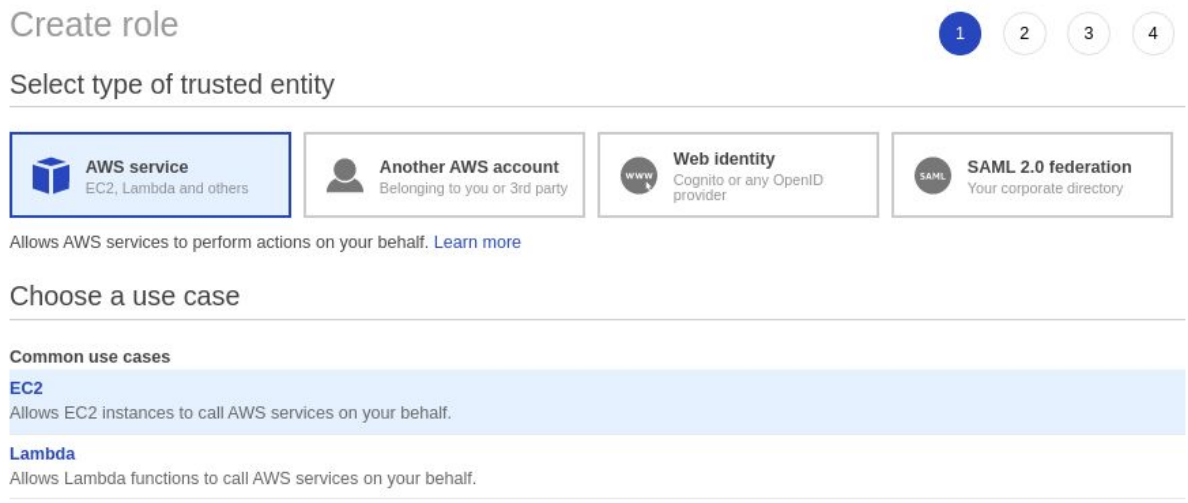
ASSESSMENT ON :

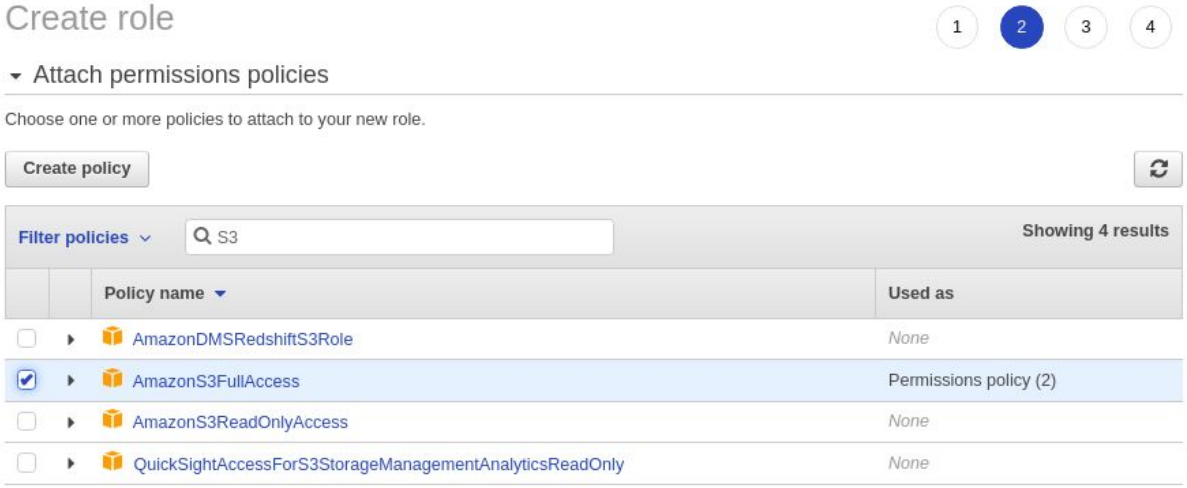
IAM

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

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 S3 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* dikshaS3fullAccess_EC2

