Class: AWS Administration

Project: #1 – AWS Cloud Formation Stacks

Version: 4.0

Author: Curt Chambers

**Lab Synopsis**

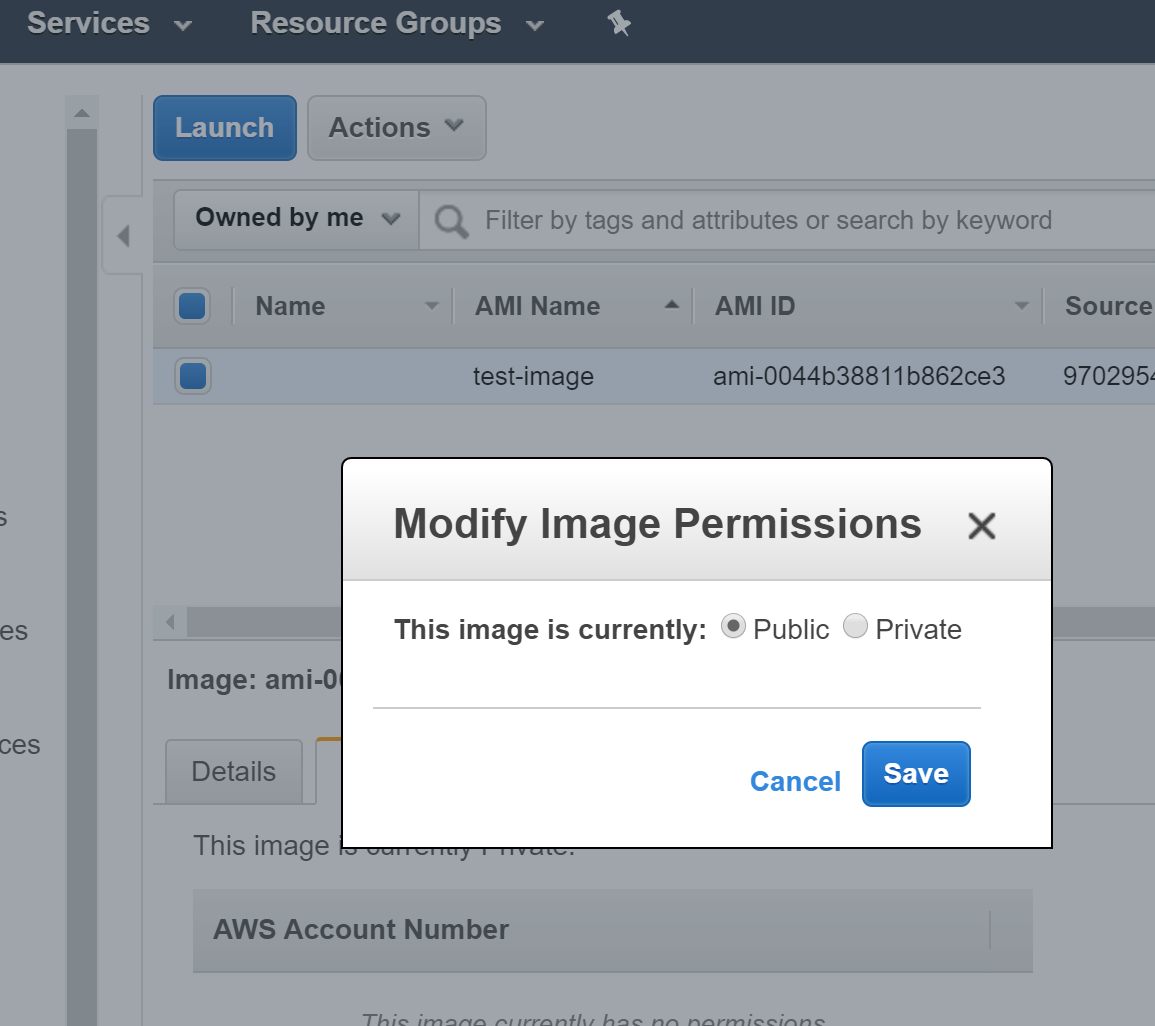
Create a stack of AWS infrastructure from TWO Cloud Formation templates. Be able to store the templates in S3 & launch from the AWS CLI. After testing the created infrastructure tear down the stack automatically with Cloud Formation.

