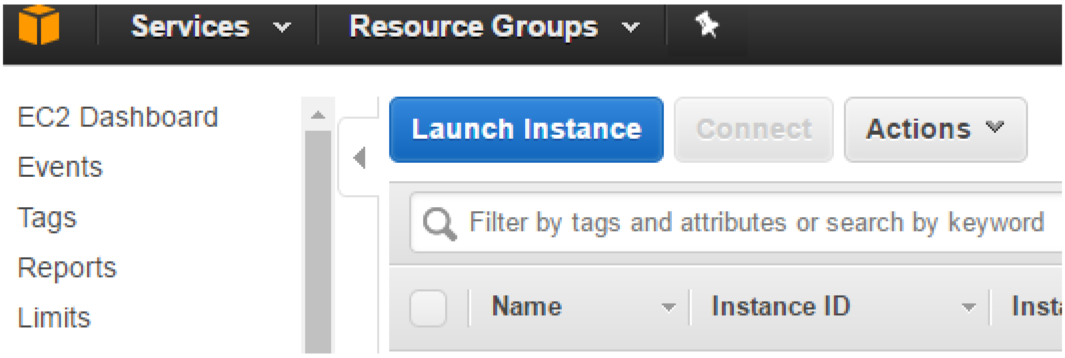
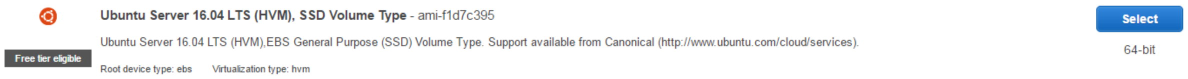
AWS

Task 1: Create an EC2 instance

EC2 – Launch Instance



Install Ubuntu 16.04

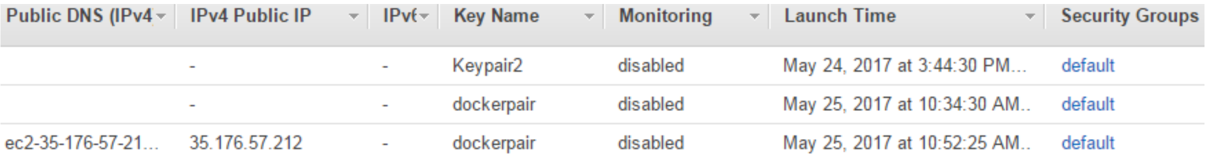


Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-f1d7c395

Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

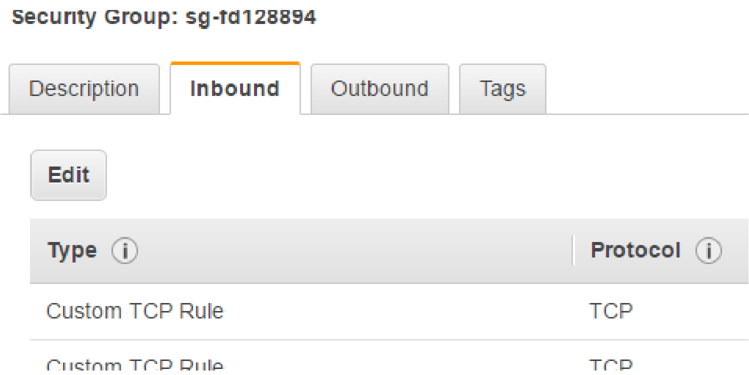
Root device type: ebs Virtualization type: hvm

Go to security groups – default to configure firewall



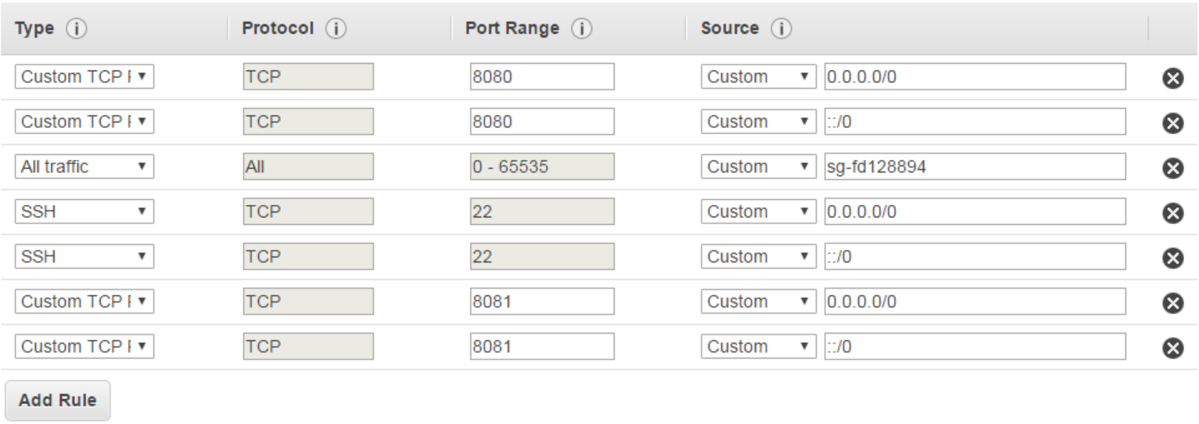
Public DNS (IPv4	IPv4 Public IP	IPv(	Key Name	Monitoring	Launch Time	Security Groups
	-	-	Keypair2	disabled	May 24, 2017 at 3:44:30 PM...	default
	-	-	dockerpair	disabled	May 25, 2017 at 10:34:30 AM..	default
ec2-35-176-57-21...	35.176.57.212	-	dockerpair	disabled	May 25, 2017 at 10:52:25 AM..	default

