

## IAM IDENTITY ACCESS MANAGEMENT SERVICE

### SECURITY ENGINEERING – USER LEVEL OF SECURITY – GLOBAL SPECIFIC SERVICE

user creation

Group creation

Policy

Roles

(\*) Search IAM

(\*) create user group



(\*) add account alias

Account ID

 509636718401

Account Alias

kathireshev **Edit** | **Delete**

Sign-in URL for IAM users in this account

 <https://kathireshev.signin.aws.amazon.com/console>

## (\*) create user group

IAM > User groups > Create user group

### Create user group

#### Name the group

##### User group name

Enter a meaningful name to identify this group.

10amec2




Maximum 128 characters. Use alphanumeric and "+, @, \_" characters.

## (\*) Give permissions as per the client needs client needs full ec2 permission in the search box type ec2 and select amazon ec2 full access permissions.

**Attach permissions policies - *Optional*** (Selected 1/842) [Info](#)

You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

28 matches

<input type="checkbox"/>	Policy name	Type	Description
<input checked="" type="checkbox"/>	 AmazonEC2FullAccess	AWS managed	Provides full access to Am
<input type="checkbox"/>	 AmazonEC2RoleforSSM	AWS managed	This policy will soon be dep
<input type="checkbox"/>	 AmazonEC2RoleforAWSCodeDeploy	AWS managed	Provides EC2 access to S3

## (\*) press create group. group created

## (\*) create users

IAM > Users

**Users (0)** [Info](#)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

<input type="checkbox"/>	User name	Groups	Last activity	MFA	Password age	Active key age
No resources to display						

## (\*) click on add users and enter the name of the user

**Specify user details**

**User details**

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, =, @, \_ (hyphen)

☐ Provide user access to the AWS Management Console - optional

If you're providing console access to a person, it's a best practice [to](#) manage their access in IAM Identity Center.

☐ If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

(\*) create user name provide user access to AWS management console and password for the user

### Specify user details

#### User details

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = \_ . - (hyphen)

☒ Provide user access to the AWS Management Console - optional

If you're providing console access to a person, it's a best practice [to](#) manage their access in IAM Identity Center.

**Are you providing console access to a person?**

☐ Specify a user in Identity Center - Recommended

We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

☒ I want to create an IAM user

We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

#### Console password

☐ Autogenerated password

You can view the password after you create the user.

☒ Custom password

Enter a custom password for the user:

• Must be at least 8 characters long

• Must include at least three of the following mix of character types: uppercase letters [A-Z], lowercase letters [a-z], numbers [0-9], and symbols [!@#\$%^&\*(){}~.-+;:"]

☐ Show password

☒ Users must create a new password at next sign-in (recommended).

Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

**If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user.** [Learn more](#)

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

(\*) set permissions add user to group and press next

### Set permissions

Add user to an existing group or create a new one. Using groups is a best practice way to manage user's permissions by job functions. [Learn more](#)

#### Permissions options

☒ Add user to group

Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions

Copy all group memberships, attached managed policies, and inline policies from an existing user.

☐ Attach policies directly

Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

#### User groups (1/1)

Search groups

Group name	Users	Attached policies	Created
<input checked="" type="checkbox"/> 10amec2group	0	<a href="#">AmazonEC2FullAccess</a>	2023-04-28 (6 minutes ago)

[Create group](#)

**Permissions boundary - optional**

Set a permissions boundary to control the maximum permissions for this user. Use this advanced feature used to delegate permission management to others. [Learn more](#)

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

(\*) review and create – create user

### Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

#### User details

User name	Console password type	Require password reset
varshini	Custom password	Yes

#### Permissions summary

Name	Type	Used as
<a href="#">10amec2group</a>	Group	Permissions group
<a href="#">IAMUserChangePassword</a>	AWS managed	Permissions policy

