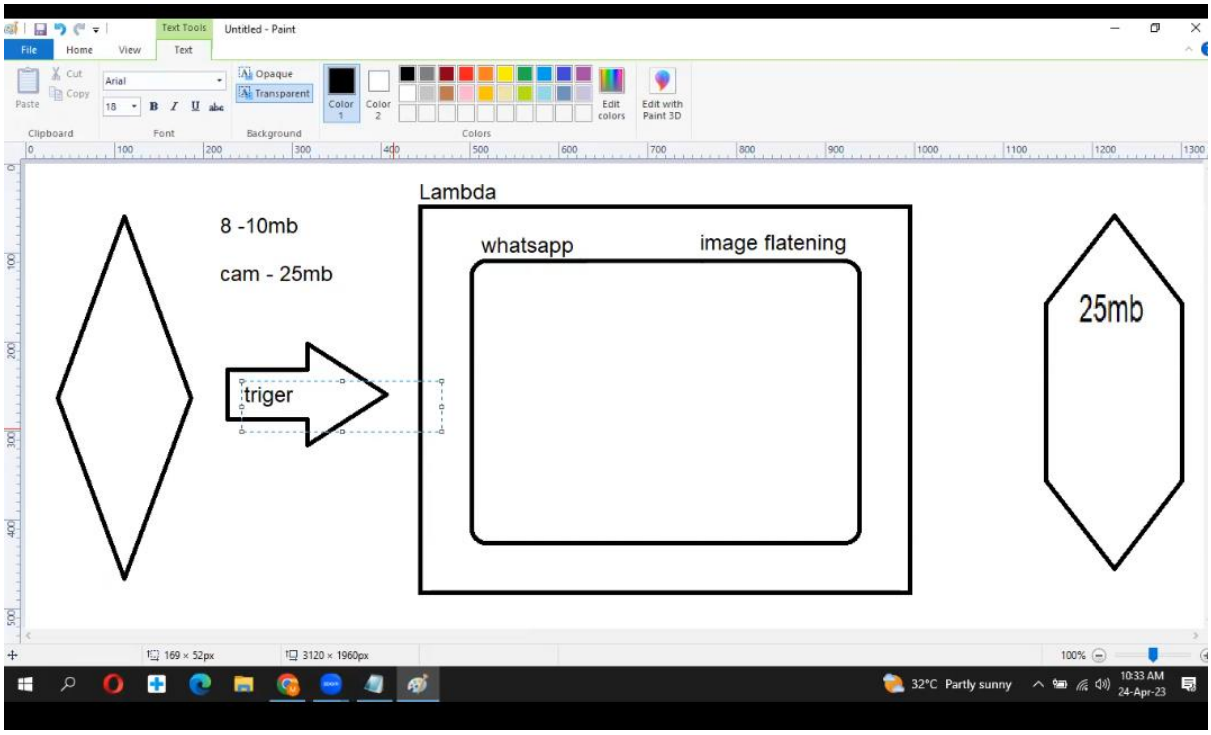
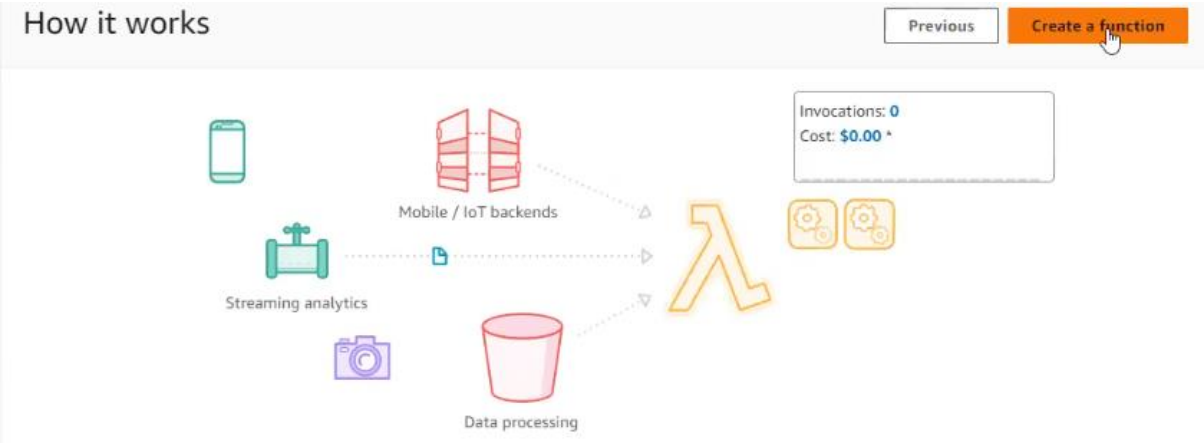
LAMBDA

LAMBDA SERVERLESS developer tool



LAMBDA RUNS ON TRIGER



Compute -- lambda

Compute

AWS Lambda
lets you run code without thinking about servers.

You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

.net

Python

.net core

Node js

Ruby

LAMBDA FUNCTION

Senior developer

Code – python

In this scenario

We use python code to delete everything we mentioned in that code

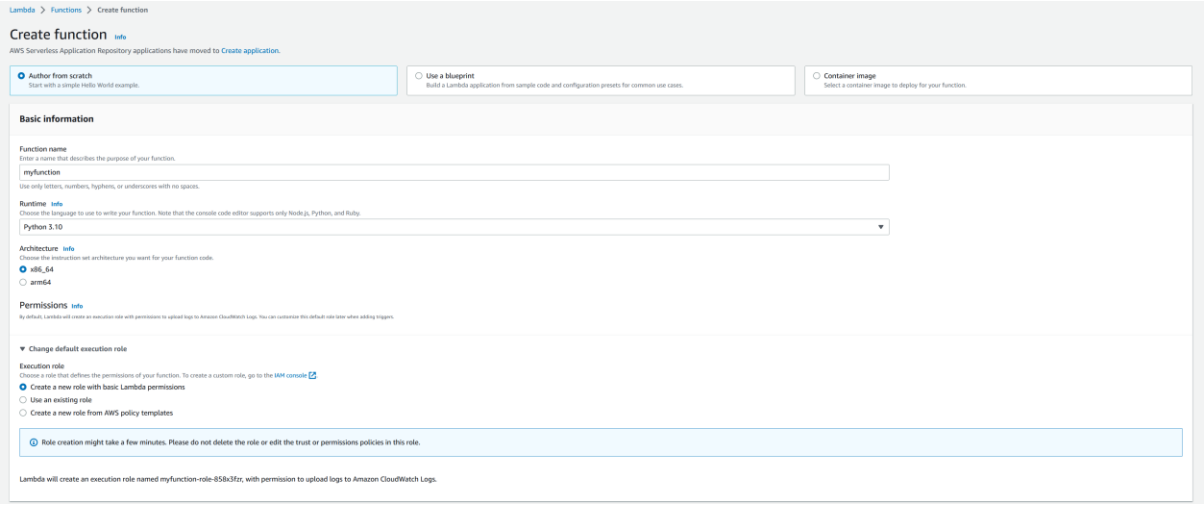
9:00 to 6:00 clk

(*) create EC2 server in Mumbai, Singapore and London

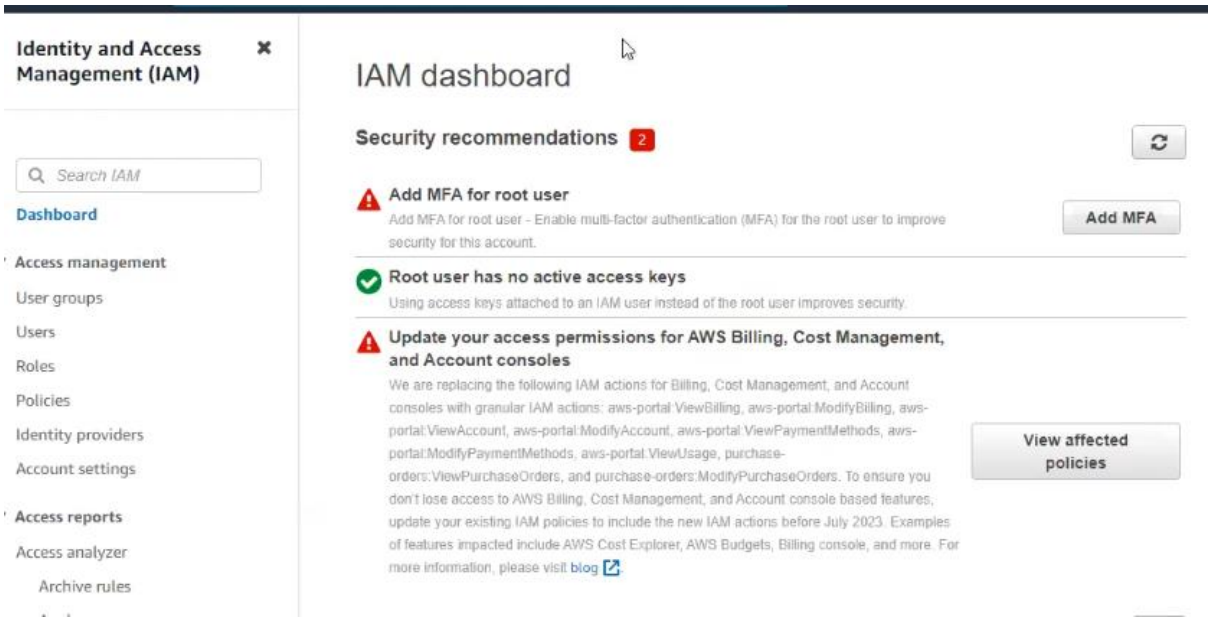
(*) refer delete python code 78 lines cleanall.py