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

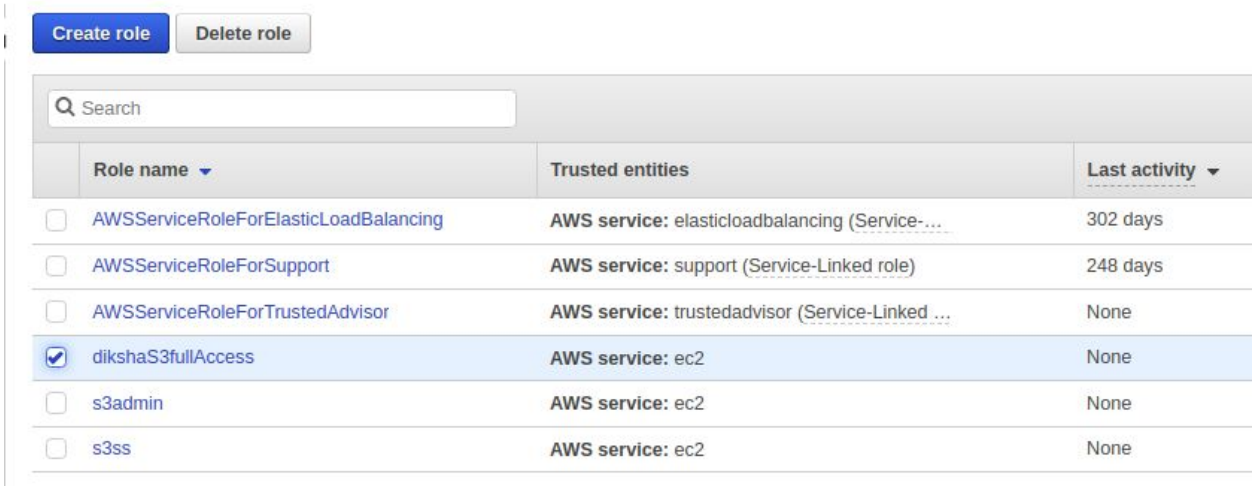
Policies AmazonS3FullAccess

Permissions boundary Permissions boundary is not set

The new role will receive the following tags

Key	Value
Owner	Diksha

STEP 2: Role has been created

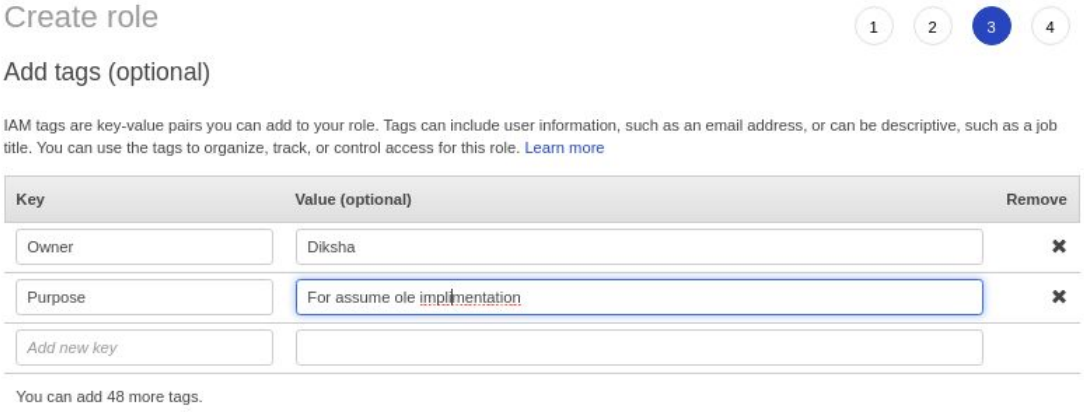


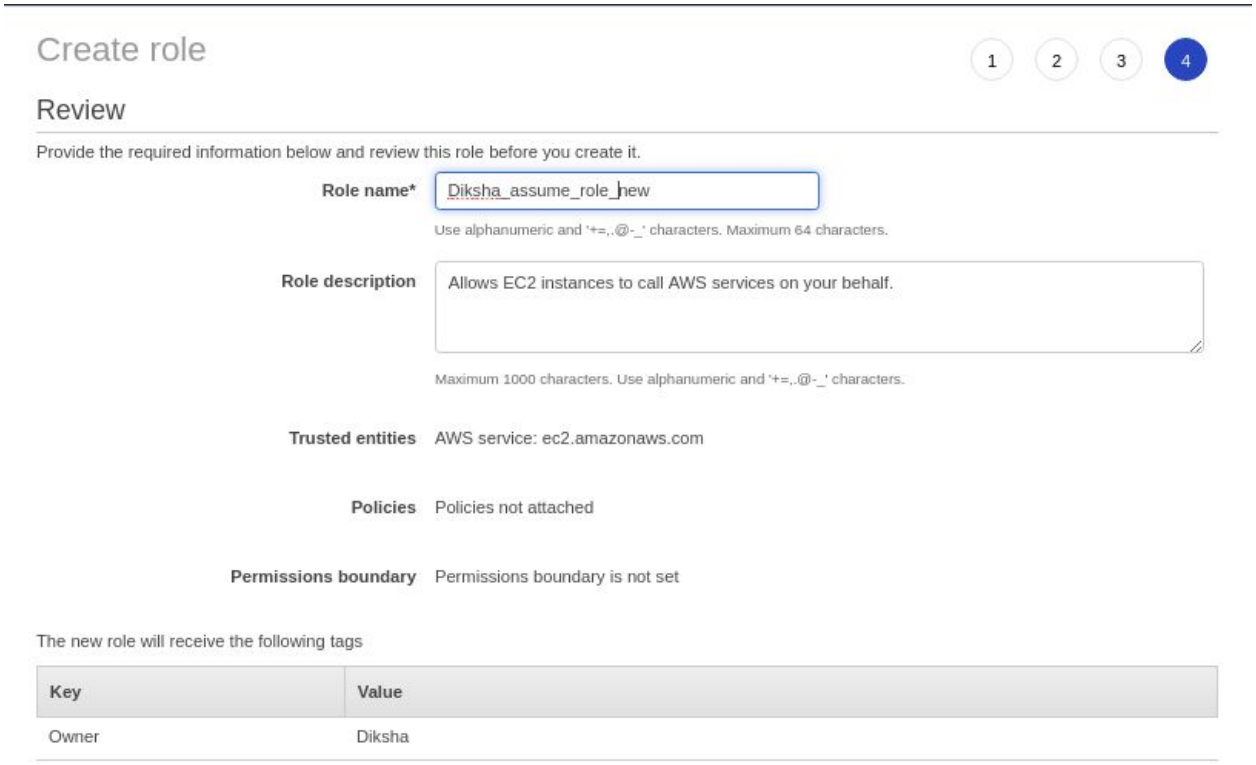
****

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

ANS:

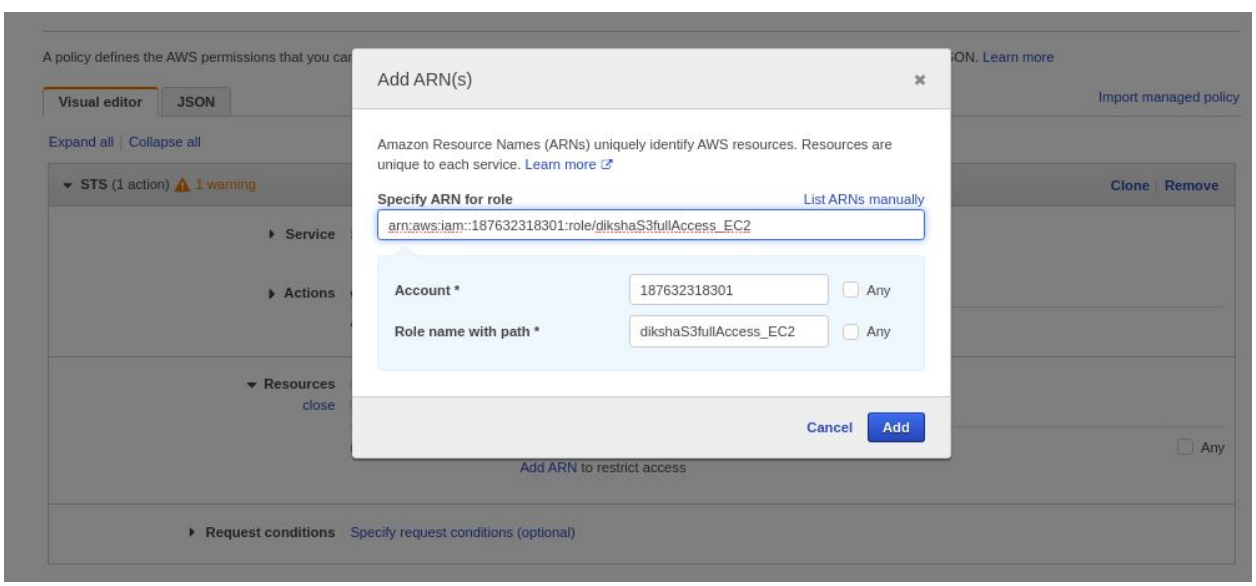
<u>STEP 1:</u> Create a new role “Diksha_assume_role”



