



IAM Permission Boundary

An IAM (Identity and Access Management) permission boundary is a feature in AWS (Amazon Web Services) that defines the maximum permissions an IAM entity (such as a user or role) can have. It acts as a policy that sets the maximum permissions that the entity can be granted regardless of any other policies attached to it.

By setting a permission boundary, you can control the maximum level of access an IAM entity can have, even if policies are attached that would otherwise grant broader permissions. This helps enforce security and compliance requirements by ensuring that certain IAM entities cannot exceed a predefined level of access.



Use cases of IAM Permission Boundary:

IAM permission boundaries have several use cases:

1. **Security Control:** Permission boundaries help enforce the principle of least privilege by restricting the maximum permissions that IAM entities can have. This reduces the risk of unauthorized access and potential security breaches.
2. **Compliance Requirements:** Many regulatory standards require organizations to implement strict access controls and enforce separation of duties. IAM permission boundaries provide a mechanism to meet these compliance requirements by limiting the scope of access for IAM entities.
3. **Complex Access Requirements:** In environments with complex access requirements, where users or roles may have multiple policies attached, permission boundaries offer a way to set clear limits on access levels, ensuring that entities cannot exceed certain permissions regardless of other policies.
4. **Third-party Access:** When granting access to third-party entities, such as vendors or partners, IAM permission boundaries can be used to restrict their access to only the necessary resources and actions, reducing the risk of potential misuse or unauthorized access.
5. **Resource Isolation:** Permission boundaries can be used to isolate resources within an organization. By defining strict boundaries for different teams or departments, organizations can ensure that each group only has access to the resources required for their specific tasks, enhancing resource isolation and security.

Overall, IAM permission boundaries are a powerful tool for managing access control in AWS environments, providing granular control over permissions and helping organizations maintain security and compliance standards.



What are we doing in this Lab?

In IAM permission boundaries, we're defining the maximum permissions that an IAM entity (like a user or role) can have within AWS services. The end goal is to enhance security and compliance by limiting the scope of access for these entities, ensuring they cannot surpass predefined boundaries, even if other policies would grant broader permissions. It helps

organizations maintain control over access levels and reduce the risk of unauthorized actions within their AWS environments.

😊 To begin with the Lab:

1. So, login to AWS console and navigate to EC2, there you need to launch an instance.
2. Once the instance is launched now you have to go to IAM and there you are going to create an IAM user.
3. Now attach EC2 read only permission to this IAM user.

The screenshot shows the AWS IAM Permissions page. At the top, there are tabs for Permissions, Groups, Tags (1), Security credentials, and Access Advisor. The Permissions tab is selected. Below the tabs, it says "Permissions policies (1)". A note states "Permissions are defined by policies attached to the user directly or through groups." There is a search bar and a filter dropdown set to "All types". A table lists one policy:

Policy name	Type	Attached via
AmazonEC2ReadOnlyAccess	AWS managed	Directly

4. Now you are going to login with this IAM user in another browser. Then navigate to EC2. Here you can see that the permission is for read only so, you can only see the instance.

The screenshot shows the AWS EC2 Dashboard. On the left, a sidebar menu includes EC2 Dashboard, Instances, Images, Elastic Block Store, Network & Security, CloudShell, and Feedback. The Instances section is expanded, showing sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and AMIs. The main content area displays the "Resources" section with metrics for Instances (running: 1), Auto Scaling Groups (0), Dedicated Hosts (0), Elastic IPs (0), Instances (1), Key pairs (2), Load balancers (0), Placement groups (0), Security groups (11), Snapshots (1), and Volumes (1). Below this is the "Launch instance" section with a "Launch instance" button and a "Migrate a server" button. A note says "Note: Your instances will launch in the Asia Pacific (Mumbai) Region". The "Service health" section shows an error message: "An error occurred: An error occurred retrieving service health information". The "Zones" section lists Zone name and Zone ID. The "Account attributes" section shows a "View all AWS Free Tier offers" button. On the right, there is an "EC2 Free Tier Info" section with a note about the end of month forecast and a detailed error message for a user who is not authorized to perform certain actions due to missing identity-based policy.

5. Here you can see this instance. But if you will try to stop this instance or terminate this instance then you will get an error.

