

Home / AWS / Guided Lab / Creating IAM Policies

# Creating IAM Policies

Level: Fundamental

Identity And Access Management    Amazon Web Services



0h 28m 45s left



End Lab

Open Console

Validation

## Lab Credentials

User Name ⓘ

Whiz\_User\_80425.36691910



Password ⓘ

66417b62-1221-4842-9ec5-b8d9a3fe686c



Access Key ⓘ

AKIAWPK3VFXHM2HFFU6B



Secret Key ⓘ

D0Rz+6GoJxCYUjAQOj9C3sC9u64xb6ce5L52v0ry






## Lab Resources

No Lab Resources Found

## Support Documents

1. [FAQs and Troubleshooting](#)

Need help?

-  How to use Hands on Lab
-  Troubleshooting Lab
-  FAQs



[Submit Feedback](#)

[Share](#)

Lab Overview

Lab Steps

Lab Validation

-  Cloud Administrator
-  Security

# Lab Steps

## Task 1: Sign in to AWS Management Console

1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
  - Leave the Account ID as default. Never edit/remove the 12-digit Account ID present in the AWS Console. Otherwise, you cannot proceed with the lab.



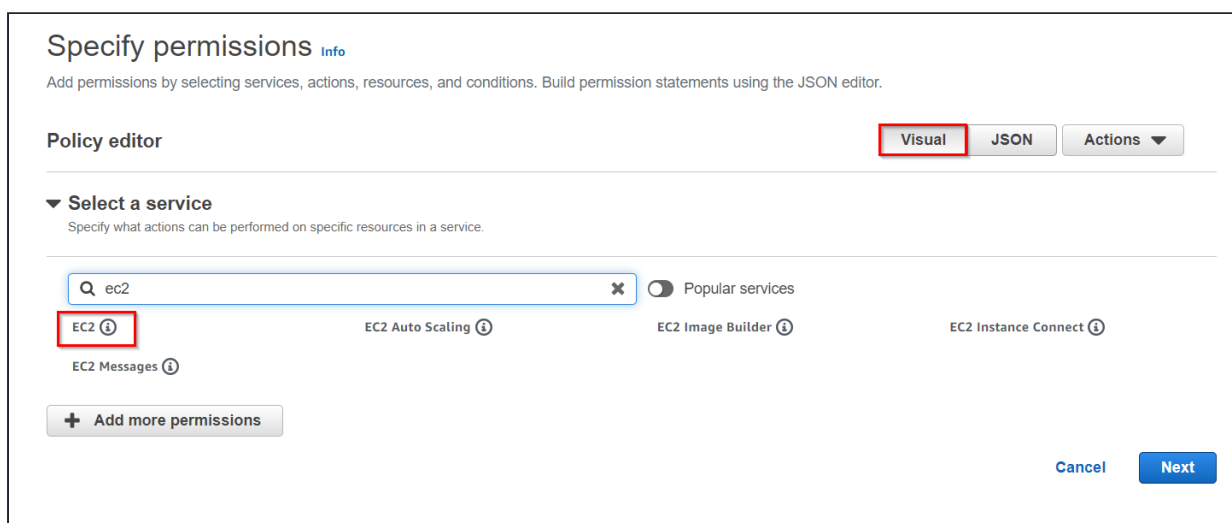
East (N. Virginia) us-east-1.

**Note :** If you face any issues, please go through [FAQs and Troubleshooting for Labs](#).

## Task 2: Creating an IAM Policy for EC2

In this task, we are going to create an IAM policy specifically for the EC2 (Elastic Compute Cloud) service. EC2 is a core AWS service that provides virtual servers in the cloud. By creating an IAM policy for EC2, users can define the permissions and actions that are allowed or restricted for EC2 resources.

1. Navigate to the **Services** menu at the top, then click on **IAM** in the **Security, identity, & Compliance** section.
2. In the left menu, select **Policies**.
3. Click on **Create Policy** button.
4. Under **Visual**, Type **EC2** in the search box and select **EC2**.



