

Lab 2- Infrastructure as Code - Operations

Day 2 Operations - While Infrastructure as Code provides us the ability to quickly create resources across infrastructure, it's important that modifications to the existing infrastructure are also made through code as well. Rather than a manual process, implementing changes through code ensures consistency while providing a path to continuous integration and automated testing. The initial templates, and modifications to any templates, generally follow processes such as code reviews and the use of version control tools, such as GitHub.

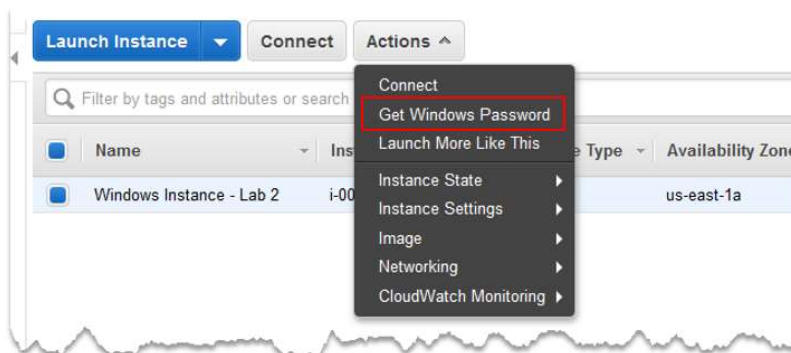
Goal – This lab is broken up into two (2) parts. The first section of this lab will teach users how to create and apply change sets to modify existing CloudFormation stacks. The second part of the lab will walk users on how to deploy additional components, using a CloudFormation template, to assist with data protection in the AWS environment.

Prerequisites – Completion of Lab 1. A Remote Desktop application to connect to a Windows Server.

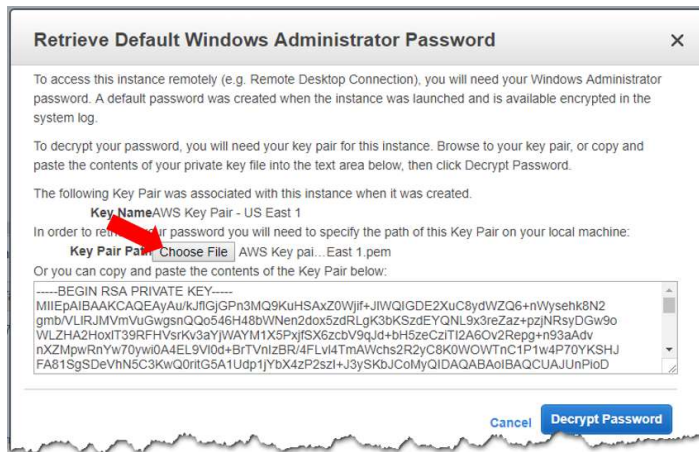
1. Now that you've deployed AWS resources let's try to make use of them. Navigate to the EC2 console by **Services** and selecting **EC2**. Click **Instances** on the left.



2. Prior to connecting to the instance, you need to retrieve the local administrator password. To do this, you need to decrypt it using the EC2 Key Pair that was created earlier. Select **Windows Instance – Lab 2** instance in the console, click **Actions**, and select **Get Windows Password**.



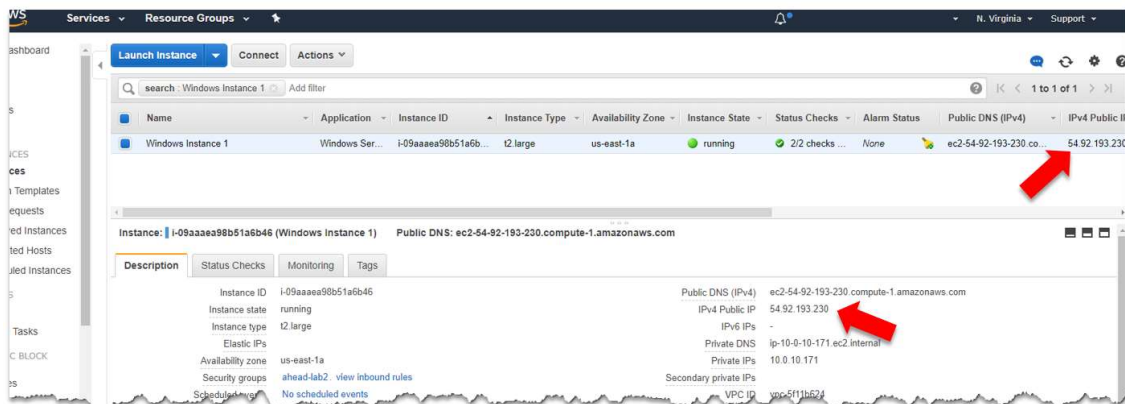
3. Select the **Choose File** button and **select the Private Key** that was downloaded in Lab 1. Alternatively, you can paste the private key into the text box. **Click Decrypt Password.**



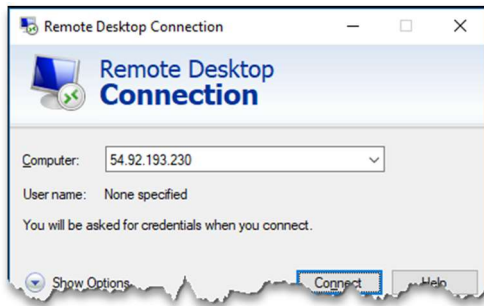
4. Copy the password from the box and click Close.



