NetBrain® Integrated Edition 10.1

Quick Setup Guide (AWS)

Version 10.1 | Last Updated 2022-09-16

Contents

1. AWS API Access Overview

NetBrain uses API (more specifically, Boto3 SDK) to retrieve the data from AWS. There are different ways to configure access to AWS, and we will explore each method in detail.

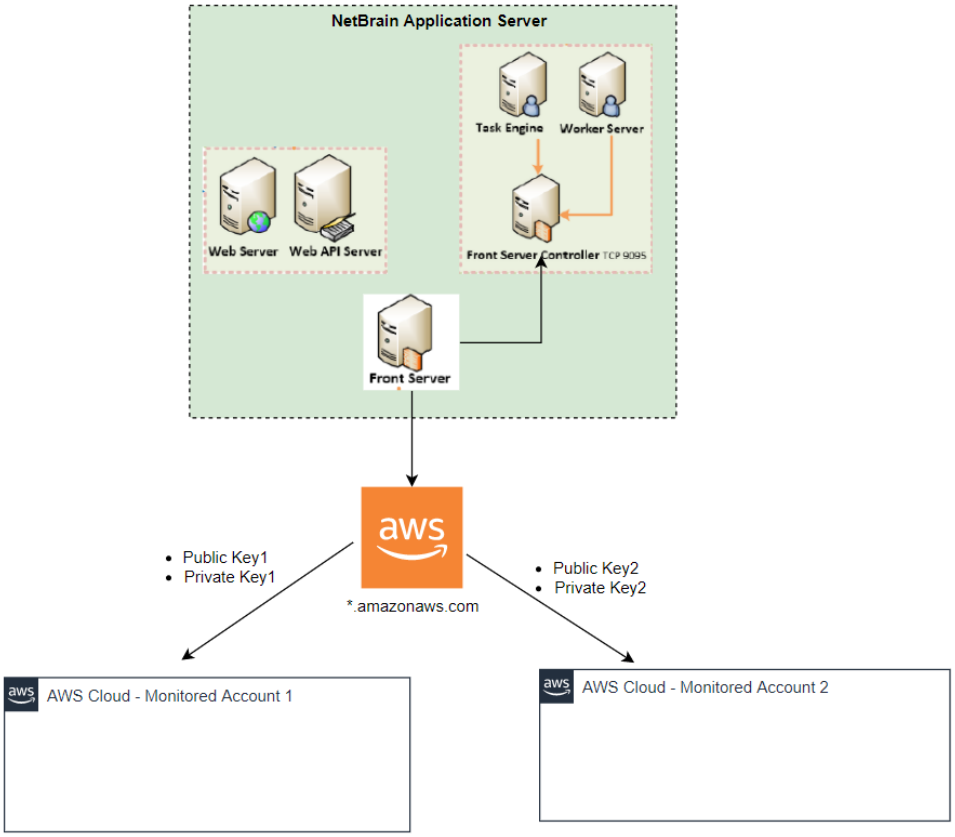
1. **Key-based Access**: Set up public and private keys so the NetBrain IE system can use static key(s) to discover AWS resources.
2. **Role-based Access**: Set up different roles for the NetBrain IE system to access AWS accounts, and it doesn't require any static key.
3. **Combined Access**: Configure the key-based access for one master account and then access the monitored accounts via the role-based access method.

1.1. Key-based Access Overview

NetBrain requires AWS public key and secrete key to be configured to access the data from AWS for key-based access. NetBrain will use the configured credentials to send HTTP requests via Front Server. Therefore, Front Server is required to access the Amazon AWS websites from an Internet access perspective: ***.amazonaws.com.**

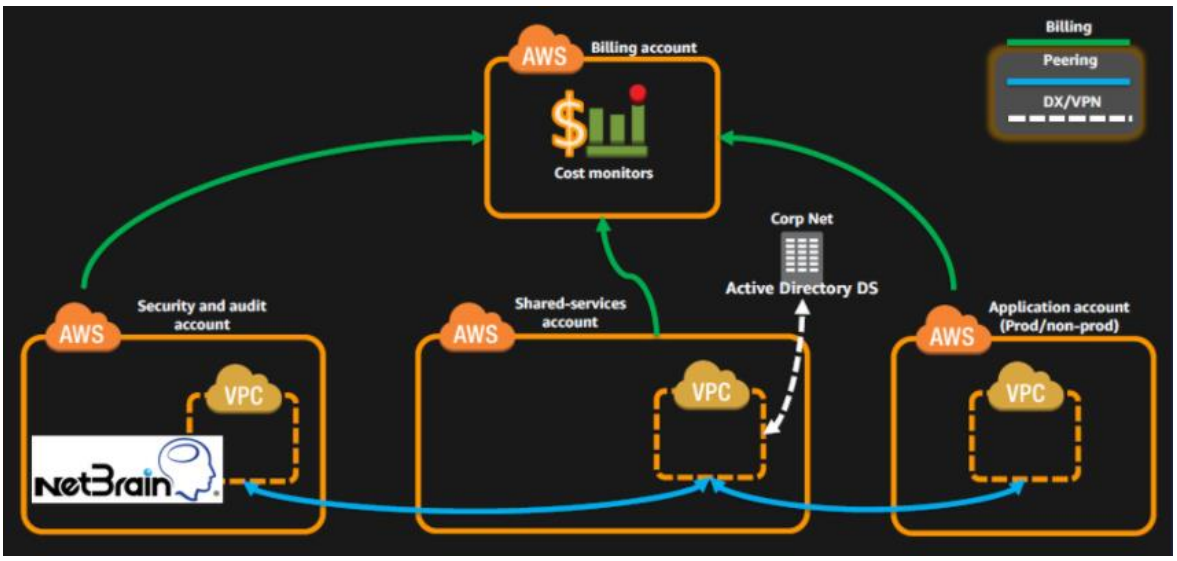
The following diagram shows how to configure the NetBrain servers to access your different AWS accounts, named monitored accounts (where the infrastructure data resides). In this deployment model, you will need to create static keys (including public and private keys) for each account and use these keys to access AWS resources.

As the requirement is to access the Amazon AWS website from the Front Server, you may deploy the NetBrain Front Servers in your on-prem data center or AWS. And there is no limitation on how to deploy NetBrain Front Servers. If you have traditional devices, CPE devices, or devices in the colocation to be discovered, make sure that the Front Server has access to these devices.



1.2. Role-based Access Overview

Role-based access requires you to configure the proper roles for NetBrain to assume for data retrieval. The following diagrams demonstrate the high-level concepts of role-based access deployment:



There are two types of accounts:

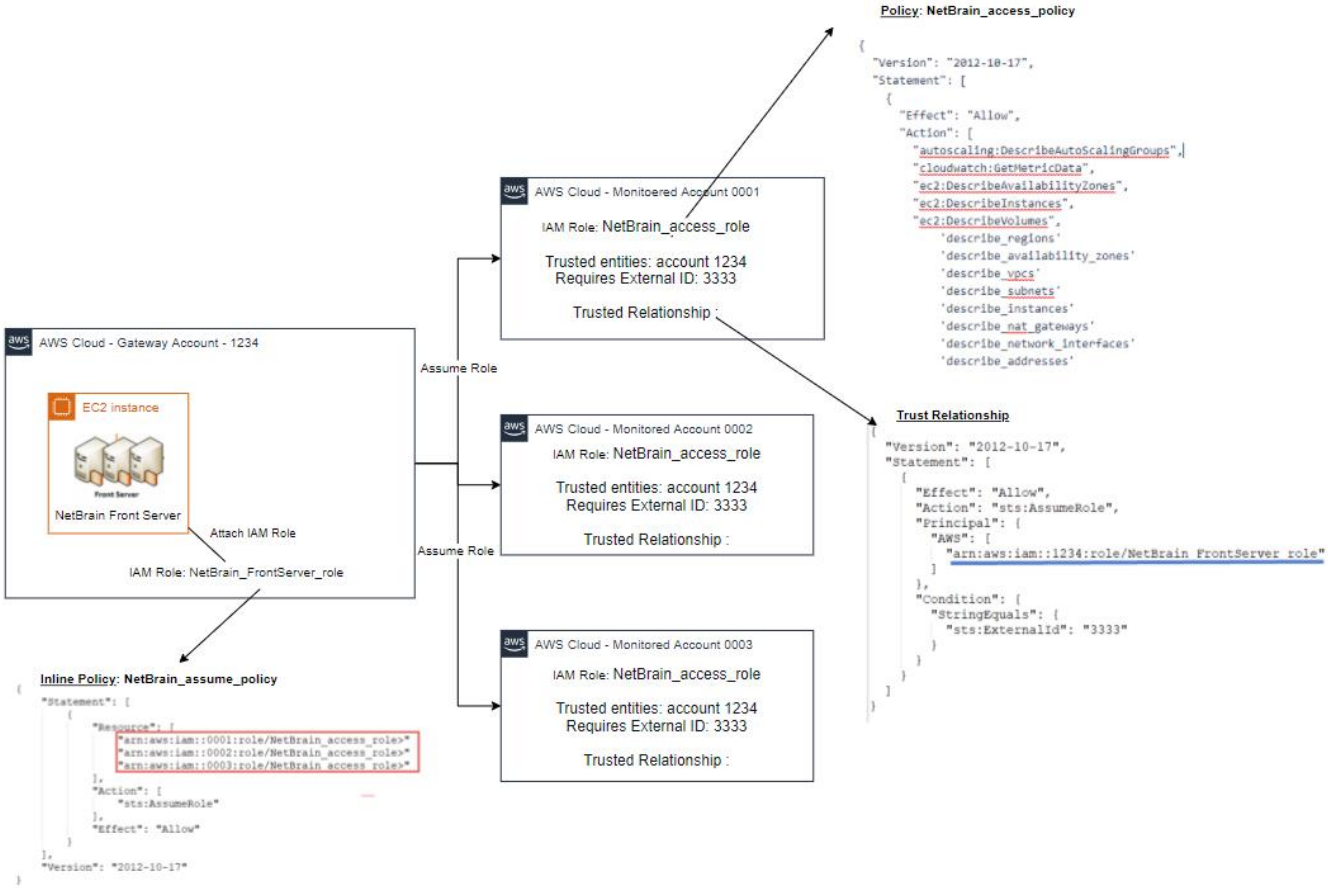
1. **Gateway Account**: Gateway account delegates access to other accounts. It is typically the account for monitoring, security, and auditing purposes in multi-account architecture.
2. **Monitored Accounts**: Accounts that host infrastructure data and need to be discovered.

The solution requires the NetBrain Front Server to run on an EC2 instance in a gateway account. In the account to be monitored, a role needs to be created to delegate and authorize access from the EC2 instance in the gateway account.

Once the proper role and policy have been configured, NetBrain Front Server can read the network configurations and run statistics from the monitored accounts.

The following diagram shows a detailed structure of this deployment.

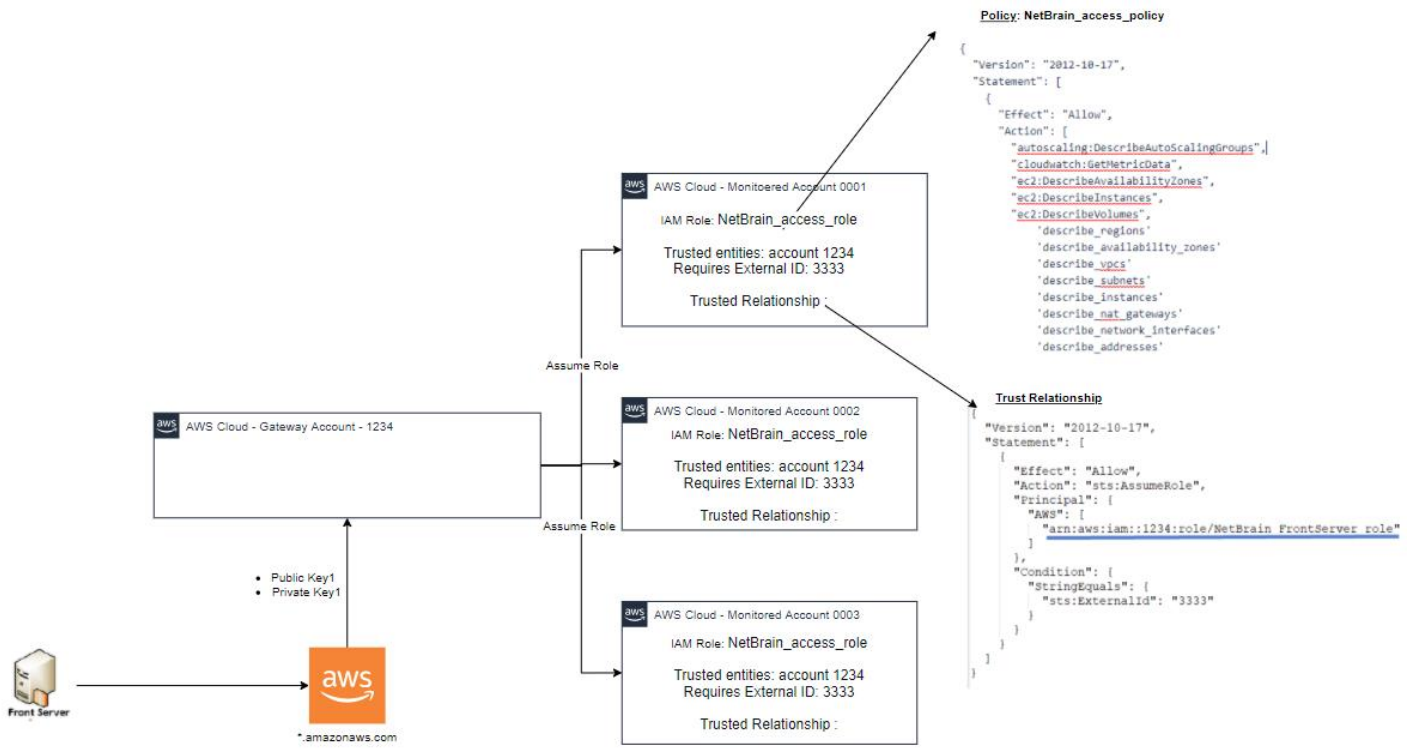
> **Note:** You only need to install the Front Server within an EC2 instance to assume proper roles. You can still have other NetBrain components in your on-prem Data Centers for communication purposes if you have IPSec or direct connections to the cloud environment.



1.3. Combined Access Overview

You sometimes don't want to permit EC2 instances to assume the role due to security or other considerations. Then, you can leverage the combined access method.

As depicted in the following diagram, we use key-based access to access the gateway account. The created user can assume the role in the monitored accounts. This way, you can install the Front Server anywhere if it has access to the AWS website.



2. Setting Up Key-based Access

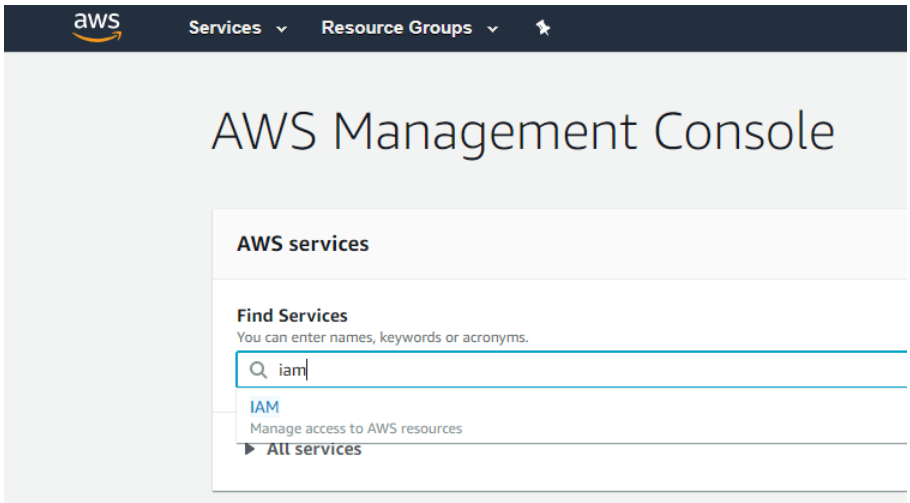
This chapter will guide you through the details of how to set up key-based access for your AWS accounts.