**Tags - optional**

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

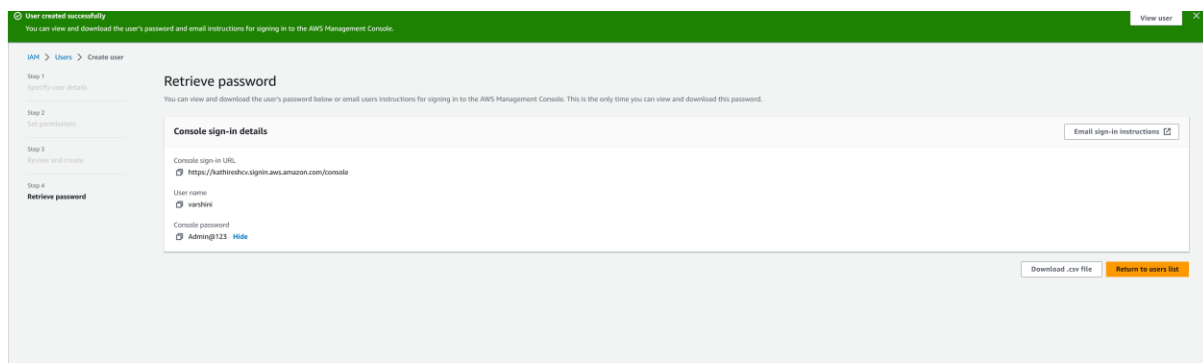
No tags associated with the resource.

[Add new tag](#)

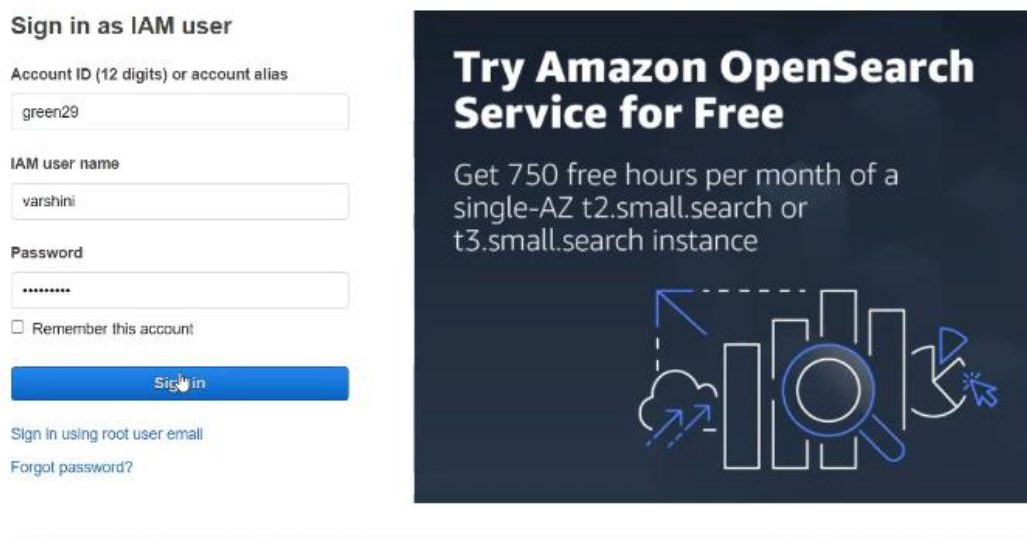
You can add up to 50 more tags.

[Cancel](#) [Previous](#) [Create user](#)

