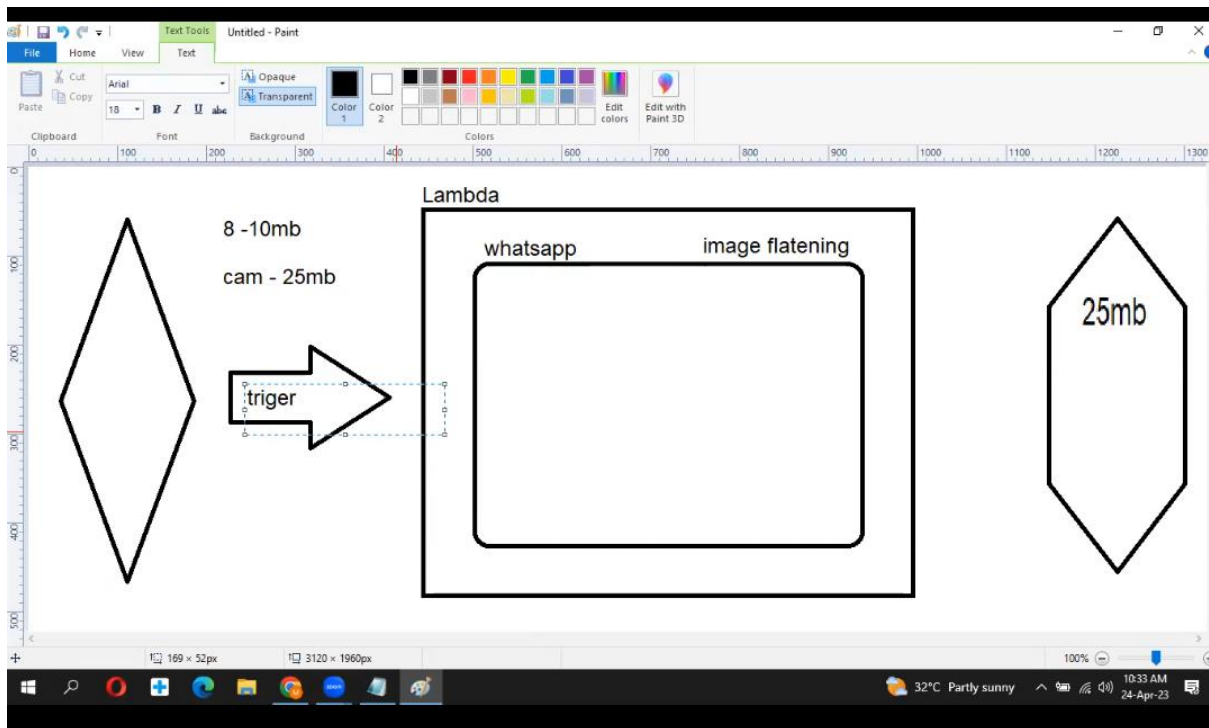


LAMBDA

LAMBDA SERVERLESS developer tool



LAMBDA RUNS ON TRIGGER



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

The screenshot shows the 'Create function' page in the AWS Lambda console. The breadcrumb navigation at the top is 'Lambda > Functions > Create function'. Below the title 'Create function' is a link to 'AWS Serverless Application Repository applications have moved to Create application.' There are three tabs: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The 'Basic information' section includes a 'Function name' field with 'myfunction', a 'Runtime' dropdown set to 'Python 3.10', and an 'Architecture' dropdown set to 'x86_64'. The 'Permissions' section has a 'Change default execution role' link and radio buttons for 'Create a new role with basic Lambda permissions' (selected), 'Use an existing role', and 'Create a new role from AWS policy templates'. A note at the bottom states: 'Lambda will create an execution role named myfunction-role-85tk3for, with permission to upload logs to Amazon CloudWatch Logs.'

(*) IAM Dashboard create roles

The screenshot shows the AWS IAM dashboard. The left sidebar is titled 'Identity and Access Management (IAM)' and contains a search bar and a menu with 'Dashboard' (selected), 'Access management', 'User groups', 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Access reports', 'Access analyzer', and 'Archive rules'. The main content area is titled 'IAM dashboard' and features 'Security recommendations' with a red badge showing '2'. The recommendations listed are: 'Add MFA for root user' (with an 'Add MFA' button), 'Root user has no active access keys' (with a green checkmark), and 'Update your access permissions for AWS Billing, Cost Management, and Account consoles' (with a 'View affected policies' button).

