





1. Create a Role with full access to S3

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

[API Gateway](#) [CodeDeploy](#) [EMR](#) [KMS](#) [RoboMaker](#)
[AWS Backup](#) [CodeGuru](#) [ElastiCache](#) [Kinesis](#) [S3](#)


* Required

Cancel




Next: Permissions

▼ Attach permissions policies

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

Create policy 

Filter policies ▼ Showing 10 results

	Policy name ▼	Used as
<input type="checkbox"/>	 AmazonDMSRedshiftS3Role	None
<input checked="" type="checkbox"/>	 AmazonS3FullAccess	Permissions policy (25)
<input type="checkbox"/>	 AmazonS3ReadOnlyAccess	Permissions policy (1)
<input type="checkbox"/>	AWSLambdaS3ExecutionRole-1a3ddce4-a989-4828-88a4-7f5e701af3ef	Permissions policy (1)
<input type="checkbox"/>	AWSLambdaS3ExecutionRole-34f3178e-e3ee-4238-8c64-0e27432978a2	Permissions policy (1)
<input type="checkbox"/>	AWSLambdaS3ExecutionRole-9954a81d-a49b-408c-aa07-78234306319f	Permissions policy (1)
<input type="checkbox"/>	AWSLambdaS3ExecutionRole-c8b9c2ec-9d50-4969-8163-87e2da653db6	Permissions policy (1)

* Required

Cancel

Previous

Next: Tags

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

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.

```

ayush@ayush:~/github/ttnbootcamp-tothenew$ aws iam get-role --role-name ec2-s3-ayush
{
  "Role": {
    "Path": "/",
    "RoleName": "ec2-s3-ayush",
    "RoleId": "AROASXL6B65OWM5OP6DNW",
    "Arn": "arn:aws:iam::187632318301:role/ec2-s3-ayush",
    "CreateDate": "2020-02-27T03:36:57Z",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "ec2.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    },
    "Description": "Allows EC2 instances to call AWS services on your behalf",
    "MaxSessionDuration": 3600,
    "Tags": [
      {
        "Key": "Name",
        "Value": "ec2-s3-ayush"
      },
      {
        "Key": "owner",
        "Value": "ayush"
      }
    ]
  }
}

```

```

{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": { "AWS": "arn:aws:iam::187632318301:role/ec2-s3-ayush" },
    "Action": "sts:AssumeRole"
  }
}

```

```

ayush@ayush:~/github/ttnbootcamp-tothenew$ aws iam create-role --role-name ec2-s3-trust --assume-role-policy-document file://trust-policy.json
{
  "Role": {
    "Path": "/",
    "RoleName": "ec2-s3-trust",
    "RoleId": "AROASXL6B6503L63YZZFG",
    "Arn": "arn:aws:iam::187632318301:role/ec2-s3-trust",
    "CreateDate": "2020-02-27T04:03:38Z",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": {
        "Effect": "Allow",
        "Principal": {
          "AWS": "arn:aws:iam::187632318301:role/ec2-s3-ayush"
        },
        "Action": "sts:AssumeRole"
      }
    }
  }
}

```

```

ayush@ayush:~/github/ttnbootcamp-tothenew$ aws iam attach-role-policy --role-name ec2-s3-trust --policy-arn "arn:aws:iam::aws:policy/AmazonS3FullAccess"
ayush@ayush:~/github/ttnbootcamp-tothenew$ aws iam list-attached-role-policies --role-name example-role

An error occurred (NoSuchEntity) when calling the ListAttachedRolePolicies operation: The role with name example-role cannot be found.
ayush@ayush:~/github/ttnbootcamp-tothenew$ aws iam list-attached-role-policies --role-name ec2-s3-trust
{
  "AttachedPolicies": [
    {
      "PolicyName": "AmazonS3FullAccess",
      "PolicyArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
    }
  ]
}
ayush@ayush:~/github/ttnbootcamp-tothenew$

```

Reference: <https://aws.amazon.com/premiumsupport/knowledge-center/iam-assume-role-cli/>

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

search : 09852355b244b0c22

✕

Add filter

Name

▼

Instance ID

▼

Instance Type

▼

Availability Zone

▼

Instance State

▼

Status Checks

▲

Alarm Status

iam-ayush

i-09852355b244b0c22

t2.micro

us-east-1c

running

Initializing

None

```
ayush@ayush:~/github/ttnbootcamp-tothenew$ aws iam create-instance-profile --instance-profile-name ec2-s3-trust-Instance-Profile
{
  "InstanceProfile": {
    "Path": "/",
    "InstanceProfileName": "ec2-s3-trust-Instance-Profile",
    "InstanceProfileId": "AIPASXL6B6503H5LFXOXD",
    "Arn": "arn:aws:iam::187632318301:instance-profile/ec2-s3-trust-Instance-Profile",
    "CreateDate": "2020-02-27T04:26:56Z",
    "Roles": []
  }
}
```

```
ubuntu@ip-172-31-71-45:~$ aws s3 ls
```

```
An error occurred (AccessDenied) when calling the ListBuckets operation: Access Denied
```

```
ubuntu@ip-172-31-71-45:~$
```



```

ayush@ayush:~/github/ttnbootcamp-tothenew$ aws iam add-role-to-instance-profile
--role-name ec2-s3-trust --instance-profile-name ec2-s3-trust-Instance-Profile
ayush@ayush:~/github/ttnbootcamp-tothenew$ aws ec2 associate-iam-instance-profil
e --instance-id i-09852355b244b0c22 --iam-instance-profile Name=ec2-s3-trust-Inst
ance-Profile
{
  "IamInstanceProfileAssociation": {
    "AssociationId": "iip-assoc-0c49a66c4185c1647",
    "InstanceId": "i-09852355b244b0c22",
    "IamInstanceProfile": {
      "Arn": "arn:aws:iam::187632318301:instance-profile/ec2-s3-trust-Inst
ance-Profile",
      "Id": "AIPASXL6B6503H5LFX0XD"
    },
    "State": "associating"
  }
}
ayush@ayush:~/github/ttnbootcamp-tothenew$ aws ec2 describe-iam-instance-profile
-associations
{
  "IamInstanceProfileAssociations": [
    {
      "AssociationId": "iip-assoc-0c49a66c4185c1647",
      "InstanceId": "i-09852355b244b0c22",
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::187632318301:instance-profile/ec2-s3-trust-
Instance-Profile",
        "Id": "AIPASXL6B6503H5LFX0XD"
      },
      "State": "associating"
    }
  ]
}
ayush@ayush:~/github/ttnbootcamp-tothenew$ █

