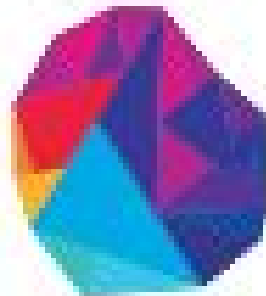


# **ASSESSMENT**

## **ON :**

## **IAM**

**TO  
THE  
NEW**



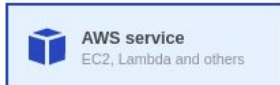
## 1. Create a Role with full access to S3.

STEP 1: S3 > go to Roles > Create Role

### Create role

1 2 3 4

#### Select type of trusted entity



Allows AWS services to perform actions on your behalf. [Learn more](#)



**Another AWS account**  
Belonging to you or 3rd party



**Web identity**  
Cognito or any OpenID provider



**SAML 2.0 federation**  
Your corporate directory

#### Choose a use case

##### 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.

### Create role

1 2 3 4

#### ▼ Attach permissions policies

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

Create policy

Filter policies ▼

Q S3

Showing 4 results

	Policy name ▼	Used as
<input type="checkbox"/>	▶  AmazonDMSRedshiftS3Role	None
<input checked="" type="checkbox"/>	▶  AmazonS3FullAccess	Permissions policy (2)
<input type="checkbox"/>	▶  AmazonS3ReadOnlyAccess	None
<input type="checkbox"/>	▶  QuickSightAccessForS3StorageManagementAnalyticsReadOnly	None

## Create role

1

2

3

4

### Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
Owner	Diksha	✕
Purpose	Providing <u>S3</u> full access	✕
Add new key		

## Create role

1

2

3

4

### Review

Provide the required information below and review this role before you create it.

**Role name\***

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

**Role description**

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

**Trusted entities**

AWS service: ec2.amazonaws.com

**Policies** [AmazonS3FullAccess](#) **Permissions boundary**

Permissions boundary is not set

The new role will receive the following tags

Key	Value
Owner	Diksha

STEP 2: Role has been created

Create role		Delete role
<input type="text"/> Search		
Role name ▾	Trusted entities	Last activity ▾
<input type="checkbox"/> AWSServiceRoleForElasticLoadBalancing	AWS service: elasticloadbalancing (Service-...	302 days
<input type="checkbox"/> AWSServiceRoleForSupport	AWS service: support (Service-Linked role)	248 days
<input type="checkbox"/> AWSServiceRoleForTrustedAdvisor	AWS service: trustedadvisor (Service-Linked ...	None
<input checked="" type="checkbox"/> dikshaS3fullAccess	AWS service: ec2	None
<input type="checkbox"/> s3admin	AWS service: ec2	None
<input type="checkbox"/> s3ss	AWS service: ec2	None

\*\*\*\*\*

**2. Create another role which has the policy to assume the previous Role.**

**ANS:**

STEP 1: Create a new role “Diksha\_assume\_role”

## Create role

1 2 3 4

### Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
Owner	Diksha	✕
Purpose	For assume ole implementation	✕
Add new key		

You can add 48 more tags.

## Create role

1

2

3

4

### Review

Provide the required information below and review this role before you create it.

**Role name\***

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

**Role description**

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

**Trusted entities** AWS service: ec2.amazonaws.com

**Policies** Policies not attached

**Permissions boundary** Permissions boundary is not set

The new role will receive the following tags

Key	Value
Owner	Diksha

### STEP 2: Create a new policy. Select service STS and action assume role

Go to resources(specific) and Copy the ARN of s3 full access(i.e your previous role"dikshaS3fullAccess") and paste it and click on add.

A policy defines the AWS permissions that you can grant to an IAM role. Learn more

Visual editor JSON

Expand all Collapse all

▼ STS (1 action) ⚠ 1 warning

► Service

► Actions

▼ Resources close

► Request conditions Specify request conditions (optional)

Add ARN(s)

Amazon Resource Names (ARNs) uniquely identify AWS resources. Resources are unique to each service. Learn more

Specify ARN for role List ARNs manually

Account \*  ☐ Any

Role name with path \*  ☐ Any

Cancel Add

Import managed policy

Clone Remove

☐ Any

### STEP 3: Review your policy

▶ Service

STS

▶ Actions

Manual actions

\*

AssumeRole

▼ Resources

☒ Specific

[close](#)

☐ All resources

role ?

arn:aws:iam::187632318301:role/dikshaS3fullAccess

EDIT

✕

☐ Any

[Add ARN to restrict access](#)

user ?