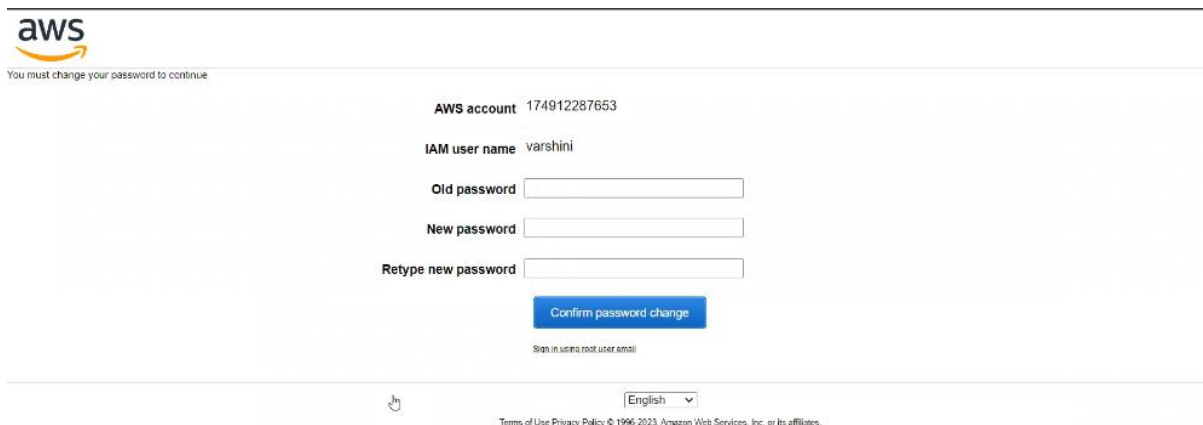
(\*) user created successfully



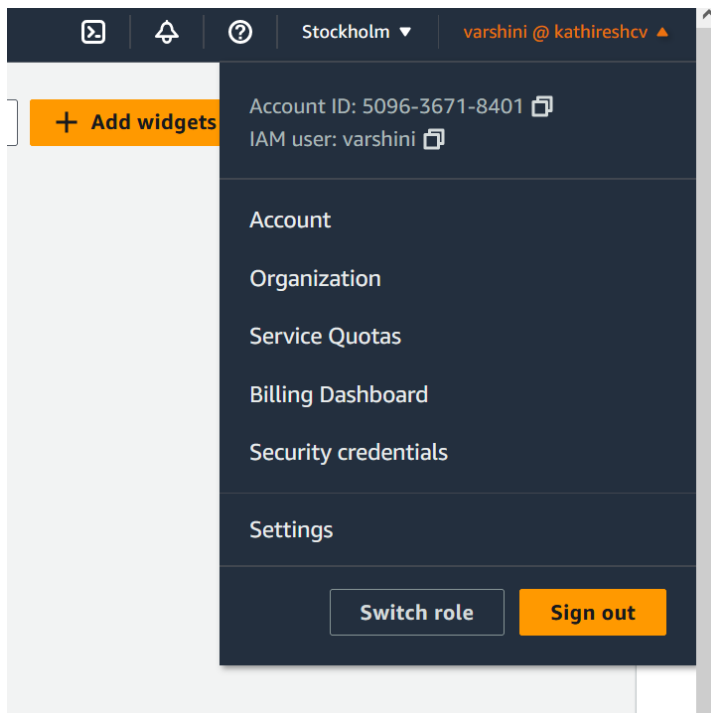
(\*) open incognito tab paste the url in console sign in details login to the account



(\*) change the password to continue



(\*) IAM USER



(\*) now the user is created with ec2 full access permissions user can access only EC2 and the user cannot access S3 or other components.

## POLICIES

### 800+ POLICIES IN AWS

IAM > Policies

Policies (1081) [info](#)

A policy is an object in AWS that defines permissions.

Filter policies by property or policy name and press enter. 38 matches

Properties [Clear filters](#)

Type	name	Type	Used as	Description
Used as	AmazonEC2FullAccess	AWS managed	Permissions policy (...)	Provides full access to Amazon EC2 via the AWS Management Console.
<input type="radio"/>	AmazonEC2RoleforSSM	AWS managed	None	This policy will soon be deprecated. Please use AmazonSSMManagedInstanceCore policy to enable AWS Systems Manager service core functionality on EC2 instances. For more information see https://doc...
<input type="radio"/>	AmazonEC2RoleforAWSCodeDeploy	AWS managed	None	Provides EC2 access to S3 bucket to download revision. This role is needed by the CodeDeploy agent on EC2 instances.
<input type="radio"/>	AmazonEC2ContainerRegistryFullAccess	AWS managed	None	Provides administrative access to Amazon ECR resources
<input type="radio"/>	AmazonEC2ContainerRegistryReadOnly	AWS managed	None	Provides read-only access to Amazon ECR Container Registry repositories.
<input type="radio"/>	AmazonElasticMapReduceforEC2Role	AWS managed	None	Default policy for the Amazon Elastic MapReduce for EC2 service role.
<input type="radio"/>	AmazonEC2ReadOnlyAccess	AWS managed	None	Provides read only access to Amazon EC2 via the AWS Management Console.

