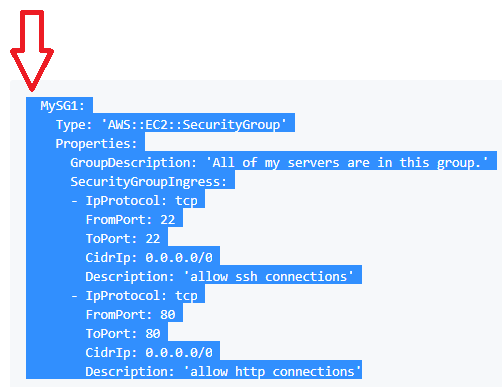
**CloudFormation Lab**

In this lab, we will create a CloudFormation stack that consists of one virtual machine, one security group, and one S3 bucket.

**IMPORTANT - COPY ALL SPACES**

During this lab, you will cut and paste lines from this lab into your CloudFormation script. Make sure you cut and paste everything. There are spaces and lines in some of the places that you may not see.  
For example, when you copy and paste within the security group later on in this lab, there are two spaces on the first line that needs to be copied. If you don't include those lines, your script will not function. YAML cares a LOT about spaces.

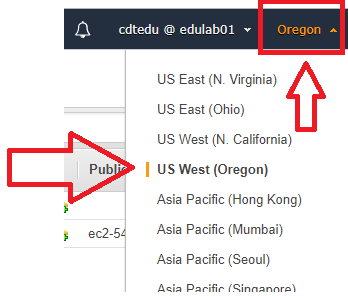
* You will know if you incorrectly copied because it will look like this:  
  
* You will know when you correctly copy everything as it will look like this:  
  

**IMPORTANT - MAKE NEW LINES**

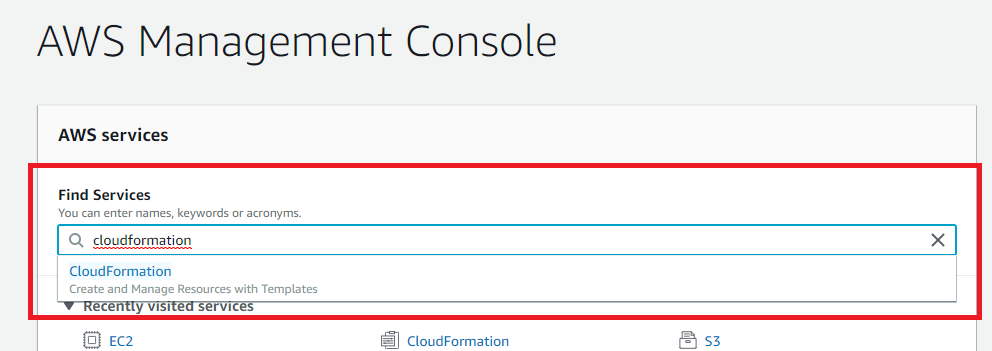
Before you copy and past new sections, be sure to start a new line, and also be sure to start at the beginning of the line.  
  
In CloudFormation, when you press enter to start a new line, it will not start at the beginning of the line.  
  
Make sure you backspace to the beginning of the new line.

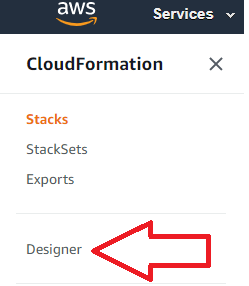
**Getting Started**

Navigate to: url to be provided  
username: to be provided  
password: to be provided

