AWS Migration Hub Orchestrator

User Guide

AWS Migration Hub Orchestrator: User Guide

Table of Contents

What is AWS Migration Hub Orchestrator?

AWS Migration Hub Orchestrator simplifies and automates the migration of servers and enterprise applications to AWS. It provides a single location to run and track your migrations.

With Migration Hub Orchestrator, you can migrate SAP NetWeaver based applications, such as S/4HANA, BW4HANA, ECC on HANA, and others to AWS and rehost supported custom applications to Amazon EC2. Migration Hub Orchestrator offers templates to create a migration workflow that can be customized to fit your unique migration requirements. Migration Hub Orchestrator automates the steps in your chosen workflow and displays the status of migration.

You can access Migration Hub Orchestrator from the https://console.aws.amazon.com/migrationhub/ or from the AWS Command Line Interface.

Related services

If you are new to Migration Hub, you can refer to the following guides.

- Application Discovery Service
- AWS Application Migration Service
- AWS Launch Wizard for SAP

Pricing

AWS Migration Hub Orchestrator is available to you at no additional cost. You only pay for the AWS resources that you provision for migrations.

Setting up

Sign up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all AWS services, including Migration Hub Orchestrator. You are charged only for the services that you use.

If you already have an AWS account, skip this step.

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

   When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to an administrative user, and use only the root user to perform tasks that require root user access.

Create an IAM user

By default, an administrator account inherits all of the policies that are required to access Migration Hub Orchestrator. To create an **administrator user**, follow the steps in Creating your first IAM admin user and user group.

To create a **non-administrative** IAM user for use with Migration Hub Orchestrator, we recommend that you create these IAM users:

- To access the console, create a user with both the `AWSMigrationHubFullAccess` and the `AWSMigrationHubOrchestratorConsoleFullAccess` managed policies attached.
- To enable the Migration Hub Orchestrator plugin to communicate with your servers, create a user with the `AWSMigrationHubOrchestratorPlugin` managed policy attached.
- To enable the instances to communicate with the Migration Hub Orchestrator plugin, create a user with the `AWSMigrationHubOrchestratorInstanceRolePolicy` managed policy attached.

Alternatively, you can create one user with all the managed policies attached. For more information, see AWS managed policies for Migration Hub Orchestrator.

When creating non-administrative IAM users, follow the Grant least privilege security best practice and grant users minimum permissions.

To create a non-administrator IAM user to use with Migration Hub Orchestrator

1. In AWS Management Console, navigate to the IAM console.
2. Follow the instructions in Creating an IAM user in your AWS account.

While following the instructions, ensure that you:

- Select both **Programmatic access** and **AWS Management Console access** as the type of access.
- Choose the option to **Attach existing policies to user directly** on the **Set permission** page. Then, choose the managed IAM policy **AWSMigrationHubFullAccess**, **AWSMigrationHubOrchestratorConsoleFullAccess**, or **AWSMigrationHubOrchestratorPlugin** from the list of policies.
- Follow the guidance in the **Important** note about saving the user's new access key ID and secret access key in a safe and secure place.

Orchestrate migrations with Migration Hub Orchestrator

You can simplify and automate the migration of your on-premises servers and applications to AWS Cloud using Migration Hub Orchestrator.

You can use templates to create migration workflows in Migration Hub Orchestrator. Define the servers and applications that you want to migrate to AWS, configure the Migration Hub Orchestrator plugin on-premises, and then create a migration workflow using one of the following templates.

Note
The Migration Hub Orchestrator plugin must be able to communicate with the source and target environments to orchestrate and automate migrations. The version of the plugin that is deployed in vCenter supports VMware vCenter Server 6.0, 6.5, 6.7 and 7.0.

Define applications

Define applications by adding a data source and grouping the servers as applications.

- the section called “Add data source” (p. 4)
- the section called “Group servers” (p. 4)

Add data source

Get metadata about the source servers and applications that you want to migrate to AWS. You can use one of the following methods to collect the data.

- **Migration Hub import** – Import information about your on-premises servers and applications into Migration Hub. For more information, see Migration Hub Import in the *Application Discovery Service User Guide*.
- **AWS Agentless Discovery Connector** – The Discovery Connector is a VMware appliance that collects information about VMware virtual machines (VMs). For more information, see AWS Agentless Discovery Connector in the *Application Discovery Service User Guide*.
- **AWS Application Discovery Agent** – The Discovery Agent is AWS software that you install on your on-premises servers and VMs to capture system information, as well as information about the network connections between systems. For more information, see AWS Application Discovery Agent in the *Application Discovery Service User Guide*.

Group servers

To use Migration Hub Orchestrator, you must group servers as applications.

1. In AWS Migration Hub console, select **Discover**, **Servers**.
2. In the servers list, select each server that you want to group into a new or existing application.
3. To create your application, or add to an existing one, choose **Group as application**.

4. In the **Group as application** dialog box, choose **Group as a new application** or **Add to an existing application**.
5. Select **Group**.

To view and edit your applications in the AWS Migration Hub console, go to **Discover** > **Servers**.

Download and deploy the Migration Hub Orchestrator plugin

The Migration Hub Orchestrator plugin is a virtual appliance that you can install in your on-premises VMware environment. To deploy the plugin as a virtual machine (VM) in your VMware environment, download the plugin Open Virtualization Archive (OVA) file.

1. Sign in to the https://console.aws.amazon.com/migrationhub/.
2. In the left navigation pane, choose **Orchestrate**.
3. On the **Migration Hub Orchestrator** page, choose **Download plugin**.
4. After the plugin is downloaded to your on-premises VMware environment, you can deploy it in vCenter. Sign in to vCenter as a VMware administrator.

   We recommend at least 8 GB of RAM and at least 4 CPUsfor the VM.
5. Deploy the OVA file that you downloaded. The OVA file includes the plugin and a CLI that can be used to access the Migration Hub Orchestrator API.
6. Sign in to the plugin using an SSH client.

   ```
   ssh ec2-user@PluginIPAddress
   ```

   When prompted for a password, enter the default password, **plugin@123**. You must change your password when you first sign in.
7. See Configure plugin to configure the Migration Hub Orchestrator plugin.

Tip
If you would like to use the plugin for multiple VMsyou can export the OVA file after you configure it, and import it to your desired source VM.