(*) create function in lambda

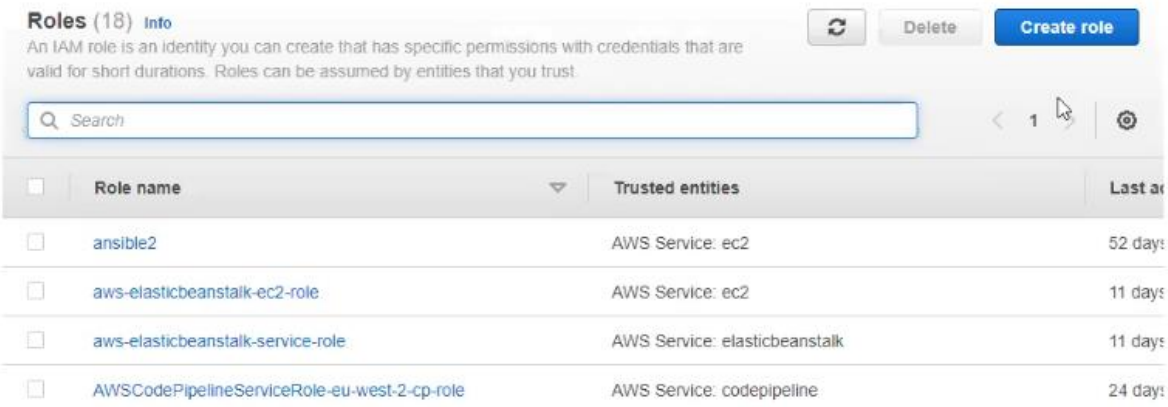
(*) in permissions we need to create roles using IAM function