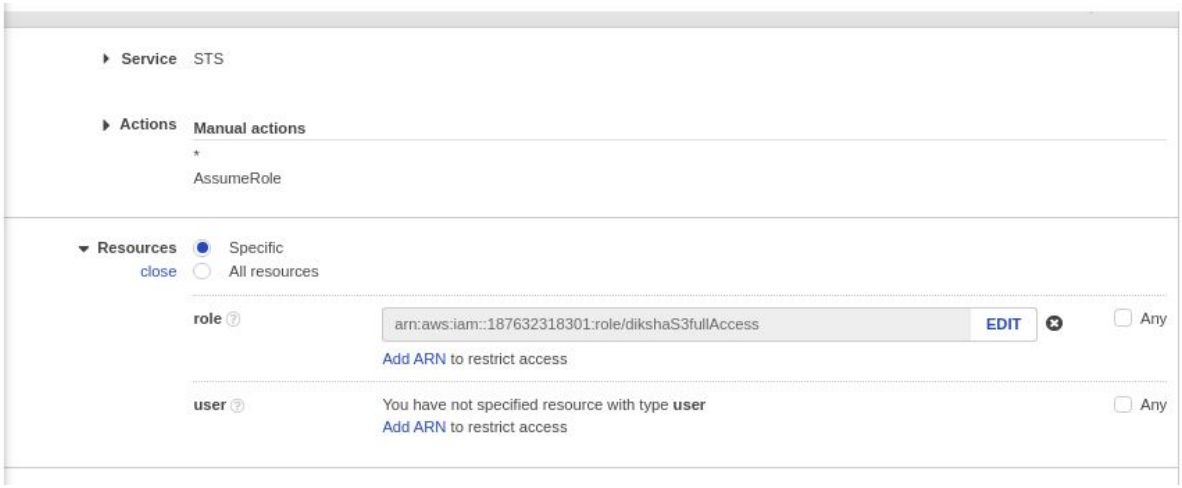


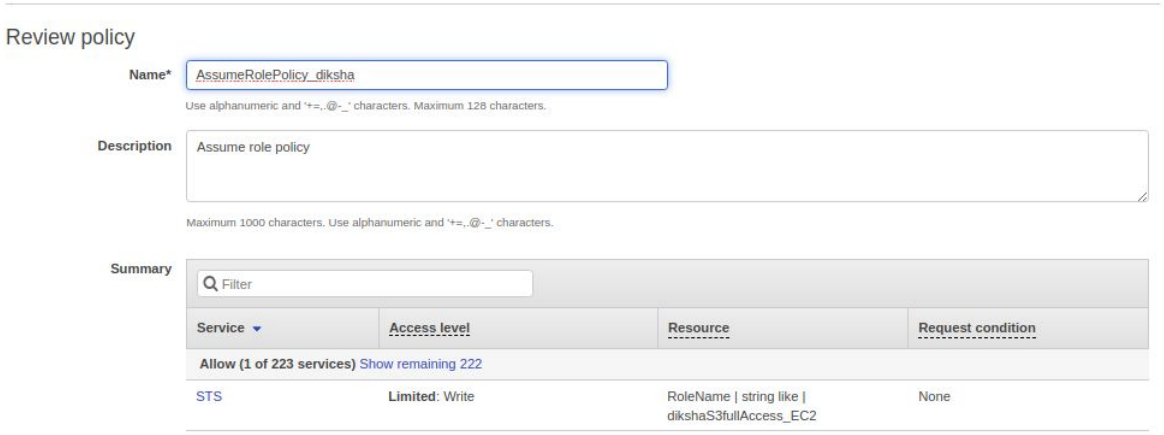
STEP 2: Create a new policy.Select service STS and action assume role

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.

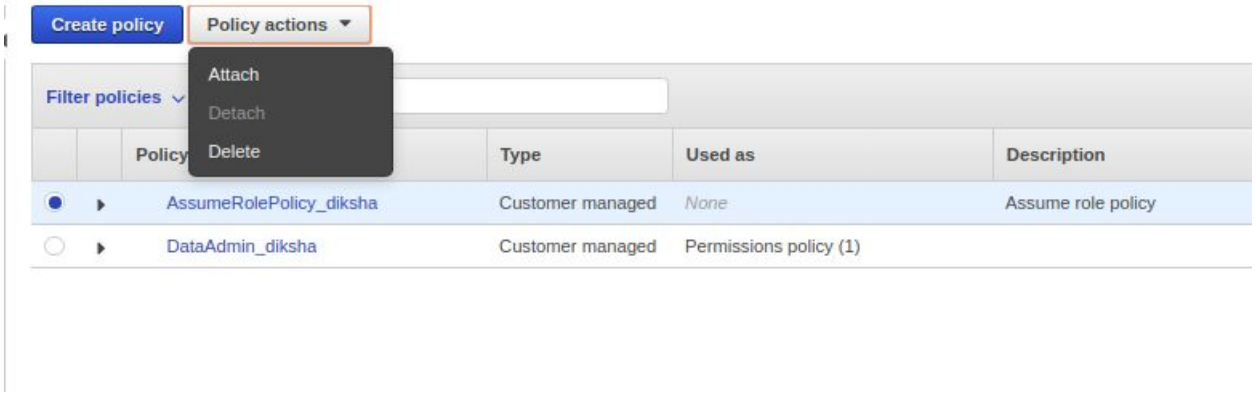


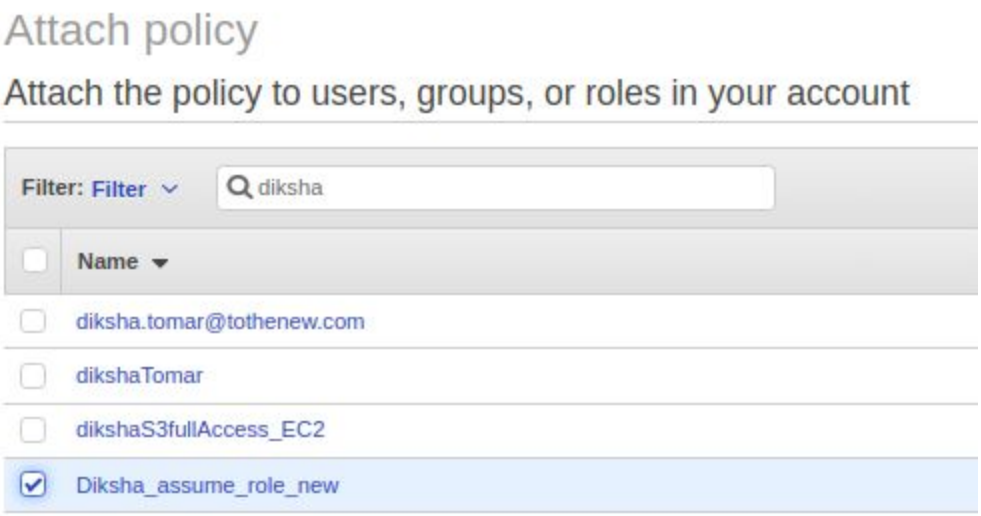
STEP 3: Review your policy



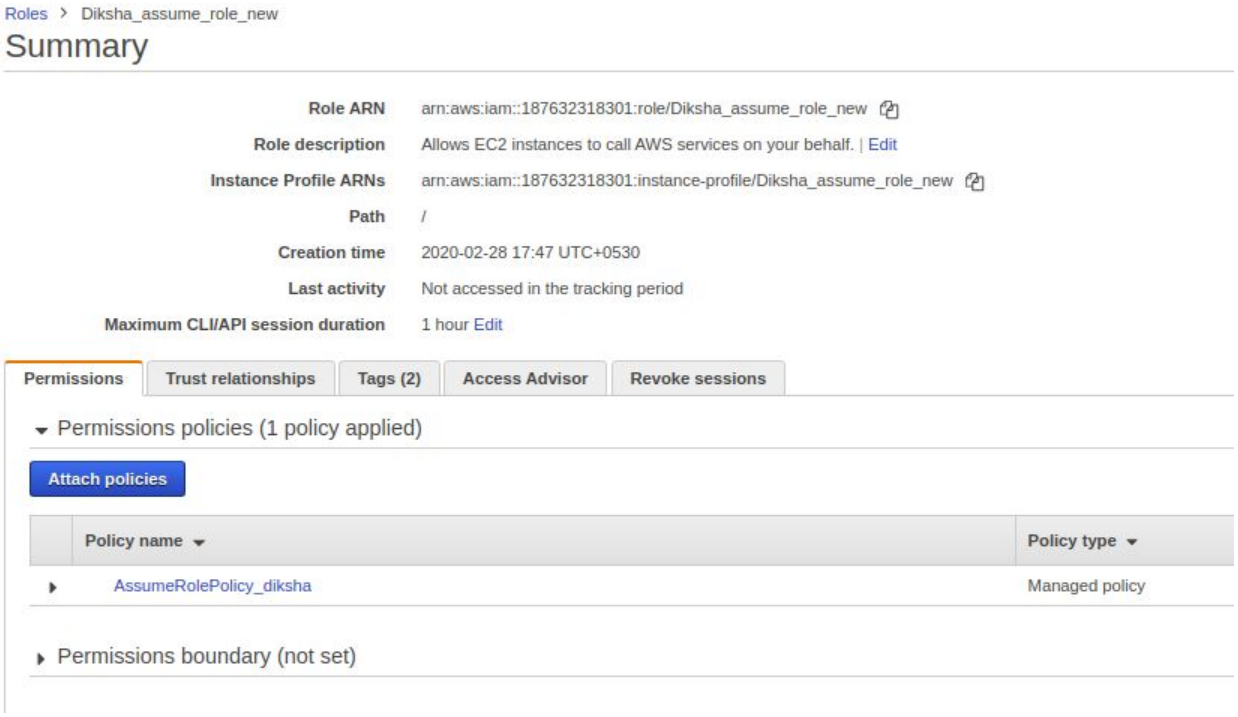


STEP 4: Attach the policy “AssumeRolePolicy_diksha” to the new role “Diksha_assume_role_new”