You have not specified resource with type **user**

☐ Any

[Add ARN to restrict access](#)

#### Review policy

Name\*

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

Description

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

Summary

Service ▼	Access level	Resource	Request condition
Allow (1 of 223 services) <a href="#">Show remaining 222</a>			
STS	Limited: Write	RoleName   string like   dikshaS3fullAccess_EC2	None

### STEP 4: Attach the policy “AssumeRolePolicy\_diksha” to the new role “Diksha\_assume\_role\_new”

Create policy

Policy actions ▼

Filter policies ▼

Attach

Detach

Delete

	Policy	Type	Used as	Description
<input checked="" type="radio"/>	▶ AssumeRolePolicy_diksha	Customer managed	None	Assume role policy
<input type="radio"/>	▶ DataAdmin_diksha	Customer managed	Permissions policy (1)	

## Attach policy

Attach the policy to users, groups, or roles in your account

Filter: Filter

<input type="checkbox"/>	Name
<input type="checkbox"/>	diksha.tomar@tothenew.com
<input type="checkbox"/>	dikshaTomar
<input type="checkbox"/>	dikshaS3fullAccess_EC2
<input checked="" type="checkbox"/>	Diksha_assume_role_new

**STEP 5:** Click the new role and check that the new role contains the assume role.

Roles > Diksha\_assume\_role\_new

### Summary

Role ARN	arn:aws:iam::187632318301:role/Diksha_assume_role_new
Role description	Allows EC2 instances to call AWS services on your behalf.   <a href="#">Edit</a>
Instance Profile ARNs	arn:aws:iam::187632318301:instance-profile/Diksha_assume_role_new
Path	/
Creation time	2020-02-28 17:47 UTC+0530
Last activity	Not accessed in the tracking period
Maximum CLI/API session duration	1 hour <a href="#">Edit</a>

Permissions Trust relationships Tags (2) Access Advisor Revoke sessions

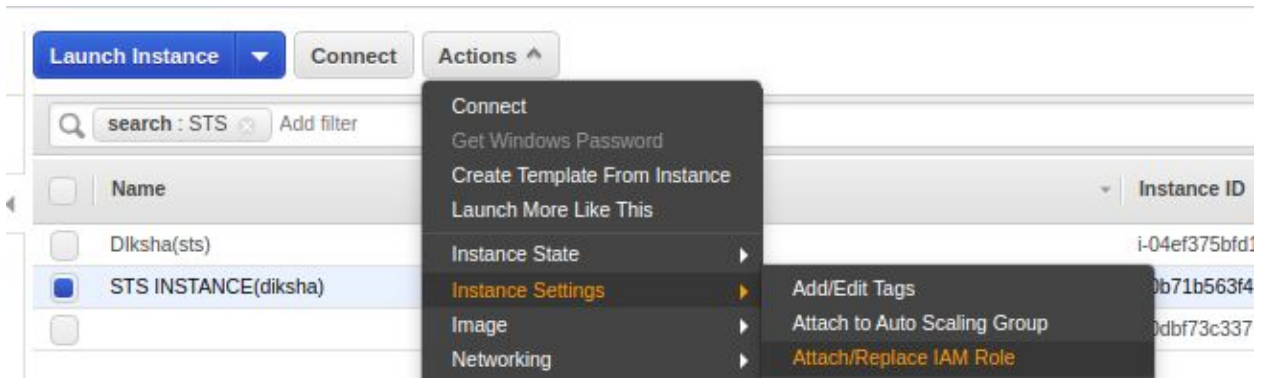
▼ Permissions policies (1 policy applied)

[Attach policies](#)

Policy name	Policy type
AssumeRolePolicy_diksha	Managed policy

► Permissions boundary (not set)

**STEP 6:** Now create an ec2 instance and attach to the “Diksha\_assume\_role\_new” created



[Instances](#) > Attach/Replace IAM Role

## Attach/Replace IAM Role

Select an IAM role to attach to your instance. If you don't have any IAM roles, choose Create new IAM role to create a role in the IAM console. If an IAM role is already attached to your instance, the IAM role you choose will replace the existing role.

