Lab 2: Identity and Access Management(IAM)

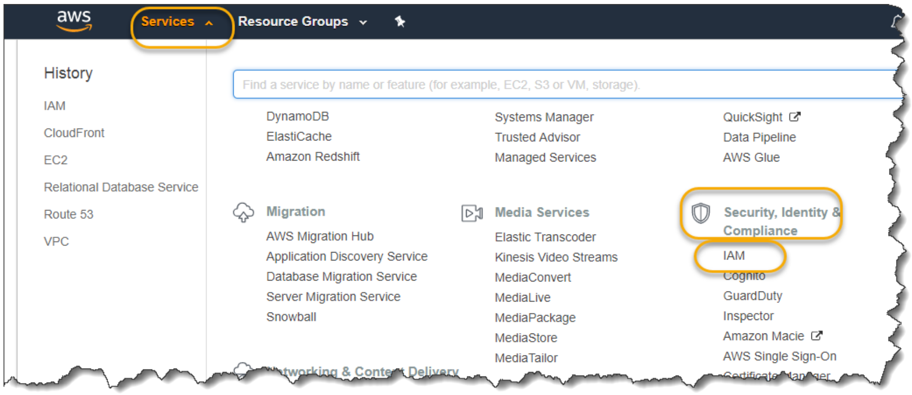
In this lab we are going to create some users attach an AWS managed policy to them. We are next going to create, attach and test customer managed policy on one of the users. We will next create and test an EC2 service role

Task Breakdown

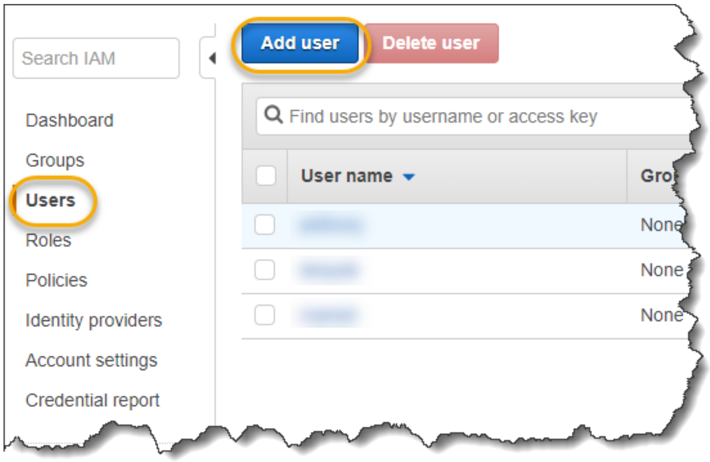
- Create users and attach Policies
- Log in using an IAM user
- Create a group and add users to the group
- Create a custom policy and test it out
- Create and use an IAM Role
- Manage IAM through CLI

Task 1: Create Users and attach Policies

1. Click **Services** and under **Security, Identity & Compliance** click on **IAM**

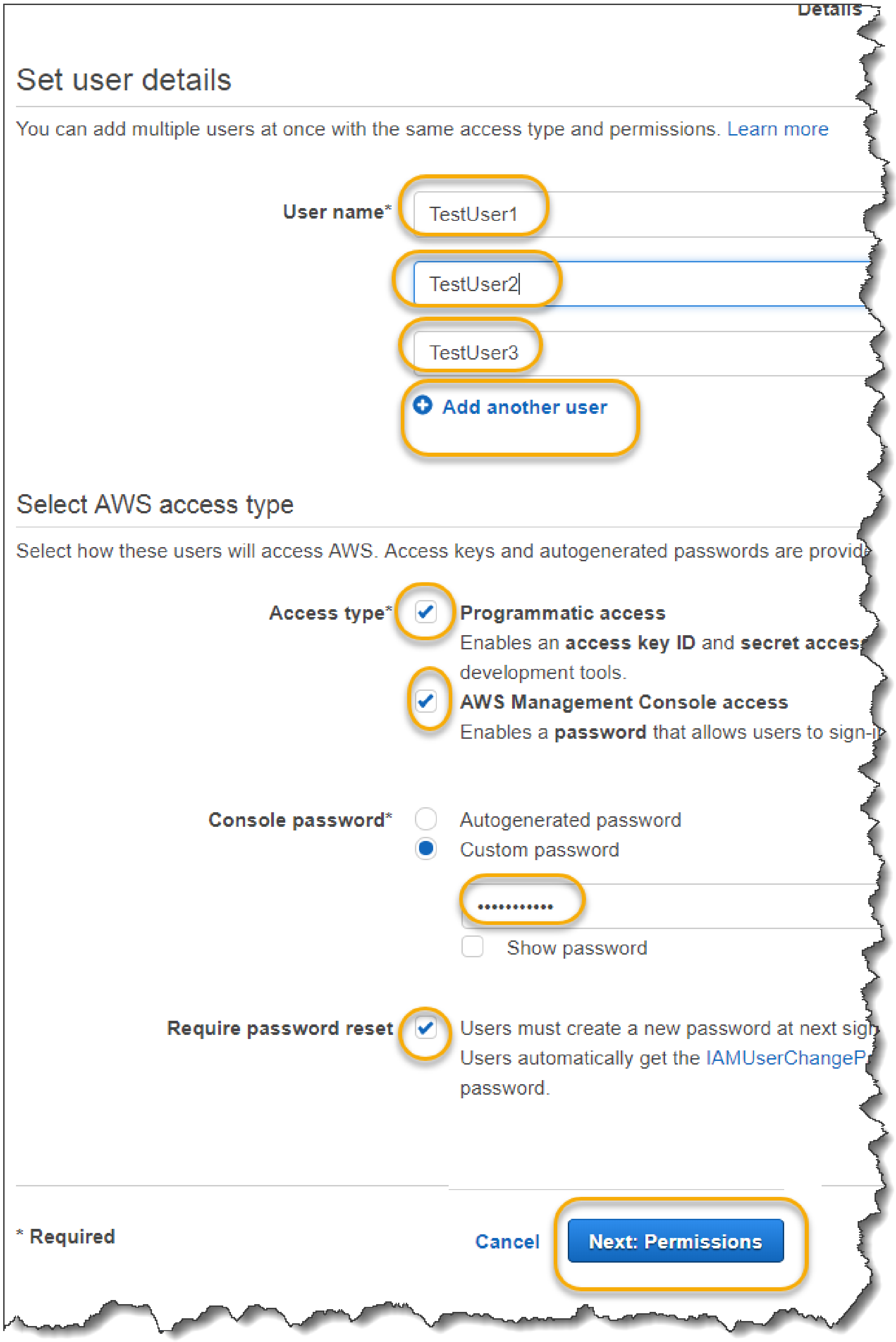


2. On the right on the screen click **Users** and then click **Add user**



□	User name ▾	Gro
□	[redacted]	None
□	[redacted]	None
□	[redacted]	None

3. Click **Add another user** and enter 3 usernames **Testuser1, Testuser2, Testuser3**. Select both **Programmatic** and **AWS Management Console access,** set a **Custom password** and click **Next: Permissions**



Details

Set user details

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

User name* TestUser1

TestUser2

TestUser3

Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provid

Access type* ☑ **Programmatic access**
Enables an **access key ID** and **secret acces**
development tools.

☑ **AWS Management Console access**
Enables a **password** that allows users to sign-i

Console password* ○ Autogenerated password
● Custom password

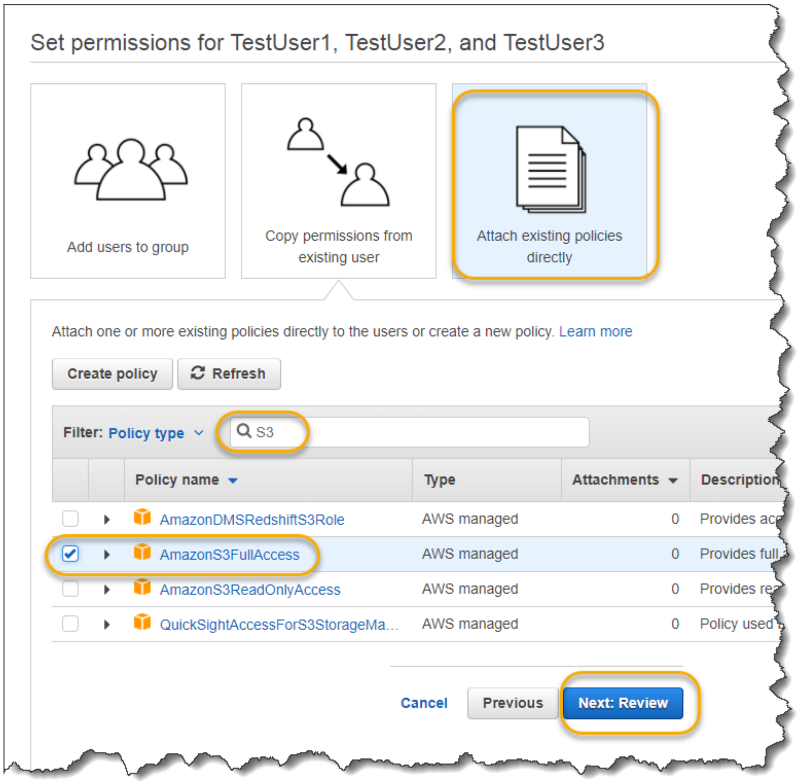
•••••••••••

□ Show password

Require password reset ☑ Users must create a new password at next sig
Users automatically get the IAMUserChangeP
password.