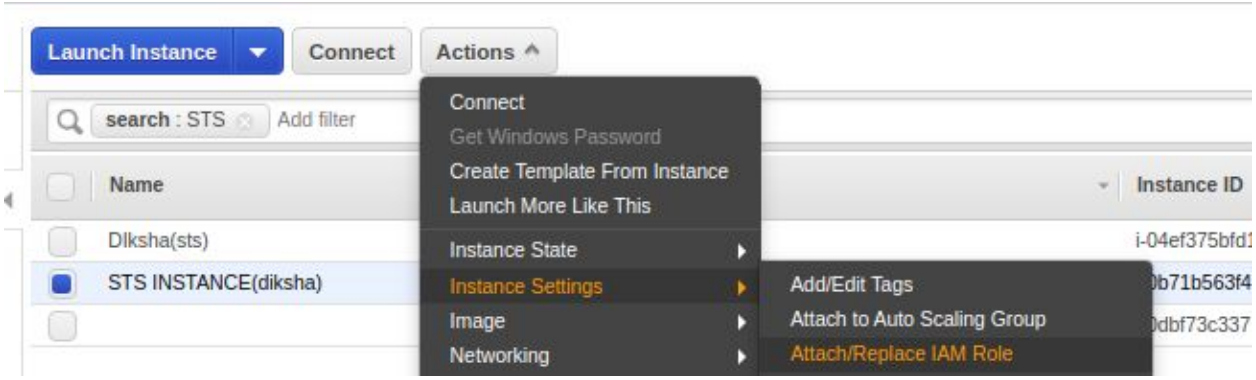


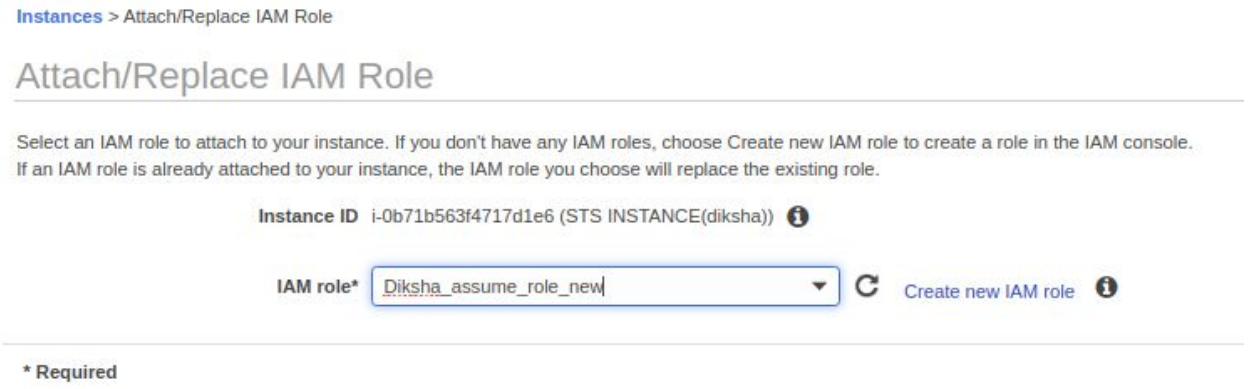


STEP 5: Click the new role and check that the new role contains the assume role.

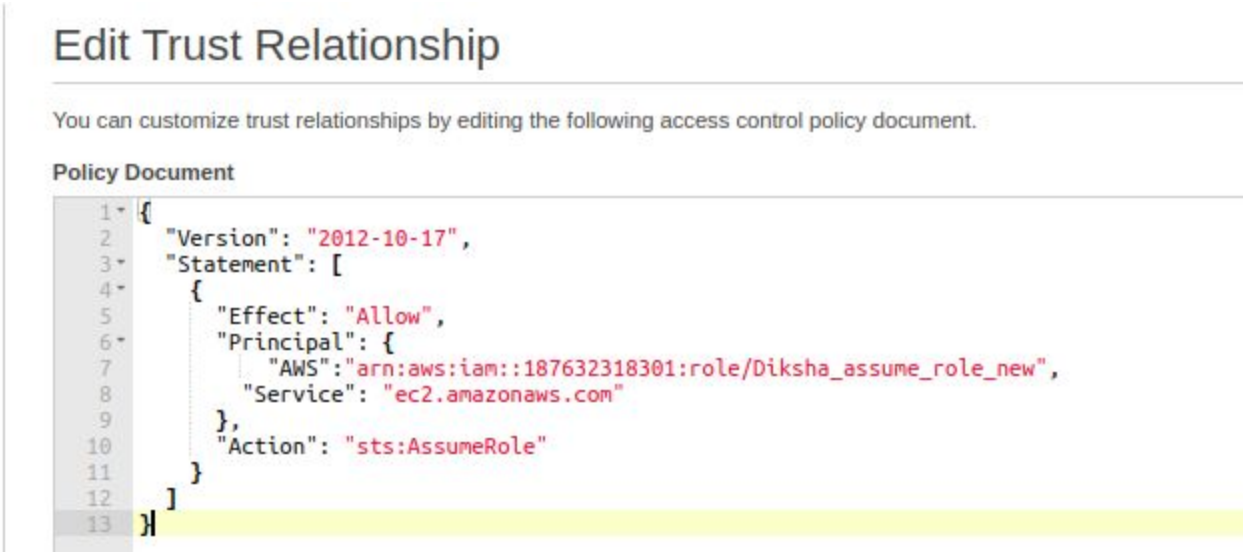


STEP 6: Now create an ec2 instance and attach to the “Diksha_assume_role_new” created





STEP 7: Now add the arn of new role i.e “Diksha_assume_role_new” to old role in trust relationship



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 libjbig0 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::1876323183
01:role/dikshaS3fullAccess_EC2 --role-session-name dikshasession
{
    "Credentials": {
        "AccessKeyId": "ASIASXL6B65O5VEUOL7R",
        "SecretAccessKey": "P79V7XrMQ7vIJVibwnFj5KQJFI44bMdM/eYfJ4Tu",
        "SessionToken": "FwoGZXIvYXdzECMaDD3e70/XXo7kw556YiKxAX2hocI9tm4U/gGvpC9
Y+aJYakuzzEx8YpbpID9HuGHoaGgCpCat6d8FWqsCPbGQbfMxPncznfn3ZQFeQgpgxWq3euNsvHI60Rn
h9tj6cncdj67AMjv7eVrmRHyKIqIF/FJutRen1e+qC3lj7yut7UkfDZFxGl8vkPLF19mfJYcwqYhAgS0
+k9ILvbBYUZ1PYeabGmpQqafR4MQ4moYlnZKZFMcG6Lgg8xM0zVK/Ea/3/yiunuXyBTItBfD955TpEKL
J8d08t2HnAwoKvA+fDjU5p0CA+xWJA/BA27eQxhHBIQXSB2ZF",
        "Expiration": "2020-02-28T18:34:38Z"
    },
    "AssumedRoleUser": {
        "AssumedRoleId": "AROASXL6B65OSMPECWCJR:dikshasession",
        "Arn": "arn:aws:sts::187632318301:assumed-role/dikshaS3fullAccess_EC2/di
kshasession"
    }
}
```

Now export it:

```
ubuntu@ip-172-31-4-230:~$ export AWS_ACCESS_KEY_ID=ASIASXL6B65O5VEUOL7R
ubuntu@ip-172-31-4-230:~$ export AWS_SECRET_ACCESS_KEY=^C
ubuntu@ip-172-31-4-230:~$ export AWS_SECRET_ACCESS_KEY=P79V7XrMQ7vIJVibwnFj5KQJF
I44bMdM/eYfJ4Tu
ubuntu@ip-172-31-4-230:~$ export AWS_SESSION_TOKEN=FwoGZXIvYXdzECMaDD3e70/XXo7kw
556YiKxAX2hocI9tm4U/gGvpC9Y+aJYakuzzEx8YpbpID9HuGHoaGgCpCat6d8FWqsCPbGQbfMxPnczn
fn3ZQFeQgpgxWq3euNsvHI60Rnh9tj6cncdj67AMjv7eVrmRHyKIqIF/FJutRen1e+qC3lj7yut7UkfD
ZFxGl8vkPLF19mfJYcwqYhAgS0+k9ILvbBYUZ1PYeabGmpQqafR4MQ4moYlnZKZFMcG6Lgg8xM0zVK/E
a/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:

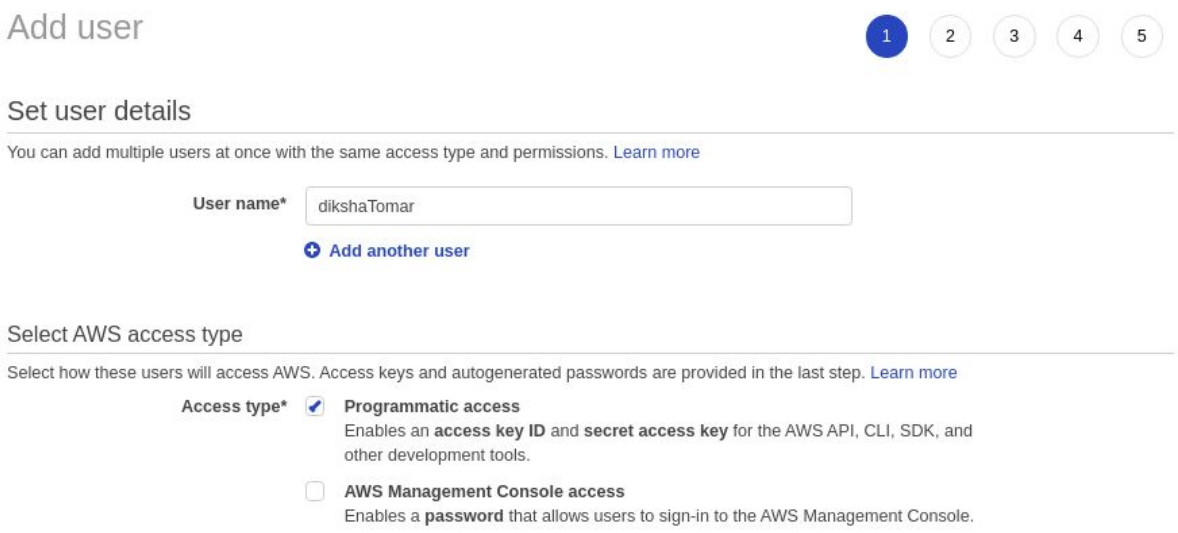
Get*,

List*,

Put*,

ARN: Input and output Buckets (no conditions)

STEP 1: Create a user



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
	data_Administrator	S3ListPutGet
	DataAdmin	s3readwritelistpooja
	DataAdmin-Chhavi	None
	DataAdministrator	None
✔	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

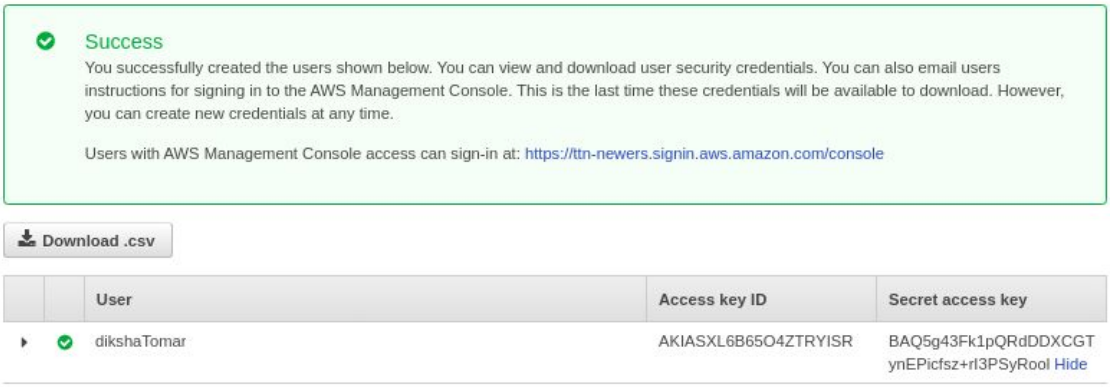
The user shown above will be added to the following groups.