Instance ID i-0b71b563f4717d1e6 (STS INSTANCE(diksha)) ⓘ

IAM role\*  ⓘ [Create new IAM role](#) ⓘ

\* Required

**STEP 7:** Now add the arn of new role i.e “Diksha\_assume\_role\_new” to old role in trust relationship

## Edit Trust Relationship

You can customize trust relationships by editing the following access control policy document.

**Policy Document**

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "AWS": "arn:aws:iam::187632318301:role/Diksha_assume_role_new",
8         "Service": "ec2.amazonaws.com"
9       },
10      "Action": "sts:AssumeRole"
11    }
12  ]
13 }
```

**STEP 8:** Now you have to ssh to the instance created and update it. Also install aws cli



```
diksha@diksha:~/Downloads$ ssh -i "diksha_awskey.pem" ubuntu@ec2-3-82-163-44.compute-1.amazonaws.com
The authenticity of host 'ec2-3-82-163-44.compute-1.amazonaws.com (3.82.163.44)' can't be established.
ECDSA key fingerprint is SHA256:ENVcN0fWGjCrdViekEK5a2TcUfvt+Vx7N68ReWvLRX8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-3-82-163-44.compute-1.amazonaws.com,3.82.163.44' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1057-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Feb 28 17:28:13 UTC 2020

System load:  0.0               Processes:    87
Usage of /:   13.6% of 7.69GB   Users logged in: 0
Memory usage: 14%              IP address for eth0: 172.31.4.230
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

```
ubuntu@ip-172-31-4-230:~$ sudo apt install awscli
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  docutils-common libjpeg-turbo8 libjpeg8 liblcms2-2 libpaper-utils
```

\*\*\*\*\*

**3. Attach this to an instance and get an sts token.**

```
ubuntu@ip-172-31-4-230:~$ aws sts assume-role --role-arn arn:aws:iam::187632318301:role/dikshaS3fullAccess_EC2 --role-session-name dikshasession
{
  "Credentials": {
    "AccessKeyId": "ASIASXL6B6505VEUOL7R",
    "SecretAccessKey": "P79V7XrMQ7vIJVibwnFj5KQJFI44bMdM/eYfJ4Tu",
    "SessionToken": "FwoGZXIvYXdzECMaDD3e70/XXo7kw556YiKxAX2hocI9tm4U/gGvpC9Y+aJYakuzzEx8YpbbID9HuGHoaGgCpCat6d8FWqsCPbGQbfMxPncznfn3ZQFeQgpgxWq3euNsvHI60Rnh9tj6cncdj67AMjv7eVrmRHyKIqIF/FJutRen1e+qC3lj7yut7UkfDZFxGL8vkPLF19mfJYcwqYhAgS0+k9ILvbBYUZ1PYeabGmpQqafR4MQ4moYlnZKZFMcG6Lgg8xM0zVK/Ea/3/yiunuXyBTItBfD955TpEKLJ8d08t2HnAwoKvA+fDjU5p0CA+xWJA/BA27eQxhHBIQXSB2ZF",
    "Expiration": "2020-02-28T18:34:38Z"
  },
  "AssumedRoleUser": {
    "AssumedRoleId": "AR0ASXL6B650SMPECWCJR:dikshasession",
    "Arn": "arn:aws:sts::187632318301:assumed-role/dikshaS3fullAccess_EC2/dikshasession"
  }
}
```

Now export it:

```
ubuntu@ip-172-31-4-230:~$ export AWS_ACCESS_KEY_ID=ASIASXL6B6505VEUOL7R
ubuntu@ip-172-31-4-230:~$ export AWS_SECRET_ACCESS_KEY=^C
ubuntu@ip-172-31-4-230:~$ export AWS_SECRET_ACCESS_KEY=P79V7XrMQ7vIJVibwnFj5KQJFI44bMdM/eYfJ4Tu
ubuntu@ip-172-31-4-230:~$ export AWS_SESSION_TOKEN=FwoGZXIvYXdzECMaDD3e70/XXo7kw556YiKxAX2hocI9tm4U/gGvpC9Y+aJYakuzzEx8YpbbID9HuGHoaGgCpCat6d8FWqsCPbGQbfMxPncznfn3ZQFeQgpgxWq3euNsvHI60Rnh9tj6cncdj67AMjv7eVrmRHyKIqIF/FJutRen1e+qC3lj7yut7UkfDZFxGL8vkPLF19mfJYcwqYhAgS0+k9ILvbBYUZ1PYeabGmpQqafR4MQ4moYlnZKZFMcG6Lgg8xM0zVK/Ea/3/yiunuXyBTItBfD955TpEKLJ8d08t2HnAwoKvA+fDjU5p0CA+xWJA/BA27eQxhHBIQXSB2ZF
ubuntu@ip-172-31-4-230:~$
```

