Access Policy and IAM

Primary Owner aes-ops-support-l3 (LDAP)

Last modified 5 months ago by burnthm.

Access Policy and IAM

Amazon Opensearch Service offers numerous security features, including fine-grained access control, IAM, SAML, Cognito authentication for Dashboards, encryption, and VPC access.

Public Documentation - https://docs.aws.amazon.com/opensearch-service/latest/developerguide/ac.html

Access Policy

Access policies controls whether a request is accepted or rejected when it reaches the Amazon Opensearch Service domain. If you specify an account, user, or role in this policy, you must sign your requests.

- Amazon Opensearch Service offers several ways to configure access to your domains. For more information, see Identity and Access Management and Fine-grained access control in Amazon Opensearch Service.
- The AWS console provides preconfigured access policies that you can customize for the specific needs of your domain. You also can import access policies from other AOS domains. For information about how these access policies interact with VPC access, see About access policies on VPC domains.

```
To modify Access Policy, you can go to 'Actions' -> 'Modify Access Policy'
```

IDENTITY AND ACCESS MANAGEMENT

AMAZON ELASTICSEARCH SERVICE OFFERS SEVERAL WAYS TO CONTROL ACCESS TO YOUR DOMAINS. IAM OFFERS VARIOUS POLICY TYPES AND DEALS WITH HOW THEY INTERACT WITH EACH OTHER, AND HOW TO

CREATE YOUR OWN CUSTOM POLICIES.

Types of Policies

AOS supports three types of access policies:

- Resource-based policies
- Identity-based policies
- IP-based policies

Making and Signing Amazon Opensearch requests

All requests to the Amazon Opensearch Service configuration API must be signed, even if you configure a completely open resource-based access policy. If your policies specify IAM users or roles, requests to the Opensearch APIs also must be signed using AWS Signature Version 4. The signing method differs by API:

- To make calls to the Amazon OpenSearch configuration API, we recommend that you use one of the AWS SDKs. The SDKs greatly simplify the process and can save you a significant amount of time compared to creating and signing your own requests.

```
If you use one of the SDKs, such as Boto 3, the SDK automatically handles the request
```

- To make calls to the Opensearch APIs, you must sign your own requests. For sample code to sign requests in Python client, see Signing HTTP requests to Amazon Opensearch Service.

```
If your domain access policy includes IAM users or roles (or you use an IAM master use
```

- To sign curl requests, you can use the aws sigv4 option with version 7.75.0 or later.

When policies collide

Complexities arise when policies disagree or make no explicit mention of a user. Understanding How IAM Works in the IAM User Guide provides a concise summary of policy evaluation logic:

- By default, all requests are denied.
- An explicit allow overrides this default.
- An explicit deny overrides any allows.

For example, if a resource-based policy grants you access to a domain subresource (an Opensearch index or API), but an identity-based policy denies you access, you are denied access. If an identity-based policy grants access and a resource-based policy does not specify whether or not you should have access, you are allowed access. See the following table of intersecting policies for a full summary of outcomes for domain subresources.

	Allowed in resource-based policy	Denied in resource-based policy	Neither allowed nor denied in resource-based policy
Allowed in identity-based policy	Allow	Deny	Allow
Denied in identity-based policy	Deny	Deny	Deny
Neither allowed nor denied in identity-based policy	Allow	Deny	Deny