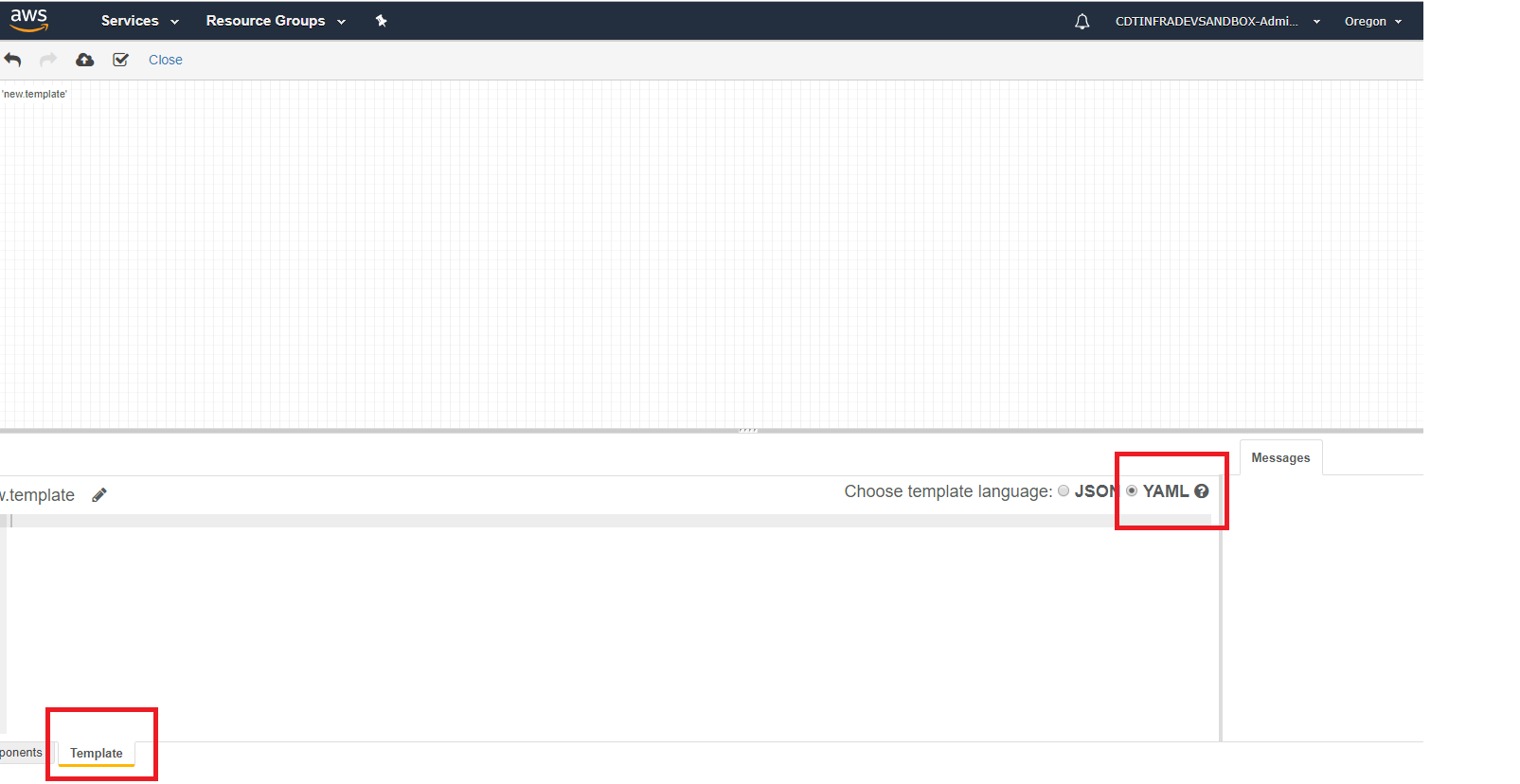
* After you log into AWS, the first thing we need to do is ensure you are in the correct region.
* In the upper-right hand corner as shown in the picture, make sure it says "Oregon." If it does not, click the link and select it from the menu as shown:  
  

**Steps**

* At the home screen, click on the CloudFormation link, or type cloudformation in the find services dialogue box.  
  
* From the CloudFormation Console, click on the Designer link in the left pane.

[](https://camo.githubusercontent.com/14d17539ab7ef087b504c8182cfced78497a458a/68747470733a2f2f6364742d696e74726f636c6f75642e73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f6c616230312f696e6465785f66696c65732f64657369676e65722e706e673f69643d31)

* At CloudFormation Dashboard, change the editor (lower pane) to "Template" tab and Language "YAML."

[](https://camo.githubusercontent.com/fd858fb22a3c1f2762c952e8d0755fe191986766/68747470733a2f2f6364742d696e74726f636c6f75642e73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f6c616230312f696e6465785f66696c65732f7365743279616d6c2e706e673f69643d32)

**Server**

Now we will create a virtual machine, which is referred to as an instance. Paste the following code into the editor:

Resources:

MyServer01:

Type: 'AWS::EC2::Instance'

Properties:

Tags:

-

Key: Name

Value: student\_server

SecurityGroupIds:

- !Ref MySG1

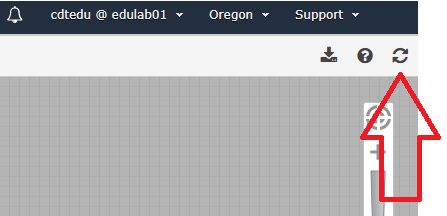
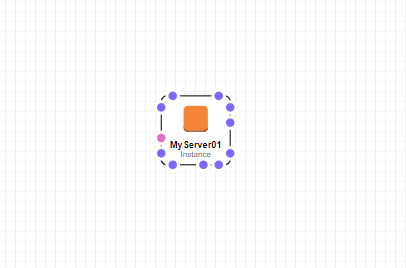
- sg-0ae9944a

ImageId: ami-04b762b4289fba92b

InstanceType: t2.micro

KeyName: forgetme

IamInstanceProfile: CdtStudentInstance

* Look for where the code shows Value: student\_server. Change student\_server to your first name and last initial.  
  This will be used to identify your server later in the lab.
* Click the Refresh Diagram button at the top.  
  
* When done, you will see the server you created in the screen similar to this:  
  

**IMPORTANT**

* For the purposes of this lab, do not click on or move the objects in the diagram. If you do, CloudFormation adds information to the script that will be confusing if you are new to CloudFormation.

**Security Group**

For security reasons, by default, servers in AWS cannot be accessed.  
This is due to all network traffic being blocked unless you create a rule.  
We will create a rule that tells AWS to allow web traffic, and remote access traffic to our server.  
What this means is we will be able to connect to the operating system directly.  
The web traffic rule would allow us to host a website on this server.

Let's create the rule now.

* Copy and paste the following into a new line in your CloudFormation script. Don't forget to add a new line as discussed in the Make New Lines section.

MySG1:

Type: 'AWS::EC2::SecurityGroup'

Properties:

GroupDescription: 'All of my servers are in this group.'

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp: 0.0.0.0/0

Description: 'allow ssh connections'

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 0.0.0.0/0

Description: 'allow http connections'

* Click the refresh diagram button
* You will now see the diagram includes a security group. You will also notice an arrow pointing to the security group.

**Code Check**

At this point, your CloudFormation script should look like the below.  
Feel free to replace what you have with this if you feel there are mistakes in your script. If you see the instance and security group, you're probably fine. If you do replace your script, remember the change student\_server as you did previously.

Resources:

MyServer01:

Type: 'AWS::EC2::Instance'

Properties:

Tags:

-

Key: Name

Value: student\_server

SecurityGroupIds:

- !Ref MySG1

- sg-0ae9944a

ImageId: ami-04b762b4289fba92b

InstanceType: t2.micro

KeyName: forgetme

IamInstanceProfile: CdtStudentInstance

MySG1:

Type: 'AWS::EC2::SecurityGroup'

Properties:

GroupDescription: 'All of my servers are in this group.'

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp: 0.0.0.0/0

Description: 'allow ssh connections'

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 0.0.0.0/0

Description: 'allow http connections'

**S3 bucket**

Your server might need a place to store things for later use. There are many ways to store data in the cloud.  
One of the most powerful ways to store data is using what is called "block-level object storage" or "blob" storage.  
Don't let the complex description fool you though. All it means is it's used to store and access files.  
AWS has a service called S3, or Simple Storage Service, to meet this need.  
The images you see in this instruction set are stored and accessed from AWS S3.  
Let's build one for ourselves now.

Copy and paste the following code into your script and click the Refresh Diagram button.

MyBucket:

Type: AWS::S3::Bucket

You will now see a reference to the S3 bucket in your diagram.

**Code Check**

We are now done building out our template. Double-check that your script matches the below.  
Feel free to replace your script with the below. If you see the instance, security group, and s3 bucket, you're likely fine. If you do replace your script, remember to change student\_server as you did previously.

Resources:

MyServer01:

Type: 'AWS::EC2::Instance'

Properties:

Tags:

-

Key: Name

Value: student\_server

SecurityGroupIds:

- !Ref MySG1

- sg-0ae9944a

ImageId: ami-04b762b4289fba92b

InstanceType: t2.micro

KeyName: forgetme

IamInstanceProfile: CdtStudentInstance

MySG1:

Type: 'AWS::EC2::SecurityGroup'

Properties:

GroupDescription: 'All of my servers are in this group.'