(\*) create policy

(\*) choose a service

(\*) select customizable permissions

(\*) client requested S3 list buckets

## Create policy

1 2 3

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editorJSON

Import managed policy

Expand allCollapse all

S3 (1 action)CloneRemove

ServiceS3

Actions

Specify the actions allowed in S3

Filter actions

Manual actions (add actions)

All S3 actions (s3:\*)

Access level

List (1 selected)Expand allCollapse all

ListAccessPoints

ListAccessPointsForObjectLambda

ListAllMyBuckets

ListBucket

ListBucketMultipartUploads

ListBucketVersions

ListJobs

ListMultiRegionAccessPoints

ListStorageLensConfigurations

ListMultipartUploadParts

Read

Tagging

Write

Permissions management

Resources

The actions you chose support all resources.

Request conditionsSpecify request conditions (optional)

Add additional permissions

Character count: 125 of 6,144.

CancelNext: Tags

(\*) review policy add name for the policy – create policy

## Create policy

1 2 3

### Review policy

Name\* varshinis3policy

Use alphanumeric and \*+\_,@- characters. Maximum 128 characters.

Description varshinis3policy

Maximum 1000 characters. Use alphanumeric and \*+\_,@- characters.

Summary

Filter

Service	Access level	Resource	Request condition
Allow (1 of 374 services) Show remaining 373			
S3	Limited: List	All resources	None

Tags

Key	Value
-----	-------

No tags associated with the resource.

\* Required

CancelPreviousCreate policy

(\*) applying permission to the user – go to user -- select user – add permissions

The screenshot shows the AWS IAM console interface for a user named 'varshini'. The left sidebar contains navigation links for Identity and Access Management (IAM), Access management, Access reports, and IAM console. The main content area displays the user's profile, including their ARN, creation date, and console access status. Below this, the 'Permissions' tab is active, showing a list of permissions policies attached to the user. The policies listed are 'AmazonEC2FullAccess' and 'IAMUserChangePassword', both managed by AWS. A 'Permissions boundary' section is also visible, indicating that no boundary is currently set for the user.

**Summary**

ARN: `arn:aws:iam::509636718401:user/varshini`  
Created: April 28, 2023, 22:12 (UTC+05:30)  
Console access: **Enabled without MFA**  
Last console sign-in: **Today**  
Access key 1: **Not enabled**  
Access key 2: **Not enabled**

**Permissions policies (2)**

Policy name	Type	Attached via
<a href="#">AmazonEC2FullAccess</a>	AWS managed	Group <a href="#">10arnec2group</a>
<a href="#">IAMUserChangePassword</a>	AWS managed	Directly

**Permissions boundary (not set)**

Set a permissions boundary to control the maximum permissions for this user. Use this advanced feature used to delegate permission management to others. [Learn more](#)

**Generate policy based on CloudTrail events**

You can generate a new policy based on the access activity for this user; then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. [Learn more](#)

[Generate policy](#)

No requests to generate a policy in the past 7 days.

(\*) add permission

The screenshot shows the 'Add permissions' step in the AWS IAM console for the user 'varshini'. The 'Permissions options' section is active, showing three options: 'Add user to group', 'Copy permissions', and 'Attach policies directly'. The 'Attach policies directly' option is selected. Below this, the 'Permissions policies' section shows a search for 'var' resulting in one match: 'varshiniPolicy', which is a 'Customer managed' policy. The 'Attach entities' column shows '0' entities attached. The 'Next' button is highlighted in orange.