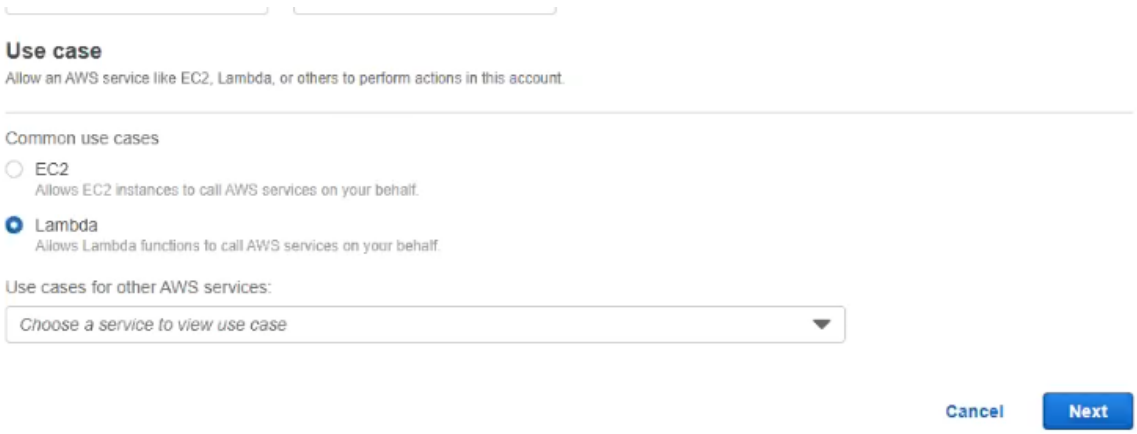
(*) IAM Dashboard create roles



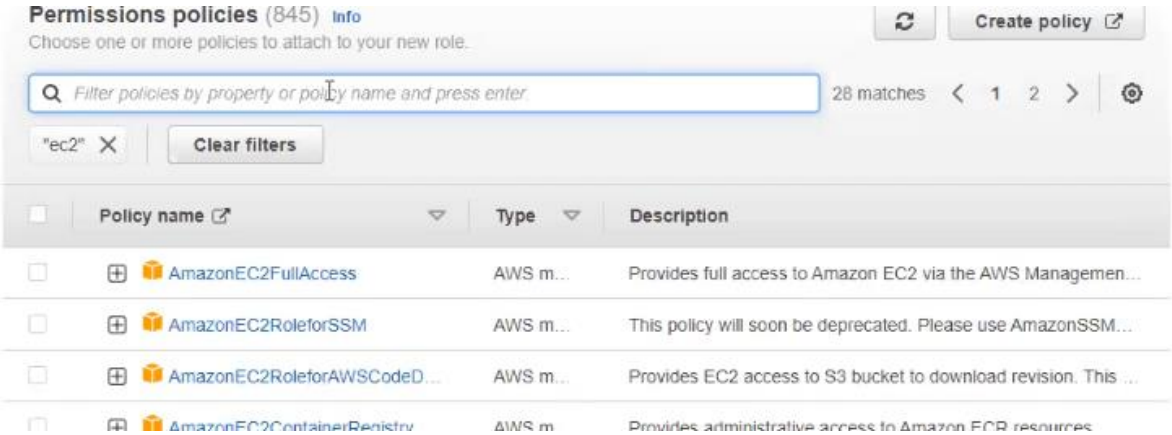
(*) create roles



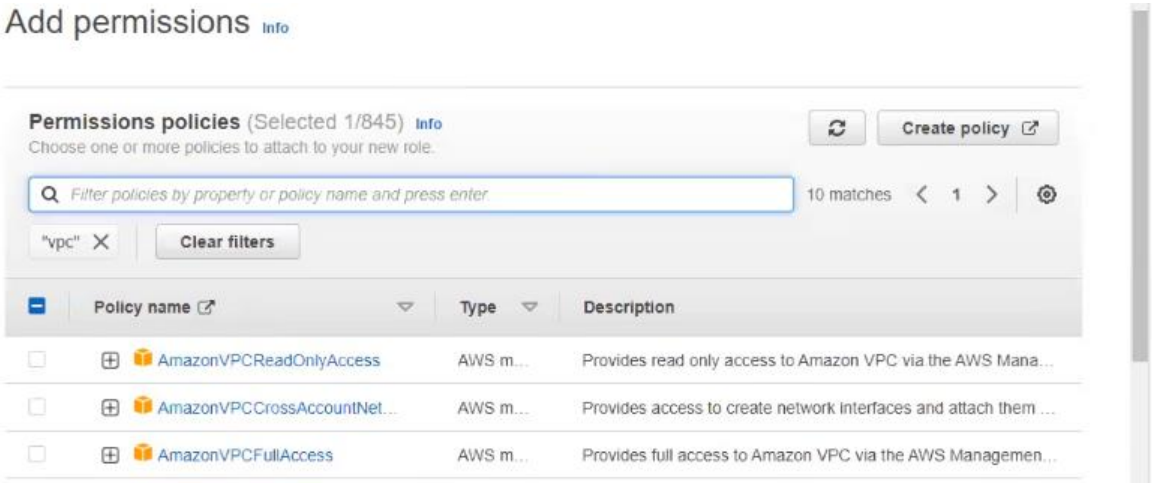
(*) use case lambda



(*) EC2 full access

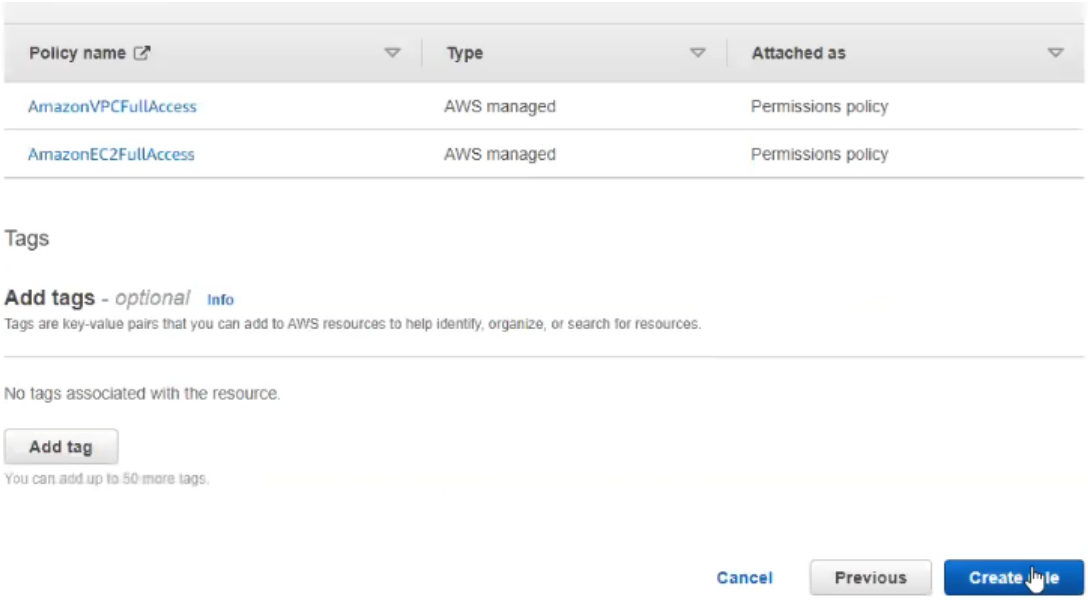


(*) vpc full access

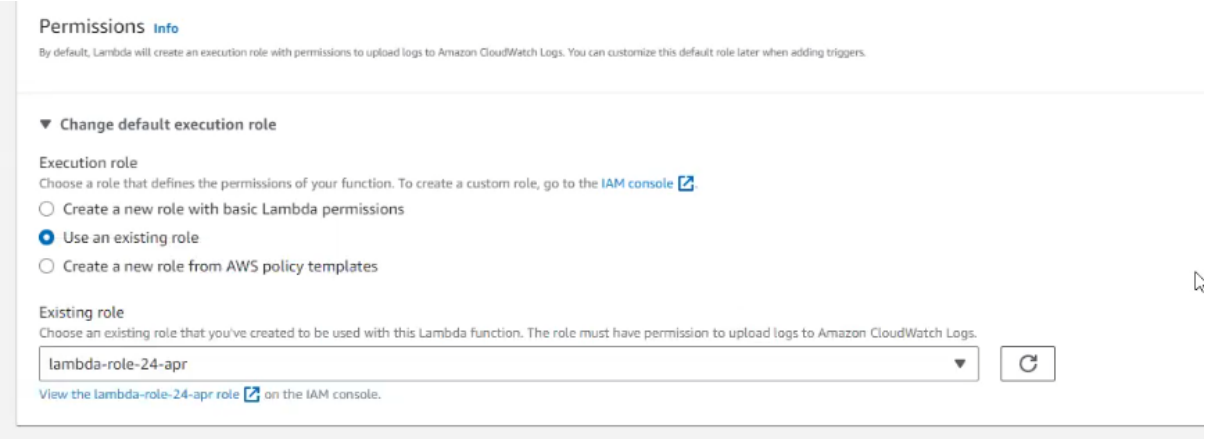


(*) name and review

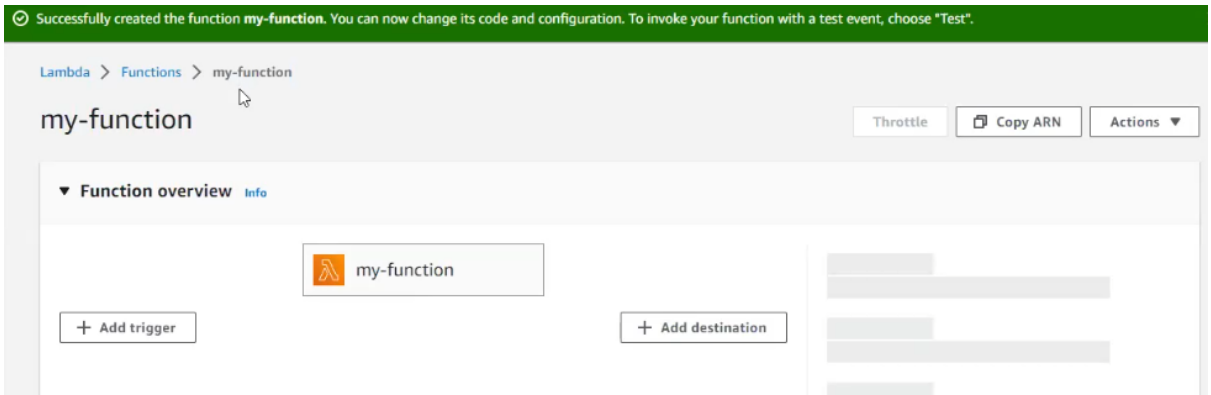
(*) create role



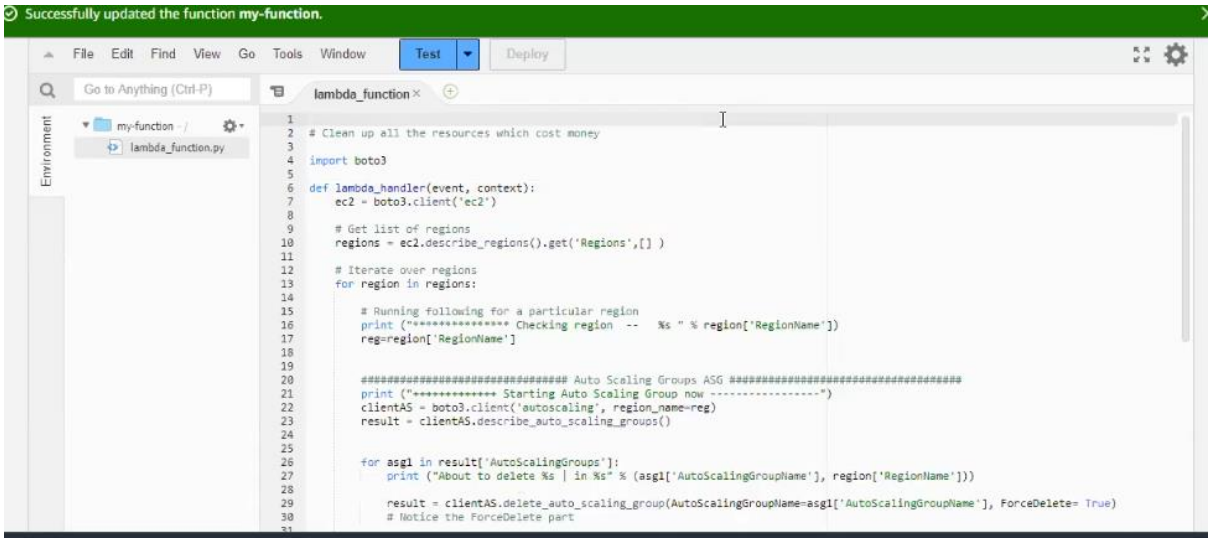
(*) in permissions apply IAM ROLE – CREATE function



(*) function has been created

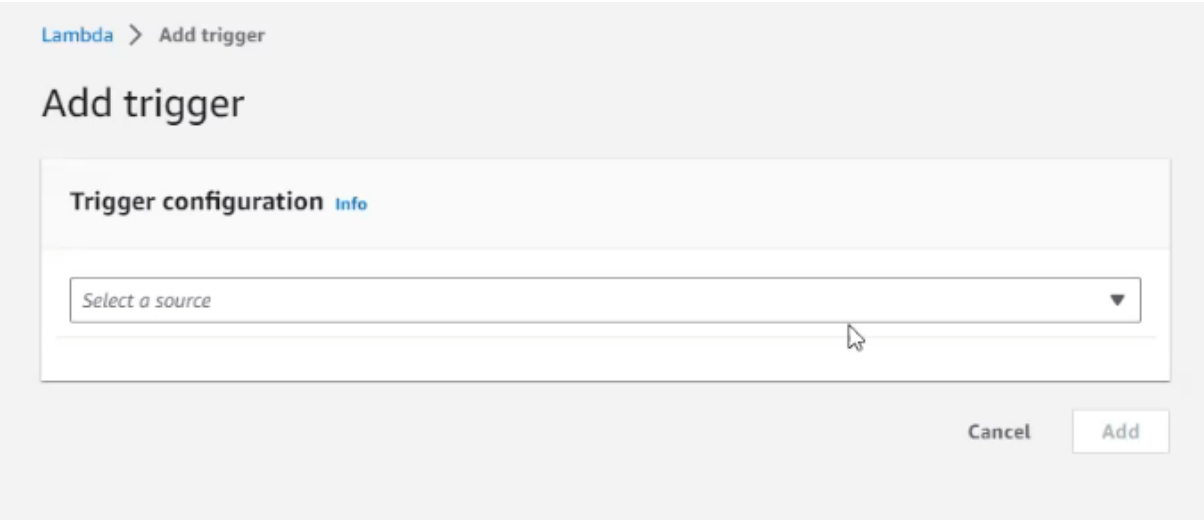


(*) copy and paste the python cleanall.py code in the editor -- deploy



(*) if we test the code the ec2 and other functions in all regions get terminated

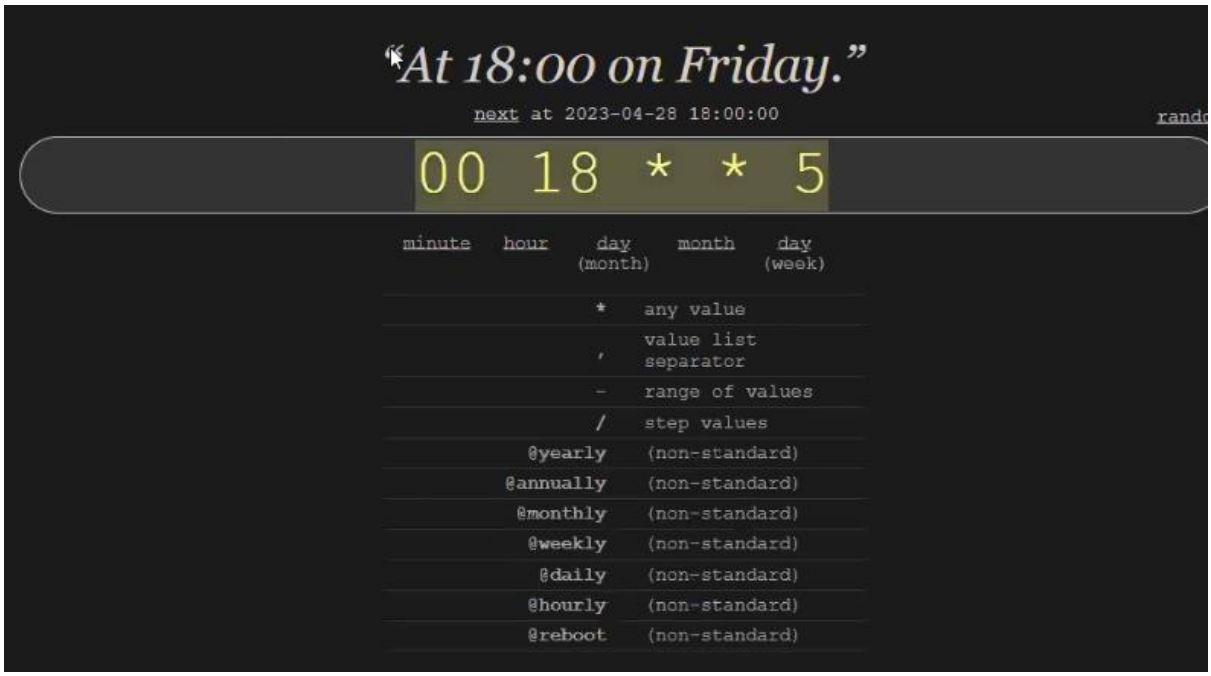
(*) event based and time based trigger – time based