Specify permissions [Info](#)

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor Visual JSON Actions ▼

▼ Select a service

Specify what actions can be performed on specific resources in a service.

Q ec2 × Popular services

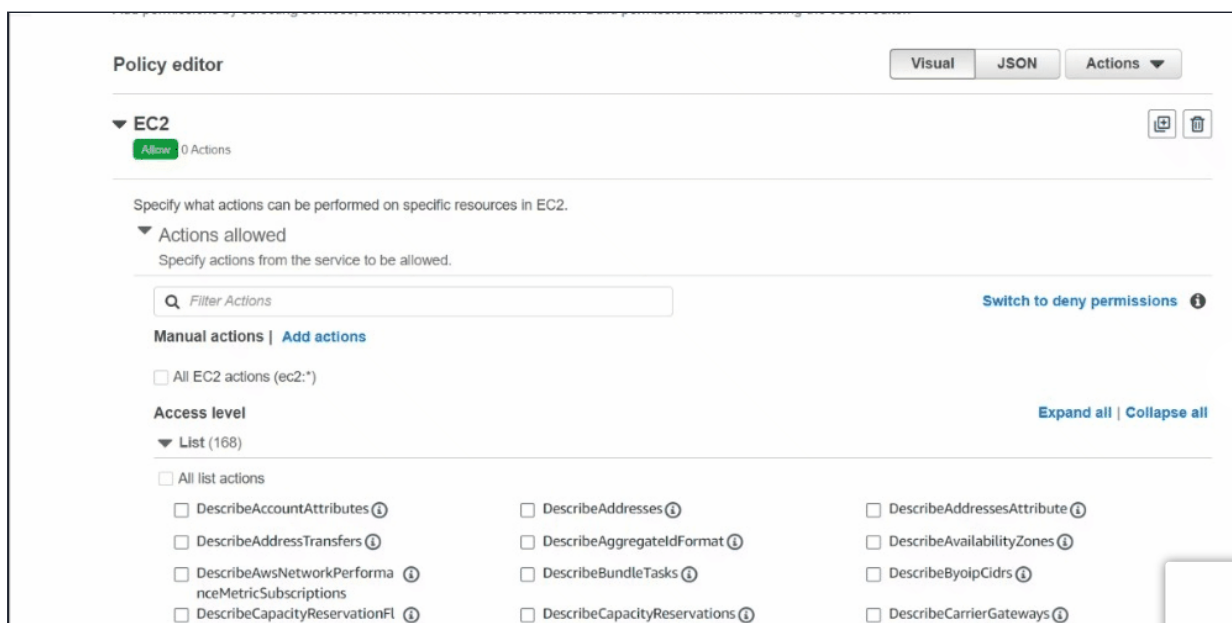
EC2 EC2 Auto Scaling EC2 Image Builder EC2 Instance Connect

EC2 Messages

+ Add more permissions

Cancel Next

5. In the **Actions**, specify the actions allowed in EC2. For this service, We'll choose **List**.
6. Click on **Resources**, scroll down and choose **All resources** so that there is no need to specify the resource ARN.



Policy editor Visual JSON Actions ▼

▼ EC2 Allow 0 Actions

Specify what actions can be performed on specific resources in EC2.

▼ Actions allowed

Specify actions from the service to be allowed.

Q Filter Actions Switch to deny permissions

Manual actions | [Add actions](#)

☐ All EC2 actions (ec2:\*)

Access level Expand all | Collapse all

▼ List (168)

☐ All list actions

☐ DescribeAccountAttributes ☐ DescribeAddresses ☐ DescribeAddressesAttribute

☐ DescribeAddressTransfers ☐ DescribeAggregateIdFormat ☐ DescribeAvailabilityZones

☐ DescribeAwsNetworkPerformanceMetricSubscriptions ☐ DescribeBundleTasks ☐ DescribeByoipCidrs

☐ DescribeCapacityReservations ☐ DescribeCarrierGateways

7. Now scroll up and If you click on the JSON, you can see the policy we created.



8. Click on **Next**

9. Review:

- Name : Enter **EC2Policy**
- Description : Enter **EC2 Full Read and List access**
- You can see the access level.
- Review the policy and then click on **Create policy**.