**Permissions options**

- ☐ Add user to group: Add user to an existing group, or create a new one. We recommend using groups to manage user permissions by job function.
- ☐ Copy permissions: Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.
- ☒ **Attach policies directly**: Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

**Permissions policies (1/1083)**

Policy name	Type	Attached entities
<a href="#">varshiniPolicy</a>	Customer managed	0

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

(\*) attach policies directly

The screenshot shows the 'Attach policies directly' step in the AWS IAM console for the user 'varshini'. The 'Permissions options' section is active, showing three options: 'Add user to group', 'Copy permissions', and 'Attach policies directly'. The 'Attach policies directly' option is selected. Below this, the 'Permissions policies' section shows a search for 'var' resulting in one match: 'varshiniPolicy', which is a 'Customer managed' policy. The 'Attach entities' column shows '0' entities attached. The 'Next' button is highlighted in orange.

**Permissions options**


- ☐ Add user to group: Add user to an existing group, or create a new one. We recommend using groups to manage user permissions by job function.
- ☐ Copy permissions: Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.
- ☒ **Attach policies directly**: Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

**Permissions policies (1/1083)**

Policy name	Type	Attached entities
<a href="#">varshiniPolicy</a>	Customer managed	0

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

(\*) insufficient permission error

**Failed to create bucket**

To create a bucket, `s3:CreateBucket` permissions are required.

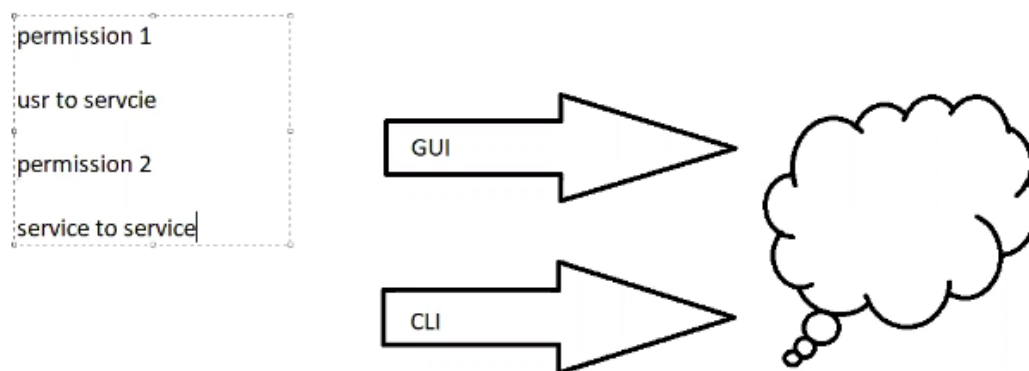
To apply the bucket owner enforced or bucket owner preferred setting for Object Ownership, `s3:PutBucketOwnershipControls` permissions are required.

View your permissions in the [IAM console](#). [Identity and Access Management in Amazon S3](#)

► API response

CancelCreate bucket

## ROLES



**Identity and Access Management (IAM)**

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings

Access reports

- Access analyzer
- Archive rules
- Analysers
- Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Related consoles

- [IAM Identity Center](#)
- [AWS Organizations](#)


**Roles (3)**

Search


Role name	Trusted entities	Last activity
<input type="checkbox"/> <a href="#">aws-elasticbeanstalk-ec2-role</a>	AWS Service: ec2	-
<input type="checkbox"/> <a href="#">aws-elasticbeanstalk-service-role</a>	AWS Service: elasticbeanstalk	-
<input type="checkbox"/> <a href="#">AWSServiceRoleForElasticLoadBalancing</a>	AWS Service: elasticloadbalancing (Service-Linked Role)	3 days ago
<input type="checkbox"/> <a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linked Role)	-
<input type="checkbox"/> <a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service-Linked Role)	-

**Roles Anywhere**


Manage

**Access AWS from your non AWS workloads**

Authenticate your non AWS workloads and securely provide access to AWS services.

**X.509 Standard**

Use your own existing PKI infrastructure or use [AWS Certificate Manager Private Certificate Authority](#) to authenticate identities.

**Temporary credentials**

Use temporary credentials with ease and benefit from the enhanced security they provide.



## (\*) Create a role – select trusted entity – aws service – use case – EC2 --Next

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

Select trusted entity [info](#)

Trusted entity type

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

☐ **AWS account**  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**  
Create a custom trust policy to enable others to perform actions in this account.