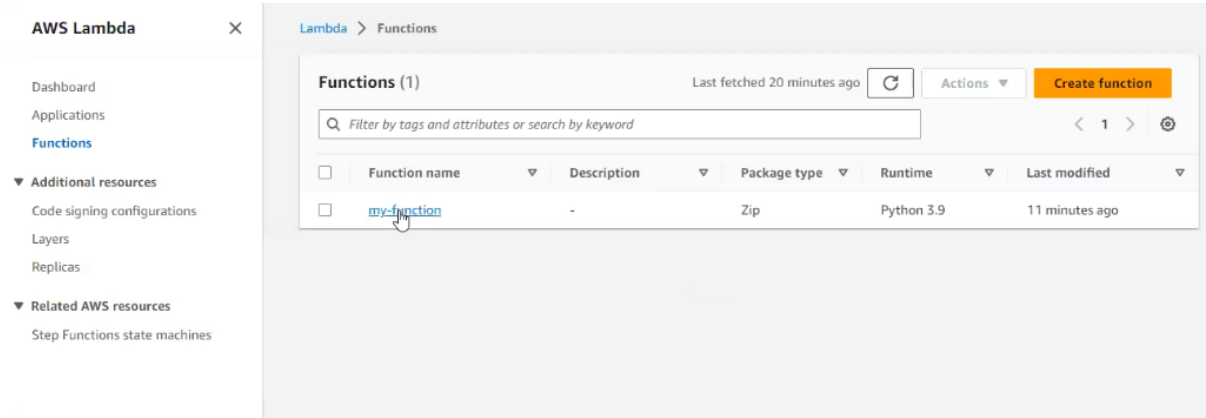
(*) event trigger

Event bridge uses cron

Cron is used for scheduling



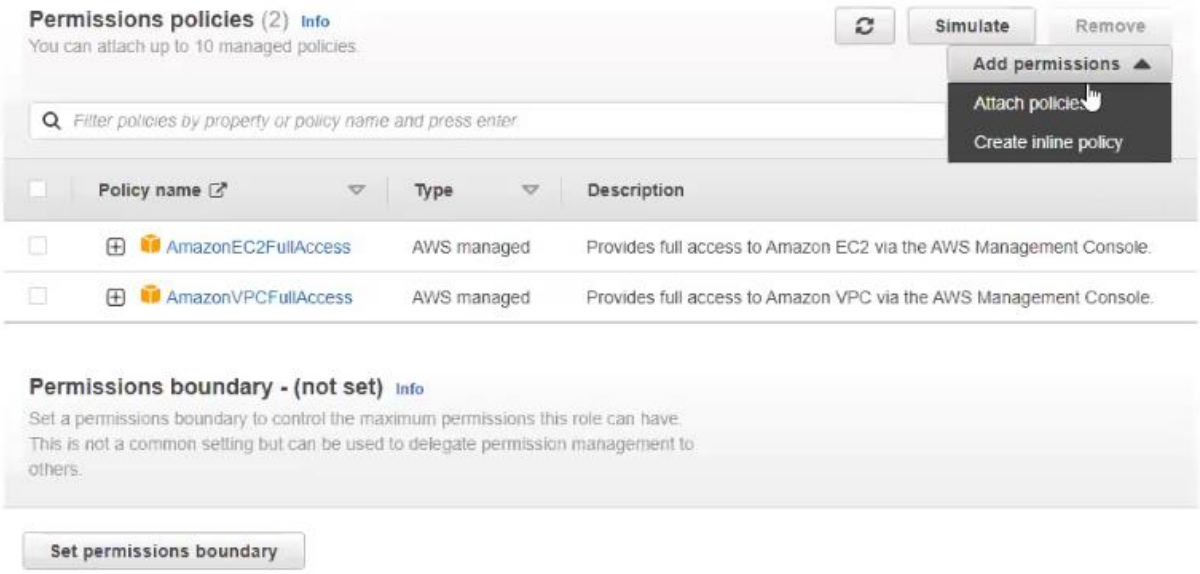
(*) go to lambda functions



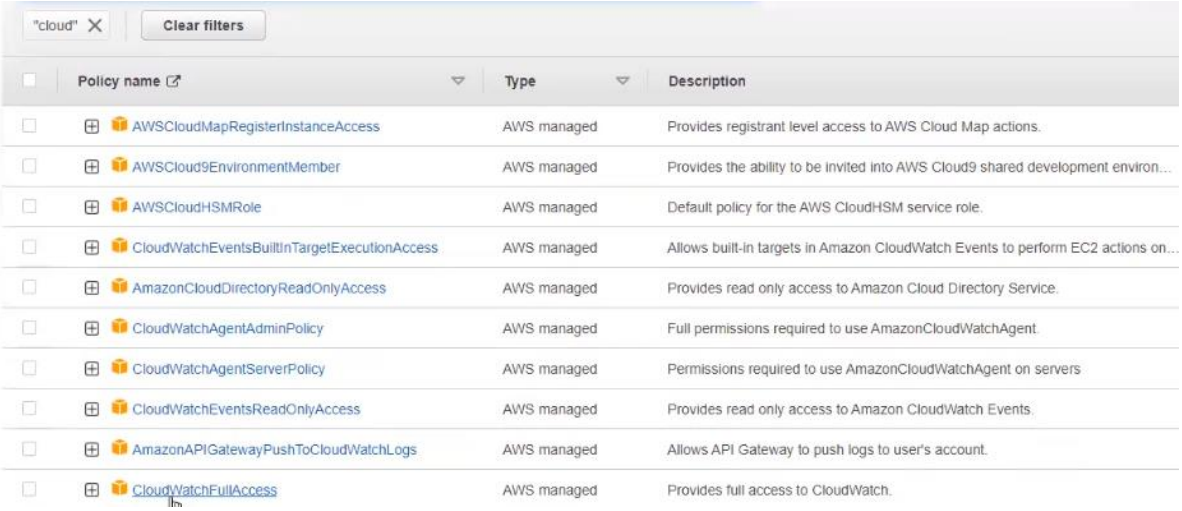
(*) we are going to give access to cloud watch role

(*) go to IAM select existing role

(*) attach policies



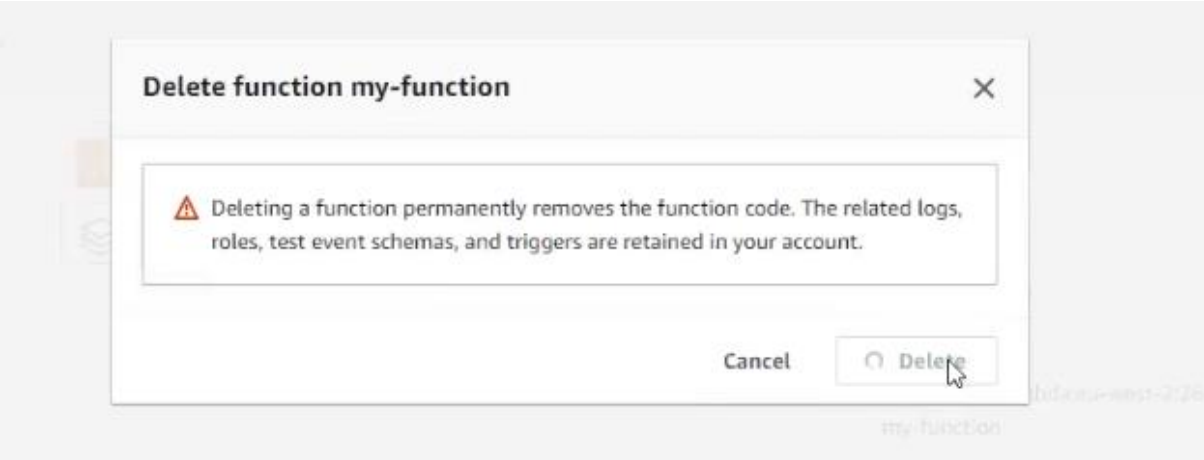
(*) cloud watch full access—add permissions



"cloud" ✕ Clear filters

Policy name	Type	Description
AWSCloudMapRegisterInstanceAccess	AWS managed	Provides registrant level access to AWS Cloud Map actions.
AWSCloud9EnvironmentMember	AWS managed	Provides the ability to be invited into AWS Cloud9 shared development environ...
AWSCloudHSMRole	AWS managed	Default policy for the AWS CloudHSM service role.
CloudWatchEventsBuiltInTargetExecutionAccess	AWS managed	Allows built-in targets in Amazon CloudWatch Events to perform EC2 actions on...
AmazonCloudDirectoryReadOnlyAccess	AWS managed	Provides read only access to Amazon Cloud Directory Service.
CloudWatchAgentAdminPolicy	AWS managed	Full permissions required to use AmazonCloudWatchAgent.
CloudWatchAgentServerPolicy	AWS managed	Permissions required to use AmazonCloudWatchAgent on servers
CloudWatchEventsReadOnlyAccess	AWS managed	Provides read only access to Amazon CloudWatch Events.
AmazonAPIGatewayPushToCloudWatchLogs	AWS managed	Allows API Gateway to push logs to user's account.
CloudWatchFullAccess	AWS managed	Provides full access to CloudWatch.