2.1. Creating AWS Access Policy in Amazon Console

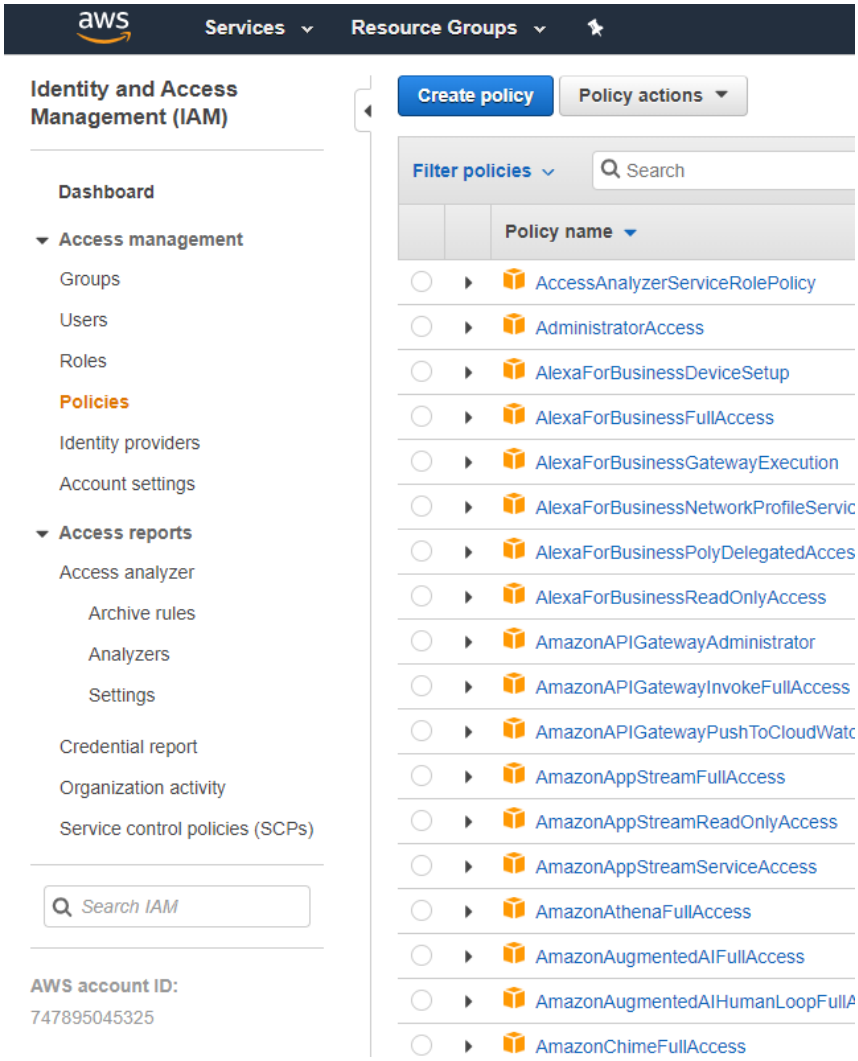
The AWS access policy defines the minimal scope of permissions that enables NetBrain to retrieve the data to build the data model and use the CloudWatch API to monitor the services running in your AWS account.

> **Note:** You can create and use the policy anytime when enabling NetBrain to access your AWS account.

1. Go to **Identity and Access Management (IAM)** in your Amazon Console.



2. Go to **Policies** and click **Create policy**.



3. Select the **JSON** tab, and paste the predefined policy in JSON as follows:

Create policy

1 2 3

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

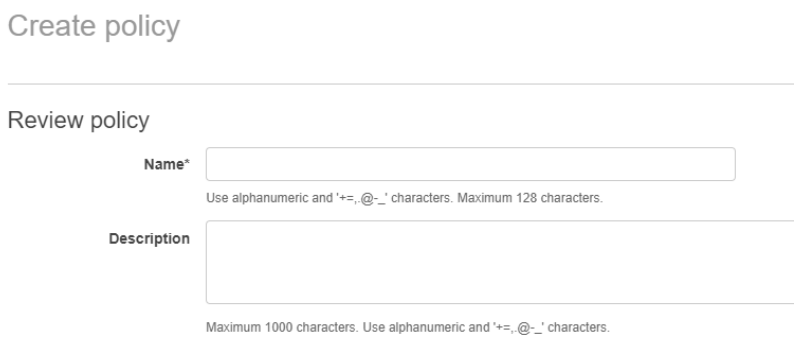
Visual editor | JSON | Import managed policy

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
          "Action": [
              "autoscaling:Describe*",
              "autoscaling-plans:Describe*",
              "autoscaling-plans:GetScalingPlanResourceForecastData",
              "cloudwatch:Describe*",
              "cloudwatch:Get*",
              "cloudwatch:List*",
              "directconnect:Describe*",
              "ec2:Describe*",
              "ec2:Get*",
              "ec2:SearchTransitGatewayRoutes",
              "network-firewall:DescribeFirewall",
              "network-firewall:DescribeFirewallPolicy",
              "network-firewall:DescribeRuleGroup",
              "network-firewall:ListFirewallPolicies",
              "network-firewall:ListFirewalls",
              "network-firewall:ListRuleGroups",
              "network-firewall:ListTagsForResource",
              "elasticloadbalancing:Describe*"
          ],
          "Effect": "Allow",
          "Resource": "*"
      }
  ]
}
```


```
{
  "Version": "2012-10-17",
  "Statement": [
      {
        "Action": [
          "autoscaling:Describe*",
          "autoscaling-plans:Describe*",
          "autoscaling-plans:GetScalingPlanResourceForecastData",
          "cloudwatch:Describe*",
          "cloudwatch:Get*",
          "cloudwatch:List*",
          "directconnect:Describe*",
          "ec2:Describe*",
          "ec2:Get*",
          "ec2:SearchTransitGatewayRoutes",
          "network-firewall:DescribeFirewall",
          "network-firewall:DescribeFirewallPolicy",
          "network-firewall:DescribeRuleGroup",
          "network-firewall:ListFirewallPolicies",
          "network-firewall:ListFirewalls",
          "network-firewall:ListRuleGroups",
          "network-firewall:ListTagsForResource",
```

```
            "elasticloadbalancing:Describe*"
        ],
        "Effect": "Allow",
        "Resource": "*"
    }
  ]
}
```

4. Click **Review Policy** and enter the policy name in the **Name** field (i.e., NetBrain_access_policy).



5. Click **Create policy**.

2.2. Enabling Access to Your Amazon Account Using Key-based Access

NetBrain must identify all virtualized infrastructure components in your AWS environment to get the information required to build the data model. This information is used to understand the context of your applications, services, and hosts. To enable it, you need to authorize NetBrain to access your Amazon metrics.

You can enable NetBrain to access your AWS metrics by either using a private access key (key-based access) or defining a special role for NetBrain (role-based access). In either case, make sure that your Front Server (used for data retrieval) has a connection to AWS by configuring your proxy for Front Server or whitelist ***.amazonaws.com** in your firewall settings.

NetBrain can use access keys to enable secure REST or Query protocol requests to the AWS service API. You will need to generate an Access Key ID and a secret access key so NetBrain can use them to get the metrics from Amazon Web Services.

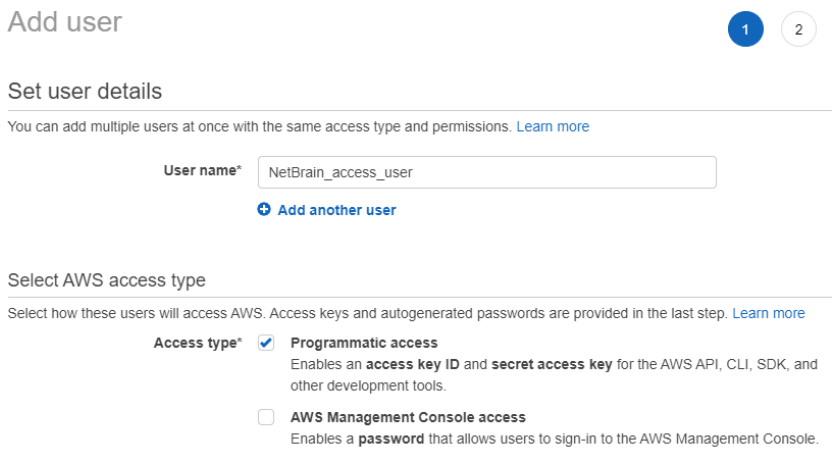
> **Note:** If you add multiple AWS accounts to NetBrain, you must repeat these steps for each account.

Prerequisites:

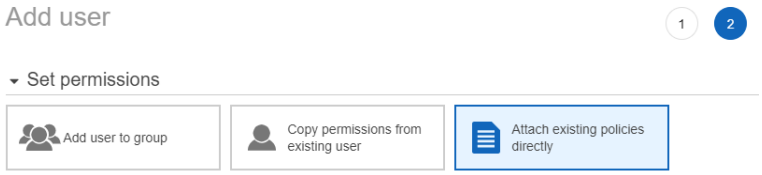
- Rights to create a new AWS user
- AWS account ID
- The Amazon Access Key ID and secret access key

Proceed with the following steps:

1. In the Amazon IAM Console, click **Users** > **Add user**.
2. Enter a name for the key, for example, **NetBrain_access_user**.
3. In the **Select AWS access type** area, select the **Programmatic access** check box and click **Next: Permissions**.



4. Click **Attach existing policies directly** and select the monitoring policy you have defined: **NetBrain_access_policy**, then click **Next: Review**.

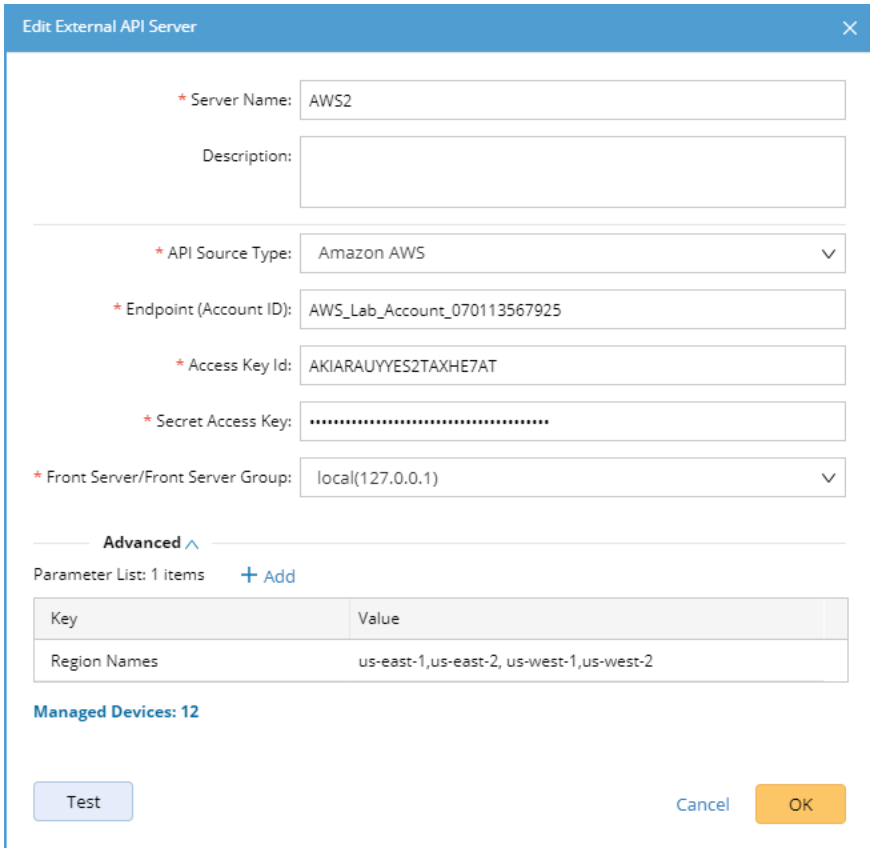


5. Review the user details and click **Create user**.
6. Store the Access key ID name (AKID) and secret access key values. You can either download the user credentials or click **Show** to copy the credentials displayed online.

2.3. Configuring NetBrain to Access AWS Using Key-based Access

Once you've granted AWS access to NetBrain, you need to connect NetBrain to your Amazon AWS account.

1. On the Domain Management page, select **Operations > Discover Settings > API Server Manager** from the quick access toolbar.



2. In the **Server Name** field, enter a meaningful name that can uniquely identify your AWS account.

3. Create a new external API server and select **Amazon AWS** as the **API Source Type**.

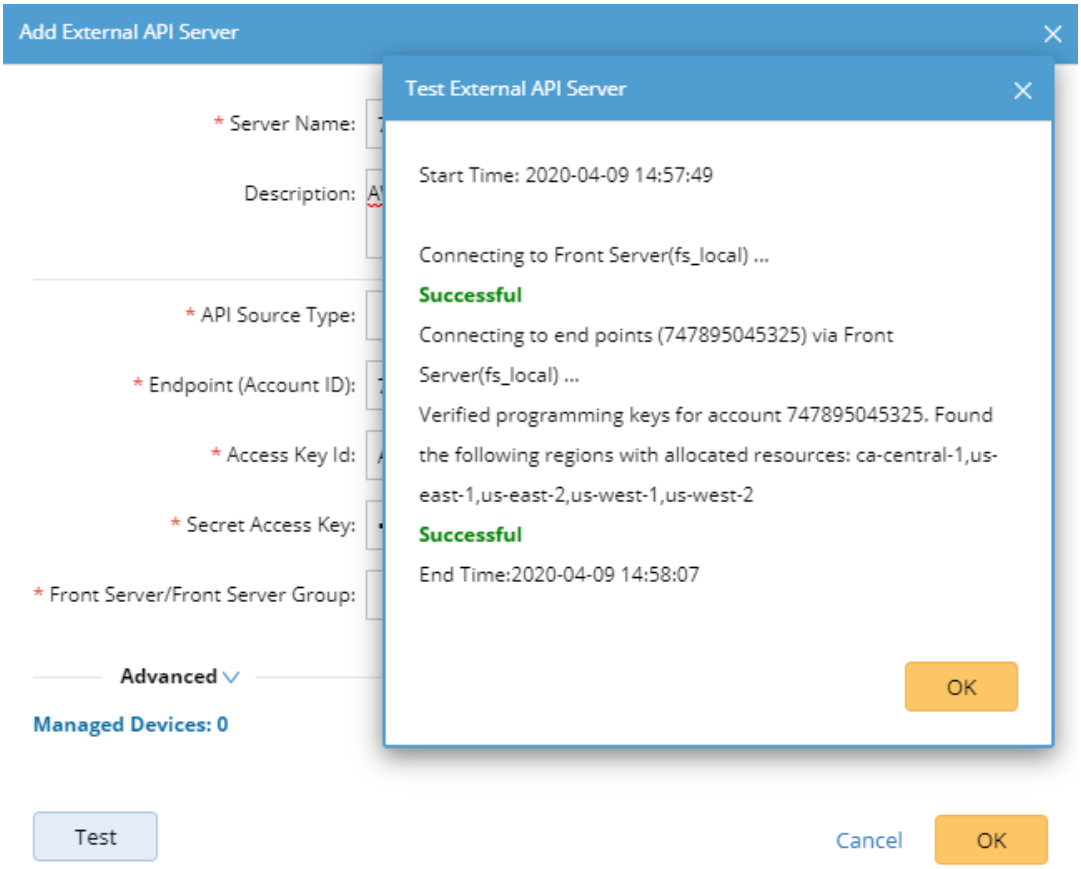
   1) In the **Access Key Id** field, paste the identifier of the key you created in AWS for NetBrain access.

   2) In the **Secret Access Key** field, paste the value of the key you created in AWS for NetBrain access.

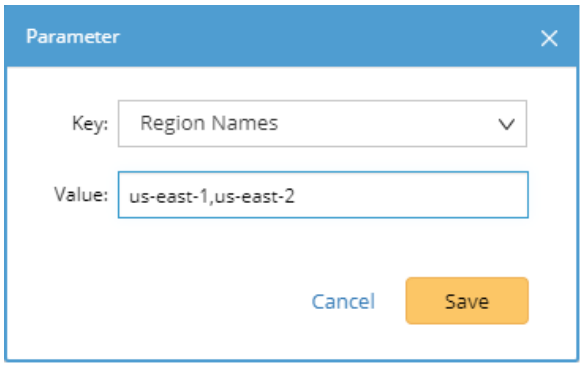
   3) In the **Endpoint (Account ID)** field, enter the AWS account identifier.

   4) Click **Test** to verify the connection.

   5) Click **OK** to save the connection.



4. Once the connection is verified and saved, you can proceed to [Discovering AWS Network in NetBrain Domain] to start the data retrieval process.

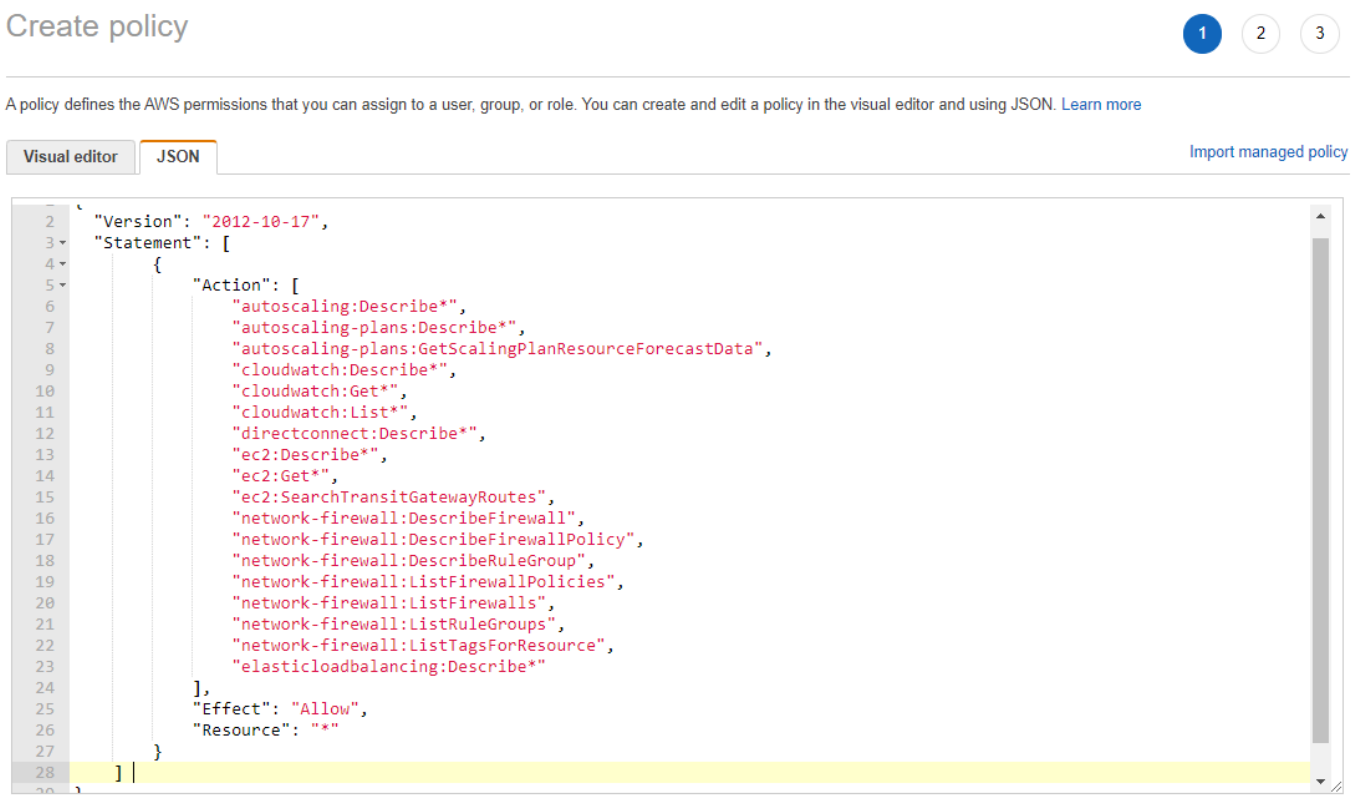
> **Note:** By default, NetBrain queries all regions in your AWS accounts for data retrieval. NetBrain will further identify whether there are resources for these regions based on whether the ENI interface exists in these regions. If you only want to retrieve the data for specific regions, you can specify the regions you want NetBrain to access in the **Parameter List** field.
>
> 
>

3. Setting Up Role-based Access

This chapter will guide you through how to set up role-based access for your AWS accounts.

3.1. Creating AWS Access Policy and Role for Monitored Accounts

1. Go to **Policies** in **Identity and Access Management (IAM)**.
2. Create a new resource access policy to grant read access to the services for monitoring purposes.

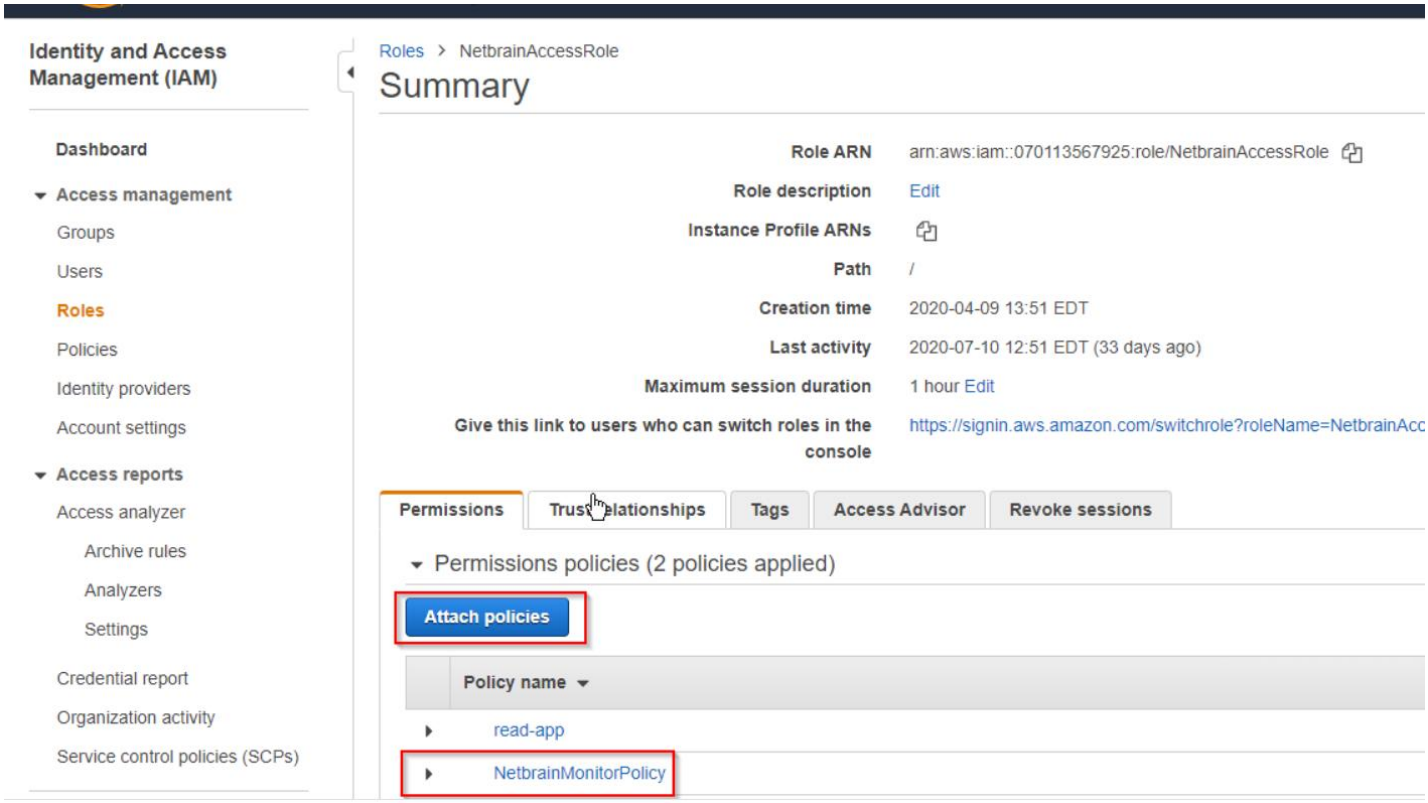