**Policy details**

**Policy name**  
Enter a meaningful name to identify this policy.  
  
Maximum 128 characters. Use alphanumeric and '+', '@', '-' characters.

**Description - optional**  
Add a short explanation for this policy.  
  
Maximum 1,000 characters. Use alphanumeric and '+', '@', '-' characters.

**Permissions defined in this policy** [Info](#) Edit

Permissions in the policy document specify which actions are allowed or denied.

**Allow (1 of 379 services)** Show remaining 378 services

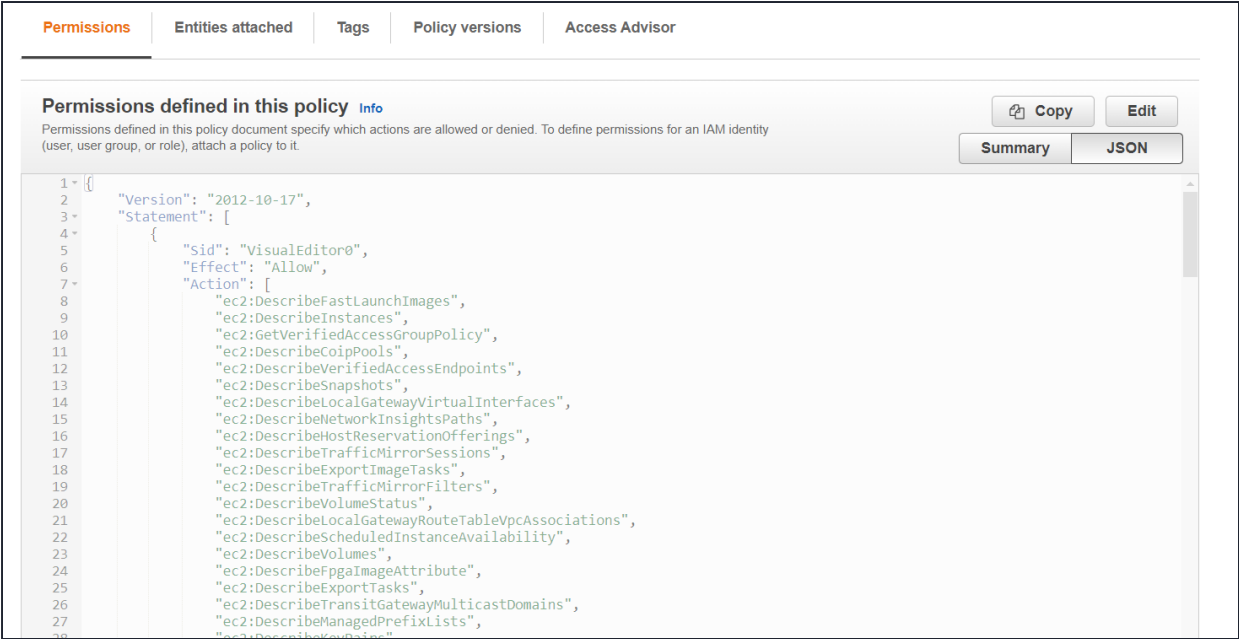
Service	Access level	Resource	Request condition
EC2	Full: List	All resources	None