Configure the Migration Hub Orchestrator plugin

To configure the Migration Hub Orchestrator plugin using **plugin setup** commands, create a bash shell session in the plugin Docker container using the following command.

```
docker exec -it mhub-orchestrator-plugin bash
```

The **plugin setup** command runs all of the following commands in succession, but you can also run them individually:

- **plugin setup --aws-configurations**
- **plugin setup --vcenter-configurations**
- **plugin setup --remote-server-configurations**

Run the following command to set up all of the plugin configurations at the same time. Then, enter the information for AWS configurations, vCenter configurations, and remote server configurations.

```
plugin setup
```

Topics

Set up AWS configurations

Set up AWS configurations using the `plugin setup` command or the `plugin setup --aws-configurations` command.

1. Enter **Y** for yes to **Have you setup IAM permissions...**. You set up these permissions when you created an IAM user to access the plugin using the `AWSMigrationHubOrchestratorPlugin` managed policy following the steps in Setting up.
2. Enter the IAM profile that you created in the Migration Hub Orchestrator plugin using the following command.

   ```
   aws configure --profile <profile-name>
   ```
3. Enter your `access_key` and `secret_key` from the AWS account that has the IAM user that you created to access the plugin.
4. Enter a Region. For example, `us-west-2`. Choose a Region that suits your needs from the Regions that Migration Hub Orchestrator uses. For a list of these Regions, see Migration Hub Orchestrator endpoints in the *AWS General Reference*.
5. Enter **Y** for yes to **Upload plugin related metrics to Migration Hub Orchestrator?** Metrics data helps AWS to provide you with support.
6. Enter **Y** for yes to **Upload plugin related logs to Migration Hub Orchestrator?** Log data helps AWS to provide you with support.

Your configuration setup may look similar to this example.

```
plugin setup --aws-configurations
Have you setup IAM permissions in your AWS account as per the user guide? [Y/N]: Y
IAM Profile name: <profile-name>
Upload plugin related metrics to Migration Hub Orchestrator? By default plugin will upload
 metrics. [Y/N]: Y
Upload plugin related logs to Migration Hub Orchestrator? By default plugin will upload
 logs. [Y/N]: Y
Plugin configurations are saved successfully
Start registering plugin
Start registering plugin
Plugin is registered successfully.
```

Set up vCenter configurations

Set up vCenter configurations using the `plugin setup` command or the `plugin setup --vcenter-configurations` command.

1. Enter **Y** or **N** to **Would you like to authenticate using VMware vCenter credentials** based on your preference.

   Note
   Authenticating using VMware vCenter credentials requires that VMware tools are installed on the target servers.

   Enter the **Host Url**, which can be the vCenter IP address or the URL. Then, enter the **Username** and **Password** for VMware vCenter.
2. Enter **Y** for yes to **Do you have Windows machines managed by VMware vCenter** if you want to configure Windows servers. Then, enter the **Username** and **Password** for Windows.

   Note
   If your Windows Remote Server belongs to an Active Directory domain, you must enter the user name as *domain-name**username* when using the CLI to provide source server configurations. For example, if the name of your domain is exampledomain and your user name is Administrator, then the user name you enter in the CLI is **exampledomain \Administrator**.
3. Enter **Y** for yes to **Setup for Linux using VMware vCenter** if you want to configure Linux servers. Then, enter the **Username** and **Password** for Linux.
4. Enter **Y** for yes to the **Would you like to setup credentials for servers outside vCenter using NTLM for Windows** and **SSH/Cert based for Linux** questions if you want to set up source server credentials for servers outside of vCenter.
5. For **Would you like to use the same Windows credentials used during vCenter setup**, enter **Y** for yes if the credentials for the Windows machines that are managed outside of vCenter are the same as the credentials provided when configuring credentials for vCenter Windows machines. Otherwise, enter **N** for no.

   If you answer **Y** for yes, the following questions are asked.

   a. Enter **Y** for yes to **Are you okay with the plugin accepting and locally storing server certificates on your behalf during first interaction with windows servers?**.
   b. Enter **1** for **Enter your options** if you want to configure SSH authentication.

      If you choose to use SSH authentication, you must copy the generated key credentials to your Linux servers. For more information, see Set up key-based authentication on Linux servers (p. 9).

Your configuration setup may look similar to this example.

```
Start setting up vCenter configurations for remote execution
Note: authenticating using VMware vCenter credentials requires VMware tools to be installed
 on the target servers
Would you like to authenticate using VMware vCenter credentials? [Y/N]: Y
Host Url for VMware vCenter: host-url
Username for VMware vCenter: username
Password for VMware vCenter:
Successfully stored vCenter credentials...
Setup for Windows using VMware vCenter? [Y/N]: Y
Username for Windows: username
Password for Windows:
Successfully stored vCenter windows credentials...
Setup for Linux using VMware vCenter? [Y/N]: Y
Username for Linux: username
Password for Linux:
```

```
Successfully stored vCenter linux credentials...
Would you like to setup credentials for servers outside vCenter using NTLM for windows and
 SSH/Cert based for linux? [Y/N]: Y
Would you like to use the same Windows credentials used during vCenter setup? [Y/N]: Y
Are you okay with plugin accepting and locally storing server certificates on your behalf
 during first interaction with windows servers? These certificates will be used by plugin
 for secure communication with windows servers [Y/N]:Y
Successfully stored windows server credentials...
Please note that all windows server certificates are stored in directory /opt/amazon/mhub-
orchestrator-plugin/remote-auth/windows/certs

Please note the IP address of the plugin and run the script specified in the user
 documentation on all the windows servers in your inventory
Would you like to setup credentials for servers not managed by vCenter using SSH/Cert based
 for Linux? [Y/N]: Y
Choose one of the following options for remote authentication:
1. SSH based authentication
2. Certificate based authentication
Enter your options [1-2]: 1
Would you like to use the same Linux credentials used during vCenter setup? [Y/N]: Y
Generating SSH key on this machine...
SSH key pair path: /opt/amazon/mhub-orchestrator-plugin/remote-auth/linux/keys/
id_rsa_assessment
Please add the public key "id_rsa_assessment.pub" to the "$HOME/.ssh/authorized_keys" file
 in your remote machines.
Your Linux remote server configurations are saved successfully.
```

Set up source server configurations

Set up source server configurations using the `plugin setup` command or the `plugin setup --remote-server-configurations` command.