Troubleshooting Issues related to Access Policy and IAM

- ***Error: UpdateElasticsearchDomainConfig: {"message":"Apply a restrictive access policy to your domain"}***
  Amazon Opensearch requires the access policy to be restrictive.
  - An easy way out of this issue would be to add restrictions based on IP addresses, by white-listing your IP address or the ARN of your IAM role in the access policy.
- ***Customer is running their code in a Lambda function that accesses their AOS cluster and they want to allow this code to only perform GETs on the cluster***
  Make sure to include the LambdaExecutionRole when you are trying to set the permissions for this in the AOS access policy.
  - By default, the ***'LambdaExecutionRole'*** is given full AOS access, advise customer on setting the *'ES GET'* permissions in the Lambda execution role.
- ***Error: An error occurred (ValidationException) when calling the CreateElasticsearchDomain operation: Enable fine-grained access control or apply a restrictive access policy to your domain***
  The above error usually indicates that the access policy is open to internet and also the FGAC is not enabled on this domain.
  - Fine Grained Access Control for AWS ES is an additional layer of security. When you are creating an Amazon ES it is required to either enable FGAC or add access policy, this is because in FGAC you are required create master credentials which can be used to provide security and prevent any possible misuse used due to exposing the cluster to the world .
  - In case FGAC is not something that interests you, you can create a domain as per the usual and create a restrictive access policy i.e., limit access by allowing only the required IAM entities or IP as per your use case.
  - Another recommendation is to restrict this access policy further (for example: limit action es:* for certain domain only and limit to some IAM user/role instead of AWS:*)
- ***Customer getting "Missing role" errors while trying to integrate Okta***
  The above errors can be caused:
  - If the users are not added inside a Group in Okta, or
  - If the Group is not added to the Group Attribute Statements
  - If the role mapping is missing between the SAML role and the roles inside the cluster (under the _opendistro/_security/api/roles endpoint).
- ***Error: "The provided execution role does not have permissions to call CreateNetworkInterface on EC2"***
  The error message is due to insufficient permission with the IAM role that Lambda should use when executing calls to your Amazon ES.
  - If the target Amazon ES domain uses VPC access, the role must have the ***'AWSLambdaVPCAccessExecutionRole'*** policy attached. This Amazon-managed policy grants Lambda access to the customer's VPC, enabling Lambda to write to the Amazon ES endpoint in the VPC.

- ***Error: "User: is not authorized to perform: es:ESHttpGet with an explicit deny"***
  The above error can be caused when there is an explicit deny on user permissions.
    - Check the access policy of the cluster and see if there are any explicit deny's.
    - If not, check if there are any resource based policies attached to your domain which could be causing this deny. To learn more, please visit How to Control Access to Your Amazon Elasticsearch service domain.
    - If there is no explicit deny via inline/managed policies attached to your IAM role, it is likely coming from your Organization Service Control Policies (SCP). Recommend checking if any SCP is blocking your action at account level.
- ***Error: "User: anonymous is not authorized to perform: es:ESHttpGet"***
  Since the Kibana endpoint doesn't support signed requests thus, if the Access Control Policy for your domain only grants access to certain IAM users or roles and you haven't configured Amazon Cognito authentication, you might receive the above mentioned error.
    - If your AOS domain uses VPC access, you might not receive this error, but the request might time out. To learn more about correcting this issue and the various configuration options available to you, see Controlling access to Kibana, About access policies on VPC domains, and Identity and Access Management in Amazon Opensearch Service.
    - In order to resolve the issue, you can configure your IP address in the Access Policy using IP based access policy or you can enable Amazon Cognito Authentication.

L1 and L2 SOP

Step 1

When a customer reaches out that they are facing issues related to access policy or IAM, look for the access policy for the AOS cluster in OpsTumbler page under *Configurations* → *Access Policies.* You can also use "AOS-GetDomainResources" tool linker tool and navigate to *Securtiy Configurations* → *Access Policy.*
Ask the customer to provide the *'IAM ARN'* they are using to access the cluster and check in the 'K2-IAM' or 'IAM-SR' if the role has necessary permissions required to access the Opensearch cluster.

Step 3

Request the customer to run *'aws sts get-caller-identity'* command via CLI, which returns details about the IAM user or role whose credentials are used to call the operation.

Step 4

Request the customer to confirm if they are using the same IAM as returned in the output of *'aws sts get-caller-identity'* to access the cluster.

Step 5

Check if the domain is VPC access based or Public access domain, look for *Configurations* → *VPC Options* in OpsTumbler page. If there is a 'Vpc Id' listed, then it is a VPC domain. Else, it is a public access domain.

Step 6

To check if the domain has FGAC enabled or not, check *'Configurations'* in Ops Tumblr. Navigate to *Configurations* → *Advanced Security Options.* If 'Enabled' is true, then FGAC is enabled.

`Note: If the issue has to be escalated,`

- for SupportOps team, choose 'CTI' as: **Category**- *AWS*, **Type**- *Support Ops Elasticsearch*, **Item**- *Access Policy Issues*, **Severity**- *3to5*
- for Service team, choose 'CTI' as: **Category**- *AWS Elasticsearch*, **Type**- *Dataplane*, **Item**- *Security*, **Severity**- *2*

Tags: inclusive_tech_exceptionX

[+]