Now you can access s3

```
ubuntu@ip-172-31-4-230:~$ aws s3 ls
2019-06-26 12:11:08 0testuser11
2018-04-20 16:59:22 187632318301-awsmacietrail-dataevent
2019-04-02 10:11:33 7testdemo
2019-03-11 04:51:59 abhimanyucftemplate
2020-02-28 10:55:02 abhishek-bootcamp
2019-03-04 06:55:23 abneesh1
2019-03-11 11:00:41 adityamun007
2020-02-26 16:26:29 akshaybuck1
2020-02-27 08:55:25 aman-khandelwal-1
2019-03-07 09:40:48 anmol-bootcamp19
2019-03-08 00:25:58 avcabc
2017-09-07 03:41:42 aws-codestar-us-east-1-187632318301
2017-09-07 04:23:01 aws-codestar-us-east-1-187632318301-codestartest2-app
2017-09-07 04:23:07 aws-codestar-us-east-1-187632318301-codestartest2-pipe
2017-09-07 03:41:48 aws-codestar-us-east-1-187632318301-codestarttest-pipe
```

\*\*\*\*\*

4. Create a group for "Data Administrator" where the user 'Alice' be a member of this group. This group will prepare the data for the analysis. So Provide the following access to the group.

Service: Amazon S3;

Action:

Get\*,

List\*,

Put\*,

ARN: Input and output Buckets (no conditions)

STEP 1: Create a user

Add user



Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

- Access type\* ☒ **Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- ☐ **AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

STEP 2: Create group "DataAdministrator\_diksha" and add the above user to this group.

## Add user to group

Create group

Refresh

Search

Showing 13 results

Group	Attached policies
<input type="checkbox"/> data_Administrator	S3ListPutGet
<input type="checkbox"/> DataAdmin	s3readwritelispooja
<input type="checkbox"/> DataAdmin-Chhavi	None
<input type="checkbox"/> DataAdministrator	None
<input checked="" type="checkbox"/> DataAdministrator_diksha	DataAdmin_diksha

## Review

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

### User details

User name	dikshaTomar
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

### Permissions summary

The user shown above will be added to the following groups.

Type	Name
Group	DataAdministrator_diksha

### Tags

The new user will receive the following tags

Key	Value
Owner	Diksha
Purpose	giving S3 access only



**Success**

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://ttn-newers.signin.aws.amazon.com/console>

[Download .csv](#)

	User	Access key ID	Secret access key
▶	✔ dikshaTomar	AKIASXL6B65O4ZTRYISR	BAQ5g43Fk1pQRdDDXCGT ynEPicfsz+r13PSyRool <a href="#">Hide</a>

STEP 3: Create policy “DataAdmin\_diksha”as per the question

Create policy

12

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 editor

JSON

Import managed policy

Expand all

Collapse all

▼ S3 (69 actions) ⚠ 2 warnings

Clone Remove

▶ Service

S3

▶ Actions

Manual actions

List\*

Get\*

Put\*

▼ Resources

☒ Specific

☐ All resources

close

▶ Request conditions

Specify request conditions (optional)

▼ Resources

☒ Specific

☐ All resources

close

accesspoint ?

Specify **accesspoint** resource ARN for the **PutAccessPointPolicy** and 2 more actions. ⓘ

Add ARN to restrict access

☐ Any

bucket ?

Any resource of type = bucket

☒ Any

object ?

Specify **object** resource ARN for the **PutObjectRetention** and 18 more actions. ⓘ

Add ARN to restrict access

☐ Any

➕ Add additional permissions

## Review policy

**Name\***

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

**Description**

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

**Summary**

Service ▾	Access level	Resource	Request condition
Allow (1 of 223 services) <a href="#">Show remaining 222</a>			
S3	Limited: List, Read, Write, Permissions management, Tagging	Multiple	None

## STEP 4: Attach the above policy to the group

### Attach Policy

Select one or more policies to attach. Each group can have up to 10 policies attached.