```
{
        "Version": "2012-10-17",
        "Statement": [
            {
      "Action": [
        "autoscaling:Describe*",
        "autoscaling-plans:Describe*",
        "autoscaling-plans:GetScalingPlanResourceForecastData",
        "cloudwatch:Describe*",
```

```
            "cloudwatch:Get*",
            "cloudwatch:List*",
            "directconnect:Describe*",
            "ec2:Describe*",
            "ec2:Get*",
            "ec2:SearchTransitGatewayRoutes",
            "network-firewall:DescribeFirewall",
            "network-firewall:DescribeFirewallPolicy",
            "network-firewall:DescribeRuleGroup",
            "network-firewall:ListFirewallPolicies",
            "network-firewall:ListFirewalls",
            "network-firewall:ListRuleGroups",
            "network-firewall:ListTagsForResource",
            "elasticloadbalancing:Describe*"
        ],
        "Effect": "Allow",
        "Resource": "*"
    }
  ]
}
```

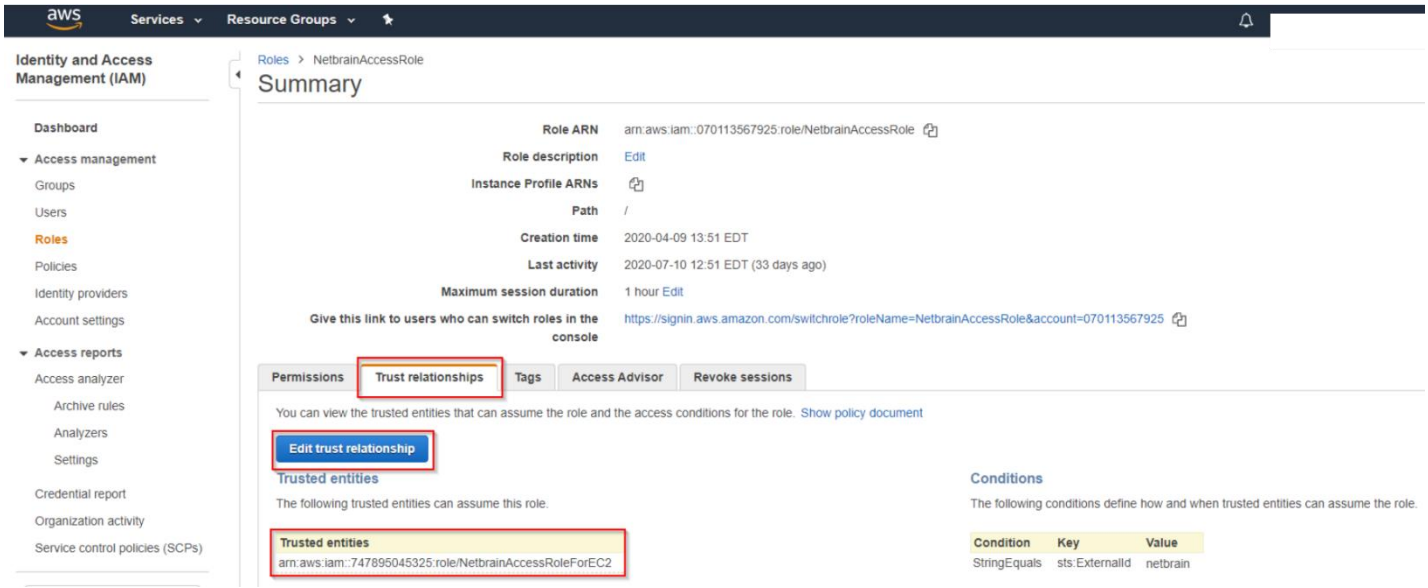
Follow the steps below to configure the role:

1. Go to **Roles** in **Identity and Access Management (IAM**).
2. Create a new role.
3. Attach the policy (created previously) to the role.



4. Go to **Trust relationships** and add the statements to allow the EC2 instance from the gateway account to assume this role.

> **Note:** The role name of the EC2 instance, for example, NetbrainAccessRoleForEC2, must match the EC2 instance role name configured in the gateway account.



The sample trust relationship JSON statements are as follows. You need to replace the account ID, role name, and External ID to reflect your specific configuration.

