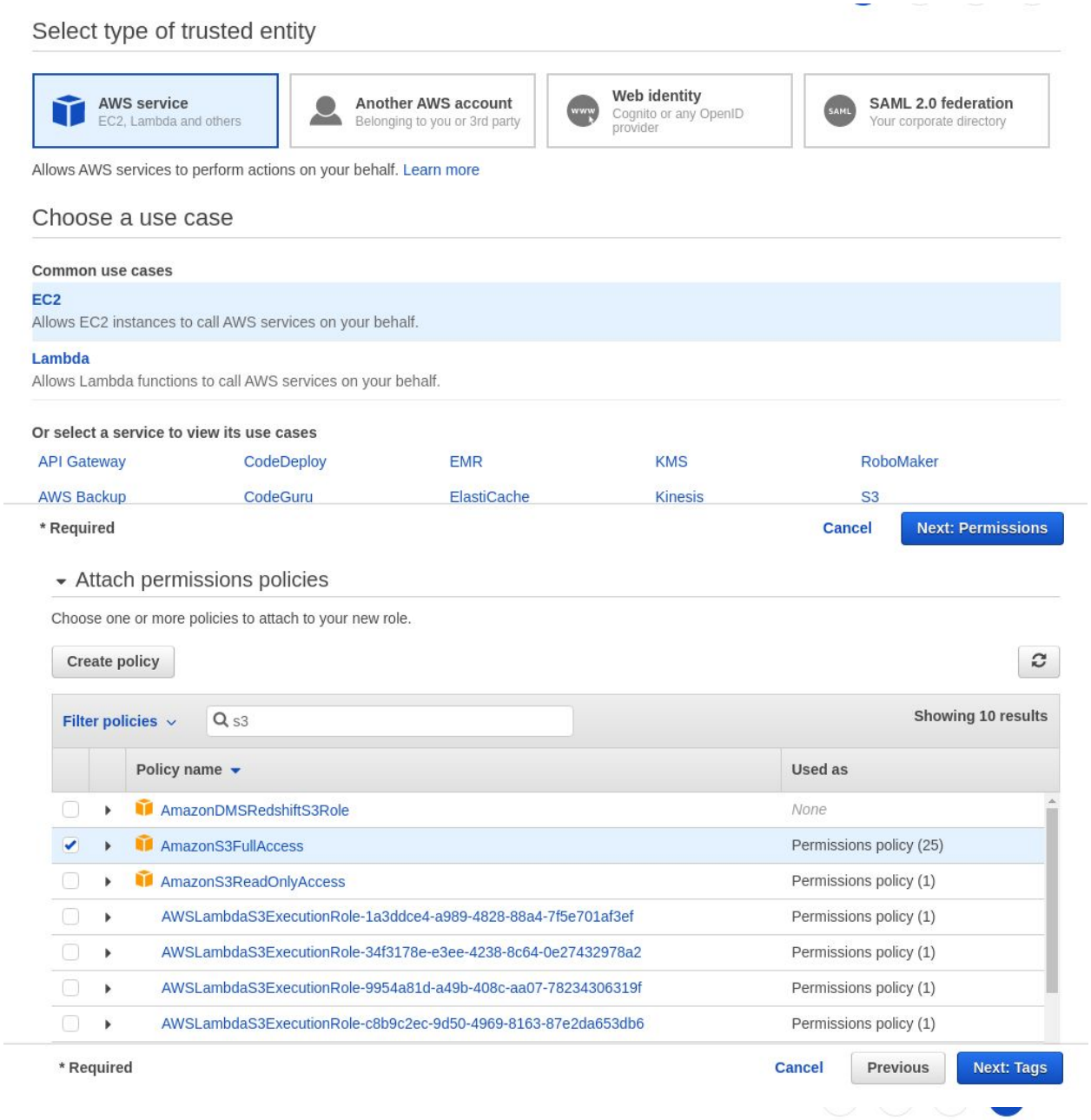
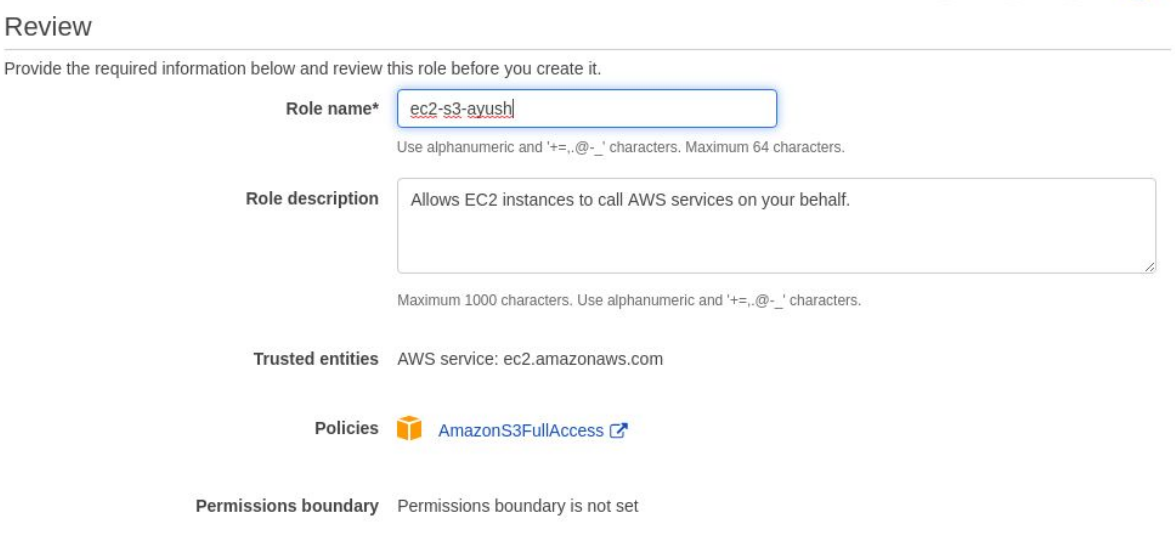
1. Create a Role with full access to S3





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

Assuming a role means asking Security Token Service (STS) to provide you with a set of temporary credentials -- role credentials -- that are specific to the role you want to assume. (Specifically, a new "session" with that role.)