On security group, open port 22 for SSH, and others as required e.g. 8080



Security Group: sg-fd128894

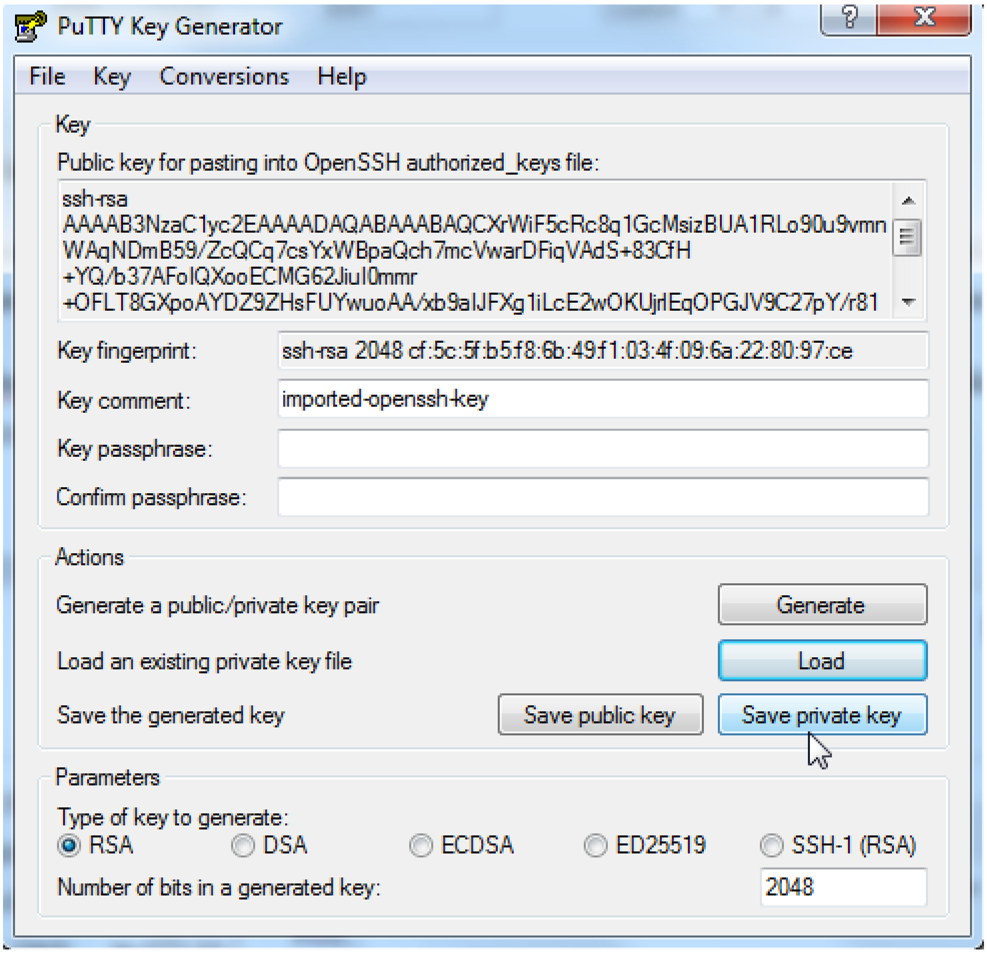
Type	Protocol
Custom TCP Rule	TCP
Custom TCP Rule	TCP



Type	Protocol	Port Range	Source	
Custom TCP I	TCP	8080	Custom	0.0.0.0/0
Custom TCP I	TCP	8080	Custom	::/0
All traffic	All	0 - 65535	Custom	sg-fd128894
SSH	TCP	22	Custom	0.0.0.0/0
SSH	TCP	22	Custom	::/0
Custom TCP I	TCP	8081	Custom	0.0.0.0/0
Custom TCP I	TCP	8081	Custom	::/0

Review and click launch instance and a screen will pop up to create keypair.

The .pem file provided by AWS to access SSH can be broken down into a private/public key pair. The private key can be created by using the .pem file on PuTTYgen – click save private key.