```
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": "sts:AssumeRole",
   "Principal": {
    "AWS": [
     "arn:aws:iam::<12-digit gateway account number>:role/<role for your EC2 Instance run Netbrain FrontServer (i.e.
NetbrainAccessRoleForEC2)>"
    ]
   },
   "Condition": {
    "StringEquals": {
     "sts:ExternalId": "<External ID generated from tenant>"
    }
   }
  }
 ]
}
```

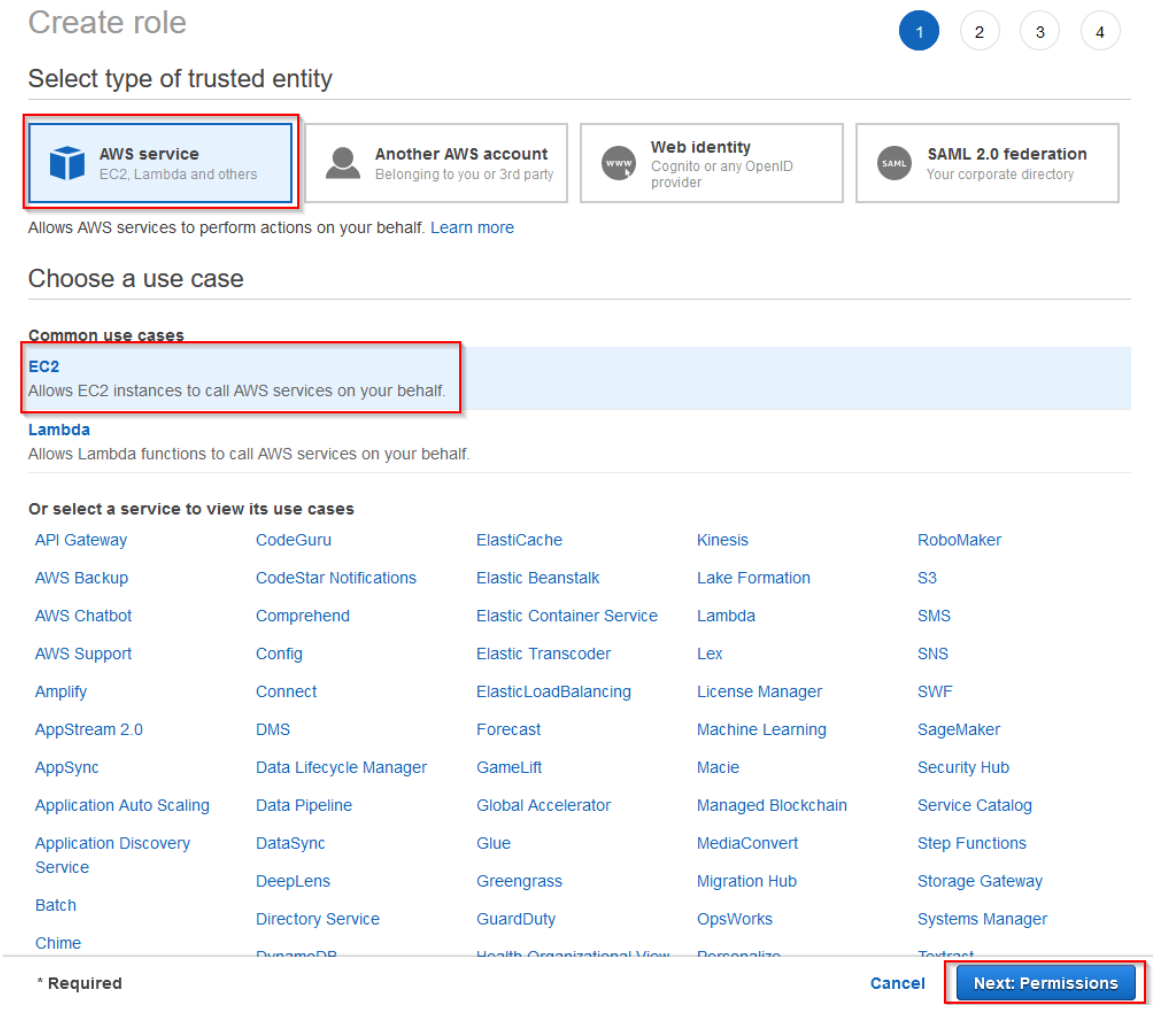
3.2. Configuring EC2 Role for NetBrain Front Server in AWS Gateway Account

This section illustrates how to create a role for an EC2 instance in the gateway account using the AWS console. This will allow the EC2 instance that hosts NetBrain system to access the monitored accounts.

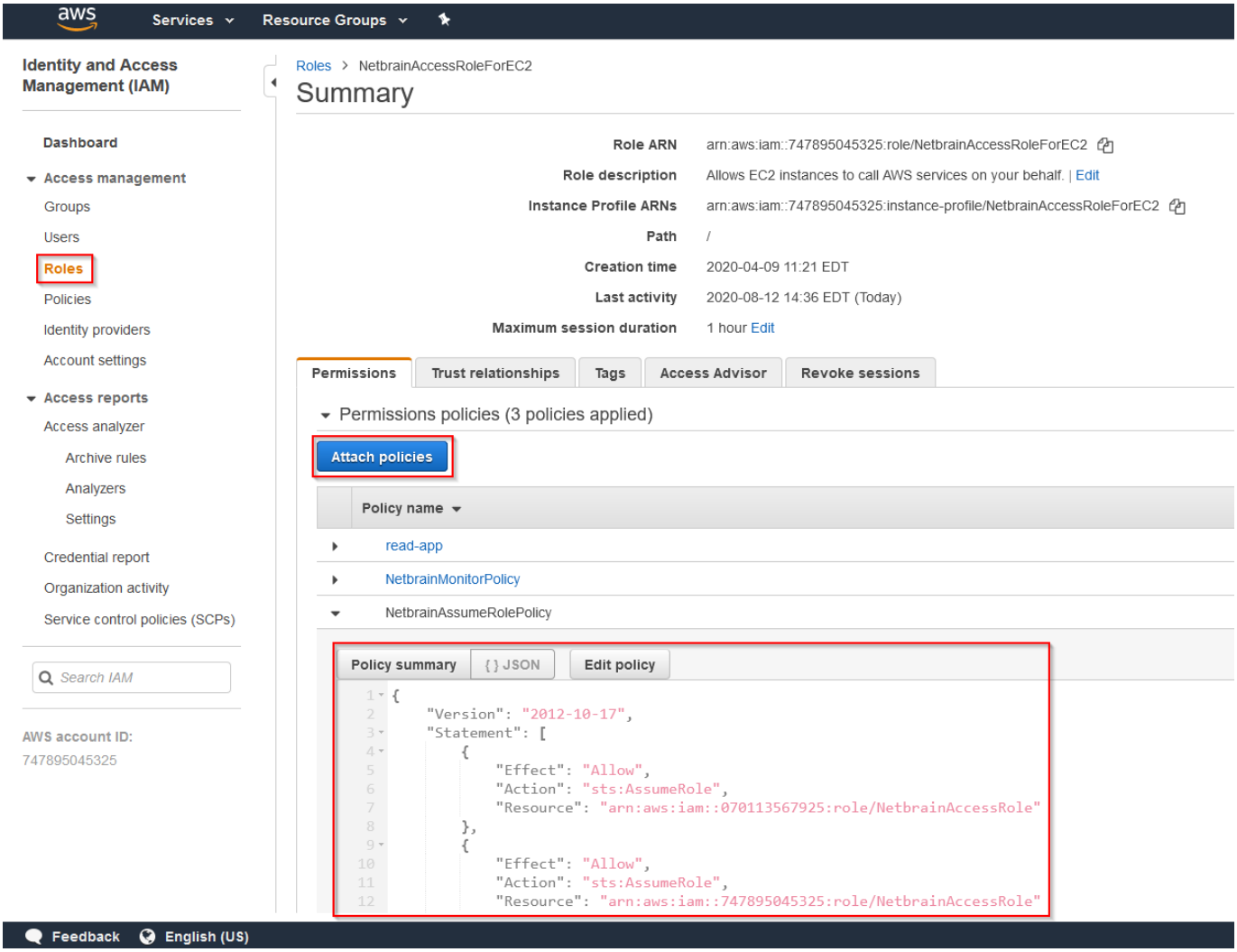
1. Go to **Roles in Identity and Access Management (IAM)** and create a new role.
2. Select **AWS service** and **EC2** for this role.
3. Enter the role name (NetbrainAccessRoleForEC2).

> **Note:** The role name shall match the one you previously picked when configuring the trusted relation in the monitored account.

4. Skip the Permissions (policy) section in the wizards. The policy will be added later.



5. After the role is successfully created, open the role and attach an inline policy to allow the EC2 instance to assume **NetbrainAccessRole** in monitored accounts.



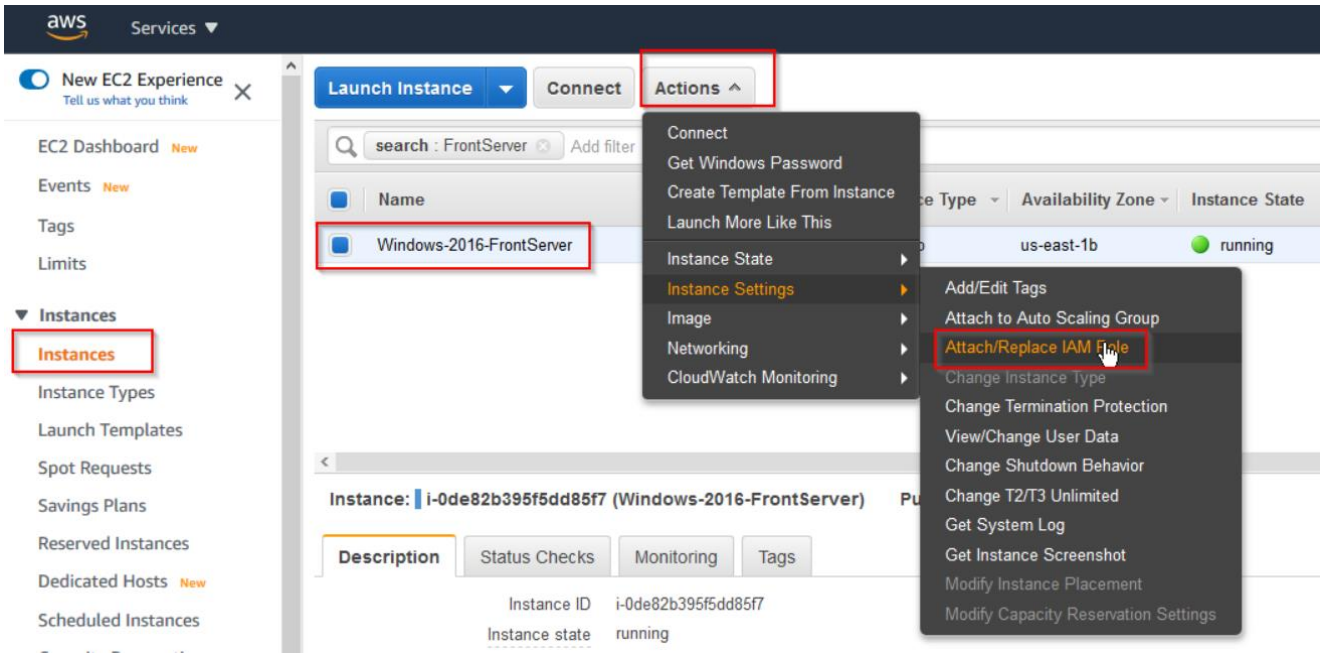
A sample policy JSON is as follows.

> **Note:** Use the account ID to monitor your environment.

```
{
  "Statement": [
    {
      "Resource": [
        "arn:aws:iam::<12-digit monitored account number>:role/<role created in previous step (NetbrainAccessRole)>"
      ],
      "Action": [
        "sts:AssumeRole"
      ],
      "Effect": "Allow"
    }
  ],
  "Version": "2012-10-17"
```

