Note: All these scripts are on our bluemoon load-generator server – Ohio region

SSH to 18.219.66.236, login as EC2 user and cd py\_scripts/cust\_ebs

Git: <https://github.com/cdw-msp/Customer-EBS.git>

1. Ebs\_creation\_encryption. Py

* This file contains the python script that creates a new EBS volume
* For this script, parameters required to create the new EBS volume are static (hardcoded).
* Parameters needed are:

1. Availability Zone: availability zone in which to create the EBS volume
2. KSM KeyID: KMS key with which you want to encrypt this volume. It can be provided in any of the following formats such as Key ID, Key alias, ARN
3. Encrypted: Boolean, should be set to true
4. Size: size of the volume in GiBs
5. Volume Type: This can be gp2 for General Purpose SSD, io1 for provisioned IOPS SSD, st1 for throughput optimized HDD, sc1 for Cold HDD, or standard for Magnetic Volumes
6. Tagspecifications: The tags to apply to the volume during creation

* Save this script locally and run the command <python filename.py>

1. Ebs\_creation\_encryptionV2. Py

* This file contains the python code to create a new encrypted EBS volume. Unlike the above code, parameters are passed as the arguments for this script.
* Save this file and run the below command from the command prompt

python <file\_name.py> <availabilityzone> <kms key arn> <size> <volume\_type> <name\_of\_volume>

1. ebs\_creation\_encryptionv3.py

* This file contains the python code to create a new encrypted EBS volume.
* In this version parameters are passed through the keyboard (using raw\_input() function)
* Save this file and run the command <python filename.py>. when prompted pass the values for the parameters

1. Existing\_vol\_encryption.py

* This file contains the python script that creates a snapshot from the existing unencrypted volume, creates an encrypted copy of the snapshot and creates a volume from the copy-snapshot
* For the current version of this script, the parameters required are

1. volumeID: The volume ID of the existing volume from which the snapshot has to be created
2. Tag specifications: tags to apply to the snapshot/copy-snap/volume creation
3. Encrypted: True (this parameter is for copy snap)
4. KmsKeyID: ARN of the KMSKEYID
5. SourceSnapshotID: The snapshot created in step1 should be referenced here (for copy-snap).
6. SourceRegion: Region in which the snapshot is created (this will be the region of the Volume from which the snapshot is created)
7. Availability Zone: availability zone in which to create the EBS volume
8. SnapShotID: This should be the ID of the copy-snap created in step 2
9. Size: size of the volume
10. Volume type: gp2

* Save this script and run the command <python filename.py> to run it.

1. S3\_creation\_encryptionv2.py

* This file contains the python script that creates a new encrypted s3bucket/encrypts the existing bucket.
* If the bucket is already existing, it loops through the objects and checks if the objects are encrypted and if not, it encrypts the objects and bucket
* For the current version of this script, the parameters required are

1. bucket\_name: name that you want to create the bucket/name of the existing bucket that you want to encrypt – passed from the keyboard using raw\_input() function.
2. LocationConstarint: Location/Region that you want to create the bucket. This is not needed if you want to create the bucket us-east-1
3. KMSMasterKeyID/AES256: ARN of the KMS key. You can get this from IAM console