* Required

4. Click **Attach existing policies,** search for, find and select the **AmazonS3FullAccess policy** and click **Next: Review.**



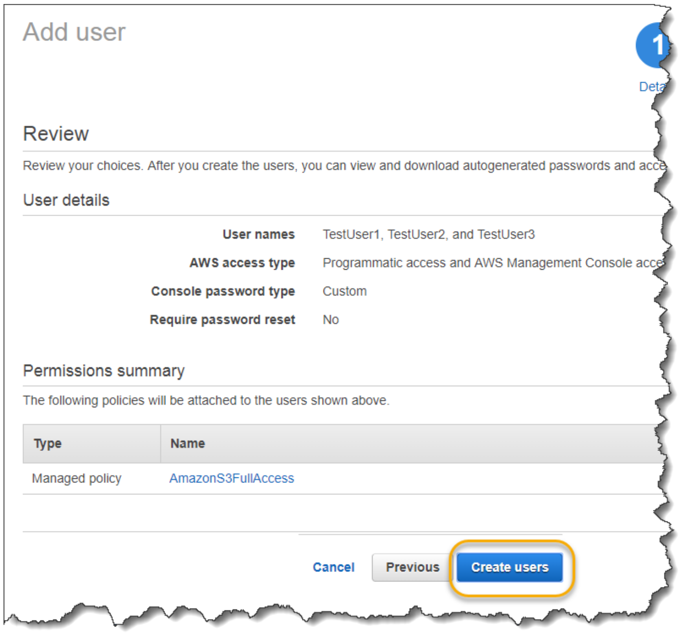
Set permissions for TestUser1, TestUser2, and TestUser3

Attach one or more existing policies directly to the users or create a new policy. Learn more

S3

		Policy name	Type	Attachments	Description
□	▸	AmazonDMSRedshiftS3Role	AWS managed	0	Provides ac
☑	▸	AmazonS3FullAccess	AWS managed	0	Provides full
□	▸	AmazonS3ReadOnlyAccess	AWS managed	0	Provides re
□	▸	QuickSightAccessForS3StorageMa...	AWS managed	0	Policy used

5. Click **Create Users.**



Add user

Review

Review your choices. After you create the users, you can view and download autogenerated passwords and acce

User details

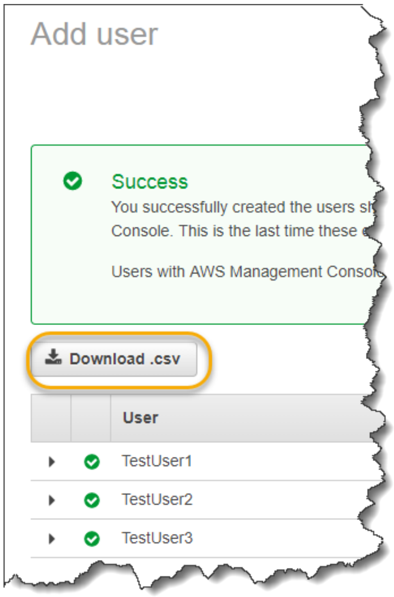
User names	TestUser1, TestUser2, and TestUser3
AWS access type	Programmatic access and AWS Management Console acce
Console password type	Custom
Require password reset	No

Permissions summary

The following policies will be attached to the users shown above.

Type	Name
Managed policy	AmazonS3FullAccess

6. Your users have been created successfully. Click **Download .csv**



Add user

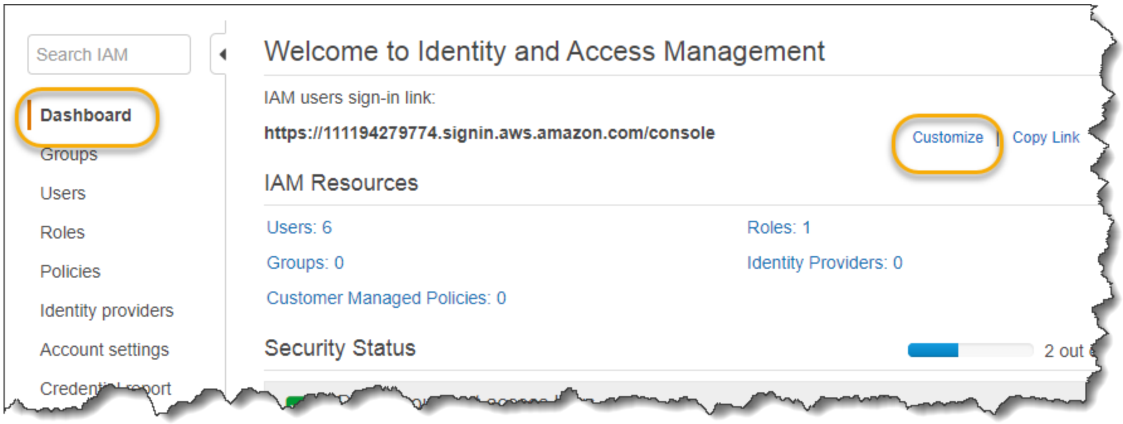
Success

You successfully created the users sh
Console. This is the last time these c

Users with AWS Management Consol

Task 2: Log in using an IAM user

1. On the main **IAM Dashboard** near the **IAM users sign-in** link click **Customize**



Welcome to Identity and Access Management

IAM users sign-in link:

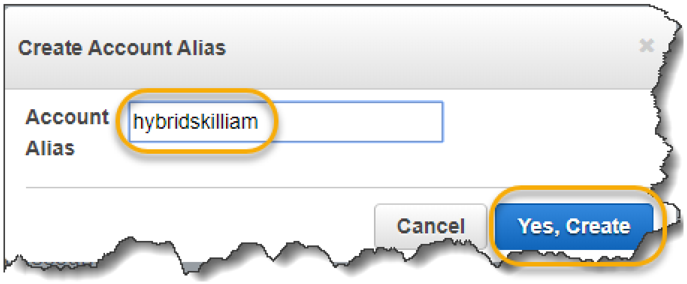
https://111194279774.signin.aws.amazon.com/console

IAM Resources

Users: 6 Roles: 1
Groups: 0 Identity Providers: 0
Customer Managed Policies: 0

Security Status 2 out

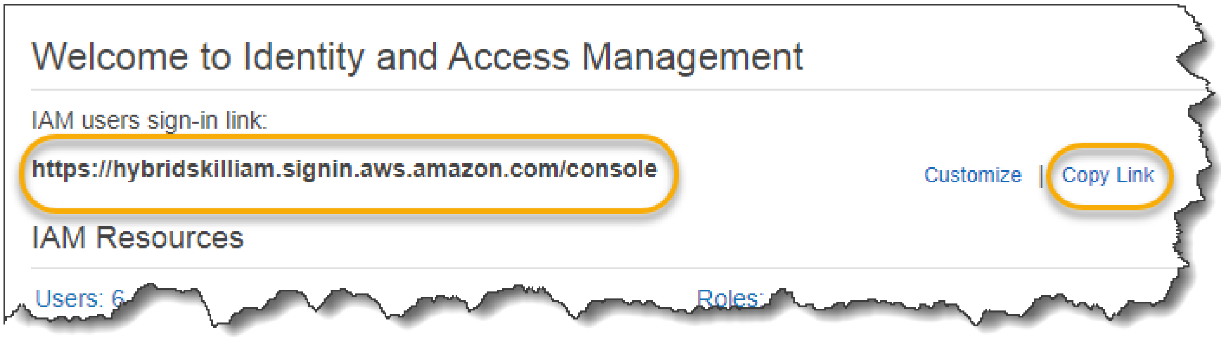
2. Enter a unique user-friendly **Account Alias** name and click **Yes, Create**.



Create Account Alias

Account Alias: hybridskilliam

3. This will be your IAM users sign-in link. Click **Copy Link.** Open an incognito browser session and visit the link

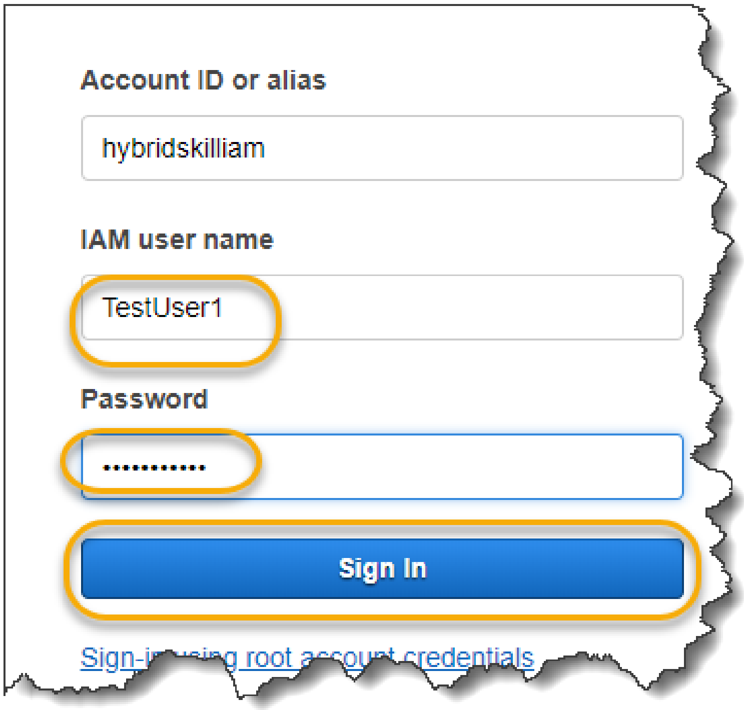


Welcome to Identity and Access Management

IAM users sign-in link:

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

4. Login in to **Testuser1's** account



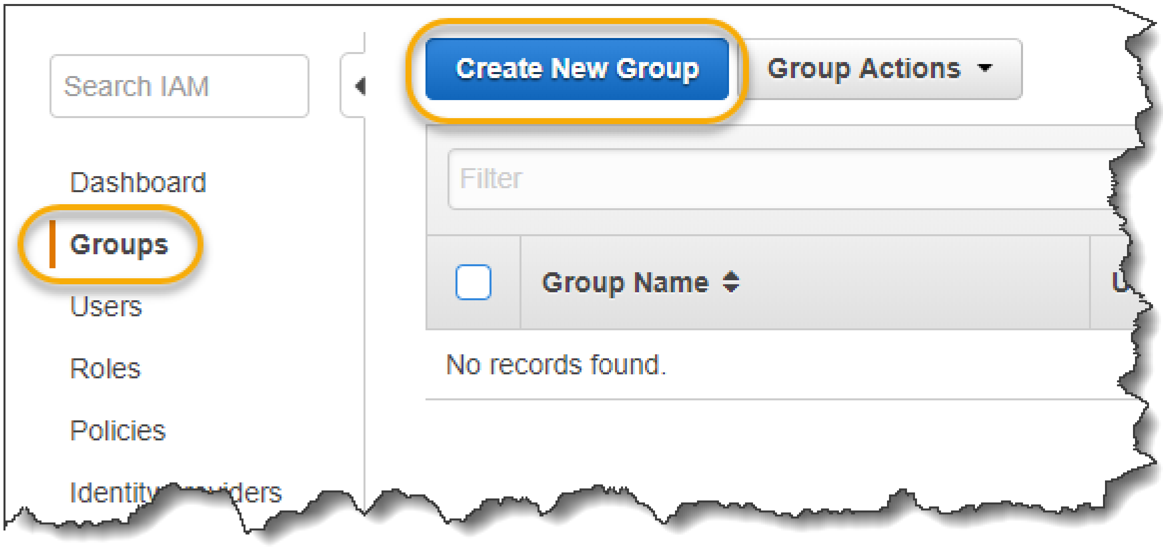
Account ID or alias: hybridskilliam

IAM user name: TestUser1

Password: •••••••••••

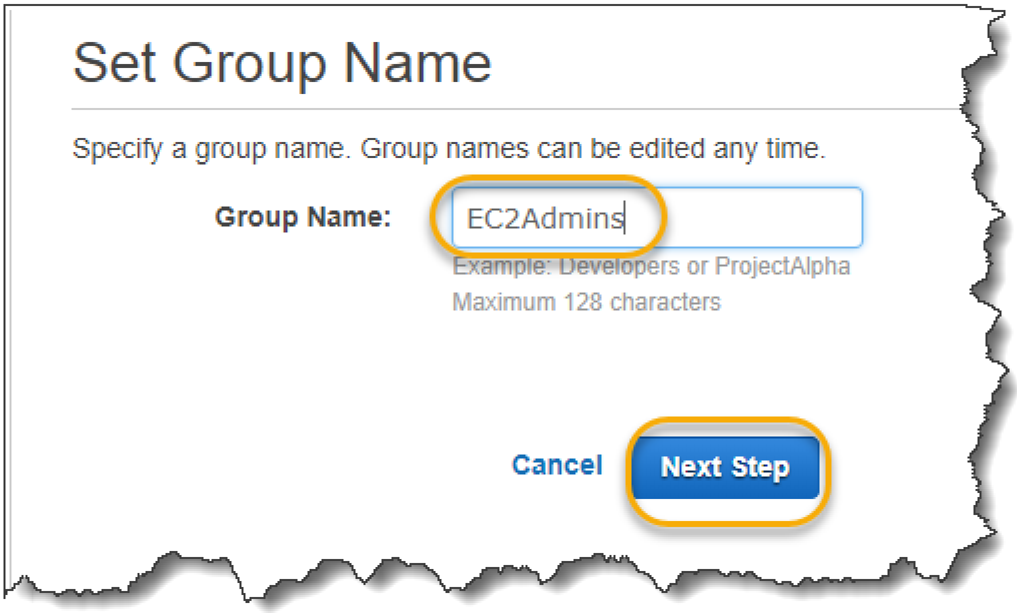
Task 3: Create a group and add users to the group

1. Click **Groups** and click **Create New Group**



No records found.