```
}
```

6. Find the EC2 instance where you run NetBrain Front Server, and attach the role to it. You can also specify the role when first launching an EC2 instance.



3.3. Configuring NetBrain System

Follow the steps below to add the accounts to monitor:

1. On the **Domain Management** page, navigate to **Operations** > **Discover Settings > API Server Manager**.

2. In the **API Server Manager** configuration page, click **Add API Server** to add an API Server entry into the table for each account to be monitored.

3. Configure the parameters in the **Edit External API Server** window as follows:

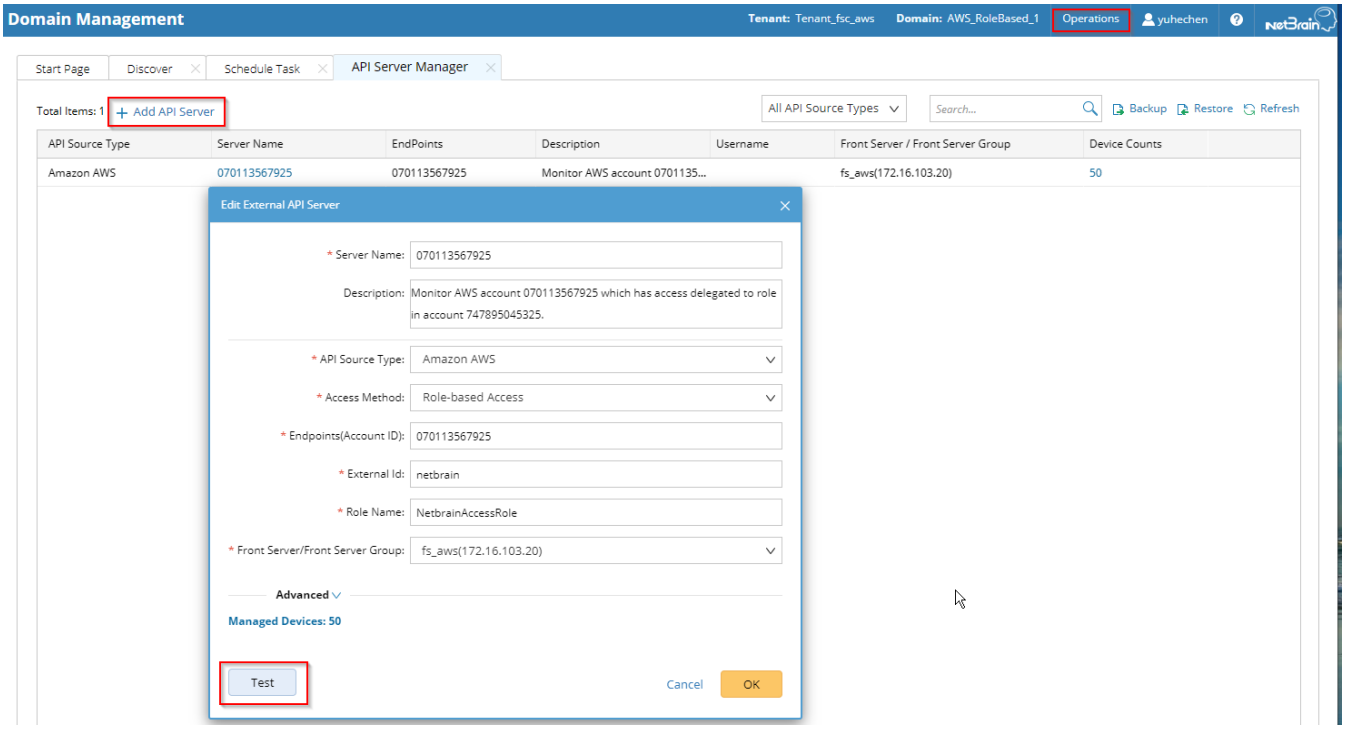
   1) **API Source Type**: Select **Amazon AWS**.

   2) **Access Method**: Select **Role-based Access**.

   3) **Endpoints (Account ID)**: Enter the AWS account ID to be monitored.

   4) **External Id**: Enter the External ID previously selected for the trust relationship in the AWS account to be monitored.

   5) **Role Name**: Enter the role name previously selected in the AWS account to be monitored.



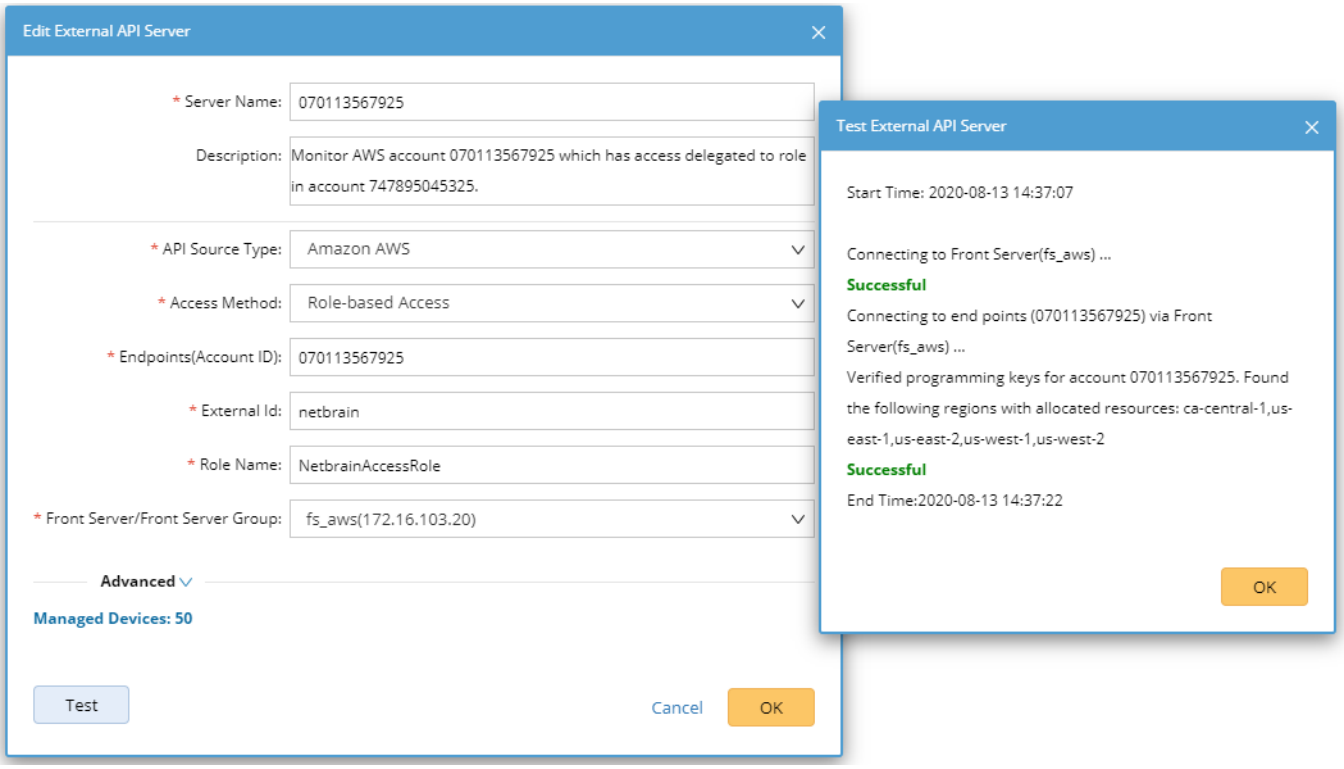
> **Tip:** Alternatively, you can call NetBrain northbound APIs to add/update/delete AWS accounts if you have integrated them with your NetOps automation flow. For more information about the APIs, refer to [Using REST API to Manage AWS Data](#).

More information about the configuration parameters is as follows:

	Display Name	***Mandatory***	***Notes***
Combined	Authentication Method	Yes	Authentication method to access account resources. Use the drop-down menu to select from KeyBase, RoleBase, or Combine.
	Endpoint (Account ID)	Yes	The AWS account to be monitored.
	Region Names	No	Comma-separated official AWS region names. Explicitly specify and limit the regions to monitor. Default to all publicly accessible regions if not specified.
Key-Based	Access Key Id	Yes	Program access key associated with an IAM user, which can be used for programmatic access to AWS account resources.
	Secret Access Key	Yes	The secret key associated with the access key for authentication purposes.
Role-Based	Role Name	Yes	Role configured in AWS account for role-based access.

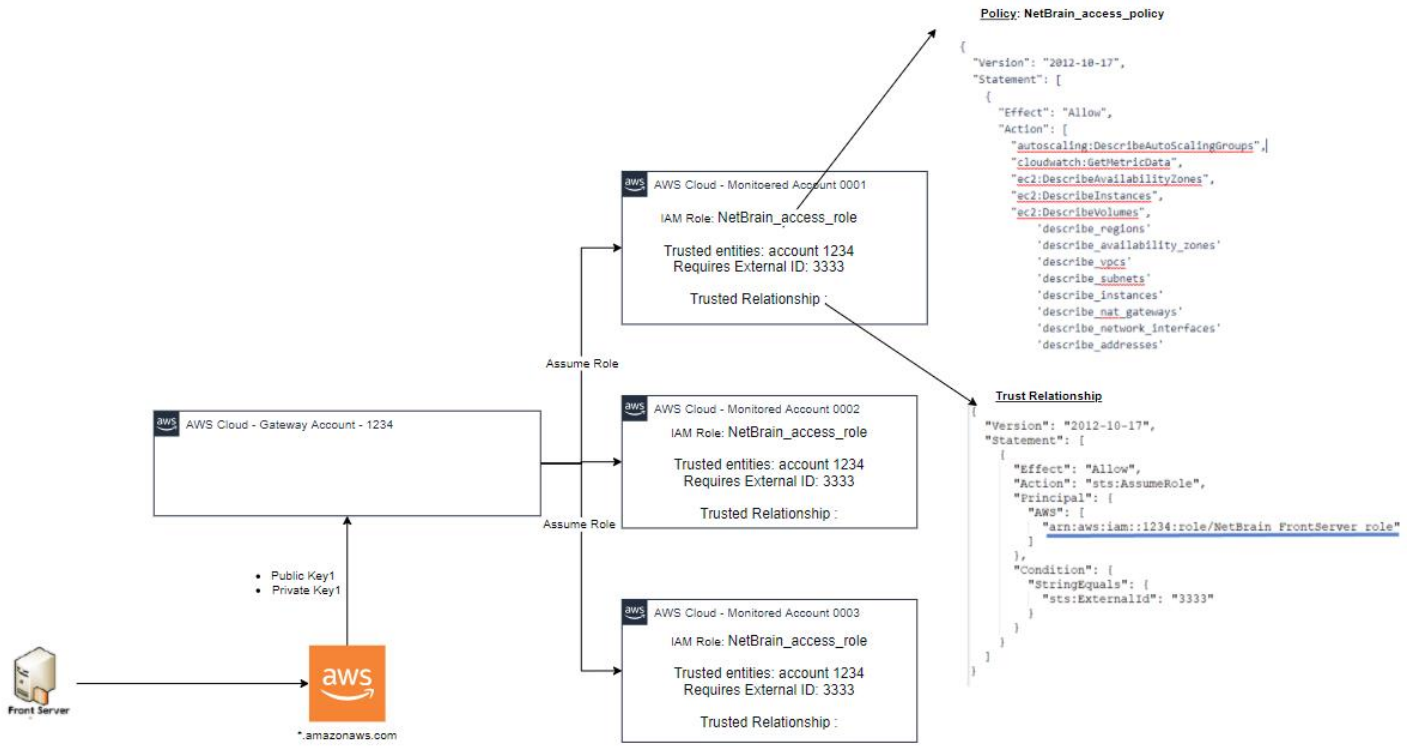
External ID	Yes	external ID configured for the role in the monitored account. As recommended by AWS, this is a mandatory field for security purposes.
Session Name	No	The Session Name will show in the CloudTrail log of the monitored account. It can be used for auditing purposes. Default to "netbrain_monitor" if not configured.

4. Click **Test** to verify that NetBrain system has access to the AWS account resources. If it fails, check if the roles and policies are configured properly.



4. Setting Up Combined Access

As shown in the diagram below, monitored accounts on the right-hand side are the accounts you will add to NetBrain for management purposes. You will need to configure the proper roles for these accounts to be accessed by the gateway account.



Compared to pure role-based access, the combined access gains access to the gateway account through key-based access, which gives you the flexibility to set up the Front Servers in any desired location.

Follow the steps below to set up the combined access:

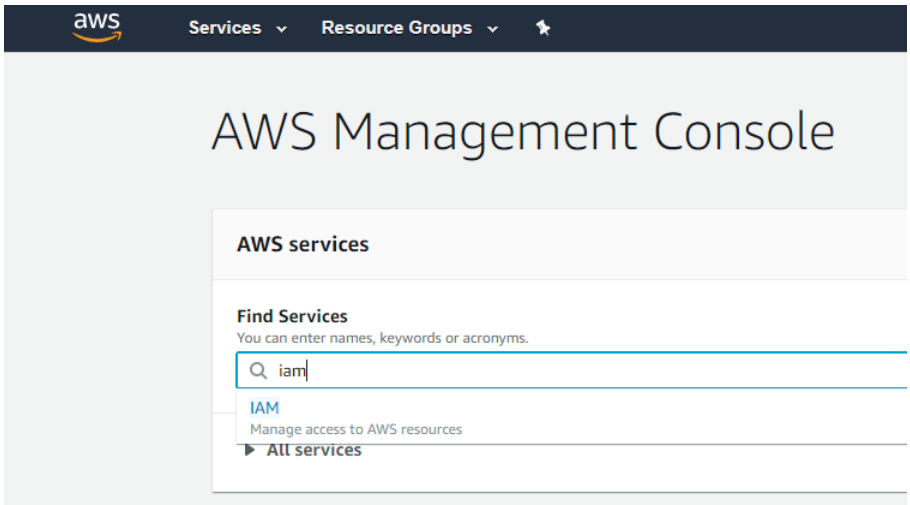
1. Creating AWS Access Policy and Role for Monitored Accounts
2. Creating Public/Secret Keys for Gateway Accounts
3. Configuring NetBrain System

4.1. Creating AWS Access Policy and Role for Monitored Accounts

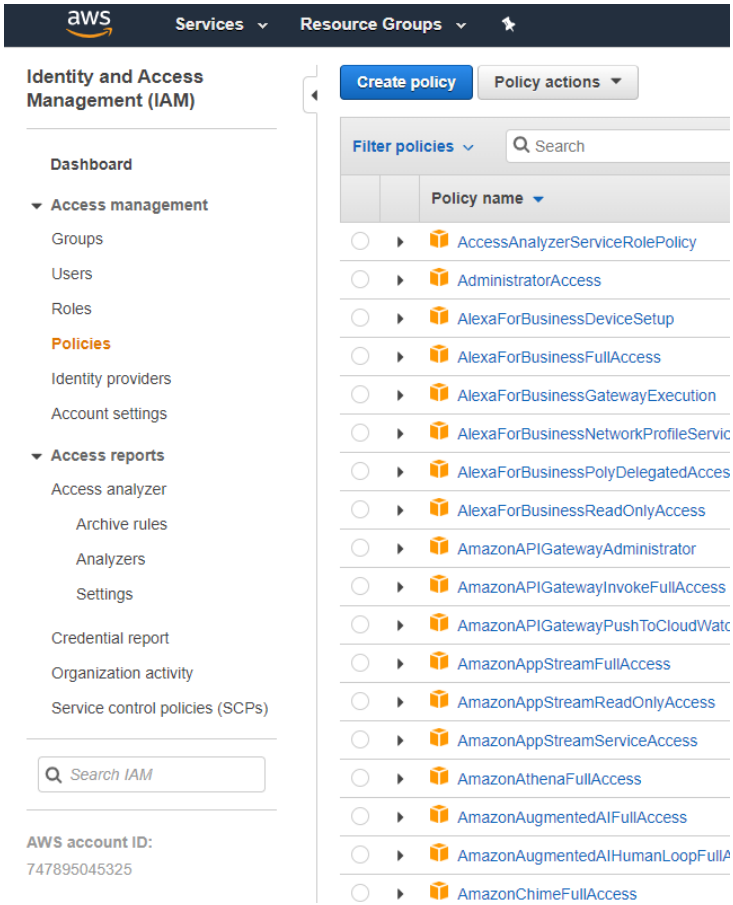
Refer to Creating AWS Access Policy and Role for Monitored Accounts.

4.2. Creating Public/Secret Keys for Gateway Accounts

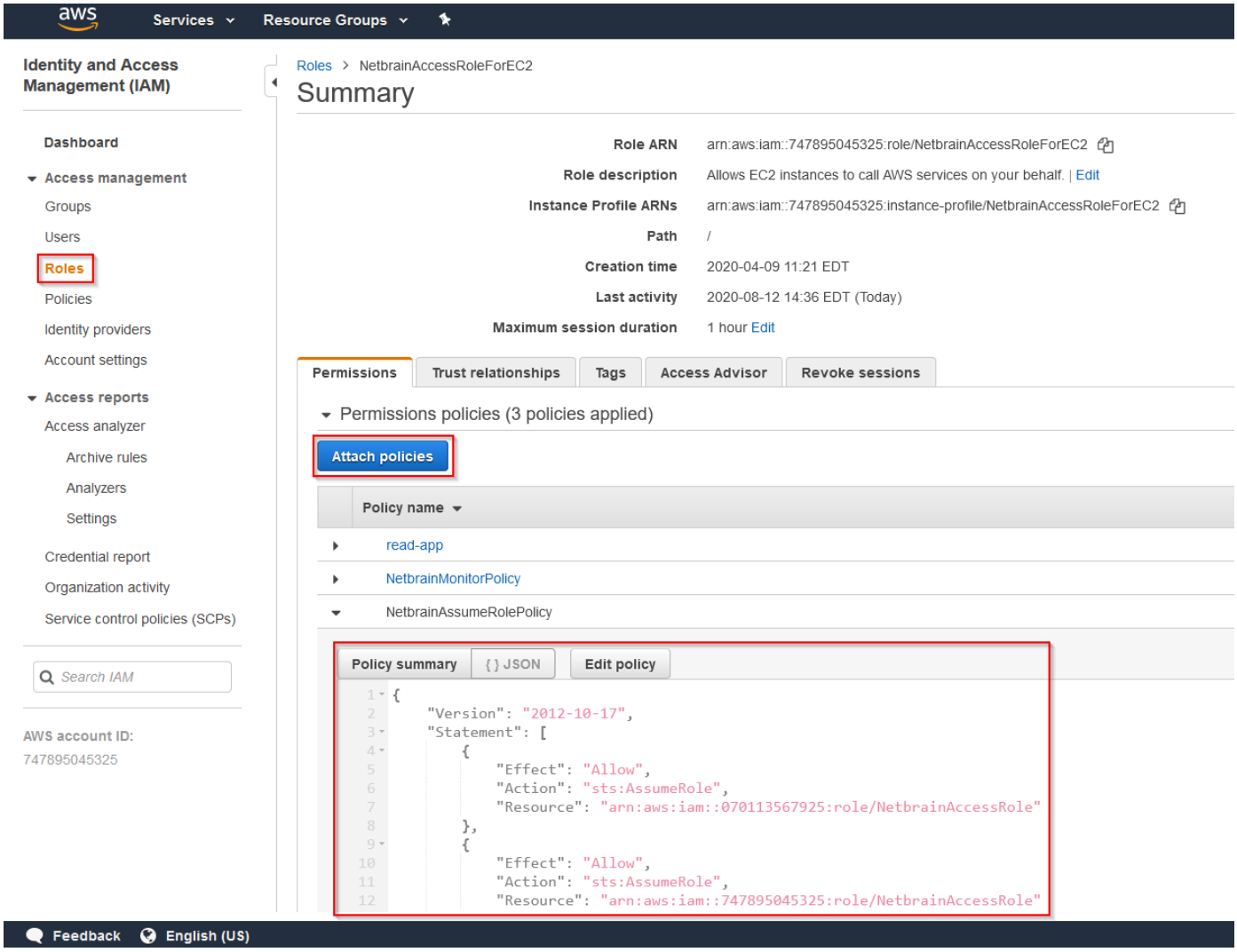
1. Go to **Identity and Access Management (IAM)** in your Amazon Console.



2. Go to **Policies** and click **Create policy**.



3. After successfully creating the role, you can open the role and attach an inline policy to allow the current role to assume NetbrainAccessRole in monitored accounts.



A Sample policy JSON is as follows.

> **Note:** Use the account ID to monitor your environment.

```
{
  "Statement": [
    {
      "Resource": [
        "arn:aws:iam::<12-digit monitored account number>:role/<role created in previous step (NetbrainAccessRole)>"
      ],
      "Action": [
        "sts:AssumeRole"
      ],
      "Effect": "Allow"
    }
  ],
  "Version": "2012-10-17"
```