(*) create roles

Roles (18) [Info](#)

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

< 1 > [Settings](#)

<input type="checkbox"/>	Role name	Trusted entities	Last ar
<input type="checkbox"/>	ansible2	AWS Service: ec2	52 days
<input type="checkbox"/>	aws-elasticbeanstalk-ec2-role	AWS Service: ec2	11 days
<input type="checkbox"/>	aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	11 days
<input type="checkbox"/>	AWSCodePipelineServiceRole-eu-west-2-cp-role	AWS Service: codepipeline	24 days

(*) use case lambda

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

- ☐ EC2
Allows EC2 instances to call AWS services on your behalf.
- ☒ Lambda
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

[Cancel](#)

[Next](#)

(*) EC2 full access

Permissions policies (845) [Info](#)

Choose one or more policies to attach to your new role.

28 matches < 1 2 > [Settings](#)

[X](#) [Clear filters](#)

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	AmazonEC2FullAccess	AWS m...	Provides full access to Amazon EC2 via the AWS Managemen...
<input type="checkbox"/>	AmazonEC2RoleforSSM	AWS m...	This policy will soon be deprecated. Please use AmazonSSM...
<input type="checkbox"/>	AmazonEC2RoleforAWSCodeD...	AWS m...	Provides EC2 access to S3 bucket to download revision. This ...
<input type="checkbox"/>	AmazonEC2ContainerRegistry	AWS m	Provides administrative access to Amazon ECR resources

(*) vpc full access

Add permissions [Info](#)

Permissions policies (Selected 1/845) [Info](#)
Choose one or more policies to attach to your new role.

10 matches < 1 >

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	AmazonVPCReadOnlyAccess	AWS m...	Provides read only access to Amazon VPC via the AWS Mana...
<input type="checkbox"/>	AmazonVPCCrossAccountNet...	AWS m...	Provides access to create network interfaces and attach them ...
<input type="checkbox"/>	AmazonVPCFullAccess	AWS m...	Provides full access to Amazon VPC via the AWS Managemen...

(*) name and review

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+-=, @ _ .' characters.

Description
Add a short explanation for this role.

Maximum 1000 characters. Use alphanumeric and '+-=, @ _ .' characters.

(*) create role

Policy name Info	Type	Attached as
AmazonVPCFullAccess	AWS managed	Permissions policy
AmazonEC2FullAccess	AWS managed	Permissions policy

Tags

Add tags - optional [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

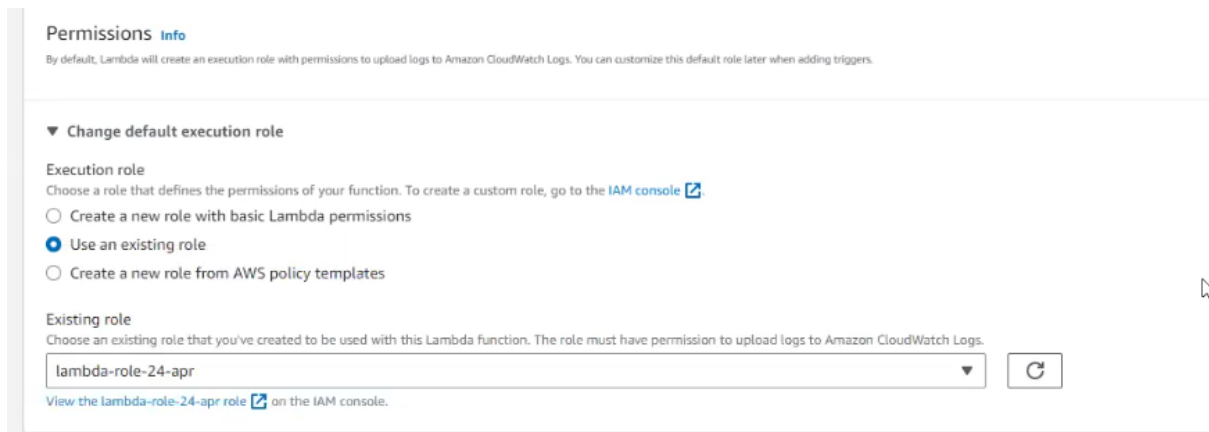
You can add up to 50 more tags.