Filter: Policy Type ▾		<input type="text" value="Dataadmin"/>	Showing 3 results	
		Policy Name ↕	Attached Entities ↕	Creation Time ↕
<input type="checkbox"/>		AWSLakeFormationDataAdmin	0	2019-08-08 23:03 UTC+05...
<input type="checkbox"/>		DataAdmin-Policy-Chhavi	0	2020-02-27 16:31 UTC+05...
<input checked="" type="checkbox"/>		DataAdmin_diksha	0	2020-02-27 16:49 UTC+05...

IAM > Groups > DataAdministrator\_diksha

### Summary


<b>Group ARN:</b>	arn:aws:iam::187632318301:group/DataAdministrator_diksha 
<b>Users (in this group):</b>	1
<b>Path:</b>	/
<b>Creation Time:</b>	2020-02-27 16:52 UTC+0530

### Users

### Permissions


### Access Advisor

This view shows all users in this group: **1 User**

User	Actions
 <a href="#">dikshaTomar</a>	<a href="#">Remove User from Group</a>

IAM > Groups > DataAdministrator\_diksha

▼ Summary

**Group ARN:** arn:aws:iam::187632318301:group/DataAdministrator\_diksha   
**Users (in this group):** 1  
**Path:** /  
**Creation Time:** 2020-02-27 16:52 UTC+0530

Users

Permissions

Access Advisor

Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

[Attach Policy](#)

Policy Name	Actions
DataAdmin_diksha	<a href="#">Show Policy</a>   <a href="#">Detach Policy</a>   <a href="#">Simulate Policy</a>

Inline Policies

\*\*\*\*\*

**5. Create a group for the "Developer group " where the user 'bob ' is a member of this group. This group with Test Newly Developed Features for which they require access to EC2 instances. Provide the following access to this group:**

**Service: Amazon EC2**

**Action: \*Instances, \*Volume, Describe\*, CreateTags;**

**Condition: Dev Subnets only**

STEP 1: Create a group "DeveloperGroup\_diksha"

## Set Group Name

Specify a group name. Group names can be edited any time.

Group Name:

DeveloperGroup\_diksha

Example: Developers or ProjectAlpha  
Maximum 128 characters

### STEP 2: Create a user garima

#### Add user



#### Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

Graina

[+](#) Add another user

#### Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type\* ☐

**Programmatic access**

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.



**AWS Management Console access**

Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password\* ☐

Autogenerated password



Custom password

Diksha@20



Show password

### STEP 3: Add the user to group "DeveloperGroup\_diksha"



## Add Users to Group

Select users to add to the group **DeveloperGroup\_diksha**

<input type="text" value="Gra"/>			
<input type="checkbox"/>	User Name ↕	Groups	Password
<input checked="" type="checkbox"/>	Graina	0	✓

**STEP 4:** Create a policy “dev\_group\_policy” and specify the action and condition as mentioned in the question(Providing arn of Dev subnet)

## Create policy

1 2

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 editor

JSON

[Import managed policy](#)

[Expand all](#) | [Collapse all](#)

▼ EC2 (113 actions) ⚠ 6 warnings

Clone Remove

▶ Service EC2

▼ Actions  
close

Specify the actions allowed in EC2 ⓘ

Switch to deny permissions ⓘ

Manual actions (add actions)

☒ ec2:\*Volumes (Edit | Remove)

☒ ec2:\*Instances (Edit | Remove)

☒ ec2:CreateTags (Edit | Remove)

☐ All EC2 actions (ec2:\*)

☒ ec2:Describe\* (Edit | Remove)

Add ARN to restrict access

Add ARN(s) ✕

Amazon Resource Names (ARNs) uniquely identify AWS resources. Resources are unique to each service. [Learn more](#)

**Specify ARN for subnet** [List ARNs manually](#)

arn:aws:ec2:us-east-1:187632318301:subnet/subnet-00b26cdd8f633e3a9

Region \*

us-east-1

☐ Any

Account \*

187632318301

☐ Any

Subnet id \*

subnet-00b26cdd8f633e3a9

☐ Any

Cancel

Add

spot-instance-req... You have not specified resource with type spot-instance-request

## Create policy

1

2

### Review policy

Name\* dev\_group\_policy

Use alphanumeric and '+', '@', '-' characters. Maximum 128 characters.