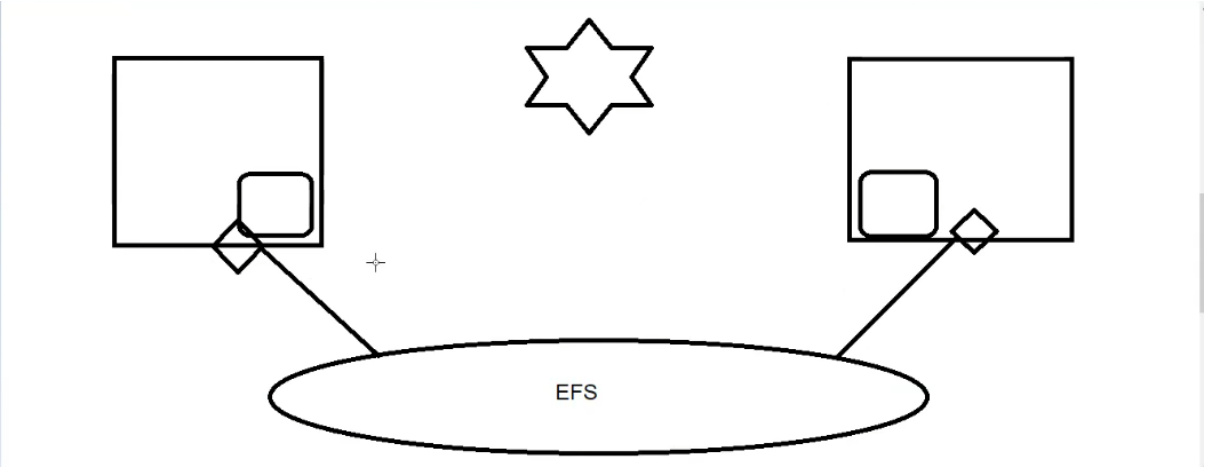
(*) action-delete function



ELASTIC FILE SYSTEM EFS

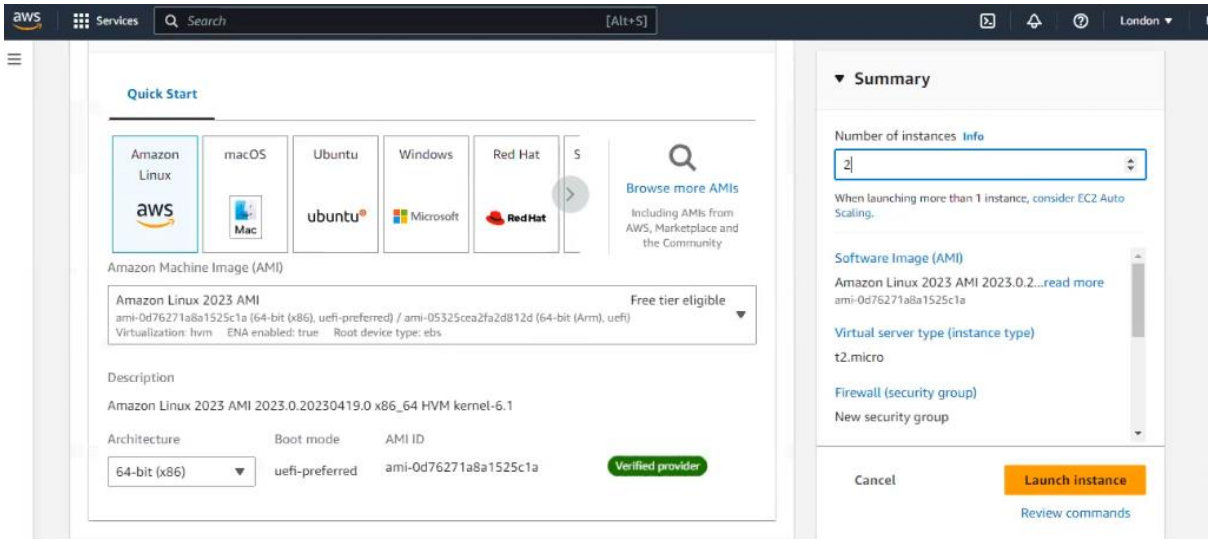
Common file system

Storage --- EFS

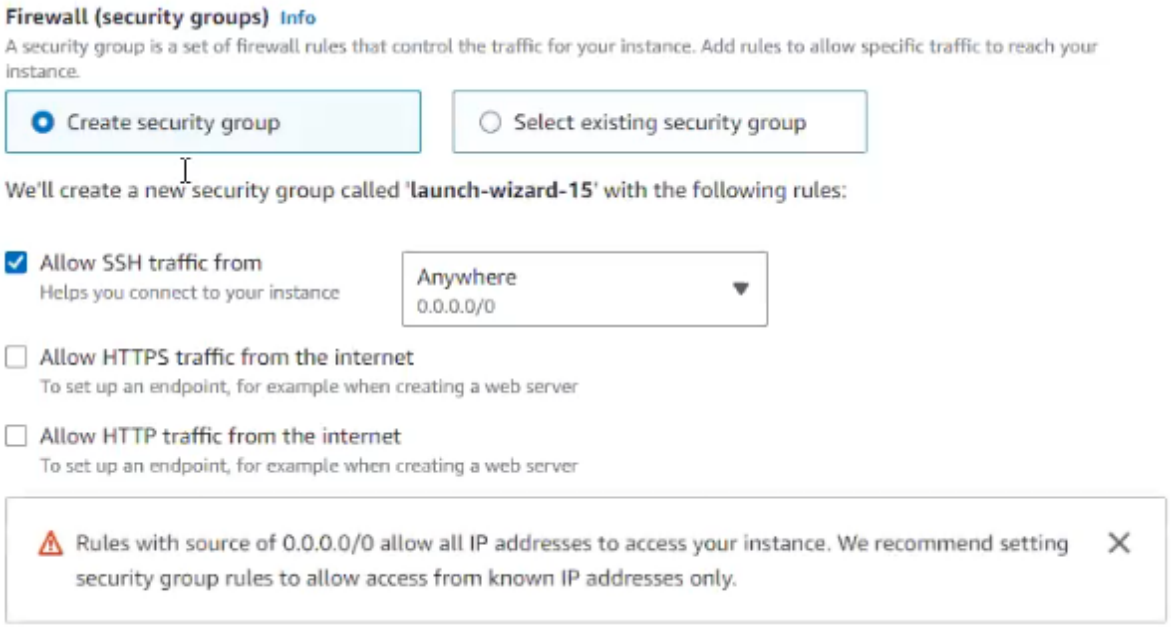


(*) First create 2 servers with same configuration ec2 linux instance

(*) NFS network file sharing



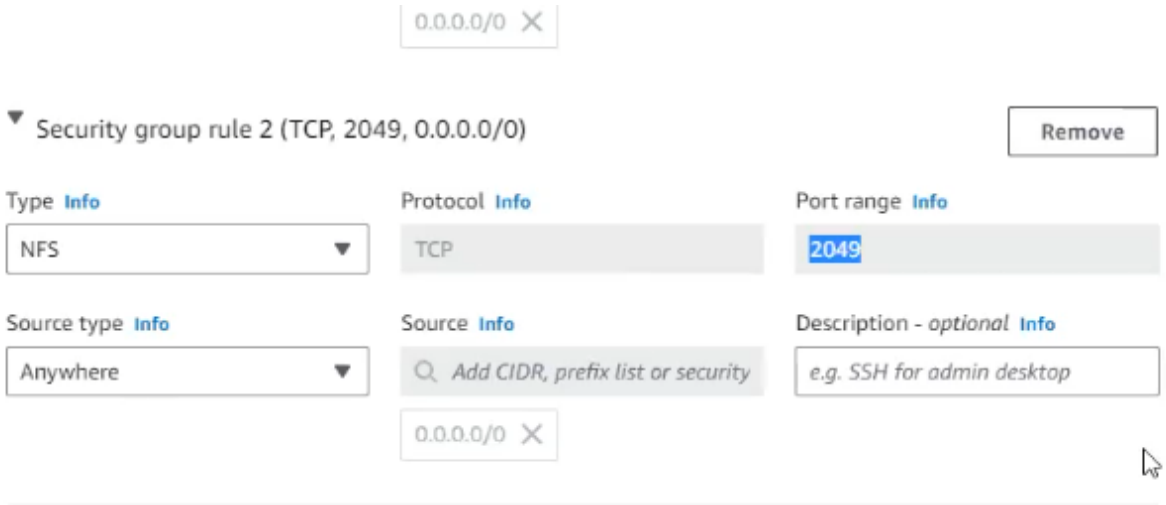
(*) create security group NFS



(*) Network settings edit

(*) add security group rule – NFS port no 2049

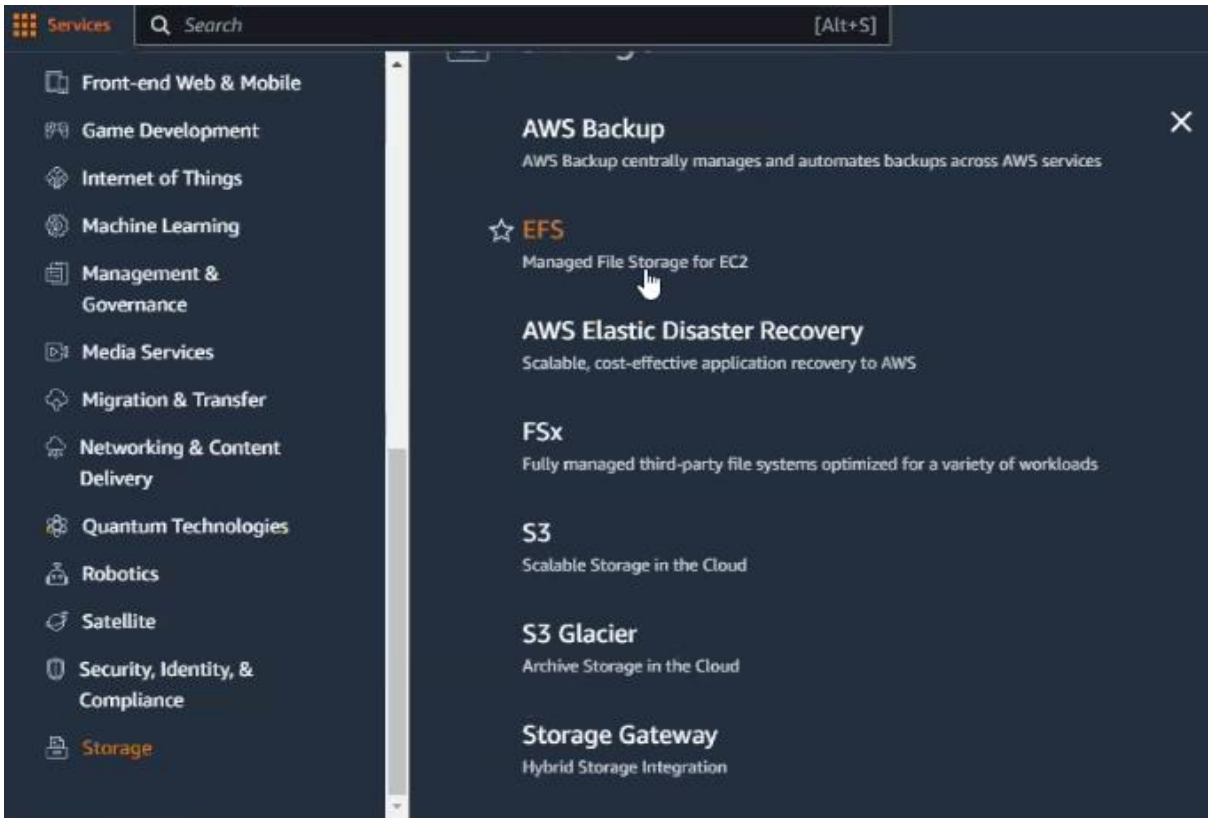
(*) source type -- anywhere

