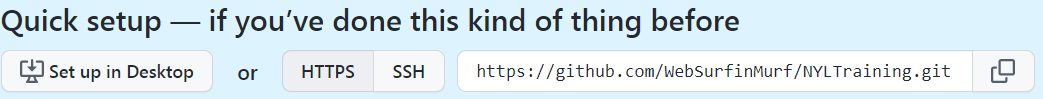
The AWS CLI is ready for use!

### Common AWS CLI commands



## Configure GIT

### Create repository

If you have created a repository, skip to the next section.  
Login to your GitHub account (see chapter [Creating GitHub account](#_Creating_GitHub_account) ) and click “create repository”  
Name it something like “NYLTraining”. Recommend making the repository private, to ensure if you checking any credentials, it won’t be shared to the public. See also [GitHub Quickstart](https://docs.github.com/en/get-started/quickstart/hello-world) guide to learn more.   
  
Copy the HTTPS URL by clicking the “copy to clickboard” button on far right of the screen.

### Install GIT

**Video**<https://www.youtube.com/watch?v=p_WSG-fsfYg>

Steps:

From command line in type:

cd  
sudo yum install git -y  
git -version

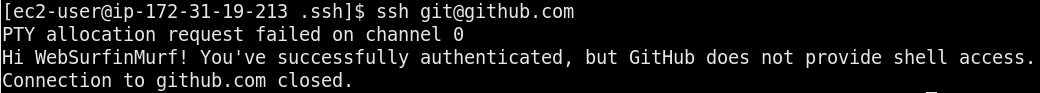
### Configure GIT

The steps here is to configure GITHUB to work with your private instance of GIT.  
These steps are simpler if you make your GIT repository public.

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

From comment line type:  
ssh-keygen -t ed25519 -C "YOURGITEMAILADDRESS"

From github.com, in upper right click the drop down and select settings.  
Then on the left choose “SSH and GPD keys”, then click “new SSH keys”.  
Enter in title something like “AWSTrainingServerkey”.  
Type:  
ssh [git@github.com](mailto:git@github.com)

You should see something like:  


In GitHub, click on your repository, then tab <Code>, and follow the directions.  
To get this to work, try running two different add origins:

cd  
echo "# Repo2" >> README.md  
git init  
git add README.md  
git commit -m "first commit"  
git branch -M main  
git remote add origin [git@github.com:YOURNAME/YOURREPO.git](mailto:git@github.com:YOURNAME/YOURREPO.git)  
git remote set-url origin [git@github.com: YOURNAME/YOURREPO.git](mailto:git@github.com:WebSurfinMurf/NYLTraining.git)  
git push -u origin main

You are configured!

## Implement Terraform

### Internet guides

**Video**

<https://www.youtube.com/watch?v=XxTcw7UTues>

<https://www.youtube.com/watch?v=LZtSZexSYG4>

**Web Page**

<https://cloudaffaire.com/how-to-install-terraform-in-aws-ec2/>

### Terraform Installation

Go to <https://www.terraform.io/downloads> , locate the link under Linux, AMD64, right click and copy the URL.  
As of 4/20 -   
<https://releases.hashicorp.com/terraform/1.1.9/terraform_1.1.9_linux_amd64.zip>

Go to your server and type “wget <https://releases.hashicorp.com/terraform/1.1.9/terraform_1.1.9_linux_amd64.zip> “  
(Change the target if a newer version is available)  
Then type:   
unzip terraform\_1.1.9\_linux\_amd64.zip  
sudo mv terraform /bin  
sudo chown -R root:root /bin/terraform  
terraform -version

sudo yum install jqy

If it doesn’t return a version, please diagnose your issues.

Type command below and copy the API token created in prerequisites [Creating Terraform account](#_Creating_Terraform_account) .  
Type the command below and enter your API token.   
terraform login

### The Essential Terraform Cheatsheet – Justin CodesCommon Terraform Commands

# Training information

## NYLife Training Curriculum

The latest recommended curriculum for NYL cloud training is available in the link below.   
<https://degreed.com/pathway/e9k6qvrw9o?orgsso=newyorklife>