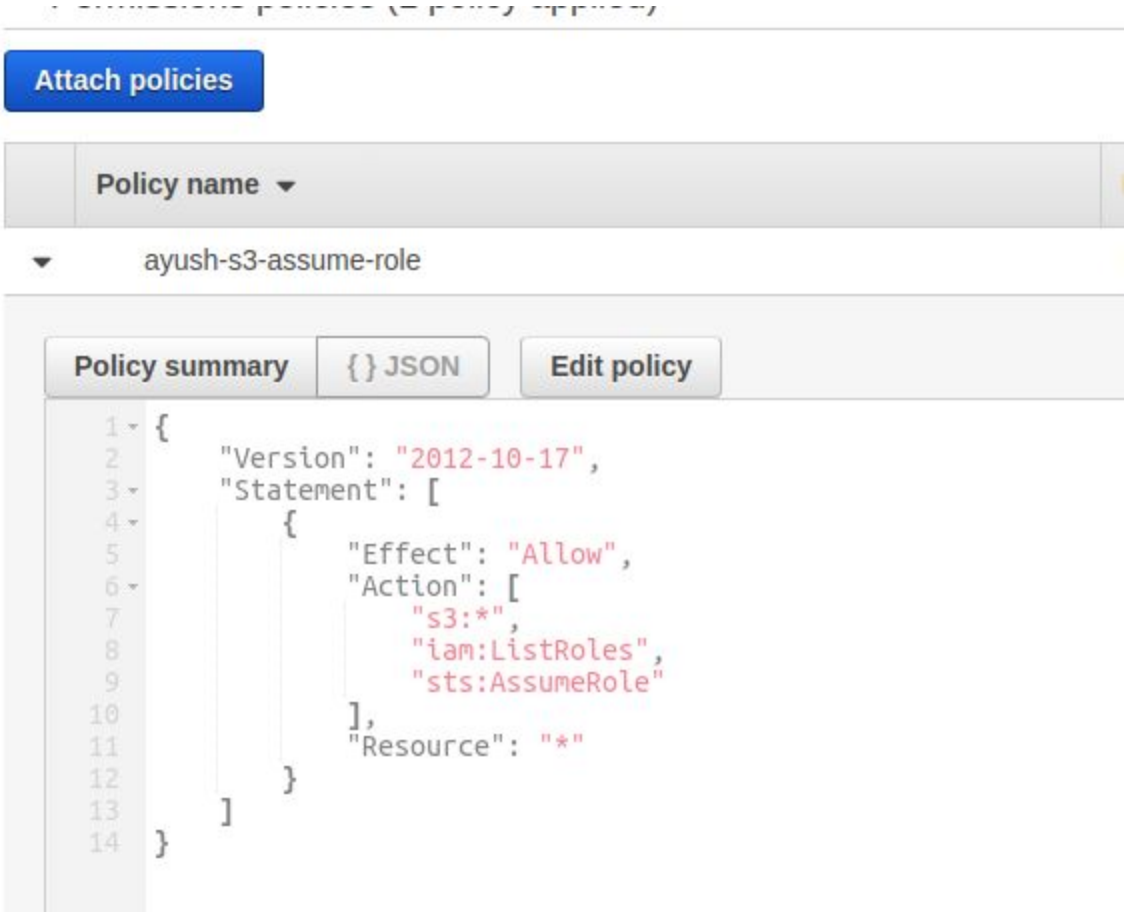
You can optionally include a policy with this request, which will serve to limit the permissions of the temporary credentials to only a subset of what the role's policies would have allowed.

You then use these credentials to make further requests. These credentials look similar to IAM user credentials with an access-key-id and secret, but the access key begins with `ASIA` instead of `AKIA` and there's a third element, called the security token, which must be included in requests signed with the temporary credentials.

When you make requests with these temporary credentials, you have the permissions associated with the role, and not your own (if you have one) because you have taken on a new identity. CloudTrail can be used to trace the role credentials back to the user who assumed the role, but otherwise the service is unaware of who is using the credentials.

tl;dr: Assuming a role means obtaining a set of temporary credentials which are associated with the role and not with the entity that assumed the role.

