Security Architecture Lab Guide

Sponsored by

August 2018

v1.0

Logistics

- **Format:** Overview presentation and lab setup, followed by paced exercises with a section recap.
- **Workshop Duration**: 1.5-2 hours
- **Target Audience:** Technical security users (security engineers, architects, DevOps) who have heard of Dome9 and know what Dome9 offers
- The organizing team is comprised of one speaker and 1-2 technical staff to help out and answer questions
- Participants bring their own laptops and have an AWS account setup (preferably beforehand) - **Please do this early on since it takes a few hours for a new AWS account to sync with a CFT template**
- Participants need to download the Cloudformation (**CFT**) template to run this lab - Please download CFT from the Github here

Table of Contents

AWS Setup

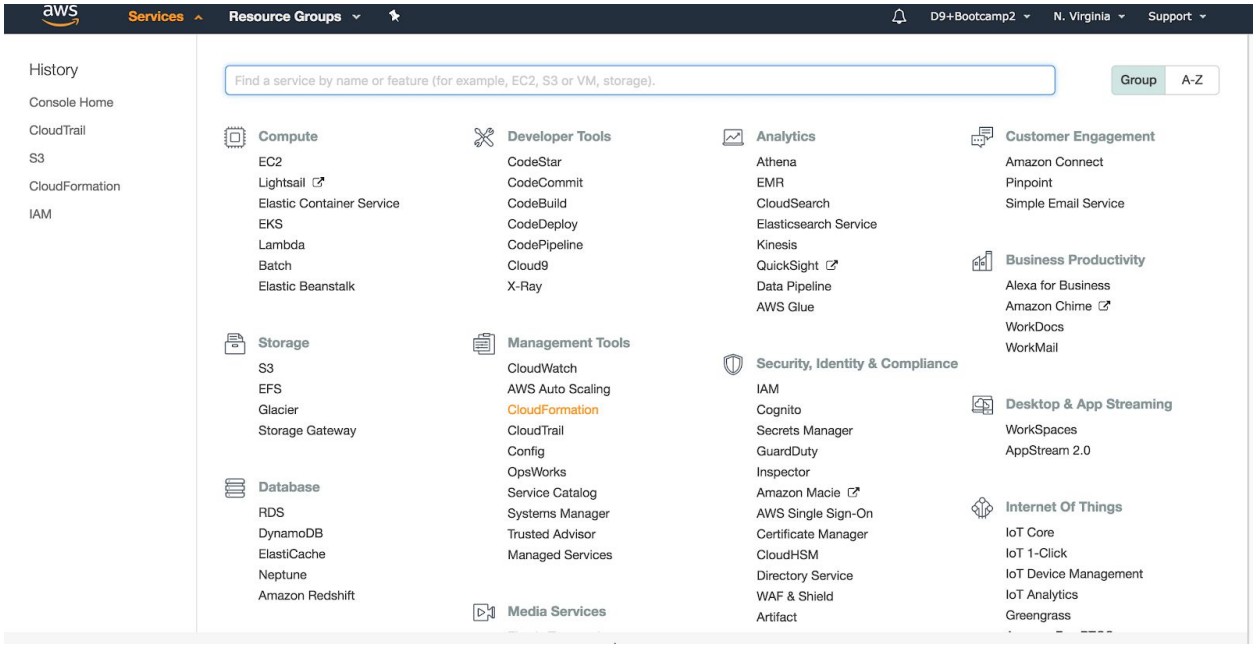
Exercise 1.1: Setup lab environment (Login to your lab AWS account)

The instructor should provide you with an AWS Account for this workshop. Ensure you have an AWS account setup before proceeding.