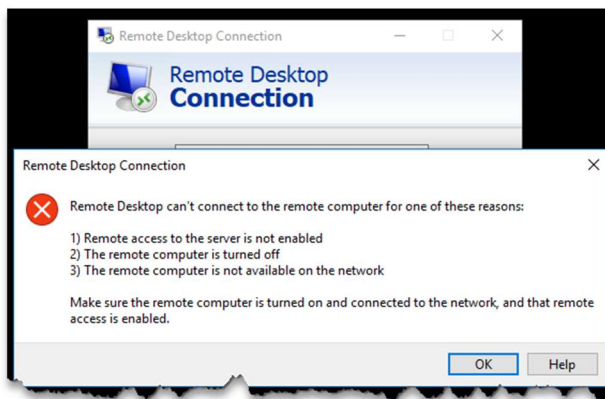
5. In the EC2 console, select the server named **Windows Instance – Lab 2** and note the public IP address.



6. As this is a Microsoft Windows 2016 Server instance, let's try to connect to it using Remote Desktop Connection (RDP). **Enter your public IP address** in the Computer field and **click the Connect button**:



7. Can't connect? Perhaps the instance isn't deployed correctly, or there is a mistake in the network configuration. This is by design for the lab, so let's take a look at the resources deployed so you can fix this problem and manage the new instance.



8. Back in the EC2 console, **look at the information** on the EC2 console and see what's wrong:

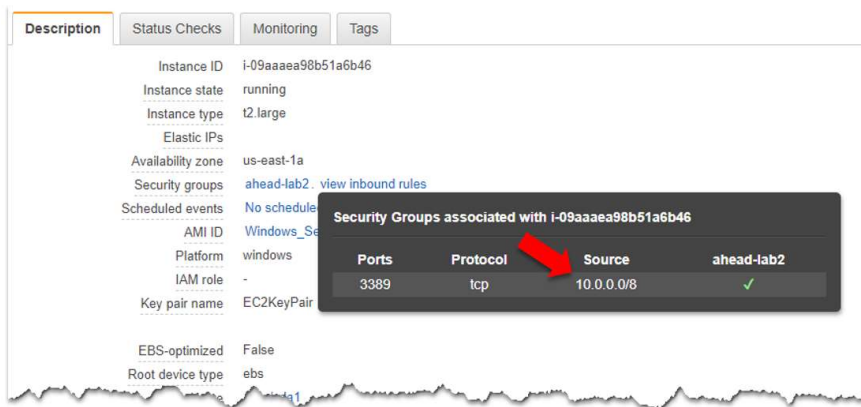
- Is the instance running as indicated by the **Instance State**? **Yes**.
- Does the console indicate that both **Status Checks** have passed? **Yes**.

Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
t2.medium	us-east-1a	running	✓ 2/2 checks ...	None
t2.medium	us-east-1a	running	✓ 2/2 checks ...	None
t2.medium	us-east-1a	running	✓ 2/2 checks ...	None
t2.medium	us-east-1c	running	✓ 2/2 checks ...	None

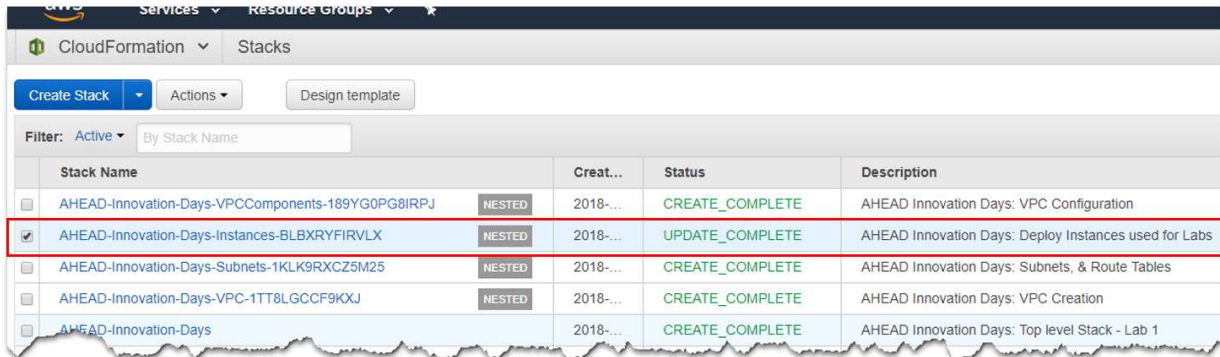
Hmm...the problem might not be the instance configuration then, so let's look at the network:

- Did you deploy an **Internet Gateway** for the VPC? **Yes**
- Is **routing** probably configured for internet access? **Yes**
- Are you allowing **tcp/3389** from the public IP address? **Nope**

Click on **view inbound rules** to see the current rules being applied. It looks like the **instances.yml** template successfully created the security group and the proper inbound rule, but it's only allowing RDP from private IP addresses in the **10.0.0.0/16** IP space. As you're trying to connect over the public Internet, you need to allow traffic from your **Public IP address**. While you could quickly fix this in the console, you're going to continue using CloudFormation to manage the existing lab environment to ensure consistency within the environment.