Description

providing access to instances in Dev subnet

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

Summary

Q Filter

Service ▼	Access level	Resource	Request condition
Allow (1 of 223 services) <a href="#">Show remaining 222</a>			
EC2	Limited: List, Read, Write, Tagging	Multiple	None

**STEP 5:** Attach the above policy to the “DeveloperGroup\_diksha”

[Create policy](#)
[Policy actions](#)

[Filter policies](#)

[Attach](#)  
[Detach](#)  
[Delete](#)

Policy	Type	Used as	Description
dev_group_policy	Customer managed	None	providing access to instances in Dev subnet

## Attach policy

Attach the policy to users, groups, or roles in your account

Filter: [Filter](#)

Showing 6 results

Name	Type
<input type="checkbox"/> lambda-developer-identity-provider-role	Role
<input type="checkbox"/> developer-group-ayush	Group
<input type="checkbox"/> Developer-Maithely	Group
<input checked="" type="checkbox"/> DeveloperGroup_diksha	Group
<input type="checkbox"/> Developers	Group
<input type="checkbox"/> Developer_Group	Group

STEP 6: Check the group that it contains the user and the policy that you created

IAM > [Groups](#) > [DeveloperGroup\\_diksha](#)

### Summary

Group ARN:	arn:aws:iam::187632318301:group/DeveloperGroup_diksha
Users (in this group):	1
Path:	/
Creation Time:	2020-02-28 12:36 UTC+0530


### Users

This view shows all users in this group: **1 User**

User	Actions
 <a href="#">Graitha</a>	<a href="#">Remove User from Group</a>

IAM > Groups > DeveloperGroup\_diksha

▼ Summary

**Group ARN:** arn:aws:iam::187632318301:group/DeveloperGroup\_diksha   
**Users (in this group):** 1  
**Path:** /  
**Creation Time:** 2020-02-28 12:36 UTC+0530

Users

Permissions

Access Advisor

Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

Attach Policy

Policy Name	Actions
dev_group_policy	<a href="#">Show Policy</a>   <a href="#">Detach Policy</a>   <a href="#">Simulate Policy</a>

Inline Policies

\*\*\*\*\*

**6. Identify the unused IAM users/credentials using AWS CLI.**

**ANS:**

STEP 1: List all users and Install jq

```
diksha@diksha:~/Downloads$ aws iam list-users
{
  "Users": [
    {
      "Path": "/",
      "UserName": "abhishek.chauhan1@tothenew.com",
      "UserId": "AIDASXL6B650Q4RMZ427Z",
      "Arn": "arn:aws:iam::187632318301:user/abhishek.chauhan1@tothenew.com",
      "CreateDate": "2020-02-19T11:03:23Z",
      "PasswordLastUsed": "2020-02-28T05:03:08Z"
    },
    {
      "Path": "/",
      "UserName": "aditya.upadhyay@tothenew.com",
      "UserId": "AIDASXL6B650YD7UUCZUJ",
      "Arn": "arn:aws:iam::187632318301:user/aditya.upadhyay@tothenew.com",
      "CreateDate": "2020-02-19T11:03:25Z",
      "PasswordLastUsed": "2020-02-28T04:46:17Z"
    },
    {
      "Path": "/",
      "UserName": "akshay.shrivastava@tothenew.com",
      "UserId": "AIDASXL6B650SGPOGZHF0",
      "Arn": "arn:aws:iam::187632318301:user/akshay.shrivastava@tothenew.com",
      "CreateDate": "2020-02-19T11:03:26Z",
      "PasswordLastUsed": "2020-02-28T04:20:30Z"
    },
    {
      "Path": "/",
      "UserName": "Alice",
      "UserId": "AIDASXL6B6506DXIQS5RS",
      "Arn": "arn:aws:iam::187632318301:user/Alice",
      "CreateDate": "2020-02-27T12:11:40Z"
    }
  ]
}
```

\*jq is like `sed` for JSON data - you can use it to slice and filter and map and transform structured data with the same ease that `sed`, `awk`, `grep` and friends let you play with text.

```
diksha@diksha:~/Downloads$ aws iam list-users | jq '.Users[] | select(.PasswordLastUsed==null) | .UserName'
"Alice"
"Alice-Chhavi"
"alice-maithely"
"asusumeuser"
"Bob"
"Bob-maithely"
"bobpooja"
"CloudChecker"
"dikshaTomar"
"Gargi_Alice"
"garima.dabral@tothenew.com"
"HAWK2.0-user"
"poojaalice"
"raghu.sharma@tothenew.com"
"s3pooja"
"vivek.yadav1@tothenew.com"
diksha@diksha:~/Downloads$ aws iam list-users
```