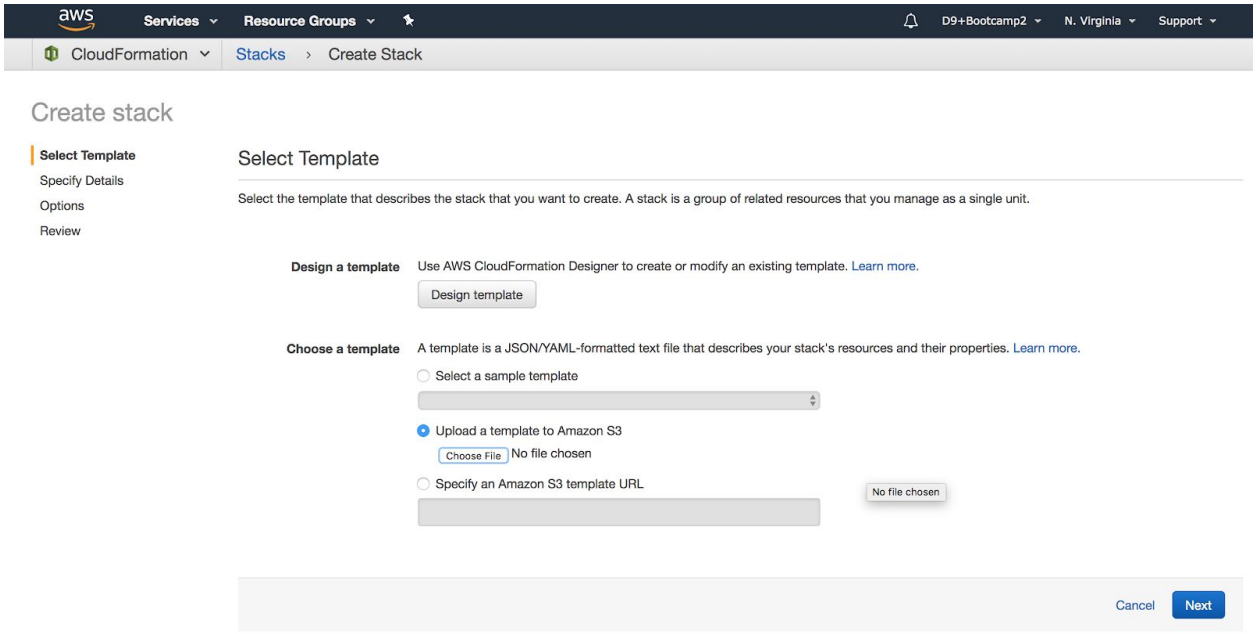
Exercise Complete!