```
}
```

4.3. Configuring NetBrain System

After you have set up the monitored accounts and gateway accounts, follow these steps to add the accounts to monitor:

1. On the **Domain Management** page, navigate to **Operations** > **Discover Settings > API Server Manager**.

2. In the **API Server Manager** configuration page, click **Add API Server** to add an API Server entry into the table for each account to be monitored.

3. Configure the parameters in the **Edit External API Server** window as follows:

   1) **API Source Type**: Select **Amazon AWS**.

   2) **Access Method**: Select **Role-based Access**.

   3) **Endpoints (Account ID)**: Enter the AWS account ID to be monitored.

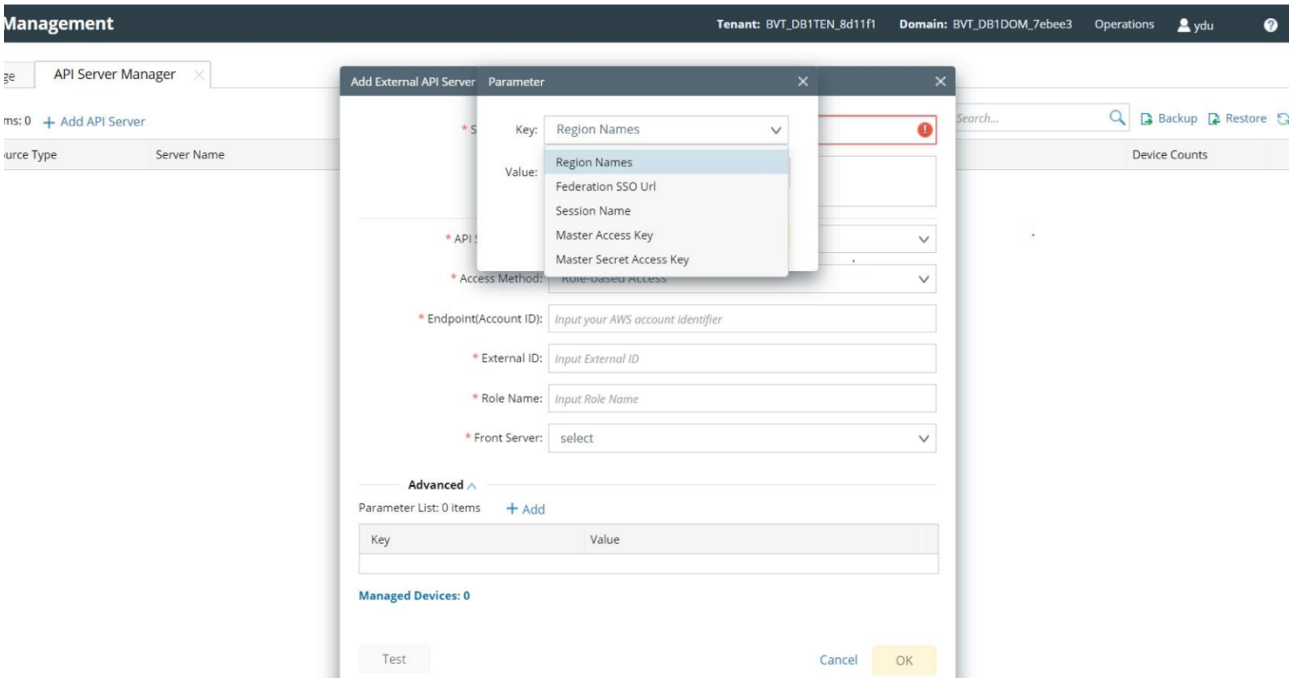
   4) **External Id**: Enter the External Id previously selected for the trust relationship in the AWS account to be monitored.

   5) **Role Name**: Enter the role name previously selected in the AWS account to be monitored.

4. In the **Advanced** section, click **+Add** and add the following keys for the gateway accounts:

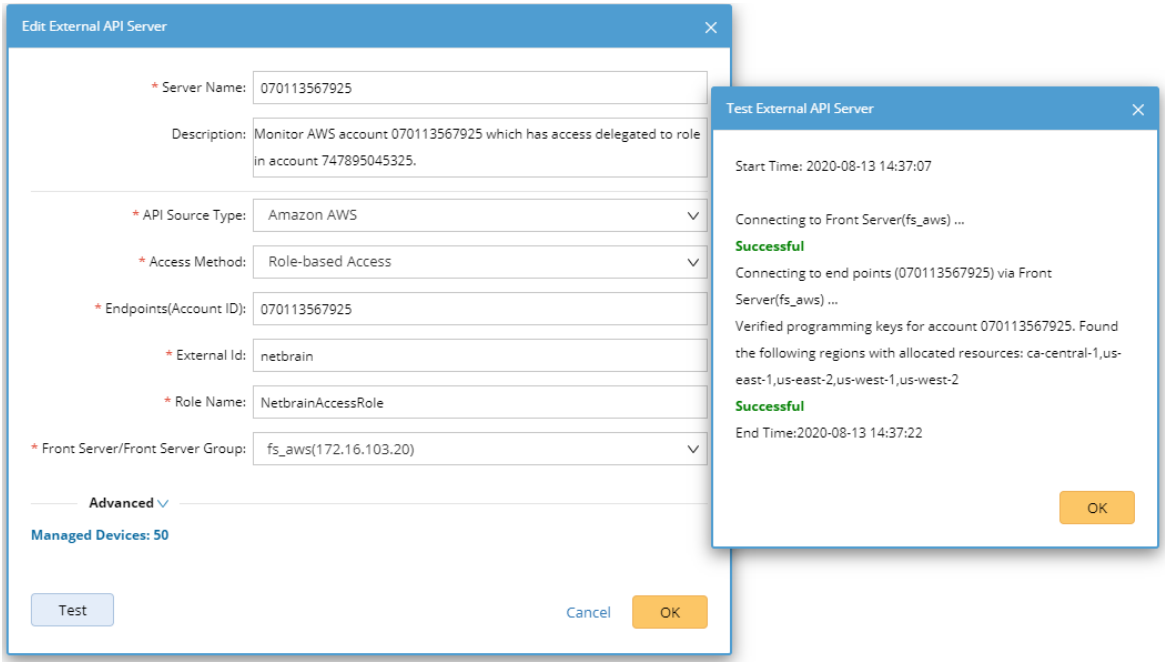
   - **Master Access Key**: This is the public key used to access the gateway account.

- **Master Secret Access Key**: This is the secret key used to access the monitored accounts.



5. Click **Test** in the **Add External API Server** window to verify the connection to the monitored accounts to ensure they are connected successfully.

6. Click **Test** in the **Edit External API Server** window to verify that NetBrain IE has access to the AWS account resources. If it fails, check if the roles and policies are configured properly.

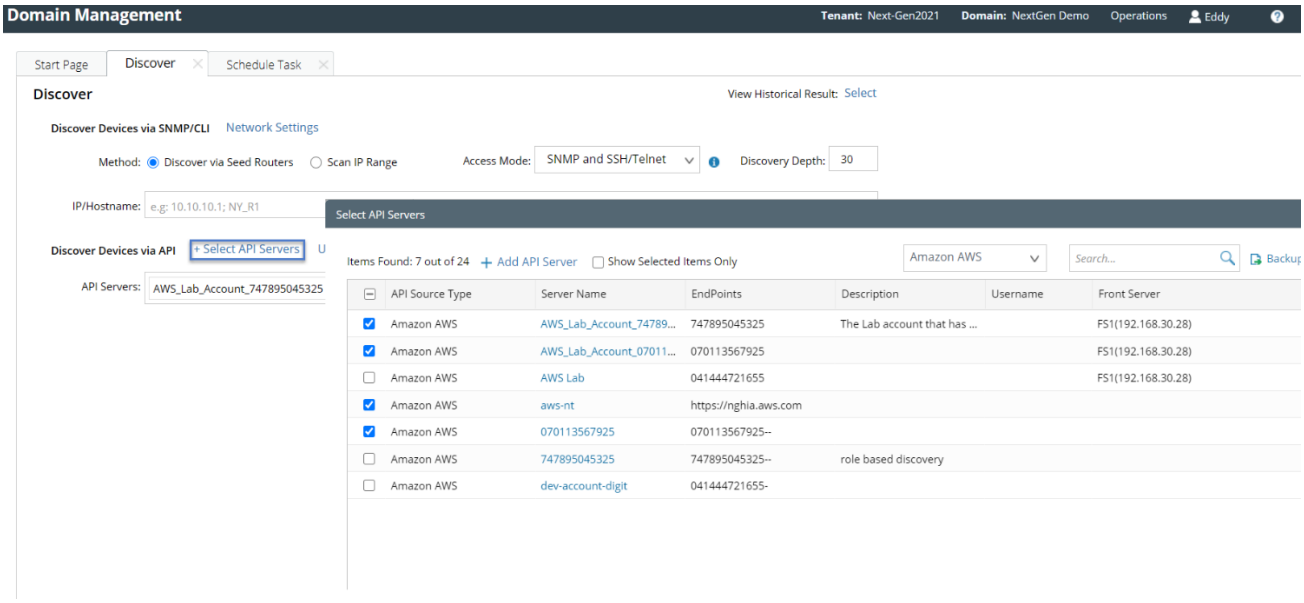


5. Discovering AWS Network in NetBrain Domain

Follow the steps below to discover the network data model in a NetBrain domain:

1. On the **Domain Management** page, select **Operations > Discover** from the quick access toolbar.

2. In the **Discover Devices via API** area, click **Select API Servers** to select the API servers you want to discover.



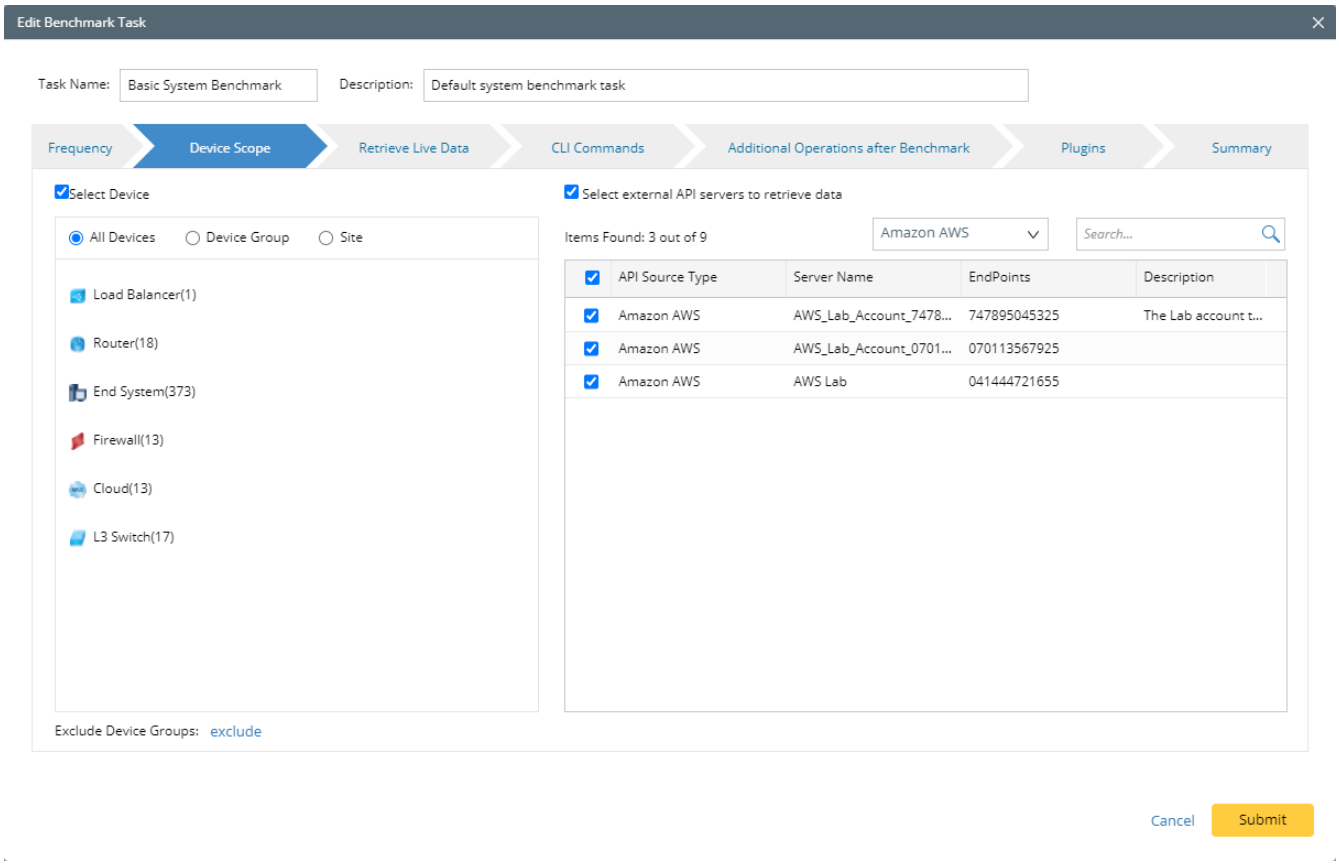
> **Note:** To build the data model correctly, NetBrain requires CLI+SNMP access to all virtual network appliances of each AWS VPC, including the customer gateway devices (CGW), virtual firewall instances, and virtual load-balancer instances.
>
> **Note:** To discover virtual appliances via SNMP/CLI, you can specify their management IP addresses in the discovery interface.

6. Auto-Updating AWS Data in NetBrain through Benchmark

The discovery only retrieves basic data of your AWS network and builds L3 topology. After the discovery, you need to execute a benchmark task to retrieve all data and build all components, including visual spaces and data views.

Example: Benchmark AWS in a NetBrain Domain.

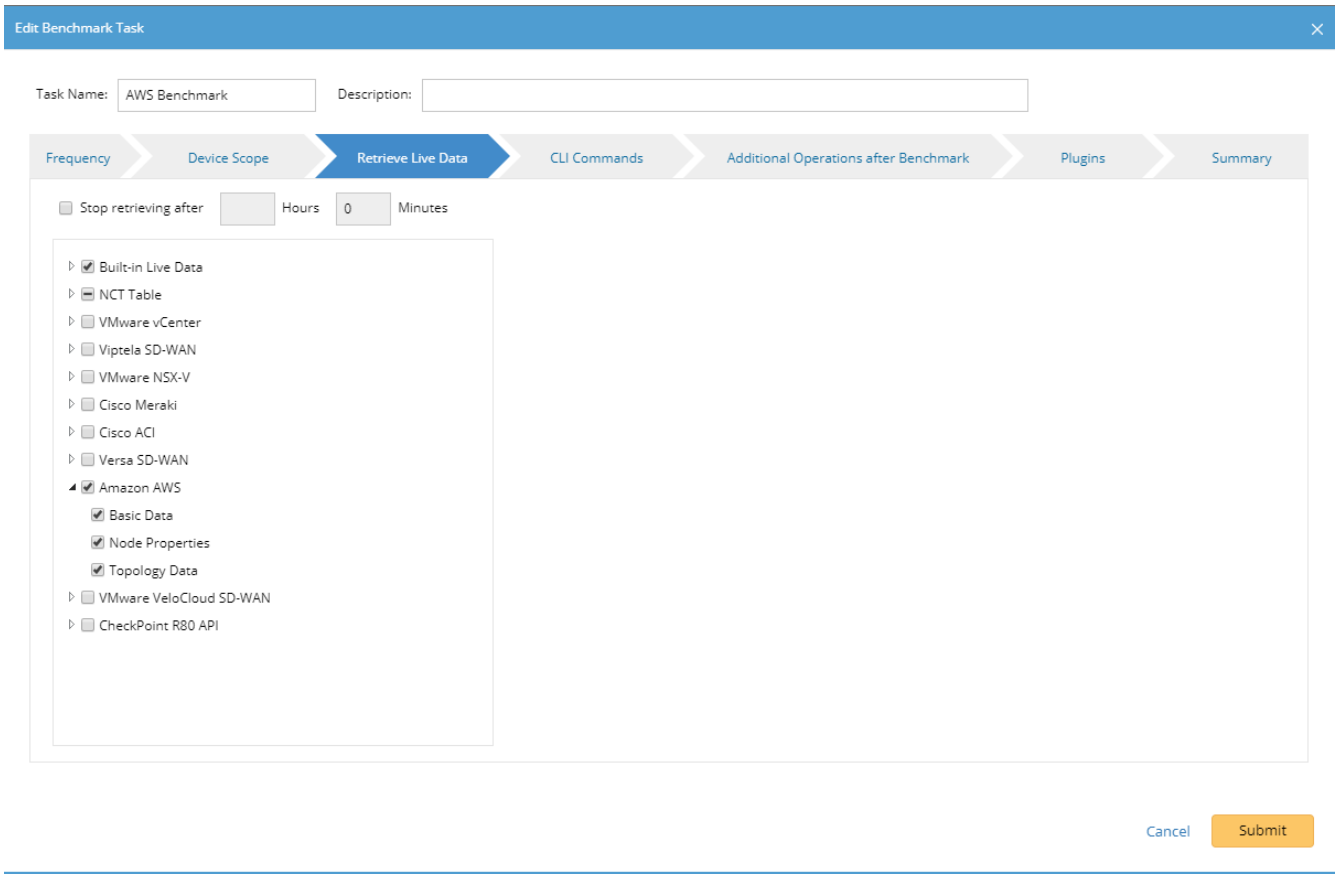
1. On the Start Page, click **Schedule Task**.
2. On the **Schedule Discovery/Benchmark** tab, click **+Add Benchmark Task**.
3. On the **Frequency** tab, define the task frequency.
4. On the **Device Scope** tab, check the **Select external API servers to retrieve data of SDN nodes** check box and select controllers.



> **Note:** As a best practice, we recommend re-using the "Basic System Benchmark" with a full benchmark task, where all devices are selected. This ensures that all AWS-connected physical or virtual devices are selected within the device scope.

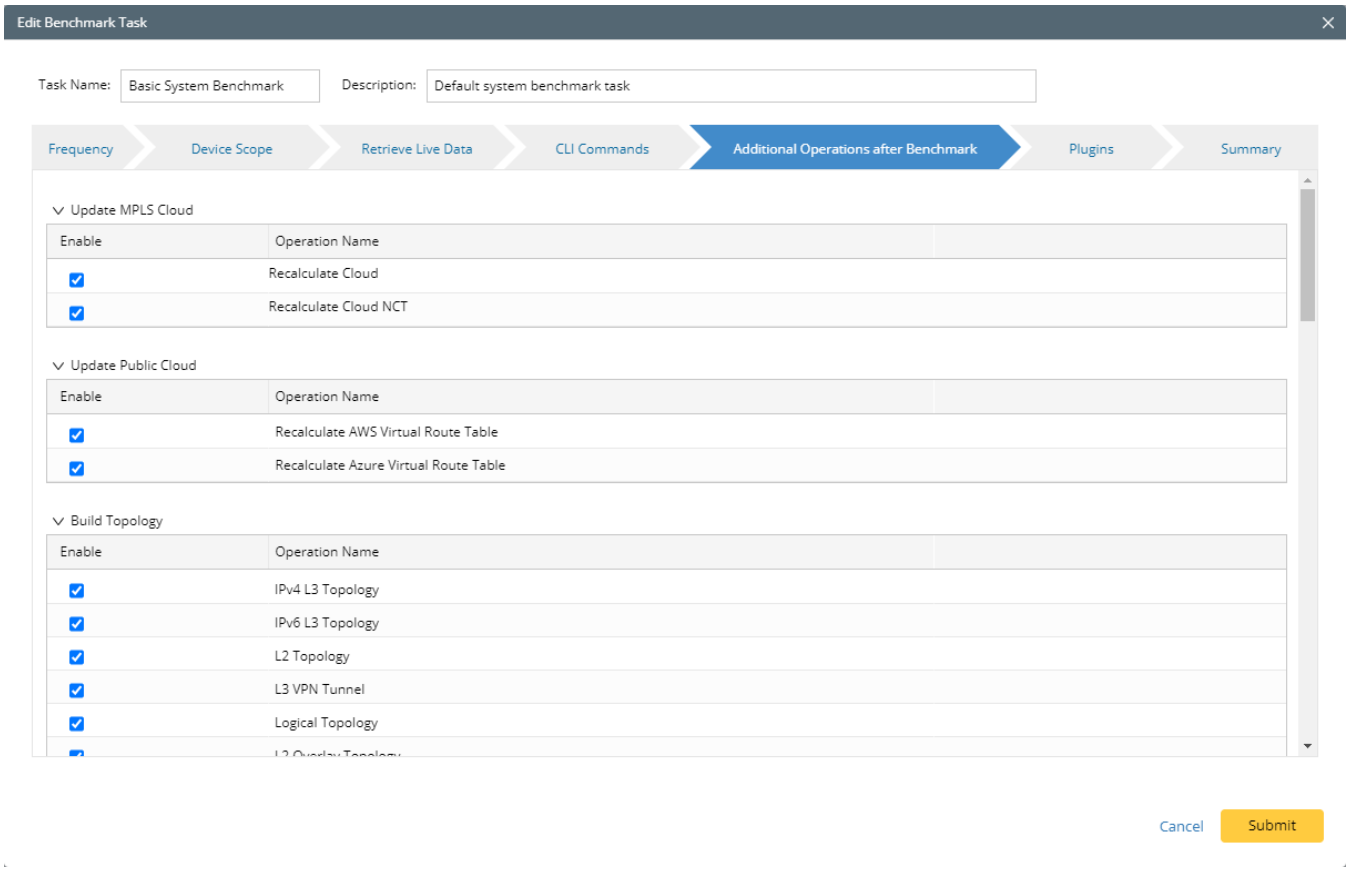
5. On the **Retrieve Live Data** tab, select the **Amazon AWS** check box, and make sure the following tables (under the NCT table) are selected:

   - AWS ENI Interface Table
   - AWS ELB Listener Table
   - AWS NAT Table
   - AWS Network ACL Table
   - AWS Security Group Table
   - AWS ELB Target Group Table
   - AWS Transit Gateway Attachments Table
   - AWS Transit Gateway Route Table
   - AWS VPC Peering Table
   - AWS PC Route Table



6. On the **Additional Operation After Benchmark** tab, select the following checkboxes:

- Update MPLS Cloud
- Update Public Cloud
- Update Build Topology

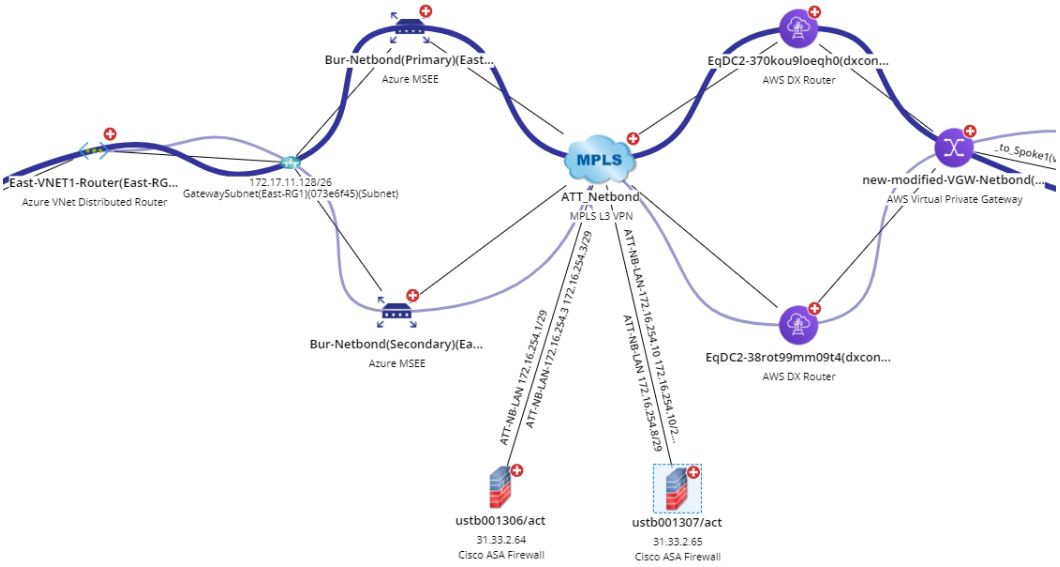


7. Click **Submit**.

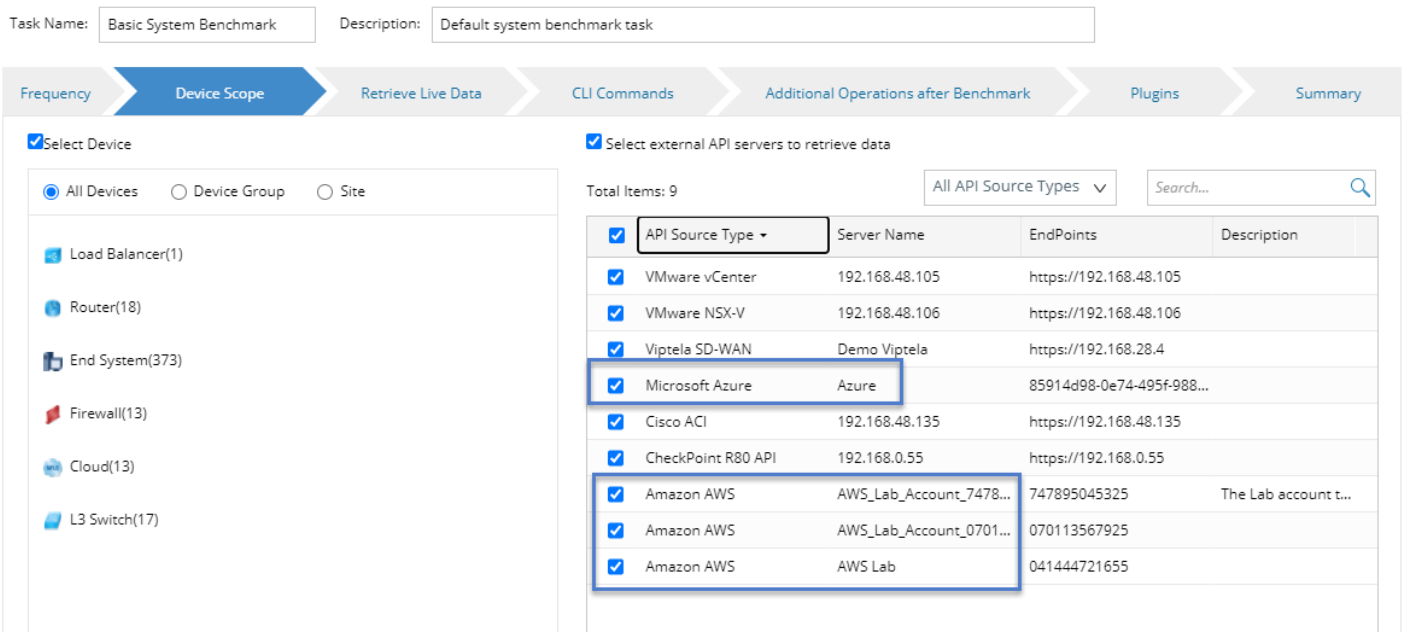
7. Working with Multi-cloud Environment

If your public cloud environment has multiple public cloud providers, you may want to discover the other public cloud providers, such as Azure and Google Cloud. Refer to their quick setup guides for details.

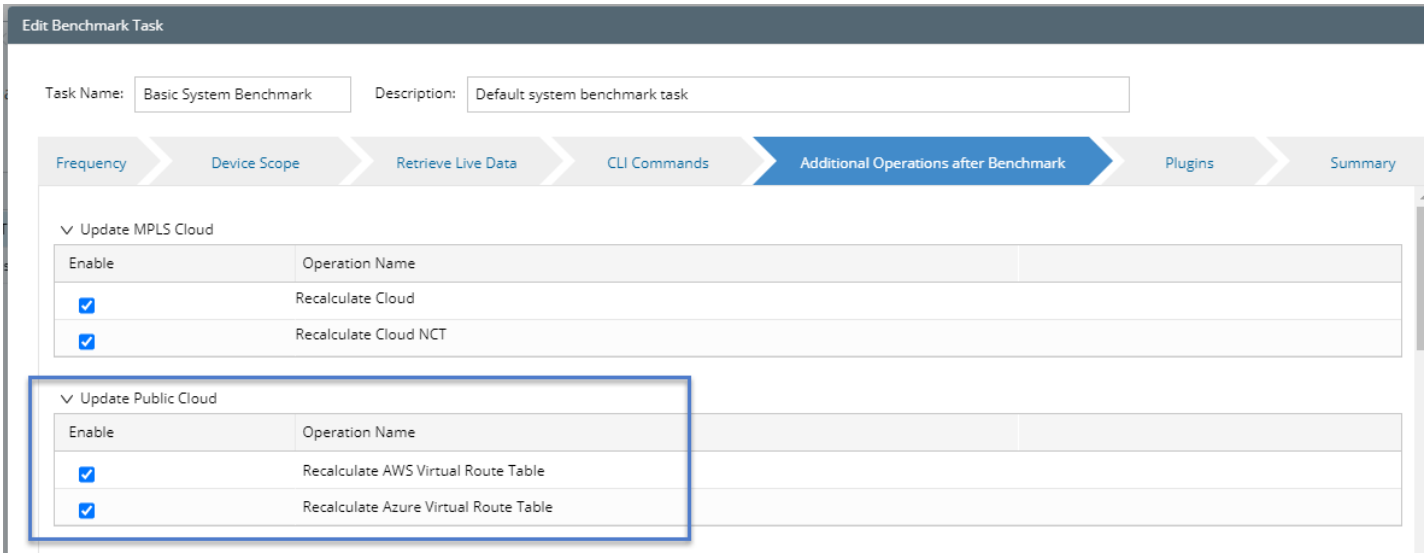
If the AWS and Azure networks are connected to your on-prem network via L3 VPN, you can use NetBrain to discover both of them. As shown in the diagram below, you need to make sure AWS and Azure are in the same benchmark task to get the entire public cloud data updated:



It is recommended to use one single benchmark task to retrieve all public cloud data. The screenshot below shows an example of retrieving the data from both AWS and Azure:



In the **Update Public Cloud** area of **Additional Operations after Benchmark** tab, make sure both **Recalculate AWS Virtual Route Table** and **Recalculate Azure Virtual Route Table** are selected.

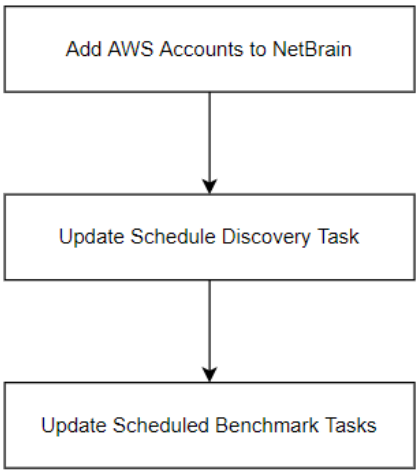


8. Using REST API to Manage AWS Data

If your organization has hundreds or even thousands of accounts, you can use the corresponding REST APIs to add these accounts to the system and manage your AWS accounts. This chapter illustrates the main flow and explains how to use these APIs.

For a complete list of APIs, refer to https://github.com/NetBrainAPI/NetBrain-REST-API-R10/tree/master/REST%20APIs%20Documentation/API%20Server%20Management.

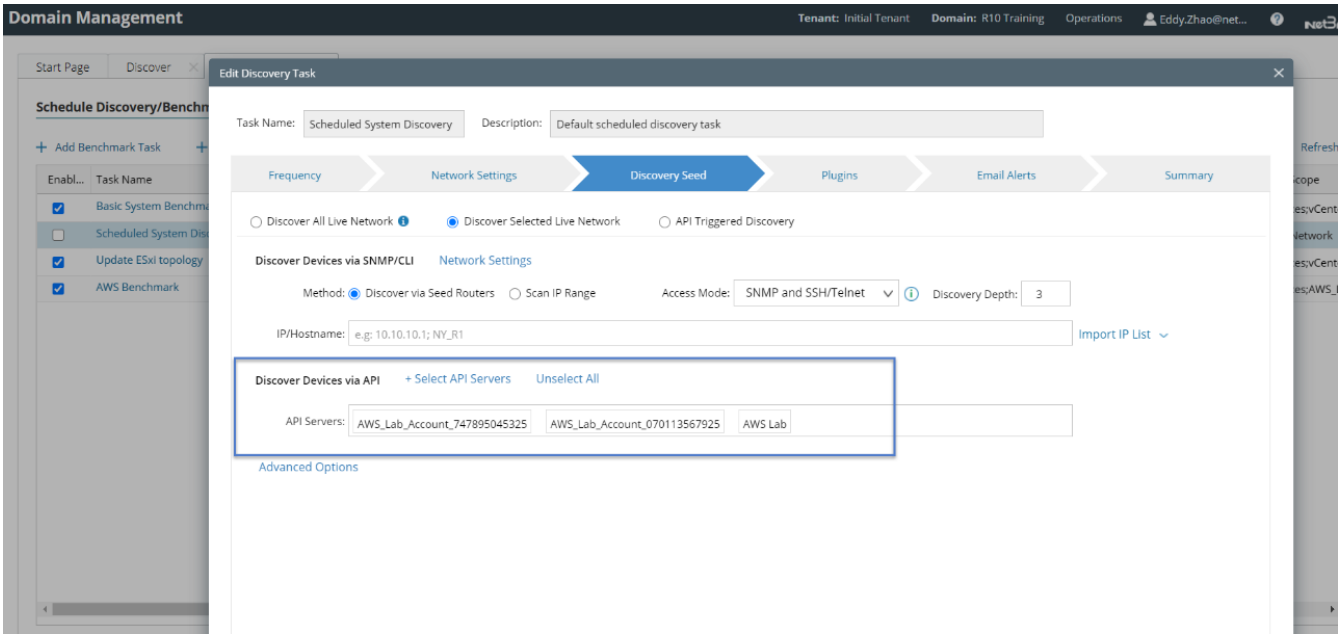
Onboarding New Accounts:



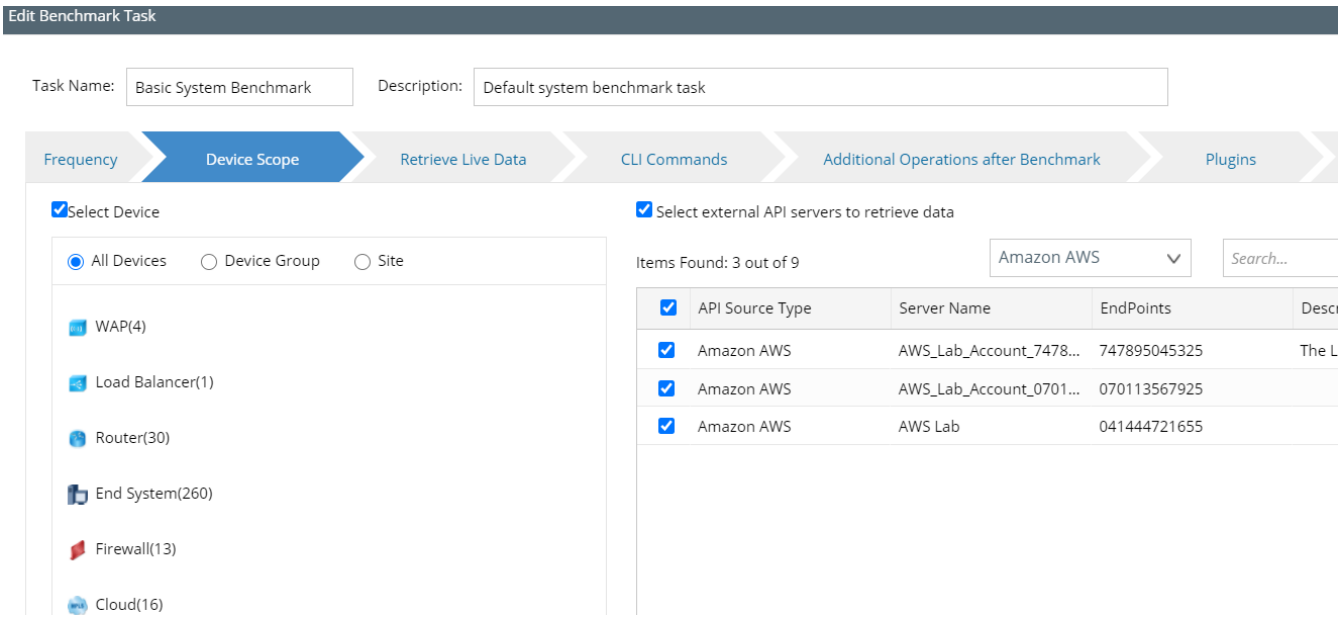
If you want to have the scripts integrated into your account onboarding process, you can use the REST APIs to perform the following tasks after adding the new accounts:

- **Add AWS Accounts to NetBrain**: You will need to define your strategy to choose what types of accounts to add to NetBrain, either by using the tag or OU (organizational unit) as a filter based on your preference.
- **Update Schedule Discovery Tasks**: After adding the AWS accounts into NetBrain, you will need to add these accounts into the scheduled discovery process.

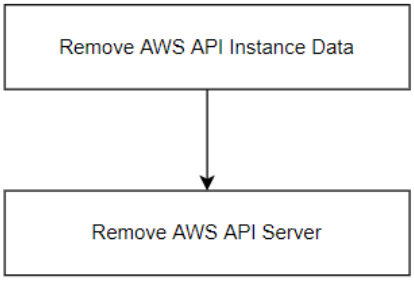
> **Note:** You only need to discover the new accounts once (when you add these new accounts to NetBrain). After the data of these accounts are discovered and initialized, you don't need to discover them for a second time. You can use the Rest API to query the discovery results (succeed or fail). If some accounts are discovered successfully, you could use the API to delete these accounts from the schedule discovery task.



- **Update Schedule Benchmark Task**: After the discovery process, the corresponding data for the AWS accounts will be added to the system. The system will then need to run the benchmark to update the AWS data. If you have selected certain AWS accounts for the discovery, you will need to add these newly added accounts to the benchmark scope, as shown in the screenshot below.

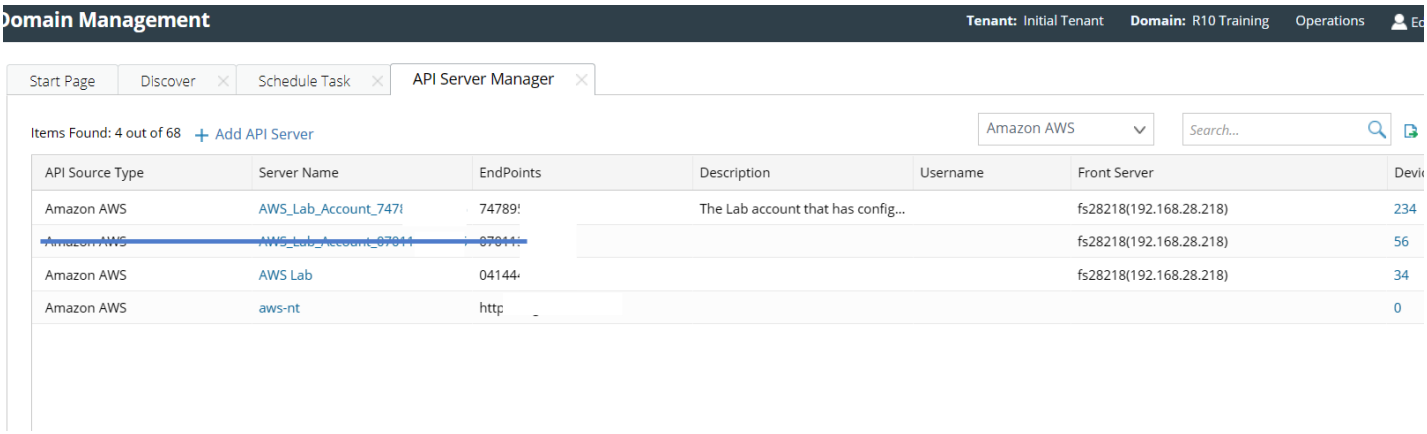


Offboarding Old Accounts:



When you want to remove some AWS accounts that are not in use, you can use the REST APIs to remove these accounts and data from NetBrain.

- **Remove AWS API Instance Data**: You will need to call this API to remove the AWS API instance data so that all the data for the current AWS API Server will be removed from the NetBrain system.
- **Remove AWS API Server**: After successfully removing the AWS API instance data, you can safely remove the AWS API server, so this server will no longer be shown in the API Server Manager.

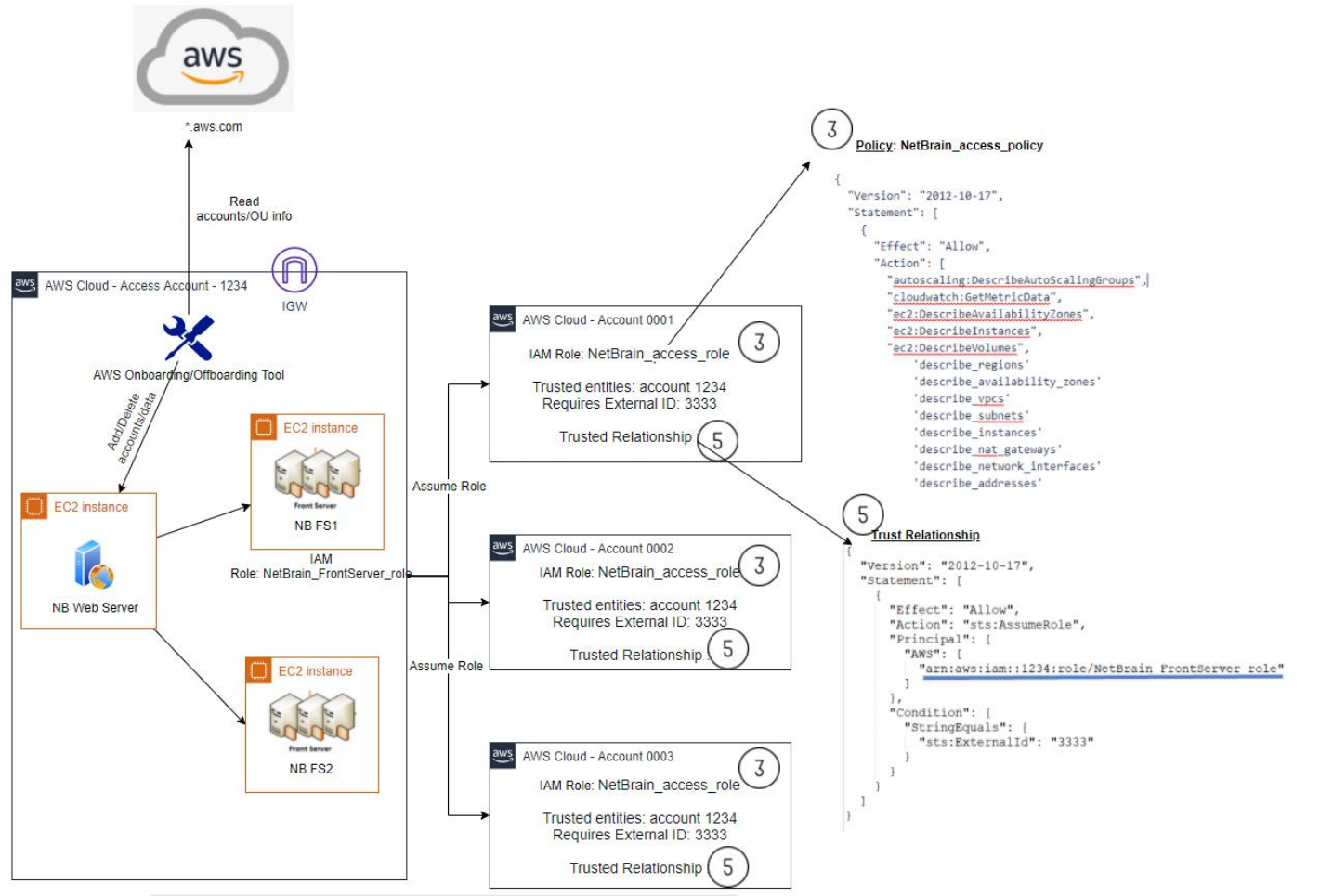


8.1. Integration with AWS Organization

Using REST API to Manage AWS Data explains how you can use the REST API to integrate with the NetBrain system and update the AWS data. Sometimes you need to create scripts with these APIs to complete complex tasks and integrate them into your account onboarding/offboarding process. Instead of creating the integration scripts, you can use the NetBrain onboarding/offboarding tool to integrate with your AWS organization. (AWS Organizations

helps you centrally manage and govern your environment as you grow and scale your AWS resources. Reference link: https://aws.amazon.com/organizations/.)

The architecture diagram is shown as follows:



The following requirements must be met to enable the proper function of the AWS onboarding/offboarding tool:

- The tool must have access to the AWS public endpoints to get the AWS organization data, and it can investigate the data to define what accounts can be added to NetBrain System.
- The tool must have access to the NetBrain web servers to use REST APIs defined in Using REST API to Manage AWS Data to update the AWS data.

> **Note:** You can contact NetBrain Support to help you deploy the tool based on your specific requirements.

Configure Access to NetBrain and your AWS Organization

You will need to configure the access to both NetBrain and your AWS organization in **config.YAML**:

```yaml
netbrain:
  base_url: "192.168.1.1"# note: it's not the desktop.html for web browser access.
  username: "admin"  # you need to use an user that has administrator role in Netbrain APP.
  password: "" # use the password associated with your username
  tenant: "AutoTestTenant" # Netbrain PSE can help to create the Tenant or you can do it by yourself.
  domain: "onboarding2" # Netbrain PSE can help to create the Domain or you can do it by yourself.
  front_servers:
    - "awswindowsfs"
  onboarding_tool_tag: Tag001 # add to description, NetBrain Onboarding Tool Tag[DO NOT DELETE]: sample_onboarding_tool_tag
aws_organizations:
  #access_key_id: "" # access key for master account to allow read access to accounts list in the organizations
  #secret_key: ""
  master_account_id: "635844821045" # master account id, for fs ec2 server to assum master account role
  master_access_role_name: "ListOrganizationRole2" # access role for master account to allow read access to accounts list in the organizations
  master_external_id: "netbrain"
  #mixed_mode_master_access_key_id: ""
  #mixed_mode_master_secret_key: ""
  access_role_name: "NetbrainAccessRole" # role name is member accounts to be assumed by Netbrain FrontServer for monitoring.
  external_id: "netbrain" # External ID required to assume the role.
  select_ous: # limit the OUs IDs to onboard. search the entire organizations if not specified.
    #- ou-1a2b
    #- ou-1a2c
  exclude_ous: # list the OUs IDs or sub OUs to be excluded from onboarding.
    #- ou-1a2b-34uvwxyz
  exclude_accounts: # list the accounts IDs to be excluded from onboarding.
    #- 1111111111111
    #- 2222222222222
  exclude_tags: # list the tags to specify which accounts to exclude.
    - Key: customer
      Value:
    - Key: purpose
      Value: sandbox