Create a role from previous role and add sts:AssumeRole in aaction



Refrence:https://aws.amazon.com/premiumsupport/knowledge-center/iam-assume-role-cli/

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

Attach this role to first ec2 instance
Generate sts credentials for second instance

```
ubuntu@ip-172-31-71-45:~$ aws iam list-roles --query "Roles[?RoleName == 'ec2-s3-inatance'].[RoleNa
me, Arn]"
[
    [
        "ec2-s3-inatance",
        "arn:aws:iam::187632318301:role/ec2-s3-inatance"
    ]
]
ubuntu@ip-172-31-71-45:~$ aws sts assume-role --role-arn "arn:aws:iam::123456789012:role/example-ro
le" --role-session-name AWSCLI-Session

An error occurred (AccessDenied) when calling the AssumeRole operation: User: arn:aws:sts::18763231
8301:assumed-role/ec2-s3-inatance/i-0a5bb40810a8795d0 is not authorized to perform: sts:AssumeRole
on resource: arn:aws:iam::123456789012:role/example-role
ubuntu@ip-172-31-71-45:~$ aws sts assume-role --role-arn ""arn:aws:iam::187632318301:role/ec2-s3-in
atance"" --role-session-name AWS-assume-role
{
    "Credentials": {
        "AccessKeyId": "ASIASXL6B65OV7WHOIE6",
        "SecretAccessKey": "N3cagQtTZr73TdiyTXylPPfW7elVcyX1QAY1nN36",
        "SessionToken": "FwoGZXIvYXdzEFEaDLS/Z6lDiXqvnepnWSKzAZ5yEpcp6V4KG2aDYOaZ9Qv90bcp8RPC1+/K7a
9qSnQTkJE4/mung4FzbU5hWSjyN0zL62hLLQXgn50ZNAZaj8OZW4jJKI2hqL95Rmmn5zXw0Ig0g9hkzVkj1FJ32ikQ1eyRPZmGT
/twa0OllMmkgjkE614p9kEYfpOkF58iF4oAdoY99IcgJMshVb3MUh5IKEHqjHWLI3yO2xX0qqNJaGD0JFkdjf2RhJGN2Le8O6F9
jDmxKKG57/IFMi3e/ZgtI9xH5r7uCsZC1WWwqNgSxAijKPQYam+oP9l0bQiUmxqmZr5LLh/fHRY=",
        "Expiration": "2020-03-01T17:02:41Z"
    },
    "AssumedRoleUser": {
        "AssumedRoleId": "AROASXL6B65OVCX7E5PAX:AWS-assume-role",
        "Arn": "arn:aws:sts::187632318301:assumed-role/ec2-s3-inatance/AWS-assume-role"
    }
}
ubuntu@ip-172-31-71-45:~$
```

Add those credentials to this instance and get access to s3

```
ubuntu@ip-10-0-2-177:~$ aws s3 ls s3://ayush-s3/

An error occurred (AccessDenied) when calling the ListObjects operation: Access Denied
ubuntu@ip-10-0-2-177:~$ export AWS_ACCESS_KEY_ID=ASIASXL6B65OV7WHOIE6
ubuntu@ip-10-0-2-177:~$ export AWS_SECRET_ACCESS_KEY=N3cagQtTZr73TdiyTXylPPfW7elVcyX1QAY1nN36
ubuntu@ip-10-0-2-177:~$ export AWS_SESSION_TOKEN=FwoGZXIvYXdzEFEaDLS/Z6lDiXqvnepnWSKzAZ5yEpcp6V4KG2
aDYOaZ9Qv90bcp8RPC1+/K7a9qSnQTkJE4/mung4FzbU5hWSjyN0zL62hLLQXgn50ZNAZaj8OZW4jJKI2hqL95Rmmn5zXw0Ig0g
9hkzVkj1FJ32ikQ1eyRPZmGT/twa0OllMmkgjkE614p9kEYfpOkF58iF4oAdoY99IcgJMshVb3MUh5IKEHqjHWLI3yO2xX0qqNJ
aGD0JFkdjf2RhJGN2Le8O6F9jDmxKKG57/IFMi3e/ZgtI9xH5r7uCsZC1WWwqNgSxAijKPQYam+oP9l0bQiUmxqmZr5LLh/fHRY
=
ubuntu@ip-10-0-2-177:~$ aws sts get-caller-identity
{
    "UserId": "AROASXL6B65OVCX7E5PAX:AWS-assume-role",
    "Account": "187632318301",
    "Arn": "arn:aws:sts::187632318301:assumed-role/ec2-s3-inatance/AWS-assume-role"
}
ubuntu@ip-10-0-2-177:~$ aws s3 ls s3://ayush-s3/
2020-03-01 12:47:58       1089 Mongo-DB.txt
2020-03-01 12:47:22       1675 codecommit
2020-03-01 12:47:37       4606 sample.war
ubuntu@ip-10-0-2-177:~$
```

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;

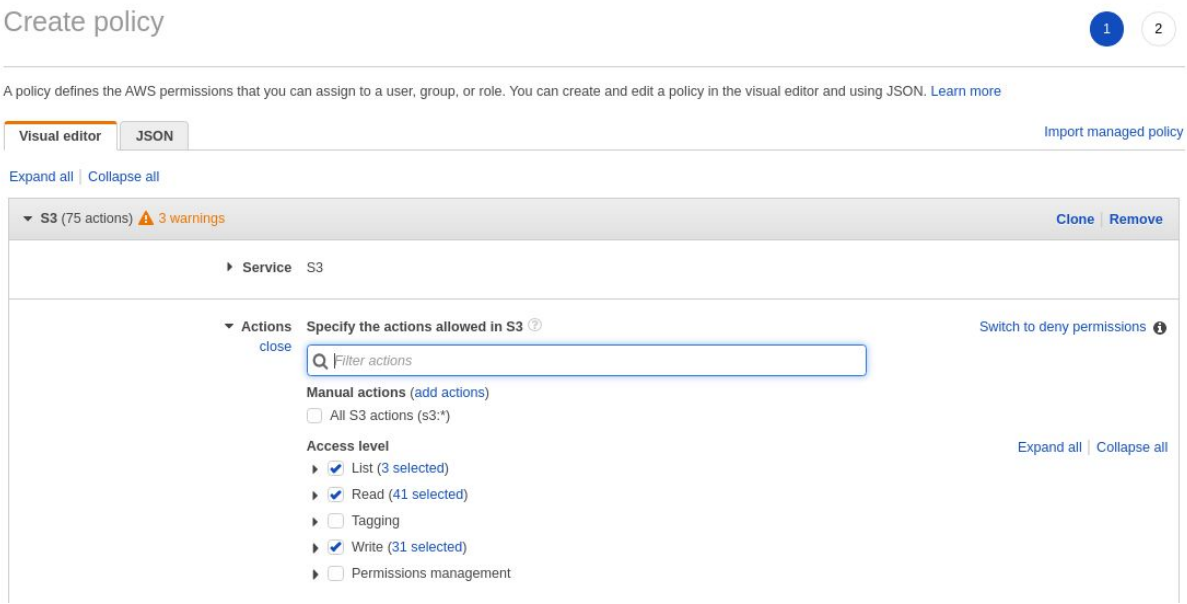
Action:

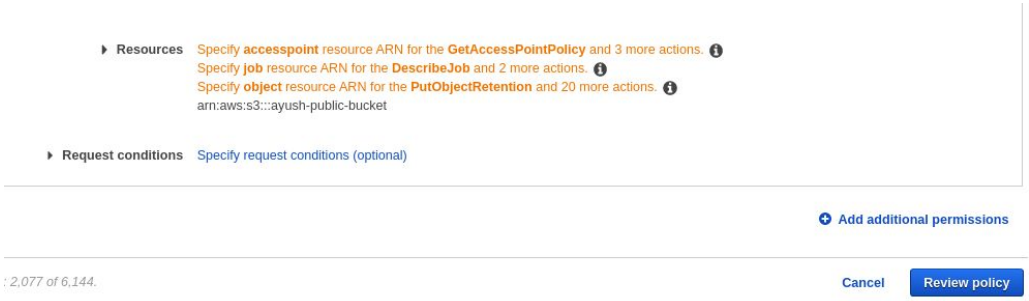
Get*,

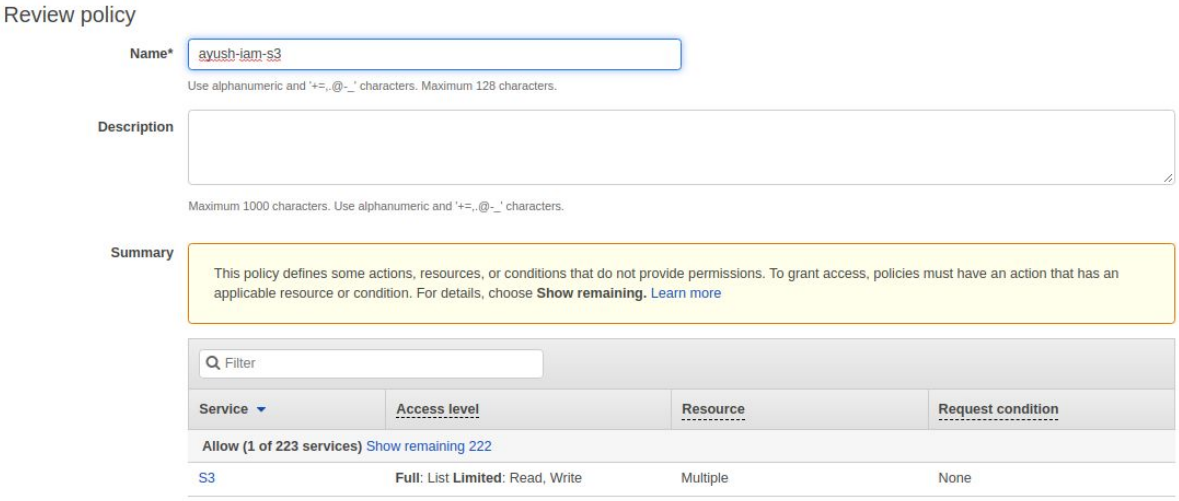
List*,

Put*,

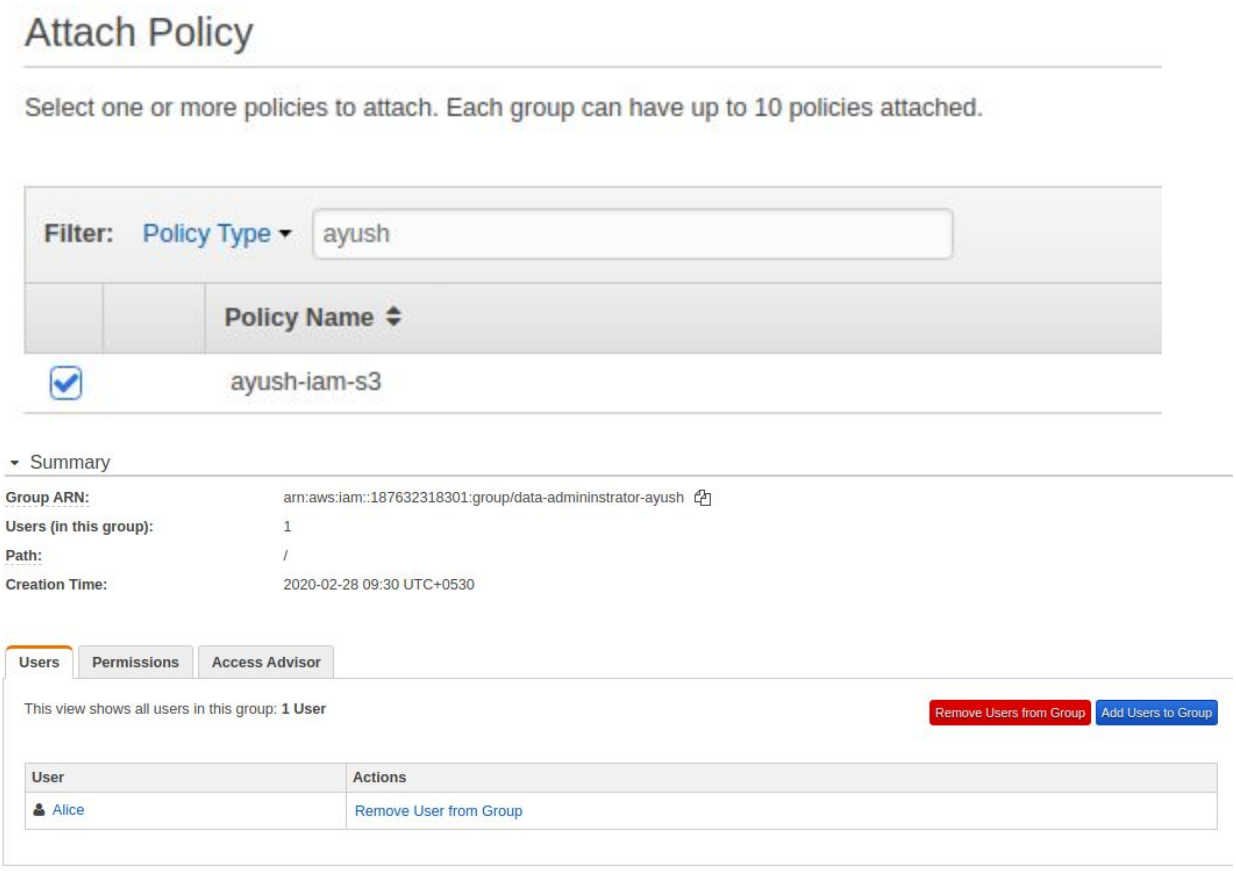
ARN: Input and output Buckets (no conditions)




Resources
Specify accesspoint resource ARN for the GetAccessPointPolicy and 3 more actions.
Specify job resource ARN for the DescribeJob and 2 more actions.
Specify object resource ARN for the PutObjectRetention and 20 more actions.
arn:aws:s3:::ayush-public-bucket
Request conditions
Specify request conditions (optional)
Add additional permissions
: 2,077 of 6,144.
Cancel
Review policy


Review policy
Name*
ayush-iam-s3
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
S3
Full: List Limited: Read, Write
Multiple