[Cancel](#)

[Previous](#)

[Create role](#)

(*) in permissions apply IAM ROLE – CREATE function



Permissions [Info](#)

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ **Change default execution role**

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions

☒ Use an existing role

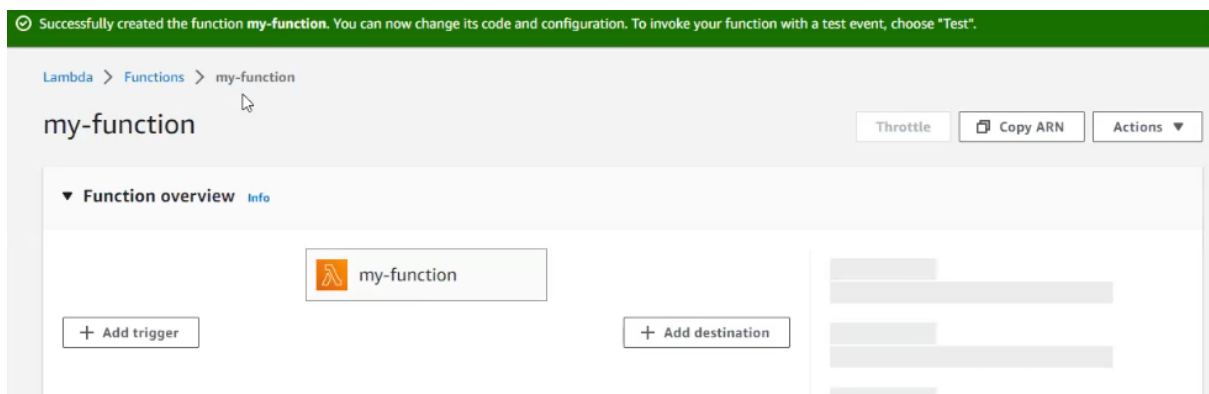
☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

lambda-role-24-apr

[View the lambda-role-24-apr role](#) on the IAM console.

(*) function has been created




Successfully created the function **my-function**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > my-function

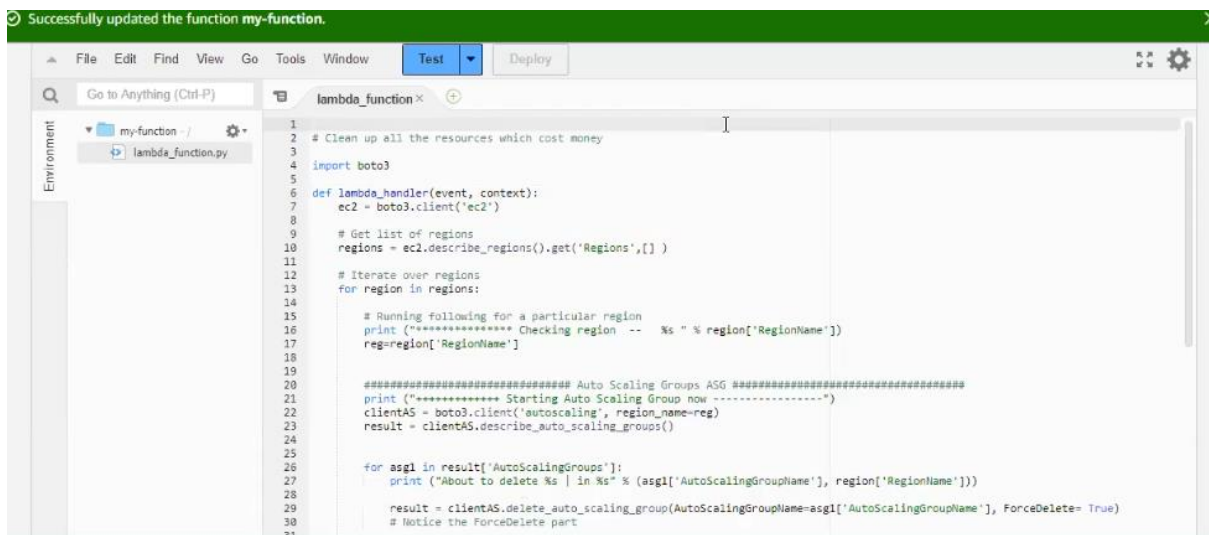
my-function [Throttle](#) [Copy ARN](#) [Actions](#)

▼ **Function overview** [Info](#)

 my-function

[+ Add trigger](#) [+ Add destination](#)

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



Successfully updated the function **my-function**.

File Edit Find View Go Tools Window [Test](#) [Deploy](#)

Go to Anything (Ctrl-P)

Environment

- my-function - /
- lambda_function.py

```
1 # Clean up all the resources which cost money
2
3
4 import boto3
5
6 def lambda_handler(event, context):
7     ec2 = boto3.client('ec2')
8
9     # Get list of regions
10    regions = ec2.describe_regions().get('Regions',[])
11
12    # Iterate over regions
13    for region in regions:
14
15        # Running following for a particular region
16        print ("***** Checking region -- %s " % region['RegionName'])
17        reg=region['RegionName']
18
19
20    ##### Auto Scaling Groups ASG #####
21    print ("***** Starting Auto Scaling Group Row -----")
22    clientAS = boto3.client('autoscaling', region_name=reg)
23    result = clientAS.describe_auto_scaling_groups()
24
25
26    for asgl in result['AutoScalingGroups']:
27        print ("About to delete %s | in %s" % (asgl['AutoScalingGroupName'], region['RegionName']))
28
29        result = clientAS.delete_auto_scaling_group(AutoScalingGroupName=asgl['AutoScalingGroupName'], ForceDelete= True)
30        # Notice the ForceDelete part
31
```

