Architecting on AWS
Exercises and Solutions

Introduction

Your instructor will ask you to perform exercises from this guide during class. Depending on the topics covered and the length of the course, you may not work on all of the exercises during class. You can use any exercises that you don't complete during class to help continue your learning after class is over.

There may also be additional exercises that your instructor asks you to perform that are not in this guide. In this case, your instructor will provide you with additional instructions.

Many of the exercises in this guide may build upon each other, so it is best if you complete each exercise before moving on to the next exercise. Work at your own pace and understand what you're doing. Remember, the goal is not to complete all of the exercises, but to further your understanding of the concepts discussed.

Note: These labs are by Colin Johnson.

Table of Contents

Exercise 1 – Introduction to AWS 6
Create your Test Instance Security Group: 6
Create your Test Instance: 7
Launch! 11
Login! 11
Delete! 12
Exercise 2 – Create Security Groups 13
Exercise 3 – Security Group Diagram 14
Exercise 4 – Build Gateway and www Servers 15
Exercise 4 – Create Gateway Instance 16
Exercise 5 – Create www Instance 17
Exercise 6 – Build an EC2 Instance using User Data 18
Exercise 7 – Create, Snapshot, and Resize EBS 19
Exercise 8 – Auto Scaling 20
Create Security Groups: 20
Create Load Balancer: 21
Create an EC2 Keypair (if necessary): 22
Create a Launch Configuration: 22
Create an Auto Scaling Group: 23
Confirm Correct Operation: 24
Scale!!! 24
Watch Scaling Happen! 24
Confirm all Instances in Service: 25
Exercise 9 – Elastic Load Balancing 26
Go to the AWS EC2 Console: 26
Create an www ELB Security Group: 27
Modify the Existing www-yourname Server Security Group: 28
Create a www Elastic Load Balancer: 29
Confirm Load Balancer Built and Instance Placed in Service! 33
Lastly, request a page through the Elastic Load Balancer: 34
Exercise 10 – Relational Database Service (RDS) 35
Create a Security Group for the RDS Database: 35
Create an RDS Database: 36
Connect to your RDS Database: 37
Create an RDS Parameter Group: 37
Modify RDS Database Settings: 39
Confirm RDS Database Settings have Changed: 40

Create an RDS Snapshot: 41
Exercise 11 – AWS S3 43
Exercise 12 – Create Public Object 44
Overview: 44
Create an S3 Bucket in the us-west-2 (Oregon) Region: 44
Select the Newly Created S3 Bucket: 44
Upload an Object: 44
Make the Uploaded Object Public: 45
View the Uploaded Object: 45
Exercise 13 – Cross Region Replication 47
Create an S3 Bucket in the us-west-2 (Oregon) Region: 47
Upload an Object to your Oregon S3 Bucket: 48
Confirm the Object has been replicated to US-Standard S3 Bucket: 48
Exercise 14 – IAM User Access 49
Overview: 49
Create an S3 Bucket in the us-west-2 (Oregon) Region: 49
Create a new IAM User: 49
Allow Limited S3 Access to Your New User: 50
Login to the AWS Console with your new user: 51
Upload an Object to the app-access-yourname Bucket: 51
Exercise 15 – CloudFormation 53
Overview: 53
Create the VPC and Web Server Stack: 53
Update VPC and Web Server Stack: 54
Delete VPC and Web Server Stack: 54
VPC with Web Server ASG.JSON 55
Exercise 16 – AWS Identity and Access Management 60
Exercise 17 – IAM Create Group User 61
Overview: 61
Create an IAM Group: 61
Create an IAM User: 62
Login with the newly created IAM User: 63
Exercise 18 – Virtual Private Cloud (VPC) 64
Option 1: Build a VPC with Public and Private Subnets 64
Option 2: Build a VPC Containing two Public Subnets and Web Servers, *optionally locking down with Network ACL* 65
Exercise 19 – Build out Simple VPC 66
Overview: 66
Create a VPC: 66

Create an Internet Gateway:.....67
Create VPC “Public” Subnets:.....68
Create Route to Public Internet:.....69

Exercise 1 – Introduction to AWS

Overview

We'll be starting the hands-on portion of the AWS Introduction module by building a security group and a server in the us-west-2 region of AWS.

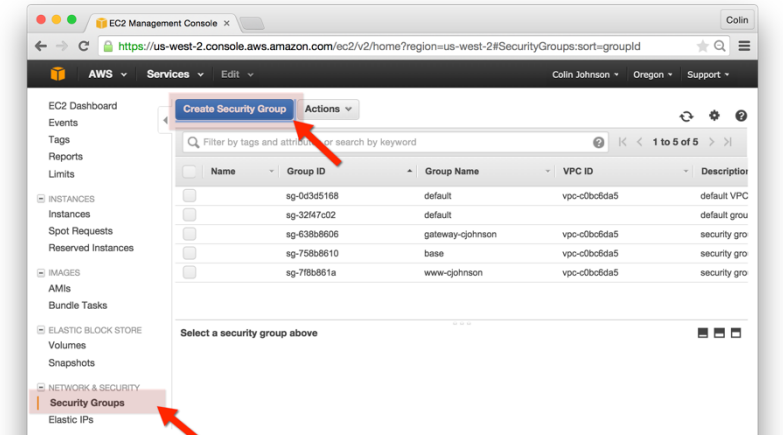
Instructions

Create your Test Instance Security Group:

The procedure will below will create a Security Group that will be used to allow port 22 (ssh) ingress to your newly created "Test" EC2 instance.

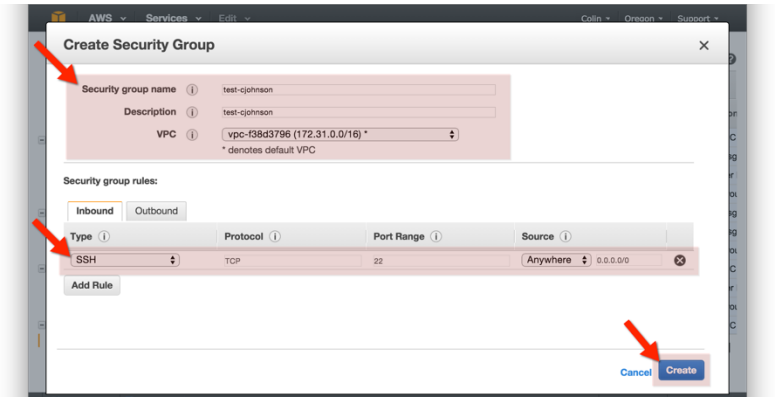
1. Go to the AWS EC2 Console and select "Security Groups" from the left-hand navigation bar.

2. Click "Create Security Group"



3. In the "Create Security Group" window, enter the following values:
   a. Security group name: test-yourname
   b. Description: security group for test-yourname Instance
   c. VPC: chose the "default VPC" – this VPC will be denoted by an asterisk at the end of the VPC name

d. Security group rules:
    i. Inbound
        1. Type: SSH
        2. Protocol: TCP
        3. Port Range: 22
        4. Source: Anywhere / 0.0.0.0/0
        5. Press “Create”

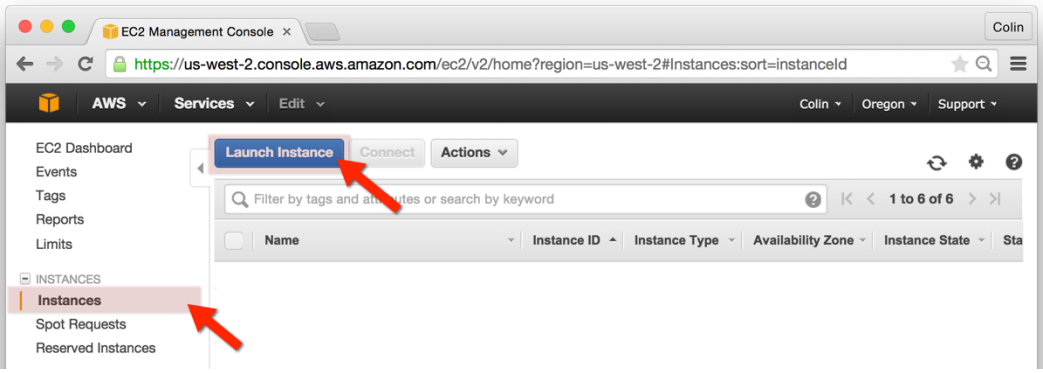


Create your Test Instance:

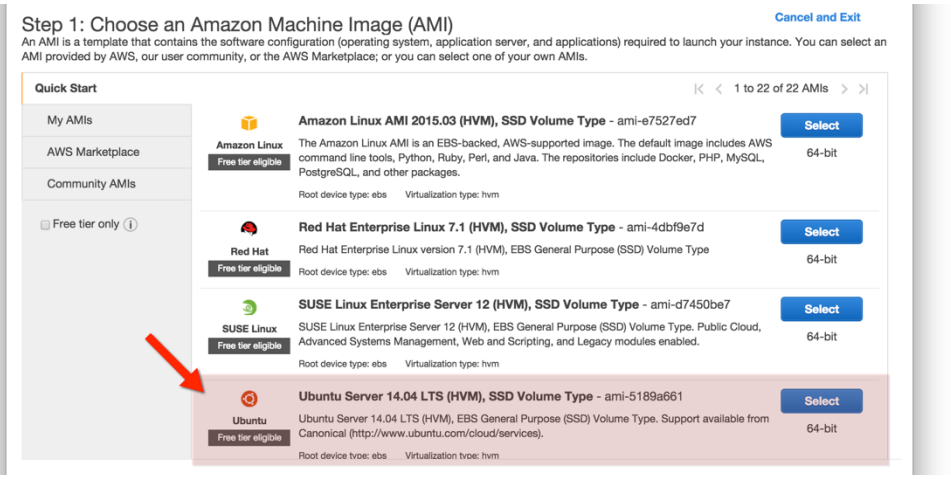
The procedure will below will create your “Test” EC2 instance.

1. Go to the AWS EC2 Console and select “Instances” from the left-hand navigation bar.

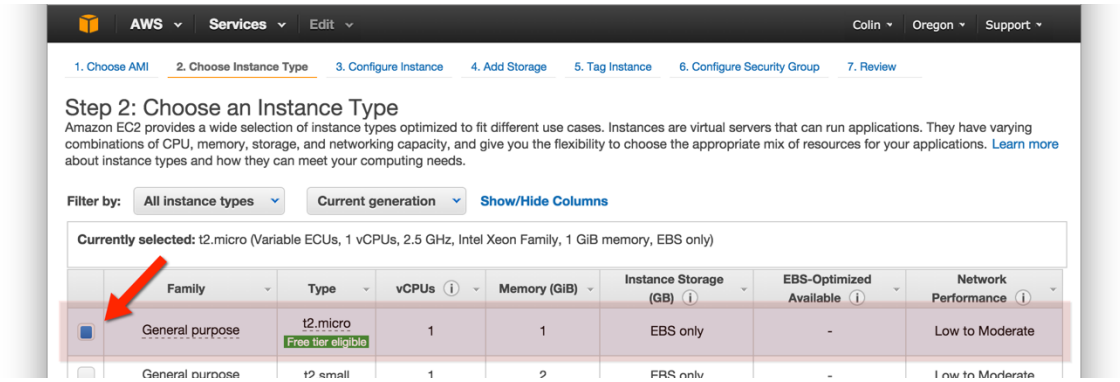
2. Select “Launch Instance”



3. Choose AMI:
    a. Choose the Ubuntu Server 14.04 LTS (HVM), SSD Volume Type AMI (ami-5189a661)
    b. Press “Select”



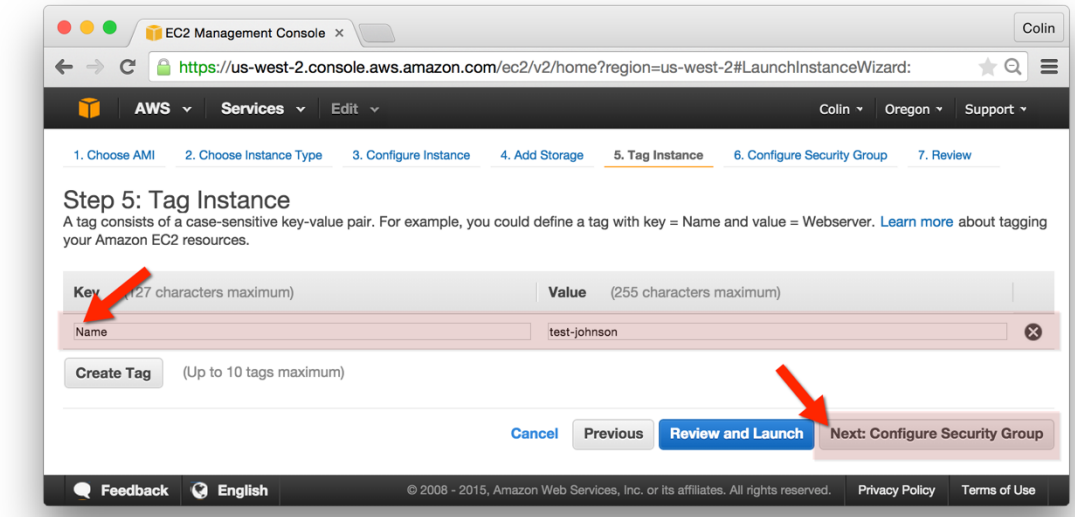
4. Choose Instance Type:
    a. Instance Type: t2.micro
    b. Press “Next: Configure Instance Details”
    c.



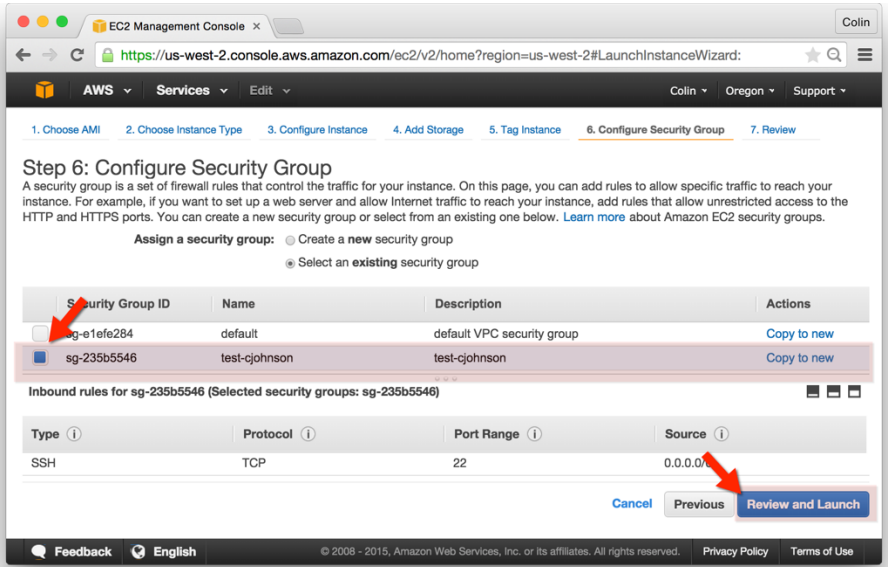
5. Configure Instance Details:
    a. Number of instances: 1
    b. Network: vpc-f38d3796 (“Default” VPC)
    c. Subnet: No preference
    d. Auto-assign Public IP: Enable
    e. IAM role: None
    f. Click “Next: Add Storage”

6. Add Storage
    a. Click “Next: Tag Instance”

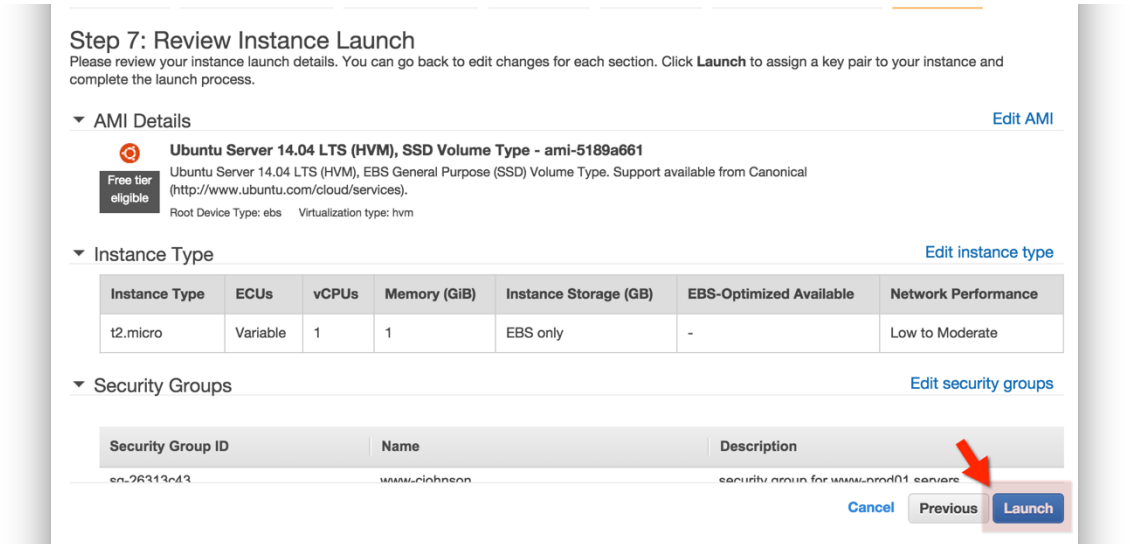
7. Tag Instance:
    a. Key=Name, Value=test-cjohnson



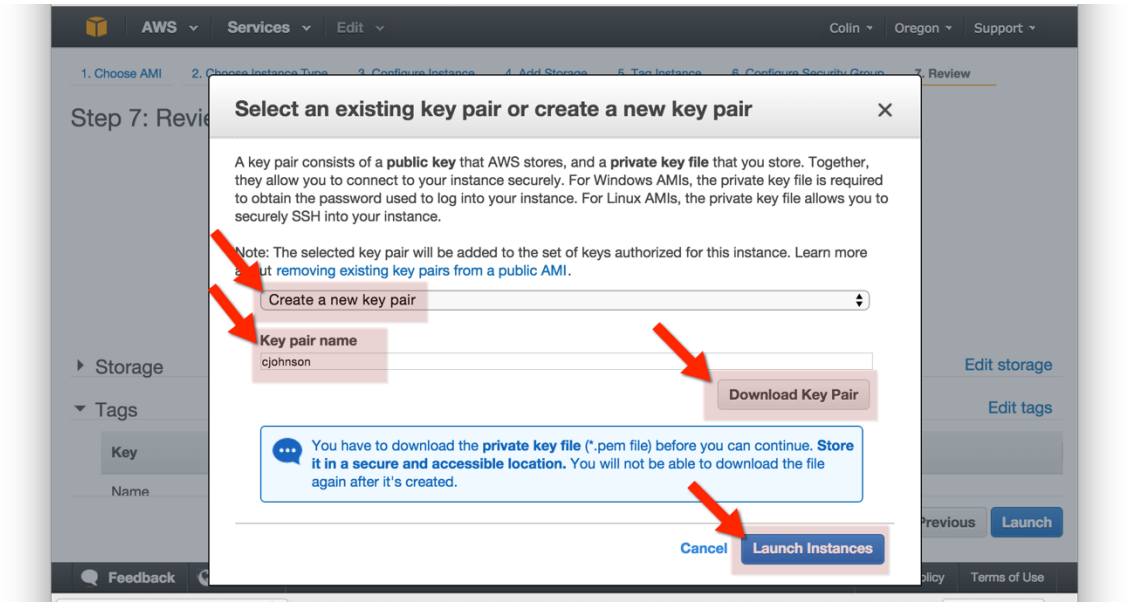
8. Configure Security Group:
    a. Choose the "test-yourname" security group created previously. *This will allow port 22 in on a your newly created "Test" instance.*
    b. Click "Review and Launch"



9. Review and Launch:
    a. Instance Type: t2.micro
    b. Security Group: test-yourname
    c. Instance Details:
        i. Network: vpc-f38d3796
        ii. Assign Public IP: Enable
    d. Tags
        i. Name: test-yourname
    e. Click “Launch”

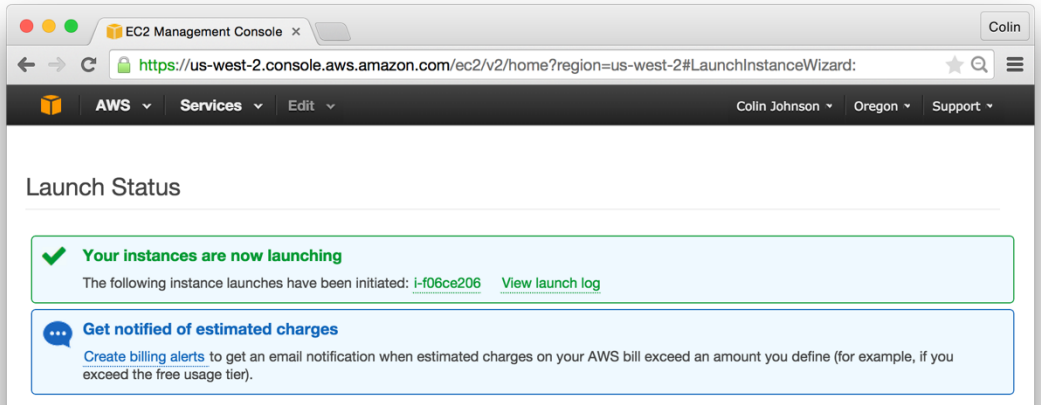


10. Keypair:
    a. Create a new key pair
    b. Key pair name: cjohnson
    c. “Click Download Keypair” – save this keypair for future labs!
    d. Click “Launch Instances”



Launch!