None


Set Group Name
Specify a group name. Group names can be edited any time.
Group Name:
data-admininstrator-ayush
Example: Developers or ProjectAlpha
Maximum 128 characters

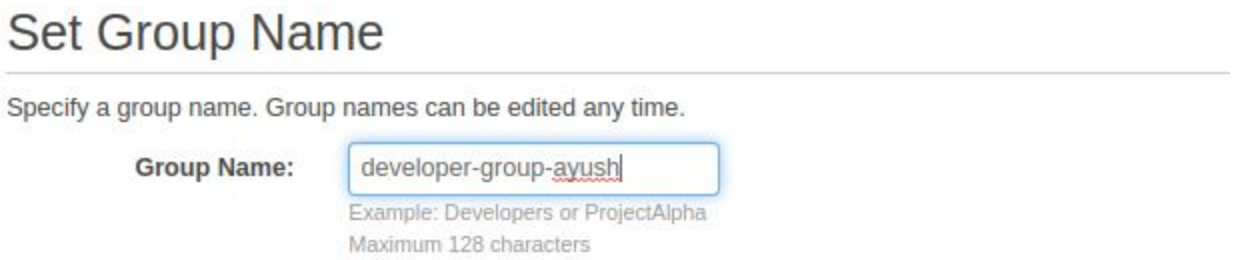


5. Create a group for the "Developer group " where the user 'bob ' is a member of this group. This group will Test Newly Developed Features for which they require access to EC2 instances. Provide the following access to this group:

Service: Amazon EC2

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

Condition: Dev Subnets only



▾ Summary

Group ARN:	arn:aws:iam::187632318301:group/developer-group-ayush
Users (in this group):	1
Path:	/
Creation Time:	2020-02-28 10:03 UTC+0530