Exercise 1.2: Deploy sandbox AWS environment

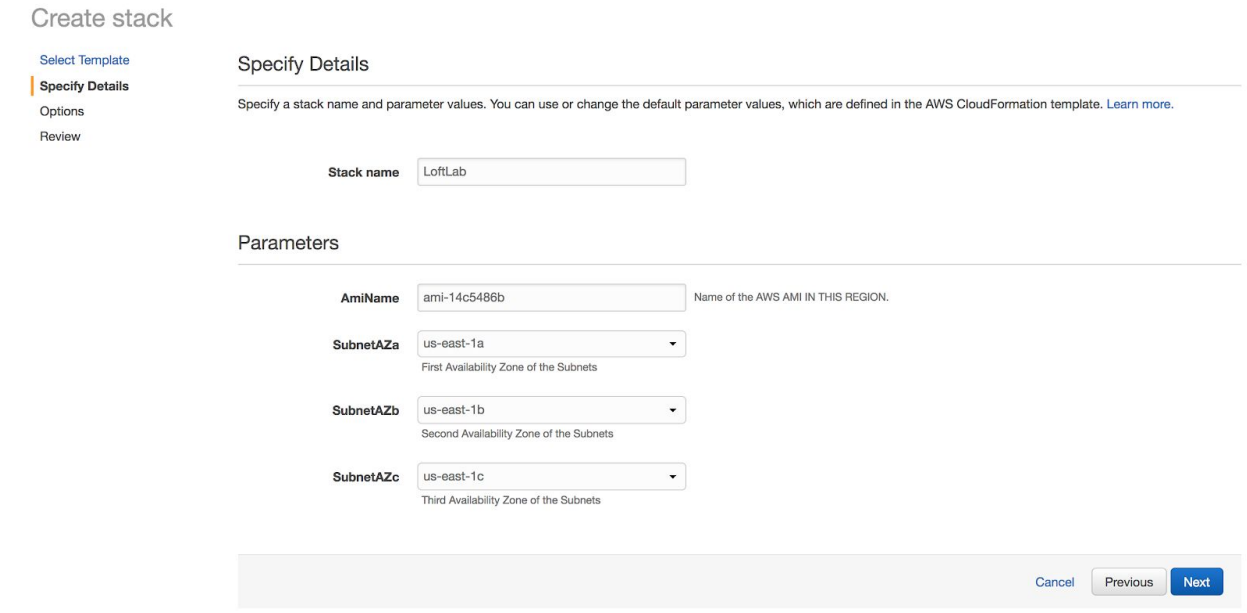
Navigate to AWS Cloudformation



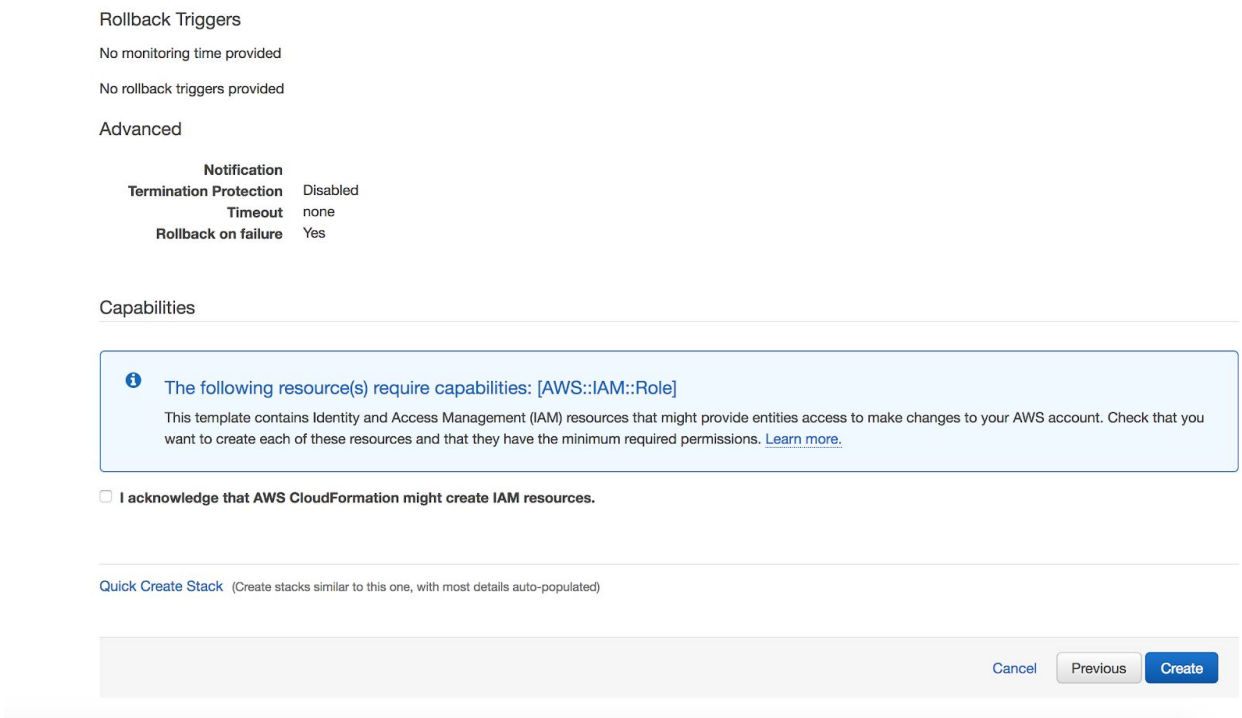
Click on create stack and select “upload a template to S3” and choose the CFT file that you downloaded and click next