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCXrWiF5cRc8q1GcMsizBUA1RLo90u9vmn
WAqNDmB59/ZcQCq7csYxWBpaQch7mcVwarDFiqVAdS+83CfH
+YQ/b37AFoIQXooECMG62JiuI0mmr
+OFLT8GXpoAYDZ9ZHsFUYwuoAA/xb9aIJFXg1iLcE2wOKUjrlEqOPGJV9C27pY/r81
```

Key fingerprint: ssh-rsa 2048 cf:5c:5f:b5:f8:6b:49:f1:03:4f:09:6a:22:80:97:ce

Key comment: imported-openssh-key

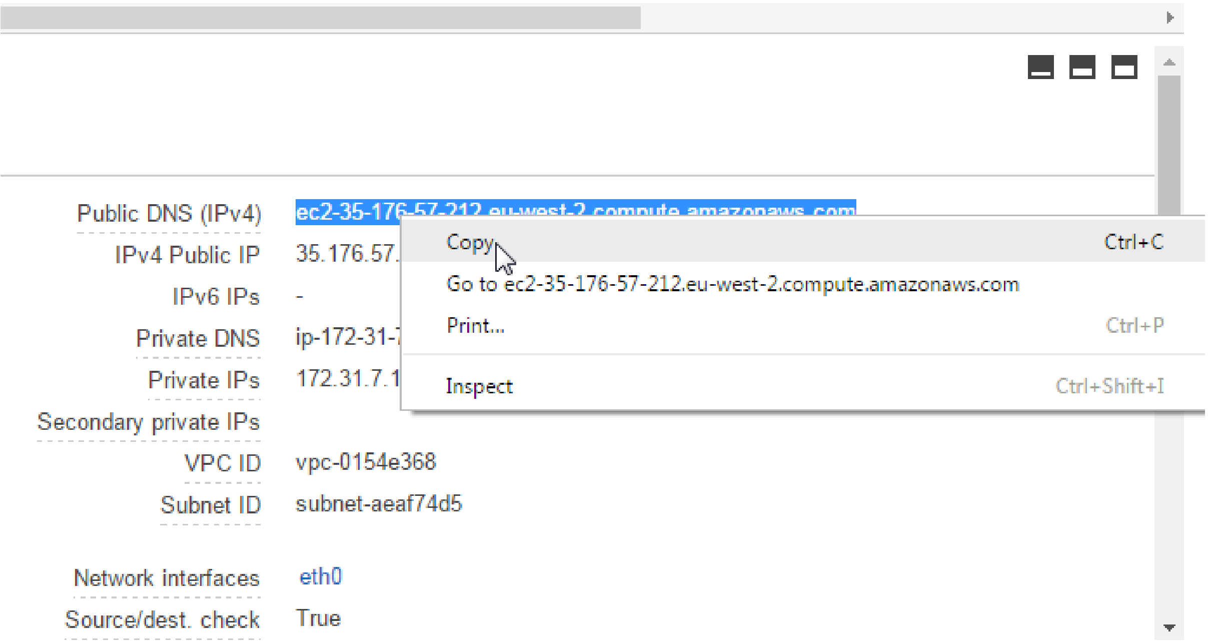
Key passphrase:

Confirm passphrase:

Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

Put your privatekey to SSH-Auth on PuTTY Config. Insert the username (Ubuntu) public IP address and port 22 to access SSH.



Public DNS (IPv4) ec2-35-176-57-212.eu-west-2.compute.amazonaws.com

IPv4 Public IP 35.176.57.

IPv6 IPs -

Private DNS ip-172-31-7

Private IPs 172.31.7.1

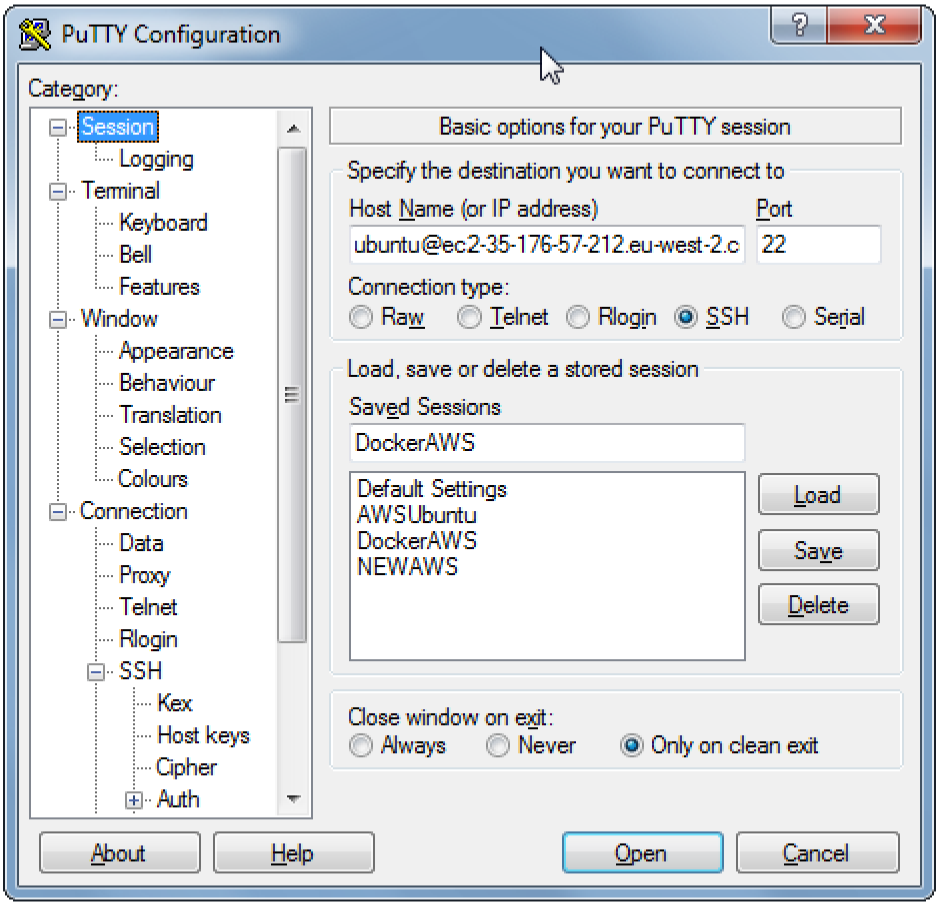
Secondary private IPs

VPC ID vpc-0154e368

Subnet ID subnet-aeaf74d5

Network interfaces eth0

Source/dest. check True



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): ubuntu@ec2-35-176-57-212.eu-west-2.c

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: DockerAWS

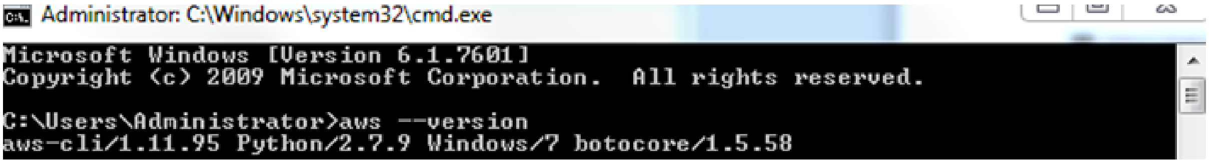
Default Settings
AWSUbuntu
DockerAWS
NEWAWS

Close window on exit: ○ Always ○ Never ● Only on clean exit

Task 2: Using the CLI

Download windows installer from:

http://docs.aws.amazon.com/cli/latest/userguide/awscli-install-windows.html#awscli-install-windows-path
http://docs.aws.amazon.com/cli/latest/userguide/tutorial-ec2-ubuntu.html
aws --version on a command line to confirm installation



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

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

Configure the CLI

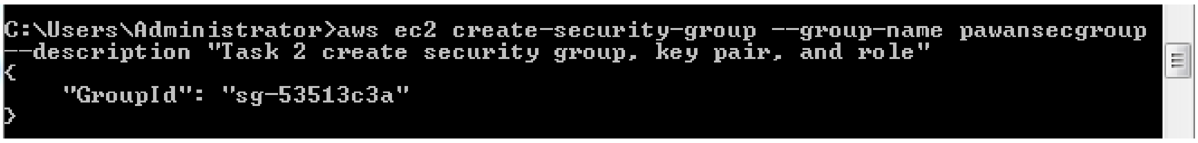
aws configure to configure the aws account

```
C:\Users\Administrator>aws configure
```

Creating a security group, key pair and role

Create security group with the following command.

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



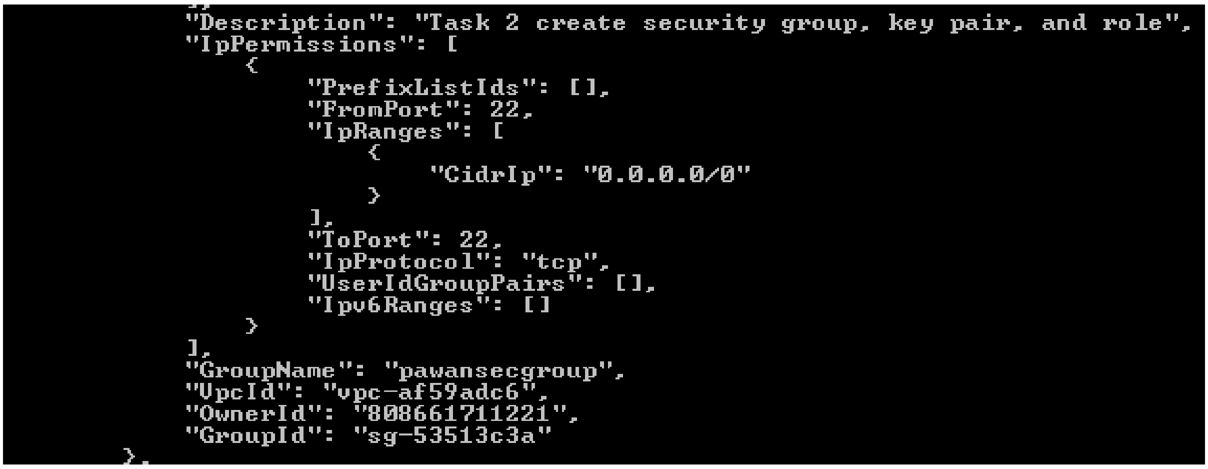
```
C:\Users\Administrator>aws ec2 create-security-group --group-name pawansecgroup
--description "Task 2 create security group, key pair, and role"
{
    "GroupId": "sg-53513c3a"
}
```

Note groupid: sg-53513c3a

To open port 22 for ssh, use the following command. Cidr can be replaced with ip address of host OS for security.

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

Use **aws ec2 describe-security-groups** command to view the change.

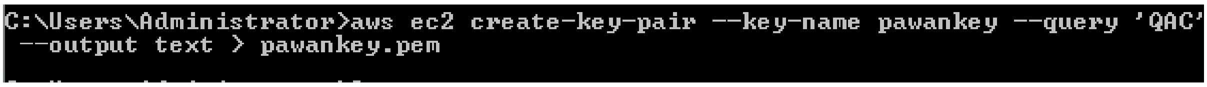