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

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

Lambda > Add trigger

Add trigger

Trigger configuration [Info](#)

Select a source ▼

Cancel Add

(*) event trigger

Event bridge uses cron

Cron is used for scheduling

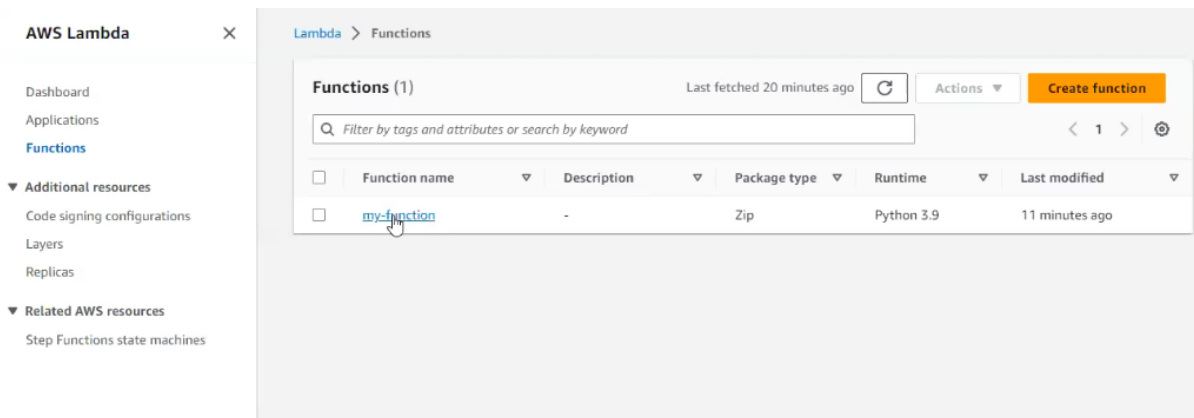
"At 18:00 on Friday."

next at 2023-04-28 18:00:00 random

00 18 * * 5

minute	hour	day (month)	month	day (week)
*				
	*			
		*		
			*	
				*
@yearly				
@annually				
@monthly				
@weekly				
@daily				
@hourly				
@reboot				

(*) go to lambda functions



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

(*) go to IAM select existing role

<input type="checkbox"/>	lambda-role	AWS Service: lambda	2 days ago
<input type="checkbox"/>	lambda-role-24-apr	AWS Service: lambda	-

(*) attach policies

Permissions policies (2) Info

You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter

Policy name	Type	Description
<input type="checkbox"/> AmazonEC2FullAccess	AWS managed	Provides full access to Amazon EC2 via the AWS Management Console.
<input type="checkbox"/> AmazonVPCFullAccess	AWS managed	Provides full access to Amazon VPC via the AWS Management Console.

Permissions boundary - (not set) Info

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others.