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp: 0.0.0.0/0

Description: 'allow ssh connections'

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 0.0.0.0/0

Description: 'allow http connections'

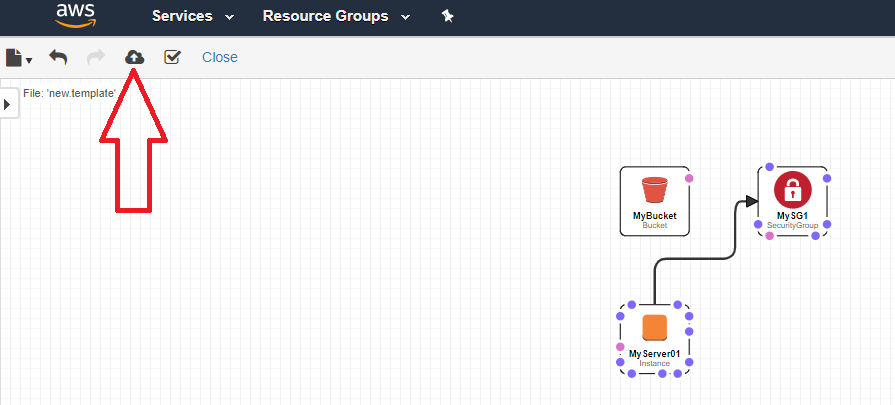
MyBucket:

Type: AWS::S3::Bucket

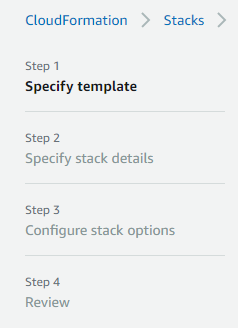
**Launch the Stack**

Now it's time to deploy, or "launch", the stack.

* Click the Create Stack button as shown

[](https://camo.githubusercontent.com/c024b1abc2a32e7581d399472aaa7d20e5023de7/68747470733a2f2f6364742d696e74726f636c6f75642e73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f6c616230312f696e6465785f66696c65732f637265617465737461636b2e706e673f69643d31)

* Launching a stack consists of 4 steps. You will see them on the left side like shown here:

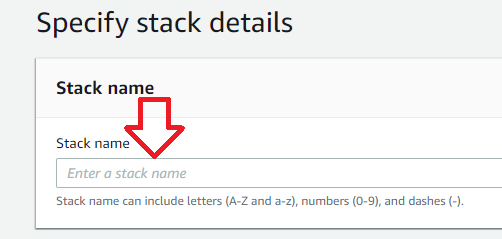
[](https://camo.githubusercontent.com/47bf8bb0f3a86a44cd101aa3ad44df3341c820f8/68747470733a2f2f6364742d696e74726f636c6f75642e73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f6c616230312f696e6465785f66696c65732f737465706f6e652e706e67)

**Step 1**

* Just click next. Nothing to change here.

**Step 2**

* Enter a name for the stack. Use your first name and last initial to help you find it later. Click Next.

[](https://camo.githubusercontent.com/71c4856e3234a85f77a5b41c292e149013fe034b/68747470733a2f2f6364742d696e74726f636c6f75642e73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f6c616230312f696e6465785f66696c65732f737461636b6e616d652e706e67)

**Step 3**

* Scroll to the bottom and click next.

**Step 4**

* You will now be in the review section. Scroll to the bottom and click next.

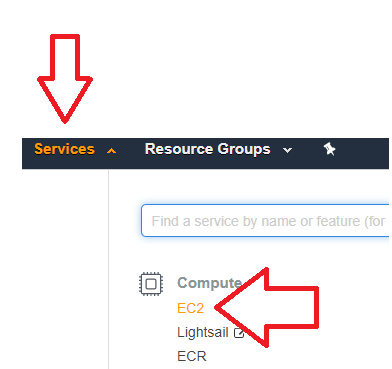
**Wait...**

* We will now wait for the resources to deploy in AWS. This takes a minute or two.
* Click the refresh button from time to time. On the right side, looks like an arrow pointing in a circle.
* If you have any errors please ask a roaming instructor.

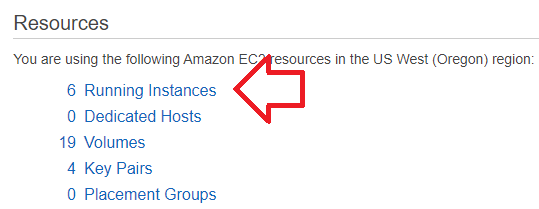
You will know when it's done because you will see the create\_in\_progress indicator turn green. If it turns red, consult an instructor.