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under 'Instances', there is a section for 'New'. The main content area displays a table titled 'Instances (1/1) info' with one row. The row details are:

- Name:** demo-instance
- Instance ID:** i-07b848600623fd9fe
- Instance state:** Running
- Instance type:** t2.micro
- Status check:** 2/2 checks passed
- Alarm status:** View alarms
- Availability Zone:** ap-south-1a
- Public IP:** ec2-13-2

Below the table, a detailed view for 'Instance: i-07b848600623fd9fe (demo-instance)' is shown. The 'Details' tab is selected. Key details include:

- Instance summary:**
 - Instance ID:** i-07b848600623fd9fe (demo-instance)
 - IPV6 address:** -
 - Hostname type:** IP name: ip-172-31-36-171.ap-south-1.compute.internal
 - Answer private resource DNS name:** IPV4 (A)
 - Auto-assigned IP address:** -
- Networking:**
 - Public IPv4 address:** 13.233.3.103 [open address]
 - Private IPv4 addresses:** 172.31.36.171
 - Public IPv4 DNS:** ec2-13-233-3-103.ap-south-1.compute.amazonaws.com [open address]
 - Elastic IP addresses:** -
- Storage:** AWS Compute Optimizer finding

6. Below you can see the error.

The screenshot shows the AWS EC2 Instances page. A red error box is displayed at the top, stating:

Failed to stop the instance i-07b848600623fd9fe
 You are not authorized to perform this operation. User: arn:aws:iam::878893308172:user/demouser is not authorized to perform: ec2:StopInstances on resource: arn:aws:ec2:ap-south-1:878893308172:instance/i-07b848600623fd9fe because no identity-based policy allows the ec2:StopInstances action. Encoded authorization failure message: gyrsOvxdael3kFe_OFZsyEdDCPvLcOtumGGMH0lhv97SnS80eJTu3wvpSO1fg0Lvpj_iu1U7bM3Rf140YnrjTvsUGOJ21IPd-lszA2MnoY8mSGA021fqZJHsVA2VmC8nf-a3twhK-aPuuhQH3ok4crhAybnpMLqlqVdpZ5BhfKkwOy0uHvYksjBGIApQbMYwZvqkeFg-4fqlxQwvY2u5y5VBxVf-1aA9_yfWXLZk1Y44aG41rlP2lrb51aTcw_Nqt8RA7tnXMHHhdLMyz-q5TRQbjfPyZYDDLoLludTFZWh6gg5vxl0SKXNtULDlmlnMB9-cwYm-KrICPJ_A_cyL50GUewcTWlruWQhMeBEUcDIOQ66th70bd2TjGSWPRB0poBMfp0lfq8rWq5o2IMhpk4StP23v_8T1D2OQ7rlpn30tbPFBPxY45113y0XHmTr8ac2menRR2HYvUI045jxNs9_wEmMsvzflWZ_ZZlcTeimNOW5xkySC65TO5INTx-A-VAemKhN7yu3zH_WEvID8W6srZ7nZDJCqqWEKG13Q2tTQ1JTEIKEGNpppy2FHrlLkmulCHJG7sedXBpV4AOqn_5kHcrLwqAlznVklm0w8xc80cVJHw_AWY4WcPvdwWCP8FUVkuW2qhecNOOpIu4i6uO8FwJ7f3-tfin5qk2RblAKUJrgxLcRkAmrlB3s7j4hM2bt6vT6kv78_xUsLDI_WQf-qgDIOHZHlzJyvwL0jbLkjigC3zO5HWtm-kPF7J2dsS6zjPfpzxEdCqRsCdBrpbw

Below the error box, the 'Instances (1/1) info' table is visible, showing the same instance details as the first screenshot.

- Now go back to your root account or main account and navigate to IAM then you are going to create a user group.
- Here, you are going to give a name to your group then select the IAM user and attach the policy for EC2 full access. Then click on create group.

User group name
Enter a meaningful name to identify this group.
demo group
Maximum 128 characters. Use alphanumeric and '+,-,@,_' characters.

Add users to the group - Optional (1/2) Info
An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

User name	Groups	Last activity	Creation time
CrossAccount-Alex	1	12 days ago	12 days ago
demouser	0	12 days ago	25 days ago

Attach permissions policies - Optional (1/925) Info
You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

Policy name	Type	Used as	Description
AmazonEC2FullAccess	AWS managed	Permissions policy (1)	Provides full access to Amazon EC2 via...

Create group

9. So, now the thing is first we attached read only permission directly to the user then we attached this user to the group and in that group, we attached EC2 full access. According to this the user has both read only and full access permission attached.

10. Now if you go back to your IAM user in another browser and again try to stop the instance. Below you can see that you are able to stop this instance.