```
            "Description": "Task 2 create security group, key pair, and role",
            "IpPermissions": [
                {
                    "PrefixListIds": [],
                    "FromPort": 22,
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "ToPort": 22,
                    "IpProtocol": "tcp",
                    "UserIdGroupPairs": [],
                    "Ipv6Ranges": []
                }
            ],
            "GroupName": "pawansecgroup",
            "VpcId": "vpc-af59adc6",
            "OwnerId": "808661711221",
            "GroupId": "sg-53513c3a"
        },
```

Create a keypair

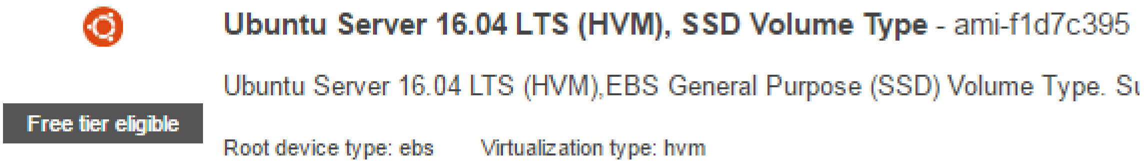
Note: KeyMaterial must be inside double quotes.

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



```
C:\Users\Administrator>aws ec2 create-key-pair --key-name pawankey --query 'QAC'
 --output text > pawankey.pem
```

Find Amazon Machine Image on EC2 dashboard



Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-f1d7c395

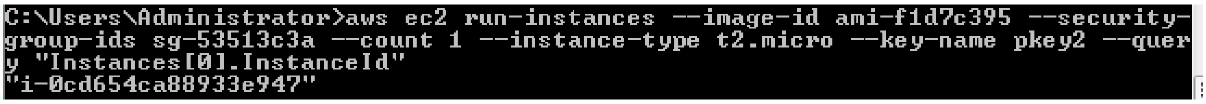
Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Su

Root device type: ebs Virtualization type: hvm

Note: ami-f1d7c395

Launch your EC2 Instance

Run the aws instance with the AMI and Group ID



```
C:\Users\Administrator>aws ec2 run-instances --image-id ami-f1d7c395 --security-
group-ids sg-53513c3a --count 1 --instance-type t2.micro --key-name pkey2 --quer
y "Instances[0].InstanceId"
"i-0cd654ca88933e947"
```

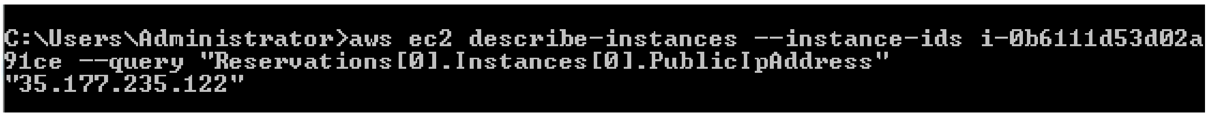
```
aws ec2 run-instances --image-id ami-f1d7c395 --security-group-ids sg-
53513c3a --count 1 --instance-type t2.micro --key-name pkey2 --query
"Instances[0].InstanceId"
```

Note: i-0cd654ca88933e947

Obtain the IP Address & SSH

To get the public ip address of instance, the following command is run

aws ec2 describe-instances --instance-ids i-0cd654ca88933e947 --query "Reservations[0].Instances[0].PublicIpAddress "



```
C:\Users\Administrator>aws ec2 describe-instances --instance-ids i-0b6111d53d02a
91ce --query "Reservations[0].Instances[0].PublicIpAddress"
"35.177.235.122"
```

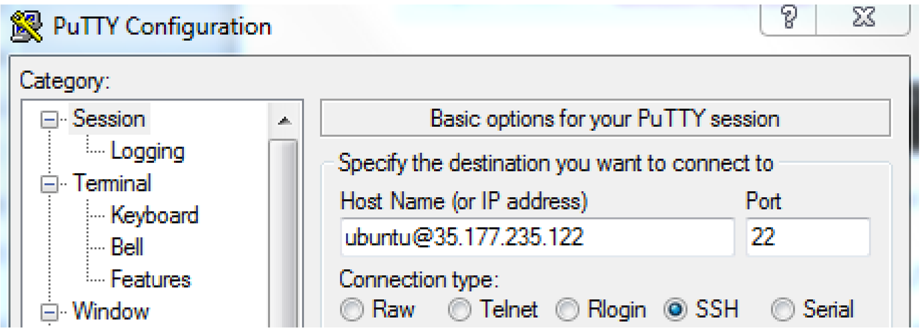
```
aws ec2 describe-instances --instance-ids i-ec3e1e2k --query
"Reservations[0].Instances[0].PublicIpAddress"
```

Create private key using the pem file created earlier.

On windows make sure there is no single quote but rather double quote on the command

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

Using PuTTYgen, create private key using the pem file.

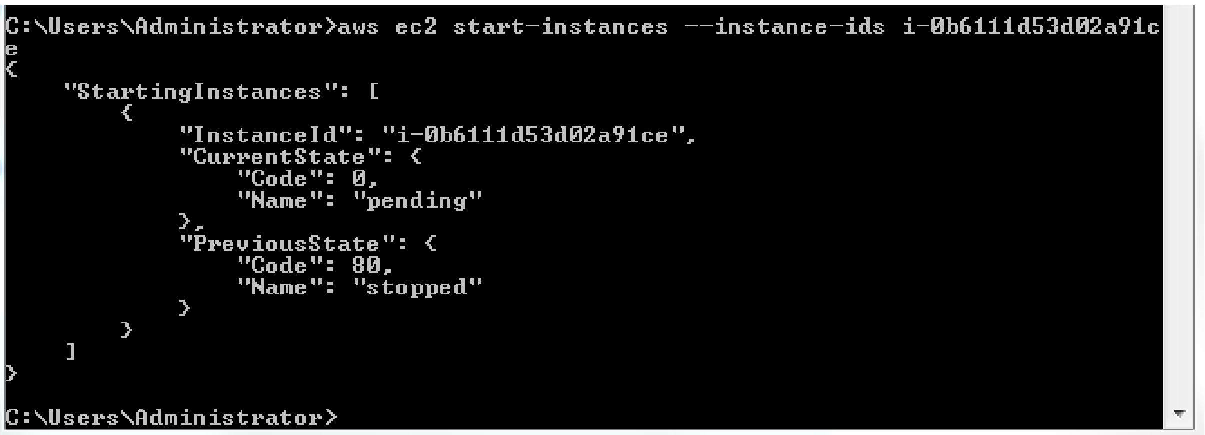

PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Basic options for your PuTTY session
Specify the destination you want to connect to

Host Name (or IP address): ubuntu@35.177.235.122

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

To start a stopped instance:



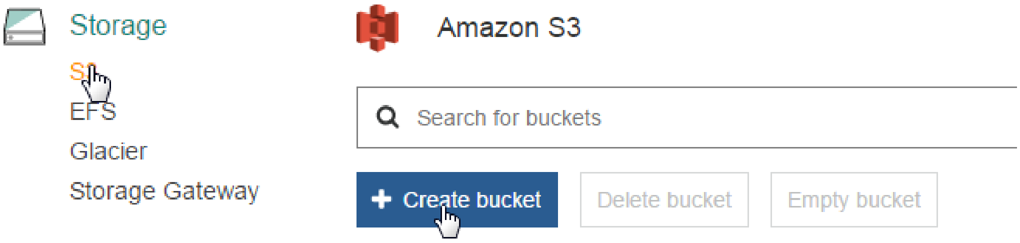
```
C:\Users\Administrator>aws ec2 start-instances --instance-ids i-0b6111d53d02a91c
e
{
    "StartingInstances": [
        {
            "InstanceId": "i-0b6111d53d02a91ce",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}