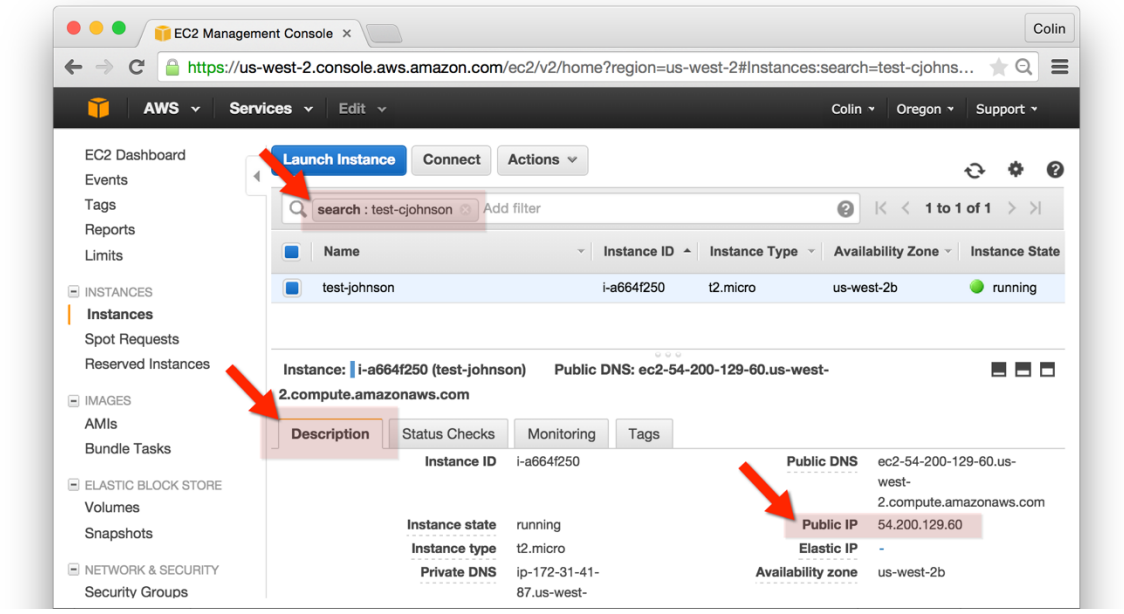
You'll see a note that "Your instances are now launching." You've created a Security Group and built an AWS EC2 Server!



Login!

You'll want to ssh into your new instance. In order to do this, you'll need the instance's Public IP address and your Keypair.

1. Go to the AWS EC2 Console and select "Instances" from the left-hand navigation bar.
2. Use the "Filter" and filter by "test-yourname" to locate your instance.
3. In the Instance Description tab, locate the Public IP and copy it.



4. With this information, you should be able to login as follows:

```
a. ssh -i ~/path/to/keyfile.pem ubuntu@x.y.z.z
```

Delete!

After you've logged in, please delete both the instance and security group. The other labs will not need to make use of the "Test" instance that we just created.

Exercise 2 – Create Security Groups

Instructions

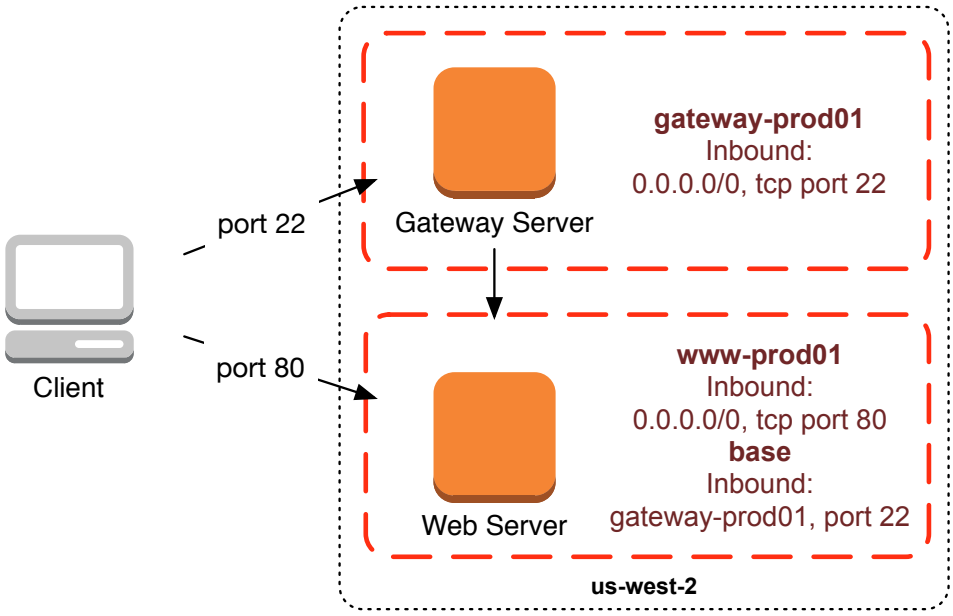
1. Create Three Groups:
   - gateway-yourname
   - Inbound, tcp port 22 from 0.0.0.0/0
   - Base
   - Inbound, tcp port 22 from gateway-yourname
   - www-yourname
   - Inbound, tcp port 80 from 0.0.0.0/0

Exercise 3 – Security Group Diagram

Instructions

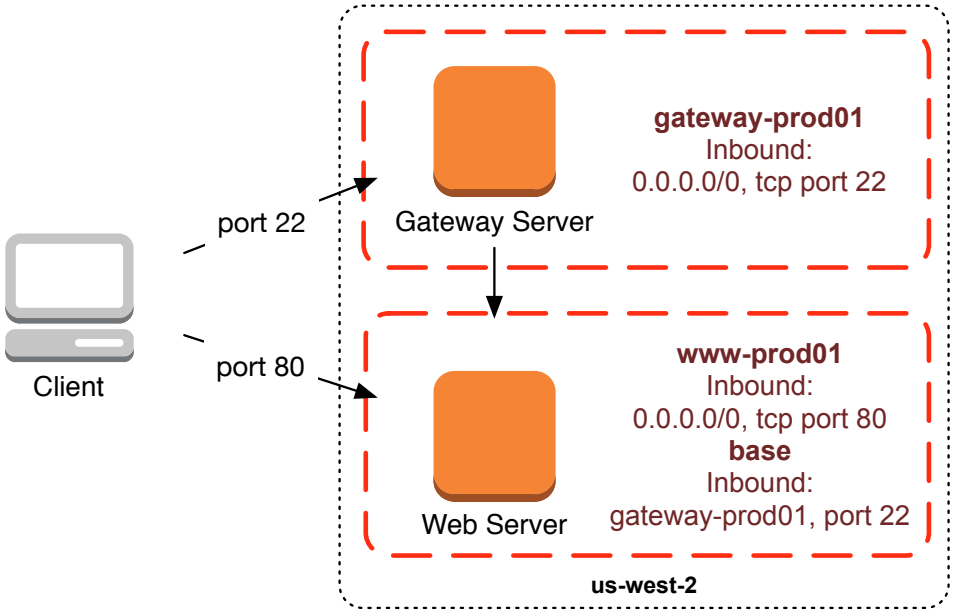
The project will accomplish the following objectives:

1. Utilize both IP addressing and Security Group Reference when creating a Security Group
2. Reference another Security Group within a Security Group's Rules



Exercise 4 – Build Gateway and www Servers

Instructions



Exercise 4 – Create Gateway Instance

Instructions

- Create EC2 Instance: “gateway-yourname”
  - Region: us-west-2
  - Select AMI: ami-5189a661 (Ubuntu)
  - Instance Type: t2.micro
  - Auto-assign Public IP: Enable
  - Tenancy: Shared
  - Security Group(s): gateway-yourname
  - Tag: Name=gateway-yourname

Exercise 5 – Create www Instance

Instructions

* Create EC2 Instance: “www-yourname”
  * Region: us-west-2
  * Select AMI: ami-5189a661 (Ubuntu)
  * Instance Type: t2.micro
  * Auto-assign Public IP: Enable
  * Tenancy: Shared
  * Security Group(s): web-yourname, base
  * Tag: Name=web-yourname

ssh into Gateway Instance

* Login to ssh Instance:
  * ssh –i ~/path/to/keypair.pem ubuntu@54.69.231.28
* Copy Keypair to Instance (for presentation only)
  * scp -i ~/path/to/keypair.pem ~/path/to/keypair.pem ubuntu@54.69.231.28:/home/ubuntu/
* Or use SSH Agent Forwarding
  * Described in command line exercises

ssh into www Instance

* Login to gateway Instance:
  * ssh –i ~/path/to/keypair.pem ubuntu@54.69.231.28
* Login to www Instance, using Internal IP (from gateway):
  * ssh –i ~/path/to/keypair.pem ubuntu@172.30.0.80
* Install webserver!
  * sudo apt-get -y install apache2

Important:

- We use an Internal IP Address, otherwise the Security Group would require that we allow in the Public IP Address of the gateway Server
- We use a Public IP Address for the www instance to allow access to apt Repository

Exercise 6 – Build an EC2 Instance using User Data

Instructions

- Create EC2 Instance: “wwwauto-yourname”
- Region: us-west-2
- Select AMI: ami-5189a661 (Ubuntu)
- Instance Type: t2.micro
- Auto-assign Public IP: Enable
- Tenancy: Shared
- Security Group(s): www-yourname, base
- Tag: Name=wwwauto-yourname
- Advanced Details -> User data
    - Paste Contents from “ec2-user-data.sh” File

Exercise 7 – Create, Snapshot, and Resize EBS

Instructions

1. Create an EBS Volume (20 GB in size).
2. Attach to the www-youname Instance.
3. Format the newly created EBS Volume.
4. Add data to the new EBS Volume.
5. Snapshot the new EBS volume.
6. Restore the EBS volume (60 GB in size, 1800 IOPS).

Exercise 8 – Auto Scaling

Overview

We will be building a fleet of Auto Scaling web servers serving a simple web page. In order to build our fleet of web servers we will be creating the following resources:

- Two Security Groups
  - An ELB Security Group
  - A web server / Backend Security Group
- A Load Balancer
- An EC2 Keypair
- A Launch Configuration
- An Auto Scaling Group

Instructions

Create Security Groups:

1. Go to the AWS EC2 Console and select "Security Groups" from the left-hand navigation bar.
2. Click "Create Security Group"
    a. Security group name: wwwelb-asg-yourname
    b. Description: wwwelb-asg-yourname
    c. VPC: use default VPC
    d. Security group rules:
        i. Inbound:
            1. Type: HTTP
            2. Protocol: TCP
            3. Port Range: 80
            4. Source: Anywhere / 0.0.0.0/0
        ii. Outbound:
            1. *If you wish to test: this rule may be able to be modified to allow only port 80 to the "www-asg" Security Group – the ELBs should not need to **initiate** outbound traffic*
    e. Security group name: www-asg-yourname
    f. Description: www-asg-yourname
    g. VPC: use default VPC
    h. Security group rules:
        i. Inbound:
            1. Rule 1:
                a. Type: HTTP
                b. Protocol: TCP
                c. Port Range: 80
                d. Source: wwwelb-asg-yourname

2. Rule 2:
    a. Type: SSH
    b. Protocol: TCP
    c. Port Range: 22
    d. Source: Anywhere / 0.0.0.0/0

ii. Outbound:
    1. *If you wish to test: this rule should be able to modified to port 80 to world – port 80 required for apt-get to complete install of Apache.*

Create Load Balancer:

1. Go to the AWS EC2 Console and select "Load Balancer" from the left-hand navigation bar.
2. Click "Create Load Blancer":
    a. Load Balancer name: wwwelb-asg-yourname
    b. Create LB Inside: My Default VPC
    c. Create an internal load balancer: unchecked
    d. Enable advanced VPC configuration: checked
    e. Listener Configuration:
        i. Load Balancer Protocol: HTTP
        ii. Load Balancer Port: 80
        iii. Instance Protocol: HTTP
        iv. Instance Port: 80
    f. Selected Subnets:
        i. Choose all available subnets.
        ii. Click "Next: Assign Security Groups"
    g. Assign Security Groups:
        i. Select the "wwwelb-asg-yourname" Security Group
        ii. Click "Next: Configure Security Settings"
    h. Configure Configure Settings:
        i. You'll be shown an "improve load balancer security screen"
        ii. Click "Next: Configure Health Check"
    i. Configure Health Check:
    Leave as default.
        i. Ping Protocol: HTTP
        ii. Ping Port: 80
        iii. Ping Path: /index.html
        iv. Response Timeout: 5 seconds
        v. Health Check Interval: 30 seconds
        vi. Unhealthy Threshold: 2
        vii. Healthy Threshold: 10
        viii. Click "Next: Add Instances"

j. Add EC2 Instances:
    i. Do not add any instances at this time.
    ii. Availability Zone Distribution:
        1. Enable Cross-Zone Load Balancing: checked
        2. Enable Connection Draining: checked
    iii. Click "Next: Add Tags"
        1. Key: Name
        2. Value: yourname
        3. Click "Create Tag"
        4. Click "Review and Create"
    iv. Review:
        1. Click "Create"

Create an EC2 Keypair (if necessary):

1. Go to the AWS EC2 Console and select "Keypairs" from the left-hand navigation bar.
2. Click "Create Key Pair"
    a. Key pair name: yourname
    b. Click "Create"
3. The keypair will be automatically downloaded.

Create a Launch Configuration:

1. Go to the AWS EC2 Console and select "Launch Configurations" from the left-hand navigation bar.
2. Click "Create Launch Configuration"
    a. Choose AMI:
        i. AMI: choose AMI ID: ami-5189a661 and press "Select"
    b. Choose Instance Type:
        i. Instance Type: t2.micro (*m3.medium if you wish to use spot pricing*)
        ii. Click "Next: Configure details"
    c. Configure Details:
        i. Name: www-asg-yourname-$date
        ii. Purchase option: *feel free to check "Request Spot Instances" if you wish. You should bid slightly more than the current spot pricing. To view Spot Pricing history:*
            1. *go to the AWS EC2 Console and select "Spot Requests"*
            2. *click the "Pricing History" button*
        iii. IAM role: None (*the IAM role could potentially allow an application running on this instance to access AWS resources*)
        iv. Monitoring:
            1. Enable CloudWatch detailed monitoring: unchecked
        v. Advanced Details:
            1. User data:
                a. Paste in the contents of the file: simple-web-server.sh

2. IP Address Type: Only assign a public IP address to instances launched in the default VPC and subnet. (default)

vi. Click "Next: Add Storage"

d. Add Storage:
   i. Click "Next: Configure Security Group"
e. Configure Security Group:
   i. Assign a Security Group:
      1. Select an existing security group
         a. Choose the "www-asg-yourname" group that you had previously created
   ii. Click "Review"
f. Review:
   i. Confirm all attributes set correctly.
   ii. Click "Create launch configuration"
g. Select Keypair:
   i. Choose a keypair that you had created previously.

Create an Auto Scaling Group:

1. Go to the AWS EC2 Console and select "Auto Scaling Groups" from the left-hand navigation bar.
2. Click "Create Auto Scaling group"
3. The "Create Auto Scaling Group" template will be presented.
   a. Select "Create an Auto Scaling group from an existing launch configuration"
      i. Select the Launch Configuration" you had created previously.
      ii. Press "Next Step"
   b. Conifgure Auto Scaling group details:
      i. Launch Configuration: <confirm this is the launch configuration you had created previously>
      ii. Group Name: www-asg-yourname
      iii. Group size: 1
      iv. Network: select the "default" VPC
      v. Subnet: choose all Subnets
      vi. Advanced Details:
         1. Load Balancing:
            a. Receive traffic from Elastic Load Balancer(s): checked
            b. Choose the Elastic Load Balancer you had created previously (wwwelb-asg-youname)
         2. Health Check Type: EC2
         3. Health Check Grace Period: 300 seconds
      vii. Click "Configure scaling policies"
         1. Keep this group at its initial size: checked
         2. Click "Next: Configure Notifications"
      viii. Configure Notifications:
         1. Click "Next: Configure Tags"

ix. Configure Tags:
    1. Key=Name
    2. Value=yourname
    3. Tag New Instances = checked
    4. Click "Review"

x. Review
    1. Review Settings
    2. Click "Create Auto Scaling group"

Confirm Correct Operation:

Confirm Auto Scaling has launched an EC2 Instance:

1. Go to the AWS EC2 Console and select "Auto Scaling Groups"
2. Select your Auto Scaling Group (name: www-asg-yourname)
3. Click on the "Scaling History" tab:
    a. Confirm that an EC2 instance has been launched.
4. Click the "Instances" tab:
    a. Configure that an EC2 Instance is listed here.

Confirm that the instance has been attached to the ELB:

1. Go to the AWS EC2 Console and select "Load Balancers"
2. Select your Elastic Load Blancer (name: wwwelb-asg-yourname)
3. Click on the "Description" tab:
    a. Status should read: "1 of 1 instances in service"

Confirm that the Instance is serving traffic:

1. Go to the AWS EC2 Console and select "Load Balancers"
2. Select your Elastic Load Blancer (name: wwwelb-asg-yourname)
3. Click on the "Description" tab:
    a. Copy the "DNS Name" and paste this into a web browser.
    b. The "Server Information" page should be returned.

Scale!!!

At this point, we have built an elastic and scalable infrastructure – our infrastructure and "application" may be simple, but we can scale servers in and out of service easily.

1. Go to the AWS EC2 Console and select "Auto Scaling Groups"
2. Select your Auto Scaling Group (name: www-asg-yourname)
3. Click on the "Details" tab and click "Edit"
    a. Change the "Desired" and "Max" capacity to 3.
    b. Click "Save"

Watch Scaling Happen!

