

AWS

Task 2 – Using the CLI (Command Line interface)

- 1) Install AWS Command line interface

AWSCLI64.msi

- 2) Confirm successful installation version

```
C:\Users\Administrator>aws --version
aws-cli/1.11.95 Python/2.7.9 Windows/7 botocore/1.5.58
```

- 3) Configuring AWS using command line interface

```
C:\Users\Administrator>aws configure
AWS Access Key ID [None]: AKIAJ2RR5GDQ57W05XUQ
AWS Secret Access Key [None]: h5Y6nSj0BEjGgYQfkJUhhKatBnRtCj03Gk16OfJU
Default region name [None]: eu-west-2
Default output format [None]: json
```

Creating a security group, key pair, and role

Creating a new security group and add a rule that allows incoming traffic over port 22 for SSH

- 4) `aws ec2 create-security-group --group-name securitygroup1h-sg --description "security group for development environment in EC2"`

```
C:\Users\Administrator>aws ec2 create-security-group --group-name securitygroup1h-sg --description "security group for development environment in EC2"
{
  "GroupId": "sg-83adc0ea"
}
```

`aws ec2 authorize-security-group-ingress --group-name securitygroup1h-sg --protocol tcp --port 22 --cidr 0.0.0.0/0`

- 5) Creating key pair that allows you to connect to the instance.

```
C:\Users\Administrator>aws ec2 create-key-pair --key-name devenv-key --query "KeyMaterial" --output text > devenv-key.pem
```

`aws ec2 create-key-pair --key-name louy-key --query "KeyMaterial" --output text > louy-key.pem`

Lheidar

6) Launch and connect to the instance:

```
aws ec2 run-instances --image-id ami-f1d7c395 --security-group-ids sg-83adc0ea --count 1 --instance-type t2.micro --key-name louy-key --query 'Instances[0].InstanceId'
```

"i-ec3e1e2k"

Notes:

Image ID ami-f1d7c395 specifies the Amazon Machine Image (AMI) that Amazon EC2 uses to bootstrap the instance.

Instance launched and running below:

The screenshot shows the AWS Management Console with a list of EC2 instances. The instance 'Academy.HL.CLICongig:0206' is selected, and its details are shown in the right pane. The details include:

- Instance ID: i-0f7e91b9400ab0e89
- Instance state: running
- Instance type: t2.micro
- Elastic IPs: None
- Availability zone: eu-west-2a
- Security groups: securitygroup-sg-83adc0ea
- Scheduled events: No scheduled events
- AMI ID: ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20170414 (ami-f1d7c395)
- Public DNS (IPv4): ec2-35-176-70-105.eu-west-2.compute.amazonaws.com
- IPv4 Public IP: 35.176.70.105
- Private DNS: ip-172-31-15-26.eu-west-2.compute.internal
- Private IPs: 172.31.15.26
- Secondary private IPs: None
- VPC ID: vpc-af59adc6
- Subnet ID: subnet-0e849d76

Associated Instance ID:

Instance: **i-0f7e91b9400ab0e89** (Academy:HL:CLICongig:0206) Public DNS: ec2-35-176-70-105.eu-west-2.compute.amazonaws.com

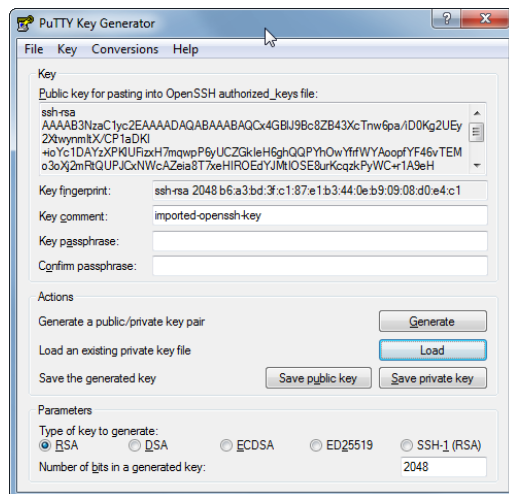
Description	Status Checks	Monitoring	Tags
Instance ID	i-0f7e91b9400ab0e89		
Instance state	running		

Command will retrieve the public IP address that you will use to connect to the instance.