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:  

Choose a service to view use case

Cancel

Next

## (\*) add permissions S3 - next

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

Add permissions [info](#)

Permissions policies (Selected 1/843) [info](#)

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

Filter policies by property or policy name and press enter

10 matches

Policy name

Type

Description

☐ **varnishPolicy**

Custom...

varnishPolicy

☐ **AmazonDMSPRedshift...**

AWS ma...

Provides access to manage S3 settings for Redshift endpoints for DMS.

☒ **AmazonS3FullAccess**

AWS ma...

Provides full access to all buckets via the AWS Management Console.

☐ **QuickSightAccessF...**

AWS ma...

Policy used by QuickSight team to access customer data produced by S3 Storage Management Analytics.

☐ **AmazonS3ReadOnly...**

AWS ma...

Provides read only access to all buckets via the AWS Management Console.

☐ **AmazonS3Outposts...**

AWS ma...

Provides full access to Amazon S3 on Outposts via the AWS Management Console.

☐ **AWSBackupService...**

AWS ma...

Policy containing permissions necessary for AWS Backup to backup data in any S3 bucket. This includes read access to all S3 objects and any decrypt access for all KMS keys.

☐ **AWSBackupService...**

AWS ma...

Policy containing permissions necessary for AWS Backup to restore a S3 backup to a bucket. This includes read/write permissions to all S3 buckets, and permissions to GenerateDataKey and DescribeKey for all KMS keys.

☐ **AmazonS3ObjectLambda...**

AWS ma...

Provides AWS Lambda functions permissions to interact with Amazon S3 Object Lambda. Also grants Lambda permissions to write to CloudWatch Logs.

☐ **AmazonS3Outposts...**

AWS ma...

Provides read only access to Amazon S3 on Outposts via the AWS Management Console.

Set permissions boundary - optional [info](#)

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel

Previous

Next

## (\*) review and create – name – create role

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

10name3a3

Maximum 64 characters. Use alphanumeric and '\*'\_@\_- characters.

Description

Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '\*'\_@\_- characters.

Step 1: Select trusted entities

1- {

2- "Version": "2012-10-17",

3- "Statement": [

4- {

5- "Effect": "Allow",

6- "Action": [

7- "sts:AssumeRole"

8- ],

9- "Principal": {

10- "Service": [

11- "ec2.amazonaws.com"

12- ]

13- }

14- ]

15- }

16- }

Step 2: Add permissions

Permissions policy summary

Policy name

Type

Attached as

AmazonS3FullAccess

AWS managed

Permissions policy

(\*) Launch an instance – amazon linux – ppk keypair – advanced settings -- IAM instance profile -- 10amec2s3 – instance created

**Name and tags** [Info](#)

Name

with role

Add additional tags

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

Si

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

Free tier eligible

ami-0b3a4110c36b9a5f0 (64-bit (x86)) / ami-0a9112a9786fcfb8 (64-bit (Arm))

Virtualization: hvm    ENA enabled: true    Root device type: ebs

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20230418.0 x86\_64 HVM gp2

Architecture

AMI ID

64-bit (x86)

ami-0b3a4110c36b9a5f0

Verified provider

(\*) advance details 10amec2s3

▼ **Advanced details** [Info](#)

Purchasing option [Info](#)

☐ Request Spot Instances

Domain join directory [Info](#)

Select

Create new directory

IAM instance profile [Info](#)

10amec2s3

arn:aws:iam::174912287653:instance-profile/10amec2s3

Create new IAM profile

Hostname type [Info](#)

IP name

DNS Hostname [Info](#)