Successfully stopped i-07b848600623fd9fe

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
demo-instance	i-07b848600623fd9fe	Stopping	t2.micro	2/2 checks passed	View alarms	ap-south-1a

11. Come back to IAM in your root account and open your user. In the dashboard of your user if you'll scroll down a little you can see an option for permission boundary.

12. You can also see how the permissions are attached to this user.

Permissions | Groups (1) | Tags (1) | Security credentials | Access Advisor

Permissions policies (2)
Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via
AmazonEC2FullAccess	AWS managed	Group demo-group
AmazonEC2ReadOnlyAccess	AWS managed	Directly

Permissions boundary (not set)

13. Now click on permissions boundary and then set permissions boundary.

▼ Permissions boundary (not set)

Set a permissions boundary to control the maximum permissions for this user. Use this advanced feature used to delegate permission management to others. [Learn more about permission boundaries](#)

[Set permissions boundary](#)

14. Now you are going to set boundary for EC2 read only permission.

Set permissions boundary on demouser

Permissions policies (1/927)
Select policy to set as the permissions boundary.

Filter by Type: All types | 1 match

Policy name	Type	Attached entities
AmazonEC2ReadOnlyAccess	AWS managed	2

[Cancel](#) [Set boundary](#)

15. Below you can also verify that the permission boundary has been set successfully.

▼ Permissions boundary (set)

Set a permissions boundary to control the maximum permissions for this user. Use this advanced feature used to delegate permission management to others. [Learn more about permission boundaries](#)

Permissions boundary:

- AmazonEC2ReadOnlyAccess (AWS managed)

[Change boundary](#) [Remove boundary](#)

16. Now you are going back to the user in another browser and try to start the instance and instantly you will get an error.

Failed to start the instance i-07b848600623fd9fe

You are not authorized to perform this operation. User: arn:aws:iam::878893308172:user/demouser is not authorized to perform: ec2:StartInstances on resource: arn:aws:ec2:ap-south-1:878893308172:instance/i-07b848600623fd9fe because no permissions boundary allows the ec2:StartInstances action. Encoded authorization failure message: SwdATTO_3EJyKN93L_7u8Y57VMyedZ00seykB2FdKfTobPQGGALoOnQGy90sA0NrWWNVNqnfz2v8XnrP0zs3BzbKb483ddQ9MKVNgMhm4svZfY8YKQh_ khBdq6PZZL2lgr9J0QaaQvasYhzDJrXXYNaDjUzeK1mula62UC06yG4QxPpQB7stCQ1c84IAQolfpRQWQlbCLhWn9Lv35W7QriPRGDDeu-nFRriOu6i- vvYT1MeRMwalgzX4JlnIVsXsk15yREpFZEEB-xTJVajFCeSoalOah35MUrMP7g4Xj_L_W_R9G3AbWevRX2EzpU4WB-QPvtvEt_DsV6KpF2b6uL- SrfGfwRYUkgkSlgIkSjYO0vzAv4D5AmnAnCOCQTCNsxEr5dfCQeWV45KkoOKFSIB4yauzBdDRKTZI25cmaHtpVrv9cNhrOlRhnu- J28GLjy50VUZKokbQ7VF2Ejdmti_oxy7QWCh6R4kGqsKl0UsaqAw5TpYExm_lstrLa9lh3UWYYUrqlskPMNp4L4_ADu5_w-yVqeqASFQVbVpjGorVyBAFBXT- Ok7jKKTZXkx2YVYVG2tdExL4xhNprKuOhb5vnHTLSryBhr9OYtpuecnVHy4XkpmHm-J5xjED0JgGvQtAHcE7YOG0Z5DhGMtjvc5o9-FabWUpDgjBMFaInd5YoSW- KxylApXlwhyggsSnRMOkGmJFgT9J6J_h2Cs4Lxfcx_p8QbnBDWbz1u

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
demo-instance	i-07b848600623fd9fe	Stopped	t2.micro	-	View alarms +	ap-south-1a	-

Instance: i-07b848600623fd9fe (demo-instance)

17. This means that permission boundary has set the hierarchy for the permission which you have attached to the IAM user.

In this lab exercise, we are demonstrating the implementation of IAM (Identity and Access Management) permission boundaries in an AWS (Amazon Web Services) environment. The end goal is to enhance security and compliance by setting clear limits on the maximum permissions that IAM entities (such as users or roles) can have, ensuring they cannot exceed predefined boundaries, even if other policies would grant broader permissions. This helps organizations maintain control over access levels and reduce the risk of unauthorized actions within their AWS environments.