Set permissions boundary











Simulate Remove

Add permissions

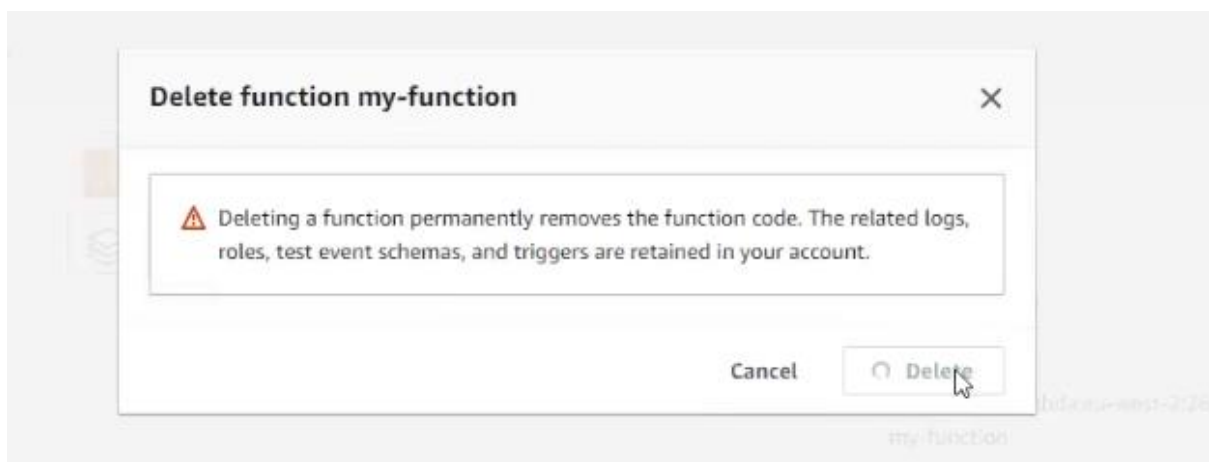
Attach policies

Create inline policy

(*) cloud watch full access—add permissions

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

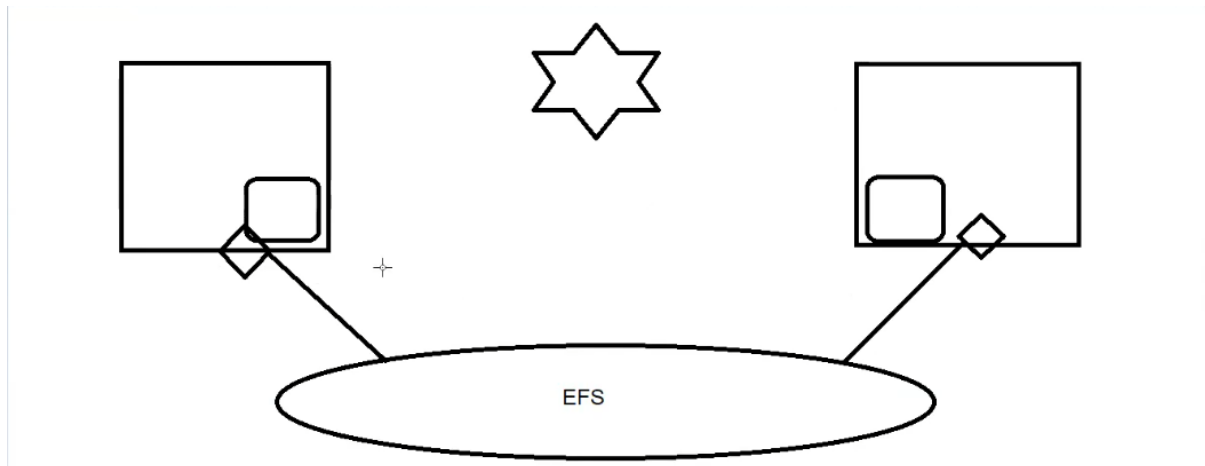
(*) action-delete function



ELASTIC FILE SYSTEM EFS

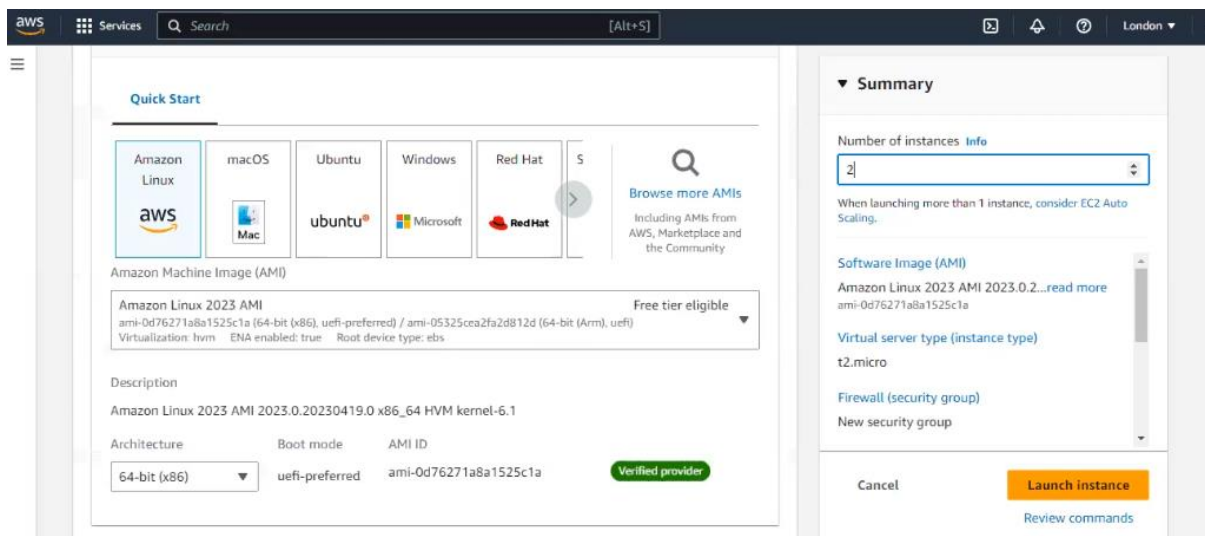
Common file system

Storage --- EFS



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

(*) NFS network file sharing



(*) create security group NFS

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group


We'll create a new security group called 'launch-wizard-15' with the following rules:

☒ Allow SSH traffic from
Helps you connect to your instance

Anywhere
0.0.0.0/0

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☐ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

 Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. [X](#)

(*) Network settings edit

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

(*) source type -- anywhere

0.0.0.0/0 [X](#)

▼ Security group rule 2 (TCP, 2049, 0.0.0.0/0)

[Remove](#)

Type [Info](#)

NFS

Protocol [Info](#)

TCP

Port range [Info](#)

2049

Source type [Info](#)

Anywhere

Source [Info](#)

[Add CIDR, prefix list or security](#)

0.0.0.0/0 [X](#)

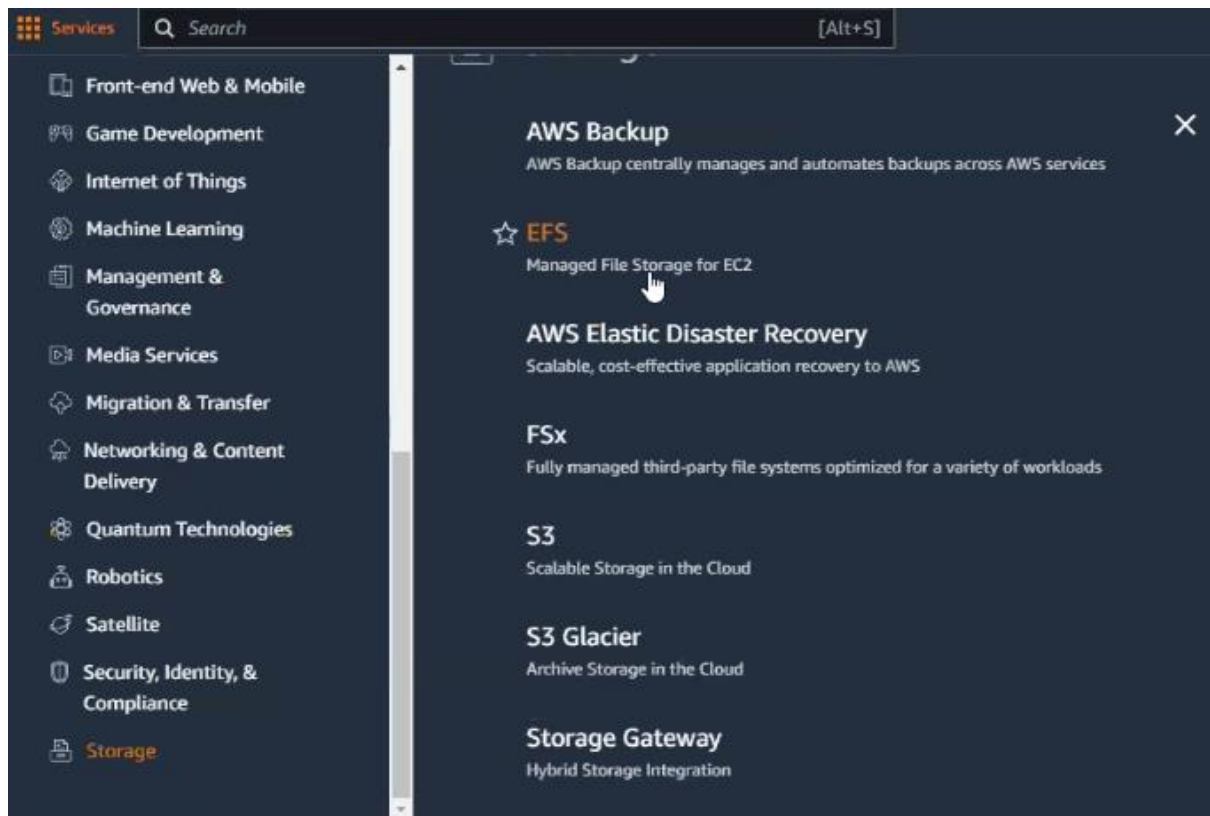
Description - optional [Info](#)

e.g. SSH for admin desktop

(*) volume 8gb --- launch instance

(*) 2 instances with same configuration created

(*) create EFS STORAGE -- EFS



Create file system

×

Create an EFS file system with recommended settings, including Elastic Throughput, Lifecycle Management, and Automatic Backups. These settings are designed to optimize the price-performance of your file system. [Learn more](#)

Name - optional
Name your file system.

Optional. Apply a name to your file system

Name can include letters, numbers, and +-=._:/ symbols, up to 256 characters.