☒ Enable IP name IPv4 (A record) DNS requests

☒ Enable resource-based IPv4 (A record) DNS requests

☐ Enable resource-based IPv6 (AAAA record) DNS requests

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...read more

ami-0G3e149Saf50eGfd5

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

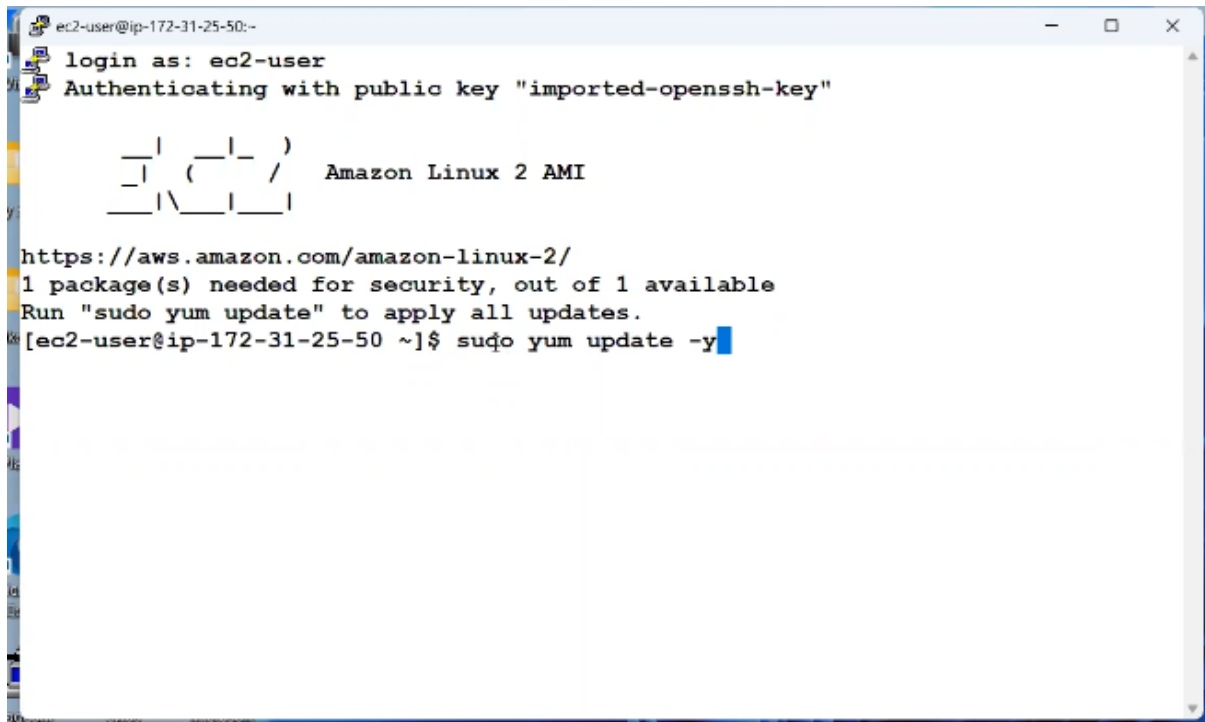
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the

Cancel

Launch instance

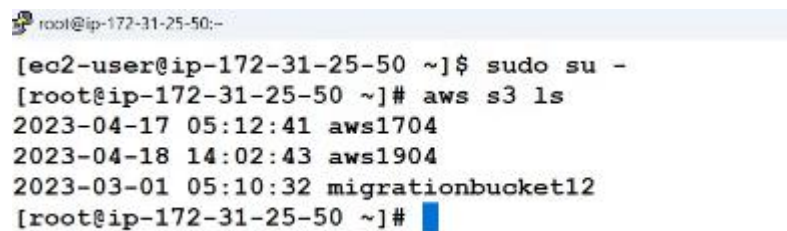
Review commands

- (\*) create instance with role and without role
- (\*) login using PUTTY configurator paste ipv4 address -- ssh- credentials
- (\*) with role white
- (\*) without role black



```
ec2-user@ip-172-31-25-50:~$  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
  
  _|  _|  _|  
 _| (  _| /  Amazon Linux 2 AMI  
 _| \ _| _|  
  
https://aws.amazon.com/amazon-linux-2/  
1 package(s) needed for security, out of 1 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-25-50 ~]$ sudo yum update -y
```