C:\Users\Administrator>
```

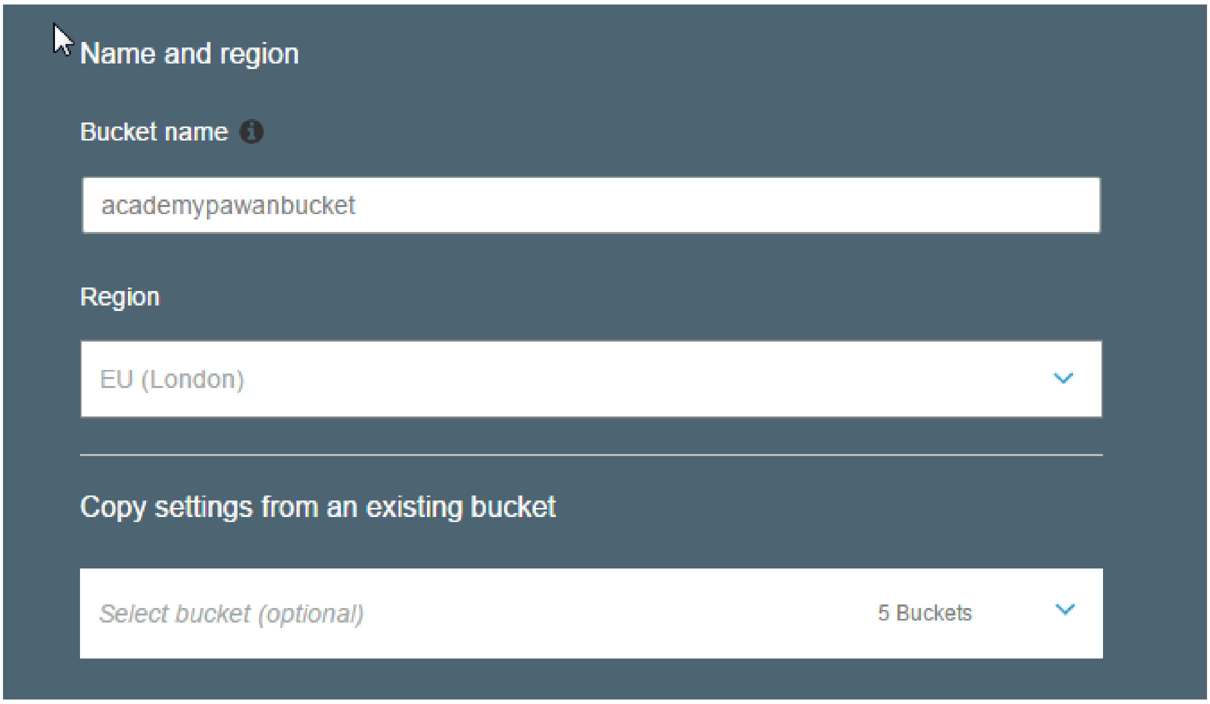
Task 3 – Using Amazon S3

Creating a Bucket in Amazon S3

Click S3 and click create bucket.



Step 1: Name, region



Name and region

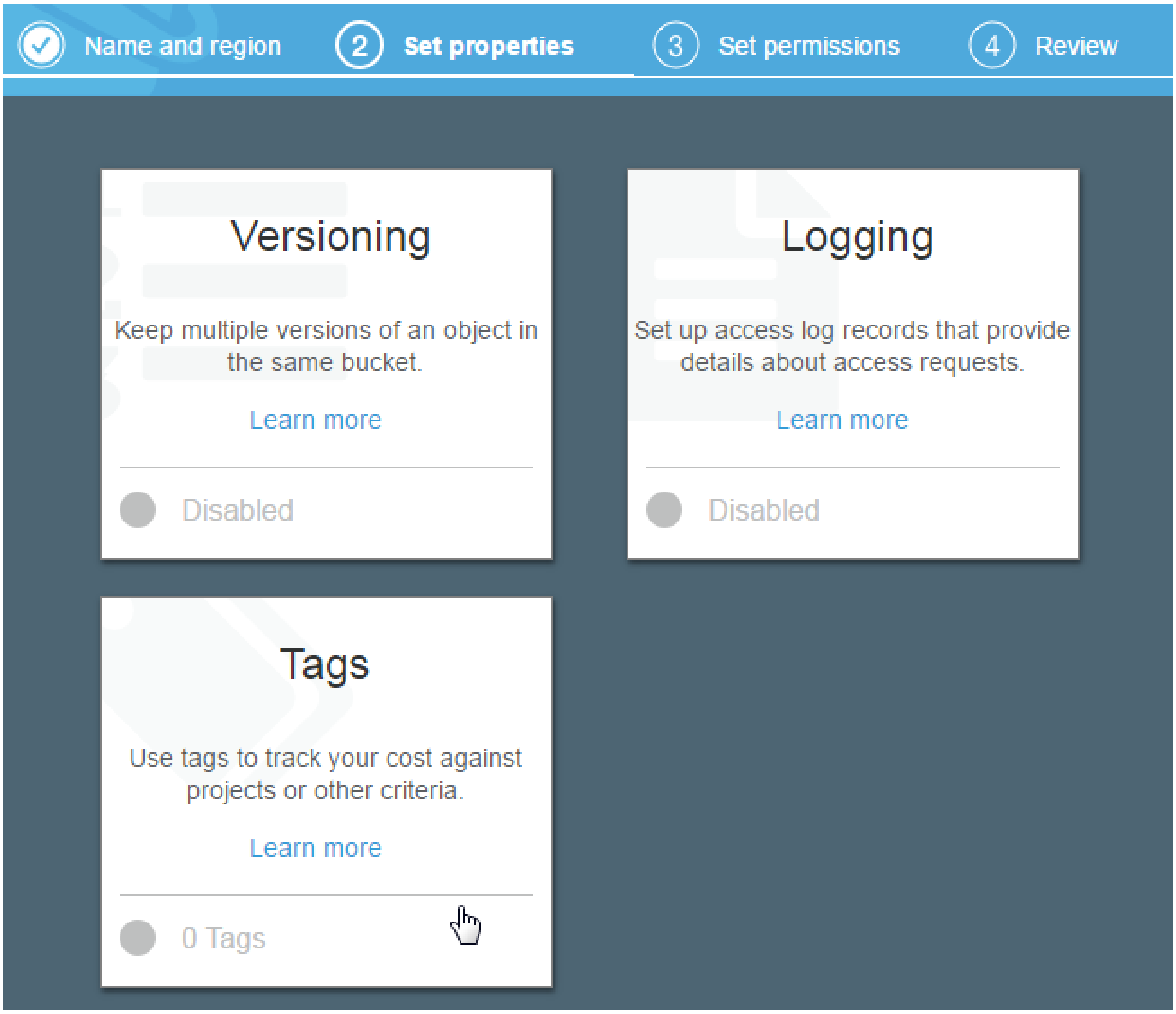
Bucket name: academypawanbucket

Region: EU (London)

Copy settings from an existing bucket

Select bucket (optional) — 5 Buckets

Step2: Set versioning, tags, logging options



Versioning

Keep multiple versions of an object in the same bucket.

Learn more

● Disabled

Logging

Set up access log records that provide details about access requests.

Learn more

● Disabled

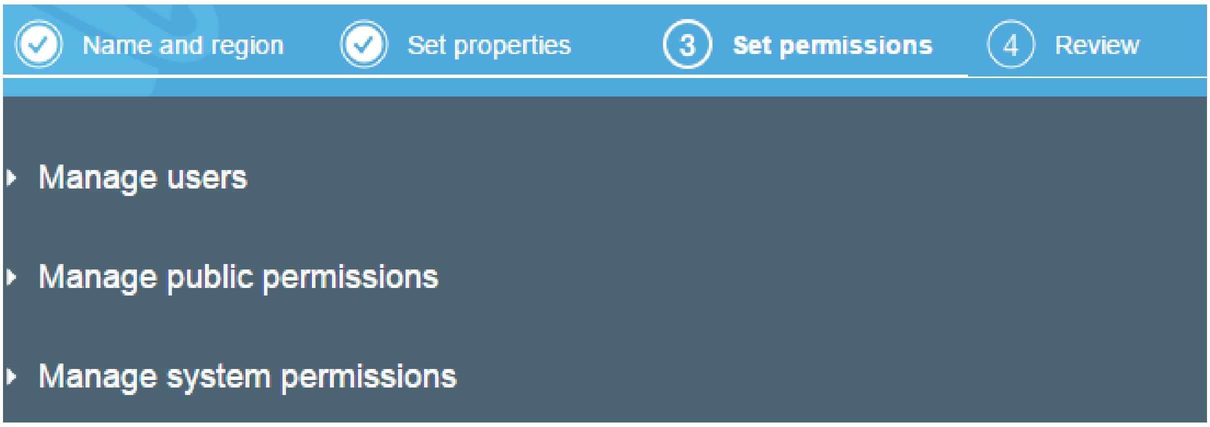