Virtual Private Cloud (VPC)
Choose the VPC where you want EC2 instances to connect to your file system.

vpc-041d925d8f7012bda
default

Cancel

Customize

Create

(*) customize

Step 1
File system settings

Step 2
Network access

Step 3 - optional
File system policy

Step 4
Review and create

File system settings

General

Name - optional
Name your file system.

my-efs

Storage class [Learn more](#)

☒ Standard
Stores data redundantly across multiple AZs

☐ One Zone
Stores data redundantly within a single AZ

Automatic backups
Automatically backup your file system data with AWS Backup using recommended settings. Additional pricing applies. [Learn more](#)

☒ Enable automatic backups

Lifecycle management
EFS Intelligent-Tiering uses Lifecycle Management to automatically achieve the right price and performance blend for your application by moving your files between the Standard and Standard-Infrequent Access storage classes. [Learn more](#)

Transition into IA
Transition files from Standard to Standard-Infrequent Access.

Transition out of IA
Transition files from Standard-Infrequent Access to Standard.

Performance settings

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.

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

Virtual Private Cloud (VPC) [Learn more](#)

Choose the VPC where you want EC2 instances to connect to your file system.

vpc-041d925d8f7012bda
default

Mount targets

A mount target provides an NFSv4 endpoint at which you can mount an Amazon EFS file system. We recommend creating one mount target per Availability Zone. [Learn more](#)

Availability zone	Subnet ID	IP address	Security groups	
eu-west-2a	subnet-086e39...	Automatic	Choose security...	Remove
eu-west-2b	subnet-0469bf...	Automatic	<input type="text"/>	Remove
eu-west-2c	subnet-0cb0e2...	Automatic	<ul style="list-style-type: none"><input type="checkbox"/> sg-01e5bb6be9288901d launch-wizard-6<input type="checkbox"/> sg-0888e1ae7bd349240 launch-wizard-1<input type="checkbox"/> sg-0a526e5e2124 launch-wizard-2	Remove

[Add mount target](#)

You can only create one mount target per Availability Zone.

[Cancel](#) [Previous](#) [Next](#)

(*) file system policy next

(*) review-- create

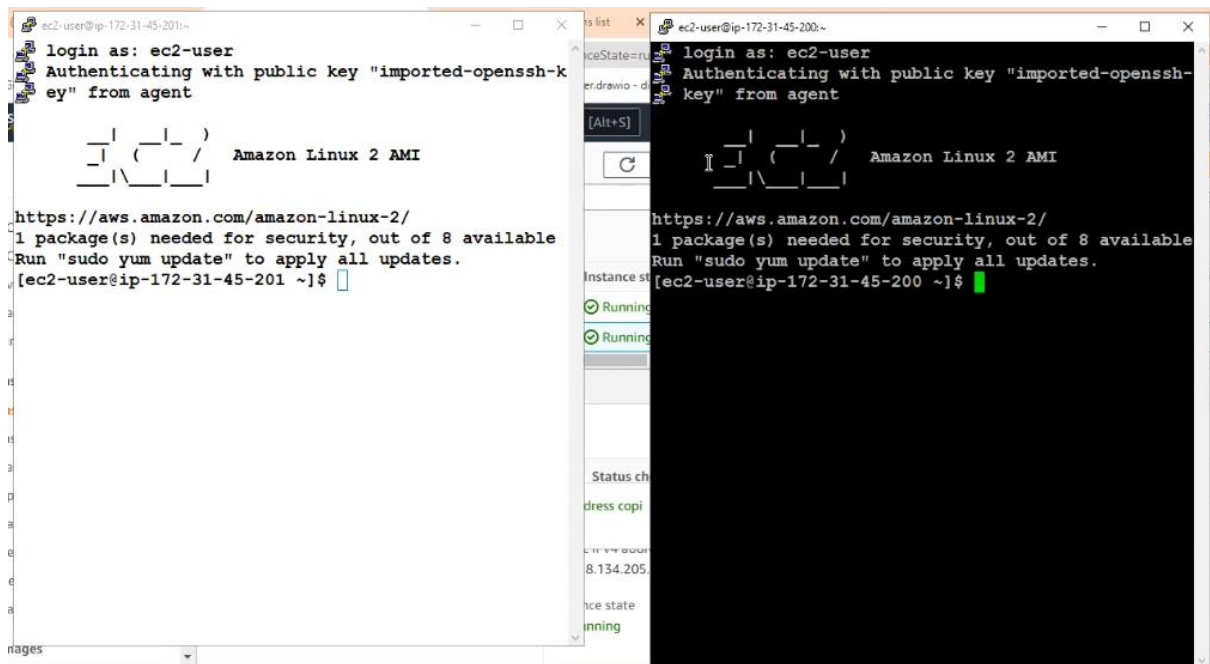
File system		
Field	Value	Is editable?
Name	my-efs	Yes
Performance mode	General Purpose	No
Throughput mode	Bursting	Yes
Encrypted	No	No
KMS Key ID	-	No
Lifecycle management	Transition into IA: 30 day(s) since last access Transition out of IA: None	Yes
Automatic backups	No	Yes
VPC ID	vpc-041d925d8f7012bda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



```

[ec2-user@ip-172-31-45-201 ~]$ sudo su
[root@ip-172-31-45-201 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-45-201 ec2-user]# mkdir test
[root@ip-172-31-45-201 ec2-user]# l
bash: l: command not found
[root@ip-172-31-45-201 ec2-user]# ls
test
[root@ip-172-31-45-201 ec2-user]#