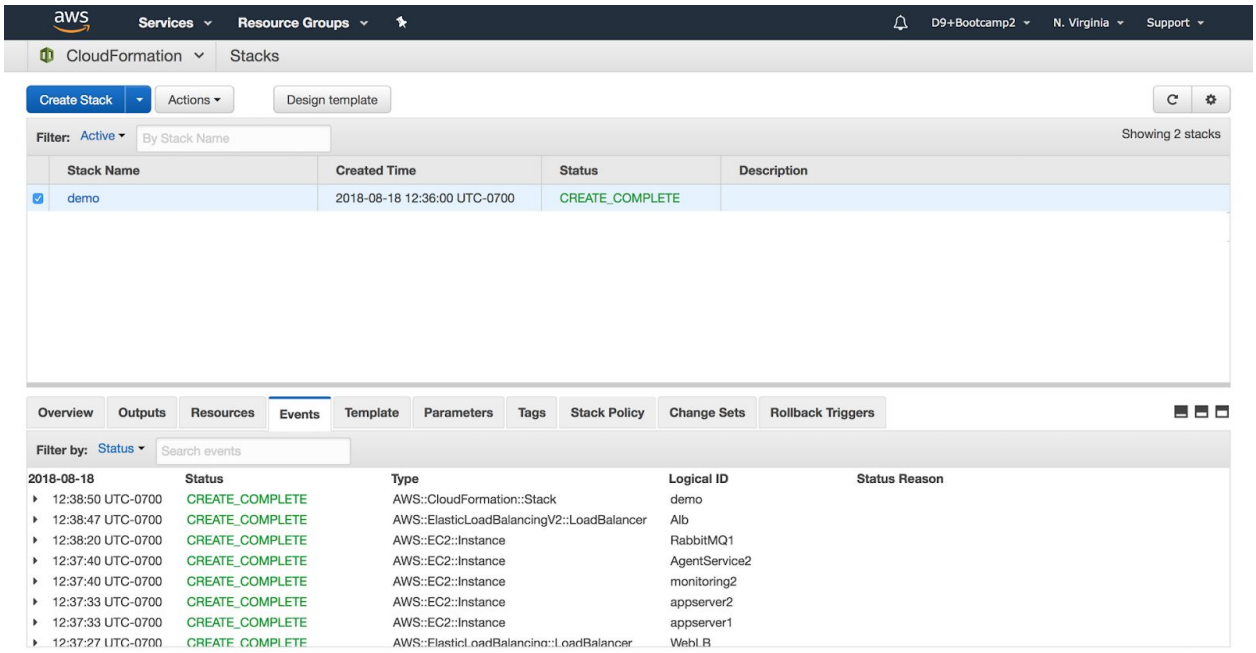
Create a stack name such as “**<yourname>LoftLab”** and select us-east1-a, 1-b, and 1-c for subnetAza, subnetAzb, subnetAzc and click next



Click next **twice** and select "**I acknowledge that AWS Cloudformation might create resources"** and click create.



You will need to wait 5 minutes for the CFT to automatically deploy the environment in your AWS account. At the end you should see the below screen:

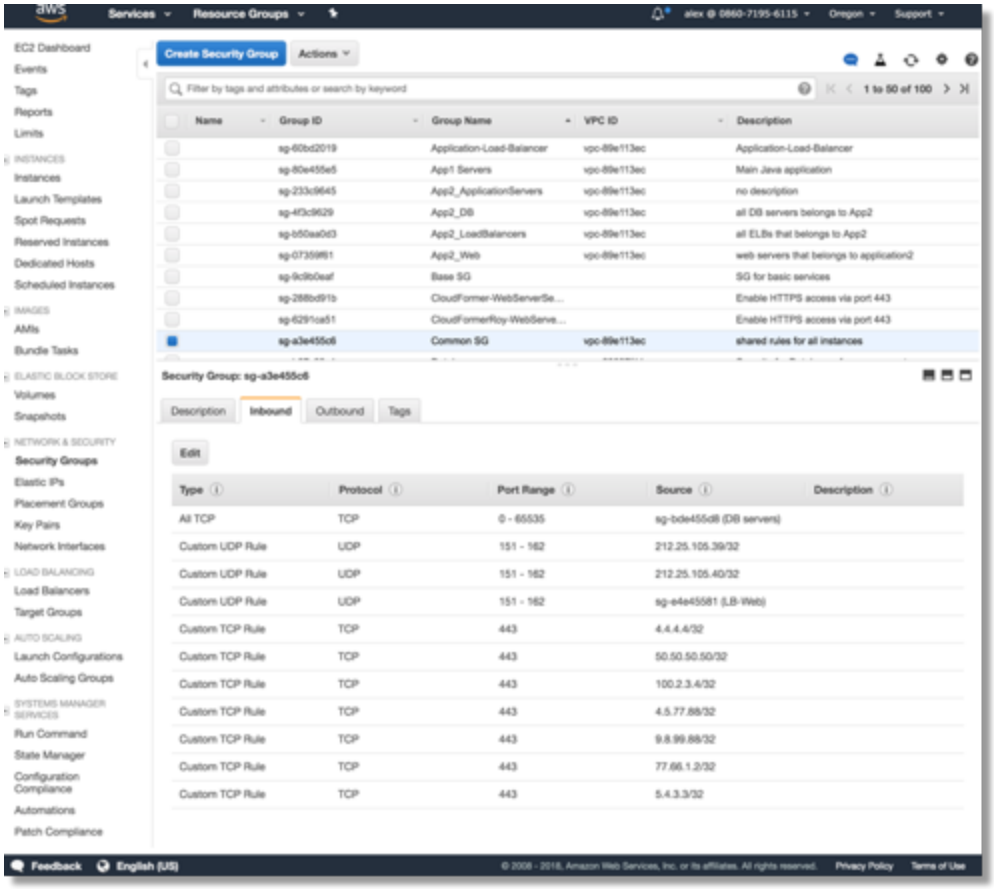


Exercise Complete! You have now deployed the sandbox AWS environment in your account.

AWS Security Policy Lab

Exercise 2.1: Identify zombie security group

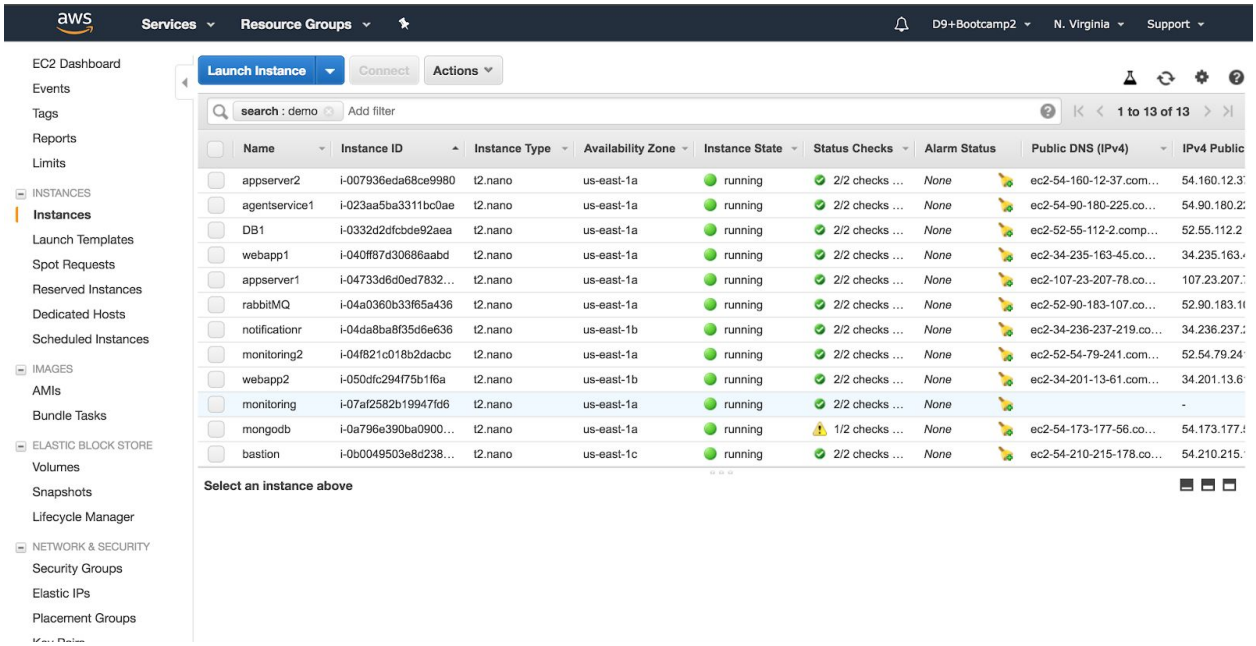
In your AWS account, navigate to **N.Virginia region**. Explore EC2 instances, Security Groups, IAM and other services.