2. Enter **EC2Admin** as your **Group Name** and click **Next Step**



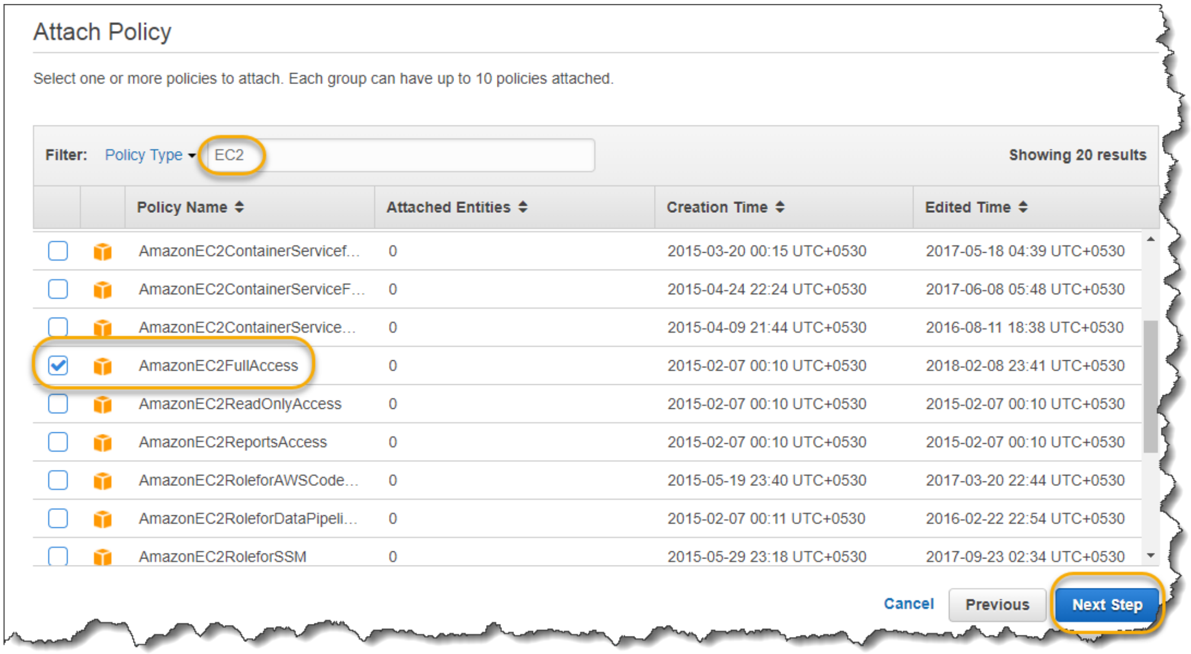
Set Group Name

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

Group Name: EC2Admins

Example: Developers or ProjectAlpha
Maximum 128 characters

3. Search for and select the **EC2FullAccess** policy and click **Next Step**



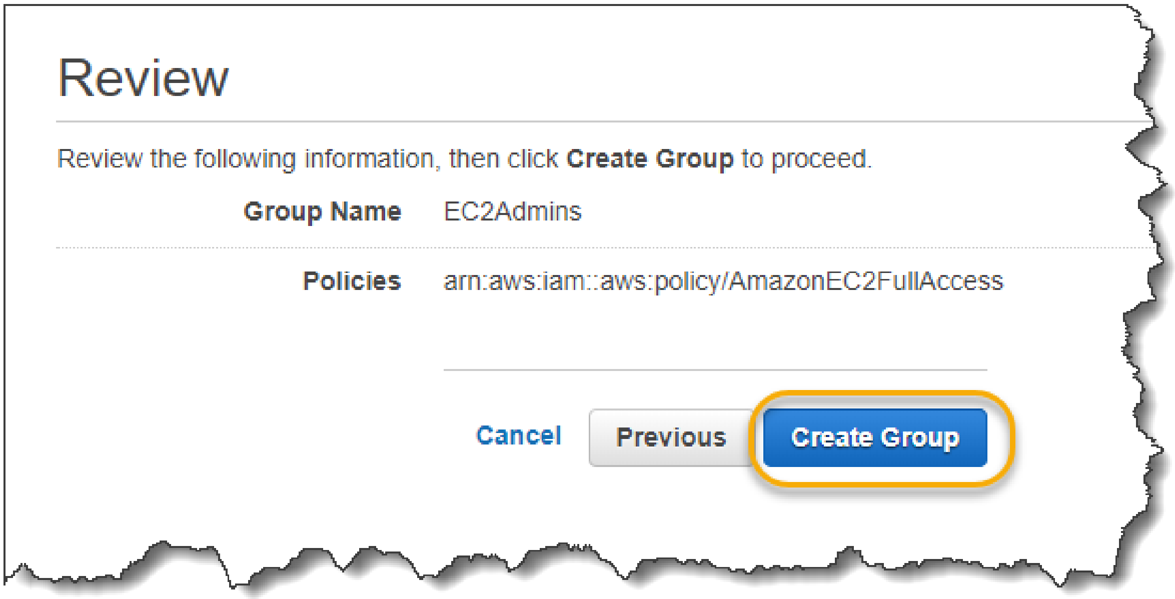
Attach Policy

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

Filter: Policy Type ▾ EC2 — Showing 20 results

	Policy Name ⬍	Attached Entities ⬍	Creation Time ⬍	Edited Time ⬍
□	AmazonEC2ContainerServicef...	0	2015-03-20 00:15 UTC+0530	2017-05-18 04:39 UTC+0530
□	AmazonEC2ContainerServiceF...	0	2015-04-24 22:24 UTC+0530	2017-06-08 05:48 UTC+0530
□	AmazonEC2ContainerService...	0	2015-04-09 21:44 UTC+0530	2016-08-11 18:38 UTC+0530
☑	AmazonEC2FullAccess	0	2015-02-07 00:10 UTC+0530	2018-02-08 23:41 UTC+0530
□	AmazonEC2ReadOnlyAccess	0	2015-02-07 00:10 UTC+0530	2015-02-07 00:10 UTC+0530
□	AmazonEC2ReportsAccess	0	2015-02-07 00:10 UTC+0530	2015-02-07 00:10 UTC+0530
□	AmazonEC2RoleforAWSCode...	0	2015-05-19 23:40 UTC+0530	2017-03-20 22:44 UTC+0530
□	AmazonEC2RoleforDataPipeli...	0	2015-02-07 00:11 UTC+0530	2016-02-22 22:54 UTC+0530
□	AmazonEC2RoleforSSM	0	2015-05-29 23:18 UTC+0530	2017-09-23 02:34 UTC+0530

4. Click **Create Group**

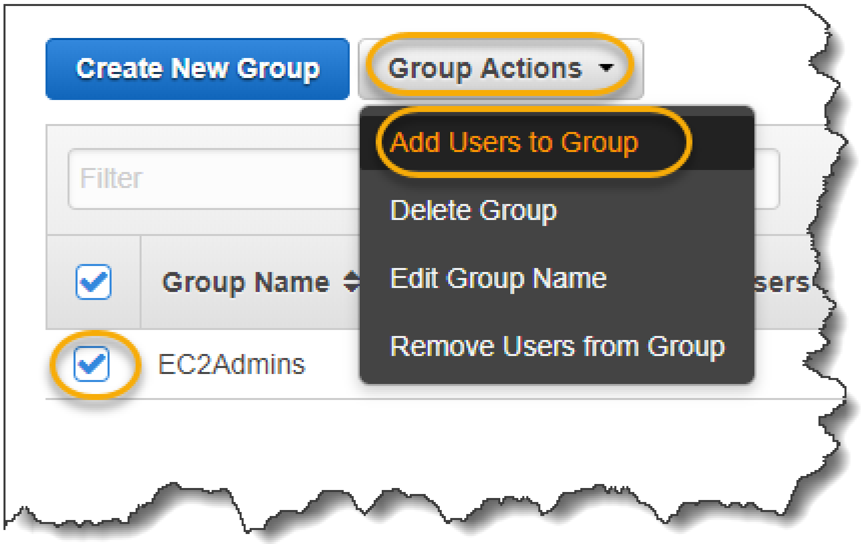


Review

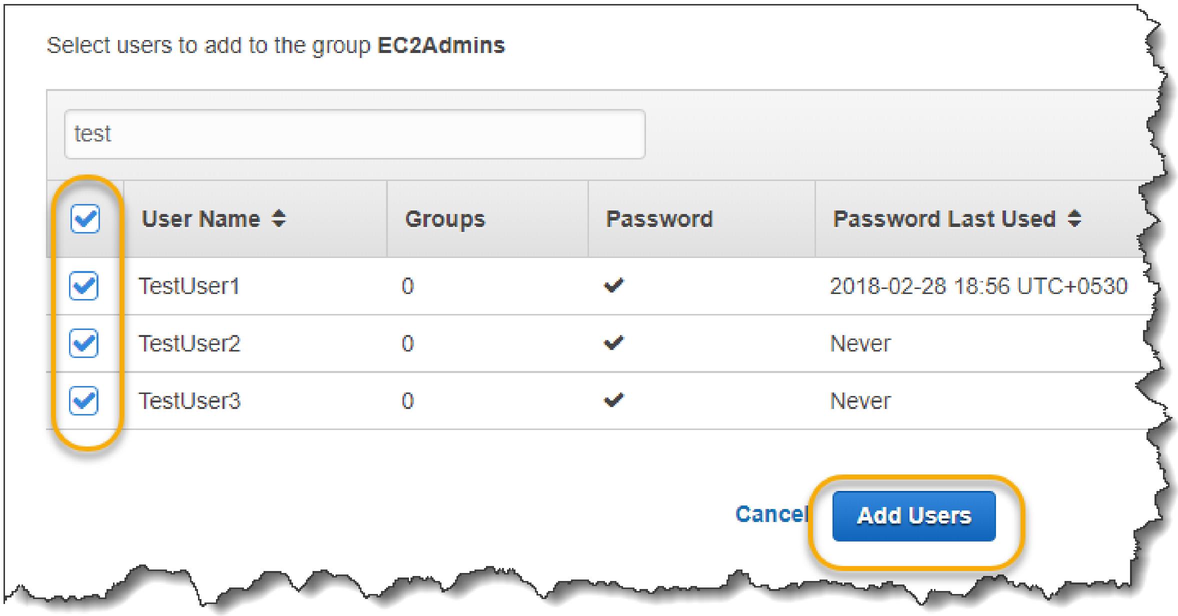
Review the following information, then click **Create Group** to proceed.

Group Name	EC2Admins
Policies	arn:aws:iam::aws:policy/AmazonEC2FullAccess

5. Select your group click **Group Actions** and click **Add Users to Group**



6. Select the 3 users you created earlier and click **Add Users**

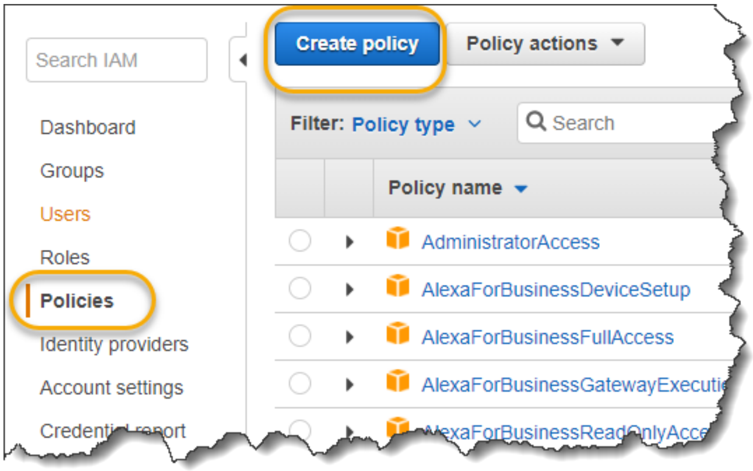

Select users to add to the group **EC2Admins**

test

☑	User Name ⬍	Groups	Password	Password Last Used ⬍
☑	TestUser1	0	✓	2018-02-28 18:56 UTC+0530
☑	TestUser2	0	✓	Never
☑	TestUser3	0	✓	Never

Task 4: Create a Custom policy

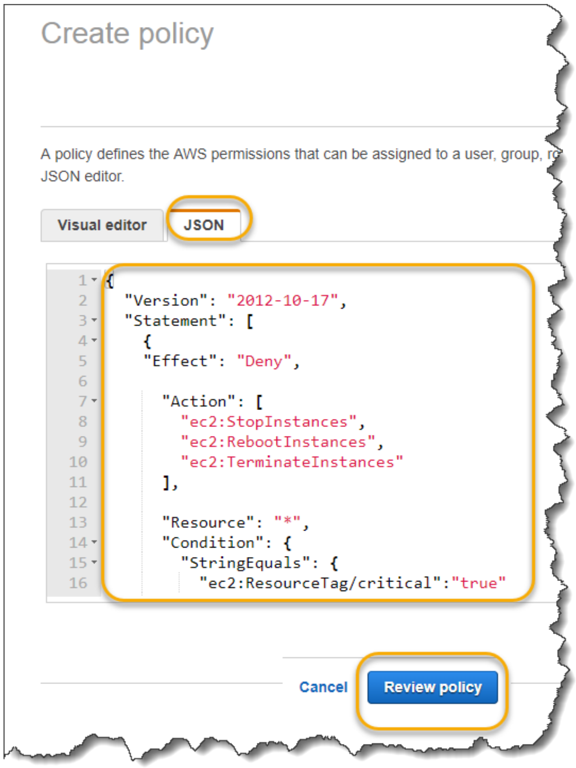
1. Click **Policies**, click **Create policy**