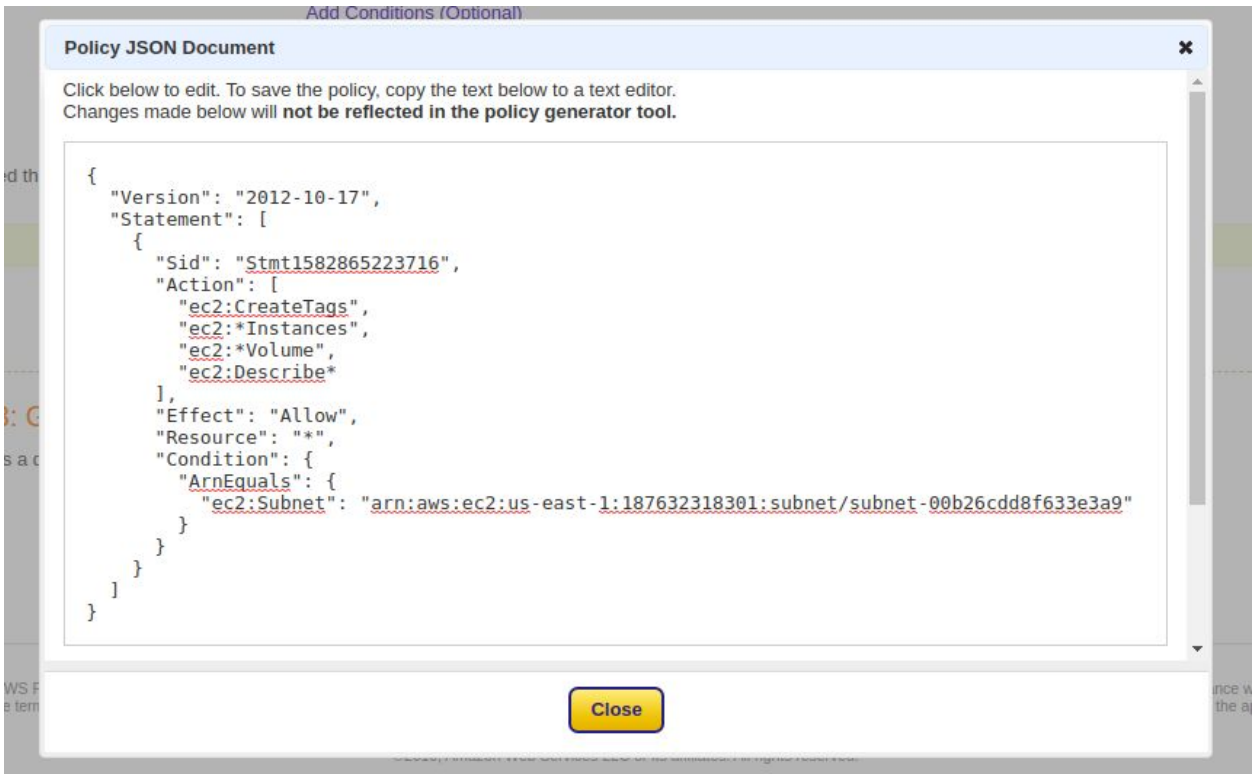
Users | Permissions | Access Advisor

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

Remove Users from Group | Add Users to Group

User	Actions
bob	Remove User from Group

\



Visual editor | JSON

Import managed policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1582865223716",
            "Action": [
                "ec2:CreateTags",
                "ec2:*Instances",
                "ec2:*Volume",
                "ec2:Describe*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "ec2:Subnet": "arn:aws:ec2:us-east-1:187632318301:subnet/subnet-00b26cdd8f633e3a9"
                }
            }
        }
```

Character count: 283 of 6,144.

Cancel | Review policy

Edit dev-policy

1 2

Review policy

Review this policy before you save your changes.

☑ Save as default

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			
EC2	**Limited**: Write, Tagging	All resources	ec2:Subnet = arn:aws:ec2:us-east-1:187632318301:subnet/subnet-00b26cdd8f633e3a9

*** Required**

Cancel Previous **Save changes**

Summary

Group ARN:	arn:aws:iam::187632318301:group/developer-group-ayush
Users (in this group):	0
Path:	/
Creation Time:	2020-02-28 10:03 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-policy	Show Policy | Detach Policy | Simulate Policy

Inline Policies

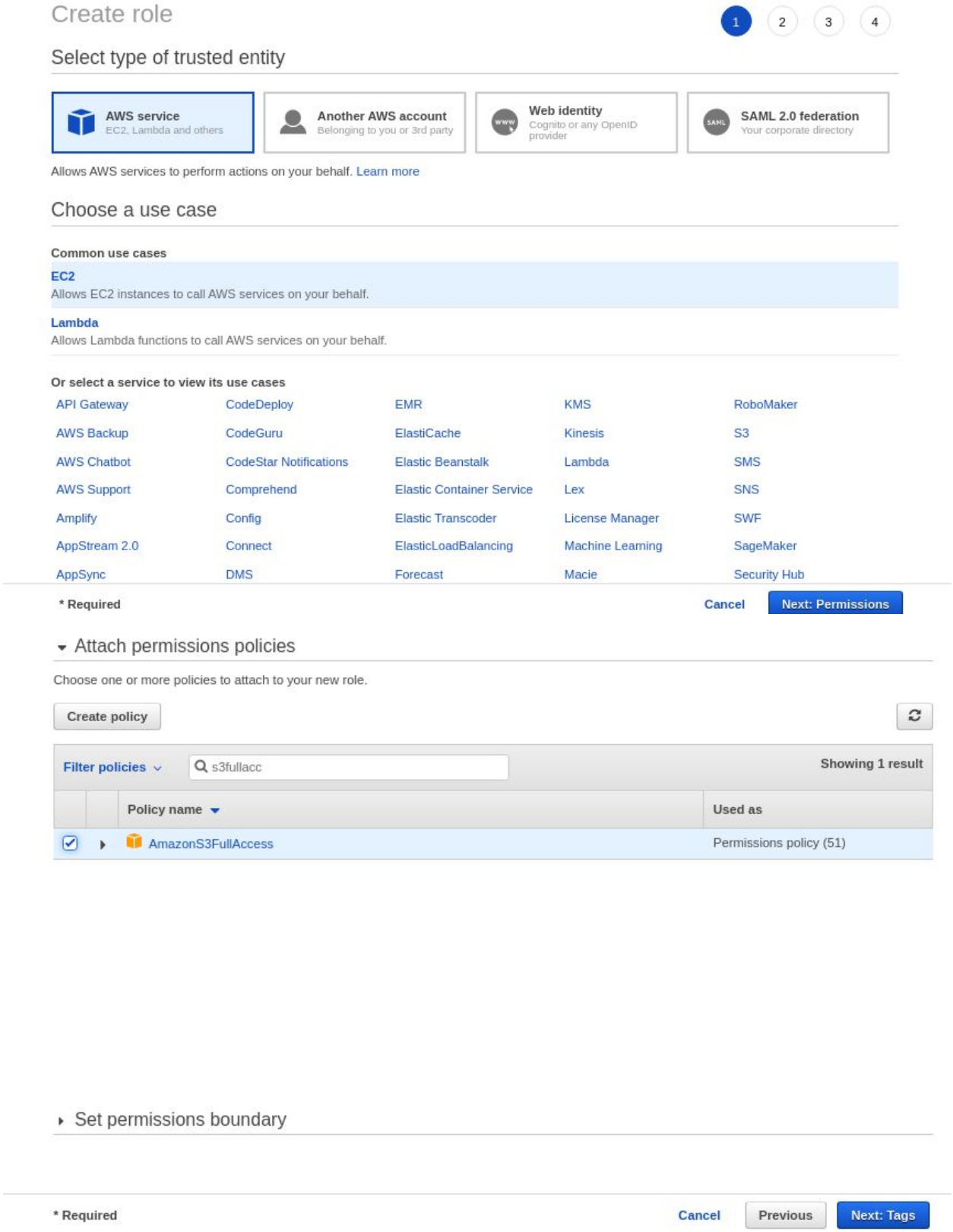
6. Identify the unused IAM users/credentials using AWS CLI.

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

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