Type	Name
Group	DataAdministrator_diksha

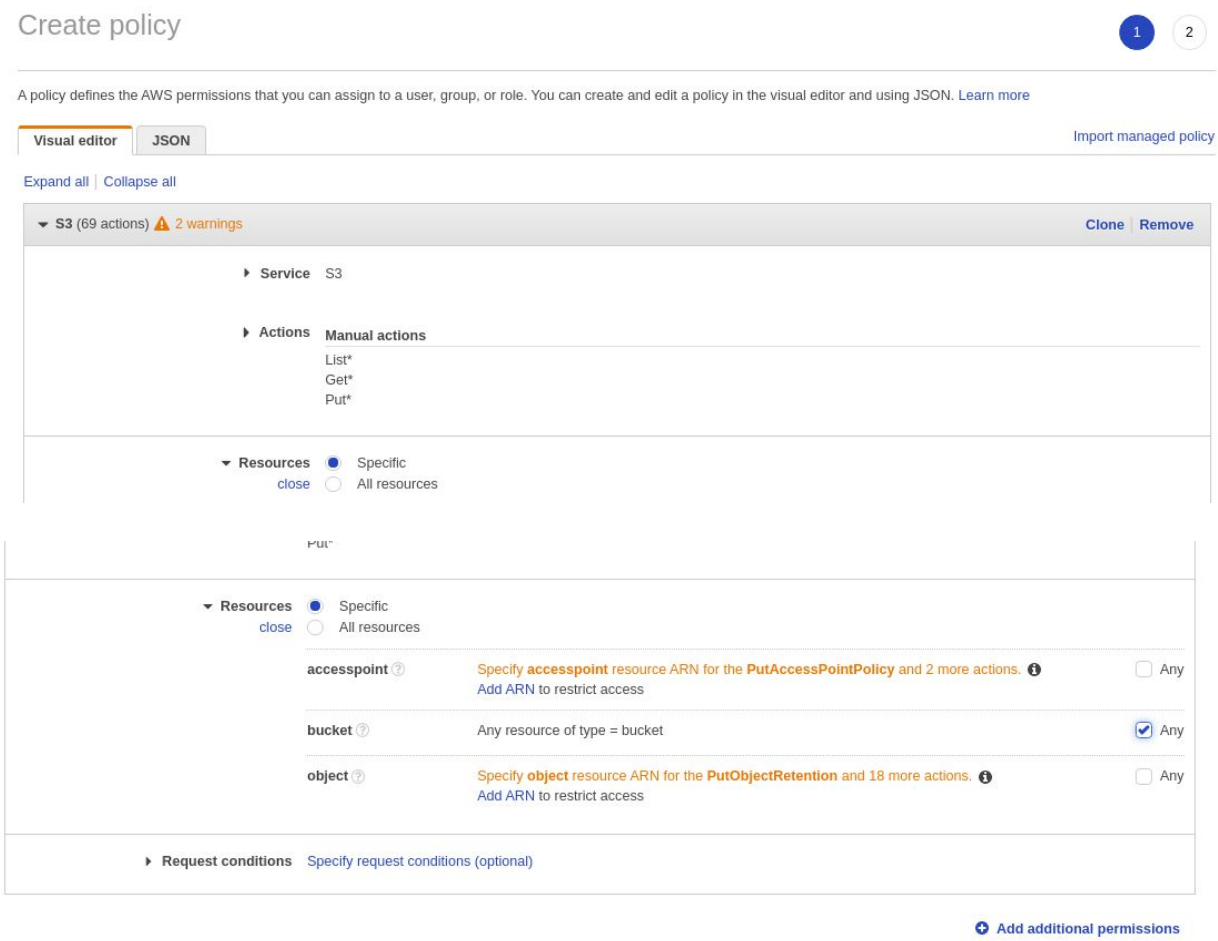
Tags

The new user will receive the following tags

Key	Value
Owner	Diksha
Purpose	giving S3 access only



STEP 3: Create policy “DataAdmin_diksha”as per the question



Review policy

Name* DataAdmin_diksha

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

Description

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

Summary

Filter

Service	Access level	Resource	Request condition
Allow (1 of 223 services) Show remaining 222			
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 Dataadmin **Showing 3 results**

	Policy Name	Attached Entities	Creation Time
☐	AWSLakeFormationDataAdmin	0	2019-08-08 23:03 UTC+05...
☐	DataAdmin-Policy-Chhavi	0	2020-02-27 16:31 UTC+05...
☑	DataAdmin_diksha	0	2020-02-27 16:49 UTC+05...

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**

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

User	Actions
dikshaTomar	Remove User from Group

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	Show Policy | Detach Policy | Simulate Policy

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

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* Graima

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.

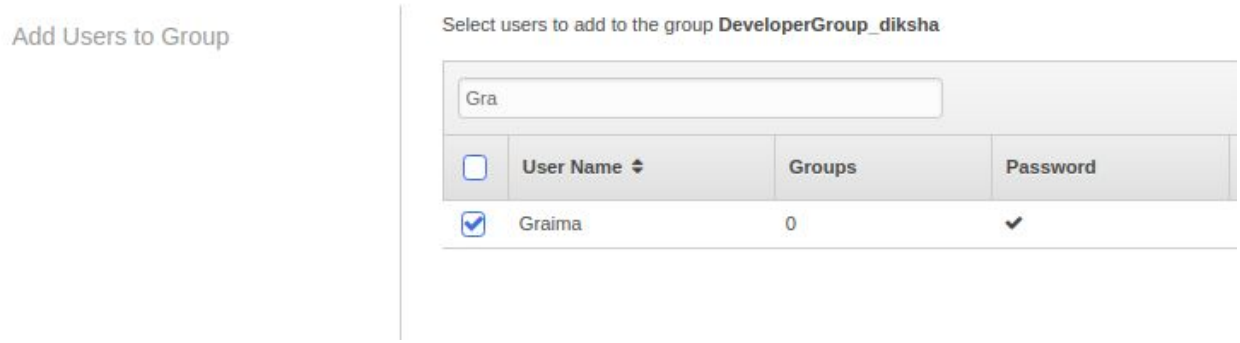
☑ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password* ○ Autogenerated password
● Custom password