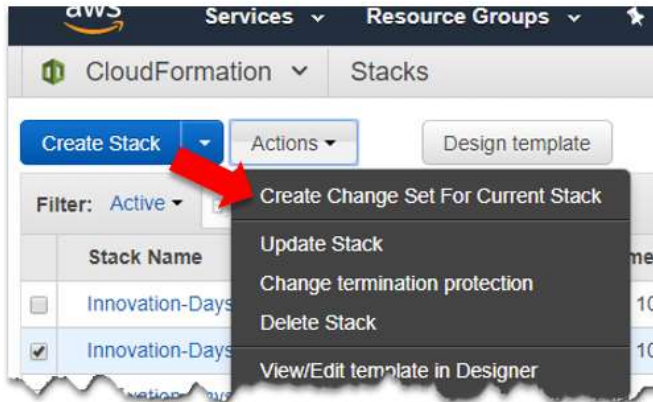


- Click on the Services menu and navigate back to the CloudFormation console. Find the Nested Stack that deployed the Security Group and instances. You can find it by the name **AHEAD-Innovation-Days-Instances-xxxxxx** or by looking at the Description in the list:

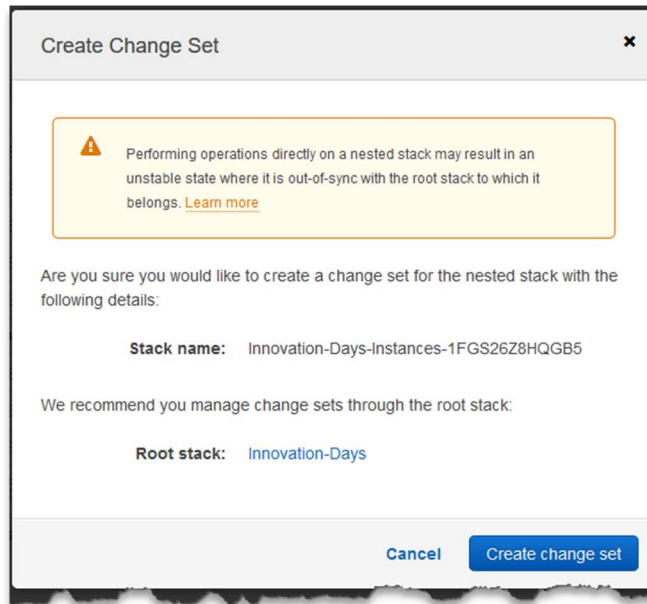


Note: Feel free to view the actual template to see the underlying code. You can view the code by clicking the stack name and expanding the “Template” subsection. Look for the “Resources” section and view the configuration for the Security Group – see where it defined the 10.0.0.0/16 network for the security group.

- You're going to create a Change Set to modify this nested stack so you can ensure that the security group is configured correctly. With the **Instances** nested stack selected, click **Create Change Set For Current Stack** from the Actions menu:



11. Read the warning that is presented and select **Create change set**:



Note: For the sake of simplicity, you are creating a change set on a child stack in the lab. In a production environment, you'll want to create the change set on the parent stack vs. a child stack to ensure the root stack is "in-sync" with all the child stacks.

12. Similar to deploying the original stack earlier, you need to provide the URL where the updated template is stored. Use the following for the **Specify an Amazon S3 template URL** and click **Next**:

https://s3.us-east-2.amazonaws.com/ahead-innovation-days/aws201/lab2/instances_v2.yml

Create change set for Innovation-Days-Instances-1FGS26Z8HQGB5 stack

Select Template

Specify Details

Options

Review

See your changes before updating your stack

Create a change set to see the changes CloudFormation will make to your stack based on the information that you submitted.

After reviewing the changes, you can execute the change set to apply the changes to your stack. [Learn more.](#)

Select Template

To create a change set, provide a template that specifies the changes for the resources and properties that you want to update your stack with. [Learn more.](#)

Choose a template

A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties. [Learn more.](#)

☐ Use current template

☐ Upload a template to Amazon S3

No file selected.

☒ Specify an Amazon S3 template URL

[View/Edit template in Designer](#)

13. Enter a change set name **Update-Security-Group** and a Description **Allow RDP to Windows Instances**. Do not change any of the Parameter fields. Click **Next**.

Create change set for Innovation-Days-Instances-1FGS26Z8HQGB5 stack

Select Template

Specify Details

Options

Review

Specify parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Specify a change set name, description, and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Change set name:

Description:

Parameters

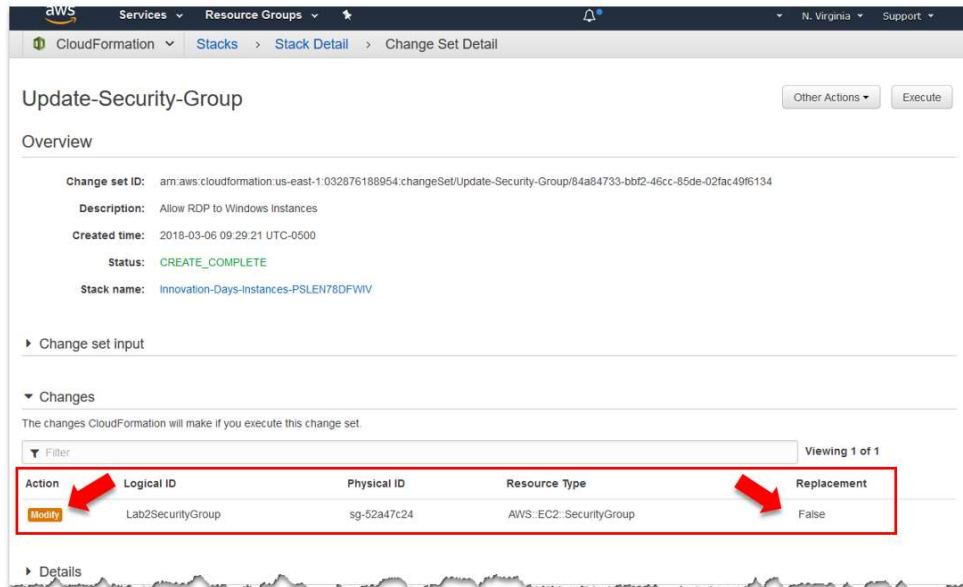
EC2KeyPair:

PublicSubnetAZ1:

PublicSubnetAZ2:

VpcId:

14. On the Options page, leave all the defaults and click **Next**.
15. Review the changes to ensure accuracy. If everything looks correct, click the **acknowledgement box** for the creation of IAM roles. Click **Create change set**.
16. Once complete, you'll be on the **Change Set Detail** page which displays information about the change set you just created. The important subsection (highlighted below) is the **Changes** section, which will list the changes that will be applied to your environment. In this case, you're modifying the Security Group to allow RDP from all IP addresses (vs. just 10.0.0.0/8). Note the action for the resources is **Modify** and that it does NOT require a resource replacement as indicated.



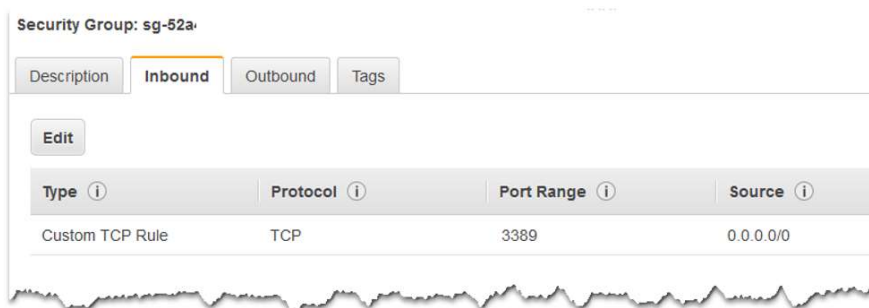
17. Now that you're aware of the impending changes to your environment let's execute the change set. Click the **Execute** button to make the desired changes. Verify you want to execute the changes by clicking **Execute** again.

You'll be redirected back to the CloudFormation console. Take a look at the **Instances** stack and notice the status is now **UPDATE_IN_PROGRESS**.

	Stack Name	Created Time	Status	Description
<input type="checkbox"/>	Innovation-Days-VPCCompon... NESTED	2018-03-06 09:26:36 UTC-0500	CREATE_COMPLETE	AHEAD Innovation D
<input checked="" type="checkbox"/>	Innovation-Days-Instances-PS... NESTED	2018-03-06 09:26:36 UTC-0500	UPDATE_IN_PROGRESS	AHEAD Innovation D
<input type="checkbox"/>	Innovation-Days-Subnets-1Q8... NESTED	2018-03-06 09:25:45 UTC-0500	CREATE_COMPLETE	AHEAD Innovation D
<input type="checkbox"/>	Innovation-Days-VPC-M38GE... NESTED	2018-03-06 09:25:20 UTC-0500	CREATE_COMPLETE	AHEAD Innovation D

18. Once the status changes to **UPDATE_COMPLETE**, you can check to see if you can connect to your EC2 instance via RDP. If you can connect, you've successfully executed a change set to modify your existing infrastructure to manage your EC2 instances.

Note: Feel free to navigate to the VPC or EC2 console to view the Security Group and verify the Inbound rules now allow port 3389 from all IP addresses rather than just 10.0.0.0/8.

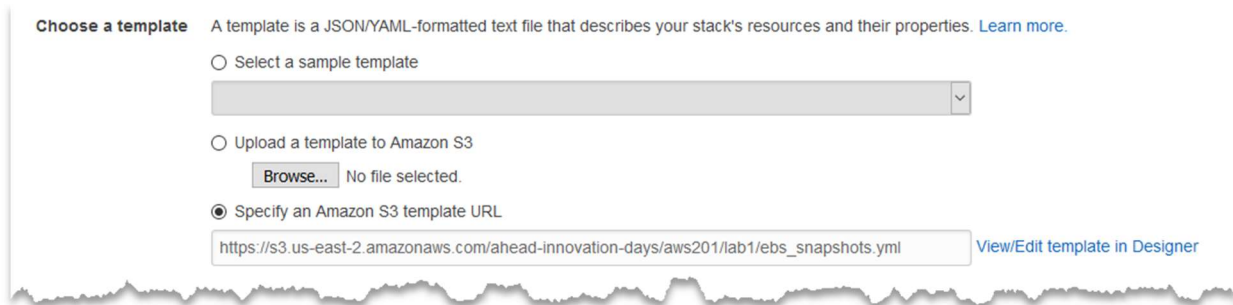


You've completed this part of the lab.

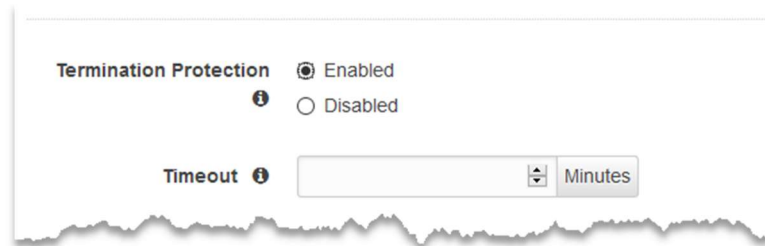
Feel free to continue to part 2 on the next page.