**Let's log into our server**

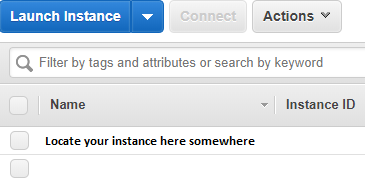
* Assuming everything went ok, click on the services tab (at the top of the screen), then click EC2 as shown:

[](https://camo.githubusercontent.com/fc6f9d1f13c2f232b35a30fc31b8cc758072c918/68747470733a2f2f6364742d696e74726f636c6f75642e73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f6c616230312f696e6465785f66696c65732f6e6176326563322e706e67)

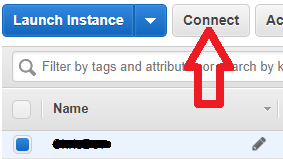
* Click Running Instances near the top (in blue)

[](https://camo.githubusercontent.com/515aeefd44133d3083fae77193f2133262bf6cb0/68747470733a2f2f6364742d696e74726f636c6f75642e73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f6c616230312f696e6465785f66696c65732f72756e6e696e67696e7374616e6365732e706e67)

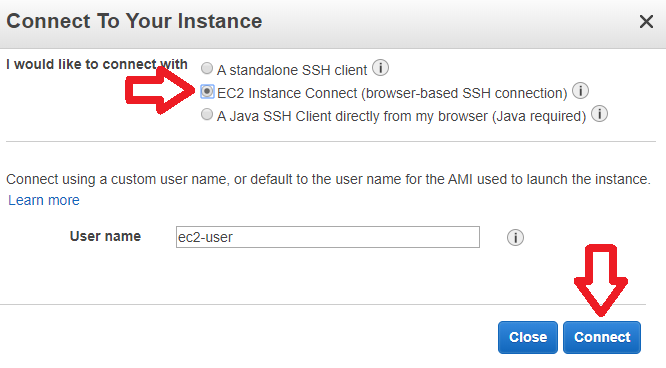
* Locate your server which should be your first name and last initial.

[](https://camo.githubusercontent.com/5351d29ff3b6198f451e06074f32e95f3b1d2183/68747470733a2f2f6364742d696e74726f636c6f75642e73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f6c616230312f696e6465785f66696c65732f6c6f63617465696e7374616e63652e706e67)

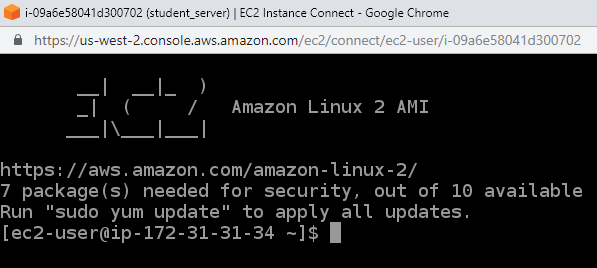
* Click the check-box to the left of it.
* Click Connect at the top.

[](https://camo.githubusercontent.com/d96d3bc08dd81c5f90f2477c2b431e3789e7805c/68747470733a2f2f6364742d696e74726f636c6f75642e73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f6c616230312f696e6465785f66696c65732f636c69636b636f6e6e6563742e706e67)

* Select EC2 Instance Connect (browser-based SSH connection)
* Click Connect

[](https://camo.githubusercontent.com/c3fcd4073ef46f6062f27db4b499003c236e8a5f/68747470733a2f2f6364742d696e74726f636c6f75642e73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f6c616230312f696e6465785f66696c65732f73656c656374636f6e6e6563742e706e67)

If everything went as planned, you will now be connected to your server which will look something like this:

[](https://camo.githubusercontent.com/b075980b2c4636399cb49b33e0e991cfd8098366/68747470733a2f2f6364742d696e74726f636c6f75642e73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f6c616230312f696e6465785f66696c65732f636f6e6e6563746564327365727665722e706e67)

* If you have never used a Linux server, this is generally all you will see. Unlike Windows, Linux is all command line, so there's nothing exciting to look at. This simply demonstrates your successful creation of an AWS Stack using the IoC tool of CloudFormation.

**End of Lab**

Congratulations! For fun, we have added a few thought-provoking questions below. If you think you have an answer, or you are interested in the answer, consult the instructor.

**Questions**

* Notice we created the instance before we created the security group. The instance refers to the security group even though the security group doesn't exist yet. Why do you think this is allowed?
* In the Designer view, why is there no line connecting the S3 Bucket to any other Resource? And what does the line between the Instance and the Security Group represent?
* Back in the CloudFormations console, if you were to select your stack and click delete, what do you think will happen?