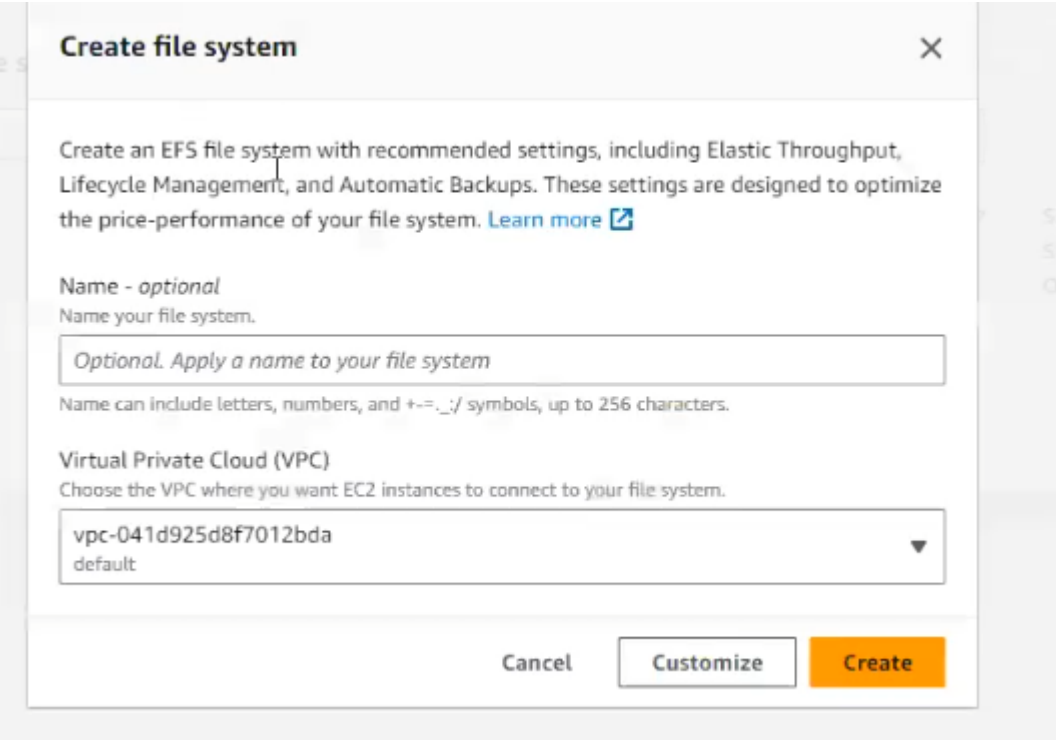


(*) volume 8gb --- launch instance

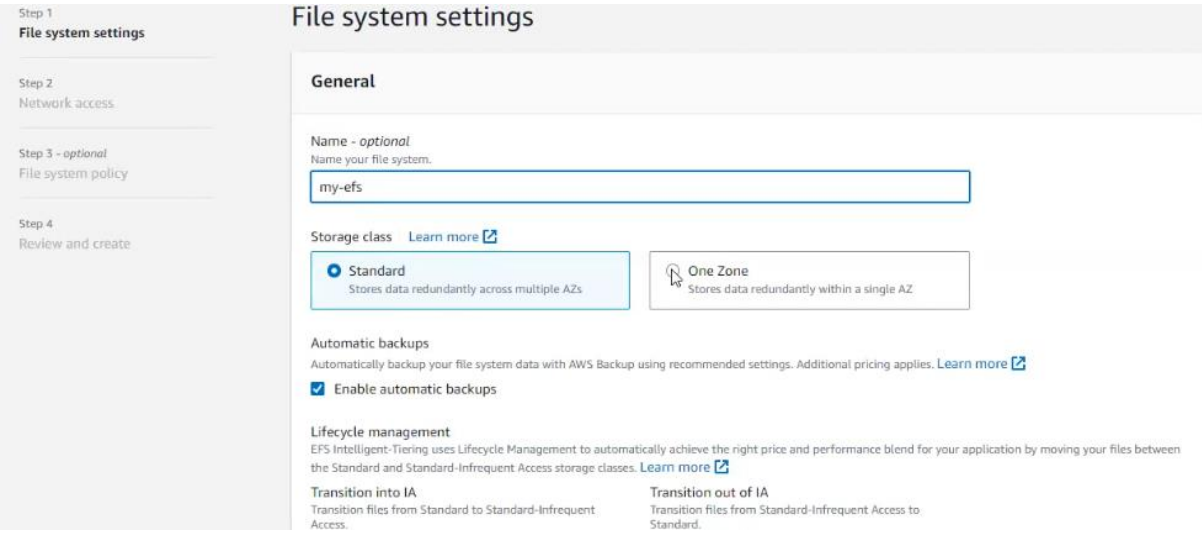
(*) 2 instances with same configuration created

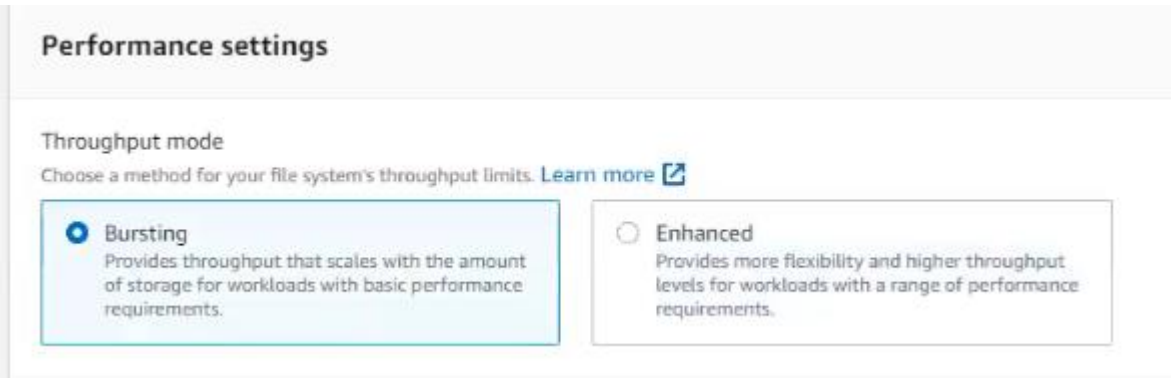
(*) create EFS STORAGE -- EFS





(*) customize





In enhanced we can select the data transferring speed --- provision

Throughput mode

Choose a method for your file system's throughput limits. Learn more

Bursting
Provides throughput that scales with the amount of storage for workloads with basic performance requirements.