1. Go to the AWS EC2 Console and select "Load Balancers"
2. Select your Elastic Load Blancer (name: wwwelb-asg-yourname)
3. Click on the "Description" tab:
    a. Status should read: "1 of 3 instances in service" (*if the instances are still coming into service*) or "3 of 3 instances in service"

Confirm all Instances in Service:

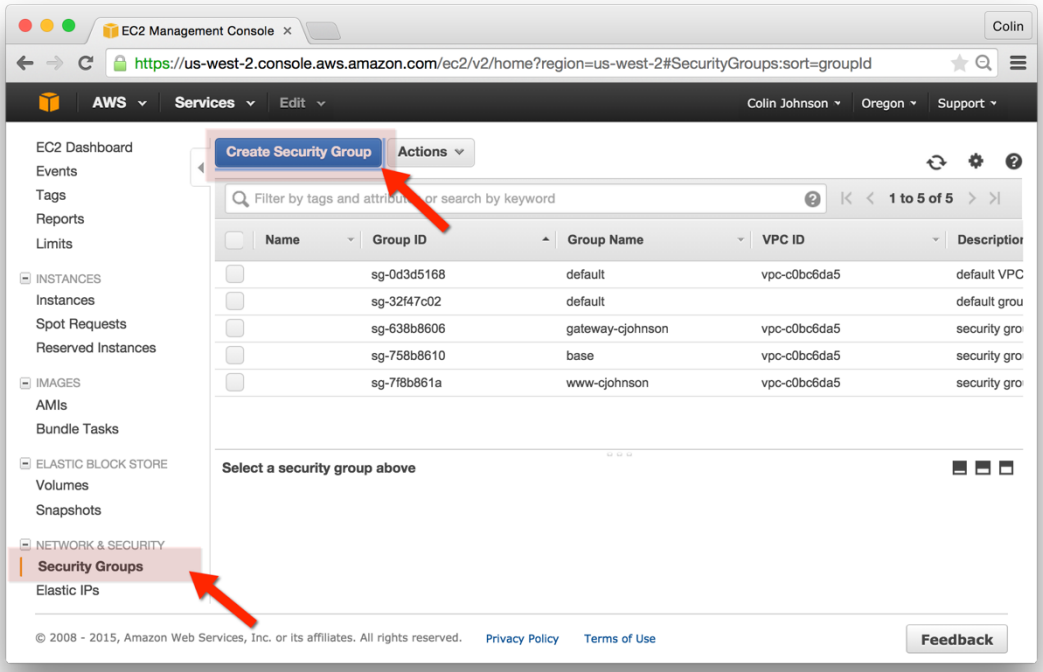
1. Go to the AWS EC2 Console and select “Load Balancers”
2. Select your Elastic Load Blancer (name: wwwelb-asg-yourname)
3. Click on the “Description” tab:
    a. Copy the “DNS Name” and paste this into a web browser.
    b. The “Server Information” page should be returned – the values should change as requests are directed at different instances.

Exercise 9 – Elastic Load Balancing

Instructions

Go to the AWS EC2 Console:

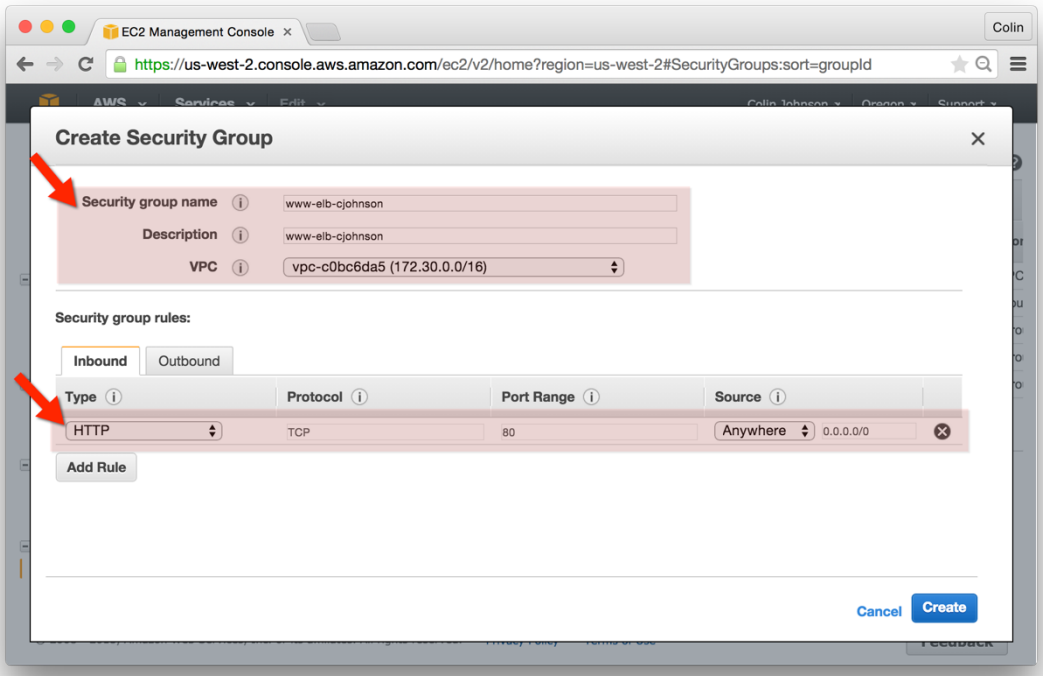
In the AWS Console, select the "Oregon" Region, select "EC2" and select "Security Groups" from the left-hand navigation bar, then click "Create Security Group":



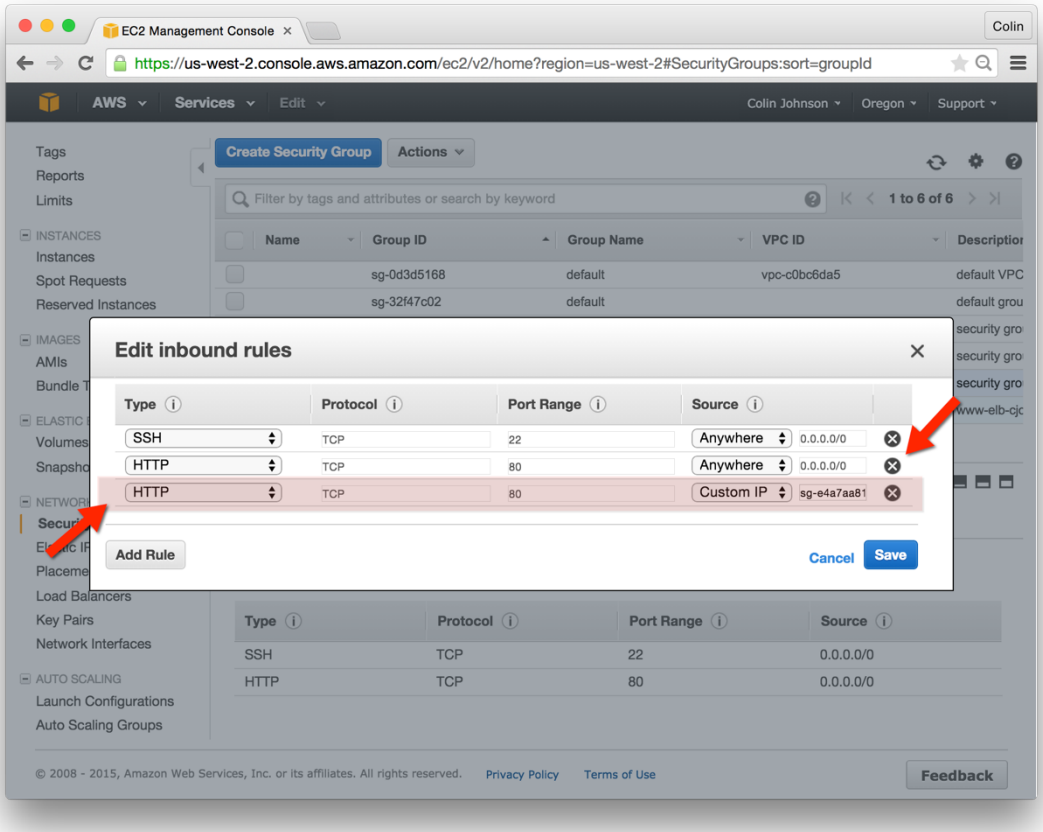
Create an www ELB Security Group:

Create a new Security Group, as follows:

- Security Group Name: www-elb-yourname
- Description: www-elb-yourname
- Inbound Rule:
  - Type: HTTP
  - Protocol: TCP
  - Port Range: 80
  - Source: Anywhere



Modify the Existing www-yourname Server Security Group:

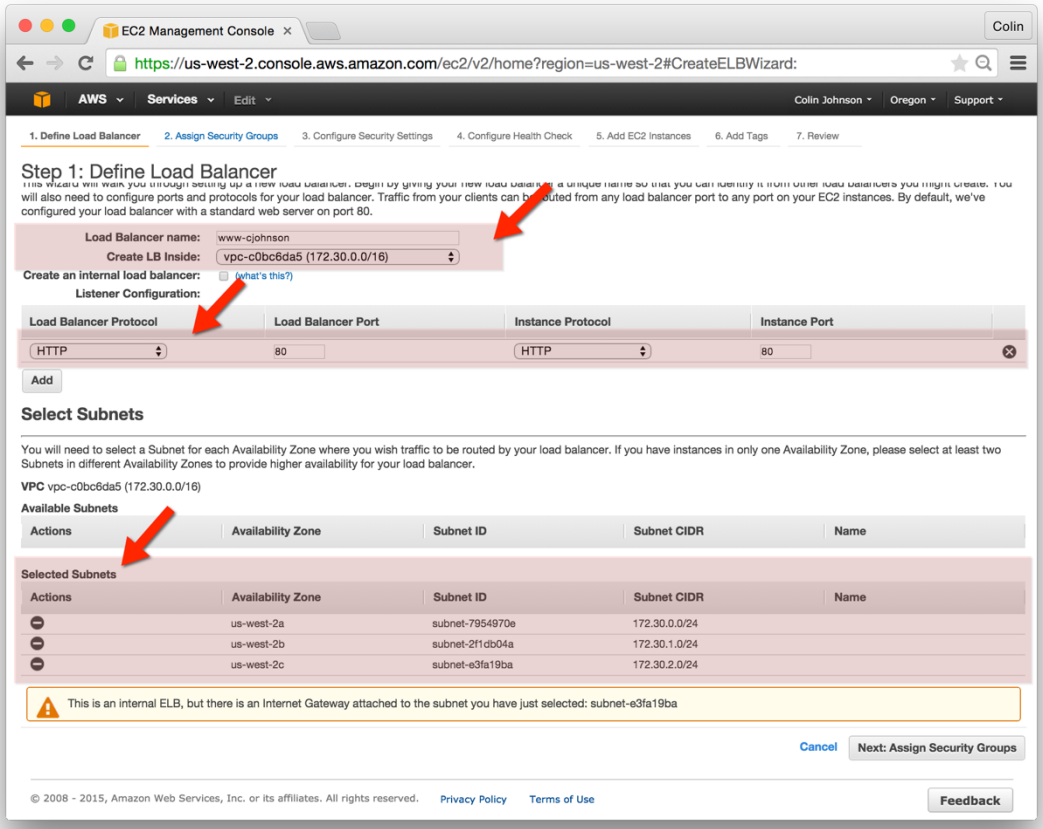


- Select the www-yourname Security Group
- Click the “Inbound” tab and select “Edit”
- Add a Rule, as Follows:
  - Type: HTTP
  - Protocol: TCP
  - Port: 80
  - Source: www-elb-yourname
- Remove the existing HTTP/Anywhere rule

Create a www Elastic Load Balancer:

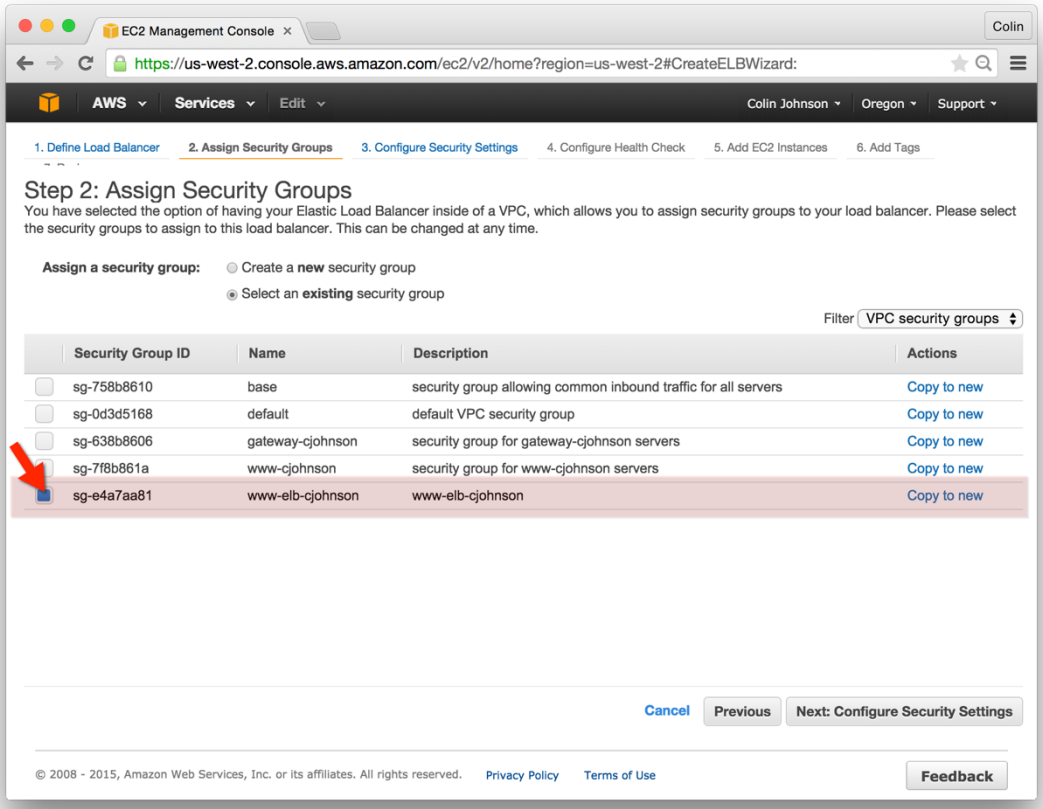
Configure Your Load Balancer:

- Load Balancer name: www-yourname
- Create LB Inside: choose the "default" VPC
- Create an internal load balancer: not selected
- Listener Configuration, setup as follows:
    a. Load Balancer Protocol: HTTP
    b. Load Balancer Port: 80
    c. Instance Protocol: HTTP
    d. Instance Port: 80
- Select Subnets:
    a. Select all Subnets



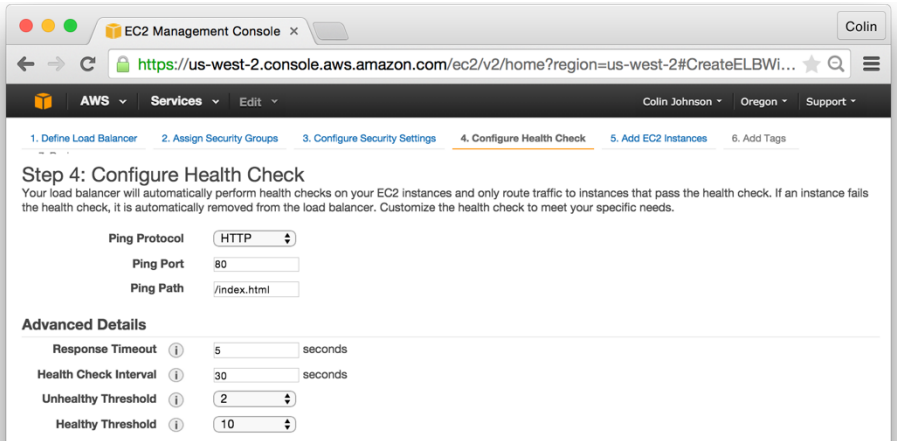
Assign Security Groups:

- Choose the www-elb-yourname Security Group



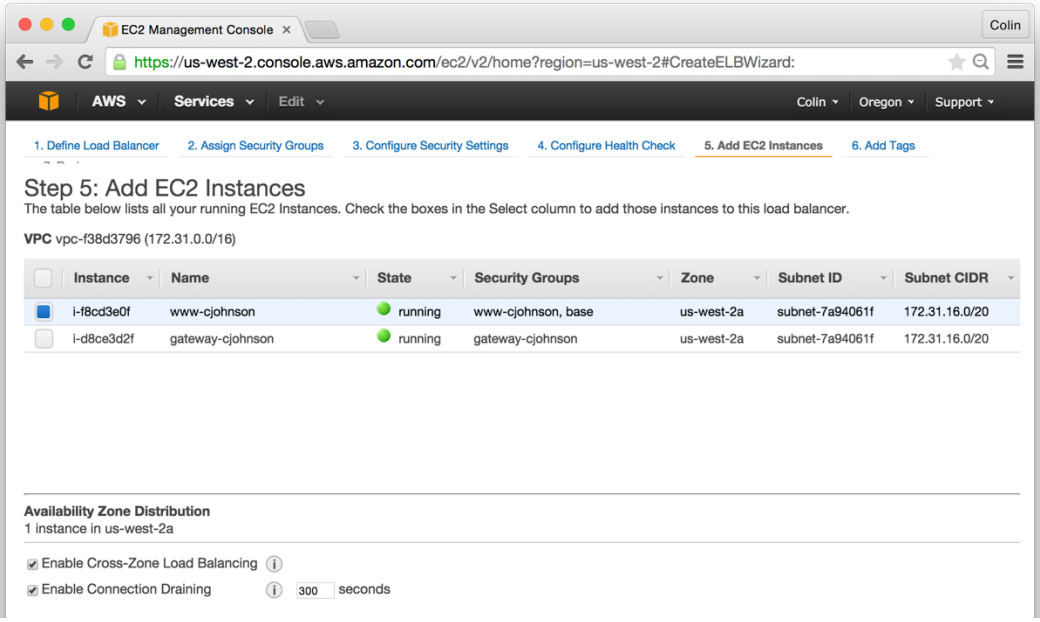
Configure Health Check:

- Ping Protocol: HTTP
- Ping Port: 80
- Ping Path: /index.html
- Advanced Details: leave as is.



