At-home coding challenge

Goals:

Demonstrates a notification of file upload to a designated bucket/prefix (e.g. /Transfer)

Demonstrates a notification on the object download from a designated bucket

Implemented a virus scanner

Installs in any account using a recognized IaC facility (CloudFormation)

Includes a self-contained test capability

Commit to Github w/ a REAMDE

Deliverable:

Lambda – build/s3-code.py, build/utils.py

Lambda – cfnhelper.zip

CFT Template – s3-deploy.yaml

Driver Script – demo.sh

Demo Driver script

The driver script consists of the following operations:

1. Unit tests – placeholder
    a. Use pytest (etc.) to validate specific function behavior (not terribly interesting)
2. Functional tests
    a. Run the lambda code as a standalone utility and verify its behavior passing it simulated events. It won't have virus scanning, but the rest of the logic can be tested. In this mode, it will return the json as stdout (in lieu of sending notifications)
    b. There are several events supplied in the test-input folder than can be fed one at a time or in bulk to the lambda
3. Build artifacts
    a. Creates the lambda and uploads artifacts to the deploy bucket s3://eft-distro-east-1/
4. Deploy using CFT
    a. The script drives cloudformation
5. Mail subscription configuration, Actual Testing
6. Teardown

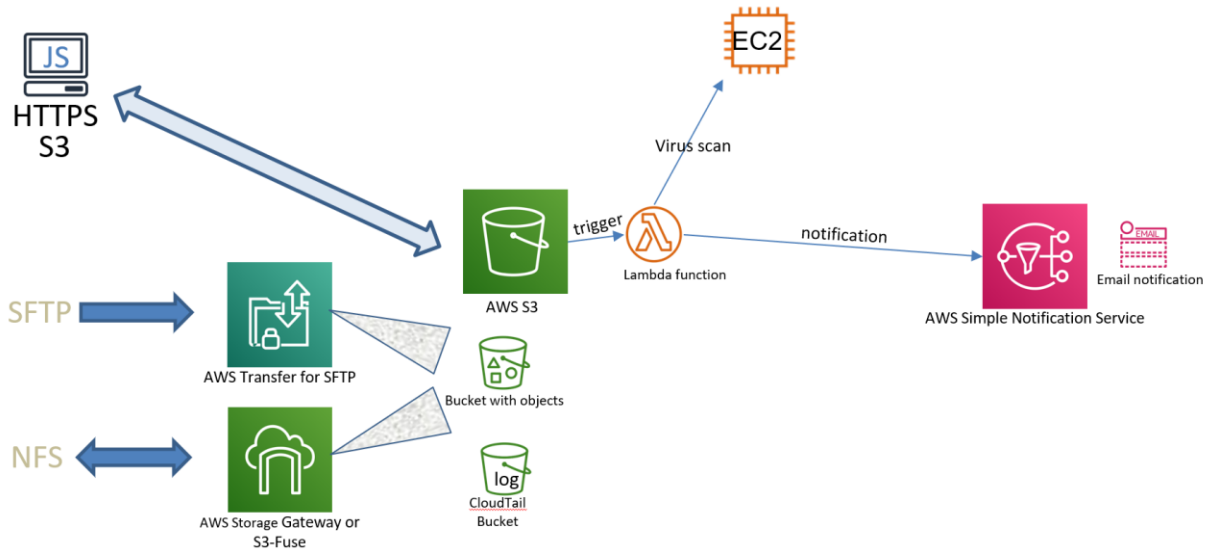
Customization

Not all of the parameters have been specified but can easily be added.

Three of the parameters are account-specific. These are only required when using the Virus Scanner as it needs to place an EC2 and configure the networking between the Lambda (not running in a VPC) and a publicly accessible EC2 scanner.

See the section under Virus Scanner to see the list of configurable parameters.

Architecture



This Python lambda is configured to listen to Object:Creation events on 2 buckets;

demo-cloudtrail-private4

demo-transfer-private4

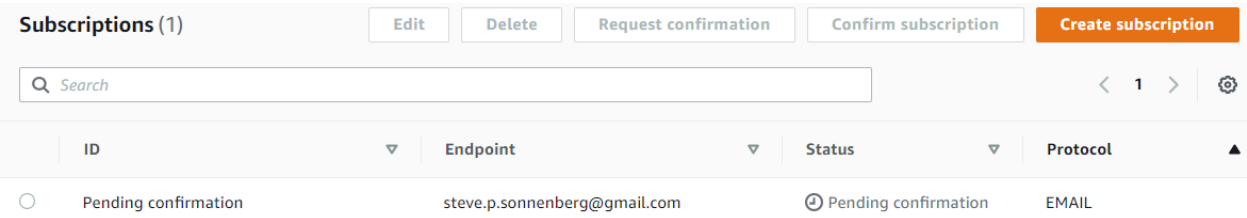
eft-distro-east-1

The 3rd bucket is used to hold the artifacts and CFT template.