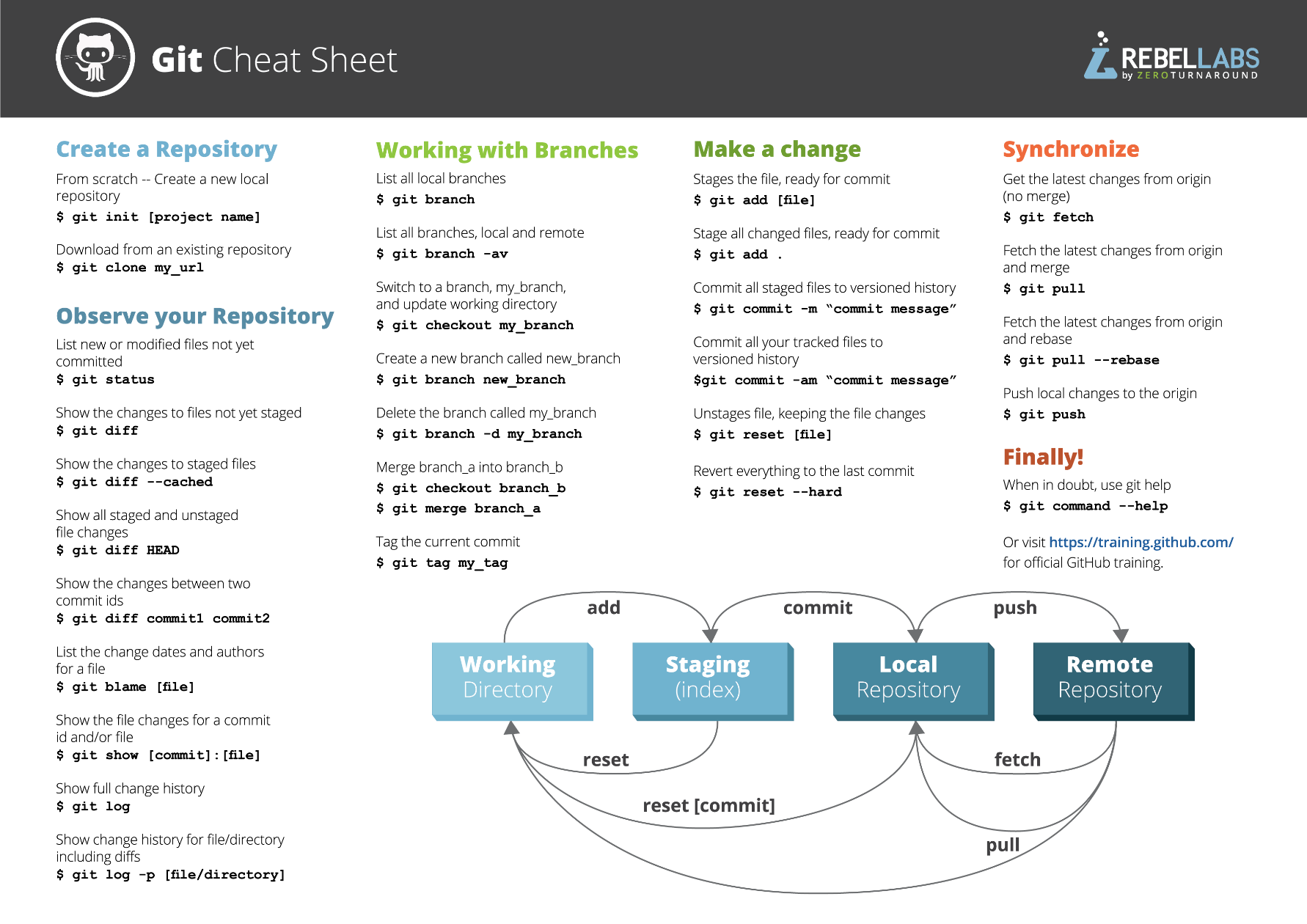
If you cannot access this link or items located on the page, please email [cloud\_services@newyorklife.com](mailto:cloud_services@newyorklife.com) for help.

The training includes using AWS, GitHub and Terraform. To take these courses it is strongly recommended you first setup accounts to use during your training sessions.  
  
These accounts are for personal use and cannot be used for any NYL assets. The accounts can be used to support the training courses. The training courses should remain in the free tier. In the event there is a charge, please see your manager about charging to your annual educational stipend.

# more information

## GIT Notes

### Cheat Sheet

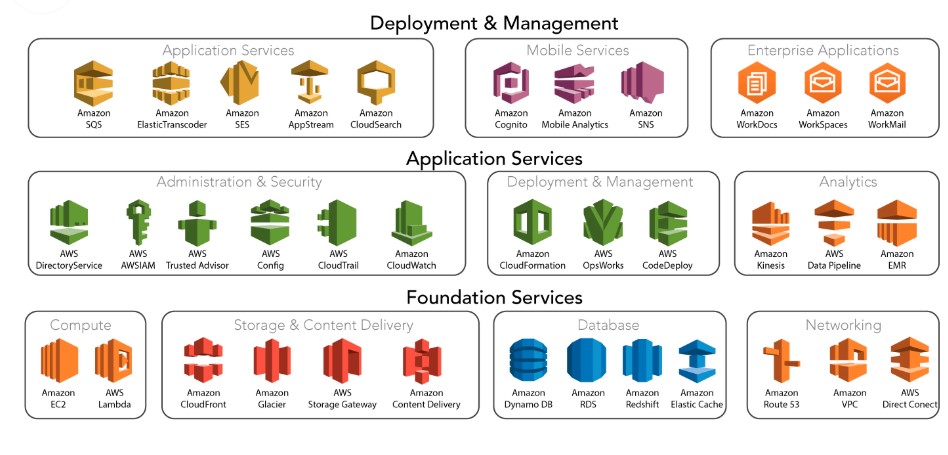


### Quick notes

|  |  |
| --- | --- |
| Command | Description |
| git checkout -b <features>/<featurename> | Create a branch |
| git push --set-upstream origin <features>/<featurename> | Sync branch to github |
| git add <filename or wildcard> | add files to branch |
| git commit -m “comment for commit” | Commit files to branch |
| git push | Pushes files to Github |
|  |  |
|  |  |
|  |  |

## AWS

### AWS Cheat Sheet - Amazon Web Services Quick Guide [2022]Terms Cheat Sheets



### AWS Mobile app

AWS has a mobile that makes it easy to start/stop your instance to save money! Choose a larger, not-free instance such as t2.large and pay 6 cents an hour, that’s about 50 cents for a days work!