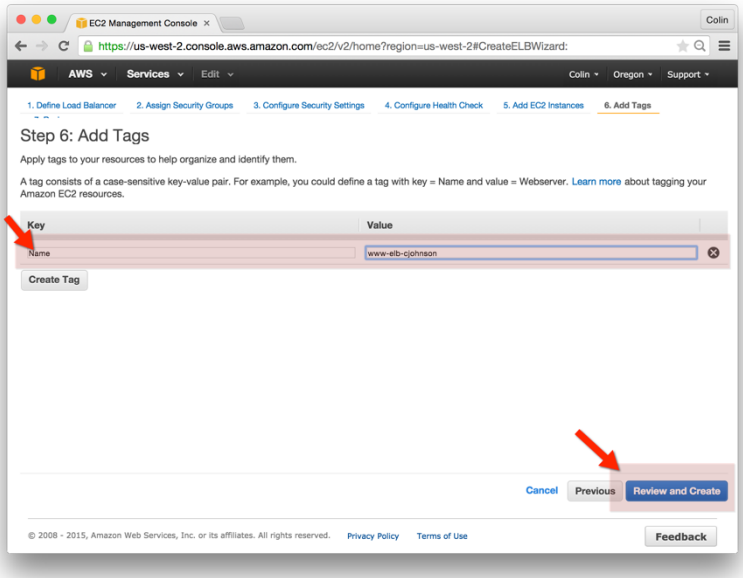
Add EC2 Instances

- Add EC2 Instances:
  - Add the www-youname Instance
- Availability Zone Distribution:
  - Enable Cross-Zone Load Balancing: checked
  - Enable Connection Draining: checked
- Click "Next: Add Tags"

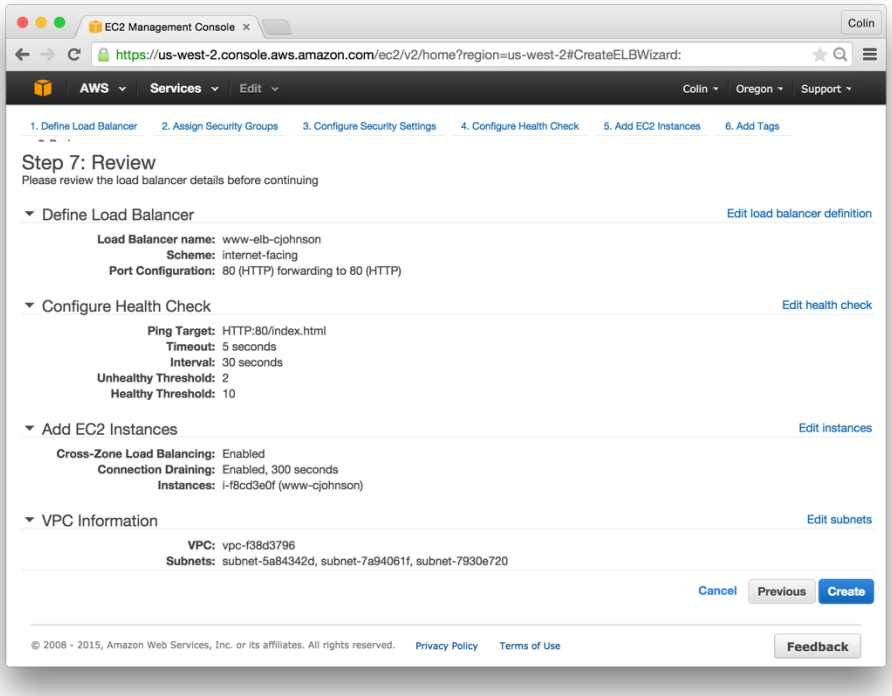


Add Tags:

- Create a Tag as Follows: “Name=www-elb-yourname”
- Click “Review and Create”

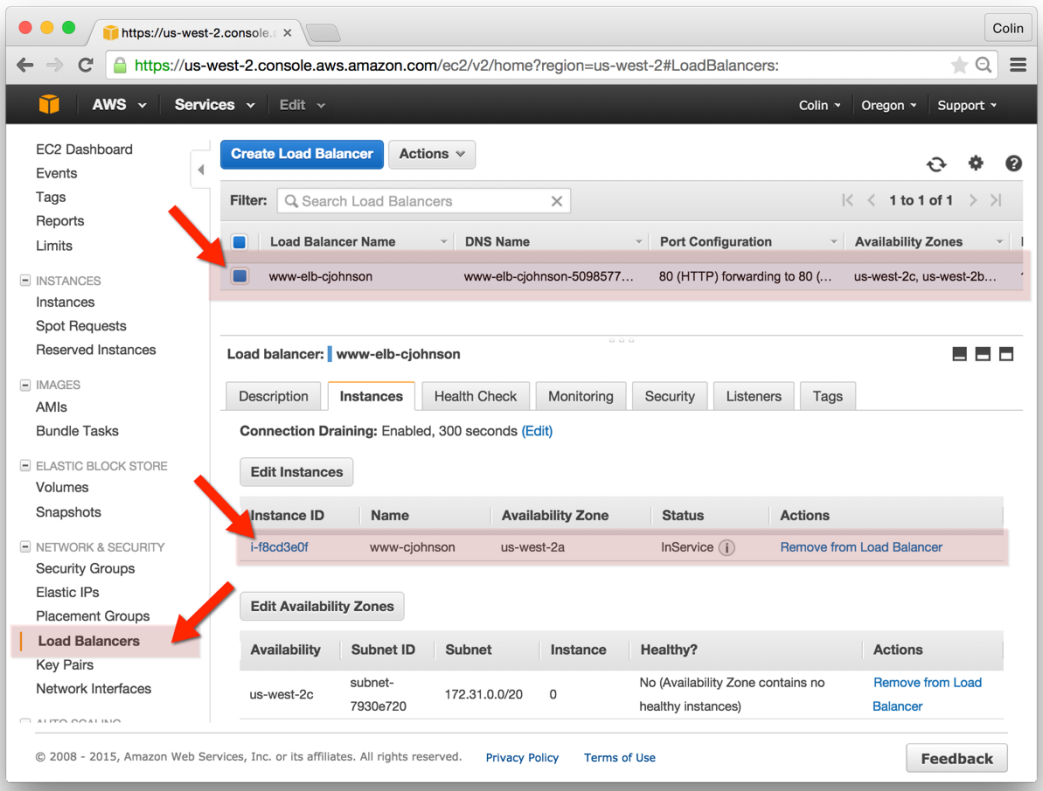


Review:



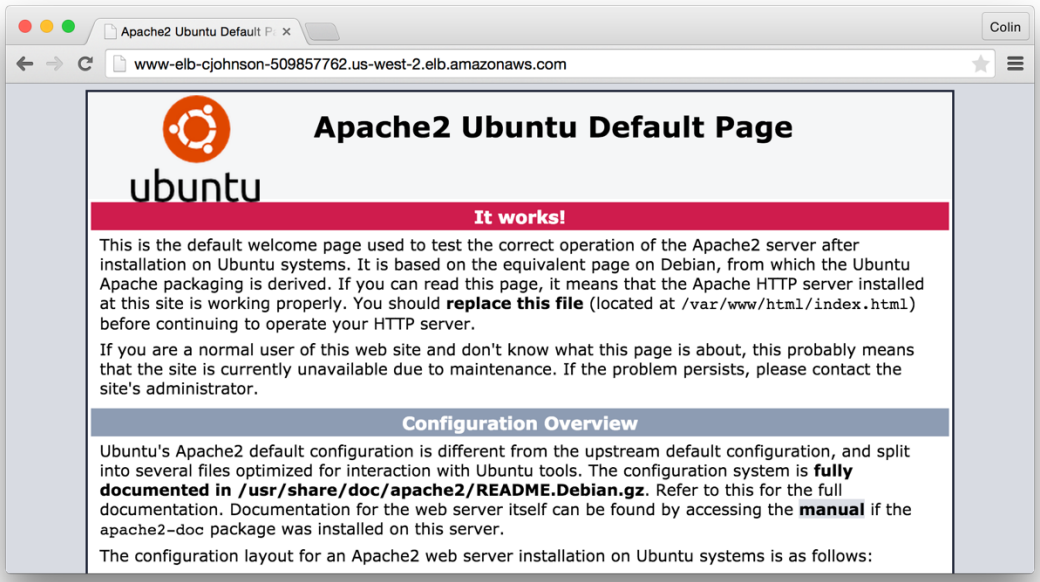
Confirm Load Balancer Built and Instance Placed in Service!

After clicking "Complete", return the the "Load Balancer" service page in the AWS EC2 Console and confirm that your www-yourname EC2 Instance is in service. An example is below:



Lastly, request a page through the Elastic Load Balancer:

- Select your www-elb-yourname Load Balancer from the EC2 AWS Console
- Click the "Description" tab and select the DNS Name
- Paste this DNS name into your web browser – this request should be directed to your back-end instance.

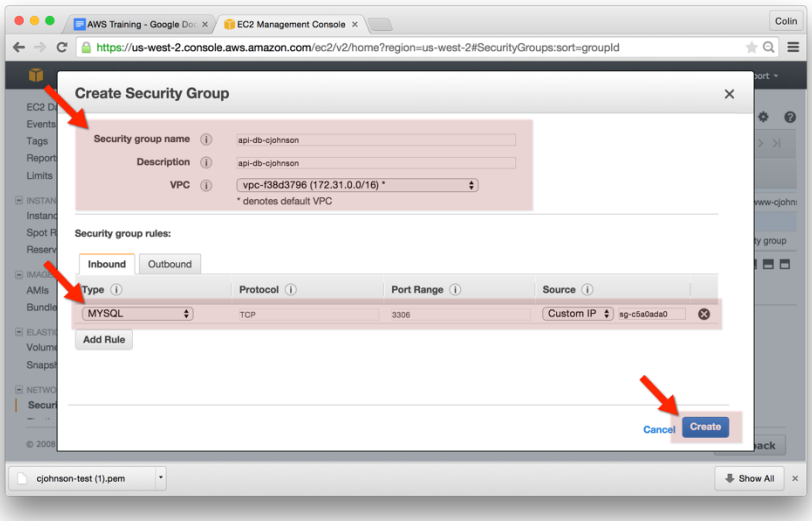


Exercise 10 – Relational Database Service (RDS)

Instructions

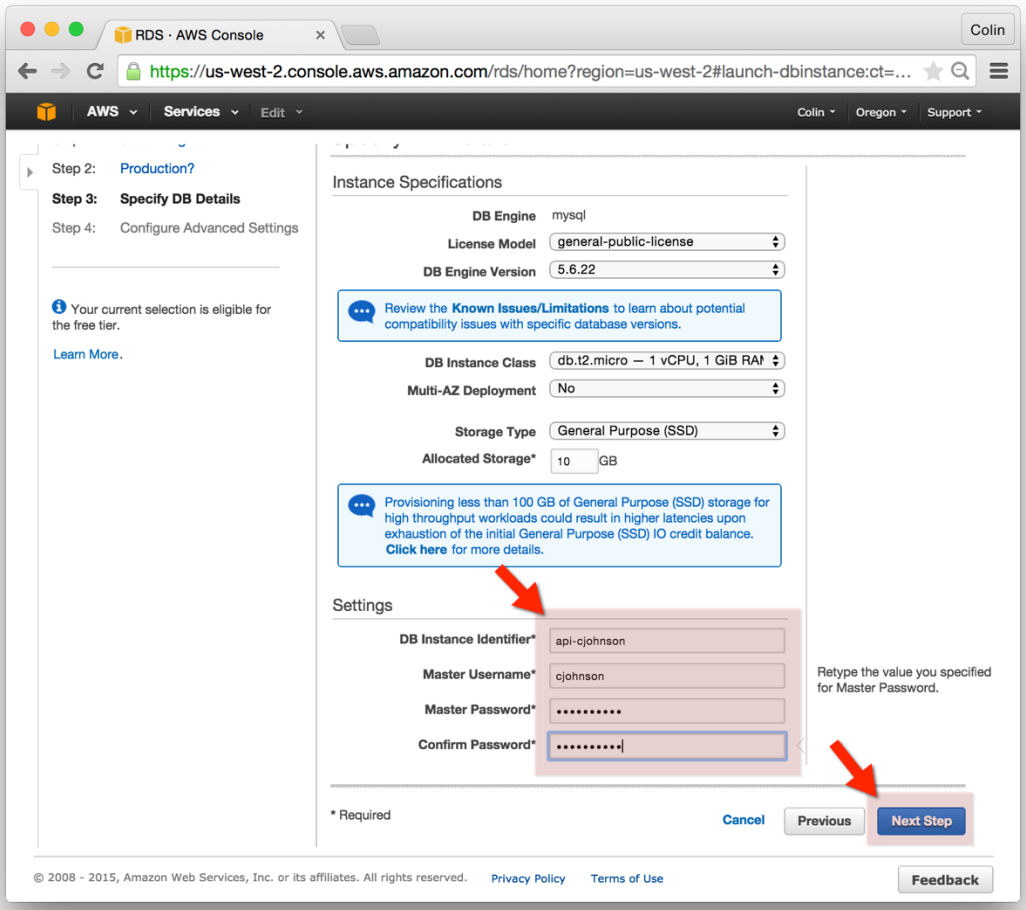
Create a Security Group for the RDS Database:

1. Go to the AWS EC2 Console and select "Security Groups"
2. Click "Create Security Group"
    a. Security group name: api-db-youname
    b. Description: api-db-yourname
    c. VPC: vpc-f38d3796
    d. Inbound Rules:
        i. Type: MYSQL
        ii. Protocol: TCP
        iii. Port Range: 3306
        iv. Source: gateway-yourname



Create an RDS Database:

1. Go to the AWS RDS Console, select "Instances" from the left-hand navigation and choose "Launch Instance"
2. Select Engine:
   a. Choose "MySQL" and press "Select"
3. Production?:
   a. Choose: "No, this instance is intended for use outside of production or under the RDS Free Usage Tier"
   b. Click "Next Step"
4. Specify DB Details:
   a. DB Engine: mysql
   b. License Model: general-license-model (*this is the only option for mysql databases*)
   c. DB Engine Version: 5.6.22
   d. DB Instance Class: db.t2.micro
   e. Multi AZ Deployment: No
   f. Storage Type: General Purpose (SSD)
   g. Allocated Storage: 10 GB
   h. DB Instance Identifier: api-yourname
   i. Master Username: yourname
   j. Master Password: <your choice>
   k. Confirm Password: <repeat above>
5. Configure Advanced Settings:
   a. Network and Security:
      i. VPC: Default VPC (vpc-f38d3796)
      ii. Subnet Group: default
      iii. Publicly Accessible: No
      iv. Availability Zone: No Preference
      v. VPC Security Groups: api-db-yourname
   b. Database Options:
      i. Database Name: yourname
      ii. DB Parameter Group: default:mysql5.6
      iii. Option Group: default: mysql-5-6
      iv. Enable Encryption: No (not available on t2.micro instance types)
   c. Backup:
      i. Backup Retention: 7 days
      ii. Backup Window: No Preference (note that backups impact performance)
   d. Maintenance:
      i. Auto Minor Version Upgrade: Yes
      ii. Maintenance Window: No Preference
   e. Press "Launch DB Instance"



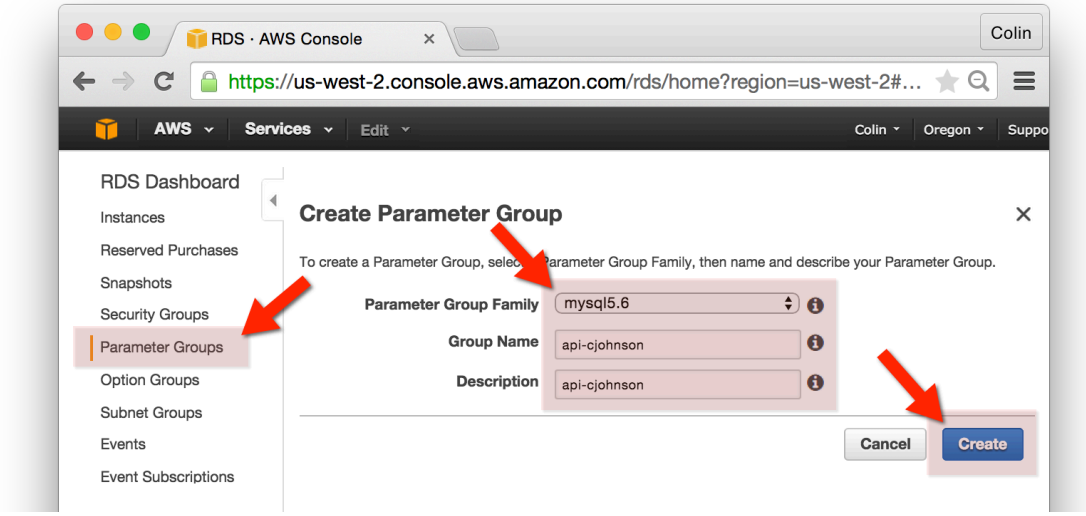
Connect to your RDS Database:

1. From the AWS RDS Console, locate your database and copy the "Endpoint"
2. Login to your gateway-yourname server.
3. Install the MySQL client by running the following command:
   a. `sudo apt-get -y install mysql-client-5.6`
4. Connect to your MySQL host as follows:
   a. `mysql -h api-cjohnson.cnn1tkhekekc.us-west-2.rds.amazonaws.com -u cjohnson -pmypassword`
   b. You'll be presented with the MySQL Command Prompt, run the show tables command and you'll notice that AWS has created a database for you:
   c. `show tables;`
   d. Disconnect from the RDS database.
   e. `exit;`

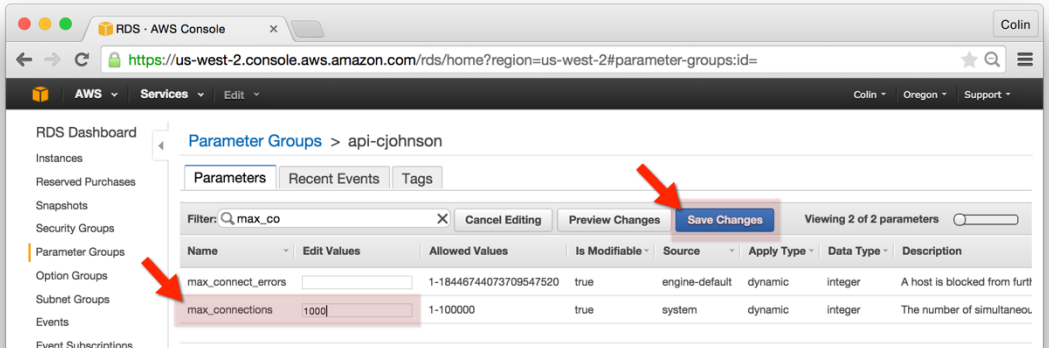
Create an RDS Parameter Group:

AWS Customers that utilize RDS often have the need to modify RDS parameters. In one particular case, the value of "max_connections" is set based on amount of RDS memory – and some customers may wish to change this value. To do so:

1. From the AWS RDS Console, select “Parameter Groups” and choose “Create Parameter Group”
2. Create a parameter group with the following options:
    a. Parameter Group Family: mysql5.6
    b. Group Name: api-yourname
    c. Description: api-yourname
    d. Click “Create”



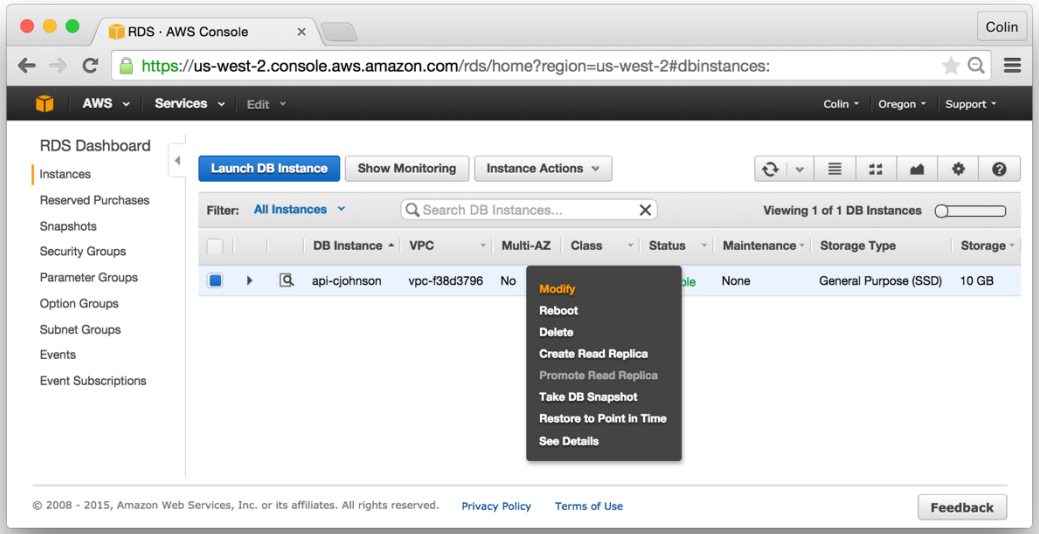
3. Select the api-yourname Parameter Group and click “Edit Parameters”
    a. In the filter field, enter “max_connections”
    b. Click the “Edit Parameters” button
    c. Enter the value 1000 in the “max_connections” Field and Click “Save Changes”