19. Navigate back to the **CloudFormation** console.
20. Click on the **Create Stack** button.
21. Select the radio button next to **Specify an Amazon S3 template URL** and enter the following URL:

https://s3.us-east-2.amazonaws.com/ahead-innovation-days/aws201/lab2/ebs_snapshots.yml

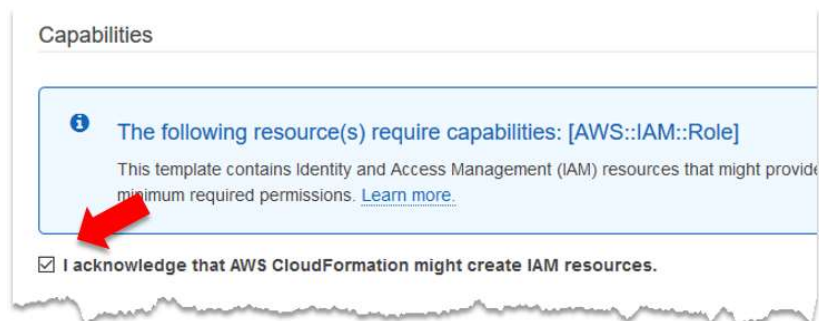


22. Enter **EBS-Snapshots** for the stack name and click **Next**
23. On the Options page, expand the **Advanced** section and select **Enabled** for Termination Protection. Click the **Next** button.



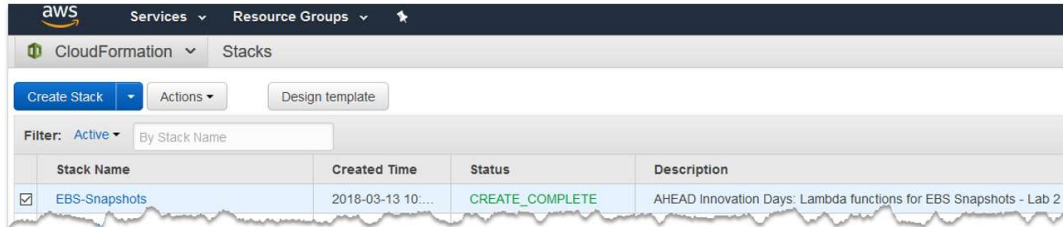
Note: Termination protection will prevent a stack from accidentally being deleted. This setting helps ensure that critical/important stacks aren't deleted without additional intervention. It's a good idea to enable this on stacks that are creating your base infrastructure or provision any other critical resources throughout your AWS deployment.

24. **Check the box** to acknowledge that AWS CloudFormation might create IAM resources. As this template will create an IAM role, the stack will not deploy resources correctly without this acknowledgment.



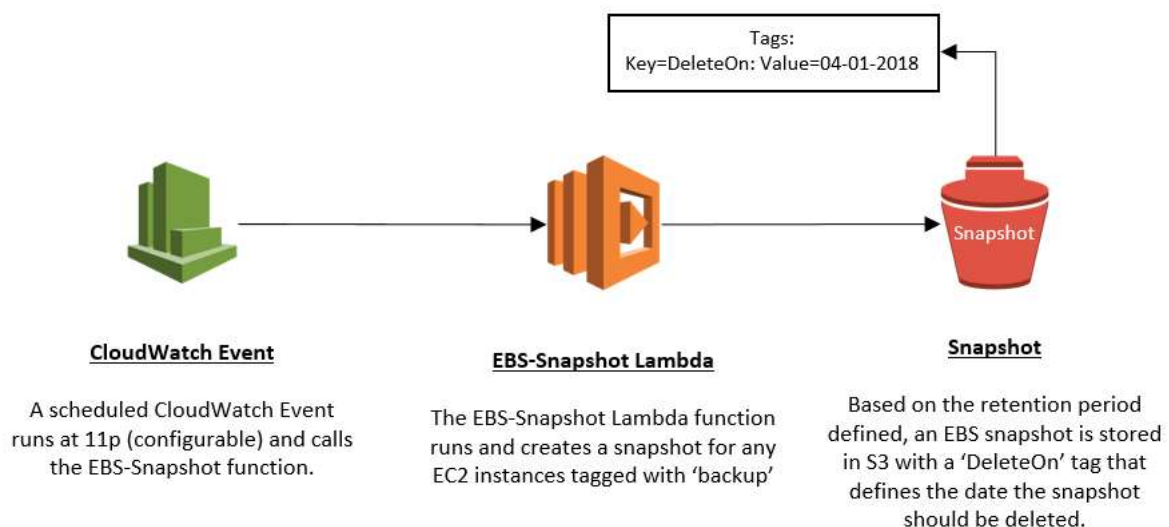
25. Click Create.

26. The stack will create an **IAM role** and **two (2) Lambda functions** that will be used to assist with creating a simple data protection strategy for the instances.



Prior to making use of the Lambda functions that were deployed, let's take a look at what they do:

EBS-Snapshot>Create-Function: This function was developed to take scheduled snapshots of EBS volumes attached to EC2 instances. When executed, the function scans EC2 instances and looks for a Tag of "Backup/backup." When it finds an instance with that Tag, it grabs the value of the "Retention" tag, if it exists. With that data, the function initiates a snapshot of the EBS volume(s) attached to the instance. The Lambda function tags the snapshot with a "DeleteOn" tag key and calculates the date for deletion based on the Retention value. The function uses seven (7) days as the default value if no Retention value is configured.