10. After creating, you will get a verification for the created Policy.



11. In the filter policies, type your policy name and click on it.

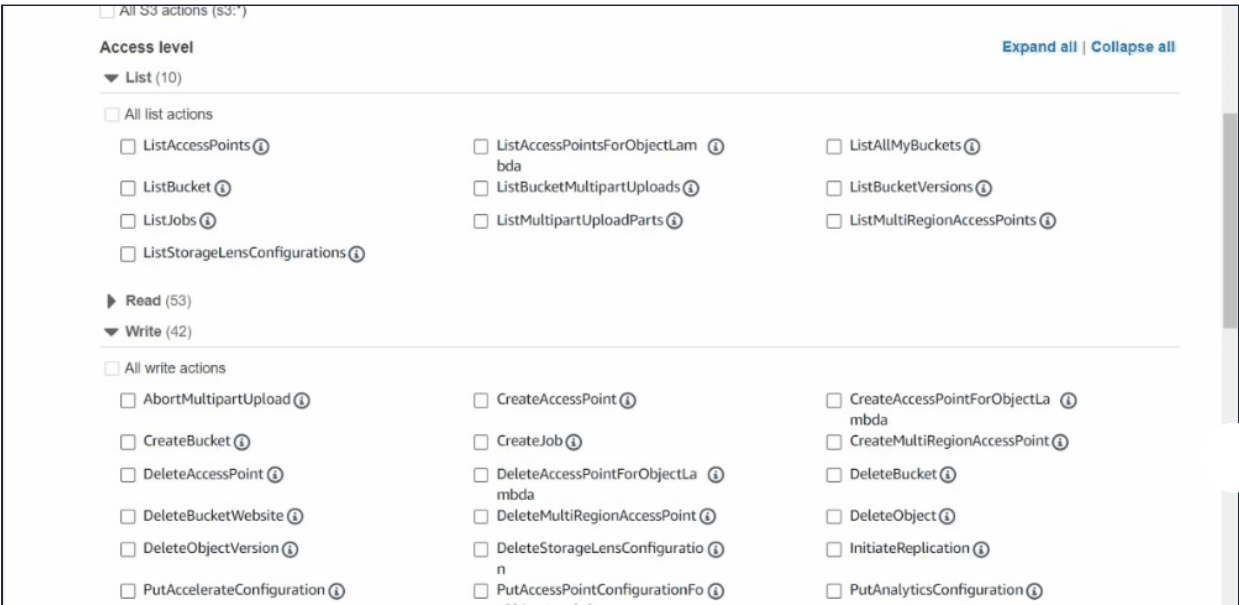
12. In the Summary, (under the JSON) you can see the policy you created.



### Task 3: Creating an IAM Policy for S3

In this task, we are going to create an IAM policy for the S3 (Simple Storage Service) service. S3 is a scalable storage service provided by AWS.

1. Click on **Create Policy** button again.
2. Under **Visual Editor**, type S3 in the search box and select **S3**.
3. In the **Actions**, specify the actions allowed in S3. For this service, we'll choose **List**, **Tagging** and **Write**.
4. Click on **Resources** and choose **All resources** so that there is no need to specify the resource ARN.



5. If you click on the JSON, you can see the policy we created.
6. Click on **Next** button.

## 7. Review:

- Name : Enter **S3Policy**
- For **Policy description**, type a description for the new policy.
- In the Summary, you can see the Access level.
- Review the policy and then click on **Create Policy**.