\*\*\*\*\*

**7. Identify all the instances having the tag key-value "backup=true" using AWS CLI.**

**ANS:** `aws ec2 describe-instances --filters`

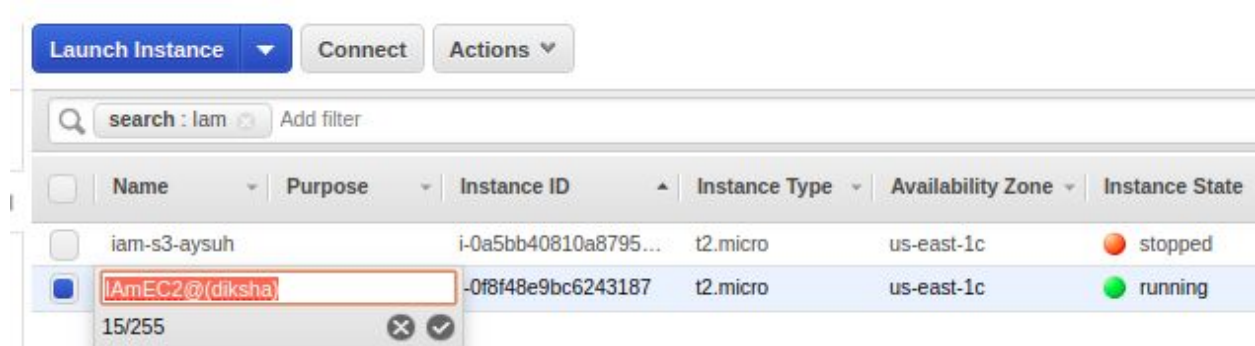
```
diksha@diksha:~/Downloads$ aws ec2 describe-instances --filters "Name=tag:backup,Values=true"
{
  "Reservations": []
}
diksha@diksha:~/Downloads$
```

\*\*\*\*\*

**8. An EC2 Instance hosts a Java-based application that accesses an s3 bucket. This EC2 Instance is currently serving production users. Create the role and assign the role to EC2 instance. Launch an EC2 instance:**

**ANS:**

STEP 1: Launch an EC2 instance





STEP 2: Create a role and attach S3fullAccess policy to it.


## Create role


1 2 3 4

### Select type of trusted entity

**AWS service**  
EC2, Lambda and others

**Another AWS account**  
Belonging to you or 3rd party

**Web identity**  
Cognito or any OpenID provider

**SAML 2.0 federation**  
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

### Choose a use case

#### 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.


Or select a service to view its use cases

## Create role

1 2 3 4

### ▼ Attach permissions policies

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

Create policy 

Filter policies ▼

Q S3fullaccess

Showing 1 result

	Policy name ▼	Used as
<input checked="" type="checkbox"/>	 AmazonS3FullAccess	Permissions policy (38)

## Create role

1

### Review

Provide the required information below and review this role before you create it.

Role name\*

diksha\_S3fullAccessToEc2

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

Role description

Allows EC2 instances to call AWS services on your behalf.

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

Trusted entities AWS service: ec2.amazonaws.com

Policies



AmazonS3FullAccess 

Permissions boundary Permissions boundary is not set



STEP 3: Attach the above role to your instance

[Instances](#) > Attach/Replace IAM Role

## Attach/Replace IAM Role

Select an IAM role to attach to your instance. If you don't have any IAM roles, choose Create new IAM role to create a role in the console. If an IAM role is already attached to your instance, the IAM role you choose will replace the existing role.