Tags

Use tags to track your cost against projects or other criteria.

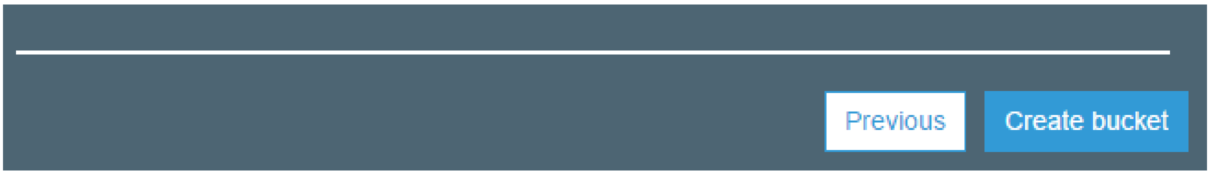
Learn more

● 0 Tags

Step 3: Users and permissions

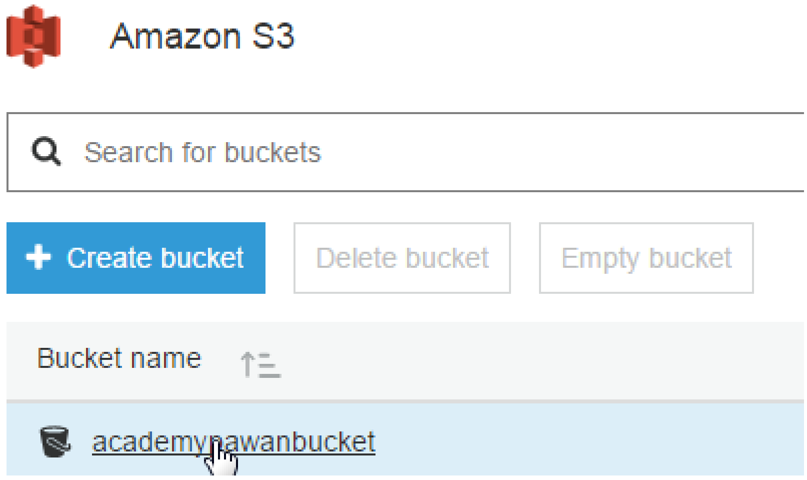


Step 4: Review and create bucket

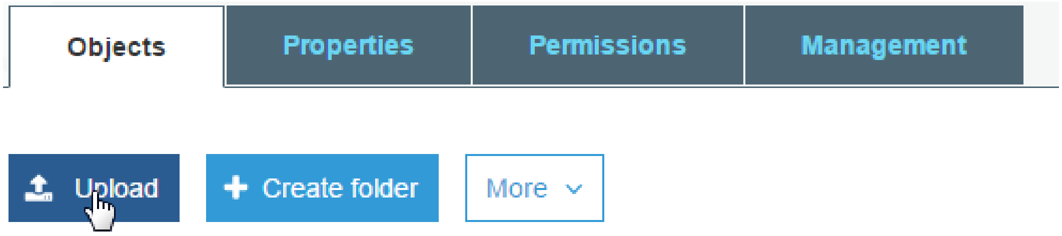


Adding objects to your bucket

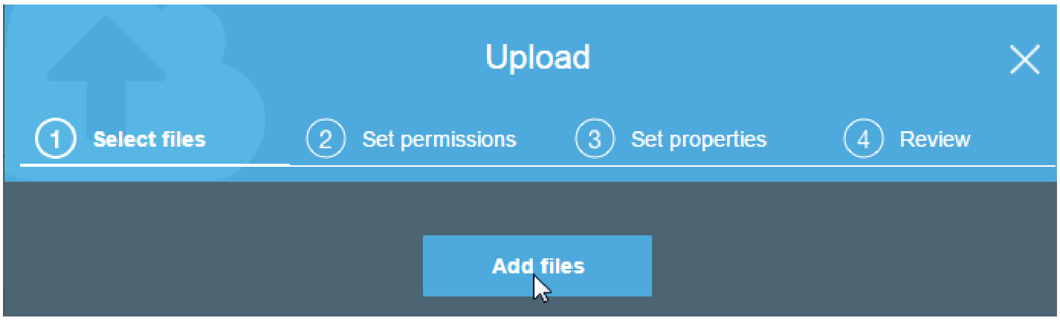
Click the bucket created



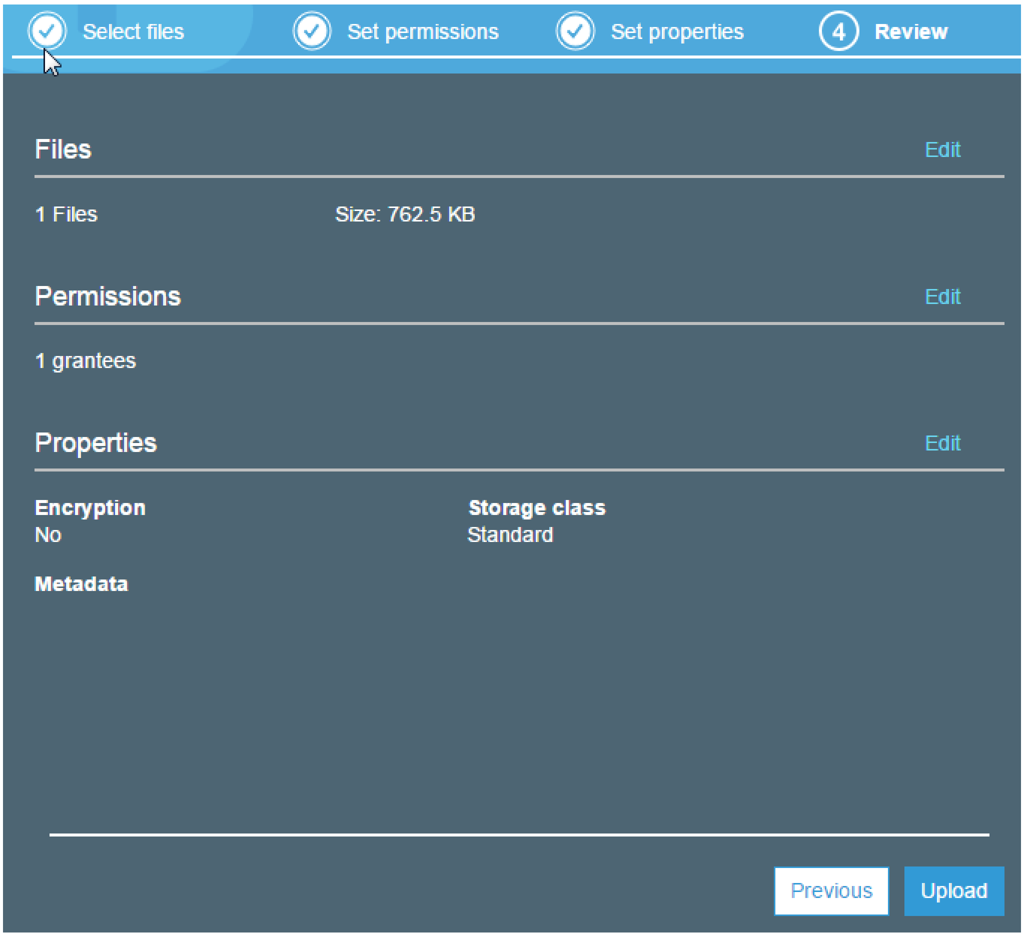
Click upload to upload a file to bucket



Add files.



Set users, permissions, encryption, metadata etc as required.



Files

1 Files Size: 762.5 KB

Permissions

1 grantees

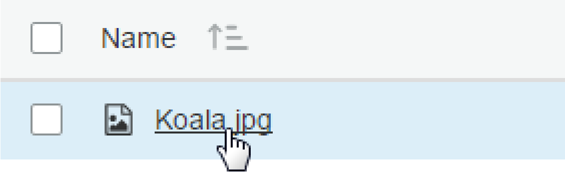