8. After creating, you will get a verification for the created policy

9. In the filter policies, type your policy name and click on it.

10. In the Summary, (under the JSON) you can see the policy you created.

The screenshot shows the AWS IAM console interface for a policy named **S3Policy**. The **Permissions** tab is selected, displaying the JSON policy document. The document defines permissions for the **VisualEditor0** SID, allowing various S3 actions. The JSON is as follows:

```

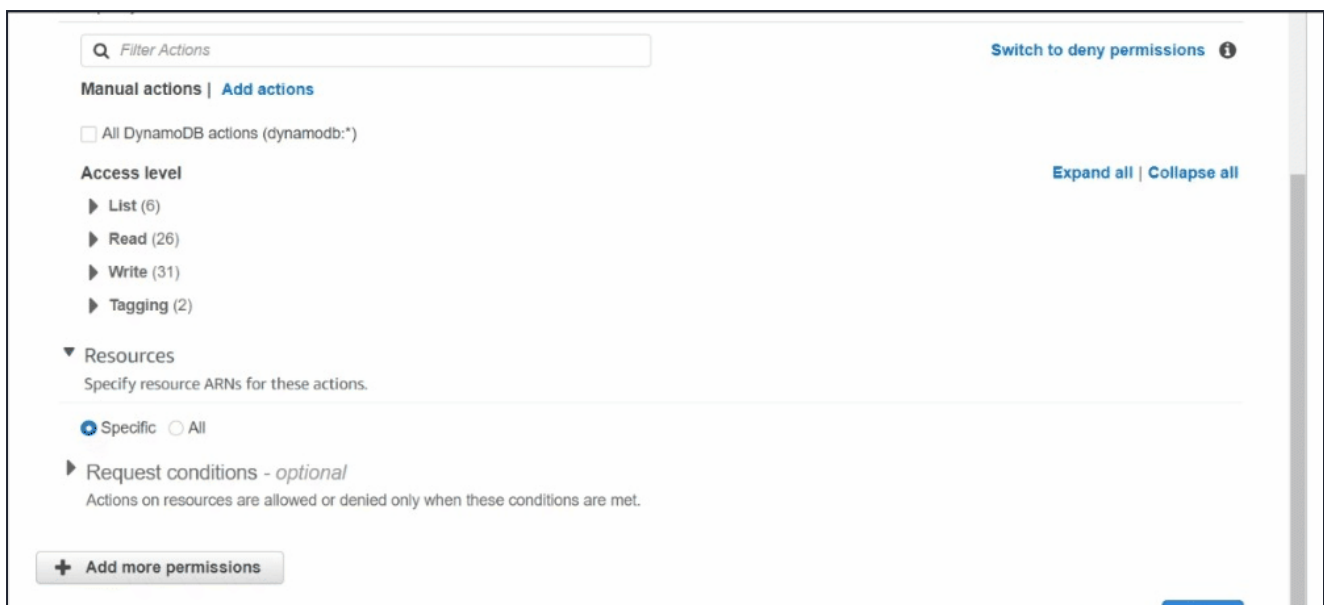
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "VisualEditor0",
6       "Effect": "Allow",
7       "Action": [
8         "s3:ListAccessPointsForObjectLambda",
9         "s3:PutAnalyticsConfiguration",
10        "s3:PutAccessPointConfigurationForObjectLambda",
11        "s3:PutStorageLensConfiguration",
12        "s3>DeleteAccessPoint",
13        "s3>CreateBucket",
14        "s3>DeleteAccessPointForObjectLambda",
15        "s3:ReplicateObject",
16        "s3>DeleteBucketWebsite",
17        "s3>DeleteJobTagging",
18        "s3:PutLifecycleConfiguration",
19        "s3:PutObjectTagging",
20        "s3>DeleteObject",
21        "s3>CreateMultiRegionAccessPoint",
22        "s3>DeleteObjectTagging",
23        "s3>DeleteStorageLensConfigurationTagging",
24        "s3:ListJobs",
25        "s3:PutReplicationConfiguration",
26        "s3>DeleteObjectVersionTagging",

```

## Task 4: Creating an IAM Policy for DynamoDB

In this task, we are going to create an IAM policy for the DynamoDB service. DynamoDB is a fully managed NoSQL database service offered by AWS.

1. Click on **Create Policy** button again.
2. Under **Visual**, type **DynamoDB** in the search box and select **DynamoDB**.
3. In the **Actions**, specify the actions allowed in DynamoDB. For this service, we'll choose **All DynamoDB actions**.
4. Click on **Resources** and choose **All resources** so that there is no need to specify the resource ARN.



The screenshot shows the 'Add permissions' interface in the AWS IAM console. At the top, there is a search bar labeled 'Filter Actions' and a link 'Switch to deny permissions' with an information icon. Below this, there are tabs for 'Manual actions' and 'Add actions'. A checkbox 'All DynamoDB actions (dynamodb:\*)' is present. The 'Access level' section lists 'List (6)', 'Read (26)', 'Write (31)', and 'Tagging (2)', each with a right-pointing arrow. To the right of this list are links 'Expand all' and 'Collapse all'. The 'Resources' section is expanded, showing 'Specify resource ARNs for these actions.' with radio buttons for 'Specific' (selected) and 'All'. Below this is a section for 'Request conditions - optional' with a right-pointing arrow and a note: 'Actions on resources are allowed or denied only when these conditions are met.' At the bottom left is a button '+ Add more permissions'.