1. Enter **Y** for yes to **Would you like to setup credentials for servers not managed by vCenter using NLTM for Windows** if you want to configure Windows servers. Enter the **Username** and **Password** for WinRM.

   Note
   If your Windows Remote Server belongs to an Active Directory domain, you must enter the user name as *`domain-name`**`username`* when using the CLI to provide source server configurations. For example, if the name of your domain is exampledomain and your user name is Administrator, then the user name you enter in the CLI is **exampledomain\Administrator**.

   Enter **Y** for yes to **Are you okay with plugin accepting and locally storing server certificates on your behalf during first interaction with windows servers?**. Windows Server certificates are stored in the directory `/opt/amazon/mhub-orchestrator-plugin/remote-auth/windows/certs`. You must copy the generated server credentials to your Windows servers. For more information, see Set up the source server configuration on Windows servers (p. 10).
2. Enter **Y** for yes to **Setup for Linux using SSH or Cert** if you want to configure Linux servers.
3. Enter **1** for **Enter your options** if you want to configure for SSH key based authentication. If you choose to use SSH authentication, you must copy the generated key credentials to your Linux servers. For more information, see Set up key-based authentication on Linux servers (p. 9).
4. Enter **2** for **Enter your options** if you want to configure for certificate-based authentication. For information about setting up certificate-based authentication, see Set up certificate-based authentication on Linux servers (p. 10).

Your configuration setup may look similar to this example.

```
Setting up target server for remote execution
```

```
Would you like to setup credentials for servers not managed by vCenter using NLTM for
 Windows [Y/N]: Y
Username for WinRM: username //Enter domain-name\username, if the server is in AD domain
Password for WinRM: password
Are you okay with plugin accepting and locally storing server certificates on your behalf
 during first interaction with windows servers? These certificates will be used by plugin
 for secure communication with windows servers [Y/N]: Y
Successfully stored windows server credentials...
Please note that all windows server certificates are stored in directory /opt/amazon/mhub-
orchestrator-plugin/remote-auth/windows/certs

Please note the IP address of the plugin and run the script specified in the user
 documentation on all the windows servers in your inventory
Would you like to setup credentials for servers not managed by vCenter using SSH/Cert based
 for Linux? [Y/N]: Y
Choose one of the following options for remote authentication:
1. SSH based authentication
2. Certificate based authentication
Enter your options [1-2]: 1
User name for remote server: username
Generating SSH key on this machine...
SSH key pair path: /opt/amazon/mhub-orchestrator-plugin/remote-auth/linux/keys/
id_rsa_assessment
Please add the public key "id_rsa_assessment.pub" to the "$HOME/.ssh/authorized_keys" file
 in your remote machines.
Your Linux remote server configurations are saved successfully.
```

Enable the Migration Hub Orchestrator plugin to communicate with source servers

Note
This step isn't necessary if you set up the Migration Hub Orchestrator plugin using vCenter credentials.

After you set up your remote server configurations, if you are using the `plugin setup` or `plugin setup --remote-server-configurations` command, you must prepare your remote servers so that the Migration Hub Orchestrator plugin can collect data from them.

Note
You must make sure that the servers are reachable using their private IP address. For further instructions on how to set up the environment through a virtual private cloud (VPC) on AWS for remote running, see the Amazon Virtual Private Cloud User Guide.

Prepare source Linux servers

Set up key-based authentication on Linux servers

If you choose to set up SSH key-based authentication for Linux when configuring source server configurations, you must perform the following steps to set up key-based authentication on your servers so that the Migration Hub Orchestrator plugin can communicate with source server.

To set up key-based authentication on your Linux servers

1. Copy the public key that was generated with the name **id_rsa_assessment.pub** from the following folder in the container:

   /opt/amazon/mhub-orchestrator-plugin/remote-auth/linux/keys.
2. Append the copied public key in the `$HOME/.ssh/authorized_keys` file for all of the remote machines. If there is no file available, create it using the `touch` or `vim` command.

3. Ensure that the home folder on the source server has a permission level of 755 or less. You can use the chmod command to restrict permissions.

Set up certificate-based authentication on Linux servers

If you choose to set up certificate-based authentication for Linux when configuring source server configurations, you must perform the following steps so that the Migration Hub Orchestrator plugin can communicate with the source server.

We recommend this option if you already have Certificate Authority (CA) set up for your application servers.

To set up certificate-based authentication on your Linux servers

1. Copy the user name that works with all of your remote servers.
2. Copy the public key of the plugin to the CA.

   The public key for the plugin can be found in the following location:

   /opt/amazon/mhub-orchestrator-plugin/remote-auth/linux/keys/id_rsa_assessment.pub

   This public key must be added to your CA for generating the certificate.
3. Copy the certificate that was generated in the previous step to the following location in the plugin:

   /opt/amazon/mhub-orchestrator-plugin/remote-auth/linux/keys

   The name of the certificate must be **id_rsa_assessment-cert.pub**.
4. Provide the certificate file name during setup.

Set up the source server configuration on Windows servers

If you choose to set up Windows when you set up the source server in the **plugin setup**, you must perform the following steps so that the Migration Hub Orchestrator plugin can communicate with the source server.

> **To understand more about the PowerShell script that's executed on the source server, read this note.**
> The script enables PowerShell remote and disables all authentication methods other than negotiate. This is used for Windows NT LAN Manager (NTLM) and sets the "AllowUnencrypted" WSMan protocol to false to ensure that the newly created listener accepts only encrypted traffic. Using the Microsoft provided script, `New-SelfSignedCertificateEx.ps1`, it creates a self-signed certificate.
> Any WSMan Instance that has an HTTP listener is removed, along with existing HTTPS listeners. Then, it creates a new HTTPS listener. It also creates an inbound firewall rule for TCP port 5986. In the final step, the WinRM service is restarted.