Enhanced
Provides more flexibility and higher throughput levels for workloads with a range of performance requirements.

Elastic (Recommended)
Use this mode for workloads with unpredictable I/O. With Elastic mode, your throughput scales automatically and you only pay for what you use.

Provisioned
Use this mode if you can estimate your workload's throughput requirements. With Provisioned mode, you configure your file system's throughput and pay for throughput provisioned.

Provisioned Throughput (MiB/s)

Add throughput value

Valid range is 1-1024 MiB/s

Maximum Read Throughput (MiB/s)

▼ Additional settings

Performance mode

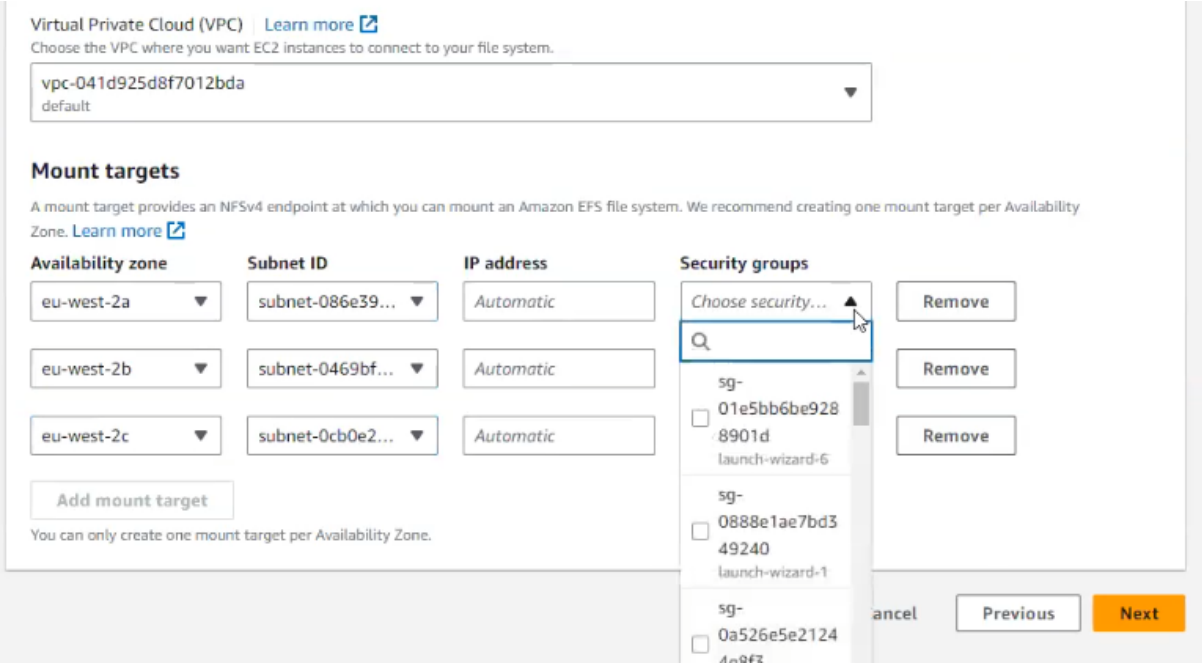
Set your file system's performance mode based on IOPS required. Learn more

General Purpose (Recommended)
Ideal for a variety of diverse workloads, including high performance and latency-sensitive applications

Max I/O
Designed for highly parallelized workloads that can tolerate higher latencies

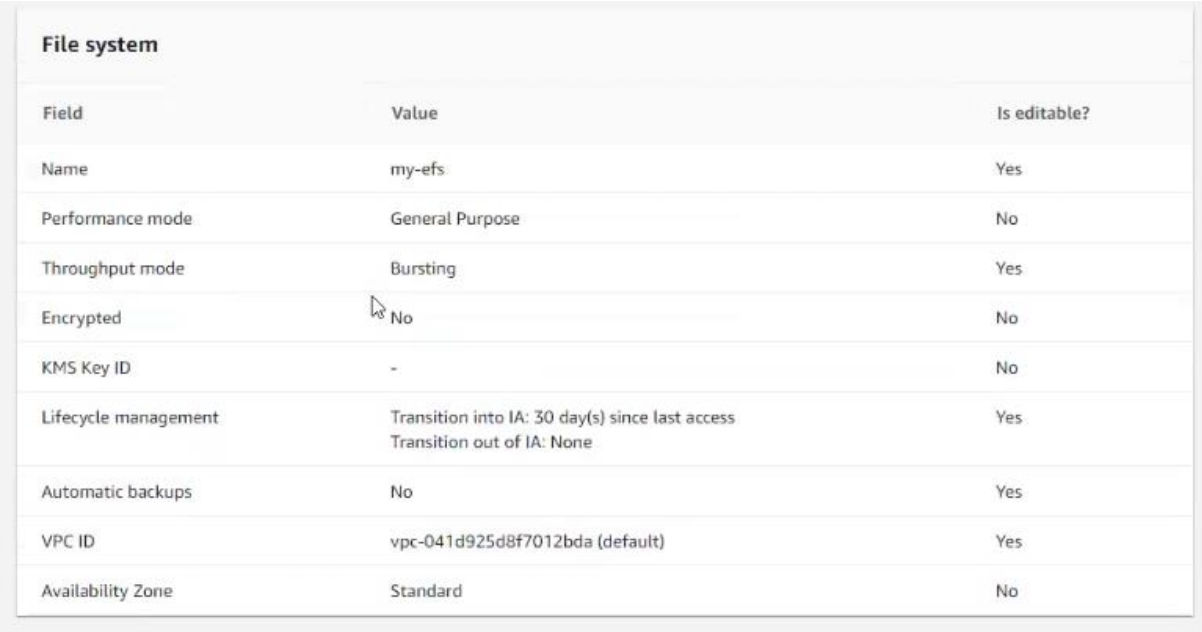
(*) network access

(*) mount targets – security groups -- nfs



(*) file system policy next

(*) review-- create



File system

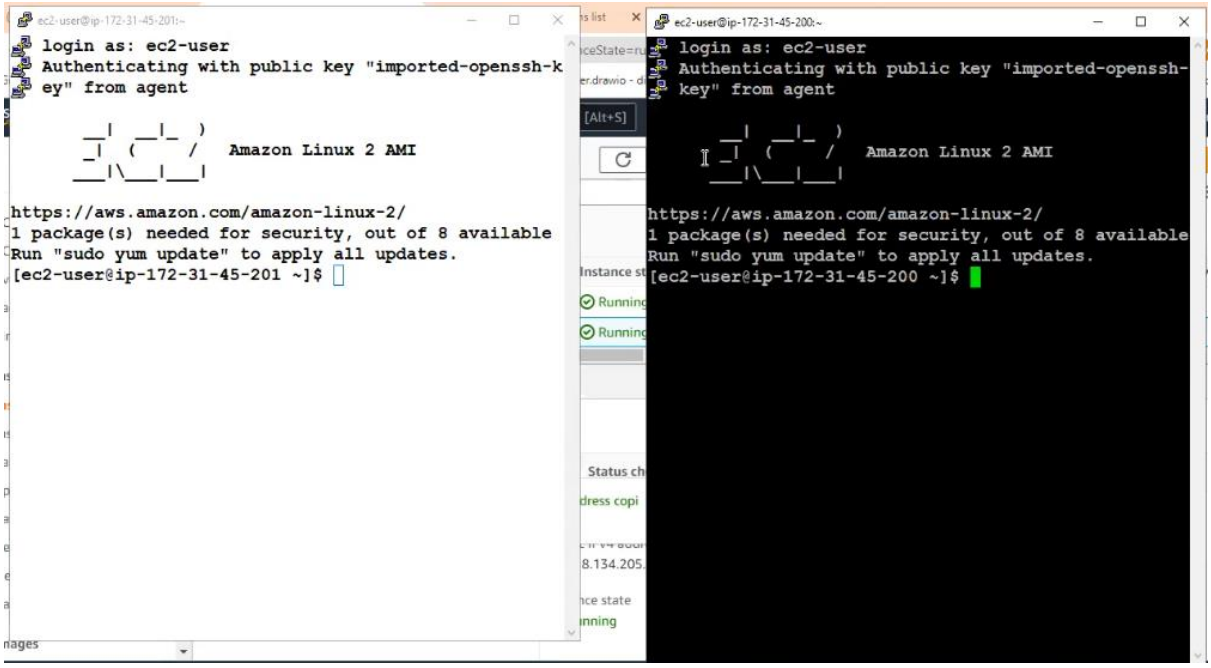
Field	Value	Is editable?
Name	my-efs	Yes
Performance mode	General Purpose	No
Throughput mode	Bursting	Yes
Encrypted	No	No
KMS Key ID	-	No
Lifecycle management	Transition into IA: 30 day(s) since last access Transition out of IA: None	Yes
Automatic backups	No	Yes
VPC ID	vpc-041d925d8f7012bda (default)	Yes
Availability Zone	Standard	No