```

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)

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](#)

▼ S3 (75 actions) ⚠ 3 warnings [Clone](#) [Remove](#)

► Service S3

▼ Actions close

Specify the actions allowed in S3 ⓘ [Switch to deny permissions ⓘ](#)

Manual actions (add actions)

☐ All S3 actions (s3:*)

Access level

► ☒ List (3 selected)

► ☒ Read (41 selected)

► ☐ Tagging

► ☒ Write (31 selected)

► ☐ Permissions management

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

► 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*

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) 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:

Example: Developers or ProjectAlpha

Maximum 128 characters

Attach Policy

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

Filter: Policy Type

	Policy Name
<input checked="" type="checkbox"/>	ayush-iam-s3

Summary

Group ARN: arn:aws:iam::187632318301:group/data-administrator-ayush

Users (in this group): 1

Path: /

Creation Time: 2020-02-28 09:30 UTC+0530

Users

This view shows all users in this group: 1 User

[Remove Users from Group](#) [Add Users to Group](#)

User	Actions
Alice	Remove User from Group

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

Set Group Name

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

Group Name:

Example: Developers or ProjectAlpha
Maximum 128 characters

▼ 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

\

Add Conditions (Optional)

Policy JSON Document

Click below to edit. To save the policy, copy the text below to a text editor. Changes made below will **not be reflected in the policy generator tool**.

```
{
  "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"
        }
      }
    }
  ]
}
```

Close

Visual editor JSON

[Import managed policy](#)

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "Stmt1582865223716",
6       "Action": [
7         "ec2:CreateTags",
8         "ec2:*Instances",
9         "ec2:*Volume",
10        "ec2:Describe*"
11      ],
12      "Effect": "Allow",
13      "Resource": "*",
14      "Condition": {
15        "ArnEquals": {
16          "ec2:Subnet": "arn:aws:ec2:us-east-1:187632318301:subnet/subnet-00b26cdd8f633e3a9"
17        }
18      }
19    }
20  ]
21 }
```

Character count: 283 of 6,144.

[Cancel](#) [Review policy](#)

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==null) | .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",
          "Architecture": "x86_64"
        }
      ]
    }
  ]
}
```

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

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

API Gateway	CodeDeploy	EMR	KMS	RoboMaker
AWS Backup	CodeGuru	ElastiCache	Kinesis	S3
AWS Chatbot	CodeStar Notifications	Elastic Beanstalk	Lambda	SMS
AWS Support	Comprehend	Elastic Container Service	Lex	SNS
Amplify	Config	Elastic Transcoder	License Manager	SWF
AppStream 2.0	Connect	ElasticLoadBalancing	Machine Learning	SageMaker
AppSync	DMS	Forecast	Macie	Security Hub


* Required

Cancel

Next: Permissions

▼ Attach permissions policies


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

Create policy 

Filter policies ▼

s3fullacc

Showing 1 result

	Policy name ▼	Used as
<input checked="" type="checkbox"/>	 AmazonS3FullAccess	Permissions policy (51)

► Set permissions boundary

* Required

Cancel

Previous

Next: 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  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

Launch Instance 

Connect

Actions 

search : ayush  Add filter

<input checked="" type="checkbox"/>	Name	Instance ID	Availability Zone	Instance State	Status Checks
<input checked="" type="checkbox"/>	iam-s3-aysuh	i-0a5bb40810a8795d0	us-east-1c	 running	 Initializing

Instance: **i-0a5bb40810a8795d0 (iam-s3-aysuh)** Public DNS: ec2-54-93-24-100.us-east-1.compute.amazonaws.com

Description

Status Checks

Monitoring

Tags

Add/Edit Tags

Connect

Get Windows Password

Create Template From Instance

Launch More Like This

Instance State

Instance Settings

Image

Networking

CloudWatch Monitoring

Add/Edit Tags

Attach to Auto Scaling Group

Attach/Replace IAM Role

Change Instance Type

Change Termination Protection

View/Change User Data

Change Shutdown Behavior

Change T2/T3 Unlimited

Get System Log

Get Instance Screenshot

Modify Instance Placement

Modify Capacity Reservation Settings

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-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:mQ00+YrjA/b8jpzr9Ts2bFxlI2rdJ9hLLG9RV8U05v4.
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:~$
```

```

ubuntu@ip-172-31-71-45:~$ 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
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
2019-03-14 10:29:37 elasticbeanstalk-us-east-1-187632318301

```

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/*"
      ]
    }
  ]
}
```