**What to build**

\*\*\* Build all Cloud Formation templates in YAML – ignore any references to JSON \*\*\*

One of the labs in the AWS Academy Cloud Foundations content provided a good basic setup for production websites sitting “behind” a load balancer in a single region production website deployment. We’ll make use of that basic setup for this design. You’ll want to build this with TWO Cloud Formation templates – we are making a few modifications from the original lab:

1. Before you start constructing the Cloud Formation templates you’ll need to build a custom AMI to use as your launch AMI for the two webservers built in template #2. See the section in this document called “Custom Web Server AMI Build” for tips on building that.
2. You WILL need to make the AMI PUBLIC after you create it so I can access it for grading when I run your templates:

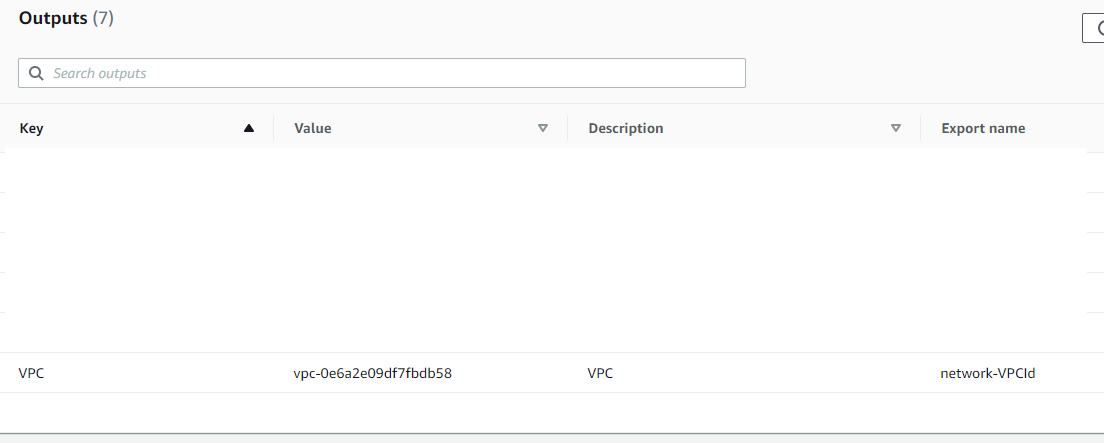


1. AWS Cloud Formation templates you will construct:

**Cloud Formation Template #1 (see the YAML outline document for a starting point)**

MUCH of the code needed for this template you can pull from the Cloud Formation Lab template you constructed! No need to reinvent the wheel.

* 1. VPC 10.0.0.0/16
  2. Internet Gateway
  3. 4 Subnets built in two availability zones:
     1. Public Subnet 1 – 10.0.1.0/24
     2. Public Subnet 2 – 10.0.2.0/24
     3. Private Subnet 1 – 10.0.3.0/24
     4. Private Subnet 2 – 10.0.4.0/24
  4. Appropriate Security Groups (2 total)
     1. One Security Group for the Application Load Balancer
     2. One Security Group used by both Web Instances
  5. You will export the values of the following from template #1 so you can import them into template #2: VPC ID, All Subnet IDs, Both Security Group IDs.
     1. Look at my example of how this works in the template #1 YAML outline under “Outputs”. Here we are exporting the VPC ID value into an export that will be named: “WhateverYouCalledYourStack-VPCId” Look at this export after template #1 runs in Cloud Formation in my example where my template is named “network”:



**Cloud Formation Template #2 (see the YAML outline document for a starting point)**

1. Application Load Balancer (A Type of Elastic Load Balancer): ALB
2. 2 Web Instances (labelled as Web Instance 1 and 2 in the diagram) (built from the custom AMI and registered with the ALB)
3. Add a Bastion Host as need be if you have trouble with your web servers so you can login into them & see what is happening - this is OPTIONAL.
4. For Project #1 we will NOT have a NAT Gateway - we’ll add this in Project #2.

**Custom Web Server AMI Build**

Before beginning construction of the Cloud Formation templates you’ll need to build out an EC2 Instance to create an AMI image from that will be used to launch the web instances shown in the diagram. Once you have a working AMI saved out you can terminate this initial EC2 instance.

Make sure to set the permissions on the AMI to public so I can see it. The construction of the AMI will result in a small EBS snapshot that will be saved in your account - this is necessary since it backs the AMI - don’t delete it.

User data to launch your initial webserver with to install the website. This will NOT have to be repeated for your Web Servers launched with the Cloud Formation template because it will already be incorporated into the AMI.

#!/bin/bash -ex

yum -y install httpd php mysql php-mysql

chkconfig httpd on

service httpd start

if [ ! -f /var/www/html/lab2-app.tar.gz ]; then cd /var/www/html

wget https://us-west-2-aws-training.s3.amazonaws.com/awsu-ilt/AWS-100-ESS/v4.2/lab-2-configure-website-datastore/scripts/lab2-app.tar.gz

tar xvfz lab2-app.tar.gz

chown apache:root /var/www/html/rds.conf.php

fi

**Architecture Diagram**