To set up a remote connection on Windows 2008 servers

1. Use the following command to check the version of PowerShell installed on your server.

```
$PSVersionTable
```

2. If the PowerShell version is not 5.1, then download and install WMF 5.1 by following the instructions at Install and Configure WMF 5.1 in the Microsoft documentation.
3. Use the following command in a new PowerShell window to ensure that PowerShell 5.1 is installed.

```
$PSVersionTable
```

To set up a remote connection on Windows 2012 and newer servers

1. Download the setup script from the following URL:

   Setup script
2. Download the `New-SelfSignedCertificateEx.ps1` from the following URL and paste the script into the same folder in which you downloaded `WinRMSetup.ps1`:

   https://github.com/Azure/azure-libraries-for-net/blob/master/Samples/Asset/New-SelfSignedCertificateEx.ps1
3. To complete the setup, run the downloaded PowerShell script on all application servers.

```
.\WinRMSetup.ps1
```

Note
If Windows Remote Management (WinRM) is not set up properly on the Windows Remote Server, an attempt to communicate will fail. If this happens, you must delete the certificate that corresponds to that server from the following location on the container:
/opt/amazon/mhub-orchestrator-plugin/remote-auth/windows/certs/*ads-server-id*.cer
After you delete the certificate, wait for the ongoing process to be retried.

Templates

Migration Hub Orchestrator offers the following templates to configure your migration workflows.

- Migrate SAP NetWeaver applications to AWS
- Rehost applications on Amazon EC2
- Rehost SQL server on Amazon EC2
- Replatform SQL server on Amazon RDS

Migrate SAP NetWeaver applications to AWS

You can automate the migration of your SAP NetWeaver based applications such as, S/4HANA, BW/4HANA, and ECC on HANA running on SAP HANA database to AWS with this template.

Prerequisites

You must meet the following requirements to create a migration workflow using this template.

- Verify that your servers and applications are on a supported operating system. For more information, see Version support for SAP deployments.
- Provide credentials of SAP HANA database instance running on your source server. These credentials are used by the Migration Hub Orchestrator plugin to communicate with the source server.

  1. Sign in to https://console.aws.amazon.com/secretsmanager/.
  2. On the AWS Secrets Manager page, select **Store a new secret**.
  3. For Secret type, select **Other type of secret** and create the following key value pairs.

Key	Value
`hana_systemdb_username`	source SAP HANA system database username
`hana_systemdb_password`	source SAP HANA system database password
`hana_saptenantdb_username`	source SAP HANA tenant database username
`hana_saptenantdb_user_password`	source SAP HANA tenant database password

     Note
     The `hana_systemdb_username` and `hana_saptenantdb_username` must have admin permissions to enable the SAP HANA System Replication and perform database backups.
  4. Select **Next** and enter a name beginning with `migrationhub-orchestrator-`*`secretname123`* in Secret name.

     Important
     The Secret ID must begin with the prefix `migrationhub-orchestrator-` and must only be followed by an alphanumeric value.
  5. Select **Next** and then, select **Store**.
- The following parameters must be the same on the source and target environments.
  - SAP SID

- SAP HANA SID
- PAS instance number
- ASCS instance number
- SAP HANA instance number
- SAP HANA database password

Target environment setup

AWS Migration Hub Orchestrator guides you to create the target environment in AWS to host your SAP NetWeaver application using AWS Launch Wizard for SAP. For more information, see Get started with AWS Launch Wizard for SAP.

Create an SAP deployment using AWS Launch Wizard for SAP. For more information, see Deploy an SAP application with AWS Launch Wizard for SAP.

> **Note**
> Migration Hub Orchestrator supports single node or multi node SAP NetWeaver stack deployment for target. You must choose to deploy the SAP NetWeaver software as part of target environment setup with Launch Wizard.

- Create a private key in the Amazon EC2 console and store it in the AWS Secrets Manager. The plugin uses this private key associated with the target instance to perform migration tasks.

  See the following steps to create a private key.

  1. Sign in to the Amazon EC2 console.
  2. In the left navigation pane, under Network & Security, select **Key Pairs**.
  3. Select **Create key pair**.
  4. Enter a name for the key pair beginning with `migrationhub-orchestrator-`*`keyname123`*.

     > **Important**
     > The Key Pair must begin with the prefix `migrationhub-orchestrator-` and must only be followed by an alphanumeric value.
  5. Select **RSA** as the Key pair type.
  6. Select **.pem** as the Private key file format.
  7. Select **Create key pair** and save the file.

  See the following steps to store the private key.

  1. Sign in to https://console.aws.amazon.com/secretsmanager/.
  2. On the AWS Secrets Manager page, select **Store a new secret**.
  3. For Secret type, select **Other type of secret** and select **Plaintext** below.
  4. Copy and paste the Private key created in Amazon EC2 console and select Next.
  5. In Secret name, enter the same name (`migrationhub-orchestrator-`*`keyname123`*) that you used for creating the key pair.
  6. Select **Next** and then, **Store**.
- To establish a connection between your source and target environments, we recommend creating a new security group with your source IP address while creating an SAP deployment with Launch Wizard.

  1. Under **Infrastructure - SAP landscape**, go to **Security groups**.
  2. Select **Create new security groups**.
  3. In Connection type, select **IP Address/CIDR**.

4. In Value, enter your source IP address.

- Launch Wizard attaches the `AmazonEC2RoleForLaunchWizard` instanceRole by default when creating the target environment. After creating the target instance with Launch Wizard, attach the `AWSMigrationHubOrchestratorInstanceRolePolicy` managed policy to `AmazonEC2RoleForLaunchWizard`. For more information, see AWS managed policies for Migration Hub Orchestrator.

Create a migration workflow

1. In https://console.aws.amazon.com/migrationhub/, select **Create migration workflow**.
2. On Choose a workflow template page, select **Migrate SAP NetWeaver on HANA applications** template.
3. Configure and submit your workflow to begin migration.

   - the section called “Details” (p. 14)
   - the section called “Application” (p. 14)
   - the section called “Source environment configuration” (p. 14)

Details

Enter a name for your workflow. Optionally, you can enter a description and add tags. If you intend to run multiple migrations, we recommend adding tags to enhance searchability. For more information, see Tagging AWS resources.

Application

Select the application you want to migrate. For more information, see Define applications.

Source environment configuration

Enter the details of the SAP source environment that you want to migrate with the Migration Hub Orchestrator.