Modify RDS Database Settings:

We will be making database changes en masse – changing the instance type, storage and max_connections parameters. To make these changes:

1. Go to the AWS RDS Console, select "Instances", right-click on your database and click "Modify"

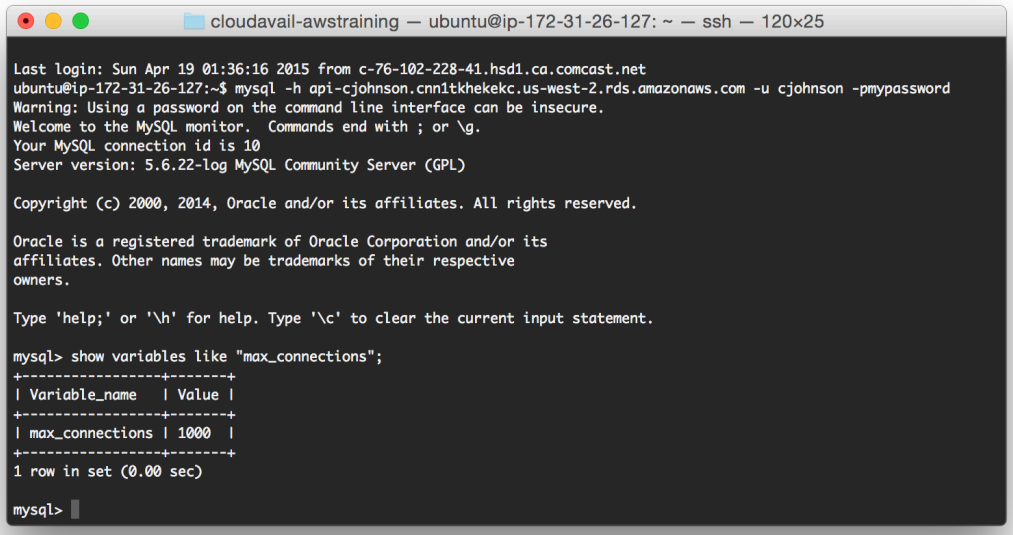


2. Change the following options:
    a. DB Instance Class: db.t2.micro
    b. Allocated Storage: 20
    c. DB Parameter Group: api-yourname
    d. Apply Immediately: checked
3. Click "Continue"
4. You'll be prompted to review the configuration changes.
5. Click "Modify DB Instance"

Confirm RDS Database Settings have Changed:

Confirmation of RDS database changes can typically be performed either through the RDS console or on the RDS instance itself. In many cases, I prefer confirming changes on the RDS instance itself. To confirm changes, do the following:

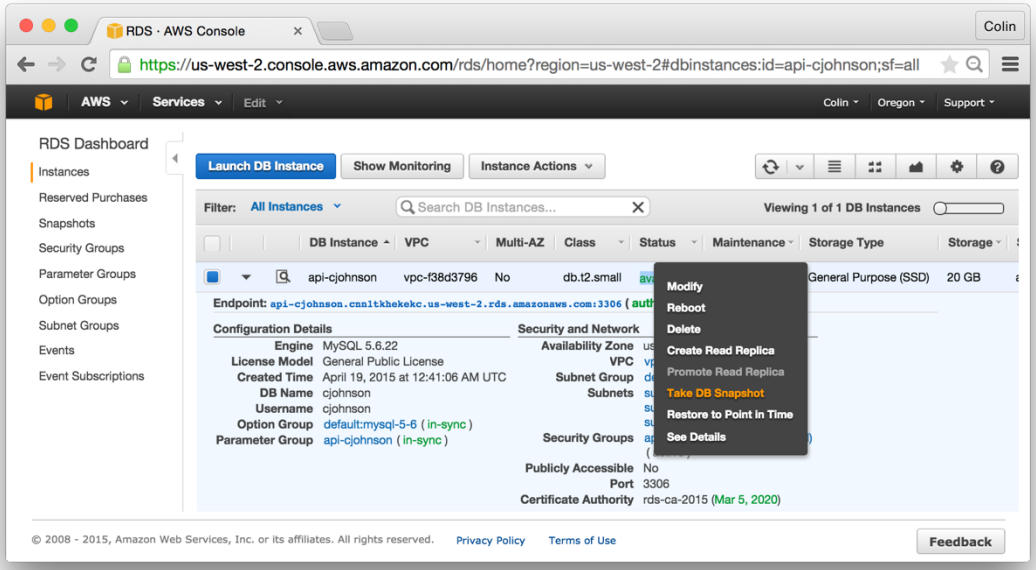
1. Login to your gateway-yourname server.
2. Connect to your MySQL host as follows:
    a. `mysql -h api-cjohnson.cnn1tkhekekc.us-west-2.rds.amazonaws.com -u cjohnson -pmypassword`
    b. You'll be presented with the MySQL Command Prompt, where you can verify that the max_connections variable has been changed:
    c. `show variables like "max_connections";`
    d. Disconnect from the RDS database.
    e. `exit;`



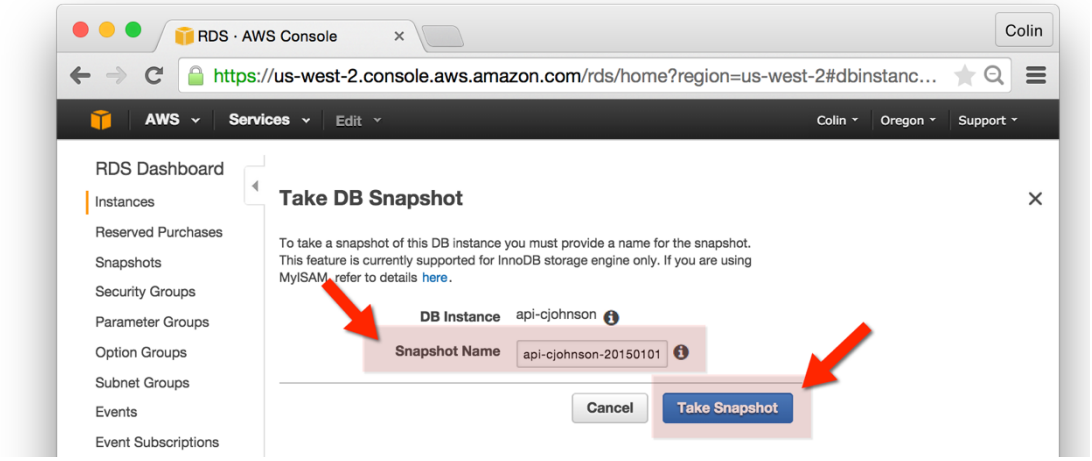
Create an RDS Snapshot:

Snapshots of an RDS Instance are a snapshot of the entire RDS instance at a point in time. Instructions for taking a snapshot are below:

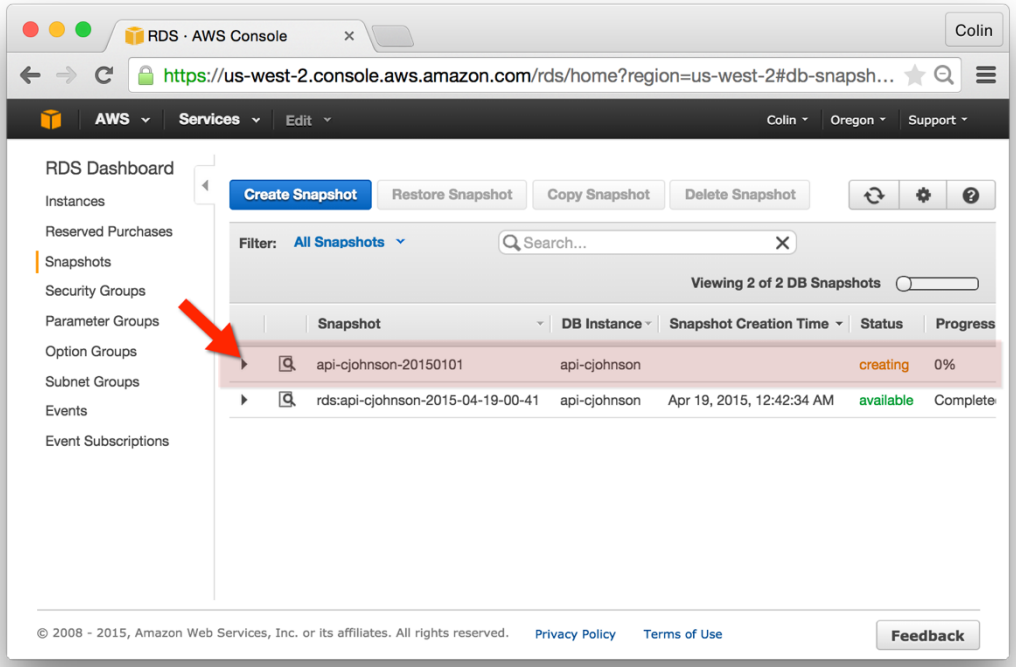
1. From the AWS RDS Console, select "Instances" and right-click on your database (your database should be named api-yourname)
2. From the drop-down menu, select "Take DB Snapshot"



3. Enter a Snapshot Name:
    a. api-yourname-$date
    b. Click "Take Snapshot"



4. Snapshot progress can be viewed in the RDS Dashboard



Exercise 11 – AWS S3

Instructions

S3 Cross Region Replication

- Have Instructions for Configuration
- Released on March 24, 2015
- Use Cross Region Replication to Automatically push Objects from one Region to Another
- Replication includes object ACLs, metadata and Reduced Redundancy Storage setting

S3 IAM Access

- Create an IAM user that:
  - has the ability to PUT/GET/DELETE objects in a given bucket
  - has no further permissions:
- Sample use case (1): ensuring an application can only access a particular S3 bucket (or S3 bucket / prefix)
- Sample use case (2): allowing users to upload or read from only a particular S3 bucket

Exercise 12 – Create Public Object

Instructions

Overview:

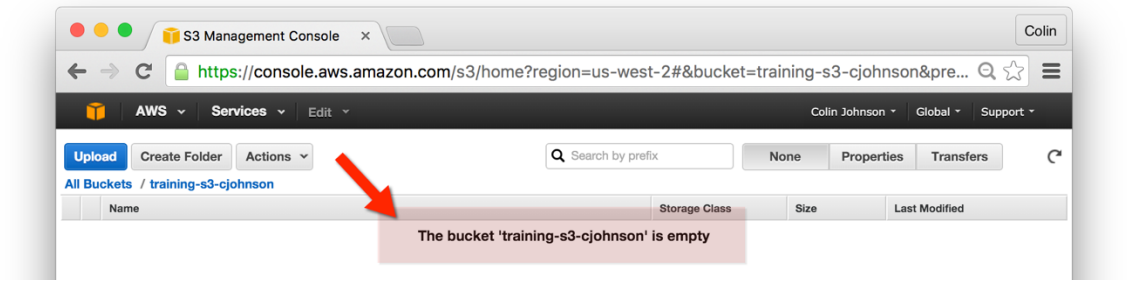
We'll be creating the following an S3 bucket, uploading an S3 object and making the uploaded object public.

Create an S3 Bucket in the us-west-2 (Oregon) Region:

1. Go to the AWS S3 Console
2. Click "Create Bucket"
    a. Bucket Name: training-s3-yourname
    b. Region: Oregon
    c. Click Create

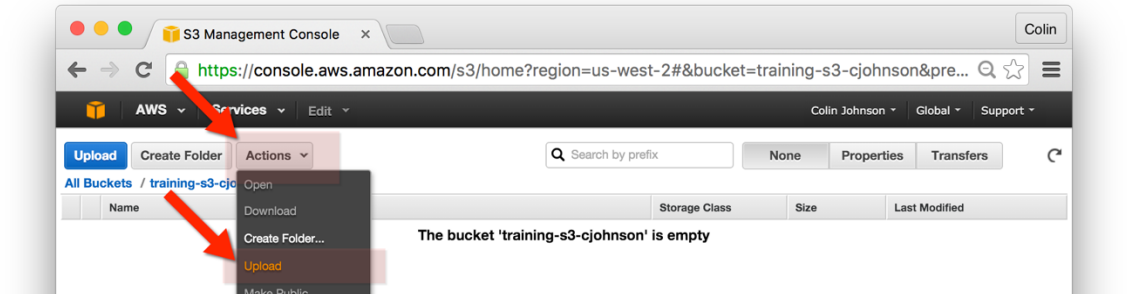
Select the Newly Created S3 Bucket:

1. Click on the S3 bucket you created in a previous step – you'll be presented with the bucket content screen (note that the bucket won't have any content).



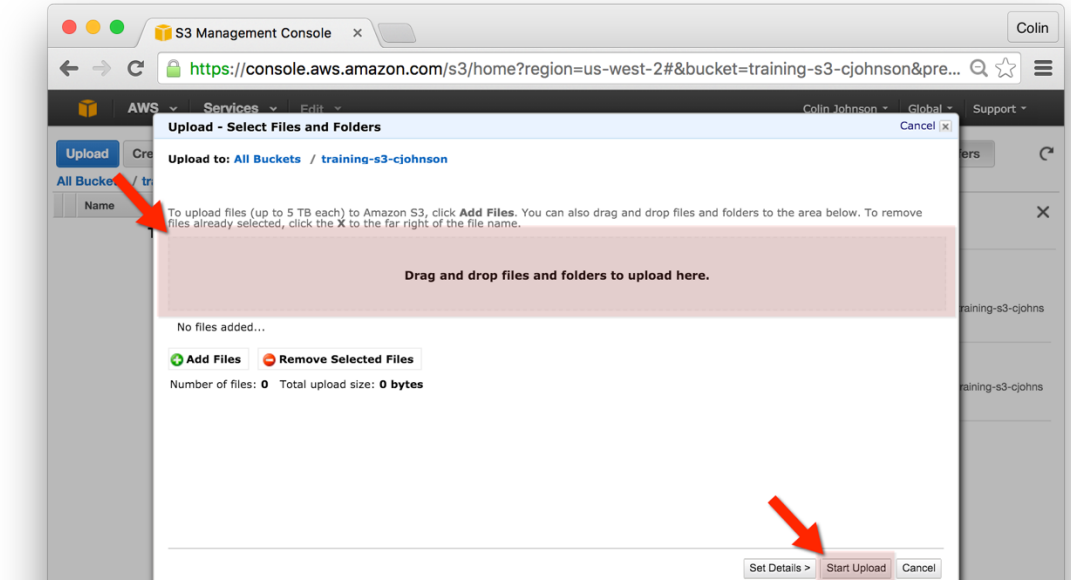
Upload an Object:

1. Click on the "Actions" button and select "Upload."



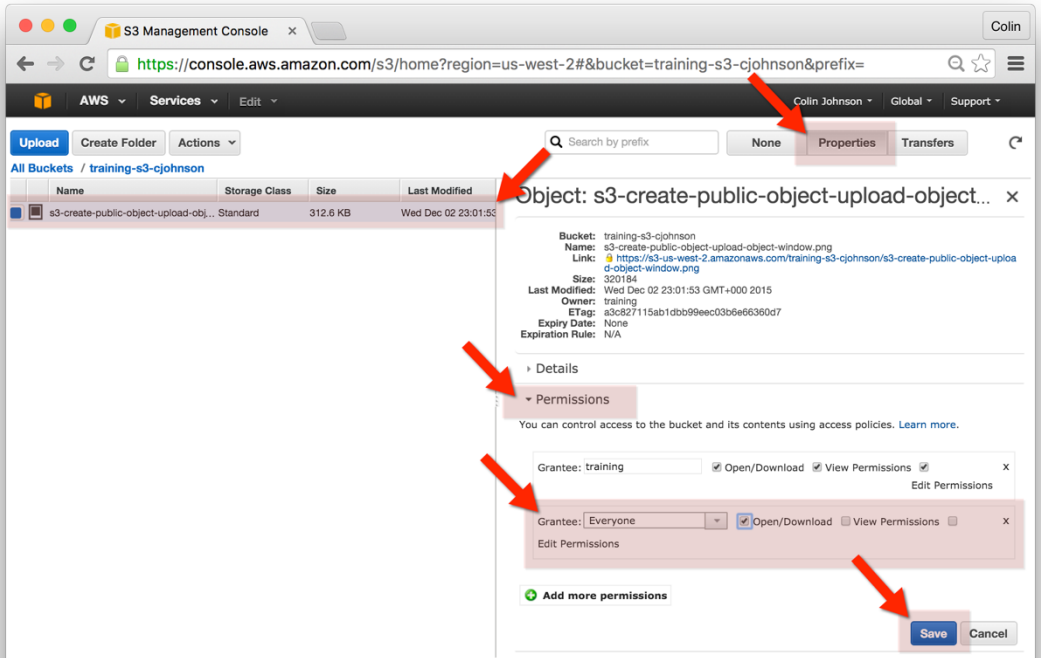
2. Drag and Drop a File onto the Upload tool. Ideally the file will be an image file in a format such as png or jpeg. Once a file is selected, click "Start

Upload”



Make the Uploaded Object Public:

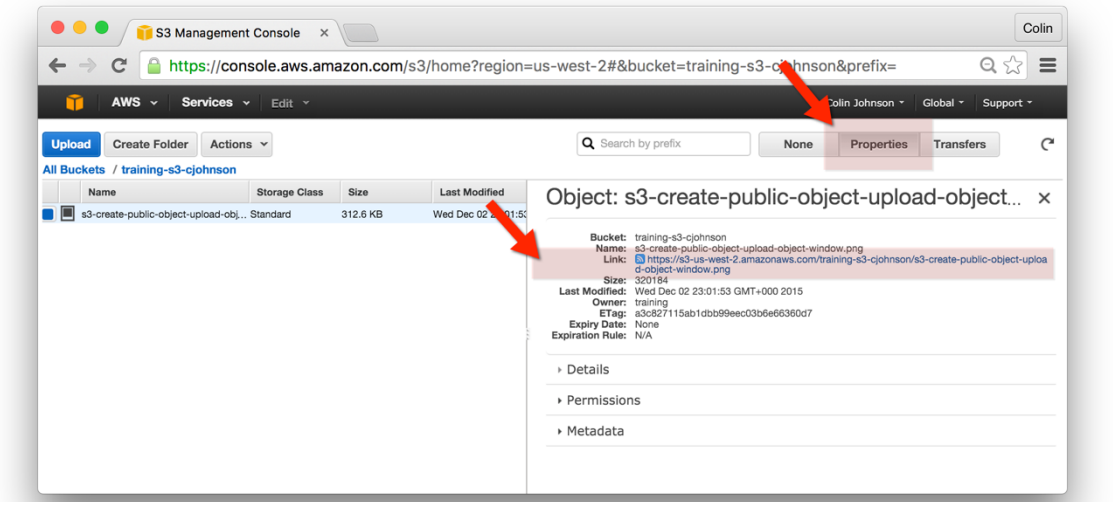
1. Select the uploaded object and select “Properties” from the upper navigation bar.
2. Select “Permissions” from the left-hand drop down.
3. Under “Permissions” click “Add more permissions.”
4. For “Grantee” type “Everyone”, click “Open/Download” and press the “Save” button to commit these changes to the Object ACL.