Hint: A zombie security group is a security group that has a permissive rule but has no instances tied to it!

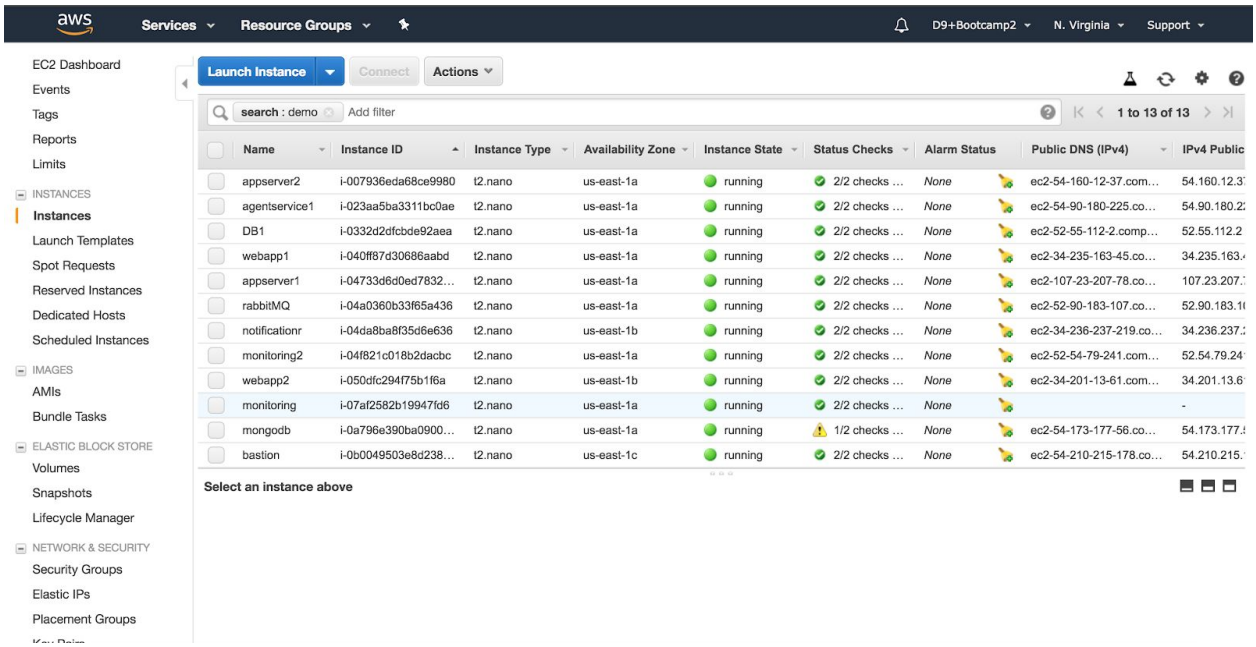
Exercise Complete! You have now found your zombie policy!

Exercise 2.2: Identify an exposed internal asset in the AWS environment - Part 1



There is an internal asset that is exposed to the public. Can you find it?

Exercise 2.3: Identify an exposed internal asset in the AWS environment - Part 2



There is another internal asset that is exposed to the public (this one is harder to find) Good luck!

Exercise Complete! You have now found your exposed assets!