```

```

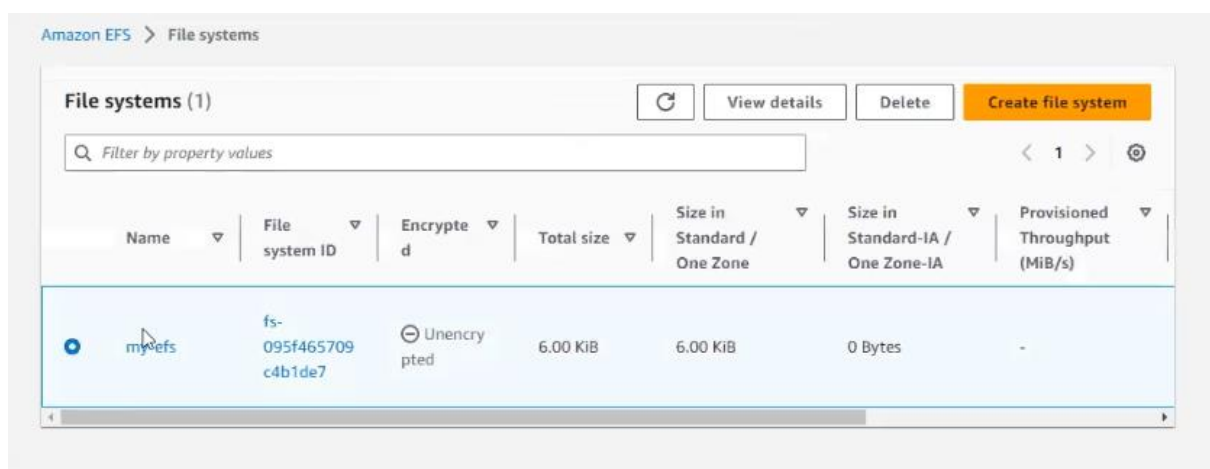
[ec2-user@ip-172-31-45-200 ~]$ sudo -i
[root@ip-172-31-45-200 ~]# pwd
/root
[root@ip-172-31-45-200 ~]# mkdir new
[root@ip-172-31-45-200 ~]# ls
new
[root@ip-172-31-45-200 ~]#

```

Create directories A TEST

B NEW

THEN ATTACH THE FILE SYSTEM



(*) SELECT ATTACH

The screenshot shows the AWS EFS console for a file system named 'my-efs' with ID 'fs-095f465709c4b1de7'. At the top right are 'Delete' and 'Attach' buttons. Below is a 'General' tab with an 'Edit' button. The 'General' section is divided into two columns. The left column contains: Performance mode (General Purpose), Throughput mode (Bursting), Lifecycle management (Transition into IA: 30 day(s) since last access, Transition out of IA: None), and Availability zone (Standard). The right column contains: Automatic backups (Disabled), Encrypted (No), File system state (Available, indicated by a green checkmark), and DNS name (fs-095f465709c4b1de7.efs.eu-west-2.amazonaws.com).

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

The screenshot shows the 'Attach' dialog box for the EFS file system. It has a title bar 'Attach' and a close button. The main text says 'Mount your Amazon EFS file system on a Linux instance. [Learn more](#)'. Below this are two radio buttons: 'Mount via DNS' (selected) and 'Mount via IP'. Under 'Mount via DNS', there are two code blocks. The first is 'Using the EFS mount helper:' with the command: `sudo mount -t efs -o tls fs-095f465709c4b1de7:/ efs`. The second is 'Using the NFS client:' with the command: `sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport fs-095f465709c4b1de7.efs.eu-west-2.amazonaws.com:/ efs`. At the bottom, there is a 'Close' button and a link to 'See our user guide for more information. [Learn more](#)'.

```
[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,wsiz=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,rsz=1048576,wsz=1048576,hard,timeo=600,
retrans=2,norevport 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