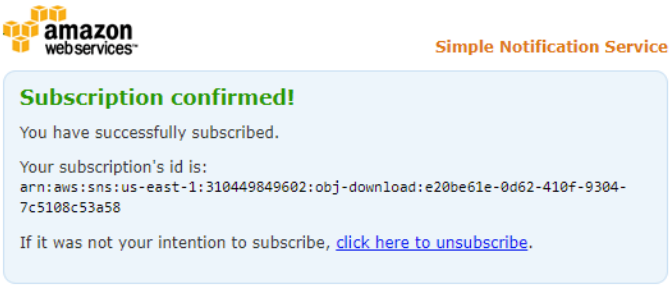
The demo-transfer-private4 bucket is used for uploading and downloading objects. Both upload and download operations trigger email notifications using SMS.

The cloudtrail bucket is used by Cloudtrail 'Data' events to record batches of read activity which once posted triggers the read-completed event.

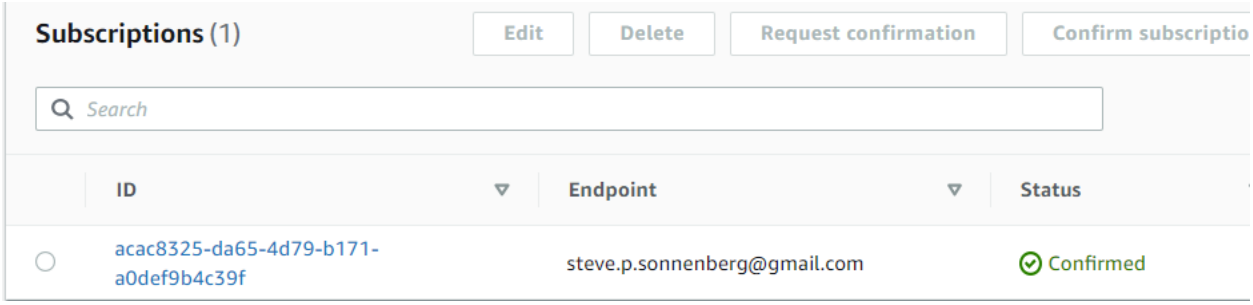
The two topics are: obj-upload and obj-download. Email won't be delivered until the recipient opts into the subscription.



After deployment, the email address will be sent 2 'Subscription Confirmation' messages. Selecting the link will generate a confirmation such as:



After confirmation the topics will show a confirmation status.

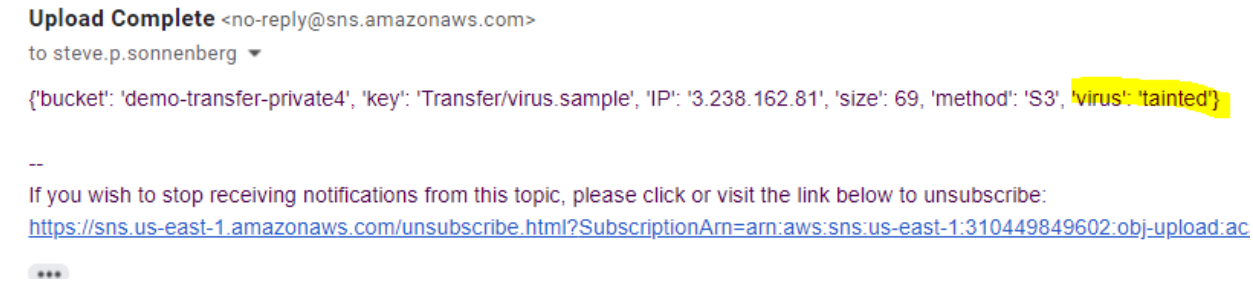


Uploading of files can be performed using the Console, AWS CLI, boto3, REST, AWS FileTransfer, s3fuse, etc. Upon receipt of an incoming file the lambda will download the object and submit to the virus-scanner (CLAM AV). If the scan is clean, it will generate a pre-signed URL and send the email notification.

Below is an example:

The email consists of JSON-formatted file attributes indicating properties of the file, how it was uploaded, the 'virus' state ('clean'), and a URL to use for download (presigned).