Instance ID i-0f8f48e9bc6243187 (IAM-EC2@diksha) ⓘ

IAM role\*



[Create new IAM role](#)

\* Required

STEP 4: SSH into your instance and run command: `$aws s3 ls`

```
ubuntu@ip-172-31-248-66:~$ aws s3 ls
2019-06-26 12:11:08 0testuser11
2018-04-20 16:59:22 187632318301-awsmacietrail-dataevent
2019-04-02 10:11:33 7testdemo
2019-03-11 04:51:59 abhimanyucftemplate
2019-03-04 06:55:23 abneesh1
2019-03-11 11:00:41 adityamun007
2020-02-26 16:26:29 akshaybuck1
2020-02-27 08:55:25 aman-khandelwal-1
2019-03-07 09:40:48 anmol-bootcamp19
2019-03-08 00:25:58 avcab
2017-09-07 03:41:42 aws-codestar-us-east-1-187632318301
2017-09-07 04:23:01 aws-codestar-us-east-1-187632318301-codestartest2-app
2017-09-07 04:23:07 aws-codestar-us-east-1-187632318301-codestartest2-pipe
2017-09-07 03:41:48 aws-codestar-us-east-1-187632318301-codestarttest-pipe
2019-06-26 05:39:55 aws-lambda-trigger-ronozor
2020-02-28 03:56:49 ayush-public-bucket
2020-02-25 07:02:11 baban-123
2018-02-14 12:28:43 cf-templates-71mx96ojlvv5-us-east-1
2019-03-27 15:57:27 cfront1
2020-02-26 11:51:54 chirag-bucket-2
2020-02-26 11:46:43 chirag-bucket1
2019-03-27 20:34:52 cloudfront8
2020-02-25 10:59:18 copy-test-delete
2020-02-26 08:17:11 diksha.static.website
2019-06-26 10:49:10 ec2-access-bucket
2019-03-28 05:23:51 ec2-ttn
2019-03-01 07:28:00 ekanshbucket
```



\*\*\*\*\*





**9. You have both production and development based instances running on your VPC. It is required to ensure that people responsible for the development instances do not have access to work on production instances for better security. Define the tags on the test and production servers and add a condition to the IAMPolicy which allows access to specific tags.**

**ANS:**

STEP 1: Create two instances in the default VPC:

1)diksha-production

2)diksha\_development

	diksha-production	Launching instance	i-081ed2a4bf8d0a5bb	t2.micro	us-east-1c	 running
	diksha_development		i-0a676810df007f623	t2.micro	us-east-1c	 running

STEP 2: Now create 2 users :

1)Dev-diksha

2)Prod-diksha

Add user

1

2

3

4

5



#### Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://ttn-newers.signin.aws.amazon.com/console>



Download .csv

	User	Email login instructions
	 Dev-diksha	<a href="#">Send email</a> 
	 Prod-diksha	<a href="#">Send email</a> 

STEP 3: Now create a policy for development server

## Create policy

1

2

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 editor

JSON

[Import managed policy](#)

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [ {
4     "Sid": "StartStopIfTags",
5     "Effect": "Allow", "Action": [
6       "ec2:StartInstances",
7       "ec2:StopInstances",
8       "ec2:DescribeTags"
9     ],
10    "Resource": "arn:aws:ec2:region:account-id:instance/*",
11    "Condition": {
12      "StringEquals": {
13        "ec2:ResourceTag/Project": "diksha_development",
14        "aws:PrincipalTag/Department": "Dev-diksha"
15      }
16    }
17  } ]
18 }
19 }
```

**STEP 4:** And similarly for production server

## Create policy

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 editor

JSON

```
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "StartStopIfTags",
6       "Effect": "Allow",
7       "Action": [
8         "ec2:StartInstances",
9         "ec2:StopInstances",
10        "ec2:DescribeTags"
11      ],
12      "Resource": "arn:aws:ec2:region:account-id:instance/*",
13      "Condition": {
14        "StringEquals": {
15          "ec2:ResourceTag/Project": "diksha-production",
16          "aws:PrincipalTag/Department": "Prod-diksha"
17        }
18      }
19    }
20  ]
21 }
```

**10. Create a policy for allowing users to set or rotate their credentials, such as their console password, their programmatic access keys, and their MFA devices.**

**STEP 1:** Create a policy and set service=IAM and give actions as per the question



Review policy

Name\*

Diksha\_change\_credentials

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

Description

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

Summary

This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose **Show remaining**. [Learn more](#)

Q Filter

Service ▾	Access level	Resource	Request condition
Allow (1 of 223 services) <a href="#">Show remaining 222</a>			
IAM	Limited: Write	UserName   string like   \${aws=username}	None

STEP 2: Policy has been created.

✔

Diksha\_change\_credentials has been created.

Create policy

Policy actions ▾