```
ayush@ayush:~$ aws ec2 describe-instances --filters "Name=tag:backup,Values=true"
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-0ab51d3c5b27777ca",
                    "InstanceId": "i-080afe86e526d1662",
                    "InstanceType": "t2.micro",
                    "KeyName": "Srima-TTN-bootcamp",
                    "LaunchTime": "2020-02-27T11:56:39.000Z",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-1e",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-10-0-3-210.ec2.internal",
                    "PrivateIpAddress": "10.0.3.210",
                    "ProductCodes": [],
                    "PublicDnsName": "",
                    "PublicIpAddress": "100.26.218.55",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-008dcd90bf26a9055",
                    "VpcId": "vpc-00470a42fc196d84e",
```

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.



Create role

1 2 3 4

Review

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

Role name* ec2-s3-inatance

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

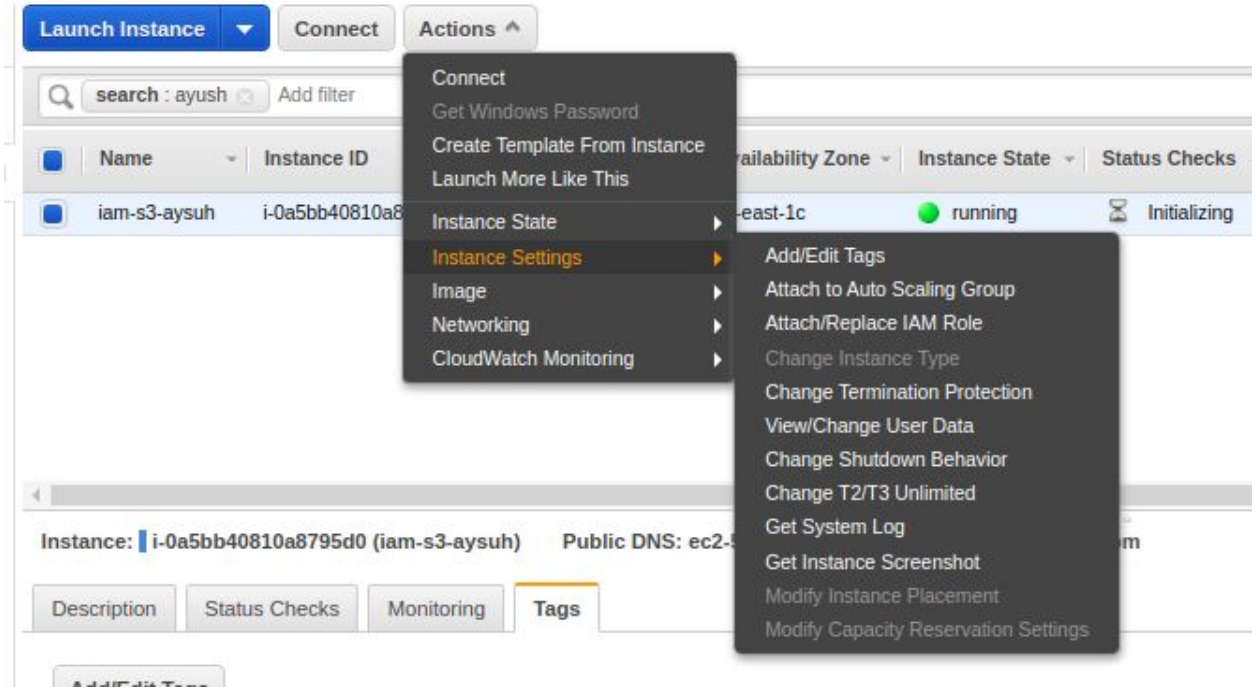
Permissions boundary Permissions boundary is not set

The new role will receive the following tags

Key	Value
NAme	s3-ec2-ayush
owner	ayush
purpose	assign

* Required

Cancel Previous Create 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 consol
If an IAM role is already attached to your instance, the IAM role you choose will replace the existing role.

Instance ID i-0a5bb40810a8795d0 (iam-s3-aysuh)

IAM role* ec2-s3-inatance Create new IAM role

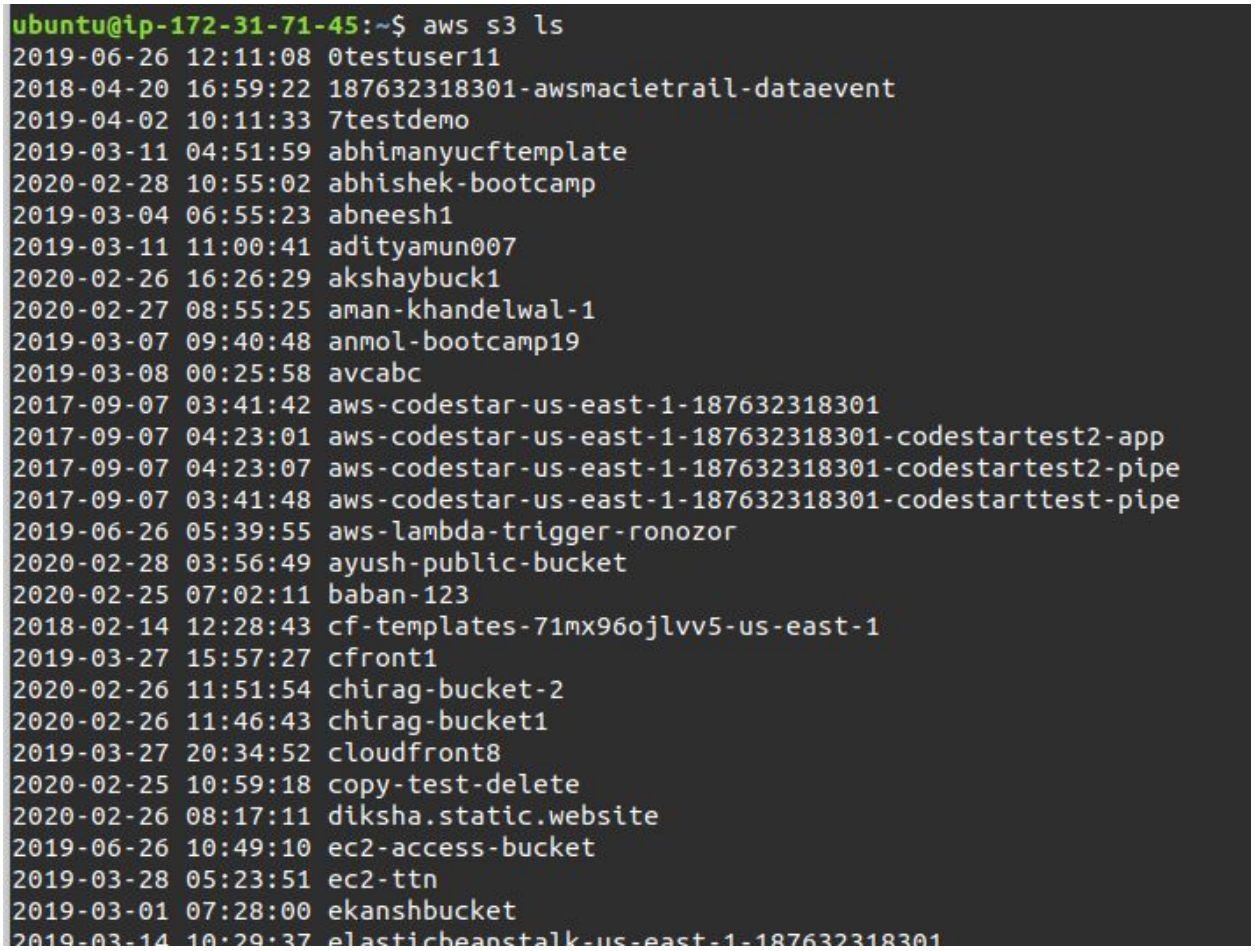
*** Required**

Attach/Replace IAM Role

IAM role operation succeeded

Close