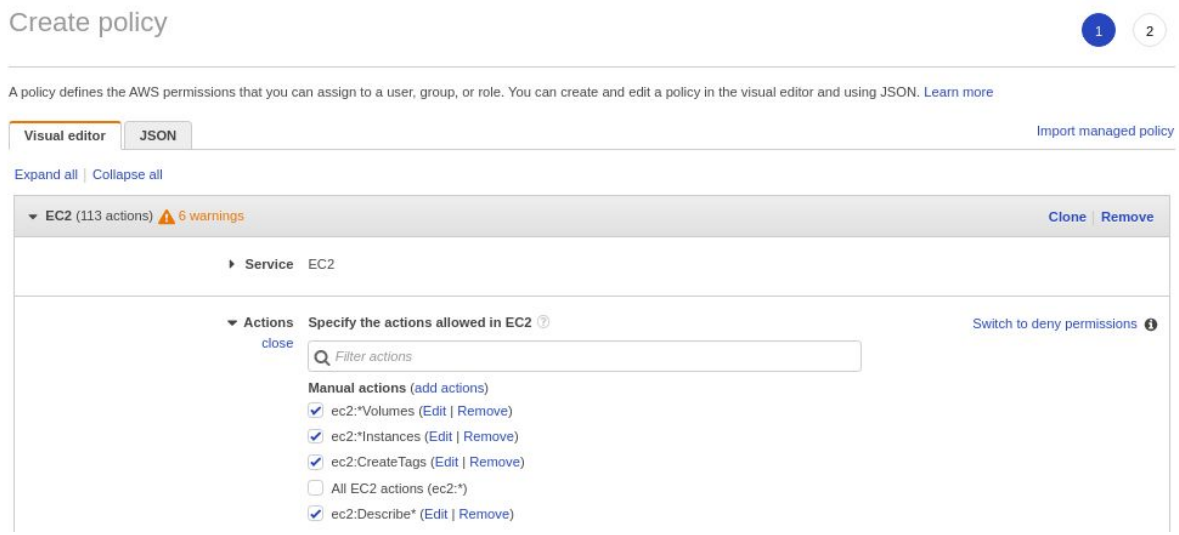
Diksha@20

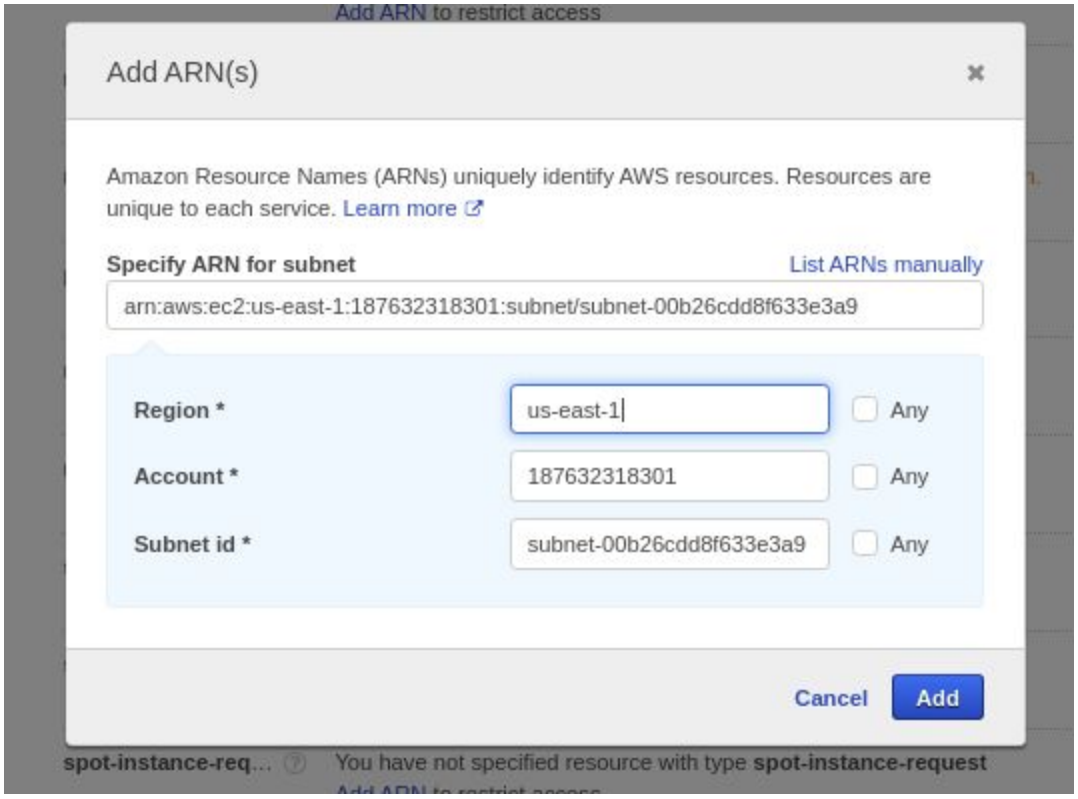
☑ Show password

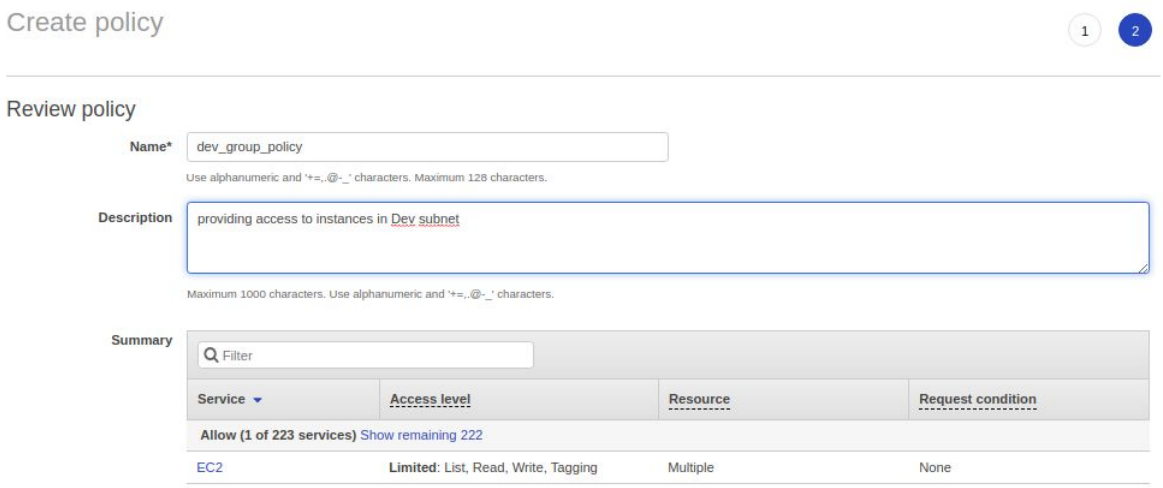
STEP 3: Add the user to group “DeveloperGroup_diksha”



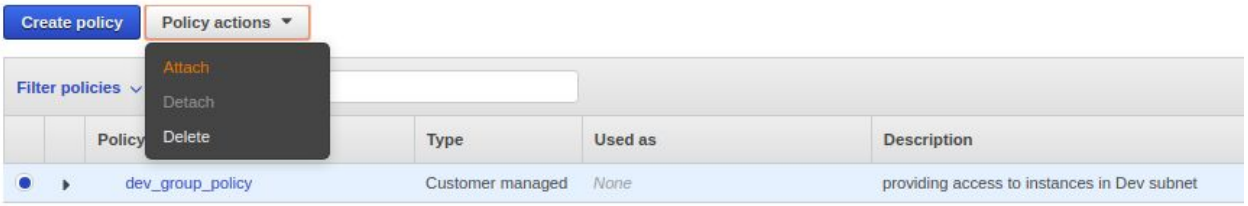
STEP 4: Create a policy “dev_group_policy” and specify the action and condition as mentioned in the question(Providing arn of Dev subnet)

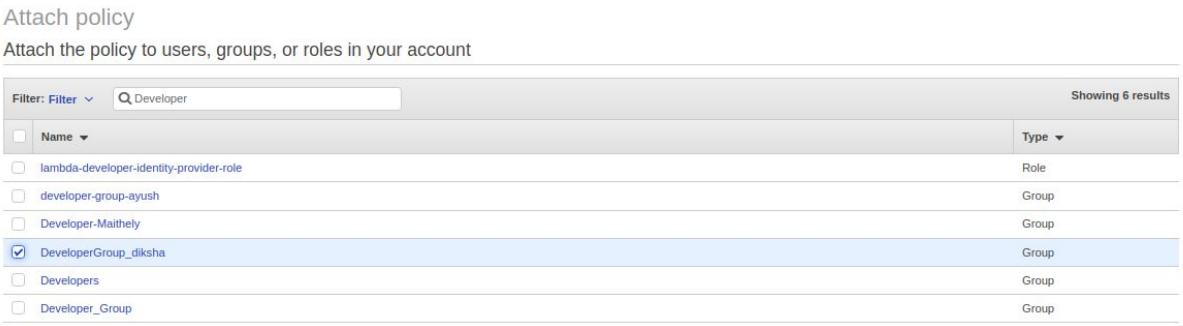




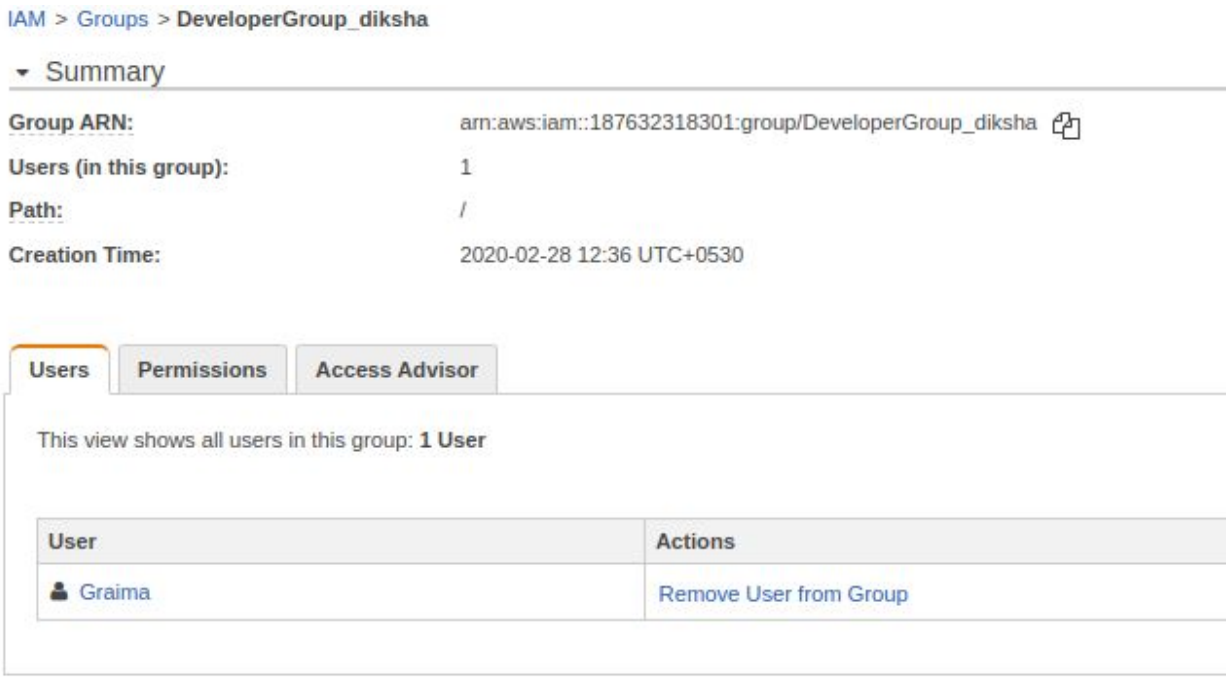


STEP 5: Attach the above policy to the “DeveloperGroup_diksha”