SAP application server configuration

- SAPSID: Enter the system ID of the SAP application that you want to migrate.
- AWS Application Discovery Service server ID for SAP application server: Select the server ID where the central instance of your source SAP application is running. The IDs in the list are available based on the application configurations made in AWS Application Discovery Service. For more information, see Define applications.

SAP HANA database configuration

- HANASID: Enter the system ID of your source SAP HANA database.
- Instance number: Enter the instance number of your source SAP HANA database.
- Database hostname: Enter the hostname of your source SAP HANA database. To find the hostname, run the `hostname` command on your database.
- AWS Application Discovery Service server ID for SAP HANA database: Select the server ID where your SAP HANA database is running. The IDs in the list are available based on the application configurations made in AWS Directory Service. For more information, see Define applications.

- Credentials: Select the credentials you created for your source HANA database in the section called "Prerequisites" (p. 12).
- Version: Migration Hub Orchestrator only supports migrations for SAP HANA database 2.0 versions. Verify that the version of your SAP HANA database is 2.0 or higher with `HDB version` command.
- Backup location: Enter the backup location of your SAP HANA database.

Migration steps

Migration Hub Orchestrator automates the migration process after you create the migration workflow. Some tasks require additional inputs and user interactions.

- By default, Launch Wizard deploys the target SAP HANA database with baseline HANA components. If the source application that is being migrated has components that have been deployed after the initial installation, check and deploy those components on the target instance.
- An SAP HANA system has several configuration (`*.ini`) files that contain properties for configuring the system as a whole and individual tenant databases, hosts, and services. SAP HANA's configuration files contain parameters for global system configuration (`global.ini`) and for each service in the system. For instance, `indexserver.ini`. Based on your application requirement, if any of these configuration files have been adjusted on the source, you need to update them on the newly deployed target system before cutover.
- Before beginning cutover, verify that your source application has been migrated properly. Step group 7 of the **Migrate SAP NetWeaver to AWS** template guides you through the necessary steps.
  - **Stop source SAP production system**: Ensure that there are no end users logged in or accessing the application before stopping the source application.
  - **Stop source HANA production system**: Verify that the HANA System Replication has completed copying data to target and gracefully stopped the source HANA database.
  - **Cutover & Start SAP application**: Start the migrated SAP application servers on the target.
  - **Verify database records**: Verify database records to validate that the application has been migrated properly.
  - **Manual post processing**: Perform any manual post-migration tasks, such as attaching interface file systems or updating end user SAPGUIconfiguration to connect to the newly migrated applications on AWS.

Rehost applications on Amazon EC2

You can rehost your custom Windows and Linux applications on Amazon EC2 using the *Rehost applications on Amazon EC2* template.

Prerequisites

You must meet the following requirements to create a migration workflow using this template.

- Verify that your applications are on a supported operating system. For more information, see Supported operating systems.
- AWS Application Migration Service must be initialized by the `admin` user of the AWS account. For more information, see Application Migration Service initialization and permissions .
- Complete the replication settings for AWS Application Migration Service. For more information, see Replication settings.
- Provide credentials in the AWS Secrets Manager to install the AWS Replication Agent on your remote server.

1. Sign in to https://console.aws.amazon.com/secretsmanager/.
2. On the AWS Secrets Manager page, select **Store a new secret**.
3. For Secret type, select **Other type of secret** and enter the following keys.
   - `access_key`
   - `secret_key`
4. Select **Next** and enter a name for the key pair beginning with `migrationhub-orchestrator-`*`secretname123`*.

   Important
   The Secret ID must begin with the prefix `migrationhub-orchestrator-` and must only be followed by an alphanumeric value.
5. Select **Next** and then, select **Store**.

- Create an IAM user and attach the **AWSApplicationMigrationAgentPolicy** policy.
- Create an IAM role with the Amazon EC2 use case to run test scripts on migrated instances. Attach the **AWSMigrationHubOrchestratorInstanceRolePolicy** and **AmazonSSMManagedInstanceCore** policies to this role. Once the role is created, update the trust policy to include SSM (`ssm.amazonaws.com`). For more information on updating a trust policy, see Modifying a role trust policy (console).
- The user running the AWS Application Migration Service must have permissions to perform the `startTest` and `startCutoverInstance` tasks. Create an IAM user and attach the **AWSApplicationMigrationFullAccess**, **AWSApplicationMigrationEC2Access**, and **AmazonEC2FullAccess** policies along with the following inline policy.

```
{
     "Effect": "Allow",
     "Action": [
         "mgn:StartCutover",
         "mgn:StartTest"
     ],
     "Resource": "*"
 },
 {
     "Effect": "Allow",
     "Action": "iam:PassRole",
     "Resource": "*",
     "Condition": {
         "StringEquals": {
             "iam:PassedToService": "ec2.amazonaws.com"
         }
     }
 }
```

Create a migration workflow

1. In https://console.aws.amazon.com/migrationhub/, select **Create migration workflow**.
2. On Choose a workflow template page, select **Rehost on Amazon EC2 using AWS Application Migration Service** template.
3. Configure and submit your workflow to begin migration.
   - the section called “Details” (p. 17)
   - the section called “Application” (p. 17)
   - the section called “Target environment configuration” (p. 17)

Details

Enter a name for your workflow. Optionally, you can enter a description and add tags. If you intend to run multiple migrations, we recommend adding tags to enhance searchability. For more information, see Tagging AWS resources.

Application

Select the application you want to migrate. For more information, see Define applications.

Target environment configuration

If you want to run test scripts on migrated instances, check the box for *I want to run test scripts on the migrated instances*.

> **Note**
> We recommend having separate workflows for Linux and Windows servers if you want to run validation tests on migrated instances.

- Test script location: Specify the Amazon S3 bucket that contains your test script. For more information, see Getting started with Amazon S3.
- IAM role: Choose the IAM role you created in the section called "Prerequisites" (p. 15).
- Script run command: Enter the **run** command for your script.

Credentials to install AWS Replication Agent: Select the credentials you created in the section called "Prerequisites" (p. 15).

Rehost SQL server on Amazon EC2

With **Rehost SQL server on Amazon EC2** template, you can rehost your SQL servers on-premises to Amazon EC2 using native backup and restore. You can also migrate databases that are encrypted with transparent data encryption.