IAM – encryption Keys – click on your key

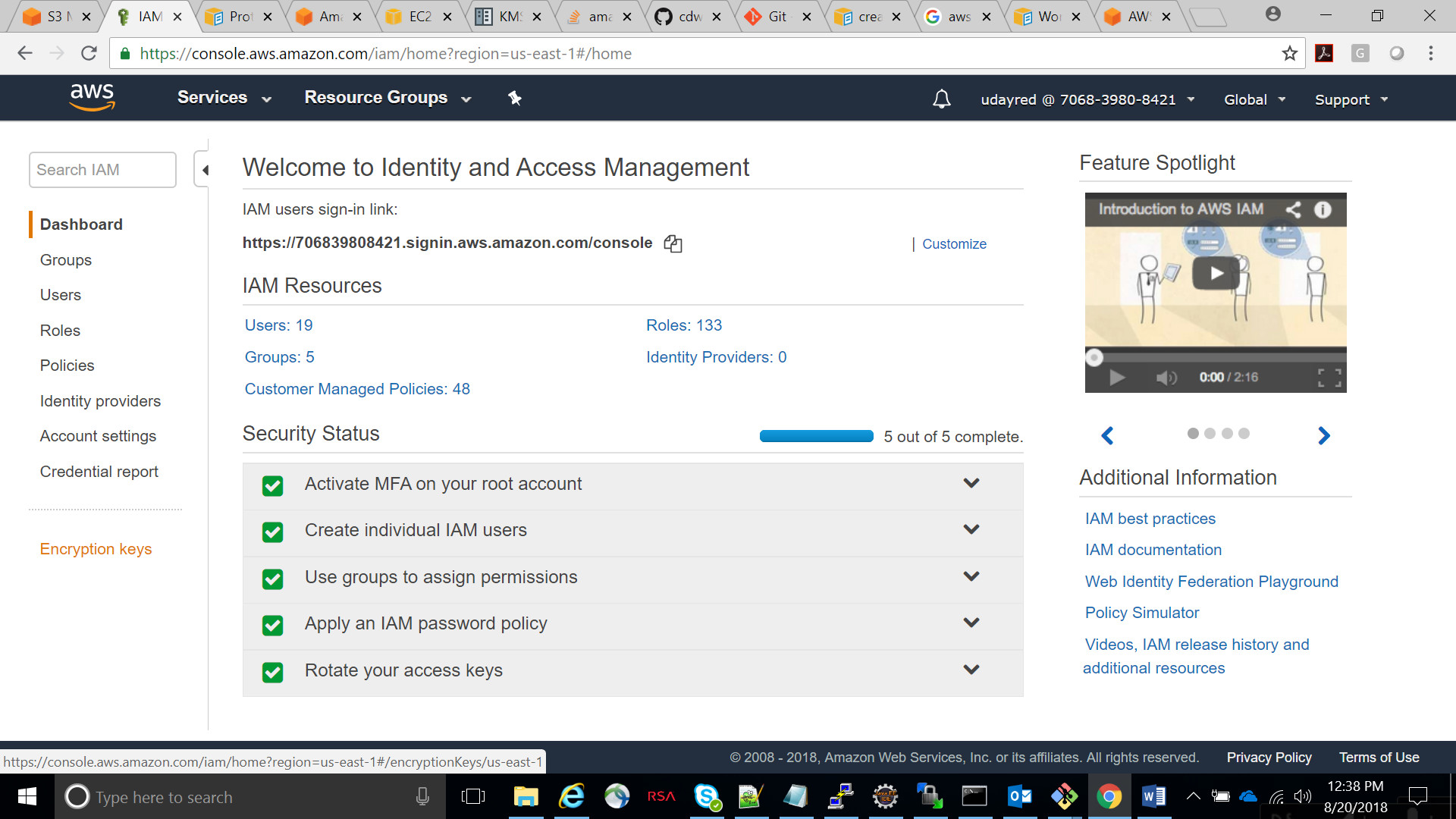
* Save this script and run the command <python filename.py> to run it. When prompted enter the S3 bucket name from the keyboard