2. Download the policy from the following link. Open in Notepad++ and copy the code

https://s3.ap-south-1.amazonaws.com/hybridskill/ec2tagdeny.json

3. Click **JSON** and paste the code in the editor and click **Review policy**



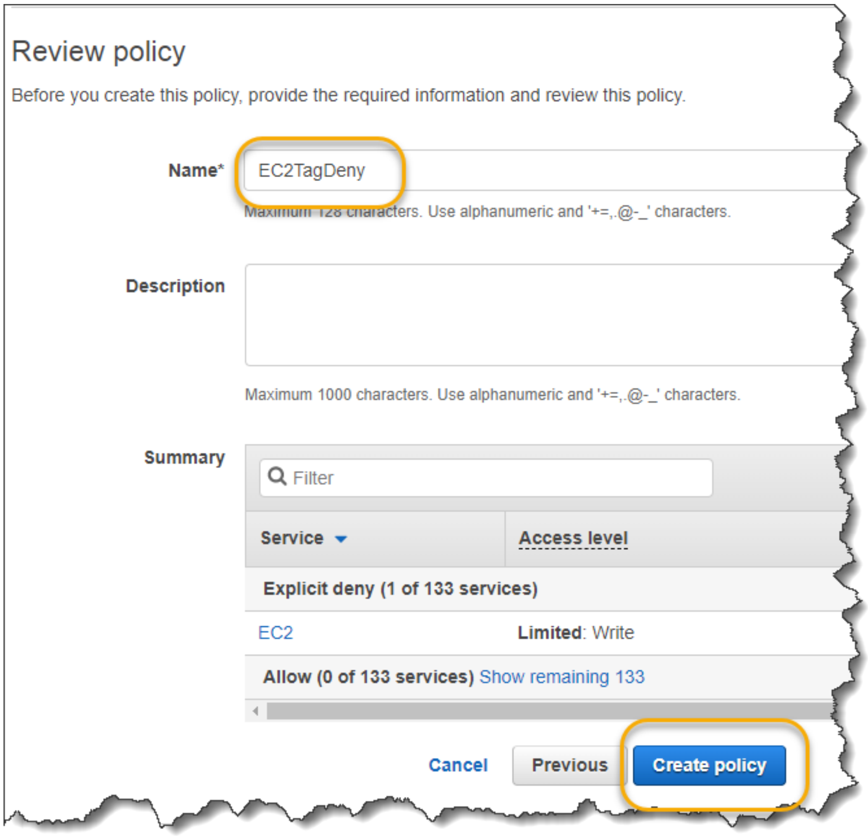
A policy defines the AWS permissions that can be assigned to a user, group, r
JSON editor.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
    "Effect": "Deny",

      "Action": [
        "ec2:StopInstances",
        "ec2:RebootInstances",
        "ec2:TerminateInstances"
      ],