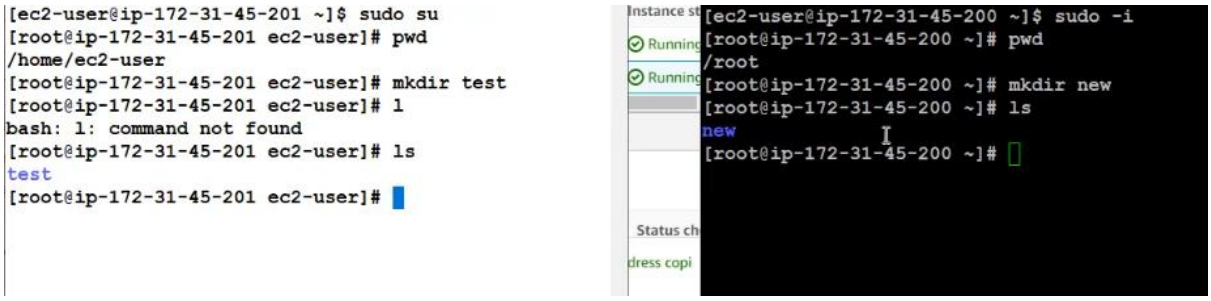
(*) file system created

Mounting the file system

(*) login to server A using putty white

(*) login to server b using putty black

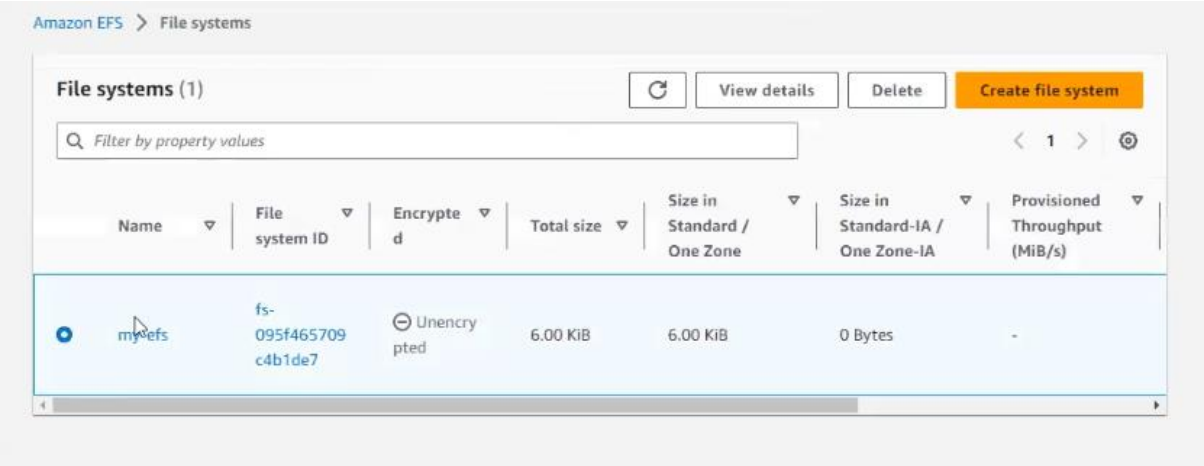




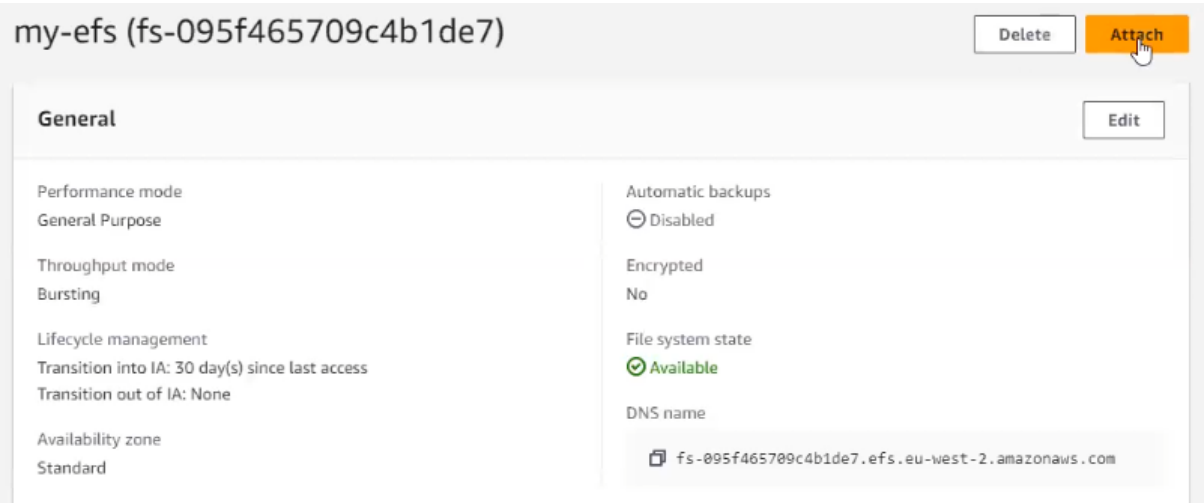
Create directories A TEST

B NEW

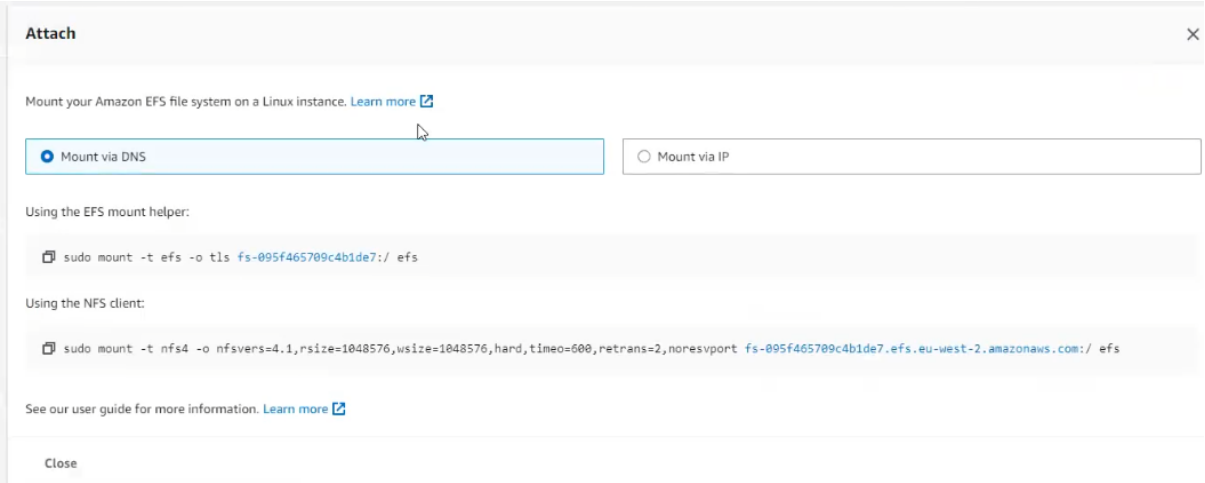
THEN ATTACH THE FILE SYSTEM



(*) SELECT ATTACH



(*) connect using NFS CLIENT copy the public link and paste it in machine A instead of efs change the directory name to test




```
[root@ip-172-31-45-201 ec2-user]# ls
test
[root@ip-172-31-45-201 ec2-user]# sudo mount -t nfs4
-o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo
=600,retrans=2,noresvport fs-095f465709c4b1de7.efs.eu
-west-2.amazonaws.com:/ test
```

(*) for machine B copy and paste the public link instead of efs change the directory name to new

```
[root@ip-172-31-45-200 ~]# ls
new
[root@ip-172-31-45-200 ~]# sudo mount -t nfs4 -o nfs
vers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,
retrans=2,noresvport fs-095f465709c4b1de7.efs.eu-wes
t-2.amazonaws.com:/ new
[root@ip-172-31-45-200 ~]#
```

(*) nothing in both the directories

```
[root@ip-172-31-45-201 ec2-user]# cd test/
[root@ip-172-31-45-201 test]# ls
[root@ip-172-31-45-201 test]#
```

```
[root@ip-172-31-45-200 ~]# cd new/
[root@ip-172-31-45-200 new]# ls
[root@ip-172-31-45-200 new]# ls
[root@ip-172-31-45-200 new]#
```

(*) then create a file in machine A NAMED hii check the same in machine b it will show the same file

(*) create a new file in machine B TOUCH HELLO it will show in machine A

```
root@ip-172-31-45-201:/home/ec2-user/test
[root@ip-172-31-45-201 test]# touch hii
[root@ip-172-31-45-201 test]# ls
hii
[root@ip-172-31-45-201 test]# ls
hello  hii
[root@ip-172-31-45-201 test]#
```

```
root@ip-172-31-45-200:~/new
[root@ip-172-31-45-200 new]# ls
hii
[root@ip-172-31-45-200 new]# touch hello
[root@ip-172-31-45-200 new]# ls
hello  hii
[root@ip-172-31-45-200 new]#
```

It is called common file system sharing