View the Uploaded Object:

1. Select the uploaded object and select “Properties” from the upper navigation bar.

2. Copy the “Link” property from the right hand-navigation and paste this into a new web browser window. The uploaded object should be served directly from AWS S3.

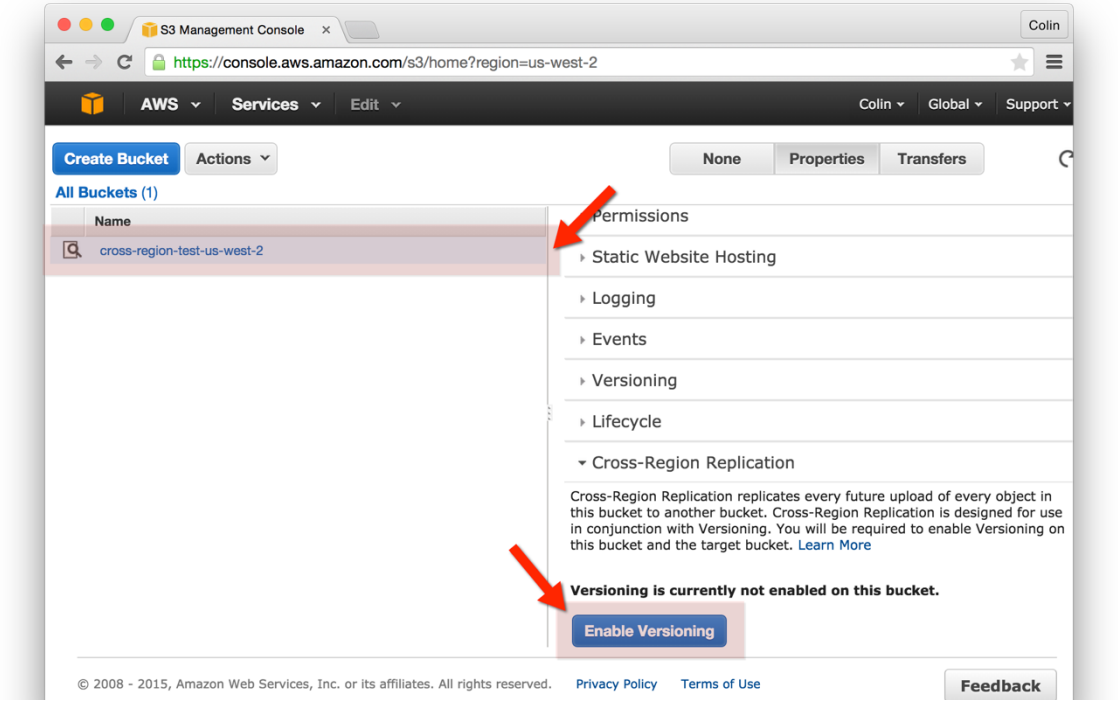


Exercise 13 – Cross Region Replication

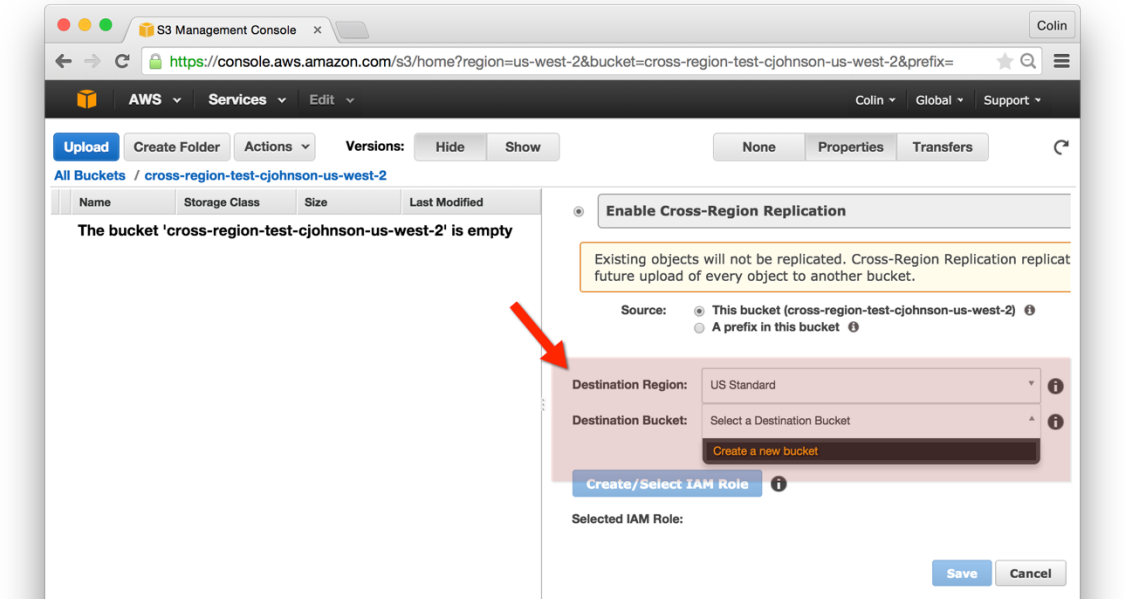
Instructions

Create an S3 Bucket in the us-west-2 (Oregon) Region:

1. Go to the AWS S3 Console
2. Click "Create Bucket"
    a. Bucket Name: cross-region-test-yourname-us-west-2
    b. Region: Oregon
    c. Click Create
3. Select the Bucket "cross-region-test-yourname-us-west-2" and click "Enable Versioning"



4. Click “Enable Cross-Region Replication” and select the following:



   a. Source: This Bucket
   b. Destination: US Standard
   c. Destination: New Bucket:
      i. Bucket Name: cross-region-test-yourname-us-standard
   d. Click “Create/Select IAM Role”
   e. Press “Save”

Upload an Object to your Oregon S3 Bucket:

1. Go to the AWS S3 Console
2. Go to the “cross-region-test-yourname-us-west-2” bucket.
3. Select “Actions” and select “Upload”
4. Drag and Drop a file to the “Upload” window and click “Start Upload”

Confirm the Object has been replicated to US-Standard S3 Bucket:

1. Go to the AWS S3 Console
2. Go to the “cross-region-test-yourname-us-standard” bucket.
3. Locate the uploaded object.

Exercise 14 – IAM User Access

Instructions

Overview:

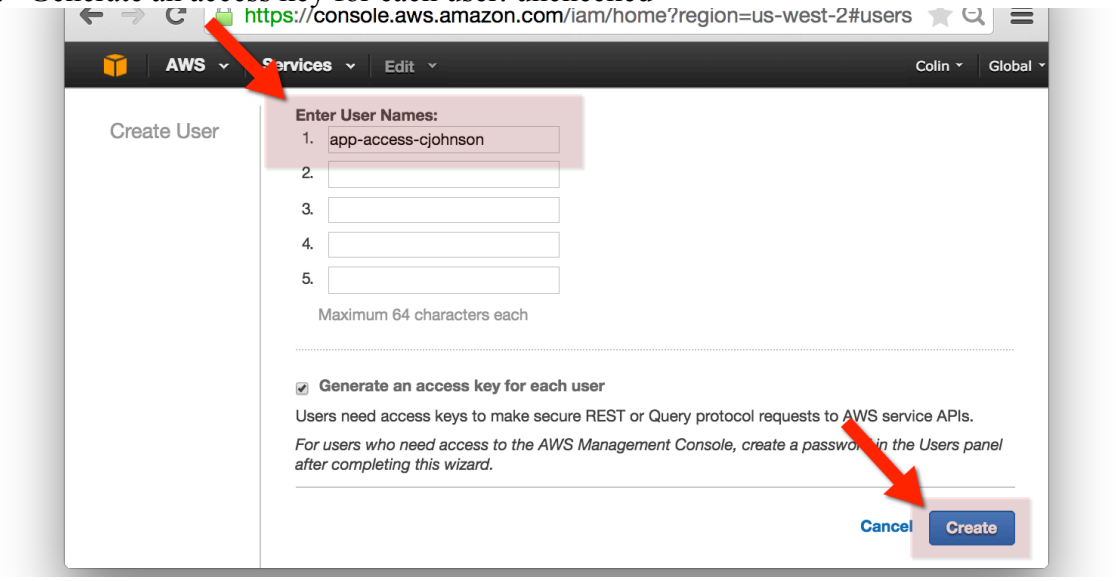
We'll be creating an IAM user that has access only limited access to an S3 bucket. This use case is fairly common – this is one common method for an application to use an AWS resource. Limiting access is particularly valuable – a misconfigured application could potential read/write/delete data from another environment or application. Limiting access using IAM is one method of ensuring an application accesses only the data it should be accessing.

Create an S3 Bucket in the us-west-2 (Oregon) Region:

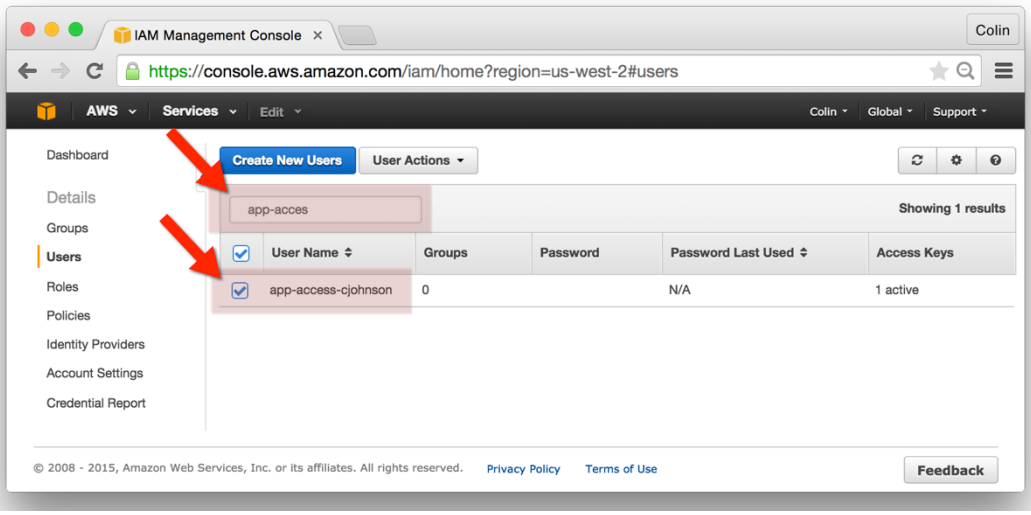
1. Go to the AWS S3 Console
2. Click "Create Bucket"
    a. Bucket Name: app-access-yourname
    b. Region: Oregon
    c. Click Create

Create a new IAM User:

1. Go to the AWS IAM Console and select "Users" from the left-hand navigation tab.
2. Click "Create New Users"
    a. Enter User Names: app-access-yourname
    b. Generate an access key for each user: unchecked



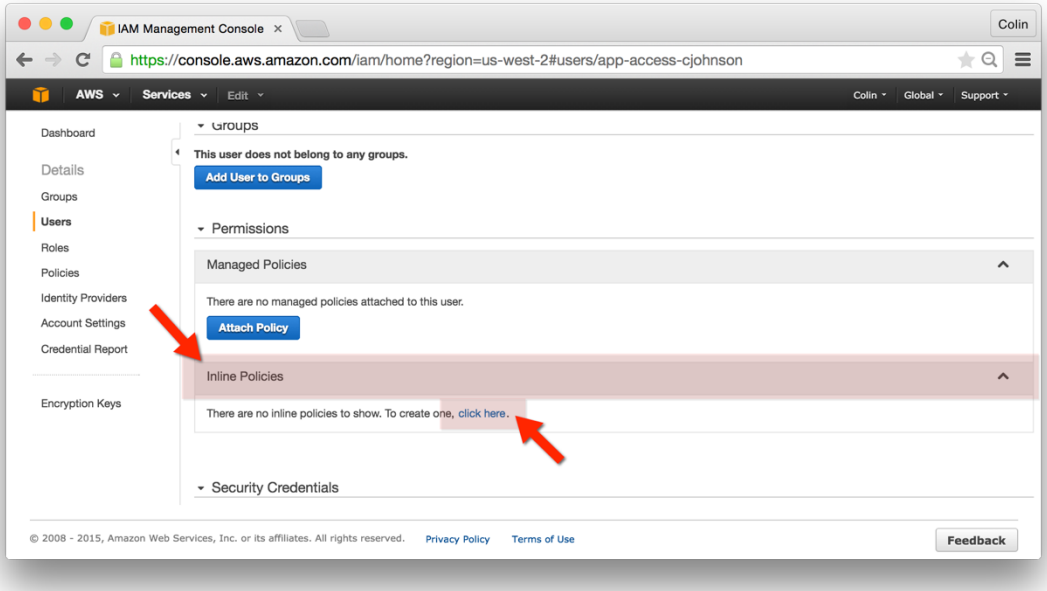
3. In the AWS IAM Console, select "Users" and search for the user app-access-yourname – click on this user.



4. Click the "Manage Password" button.
    a. Select "Assign an auto-generated password"
    b. Click "Apply"
5. Click "Show User Security Credentials"
    a. Make a note of your newly created user's password.
    b. Click "Close"

Allow Limited S3 Access to Your New User:

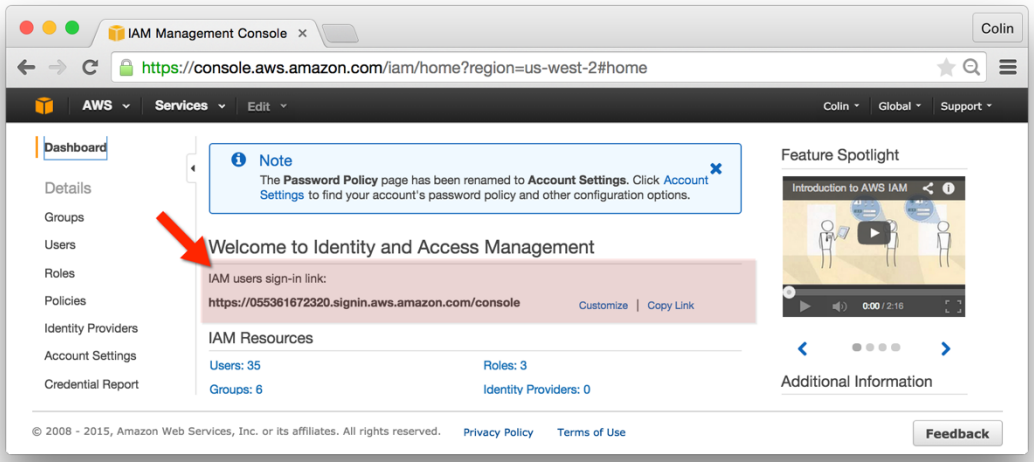
1. In the AWS IAM Console, select "Users" and search for the user app-access-yourname – click on your new user.
2. Click the arrow next to "Inline Policies" and "click here"



3. Click "Custom Policy" to create a policy:
    a. In the same directory as this file you should see an "app-access-policy.txt" file – open this document and paste the content as the Custom Policy.
    b. Click "Apply Policy"

Login to the AWS Console with your new user:

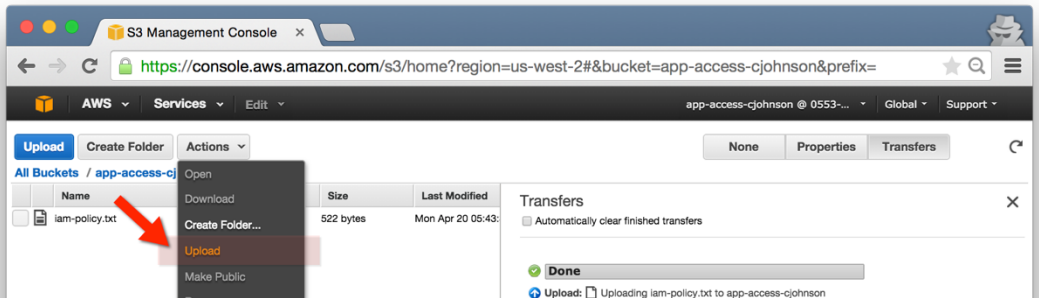
1. To get the address of the AWS Console, go to the AWS IAM Console and click "Dashboard" from the left-hand navigation.
2. Locate the "IAM users sign-in link" – it will look similar to https://055361672320.signin.aws.amazon.com/console



3. Open a new, private browser window and go to the "IAM User Sign In"

Upload an Object to the app-access-yourname Bucket:

1. In the new window, go to the AWS S3 Console.
2. Click on the "app-access-youname" Bucket
3. From the "Actions" menu, select "Upload"



    a. Drag and Drop a file to upload a file to this bucket.
4. Next, return to the AWS S3 Console and click on an S3 bucket – you should see the text "Sorry! You do not have permissions to view this bucket"

App Access Policy

```
{
 "Version": "2012-10-17",
 "Statement": [
   {
    "Effect": "Allow",
    "Action": "s3:ListAllMyBuckets",
    "Resource": "arn:aws:s3:::*"
   },
   {
    "Effect": "Allow",
    "Action": [
     "s3:ListBucket",
     "s3:GetBucketLocation"
    ],
    "Resource": "arn:aws:s3:::${aws:username}"
   },
   {
    "Effect": "Allow",
    "Action": [
     "s3:PutObject",
     "s3:GetObject",
     "s3:DeleteObject"
    ],
    "Resource": "arn:aws:s3:::${aws:username}/*"
   }
 ]
}
```

Exercise 15 – CloudFormation

Instructions

Overview:

We will use Amazon's CloudFormation to provision the following resources:

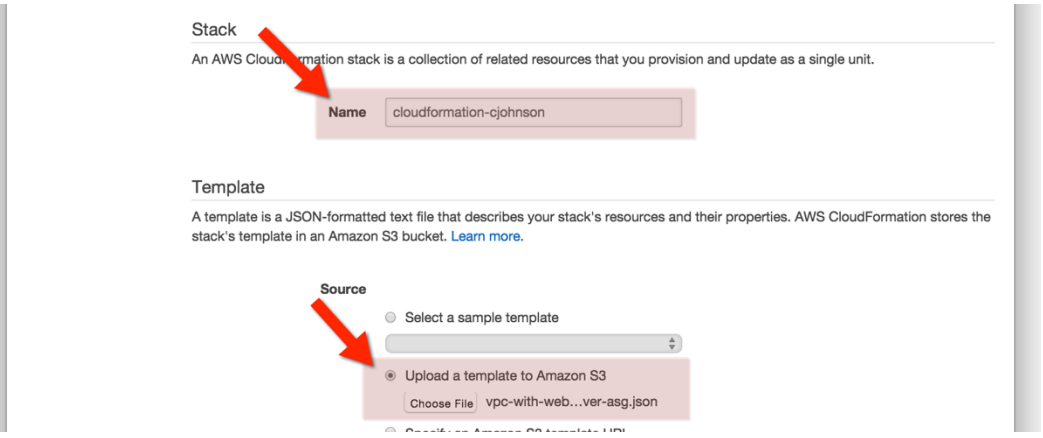
- a VPC, including
  - two Public subnets
  - an Internet Gateway
  - a Route Table, allowing access to the Public Internet
- a Web Server Security Group
- a Web Server Auto Scaling Group
- a Web Server ELB Security Group
- a Web Server ELB

The resources reference each other, as examples:

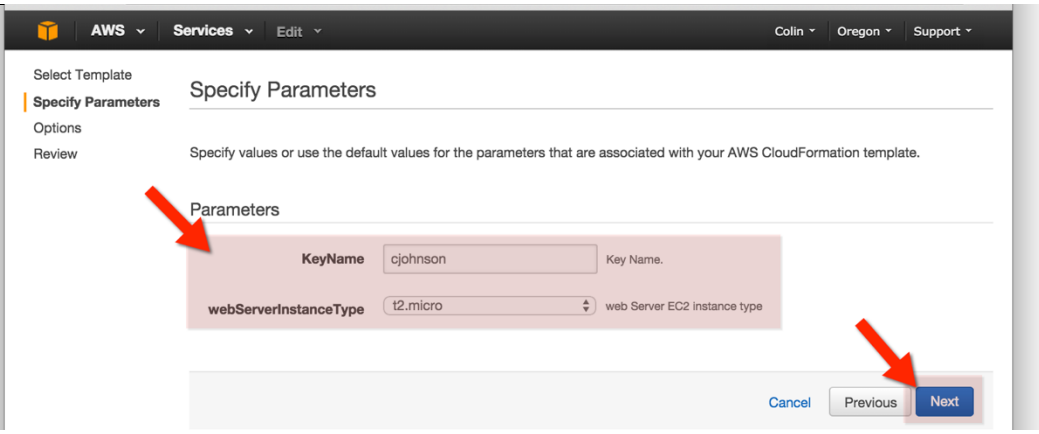
- the Web Server Security Group allows only the Web Server ELB in on port 80
- the Web Server Auto Scaling Group is associated with the Web Server ELB

Create the VPC and Web Server Stack:

1. Go to the AWS CloudFormation Console and click "Create Stack"
2. Select a Template:
   a. Name: cloudformation-yourname
   b. Source:
      i. Upload a template to Amazon S3:
         1. Click "Choose File" and browse to the file "vpc-with-webserver-asg.json"
   c. Click "Next"



3. Specific Parameters:
   a. KeyName: <your key name>
   b. webServerInstanceType: t2.micro



4. Options:
    a. Key: Owner
    b. Value: cjohnson
    c. Advanced: <leave unset>
5. Review:
    a. Click "Create"

Update VPC and Web Server Stack:

1. Go to the AWS CloudFormation Console and locate the stack that you created in the previous step. Click "Update Stack".
2. Select a Template:
    a. Name: cloudformation-yourname
    b. Template:
        i. Source: Use existing template.
    c. Click "Next"
3. Specify Parameters:
    a. webServerInstanceType: t2.small
    b. Click "Next
4. Options:
    a. Click "Next"
5. Review:
    a. Click "Update"

Note: some stack updates (and some AWS resource updates) do not take place immediately – this is the case with an Auto Scaling Group's Launch Configuration – to complete the resize of these instances they'll need to be "cycled" by terminating the given instances.