      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/critical":"true"
```

4. Enter **EC2TagDeny** as the **Name** of the policy and click **Create Policy**



Review policy

Before you create this policy, provide the required information and review this policy.

Name* EC2TagDeny

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

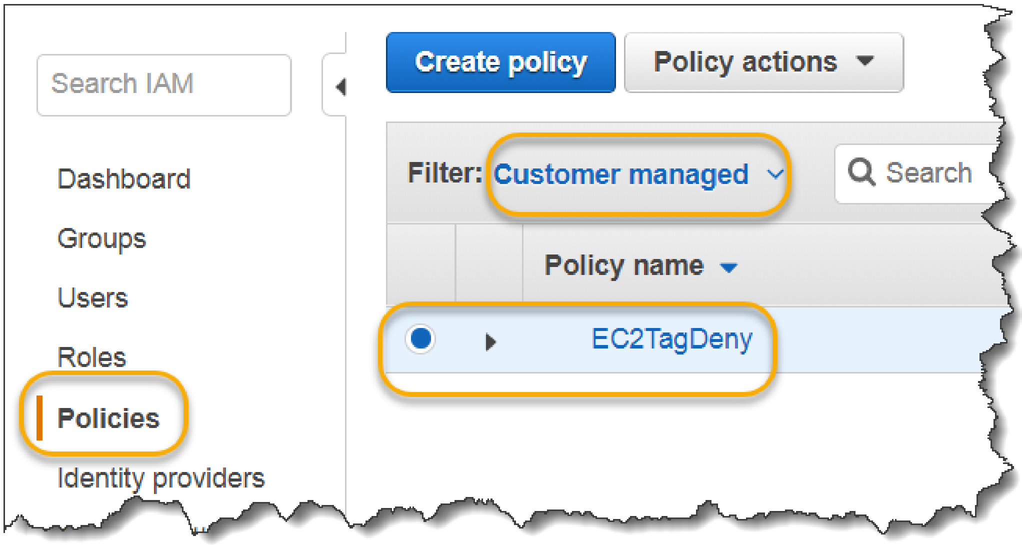
Description

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

Summary

Service	Access level
Explicit deny (1 of 133 services)	
EC2	**Limited**: Write
Allow (0 of 133 services) Show remaining 133	

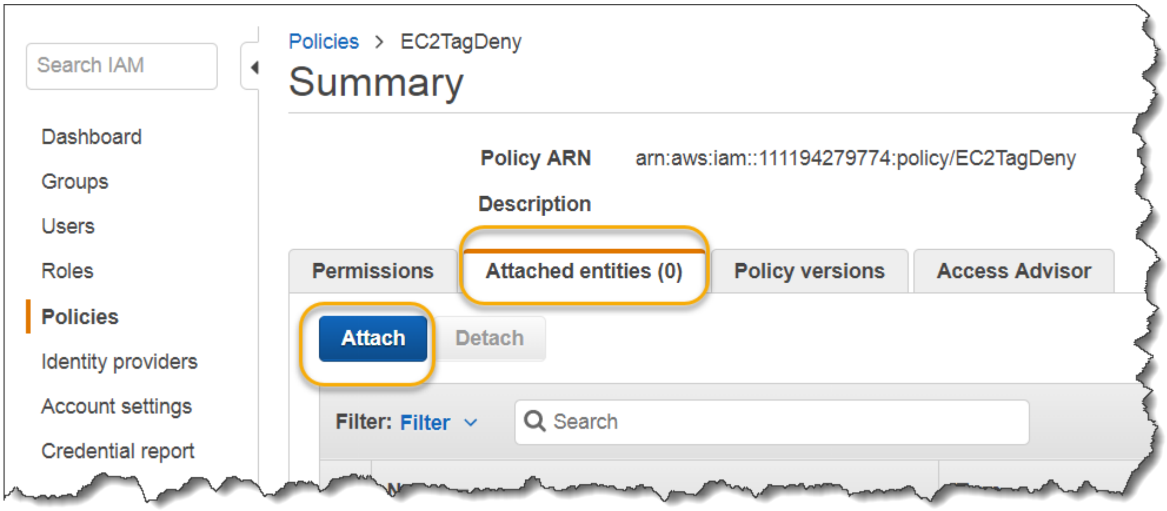
5. Click **Policies**, filter by **Customer managed policies** and click on the **EC2TagDeny** policy you created earlier to see more information about it



Filter: Customer managed

		Policy name
●	▸	EC2TagDeny

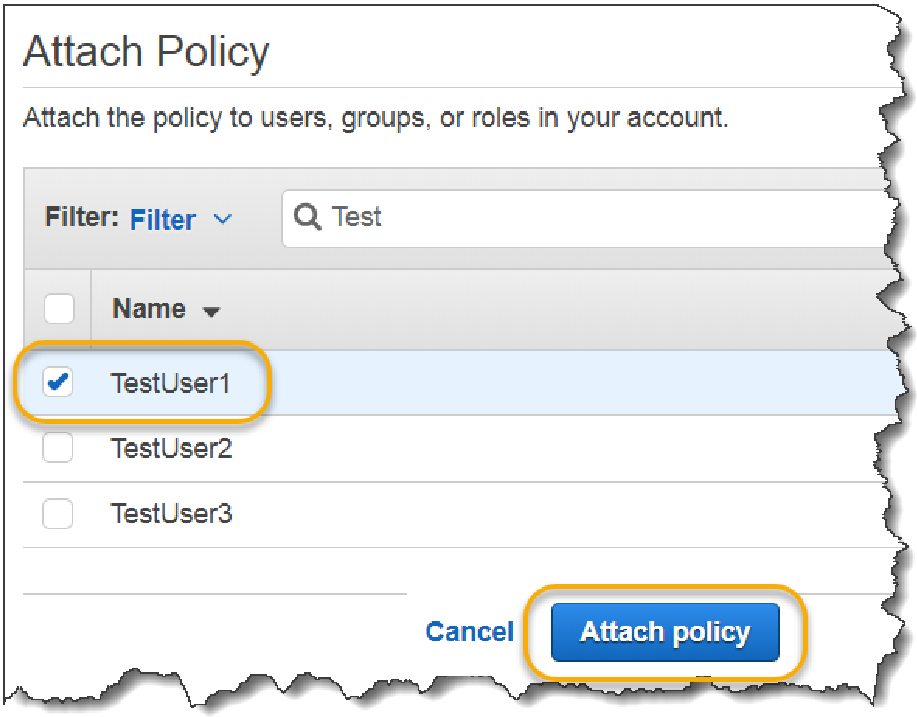
6. Click on the **Attached entities** tab and click **Attach**



Policy ARN arn:aws:iam::111194279774:policy/EC2TagDeny

Description

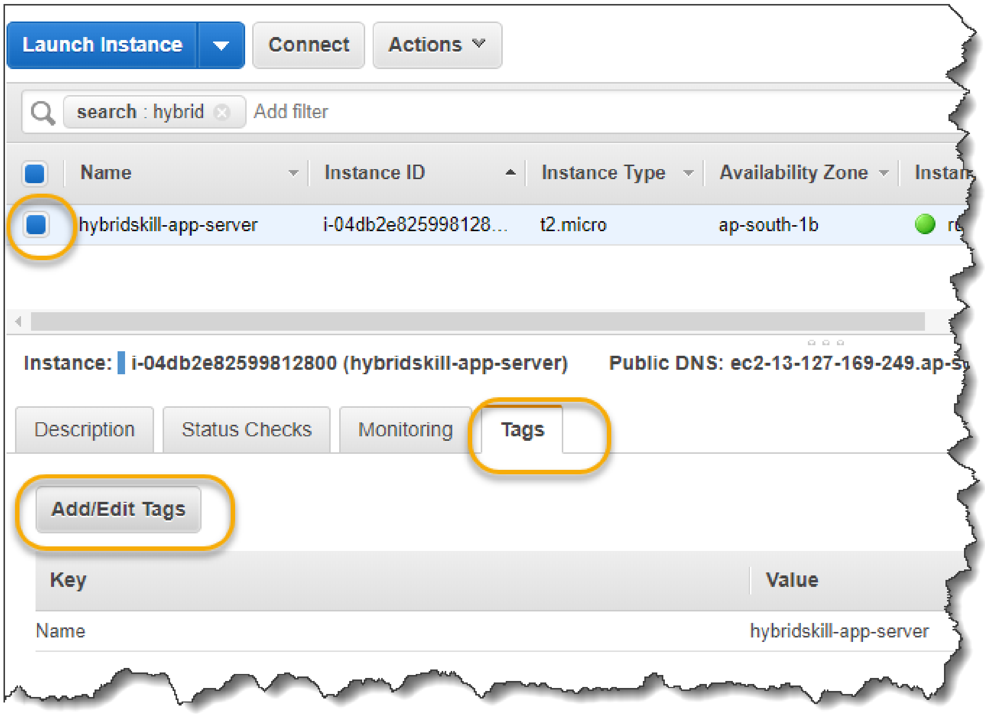
7. Select **TestUser1** and Click **Attach policy**



Attach Policy

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

8. Select an EC2 instance from your previous labs and on the lower half of the screen, click **Tags** and then **Add/Edit Tags**

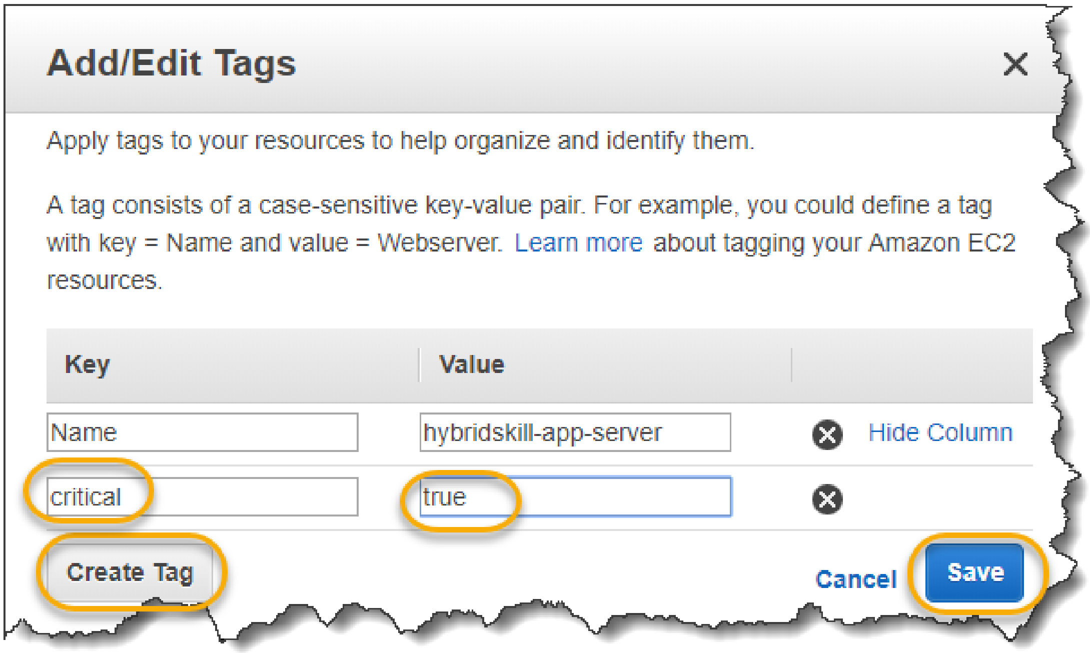


	Name	Instance ID	Instance Type	Availability Zone	Instan
■	hybridskill-app-server	i-04db2e825998128…	t2.micro	ap-south-1b	● r

Instance: i-04db2e82599812800 (hybridskill-app-server) Public DNS: ec2-13-127-169-249.ap-s

Key	Value
Name	hybridskill-app-server

9. Click **Create Tag,** enter **critical** as the **Key** and **true** as the **Value**. Click **Save**



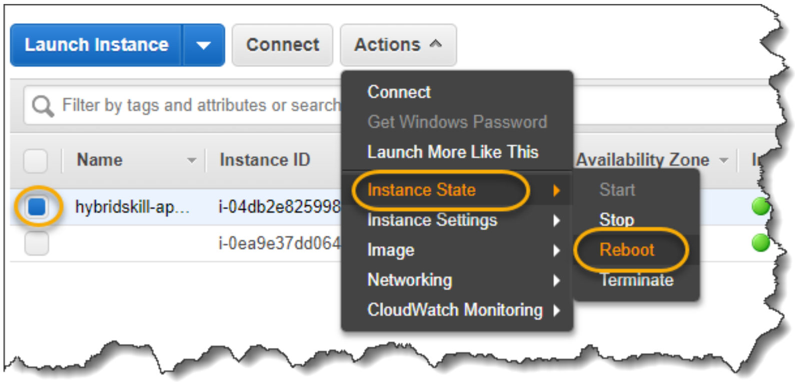
Add/Edit Tags

Apply tags to your resources to help organize and identify them.

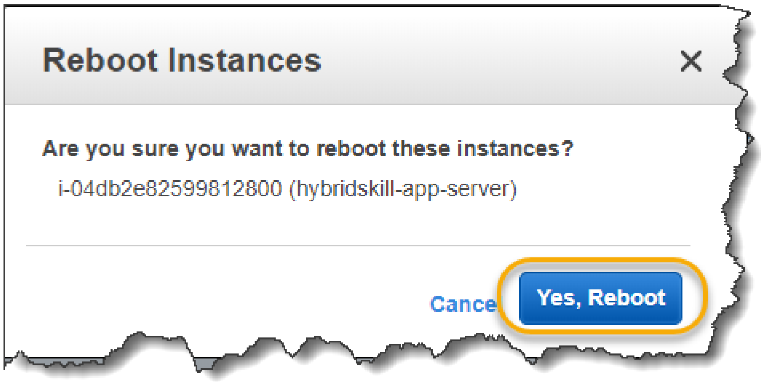
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key	Value	
Name	hybridskill-app-server	⊗ Hide Column
critical	true	⊗

10. On an incognito window login as **TestUser1. Try and Reboot, Stop** or **Terminate** the server



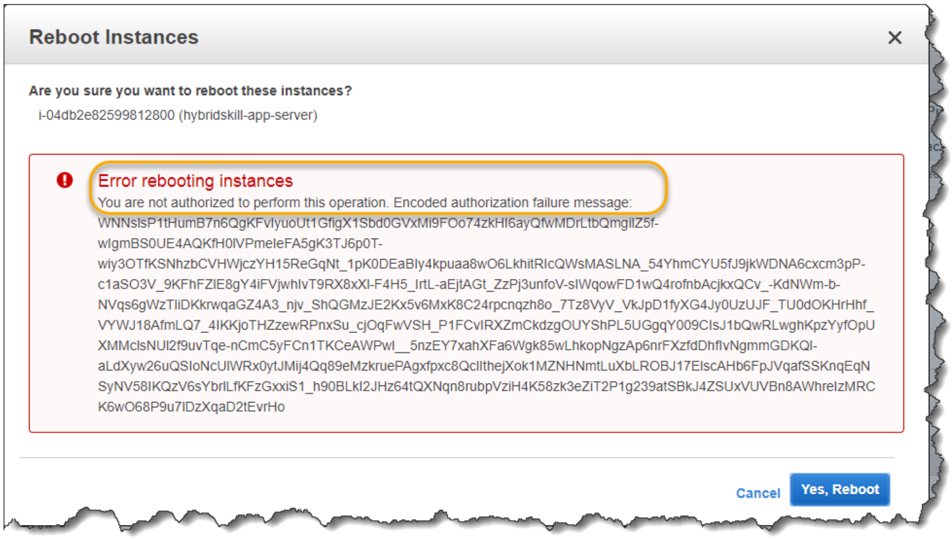
11. Let's try the reboot option first. Click **Reboot**


Reboot Instances

Are you sure you want to reboot these instances?

i-04db2e82599812800 (hybridskill-app-server)

12. You should get an error saying you are not authorized to do so.


Reboot Instances

Are you sure you want to reboot these instances?

i-04db2e82599812800 (hybridskill-app-server)

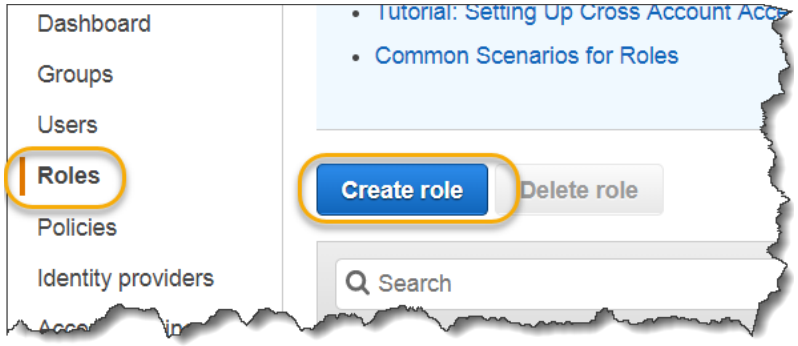
Error rebooting instances

You are not authorized to perform this operation. Encoded authorization failure message:

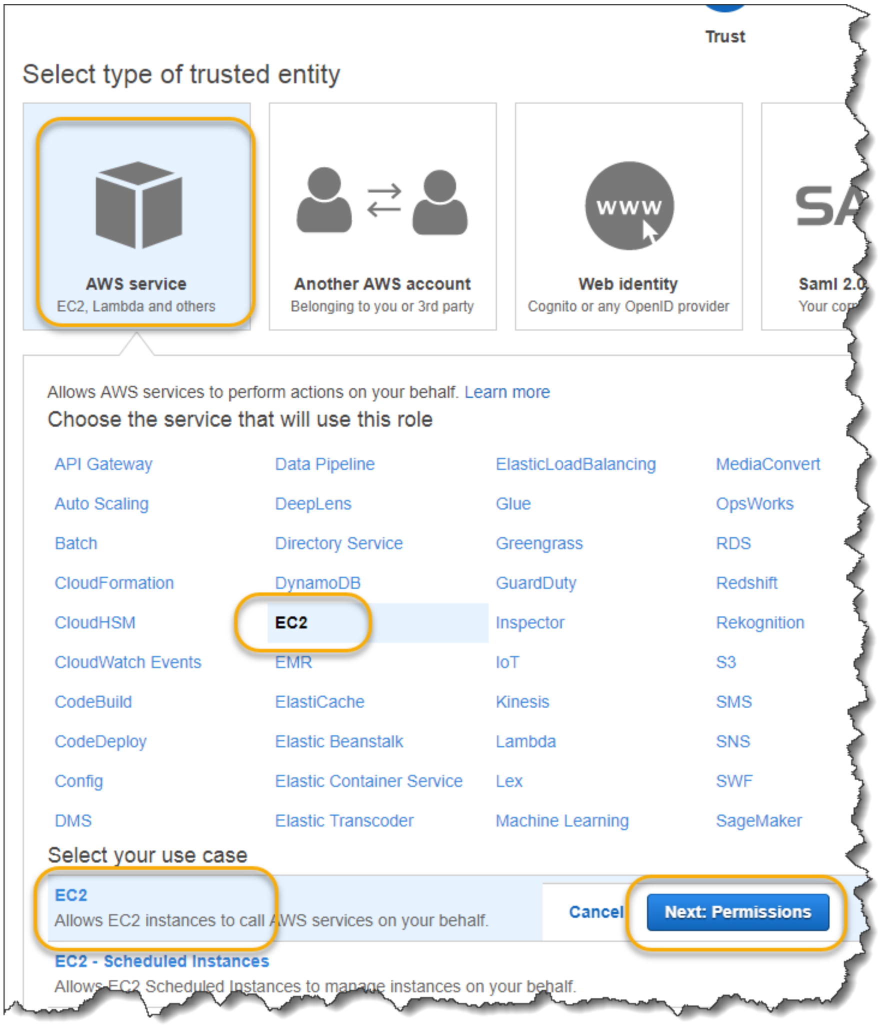
```
WNNslsP1tHumB7n6QgKFvlyuoUt1GfigX1Sbd0GVxMi9FOo74zkHl6ayQfwMDrLtbQmgilZ5f-