5. If you click on the **JSON** you can see the policy we created.

6. Click on **Next** button.

7. Review:

- Name : Enter **DynamoDBPolicy**
- For **Policy description**, type a description for the new policy.
- In the Summary, you can see the Access level.
- Review the policy and then click on **Create policy**.

8. After creating, you will get a verification for the created policy

9. In the filter policies, type your policy name and click on it.

10. In the Summary, (under the JSON) you can see the policy you created.

## Do You know?

IAM policies are a powerful tool for managing access to AWS resources, and their default deny rule and support for conditions contribute to the robust security and access control capabilities of the AWS IAM system.

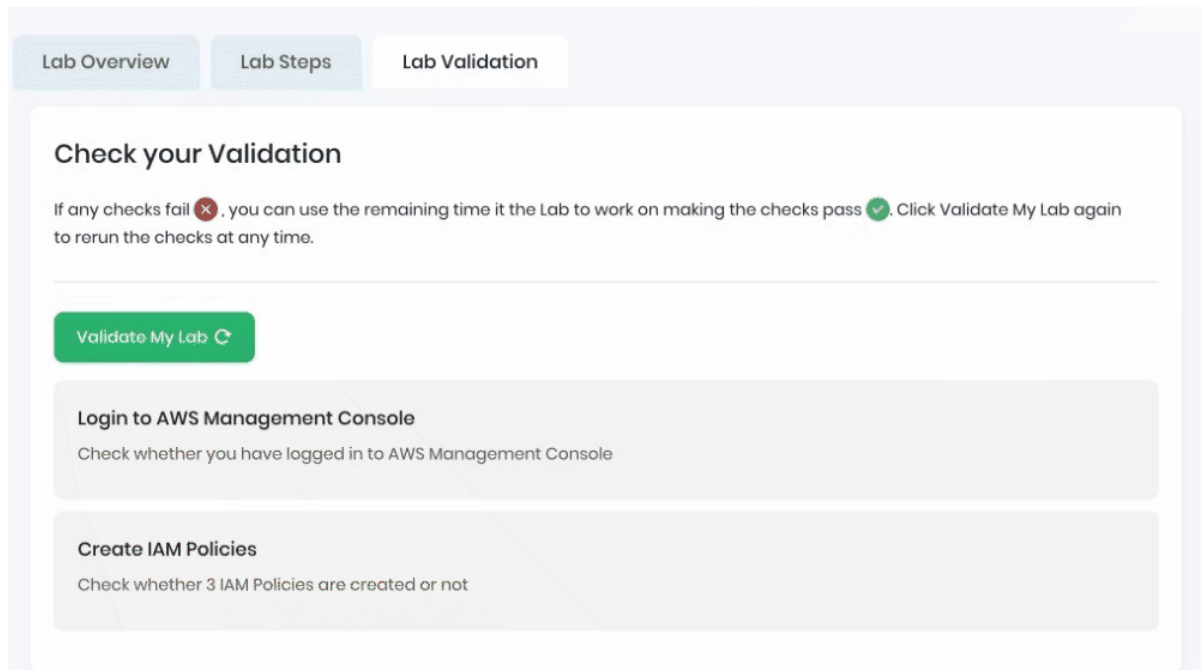
### Task 5: Validation Test

1. Once the lab steps are completed, please click on the **Validation** button on the left side panel.



2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.

3. Sample output :



## Completion and Conclusion

1. You have successfully created an IAM Policy for EC2 Service.
2. You have successfully created an IAM Policy for S3 Service.
3. You have successfully created an IAM Policy for DynamoDB service.

## End Lab

1. Sign out of the AWS Account.
2. You have successfully completed the lab.
3. Once you have completed the steps, click on **End lab** from your whizlabs dashboard.

[About Us](#) [Subscription](#) [Instructions and Guidelines](#) [FAQ's](#) [Contact Us](#)



© 2024, Whizlabs Software Pvt. Ltd.