```
ayush@ayush:~/Downloads$ ssh -i ayush-ec2.pem ubuntu@54.172.8.246
The authenticity of host '54.172.8.246 (54.172.8.246)' can't be established.
ECDSA key fingerprint is SHA256:mQOO+YrjA/b8jpzr9Ts2bFxlI2rdJ9hLlG9RV8UO5v4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.172.8.246' (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 disabled due to load higher than 1.0

 * Multipass 1.0 is out! Get Ubuntu VMs on demand on your Linux, Windows or
   Mac. Supports cloud-init for fast, local, cloud devops simulation.

     https://multipass.run/

 * Latest Kubernetes 1.18 beta is now available for your laptop, NUC, cloud
   VM Raspberry Pi, with automatic updates to the final GA release

     sudo snap install microk8s --channel=1.18/beta --classic

37 packages can be updated.
7 updates are security updates.


Last login: Thu Feb 27 10:11:01 2020 from 61.12.91.218
ubuntu@ip-172-31-71-45:~$
```



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.

We will add the below policy to all production server so that no development server can access it.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Server": "Production"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": [
        "ec2:CreateTags",
        "ec2:DeleteTags"
      ],
      "Resource": "*"
    }
  ]
}
```

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.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListUsers",
                "iam:GetAccountPasswordPolicy"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:*AccessKey*",
                "iam:ChangePassword",
                "iam:GetUser",
                "iam:*ServiceSpecificCredential*",
                "iam:*SigningCertificate*",
                "iam:ListMFADevices"
            ],
            "Resource": [
                "arn:aws:iam::*:user/${aws:username}",
                "arn:aws:iam::*:mfa/*"
            ]
        }
    ]
}
```