Section Wrap Up

- Summary of misconfiguration issues
- Dome9 Clarity

S3 Security Lab

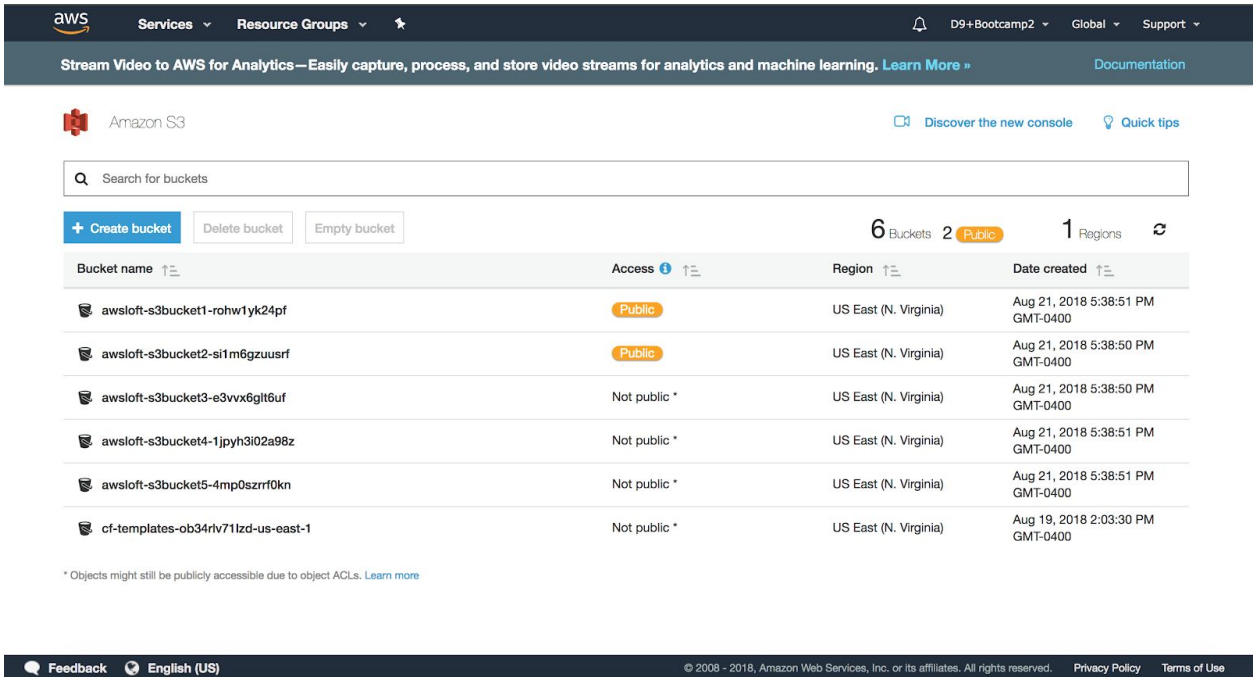
Overview of S3 bucket security

In this lab, you will build upon each exercise by starting from an exposed bucket and adding JSON policies to secure S3 buckets. There are endless realms of possibilities with S3, hence we will focus on a few key items in this lab.

Exercise 3.1: S3 Access Controls (exposed ACLs and Bucket policies)