wlgmBS0UE4AQKfH0lVPmeleFA5gK3TJ6p0T-
wiy3OTfKSNhzbCVHWjczYH15ReGqNt_1pK0DEaBly4kpuaa8wO6LkhitRlcQWsMASLNA_54YhmCYU5fJ9jkWDNA6cxcm3pP-
c1aSO3V_9KFhFZlE8gY4iFVjwhlvT9RX8xXl-F4H5_lrtL-aEjtAGt_ZzPj3unfoV-slWqowFD1wQ4rofnbAcjkxQCv_-KdNWm-b-
NVqs6gWzTliDKkrwqaGZ4A3_njv_ShQGMzJE2Kx5v6MxK8C24rpcnqzh8o_7Tz8VyV_VkJpD1fyXG4Jy0UzUJF_TU0dOKHrHhf_
VYWJ18AfmLQ7_4IKKjoTHZzewRPnxSu_cjOqFwVSH_P1FCvlRXZmCkdzgOUYShPL5UGgqY009ClsJ1bQwRLwghKpzYyfOpU
XMMclsNUl2f9uvTqe-nCmC5yFCn1TKCeAWPwl__5nzEY7xahXFa6Wgk85wLhkopNgzAp6nrFXzfdDhflvNgmmGDKQl-
aLdXyw26uQSIoNcUlWRx0ytJMij4Qq89eMzkruePAgxfpxc8QclIthejXok1MZNHNmtLuXbLROBJ17ElscAHb6FpJVqafSSKnqEqN
SyNV58IKQzV6sYbrlLfKFzGxxiS1_h90BLkl2JHz64tQXNqn8rubpVziH4K58zk3eZiT2P1g239atSBkJ4ZSUxVUVBn8AWhrelzMRC
K6wO68P9u7lDzXqaD2tEvrHo
```

Task 5: Create and use an IAM Role

1. Click **Roles** and **Create role**



2. Click **AWS service** and click on **EC2,** select your **use case** as **EC2** and click **Next: Permission**



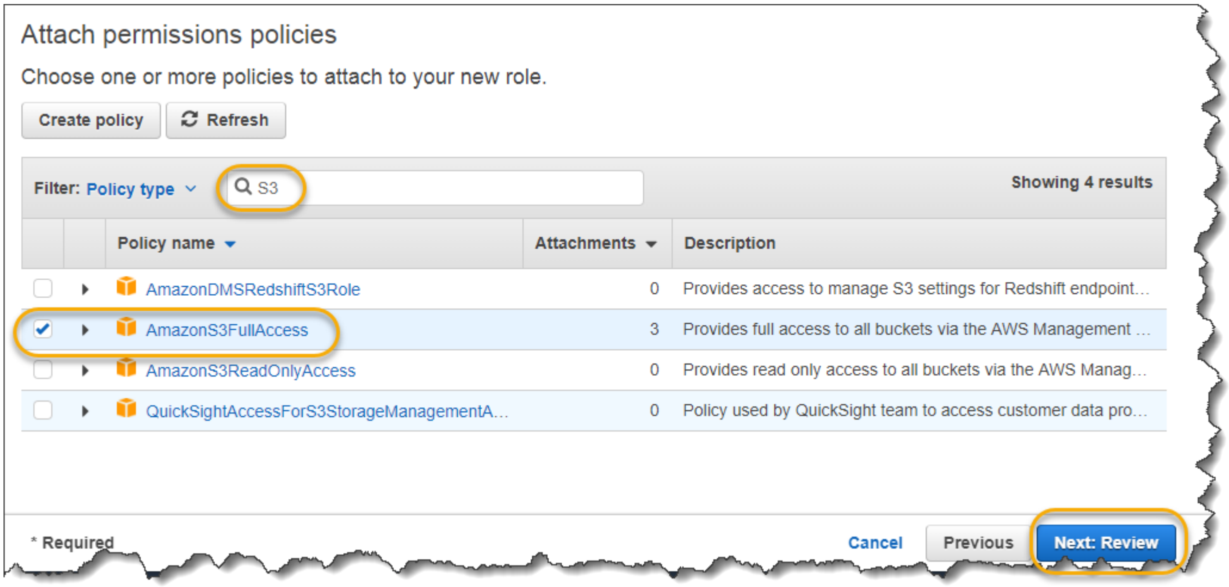
Allows AWS services to perform actions on your behalf. Learn more

Choose the service that will use this role

Allows EC2 instances to call AWS services on your behalf.

Allows EC2 Scheduled Instances to manage instances on your behalf.

3. Search for select the **AmazonS3FullAccess** for **Next: Review**



Attach permissions policies

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

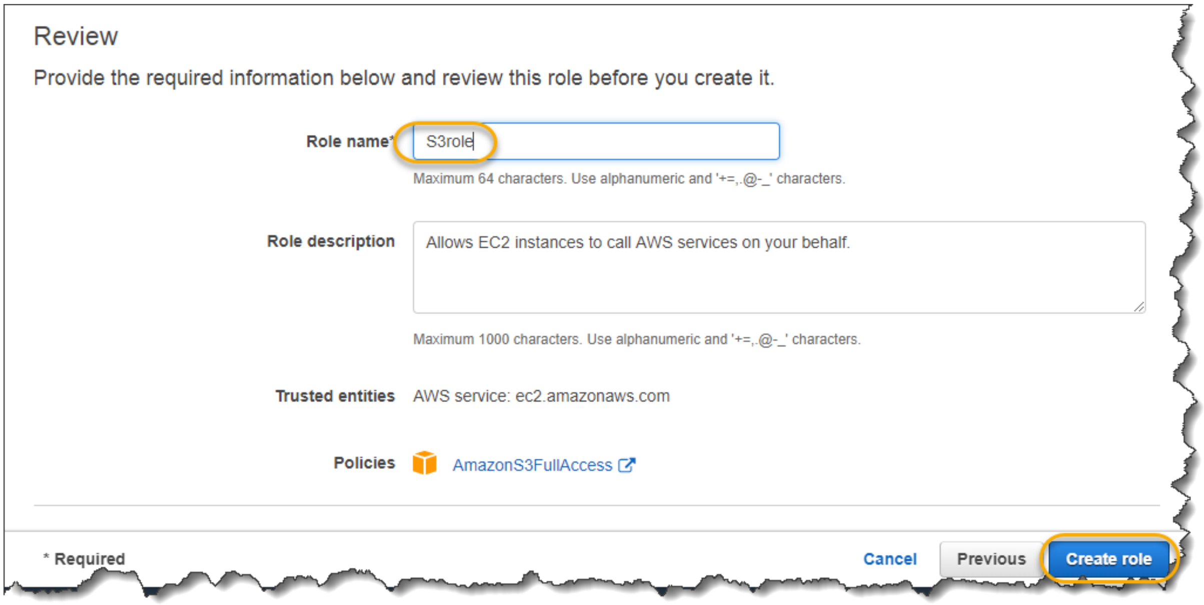
Filter: Policy type S3

Showing 4 results

		Policy name	Attachments	Description
□	▸	AmazonDMSRedshiftS3Role	0	Provides access to manage S3 settings for Redshift endpoint…
☑	▸	AmazonS3FullAccess	3	Provides full access to all buckets via the AWS Management …
□	▸	AmazonS3ReadOnlyAccess	0	Provides read only access to all buckets via the AWS Manag…
□	▸	QuickSightAccessForS3StorageManagementA…	0	Policy used by QuickSight team to access customer data pro…

* Required

4. Enter **S3role** and click **Create role**



Review

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

Role name* S3role

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

Role description Allows EC2 instances to call AWS services on your behalf.

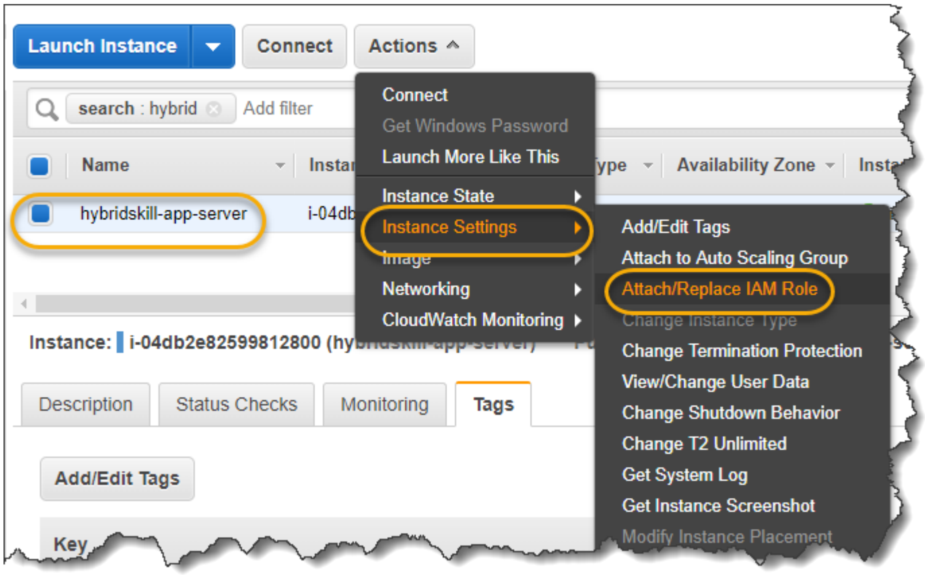
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Trusted entities AWS service: ec2.amazonaws.com

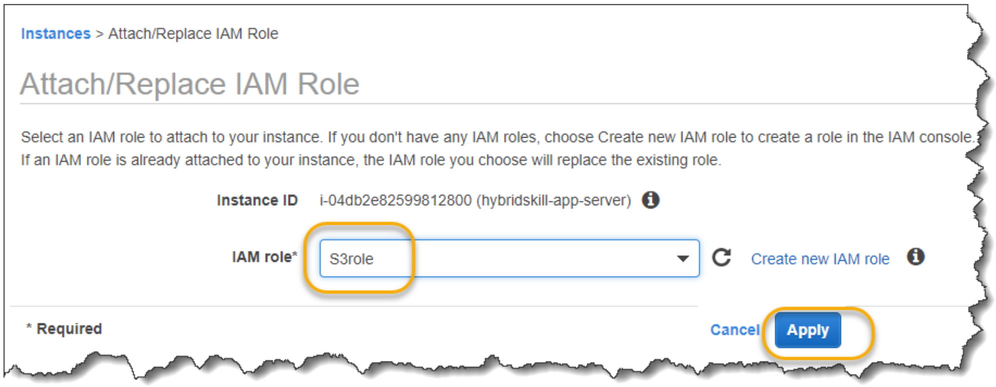
Policies AmazonS3FullAccess

* Required

5. Select one of your instances click **Actions**, **Instance Settings** and click **Attach/Replace IAM Role**



6. Select your **IAM Role** and click **Apply**



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-04db2e82599812800 (hybridskill-app-server)

IAM role* S3role

7. Log into the EC2 instance Install AWS CLI on and run the following commands