> **Note**
> This template must be used along with AWS Direct Connect. To use the template without AWS Direct Connect, send us an email at mh-orchestrator-interest@amazon.com with your AWS account number and AWS Region where you have registered the Migration Hub Orchestrator plugin.

Topics

Prerequisites

You must set up the source and target environments before creating a migration workflow.

Topics

Source environment setup

- When configuring the Migration Hub Orchestrator plugin, ensure that the user that is provided to connect to your Windows machine has the SYSAdmin permission on the SQL server instance.
- Ensure that PowerShell is enabled on the server that contains your SQL server instance.
- Install AWS.Tools on the server that contains your SQL server instance, with the following command.

```
Install-Module -Name AWS.Tools.Installer
```

  For more information, see What are AWS Tools for PowerShell?
- Create an IAM policy with the following permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "kms:GenerateDataKey",
                "kms:CreateKey"
            ],
            "Resource": "*"
        }
    ]
}
```

- Create an IAM with the preceding following attached.
- Configure a name profile for AWS Command Line Interface that uses the preceding IAM user. For more information, see Using AWS credentials.
- Install the DBA.Tools module on your Windows machine, with the following command.

```
Cmd: Install-Module dbatools
```

Target environment setup

- (*Optional*)If you want to use BYOL for SQL server, use AWS VM Import/Export to import your VM image.
- (*Optional*) Use AWS Launch Wizard to deploy your target SQL server.
  - Launch Wizard attaches the AmazonEC2RoleForLaunchWizard instance role by default when creating the target environment.
  - After creating the target environment with Launch Wizard, attach the AWSMigrationHubOrchestratorInstanceRolePolicy managed policy to AmazonEC2RoleForLaunchWizard. For more information, see AWS managed policies for Migration Hub Orchestrator.
- If you are not using Launch Wizard to create your target environment, attach the AWSMigrationHubOrchestratorInstanceRolePolicy managed policy to your instance role.
- Add the following permissions to your instance role.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "kms:Decrypt",
                "s3:ListAllMyBuckets",
                "s3:ListBucket"
            ],
            "Resource": "*"
        }
    ]
}
```

- Create a user in your target SQL server with SYSAdmin permission.
- Provide credentials in AWS Secrets Manager for the user created in your target SQL server.

  1. Sign in to https://console.aws.amazon.com/secretsmanager/.
  2. On the AWS Secrets Manager page, select **Store a new secret**.
  3. For Secret type, select **Other type of secret** and enter the following keys.
     - `username` - enter your username
     - `password` - enter your password
  4. Select **Next** and enter a name for the key pair beginning with `migrationhub-orchestrator-`*`secretname123`*.

     Important
     The Secret ID must begin with the prefix `migrationhub-orchestrator-` and must only be followed by an alphanumeric value.
  5. Select **Next** and then, select **Store**.

Create a migration workflow

1. In https://console.aws.amazon.com/migrationhub/, select **Create migration workflow**.
2. On Choose a workflow template page, select **Rehost SQL server on Amazon EC2** template.
3. Configure and submit your workflow to begin migration.

Topics

Details

Enter a name for your workflow. Optionally, you can enter a description and add tags. If you intend to run multiple migrations, we recommend adding tags to enhance searchability. For more information, see Tagging AWS resources.

Application

Select the application you want to migrate. For more information, see Define applications.

Replatform SQL server on Amazon RDS

With **Replatform SQL server on Amazon RDS** template, you can migrate your SQL servers on-premises to Amazon RDS using native backup and restore. You can also migrate databases that are encrypted with transparent data encryption.

> **Note**
> This template must be used along with AWS Direct Connect. To use the template without AWS Direct Connect, send us an email at mh-orchestrator-interest@amazon.com with your AWS account number and AWS Region where you have registered the Migration Hub Orchestrator plugin.

Topics

Prerequisites

You must set up the source and target environments before creating a migration workflow.

Topics

Source environment setup

- When configuring the Migration Hub Orchestrator plugin, ensure that the user that is provided to connect to your Windows machine has the `SYSAdmin` permission on the SQL server instance.
- Ensure that PowerShell is enabled on the server that contains your SQL server instance.
- Install AWS.Tools on the server that contains your SQL server instance, with the following command.

```
Install-Module -Name AWS.Tools.Installer
```

  For more information, see What are AWS Tools for PowerShell?
- Create an IAM policy with the following permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "kms:GenerateDataKey",
                "kms:CreateKey"
            ],
            "Resource": "*"
        }
    ]
}
```

- Create an IAM Role with the preceding policy attached.
- Configure a name profile for AWS Command Line Interface that uses the preceding IAM user. For more information, see Using AWS credentials.
- Install the `DBA.Tools` module on your Windows machine, with the following command.

```
Cmd: Install-Module dbatools
```

Target environment setup

- Deploy an Amazon RDS SQL server with the same version as the source SQL server.
- Configure the target Amazon RDS SQL server with the same parameter groups as the source SQL server.
- Deploy an Amazon EC2 instance and create an instance role.
  - Attach the `AWSMigrationHubOrchestratorInstanceRolePolicy` and `AmazonSSMManagedInstanceCore` managed policies to this role.
  - Add the following permissions to this role.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::migrationhub-orchestrator-*",
                "arn:aws:s3:::aws-migrationhub-orchestrator-*/*"
            ]
        }
    ]
}
```

- Migration Hub Orchestrator plugin creates an Amazon S3 bucket to store on-premises backups and transparent data encryptions, if the source SQL server is using it.

  For more information, see the following.
  - Importing and exporting SQL Server databases using native backup and restore
  - Support for Transparent Data Encryption in SQL Server

  Configure the option group for backup/restore and transparent data encryption, and attach the following policies to the created IAM role.

```
   {
  "Version": "2012-10-17",
  "Statement": [
      {
          "Sid": "VisualEditor0",
          "Effect": "Allow",
          "Action": [
              "kms:Decrypt",
              "s3:ListAllMyBuckets",
              "kms:DescribeKey"
          ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketAcl",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "VisualEditor2",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:AbortMultipartUpload",
                "s3:ListMultipartUploadParts"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

```
     {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

- Ensure that your Amazon RDS instance can be reached from the created Amazon EC2 instance.
- Provide credentials in AWS Secrets Manager for the user created in your target SQL server.