Delete VPC and Web Server Stack:

1. Go to the AWS CloudFormation Console and locate the stack that you created in the previous step. Click "Delete Stack."

VPC with Web Server ASG.JSON

```
{
 "AWSTemplateFormatVersion" : "2010-09-09",

 "Description" : "Create a VPC containing two subnets and an auto scaling group
containing instances with Internet access.",

 "Parameters" : {

  "webServerInstanceType" : {
   "Description" : "web Server EC2 instance type",
   "Type" : "String",
   "Default" : "t2.micro",
   "AllowedValues" : [ "t2.micro", "t2.small" ],
   "ConstraintDescription" : "must be a valid EC2 instance type."
  },
  "KeyName" : {
   "Description" : "EC2 Key Name",
   "Type" : "String"
  }
 },

 "Mappings" : {
  "AWSInstanceType2Arch" : {
   "t2.micro" : { "Arch" : "64" },
   "t2.small" : { "Arch" : "64" }
  },
  "AWSRegionArch2AMI" : {
   "us-east-1" : { "64" : "ami-7b89cc11" },
   "us-west-1" : { "64" : "ami-809df3e0" },
   "us-west-2" : { "64" : "ami-d24c5cb3" }
  },
  "AWSRegion2AZ" : {
   "us-east-1" : { "A" : "us-east-1b", "B" : "us-east-1c", "C" : "us-east-1d", "D" : "us-
east-1d" },
   "us-west-1" : { "A" : "us-west-1a", "B" : "us-west-1b", "C" : "us-west-1c"  },
   "us-west-2" : { "A" : "us-west-2a", "B" : "us-west-2b", "C" : "us-west-2c"  }
  }
 },

 "Resources" : {

  "VPC" : {
   "Type" : "AWS::EC2::VPC",
   "Properties" : {
    "CidrBlock" : "10.0.0.0/23"
   }
  },
```

```
  "InternetGateway" : {
    "Type" : "AWS::EC2::InternetGateway"
  },
  "PublicInternetRoute" : {
   "Type" : "AWS::EC2::Route",
   "DependsOn" : [ "InternetGateway", "PublicInternetRouteTable" ] ,
   "Properties" : {
    "DestinationCidrBlock" : "0.0.0.0/0",
    "GatewayId" : { "Ref" : "InternetGateway" },
    "RouteTableId" : { "Ref" : "PublicInternetRouteTable" }
   }
  },
  "VPCGatewayAttachment" : {
    "Type" : "AWS::EC2::VPCGatewayAttachment",
    "Properties" : {
      "InternetGatewayId" : { "Ref" : "InternetGateway" },
      "VpcId" : { "Ref" : "VPC" }
    }
  },
  "PublicInternetRouteTable" : {
    "Type" : "AWS::EC2::RouteTable",
    "Properties" : {
      "VpcId" : { "Ref" : "VPC" }
    }
  },
  "PublicSubnetA" : {
   "Type" : "AWS::EC2::Subnet",
   "Properties" : {
    "AvailabilityZone" : { "Fn::FindInMap" : [ "AWSRegion2AZ", { "Ref" :
"AWS::Region" }, "A" ] },
    "CidrBlock" : "10.0.0.0/25",
    "VpcId" : { "Ref" : "VPC" }
   }
  },
  "PublicSubnetB" : {
   "Type" : "AWS::EC2::Subnet",
   "Properties" : {
    "AvailabilityZone" : { "Fn::FindInMap" : [ "AWSRegion2AZ", { "Ref" :
"AWS::Region" }, "B" ] },
    "CidrBlock" : "10.0.0.128/25",
    "VpcId" : { "Ref" : "VPC" }
   }
  },
  "PublicSubnetARouteTableAssociation" : {
   "Type" : "AWS::EC2::SubnetRouteTableAssociation",
   "Properties" : {
    "RouteTableId" : { "Ref" : "PublicInternetRouteTable" },
    "SubnetId" : { "Ref" : "PublicSubnetA" }
   }
  },
```

```
   "PublicSubnetBRouteTableAssociation" : {
     "Type" : "AWS::EC2::SubnetRouteTableAssociation",
     "Properties" : {
       "RouteTableId" : { "Ref" : "PublicInternetRouteTable" },
       "SubnetId" : { "Ref" : "PublicSubnetB" }
     }
   },

   "webServerLaunchConfig" : {
     "Type" : "AWS::AutoScaling::LaunchConfiguration",
     "Properties" : {
       "AssociatePublicIpAddress" : "true",
       "KeyName" : { "Ref" : "KeyName" },
       "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                                 { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" :
"webServerInstanceType" },
                                 "Arch" ] } ] },
       "SecurityGroups" : [ { "Ref" : "webServerSecurityGroup" } ],
       "InstanceType" : { "Ref" : "webServerInstanceType" },
       "UserData": {
         "Fn::Base64": {
           "Fn::Join": [ "",
             [
               "#!/bin/bash -\n",
               "apt-get -y update\n",
               "apt-get -y install apache2\n",
               "ami_id=$(curl --silent http://169.254.169.254/latest/meta-data/ami-id)\n",
               "instance_id=$(curl --silent http://169.254.169.254/latest/meta-
data/instance-id)\n",
               "instance_type=$(curl --silent http://169.254.169.254/latest/meta-
data/instance-type)\n",
               "local_ip=$(curl --silent http://169.254.169.254/latest/meta-data/local-
ipv4)\n",
               "public_ip=$(curl --silent http://169.254.169.254/latest/meta-data/public-
ipv4)\n",
               "cat > /var/www/html/index.html <<EOF\n",
               "<html>\n",
               "<head>\n",
               "</head>\n",
               "<body>\n",
               "<h1>Server Information</h1>\n",
               "<ul>\n",
               "  <li>AMI: $ami_id</li>\n",
               "  <li>Instance ID: $instance_id</li>\n",
               "  <li>Instance Type: $instance_type</li>\n",
               "  <li>Local IP: $local_ip</li>\n",
               "  <li>Public IP: $public_ip</li>\n",
               "</ul>\n",
               "</body>\n",
```

```
            "</html>\n",
            "EOF\n"
          ]
        ]
      }
    }
  },
  "webServerSecurityGroup" : {
   "Type" : "AWS::EC2::SecurityGroup",
   "Properties" : {
    "GroupDescription" : "web Server Security Group",
    "VpcId" : { "Ref" : "VPC" },
    "SecurityGroupIngress" : [ {
     "IpProtocol" : "tcp",
     "FromPort" : "22",
     "ToPort" : "22",
     "CidrIp" : "0.0.0.0/0"
    },{
     "IpProtocol" : "tcp",
     "FromPort" : "80",
     "ToPort" : "80",
     "SourceSecurityGroupId" : { "Ref" : "webServerELBSecurityGroup" }
    } ],
    "SecurityGroupEgress" : [ {
     "IpProtocol" : "tcp",
     "FromPort" : "0",
     "ToPort" : "65535",
     "CidrIp" : "0.0.0.0/0"
    } ]
   }
  },
  "webServerAutoScalingGroup" : {
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
    "VPCZoneIdentifier" : [ { "Ref" : "PublicSubnetA" }, { "Ref" : "PublicSubnetB"
} ],
    "LaunchConfigurationName" : { "Ref" : "webServerLaunchConfig" },
    "MinSize" : "2",
    "MaxSize" : "2",
    "DesiredCapacity" : "2",
    "LoadBalancerNames" : [ { "Ref" : "webServerELB" } ],
    "Tags" : [ {
     "Key" : "Name",
     "Value" : "web-asgcfn-cjohnson",
     "PropagateAtLaunch" : "true"
    } ]
   }
  },
  "webServerELBSecurityGroup" : {
```

```
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "web Server ELB Security Group",
        "VpcId" : { "Ref" : "VPC" },
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "80",
          "ToPort" : "80",
          "CidrIp" : "0.0.0.0/0"
        } ],
        "SecurityGroupEgress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "0",
          "ToPort" : "65535",
          "CidrIp" : "0.0.0.0/0"
        } ]
      }
    },
    "webServerELB" : {
      "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties" : {
        "Subnets" : [ { "Ref" : "PublicSubnetA" }, { "Ref" : "PublicSubnetB" } ],
        "SecurityGroups" : [ { "Ref" : "webServerELBSecurityGroup" } ],
        "HealthCheck" : {
          "Target" : "HTTP:80/index.html",
          "HealthyThreshold" : "3",
          "UnhealthyThreshold" : "5",
          "Interval" : "30",
          "Timeout" : "5"
        },
        "Listeners" : [ {
          "LoadBalancerPort" : "80",
          "InstancePort" : "80",
          "Protocol" : "HTTP"
        } ],
        "CrossZone" : "true",
        "ConnectionDrainingPolicy": {
          "Enabled" : "true",
          "Timeout" : "60"
        }
      }
    }
  },
  "Outputs" : {
    "webServerELBDNSName" : {
      "Description": "The DNSName of the webServer ELB",
      "Value" : { "Fn::GetAtt" : [ "webServerELB", "DNSName" ]}
    }
  }
}
```

Exercise 16 – AWS Identity and Access Management

Instructions

Create IAM User and Group:

- Create an IAM Group
- Create an IAM User

Create a Limited Access IAM User and Test:

- Create an IAM Policy
- Create a user
- Apply Policy to user – allow access to resource by Tag
- Test Access Policy

Ec2-dev-tag-only.json

```
{
 "Version": "2012-10-17",
 "Statement": [
   {
    "Action": [
     "ec2:StartInstances",
     "ec2:StopInstances",
     "ec2:RebootInstances",
     "ec2:TerminateInstances"
    ],
    "Condition": {
     "StringEquals": {
      "ec2:ResourceTag/Owner":"${aws:username}"
     }
    },
    "Resource": [
     "arn:aws:ec2:*:*:instance/*"
    ],
    "Effect": "Allow"
   },
   {
    "Action": [
     "ec2:Describe*"
    ],
    "Resource": [
     "*"
    ],
    "Effect": "Allow"
   }
 ]
}
```

Exercise 17 – IAM Create Group User

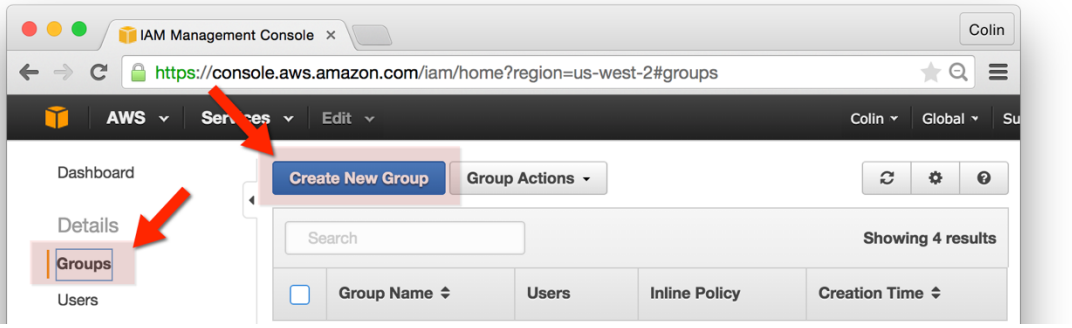
Instructions

Overview:

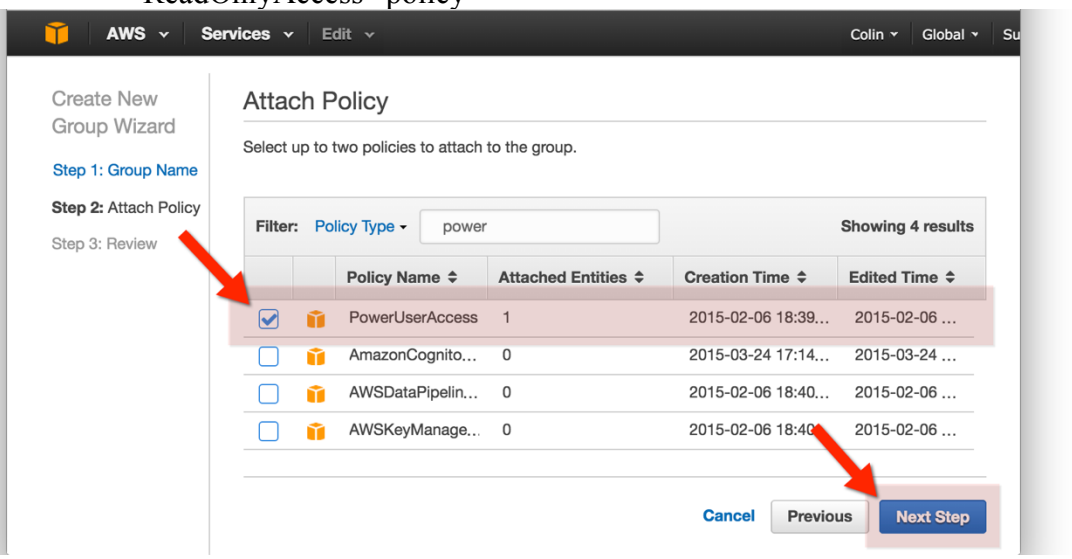
We will be creating one IAM Group named "Developers" and two "Developer" IAM Users.

Create an IAM Group:

1. Go the AWS Identity and Access Management Console and click "Groups" from the left-hand navigation.
2. Click "Create Group"



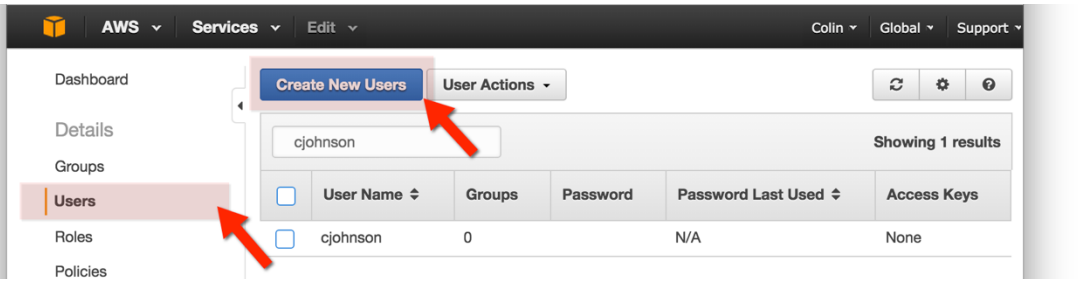
3. Group Name:
    a. Group Name: yourgroup-yourname
    b. Attach a Policy:
        i. select a policy – I might suggest the "PowerUserAccess" or the "ReadOnlyAccess" policy



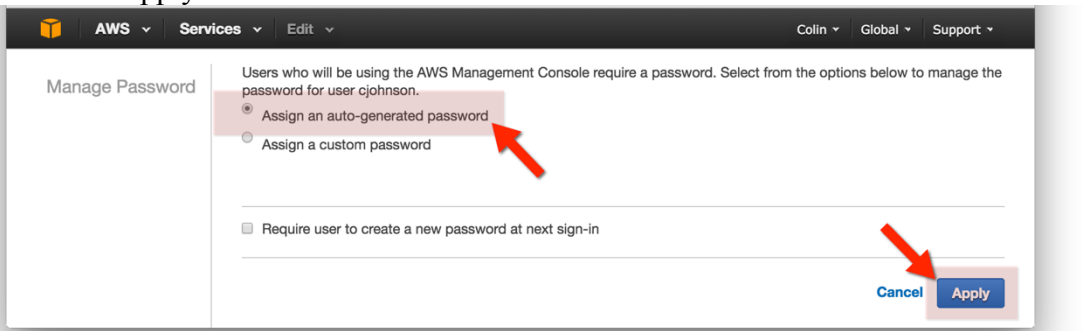
4. Review:
    a. Review the new group
    b. Click "Create Group" when done

Create an IAM User:

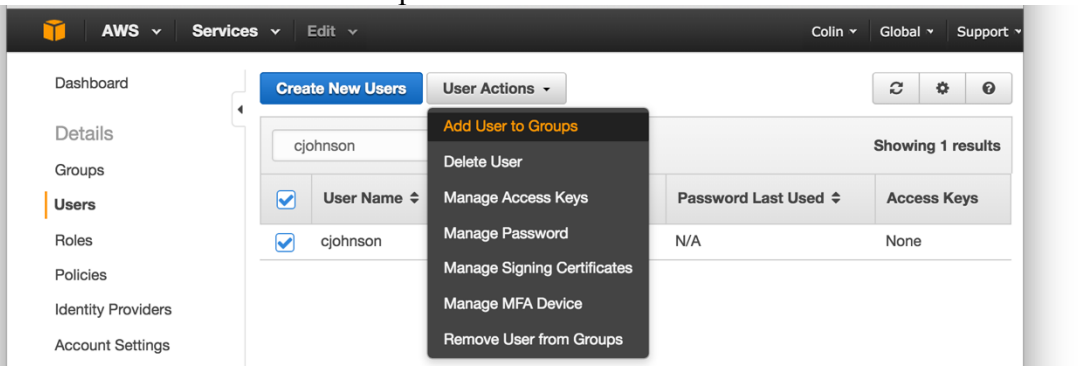
1. Go the AWS Identity and Access Management Console and click "Users" from the left-hand navigation.
2. Click "Create New Users"



3. Enter a User Name and press "Create"
4. Select the newly created user in the User List, click the "User Actions" drop-down and select "Manage Password"
    a. Choose "Assign an auto-generated password"
    b. Click Apply



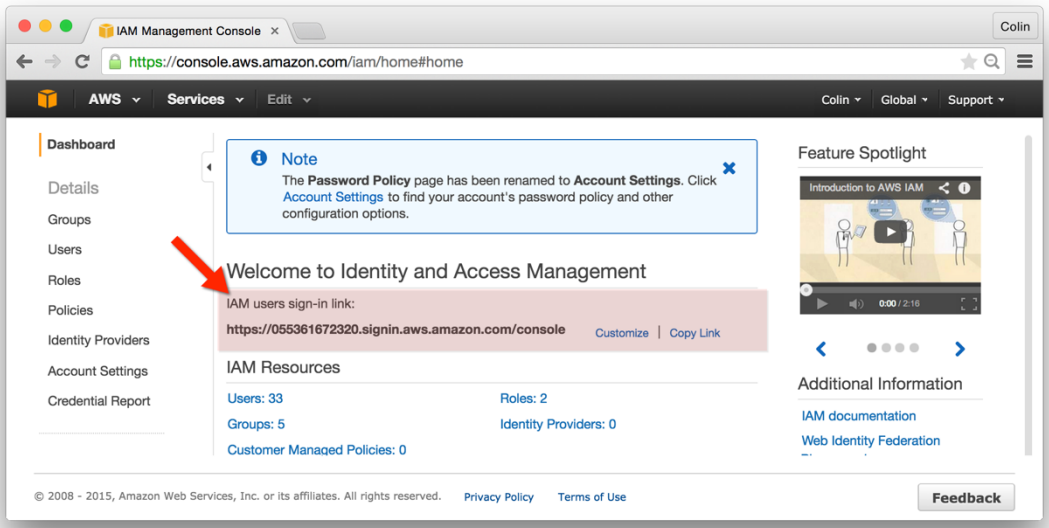
    c. Click "Show User Security Credentials" on the next screen – note this password.
    d. Press Close.
5. Select the newly created user in the User List, click the "User Actions" drop-down and select "Add User to Group"



6. Search for the "yourgroup-yourname" group created in the previous step and click "Add to Groups"

Login with the newly created IAM User:

1. Go the AWS Identity and Access Management Console and click "Dashboard" from the left-hand navigation.
2. Copy the "IAM Users Sign in Link" (it will look similar to https://055361672320.signin.aws.amazon.com/console)



3. Open a new browser window and paste this URL in.
4. Sign in as this account
5. Attempt to perform an action that was **not** granted to this account. For instance – try to delete your usual account.