EBS-Snapshot-Delete-Function: This function was developed to delete expired snapshots to eliminate the cost associated with unneeded snapshots. When executed, the function scans all the snapshots in your account and looks for a Tag of 'DeleteOn.' If the 'DeleteOn' date matches the date the Lambda function is executed on, the function deletes the snapshot.



27. Browse to the **EC2** console and select the **Windows Instance – Lab 2** instance. Click on the Actions menu and select **Instance Settings → Add/Edit Tags**.
28. **Create two new tags** as follows:
 - **Key:** backup
Value: <leave blank>
 - **Key:** Retention
Value: 30
29. **Click Save.**

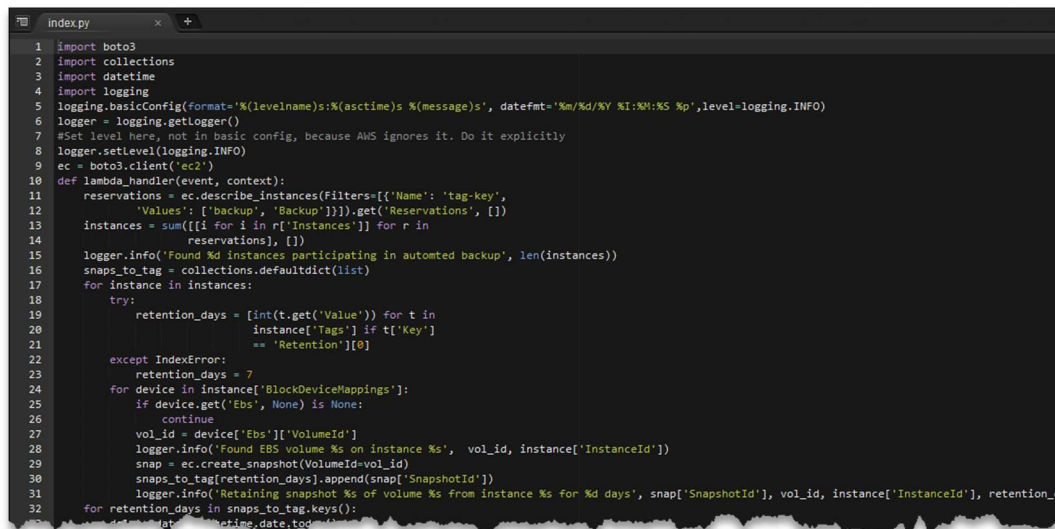
The screenshot shows the 'Add/Edit Tags' dialog box in the AWS Management Console. It has a title bar with 'Add/Edit Tags' and a close button. The main content area contains instructions: 'Apply tags to your resources to help organize and identify them.' and 'A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.' Below this is a table with two columns: 'Key' and 'Value'. The first row has 'Retention' in the Key column and '30' in the Value column. The second row has 'backup' in the Key column and an empty Value field. To the right of each row is a 'Show Column' link with a close icon. At the bottom left is a 'Create Tag' button, and at the bottom right are 'Cancel' and 'Save' buttons.

Key	Value	
Retention	30	ⓧ Show Column
backup		ⓧ Show Column

Create Tag Cancel Save

30. While still in the EC2 console, click on **Snapshots** in the navigation pane and notice there are currently no snapshots for our lab instances.

31. Navigate to the **Lambda** console under the Compute section.
32. Click the function with the name of **EBS-Snapshot-Create-Function** to open the properties of this function.
33. Scroll down to **view the code** of the Lambda function - don't worry, you don't need to be able to read it. This Lambda function is written in Python 2.7.

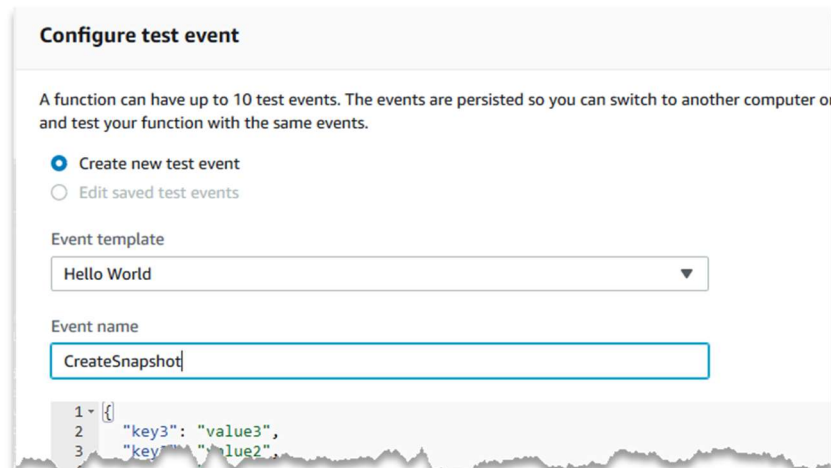


```
1 import boto3
2 import collections
3 import datetime
4 import logging
5 logging.basicConfig(format='%(levelname)s: %(asctime)s %(message)s', datefmt='%m/%d/%Y %I:%M:%S %p', level=logging.INFO)
6 logger = logging.getLogger()
7 #Set level here, not in basic config, because AWS ignores it. Do it explicitly
8 logger.setLevel(logging.INFO)
9 ec = boto3.client('ec2')
10 def lambda_handler(event, context):
11     reservations = ec.describe_instances(filters=[{'Name': 'tag-key',
12     'Values': ['backup', 'Backup']}] ).get('Reservations', [])
13     instances = sum([i for i in r['Instances'] for r in
14     reservations], [])
15     logger.info('Found %d instances participating in automated backup', len(instances))
16     snaps_to_tag = collections.defaultdict(list)
17     for instance in instances:
18         try:
19             retention_days = [int(t.get('Value')) for t in
20             instance['Tags'] if t['Key']
21             == 'Retention'][0]
22         except IndexError:
23             retention_days = 7
24         for device in instance['BlockDeviceMappings']:
25             if device.get('Ebs', None) is None:
26                 continue
27             vol_id = device['Ebs']['VolumeId']
28             logger.info('Found EBS volume %s on instance %s', vol_id, instance['InstanceId'])
29             snap = ec.create_snapshot(VolumeId=vol_id)
30             snaps_to_tag[retention_days].append(snap['SnapshotId'])
31             logger.info('Retaining snapshot %s of volume %s from instance %s for %d days', snap['SnapshotId'], vol_id, instance['InstanceId'], retention_d
32     for retention_days in snaps_to_tag.keys():
33         for snap_id in snaps_to_tag[retention_days]:
34             tag = {'Key': 'Retention', 'Value': retention_days}
35             ec.create_tags(Resources=[snap_id], Tags=[tag])
```