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 | 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	Show Policy | Detach Policy | Simulate Policy

Inline Policies

****

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

ANS:

<u>STEP 1:</u> List all users and Install jq

```
diksha@diksha:~/Downloads$ aws iam list-users
{
    "Users": [
        {
            "Path": "/",
            "UserName": "abhishek.chauhan1@tothenew.com",
            "UserId": "AIDASXL6B65OQ4RMZ427Z",
            "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": "AIDASXL6B65OYD7UUCZUJ",
            "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": "AIDASXL6B65OSGPOGZHFO",
            "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": "AIDASXL6B65O6DXIQS5RS",
            "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"
"CloudCheckr"
"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`

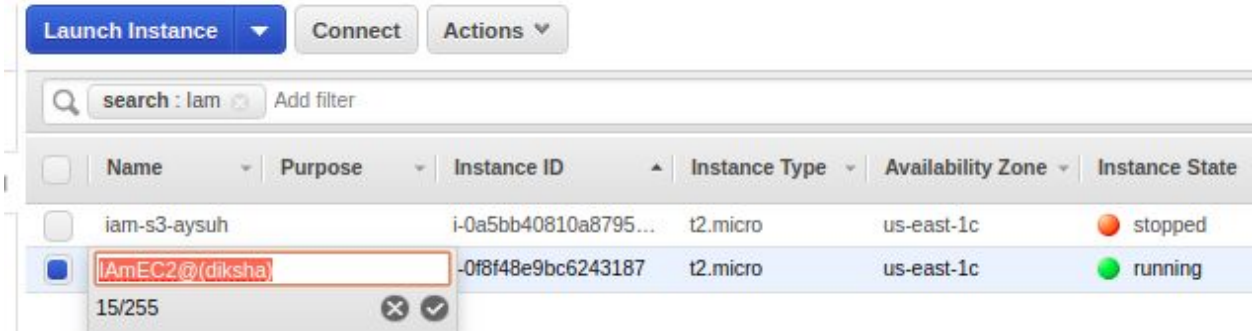
```
diksha@diksha:~/Downloads$ aws ec2 describe-instances --filters "Name=tag:backup
,Values=true"
{
    "Reservations": []
}
diksha@diksha:~/Downloads$
```

****

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 | S3fullacces | Showing 1 result

	Policy name	Used as
☑	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.

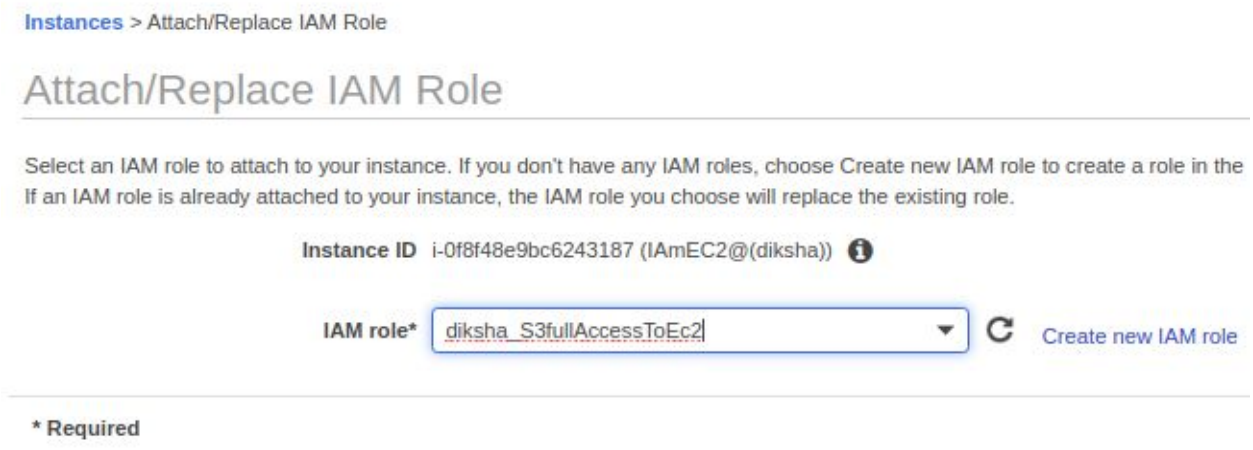
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

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



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 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
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

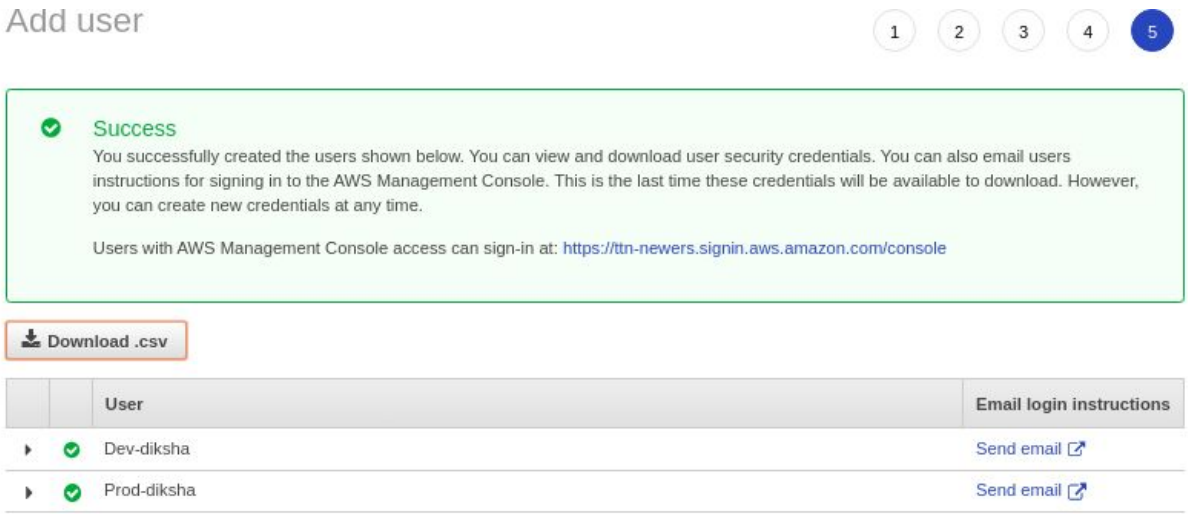
2)diksha_development

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



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 }
```

<u>STEP 4:</u> 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

Service IAM

Actions **Write**

ChangePassword	DeactivateMFADevice	EnableMFADevice
CreateAccessKey	DeleteAccessKey	ResyncMFADevice
CreateVirtualMFADevice	DeleteVirtualMFADevice	UpdateAccessKey

Resources close

- (●) Specific
- ( ) All resources

mfa You have not specified resource with type **mfa**
Add ARN to restrict access

sms-mfa You have not specified resource with type **sms-mfa**
Add ARN to restrict access

user arn:aws:iam::187632318301:user/${aws=username} EDIT

Add ARN to restrict access

Request conditions Specify request conditions (optional)

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "iam:DeactivateMFADevice",
                "iam:DeleteAccessKey",
                "iam:DeleteVirtualMFADevice",
                "iam:ResyncMFADevice",
                "iam:EnableMFADevice",
                "iam:UpdateAccessKey",
                "iam:CreateVirtualMFADevice",
                "iam:ChangePassword",
                "iam:CreateAccessKey"
            ],
            "Resource": "arn:aws:iam::187632318301:user/${aws=username}"
        }
    ]
```

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

Filter

Service	Access level	Resource	Request condition
Allow (1 of 223 services) Show remaining 222			
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