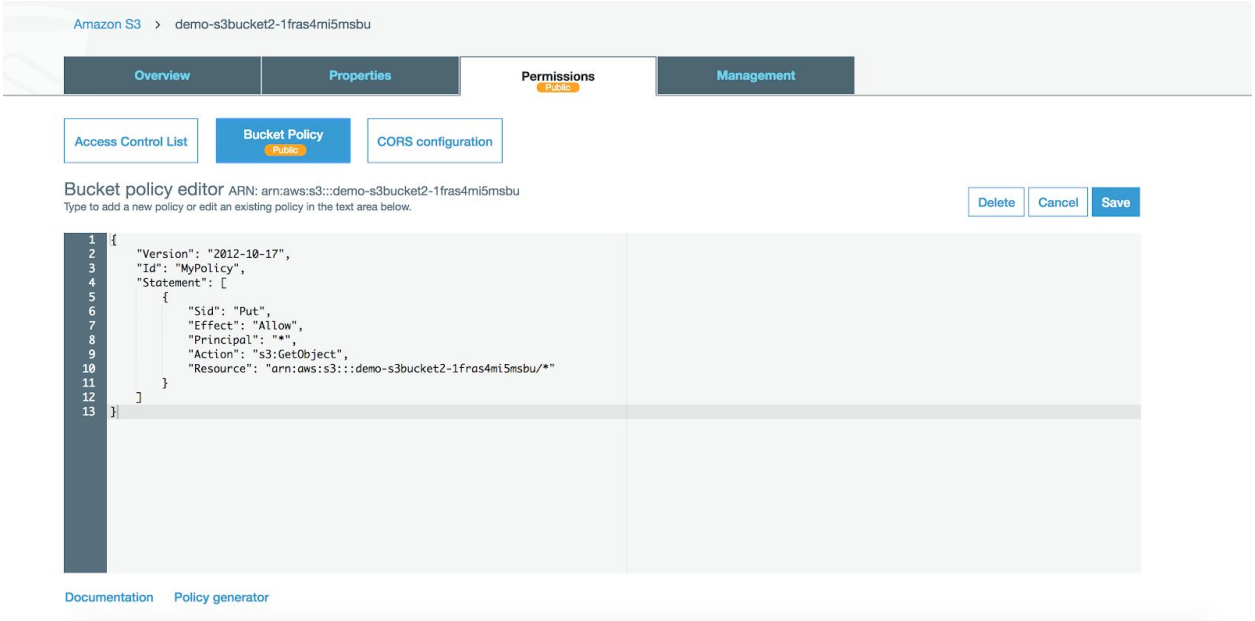
The goal of this exercise is to help you understand how to control S3 bucket access. We will focus on how to ensure specific role can interact with S3 and thereby not allowing any anonymous/outside users have access to sensitive data in the bucket. We will also look at least privilege concept, where you as an admin decide who has access (whitelist) to the most sensitive S3 operations and deny everyone else.

Navigate to the S3 console in AWS (FYI - You should see 6 buckets) We will go chronologically, starting from bucket 1



1. Fix <yourCFT>S3Bucket1 (this is straightforward)

2. Now let's move to <yourCFT>S3Bucket2. Write JSON policy (start from the existing configuration seen below)



a. Access: Ensure S3 bucket is not publicly accessible
b. Bucket Policy: Access Management Allow **S3-Role** to put and get objects in this bucket
c. IAM policy: Modify **S3-Role** to allow put and get objects for S3 (useful when you want to set overall S3 policies at a user/role level rather than bucket)
d. Bucket Policy: Least Privilege enforcement - Ensure deletion of S3 bucket can only be done based on your specific AWS account id (this is important for sensitive buckets that may have other accounts accessing it, and you want to make sure the most critical operations are whitelisted)

Exercise Complete! You have now ensured no buckets are publicly exposed!

Exercise 3.2: S3 Encryption Best Practices

Explore permissions of <yourCFT>S3Bucket3. Now that this bucket is not public and has the right permission settings, lets see how we can properly encrypt S3. A key aspect of data security is protecting data-in-flight and data at rest. Best practice dictates that all data in the cloud be encrypted both at rest as well as in flight when data is read from or written to a bucket. This can be done easily using Secure Sockets Layer/Transport Layer Security (SSL/TLS). Encrypting data in flight helps protect against man-in-the-middle and sniffing attacks.

For your <yourCFT>S3Bucket3:

1. Data at rest: Turn on server-side encryption (AES-256)
2. Data in flight: Bucket policy should only allow get/Put to Amazon S3 objects only through HTTPS

Exercise Complete! You have now following a few S3 encryption best practices

Exercise 3.3: S3 Logging

This exercise will focus on enabling logging on your S3 bucket. There are two types, object level and server level. Object level enables AWS CloudTrail trail to log data events for objects in an S3 bucket by using the Amazon S3 console. You can use it to control what buckets/prefixes,objects will be audited. Server access logging provides detailed records for the requests that are made to a bucket. This is managed by S3 alone and cannot filter what events will be logged.

For your <yourCFT>S3Bucket4:

1. Enable logging server access logging

Exercise Complete! You have now following a few logging best practices

Section wrap up

- Discuss challenges in continuous monitoring and assessment
- Dome9 Compliance Engine

Offboarding

Please navigate to the Cloudformation and click the stack and delete stack. You are now done!