```
aws configure
```

8. Set up only your region name do not provide Access Key and Secret Key Next try listing your buckets with the list command. You should be able to do so without providing keys.

```
aws s3 ls
```

9. Try and fetch the current roles STS token

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/S3role
```

Task 6: Manage IAM using CLI

Now that we have explored IAM through the console, let’s do the same through the CLI. Run the following commands on the command line interface, you setup earlier.

1. Create user

```
:~$ aws iam create-user --user-name test1
{
    "User": {
        "Path": "/",
        "UserName": "test1",
        "UserId": "AIDAJ4DSAFASDFSDFZZH3EQ",
        "Arn": "arn:aws:iam::123456789123:user/test1",
        "CreateDate": "2018-03-02T12:56:06.190Z"
    }
}
```

2. Attach policy to user

```
:~$ aws iam attach-user-policy --user-name test1 --policy-arn
arn:aws:iam::aws:policy/AmazonS3FullAccess
```

3. Assign password for console access

```
:~$ aws iam create-login-profile --user-name test1 --password test1@hybridskill
{
    "LoginProfile": {
        "UserName": "test1",
        "CreateDate": "2018-03-04T17:20:12.429Z",
        "PasswordResetRequired": false
    }
}
```

4. Create group and assign policy and add user

```
:~$ aws iam create-group --group-name Admins
{
    "Group": {
        "Path": "/",
        "GroupName": "Admins",
        "GroupId": "AGPAJWPEY3UGNUKOJ2PIA",
        "Arn": "arn:aws:iam::123456789123:group/Admins",
        "CreateDate": "2018-03-04T17:22:38.815Z"
    }
}

:~$ aws iam attach-group-policy  --group-name Admins --policy-arn
arn:aws:iam::aws:policy/AmazonEC2FullAccess

:~$ aws iam add-user-to-group --group-name Admins --user-name test1
```

5. Create a policy file with Deny

```
:~$ cat Ec2TagDenyPolicy
{
  "Version": "2012-10-17",
  "Statement": [
    {
    "Effect": "Deny",
      "Action": [
        "ec2:StopInstances",
        "ec2:RebootInstances",
        "ec2:TerminateInstances"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/critical":"true"
        }
      }


    }
  ]
}
```

6. Attach policy to the users

```
:~$ aws iam put-user-policy --user-name test1 --policy-name Ec2TagDenyPolicy --policy-
document file://./Ec2TagDenyPolicy
```

7. Generate access and secret key for awscli

```
:~$ aws iam create-access-key --user-name test1
{
    "AccessKey": {
        "UserName": "test1",
        "AccessKeyId": "AKIAJDF3324FDAC473A2JUQ",
        "Status": "Active",
        "SecretAccessKey": "LfMGhITwuElFDSFDS31EG4FSGDFKCX3",
        "CreateDate": "2018-03-04T17:54:05.036Z"
    }
}
```

8. Set a tag on EC2 box and check if policy is working fine

```
:~$ aws ec2 create-tags --resources   i-04f8608c2e791aa76  --tags
Key=critical,Value=true
```

9. Save current creds from **~/.aws/credentials** file and Configure awscli creds for test1 user

```
:~$ aws configure
```

10. Try rebooting machine

```
:~$ aws ec2 reboot-instances --instance-id i-04f8608c2e791aa76
```

11. Create a role. First create a trusted policy policy.json

```
:~$ cat policy.json

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

12. Create role

```
:~$ aws iam create-role --role-name  s3FullAccess --assume-role-policy-document  file://./policy.json
```

13. Attach policy to role:

```
:~$ aws iam attach-role-policy --role-name s3FullAccess --policy-arn arn:aws:iam::aws:policy/AmazonS3FullAccess
```

14. Create instance profile

```
:~$ aws iam create-instance-profile --instance-profile-name s3FullAccess
```

15. Attach role to profile

```
:~$ aws iam add-role-to-instance-profile --role-name s3FullAccess --instance-profile-name s3FullAccess
```

16. Log into the EC2 instance Install AWS CLI on it and run the following commands

```
:~$ aws s3 ls
```