1. Sign in to https://console.aws.amazon.com/secretsmanager/.
2. On the AWS Secrets Manager page, select **Store a new secret**.
3. For Secret type, select **Other type of secret** and enter the following keys.
   - `username` - enter your username
   - `password` - enter your password
4. Select **Next** and enter a name for the key pair beginning with `migrationhub-orchestrator-`*`secretname123`*.

   Important
   The Secret ID must begin with the prefix `migrationhub-orchestrator-` and must only be followed by an alphanumeric value.
5. Select **Next** and then, select **Store**.

Create a migration workflow

1. In https://console.aws.amazon.com/migrationhub/, select **Create migration workflow**.
2. On Choose a workflow template page, select **Rehost SQL server on Amazon EC2** template.
3. Configure and submit your workflow to begin migration.

Topics

Details

Enter a name for your workflow. Optionally, you can enter a description and add tags. If you intend to run multiple migrations, we recommend adding tags to enhance searchability. For more information, see Tagging AWS resources.

Application

Select the application you want to migrate. For more information, see Define applications.

Migration workflows

Migration Hub Orchestrator provides predefined templates that offer automation capabilities and facilitate the migration of your on-premises servers and applications to AWS. A template consists of one or more step groups that contain steps. A step can be automated or manual.

You can create a workflow with one of the following templates.

- Migrate SAP NetWeaver applications to AWS

  A template to migrate SAP NetWeaver-based applications (S/4HANA, BW4HANA, and ECC on HANA) running on SAP HANA database to AWS.
- Rehost applications on Amazon EC2

  A template to rehost applications on Amazon EC2 using AWS Application Migration Service (AWS MGN).

Add a step

You can add, reorder, and delete step groups and steps *after* you create the workflow.

1. Sign in to the https://console.aws.amazon.com/migrationhub/.
2. In the left navigation pane, choose **Orchestrate** > **Workflows**.
3. On the **Workflows** page, select the workflow that you want to customize and choose **View details**.
4. Under **Steps**, select **Add**.
5. To create a manual step, enter a **Name** for your step and choose **Add**.

   Note
   Manual steps require user intervention. After completing the step, you must update the status to enable the migration workflow to continue.
6. To create an automated step:

   - Enter a **Name** for your step.
   - Specify a **Script location**. You can upload a custom script to an Amazon S3 bucket or upload a file from your local machine. For more information, see Getting started with Amazon S3.
   - Enter the **Script run command** that Migration Hub Orchestrator can use to run your script.
   - In **Script run environment**, select **On premises** to run the script in the source environment, or select **AWS** to run the script in the target environment.
   - Based on your selection for the **Script run environment**, the list under **Server** displays the applications that you configured in the Application Discovery Service. For more information, see Define applications.

Rules and limitations

There are some rules and limitations when customizing migration workflows:

- You can make modifications to a migration workflow after it's created.
- A step must be placed within a step group. You can choose to add a step to an existing step group or create a new step group.

- A step group must have at least one step.
- A step can't be added to a step group with a status of **Completed**.
- To delete an ongoing migration workflow, you must pause it first.

Security in Migration Hub Orchestrator

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Migration Hub Orchestrator, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations

This documentation helps you understand how to apply the shared responsibility model when using Migration Hub Orchestrator. It shows you how to configure Migration Hub Orchestrator to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Migration Hub Orchestrator resources.

Contents

Data protection in Migration Hub Orchestrator

The AWS shared responsibility model applies to data protection in Migration Hub Orchestrator. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.

- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a **Name** field. This includes when you work with Migration Hub Orchestrator or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Encryption at rest

Migration Hub Orchestrator encrypts all data at rest.

Encryption in transit

Migration Hub Orchestrator inter-network communications support TLS 1.2 encryption between all components and clients.

Identity and access management for Migration Hub Orchestrator

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Migration Hub Orchestrator resources. IAM is an AWS service that you can use with no additional charge.

Topics

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Migration Hub Orchestrator.

Service user – If you use the Migration Hub Orchestrator service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Migration Hub Orchestrator features to do your work, you might need additional permissions. Understanding how

access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Migration Hub Orchestrator, see Troubleshooting Migration Hub Orchestrator identity and access (p. 39).

Service administrator – If you're in charge of Migration Hub Orchestrator resources at your company, you probably have full access to Migration Hub Orchestrator. It's your job to determine which Migration Hub Orchestrator features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Migration Hub Orchestrator, see How Migration Hub Orchestrator works with IAM (p. 32).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Migration Hub Orchestrator. To view example Migration Hub Orchestrator identity-based policies that you can use in IAM, see Identity-based policy examples for Migration Hub Orchestrator (p. 36).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (successor to AWS Single Sign-On) (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see Signature Version 4 signing process in the *AWS General Reference*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide* and Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *AWS General Reference*.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using

credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center (successor to AWS Single Sign-On). You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.

  - **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, Resources, and Condition Keys for Migration Hub Orchestrator in the *Service Authorization Reference*.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

Every IAM entity (user or role) starts with no permissions. By default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that

you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How Migration Hub Orchestrator works with IAM

Before you use IAM to manage access to Migration Hub Orchestrator, learn what IAM features are available to use with Migration Hub Orchestrator.

IAM features you can use with Migration Hub Orchestrator

IAM feature	Migration Hub Orchestrator support
Identity-based policies (p. 32)	Yes
Resource-based policies (p. 33)	Yes
Policy actions (p. 33)	Yes
Policy resources (p. 34)	Yes
Policy condition keys (p. 34)	Yes
ACLs (p. 35)	No
ABAC (tags in policies) (p. 35)	No
Temporary credentials (p. 35)	Yes
Principal permissions (p. 36)	Yes
Service roles (p. 36)	No
Service-linked roles (p. 36)	Yes

To get a high-level view of how Migration Hub Orchestrator and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for Migration Hub Orchestrator

Supports identity-based policies	Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for Migration Hub Orchestrator

To view examples of Migration Hub Orchestrator identity-based policies, see Identity-based policy examples for Migration Hub Orchestrator (p. 36).

Resource-based policies within Migration Hub Orchestrator

Supports resource-based policies	Yes

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

Policy actions for Migration Hub Orchestrator

Supports policy actions	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Migration Hub Orchestrator actions, see Actions Defined by Migration Hub Orchestrator in the *Service Authorization Reference*.