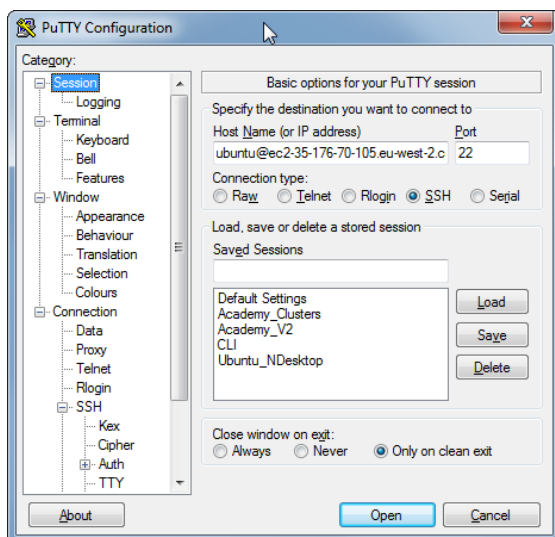
```
C:\Users\Administrator>aws ec2 describe-instances --instance-ids i-0f7e91b9400ab0e89 --query "Reservations[0].Instances[0].PublicIpAddress"
"35.176.70.105"
```

SSH on Windows:

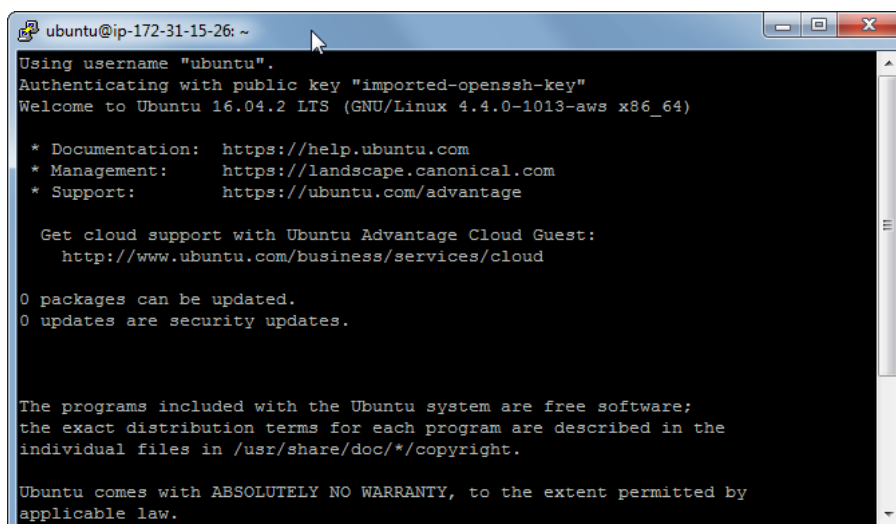
- 1) Load 'louy-key.pem' key to generate the private key



- 2) Configure the connection properties within putty which include adding to newly generated private key



- 3) Successful connection to instance after correct configuration



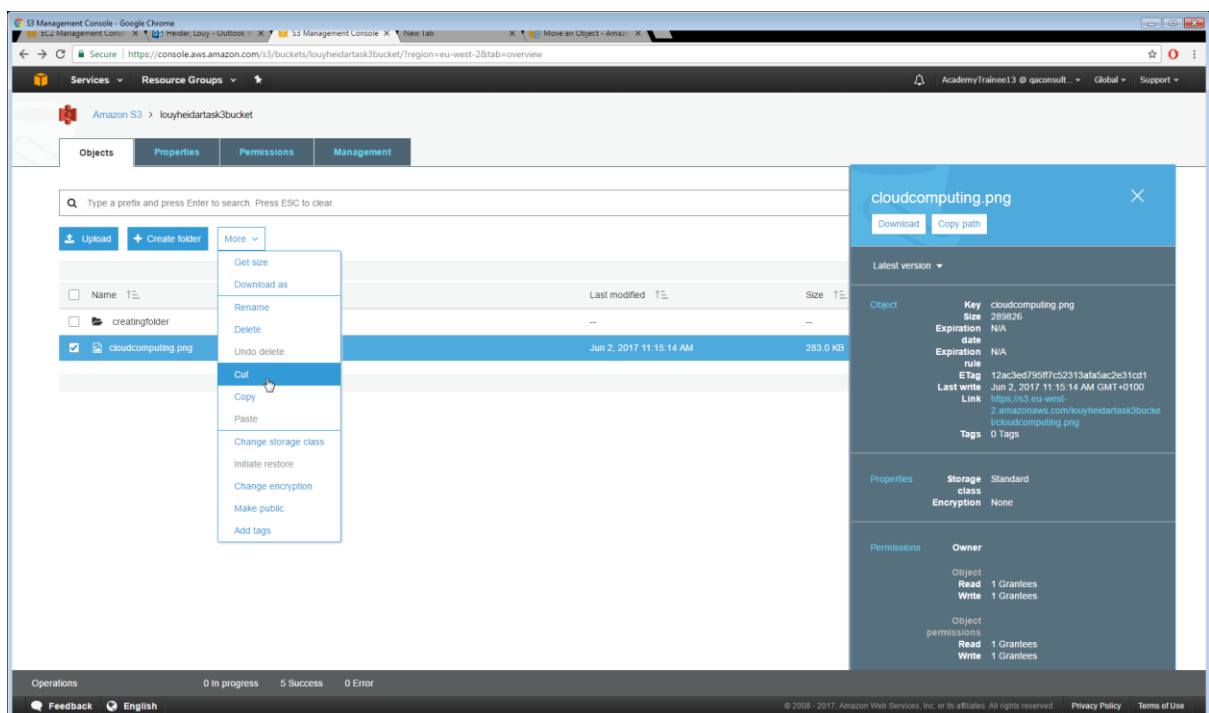
Task 3 – Using Amazon S3

Adding objects to your bucket:

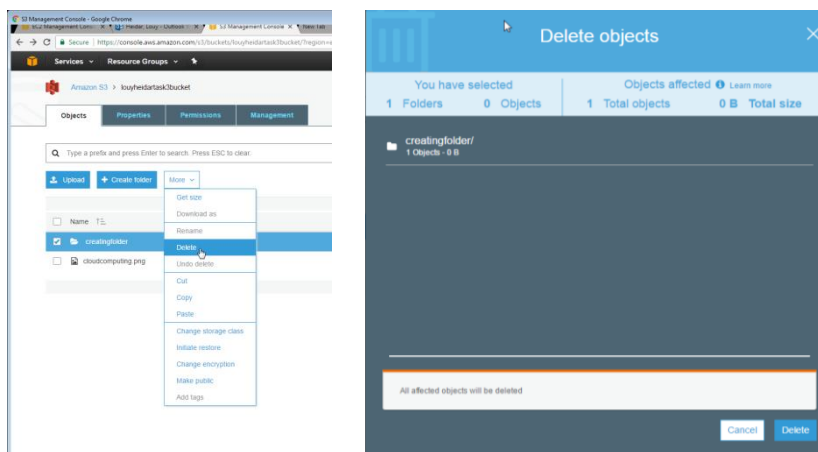
- 1) Following set instruction to create and configure bucket (QACDEVPUP_AM_Exercise_Workbook.docx)
- 2) Upload image
- 3) Moving your objects:

Select Image > Click 'Cut' > Click/Enter inside folder location > Click Paste

Note: You have to enter folder location to move item

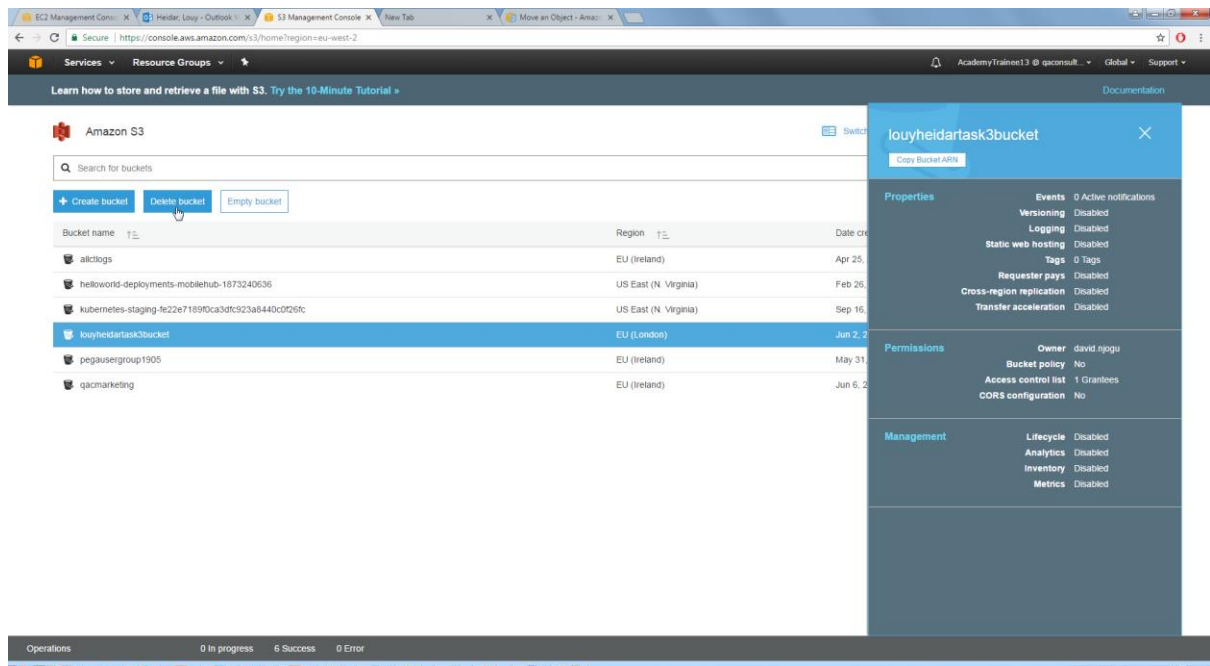


- 4) Deleting Folder: Select Folder > Click Delete > Confirmation window will appear > Click Delete to accept changes



Lheidar

5) Deleting a bucket using AWS Management Console



Note: bucket content has to be deleted to delete bucket via AWS Console
Command line can be used to remove folder and all files recursively e.g

```
AWS::S3::Bucket.delete('your_bucket', :force => true)
```