**Note:**

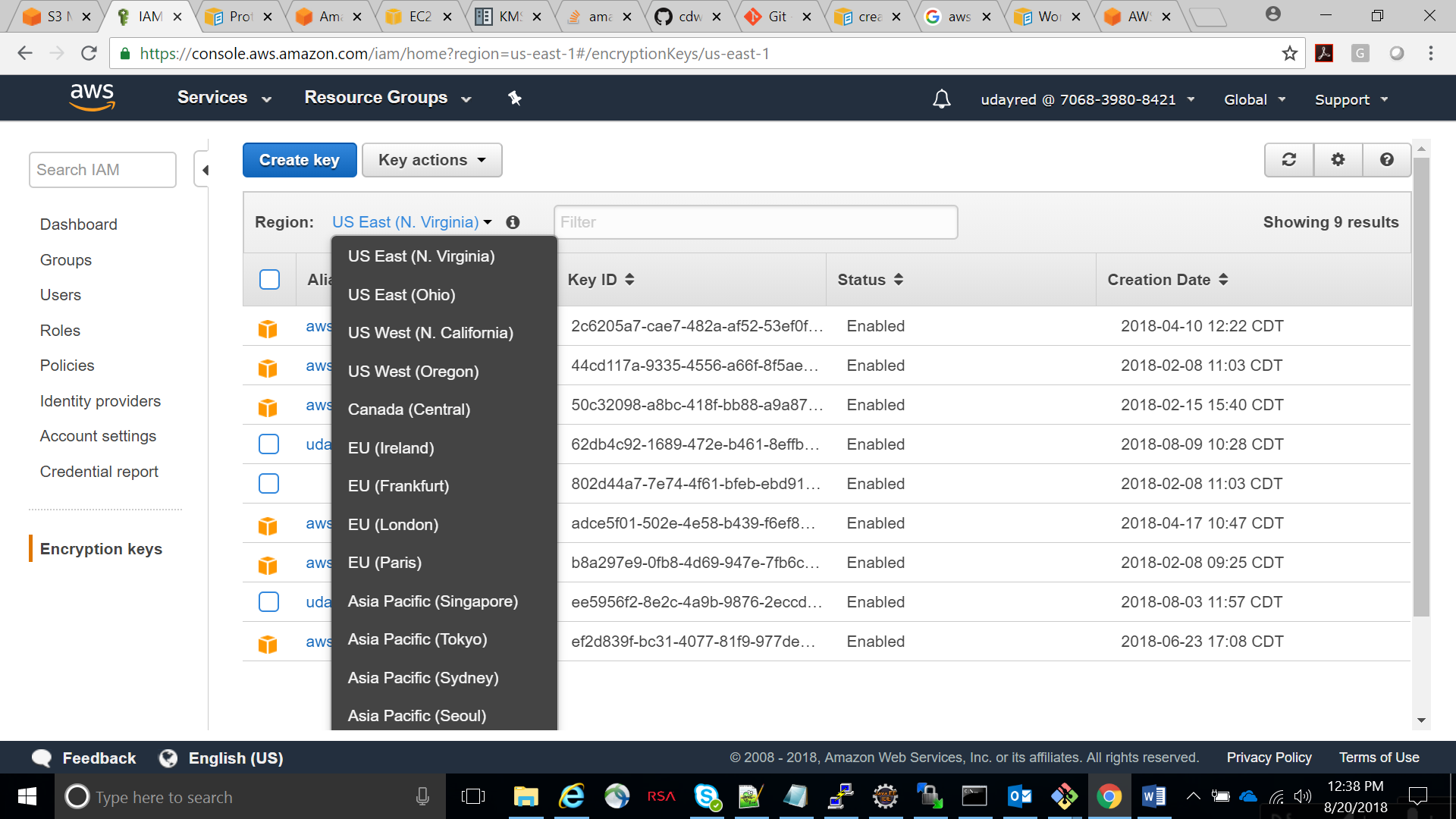
* If you want to encrypt the bucket/objects using the KMS KEYID, you will need to have a valid KMS key ID for each region (ex: us-east-1, us-east-2)
* for example, if you want to create a new bucket which is encrypted using KMS Key ID in us-east-2 region, you should have a valid KMS key in us-east-2 region and make sure you supply this key ARN (which is in us-east-2 region) in the put\_bucket\_encryption part for KMSMasterKeyID (line 38 of this script)
* The bucket gets created successfully even though you use the KMS Key ID of other region, but you will run into issues when you want to upload objects into this bucket.
* In the same way, if you want to check/encrypt the S3 bucket and it’s objects which is already existing, make sure you supply the KMS Key ID of that region in which the bucket is existing, and make sure you supply this KMS Key for objects and bucket encryption – line no 29 and 38 of this script (ex: if the bucket is in us-west-2 region, you should have a valid KMS Key ID in this region and you should use this key to encrypt the bucket and it’s objects). Otherwise, you will get a KMSnotfoundexception- if you try to encrypt using the KMS Key ID of another region.

**KMS Key creation with specific region**:

* From AWS Console, navigate to IAM and click ‘Encryption Keys’



* Next, choose the region that you want your KMS key created and click on ‘create-key’



* Next you can give your key a “Alias” and assign the key administrative permissions to the users so that this key can be used.

6. encrypt\_available\_volumes.py

* This script lists out all the unencrypted volumes which are in “available” status with respect to a region and encrypts them (snapshot/copy-snap/vol).
* Save this file and run the command <python filename.py> to run this script.