Exercise 18 – Virtual Private Cloud (VPC)

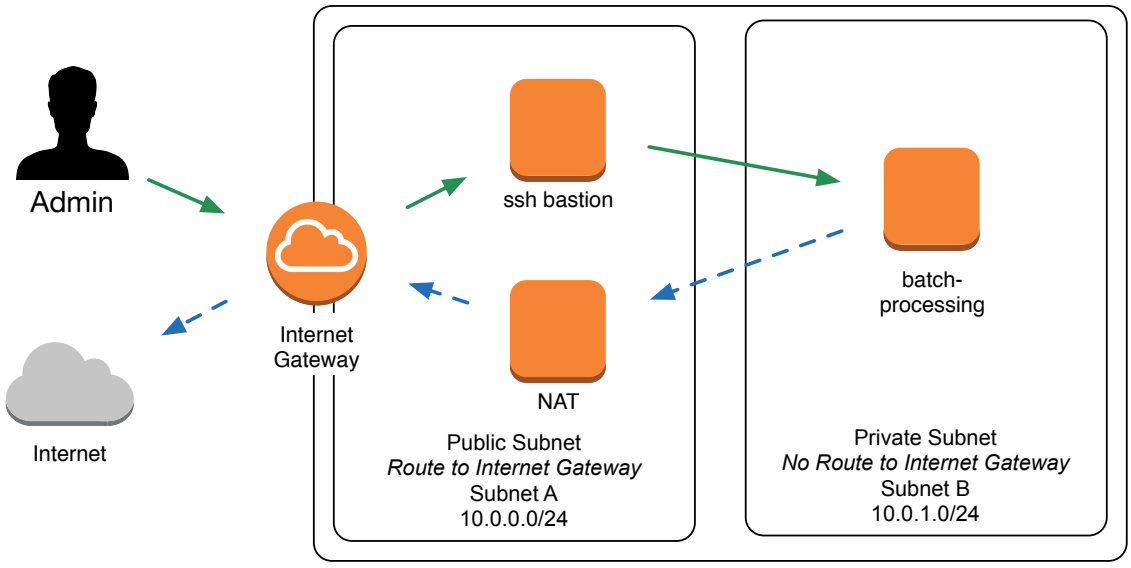
Instructions

Option 1: Build a VPC with Public and Private Subnets

Your VPC will need to include:

1. An Internet Gateway
2. Two Subnets:
    a. One Public Subnet (called Subnet A) associated with a route table that contains a Default Route to an Internet Gateway
    b. One Private Subnet (called Subnet B) associated with a route table that contains a Default Route to the NAT server
3. Two Route Tables:
    a. A Route Table with a default route to the Internet Gateway
    b. A Route Table with a default route to the NAT Server
4. A NAT Server and NAT Security Group
    a. Use AMI: ami-69ae8259
    b. Make sure to Disable "Source/Destination" Checks
    c. Security Group Allows HTTP in from batch-processing
5. An ssh Bastion Host
    a. Security Group Allows ssh in from 0.0.0.0/0
6. A Batch Processing Host
    a. Security Group Allows ssh Bastion Host in

Your VPC will be designed as follows:

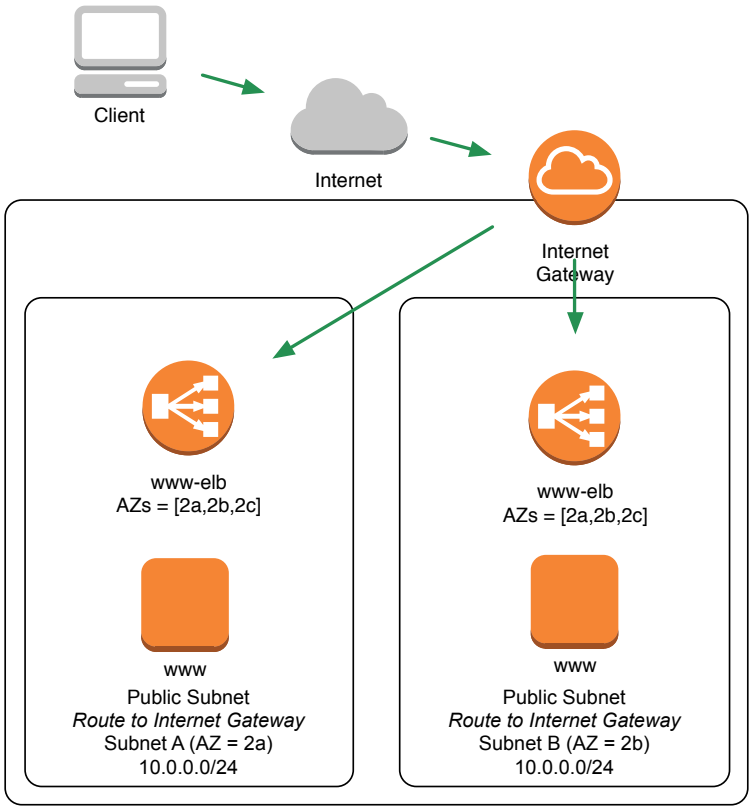


Option 2: Build a VPC Containing two Public Subnets and Web Servers, *optionally locking down with Network ACL*

Your VPC will need to include:

1. An Internet Gateway
2. Two Public Subnets:
    a. Both Subnets should be associated with a route table that contains a Default Route to an Internet Gateway
3. One Route Table:
    a. A Route Table with a default route to the Internet Gateway
4. Two Web Servers:
    a. Both will be members of the "www" Security Group
5. Optional:
    a. Lock down outbound using a Network ACL:
        i. Only allow inbound traffic on ports 22, 80
        ii. Only allow outbound traffic on ephemeral ports

Your VPC will be designed as follows:



Exercise 19 – Build out Simple VPC

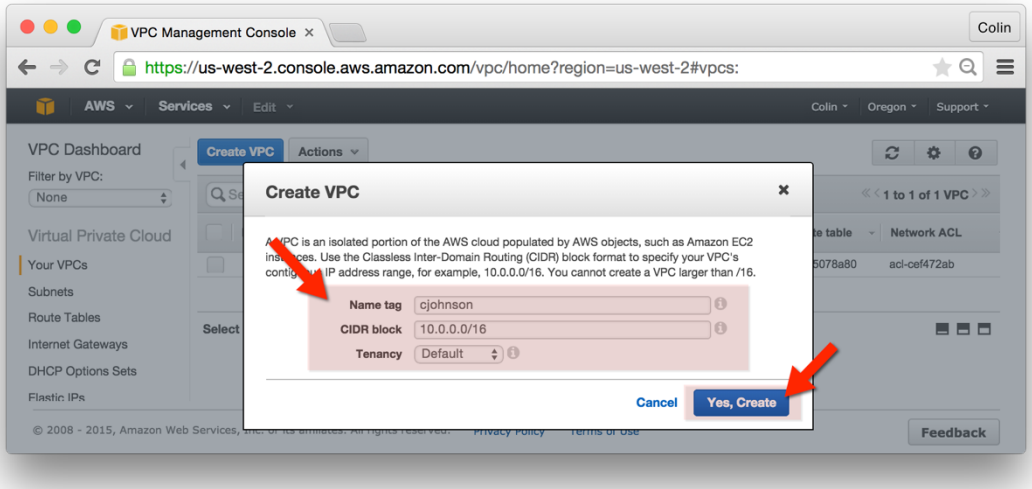
Instructions

Overview:

We will build out a simple VPC with all resources required for an instance to access the Internet – consider this instance a VPN or SSH gateway host.

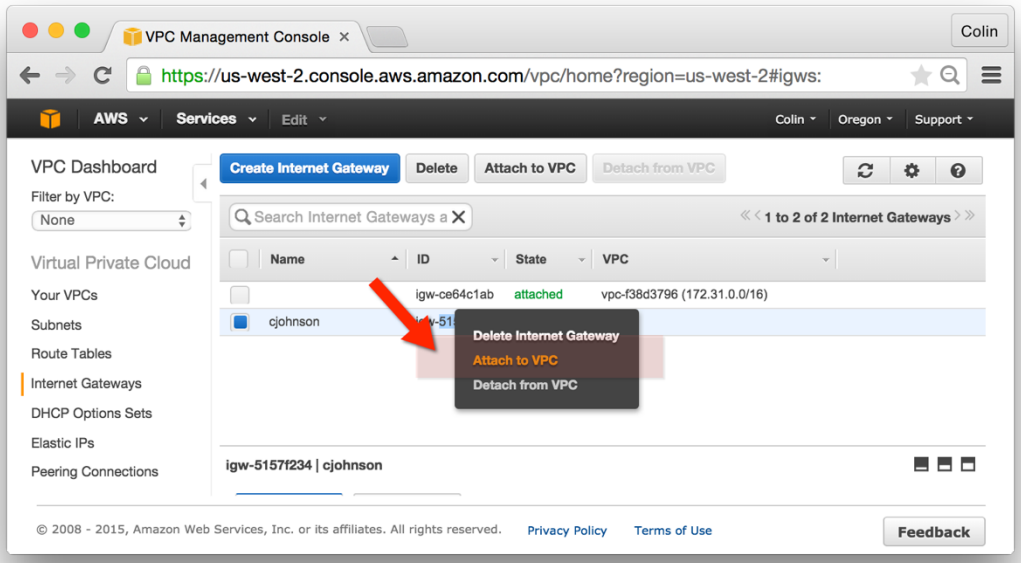
Create a VPC:

1. Go to the AWS VPC Console and click "Your VPCs" from the left-hand navigation bar.
2. Click "Create VPC"
3. In the Create VPC window, enter the following:
    a. Name tag: yourname
    b. CIDR block: 10.0.0.0/16
    c. Tenancy: Default
    d. Click "Create"



Create an Internet Gateway:

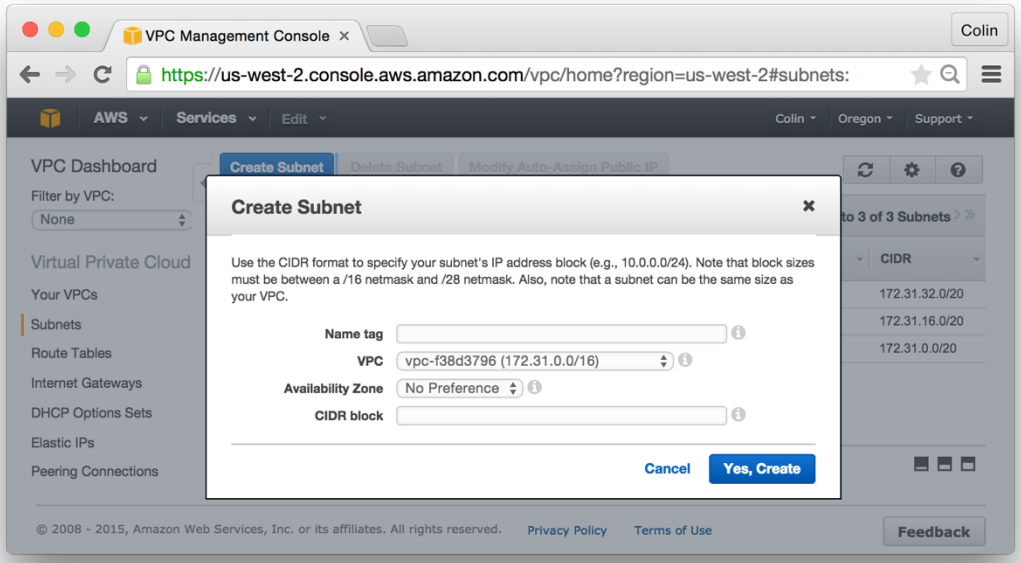
1. Go to the AWS VPC Console and click “Internet Gateways” from the left-hand navigation bar.
2. Click the “Create Internet Gateway” button.
3. In the Create Internet Gateway window, enter the following:
    a. Name tag: yourname
    b. Click “Yes, Create”
    c. The Internet Gateway will be created, in state detached.
4. Right-click your newly created Internet Gateway and select “Attach”
    a. Attach your newly created Internet Gateway to your VPC.



Create VPC "Public" Subnets:

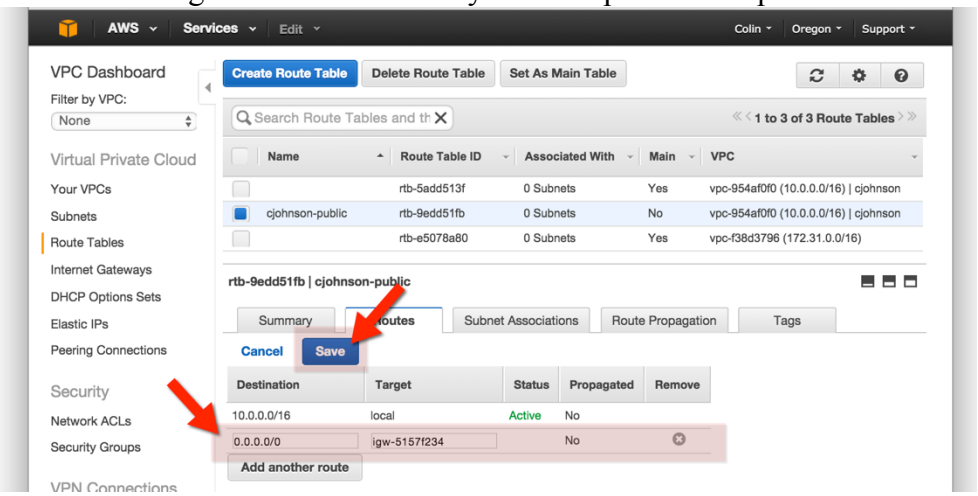
Note: the use of "Public" and "Private" subnets can confuse – when AWS documentation refers to a "Public" subnet they are referring to a subnet where instances have a Public IP address.

1. Go to the AWS VPC Console and click "Subnets" from the left-hand navigation bar.
2. Click the "Create Subnet" button and enter attributes as follows:
    a. Name tag: yourname-public-A
    b. VPC: <choose your VPC>
    c. Availability Zone: choose the "us-west-2a" Availability Zone
    d. CIDR block: 10.0.0.0/24
3. Click the "Create Subnet" button and enter attributes as follows:
    a. Name tag: yourname-public-B
    b. VPC: <choose your VPC>
    c. Availability Zone: choose the "us-west-2b" Availability Zone
    d. CIDR block: 10.0.1.0/24

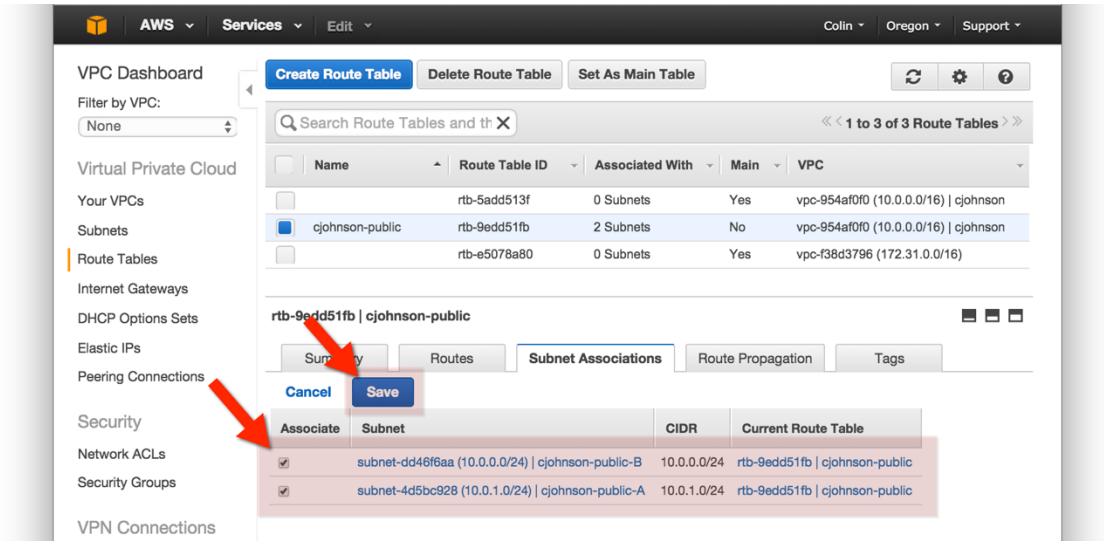


Create Route to Public Internet:

1. Go to the AWS VPC Console and click "Route Tables" from the left-hand navigation bar.
2. Click "Create Route Table" and enter attributes as follows:
    a. Name tag: yourname-public
    b. VPC: <choose your VPC>
3. Select the Route Table you just created:
    a. Click on the "Routes" tab and click "Edit"
    b. Click "Add another route" and enter information as follows:
        i. Destination: 0.0.0.0/0
        ii. Target: <Internet Gateway ID from previous step>



    c. Click on the "Subnet Associations" tab and click "Edit"
        i. Associate this Route Table with both of the previously created Public Subnets



        ii. Press Save