log_level: info # e.g. debug, info, warn, error
```

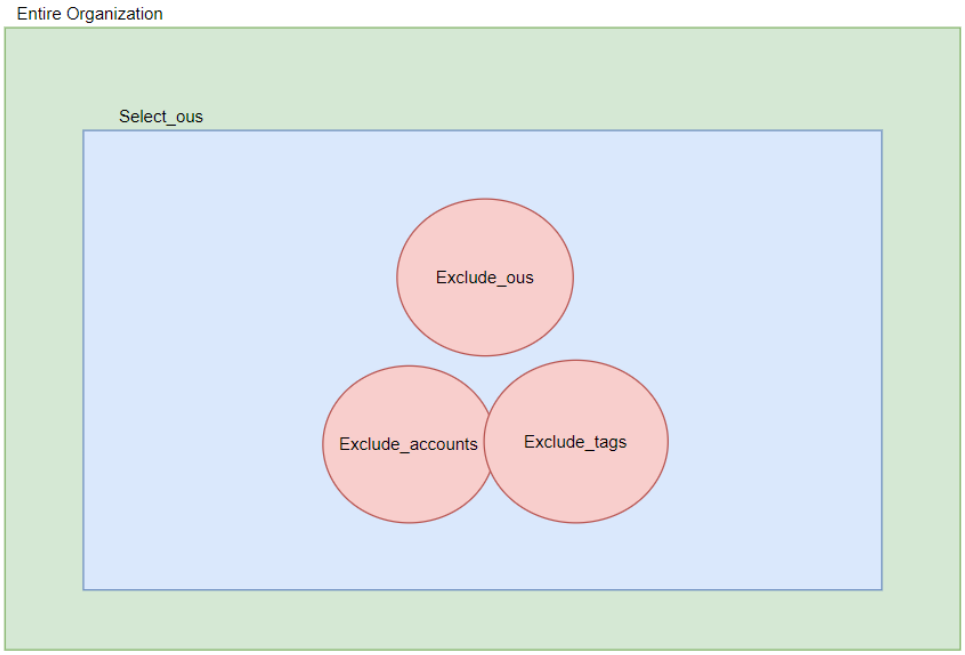
- **Access to NetBrain**: You must specify the NetBrain URL, username, password, tenant, domain, and the front server. Make sure the created user has domain management permission.
- **Access to AWS Organization**: You will need to specify the access method to the master accounts where the onboarding/offboarding tool can get the AWS organization info:
  - **Key-based Access**: Using the key-based access to configure the access key/secret key to access the AWS master account.
  - **Role-based Access**: Using the role-based access so the onboarding/offboarding tool can access the AWS master account.

You can use the combination of OU, accounts, and tag as the filter to only onboard specific accounts into the NetBrain system. The following rules should be obeyed:

1) **Select_ous**: Define the search scope and the function scope of excelude_ous, exclude_accounts, and exclude_tags. In most cases, select the OUs you want to onboard and do not leave them empty.
2) **Exclude_ous**: Define what OUs or subOUs you want to exclude.
3) **Exclude_accounts**: Define specific accounts you want to exclude.

4) **Exclude_tags**: Define tags so accounts with these tags won't be included. In most cases, you may want to exclude sandbox accounts or other types of accounts that you don't want to add to NetBrain.

The following diagram gives an overview of how the various conditions work together. The green color represents the entire organization tree. From there, you can define the select_ou to specify certain OUs you want to add to NetBrain. Within the selected OU group, you can use different types of excluding flags to exclude certain ous/accounts/tags. The final accounts added to NetBrain are the area shown in blue.



<u>Access to the Master Accounts</u>:

To access the master accounts and list all accounts within the current organization, you must configure the correct access policy. We have attached different policies for you to choose from based on your security considerations.

If your security team permits, you can use the board policy, which allows access to the entire organization:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "organizations:Describe*",
                "organizations:List*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Or, if you want more specific policies, you can use the following detailed policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "organizations:DescribeOrganization",
                "organizations:ListRoots",
                "organizations:ListTagsForResource",
                "organizations:ListOrganizationsUnitsForParent",
                "organizations:ListAccountsForParent"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

There are two ways to access the master accounts: key-based access or role-based access:

Key-based access to the Master Account

If you use the key-based access to access the master account, list organization information, select the access method as key-based access and configure the access key/secret key to the master accounts, NetBrain will access the master account and list the organization information.

```
aws_organizations:
  access_key_id: "key_id" # access key for master account to allow read access to accounts list in the organizations
  secret_key: "secret_key"
  #master_account_id: "635844821045" # master account id, for fs ec2 server to assum master account role
  #master_access_role_name: "ListOrganizationRole2" # access role for master account to allow read access to accounts list in the organizations
  #master_external_id: "netbrain"
  #mixed_mode_master_access_key_id: ""
  #mixed_mode_master_secret_key: ""
  access_role_name: "NetbrainAccessRole" # role name is member accounts to be assumed by Netbrain FrontServer for monitoring.
  external_id: "netbrain" # External ID required to assume the role.
  select_ous: # limit the OUs IDs to onboard. search the entire organizations if not specified.
    #- ou-1a2b
    #- ou-1a2c
```

Role-based Access to the Master Account

If you use role-based access to access the master account, list organization information, select the access method as role-based access and configure the role and other details, NetBrain will access the master account and list the organization information.