34. At the top of the page, click the **Test** button.

Note: The test button will allow us to manually execute the Lambda function without requiring additional AWS components, such as Config or CloudWatch.

35. Enter **CreateSnapshot** as the Event name and leave all the defaults. **Click Create**.



Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or and test your function with the same events.

☒ Create new test event
☐ Edit saved test events

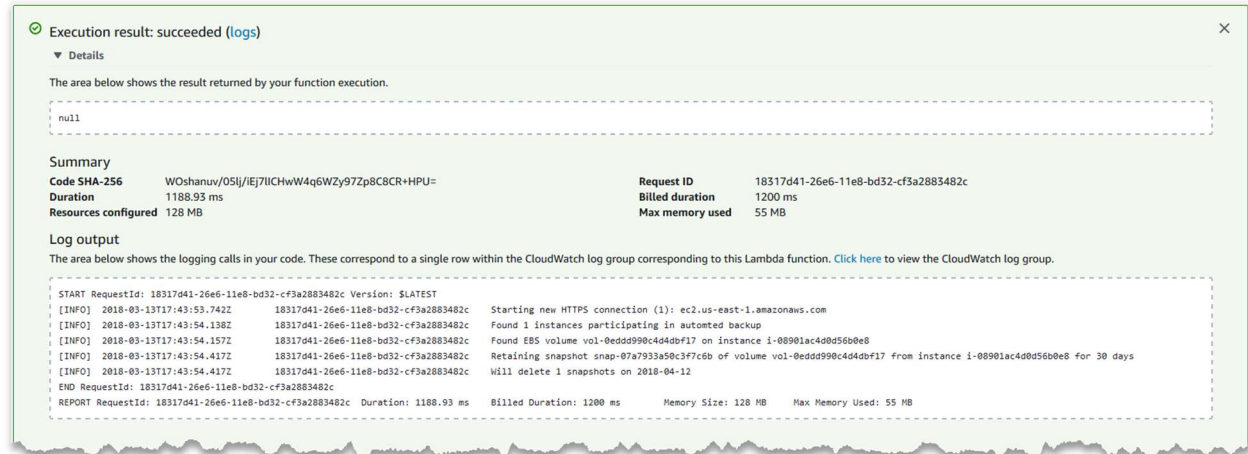
Event template
Hello World ▼

Event name
CreateSnapshot

```
1 {
2   "key3": "value3",
3   "key": "value2"
}
```

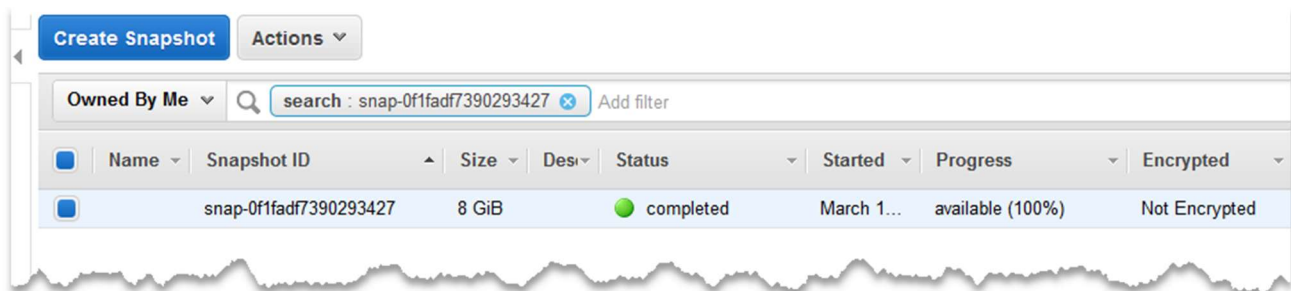
Note: For this particular Lambda function, you don't need to pass it any information to work correctly (because it's going out to look for Tags) therefore the JSON code is irrelevant for this test event.

36. You're ready to execute this Lambda function. Ensure the **CreateSnapshot** test event is selected on the drop-down and **click the Test button**.
37. Did it run successfully? Take a look at the logs – **expand the Details** for the Execution result.