Properties

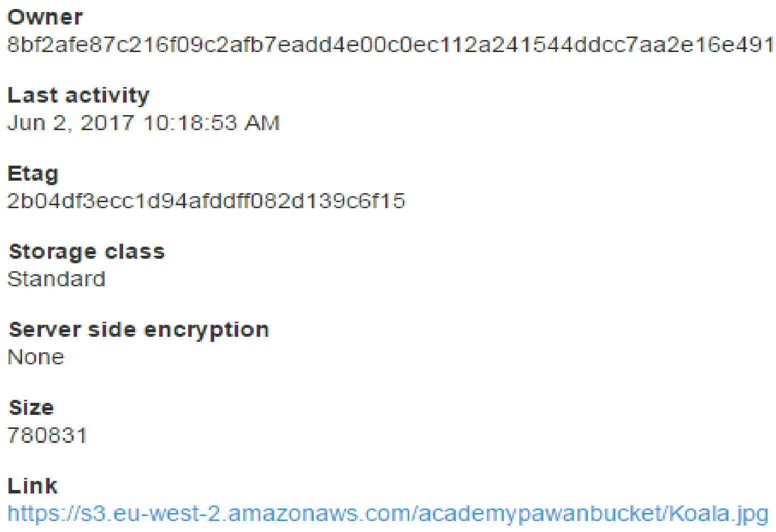
Encryption
No

Storage class
Standard

Metadata

Click the image to find its properties





Owner
8bf2afe87c216f09c2afb7eadd4e00c0ec112a241544ddcc7aa2e16e4918485c

Last activity
Jun 2, 2017 10:18:53 AM

Etag
2b04df3ecc1d94afddff082d139c6f15

Storage class
Standard

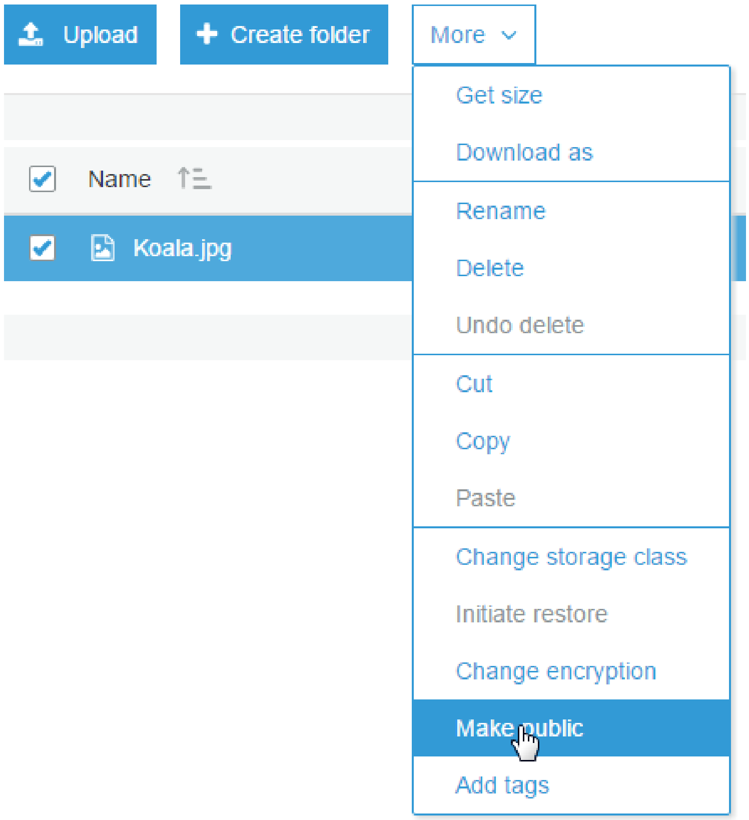
Server side encryption
None

Size
780831

Link
https://s3.eu-west-2.amazonaws.com/academypawanbucket/Koala.jpg

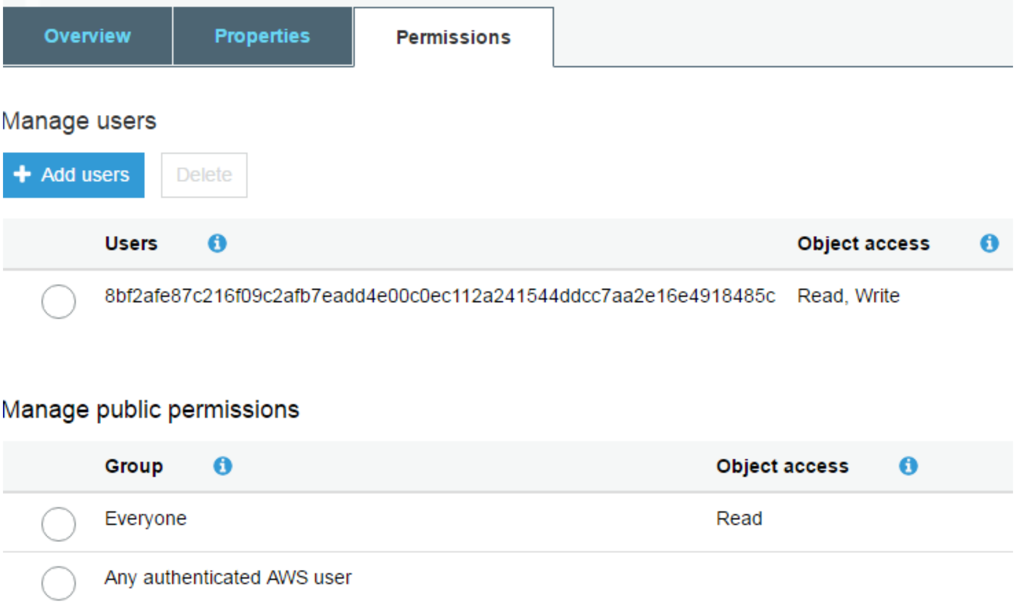
By default file is private for public use, make public.

Select image and choose make public option.



To make it private again, click the image to see its properties.