If the file is not clean, the virus-state will show 'tainted' and no url will be provided.



Below is a snippet from Cloudwatch logs:

```
START RequestId: 39ae2e7b-7955-476d-86df-c50d37f431a7 Version: $LATEST
[INFO] 2021-08-26T12:18:24.966Z Found credentials in environment variables.
[INFO] 2021-08-26T12:18:25.054Z 39ae2e7b-7955-476d-86df-c50d37f431a7 {"Records": [{"eventVersion": "2.1", "eventSource": "aws:s3", "
[INFO] 2021-08-26T12:18:25.054Z 39ae2e7b-7955-476d-86df-c50d37f431a7 demo-transfer-private4:Transfer/virus.sample file-upload
[INFO] 2021-08-26T12:18:25.694Z 39ae2e7b-7955-476d-86df-c50d37f431a7 retrieving object to submit for scan
[INFO] 2021-08-26T12:18:26.336Z 39ae2e7b-7955-476d-86df-c50d37f431a7 scan state tainted
[INFO] 2021-08-26T12:18:26.354Z 39ae2e7b-7955-476d-86df-c50d37f431a7 Found credentials in environment variables.
[INFO] 2021-08-26T12:18:26.677Z 39ae2e7b-7955-476d-86df-c50d37f431a7 Publish to arn:aws:sns:us-east-1:310449849602:obj-upload
[INFO] 2021-08-26T12:18:26.920Z 39ae2e7b-7955-476d-86df-c50d37f431a7 completed record
END RequestId: 39ae2e7b-7955-476d-86df-c50d37f431a7
```

Downloading files will not be detected until Cloudtrail posts the audit activity which may be delayed by 5-15 min.

Download Complete <no-reply@sns.amazonaws.com> 8:23 AM (9 minutes ago)

to me

{'IP': '3.238.162.81', 'method': 'aws-cli/1.18.147', 'size': 1831, 'bucket': 'demo-transfer-private4', 'key': 'Transfer/event2.json', 'time': '2021-08-26T12:20:46Z', 'recipient': 'arn:aws:iam::310449849602:user/steve'}

Virus Scanning

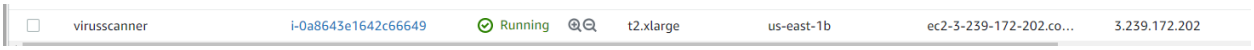
Virus scanning is performed by an EC2 running ClamAV. This is an optional component which can be configured using the 'CreateServer' parameter shown below.

Parameters (11)

Search parameters

Key	Value
AWSLinuxAmiId	ami-0947d2ba12ee1ff75
BucketName	demo-transfer-
CloudtrailBucketName	demo-cloudtrail-
CreateServer	true
DistroBucketName	eft-distro-east-1
DistroPrefix	private4
KeypairName	2020
NotificationEmail	steve.p.sonnenberg@gmail.com
PublicSubnet	subnet-2e4a6949
VirusHost	0.0.0.0
VpcId	vpc-e12a559b

The virus scanner works best when all of its signatures are loaded in memory. The lambda uses clamdscan to submit the object to the server for validation. In the future this could auto-hibernate when idle to reduce the cost of operation.



Here is the 'userdata' used to initialize the virus scanner.

```
  UserData:
    Fn::Base64: !Sub |
      #!/bin/bash -xe
      HOMEDIR=/home/ec2-user

      yum update -y

      #amazon-linux-extras install ..
      yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

      echo Installing packages...
      yum install -y clamav clamav-server clamav-update clamav-data clamav-scanner-systemd clamav-s
-systemd

      # configure virus scanner
      sed -i -e "s/^Example/#Example/" -e "s/^#TCPSocket/TCPSocket/" /etc/clamd.d/scan.conf
      freshclam
      sed -e "s/%i/scan/g" /usr/lib/systemd/system/clamd@.service > /usr/lib/systemd/system/clamd.s
e
      systemctl enable /usr/lib/systemd/system/clamd.service
      systemctl start clamd.service

      netstat -tanp | grep LISTEN

      /opt/aws/bin/cfn-signal \
        -e $? \
        --stack ${AWS::StackName} \
        --resource EC2ScannerInstance \
        --region ${AWS::Region}
```

CFN Helper

A CloudFormation helper (custom resource) is part of the deployment (not coded during this exercise) to overcome some CFN limitations such as:

- There is no way to create a directory (e.g. Transfer) using CF
- There is no way to set the 'configuration-id' of an S3 notification using CF (e.g. file-upload)
- There is no way to empty a bucket during termination/cleanup
- Etc.