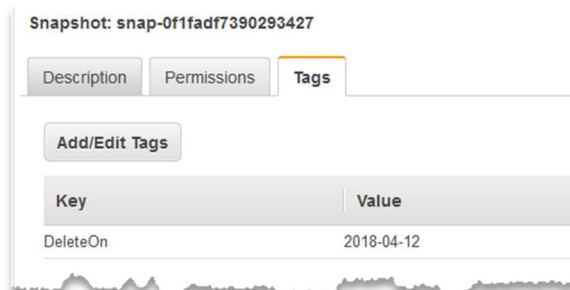


Notice that the Lambda found one (1) instance participating in the automated backup. Secondly, it found the EBS volume attached to the instance and created a snapshot of the volume. Finally, the log output states that the snapshot will be deleted thirty (30) days from now (the value for the Retention tag).

38. Browse to the EC2 console and click on **Snapshots**. You should now have a snapshot listed that was taken of the Windows instance. If you already have snapshots in your account, sort the list by the **Started** column to quickly find the latest snapshot taken.



39. Click on the **Tags** tab in the bottom pane. Notice the **DeleteOn** key with a value that equals 30 days from today's date. That tag is what the EBS-Snapshot-Delete-Function will be looking for when it is executed.



40. Navigate back to the **Lambda** console. **Create a test event** for the **EBS-Snapshot-Delete-Function** Lambda function just like you did for the EBS-Snapshot-Delete-Function in **Step 35**.
41. Execute it by clicking the **Test** button.
42. Navigate back to the **EC2** console and check to see if it deleted the snapshot.

It didn't delete it. While the function executed correctly, it didn't find any snapshots to delete. If you recall, the 'EBS-Snapshot-Delete-Function' function looks for the value of the date in the **DeleteOn** tag, and if it matches today's date, it will delete it.

43. While still in the EC2 console, click the snapshot, and select **Add/Edit Tags** from the Action menu.
44. **Modify the Value** of the tag to match today's date. Click **Save**.

Add/Edit Tags

Apply tags to your resources to help organize and identify them.

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.

Key	Value
DeleteOn	2018-03-13

[Show Column](#)

Create Tag **Cancel** **Save**

45. Navigate back to the Lambda console, open the properties of the **EBS-Snapshot-Delete-Function** function and **execute** it again. When successful, **expand the Details** and look at the logs.

Summary

Code SHA-256	ZyqgZJEGt9hcelJZISC6VgxvO442DFrEiusK/dYozl=	Request ID	69eb5a63-26e8-11e8-bff9-6ba9d63fd945
Duration	625.88 ms	Billed duration	700 ms
Resources configured	128 MB	Max memory used	51 MB

Log output

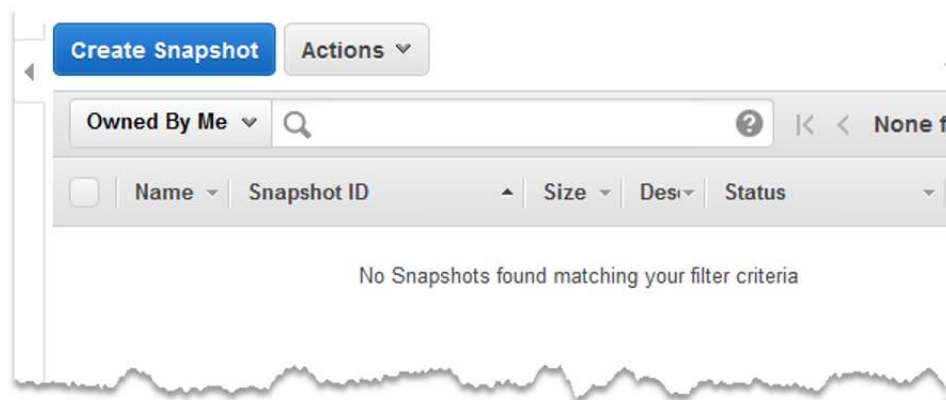
The area below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. [Click here](#) to view the CloudWatch log group.

```

START RequestId: 69eb5a63-26e8-11e8-bff9-6ba9d63fd945 Version: $LATEST
[INFO] 2018-03-13T18:00:29.123Z 69eb5a63-26e8-11e8-bff9-6ba9d63fd945 Starting new HTTPS connection (1): sts.amazonaws.com
[INFO] 2018-03-13T18:00:29.286Z 69eb5a63-26e8-11e8-bff9-6ba9d63fd945 Found account id [REDACTED] from the runtime environment
[INFO] 2018-03-13T18:00:29.299Z 69eb5a63-26e8-11e8-bff9-6ba9d63fd945 Starting new HTTPS connection (1): ec2.us-east-1.amazonaws.com
[INFO] 2018-03-13T18:00:29.502Z 69eb5a63-26e8-11e8-bff9-6ba9d63fd945 Deleting snapshot snap-0f1fadf7390293427 from S3 permanently
END RequestId: 69eb5a63-26e8-11e8-bff9-6ba9d63fd945
REPORT RequestId: 69eb5a63-26e8-11e8-bff9-6ba9d63fd945 Duration: 625.88 ms Billed Duration: 700 ms Memory Size: 128 MB Max Memory Used: 51 MB
  
```

This time it found a snapshot to delete, as the value of the tag matched today's date.

46. Navigate back to the **EC2** console and verify that the snapshot has been deleted.



That's it. You've completed this lab.

Conclusion: In this lab, participants deployed a CloudFormation change set to modify the existing AWS infrastructure using code. Then you deployed two Lambda functions to help form a data protection strategy for the EC2 instances.

Note: You can delete the stack if you wish, although nothing in the lab will incur costs if you've deleted the snapshot.