Under public permissions, everyone has read access.



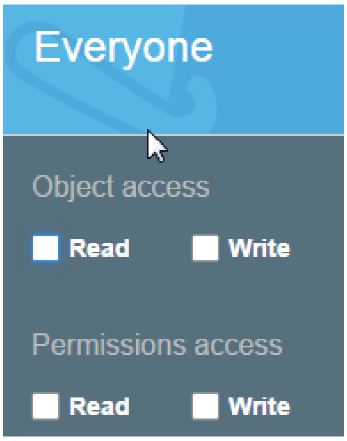
Manage users

	Users	Object access
○	8bf2afe87c216f09c2afb7eadd4e00c0ec112a241544ddcc7aa2e16e4918485c	Read, Write

Manage public permissions

	Group	Object access
○	Everyone	Read
○	Any authenticated AWS user	

Uncheck the Read access permission and save to make the file private again.



Everyone

Object access

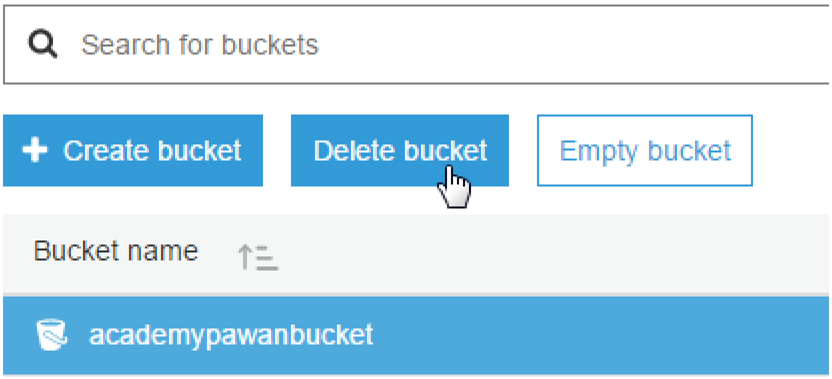
□ Read □ Write

Permissions access

□ Read □ Write

To delete a bucket, delete all objects inside the bucket.

Select the bucket and delete it.



Bucket name
academypawanbucket