- (\*) Aws s3 ls



```
root@ip-172-31-25-50:~$  
[ec2-user@ip-172-31-25-50 ~]$ sudo su -  
[root@ip-172-31-25-50 ~]# aws s3 ls  
2023-04-17 05:12:41 aws1704  
2023-04-18 14:02:43 aws1904  
2023-03-01 05:10:32 migrationbucket12  
[root@ip-172-31-25-50 ~]#
```

(\*) without role

```
ec2-user@ip-172-31-27-102:~$ ssh
login as: ec2-user
Authenticating with public key "imported-openssh-key"

      _ _ | _ _ | _ _ )
      _ | ( _ _ | _ _ /   Amazon Linux 2 AMI
      _ | \ _ _ | _ _ |

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 1 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-27-102 ~]$ sudo yum update -y
```

(\*) aws s3 ls

```
root@ip-172-31-27-102:~$ sudo su -
[ec2-user@ip-172-31-27-102 ~]$ sudo su -
[root@ip-172-31-27-102 ~]# aws s3 ls
Unable to locate credentials. You can configure credentials by running "aws configure".
[root@ip-172-31-27-102 ~]#
```

(\*) for user login in CLI go to IAM user – select user – security credentials – access keys --

Identity and Access Management (IAM)

Q Search IAM

Dashboard

Access management

User groups

**Users**

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analysers

Settings

Credential report

Organization activity

Service control policies (SCPs)

Related console

IAM Identity Center

AWS Organizations

IAM > Users > varshini

varshini

Delete

Summary

ARN

arn:aws:iam::509638718401:user/varshini

Console access

Enabled without MFA

Access key 1

Not enabled

Created

April 28, 2023, 22:12 (UTC+05:30)

Last console sign-in

Today

Access key 2

Not enabled

Permissions

Groups (1)

Tags

**Security credentials**

Access Advisor

Console sign-in

Manage console access

Console sign-in link

https://kathredochu.signin.aws.amazon.com/console

Console password

Updated 1 hour ago (2023-04-28 22:18 GMT+5:30)

Last console sign-in

1 hour ago (2023-04-28 22:17 GMT+5:30)

Multi-factor authentication (MFA)

Remove

Resync

Assign MFA device

Device type

Identifier

Created on

No MFA devices. Assign an MFA device to improve the security of your AWS environment.

Assign MFA device

Access keys

Create access key

No access keys

As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials.&#x2192;Learn more

Create access key

## Access key best practices & alternatives

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Command Line Interface (CLI)

You plan to use this access key to enable the AWS CLI to access your AWS account.

Local code

You plan to use this access key to enable application code in a local development environment to access your AWS account.

Application running on an AWS compute service

You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

Third-party service

You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

Application running outside AWS

You plan to use this access key to enable an application running on an on-premises host, or to use a local AWS client or third-party AWS plugin.

Other

Your use case is not listed here.

Alternatives recommended

Use AWS CloudShell, a browser-based CLI, to run commands. Learn more

Use the AWS CLI V2 and enable authentication through a user in IAM Identity Center. Learn more

☒ I understand the above recommendation and want to proceed to create an access key.

Cancel

Next

IAM > Users > varshini > Create access key

Step 1

Access key best practices & alternatives

Step 2 - optional

Set description tag

Step 3

**Retrieve access keys**

Retrieve access keys

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key

AKIAJNKF5AGDPI6AF0X

Secret access key

\*\*\*\*\* Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

Download .csv file

Done

root@ip-172-31-27-102:~

```
[ed2-user@ip-172-31-27-102 ~]$ sudo su -  
[root@ip-172-31-27-102 ~]# aws s3 ls  
Unable to locate credentials. You can configure credentials by running "aws configure".  
[root@ip-172-31-27-102 ~]# aws configure  
AWS Access Key ID [None]: AKIASROMVWOS7BLIPDON  
AWS Secret Access Key [None]: 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  
Default region name [None]: ap-southeast-1  
Default output format [None]: json  
[root@ip-172-31-27-102 ~]# aws s3 ls  
2023-04-17 05:12:43 aws1704  
2023-04-18 14:02:45 aws1904  
2023-04-08 08:28:55 migrationbucket12  
[root@ip-172-31-27-102 ~]#
```