Policy actions in Migration Hub Orchestrator use the following prefix before the action:

```
migrationhub-orchestrator
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "migrationhub-orchestrator:action1",
      "migrationhub-orchestrator:action2"
         ]
```

To view examples of Migration Hub Orchestrator identity-based policies, see Identity-based policy examples for Migration Hub Orchestrator (p. 36).

Policy resources for Migration Hub Orchestrator

Supports policy resources	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of Migration Hub Orchestrator resource types and their ARNs, see Resources Defined by Migration Hub Orchestrator in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions Defined by Migration Hub Orchestrator .

To view examples of Migration Hub Orchestrator identity-based policies, see Identity-based policy examples for Migration Hub Orchestrator (p. 36).

Policy condition keys for Migration Hub Orchestrator

Supports service-specific policy condition keys	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of Migration Hub Orchestrator condition keys, see Condition Keys for Migration Hub Orchestrator in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions Defined by Migration Hub Orchestrator .

To view examples of Migration Hub Orchestrator identity-based policies, see Identity-based policy examples for Migration Hub Orchestrator (p. 36).

Access control lists (ACLs) in Migration Hub Orchestrator

Supports ACLs	No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Attribute-based access control (ABAC) with Migration Hub Orchestrator

Supports ABAC (tags in policies)	No

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see What is ABAC? in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using Temporary credentials with Migration Hub Orchestrator

Supports temporary credentials	Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switching to a role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Cross-service principal permissions for Migration Hub Orchestrator

Supports principal permissions	Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, Resources, and Condition Keys for Migration Hub Orchestrator in the *Service Authorization Reference*.

Service roles for Migration Hub Orchestrator

Supports service roles	No

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
> Changing the permissions for a service role might break Migration Hub Orchestrator functionality. Edit service roles only when Migration Hub Orchestrator provides guidance to do so.

Service-linked roles for Migration Hub Orchestrator

Supports service-linked roles	Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing Migration Hub Orchestrator service-linked roles, see AWS services that work with IAM.

Identity-based policy examples for Migration Hub Orchestrator

By default, users and roles don't have permission to create or modify Migration Hub Orchestrator resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform actions on the resources that they need. The administrator must then attach those policies for users that require them.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Creating IAM policies in the *IAM User Guide*.

For details about actions and resource types defined by Migration Hub Orchestrator, including the format of the ARNs for each of the resource types, see Actions, Resources, and Condition Keys for Migration Hub Orchestrator in the *Service Authorization Reference*.

Topics

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Migration Hub Orchestrator resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see IAM Access Analyzer policy validation in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or root users in your account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Configuring MFA-protected API access in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the Migration Hub Orchestrator console

To access the Migration Hub Orchestrator console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Migration Hub Orchestrator resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (IAM users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that you're trying to perform.

To ensure that users and roles can still use the Migration Hub Orchestrator console, also attach the Migration Hub Orchestrator `ConsoleAccess` or `ReadOnly` AWS managed policy to the entities. For more information, see Adding permissions to a user in the *IAM User Guide*.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Accessing one Amazon S3 bucket

In this example, you want to grant an IAM user in your AWS account access to one of your Amazon S3 buckets, `examplebucket`. You also want to allow the user to add, update, and delete objects.

In addition to granting the `s3:PutObject`, `s3:GetObject`, and `s3:DeleteObject` permissions to the user, the policy also grants the `s3:ListAllMyBuckets`, `s3:GetBucketLocation`, and `s3:ListBucket` permissions. These are the additional permissions required by the console. Also, the `s3:PutObjectAcl` and the `s3:GetObjectAcl` actions are required to be able to copy, cut, and paste objects in the console. For an example walkthrough that grants permissions to users and tests them using the console, see An example walkthrough: Using user policies to control access to your bucket.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"ListBucketsInConsole",
         "Effect":"Allow",
```

```
         "Action":[
            "s3:ListAllMyBuckets"
         ],
         "Resource":"arn:aws:s3:::*"
      },
      {
         "Sid":"ViewSpecificBucketInfo",
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket",
            "s3:GetBucketLocation"
         ],
         "Resource":"arn:aws:s3:::examplebucket"
      },
      {
         "Sid":"ManageBucketContents",
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:GetObject",
            "s3:GetObjectAcl",
            "s3:DeleteObject"
         ],
         "Resource":"arn:aws:s3:::examplebucket/*"
      }
   ]
}
```

Troubleshooting Migration Hub Orchestrator identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Migration Hub Orchestrator and IAM.

Topics

I am not authorized to perform an action in Migration Hub Orchestrator

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but does not have the fictional `migrationhub-orchestrator:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 migrationhub-orchestrator:GetWidget on resource: my-example-widget
```

In this case, Mateo asks his administrator to update his policies to allow him to access the *my-example-widget* resource using the `migrationhub-orchestrator:GetWidget` action.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to Migration Hub Orchestrator.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Migration Hub Orchestrator. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to view my access keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (for example, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

> **Important**
> Do not provide your access keys to a third party, even to help find your canonical user ID. By doing this, you might give someone permanent access to your account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see Managing access keys in the *IAM User Guide*.

I'm an administrator and want to allow others to access Migration Hub Orchestrator

To allow others to access Migration Hub Orchestrator, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in Migration Hub Orchestrator.

To get started right away, see Creating your first IAM delegated user and group in the *IAM User Guide*.

I want to allow people outside of my AWS account to access my Migration Hub Orchestrator resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Migration Hub Orchestrator supports these features, see How Migration Hub Orchestrator works with IAM (p. 32).
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

AWS managed policies for Migration Hub Orchestrator

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to create IAM customer managed policies that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see AWS managed policies in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see AWS managed policies for job functions in the *IAM User Guide*.

AWS managed policy: AWSMigrationHubOrchestratorConsoleFullAccess

Attach the `AWSMigrationHubOrchestratorConsoleFullAccess` policy to your IAM identities.

The `AWSMigrationHubOrchestratorConsoleFullAccess` policy grants an IAM user account full access to the Migration Hub Orchestrator service through the AWS Management Console.

Permissions details

This policy includes the following permissions.

- `migrationhub-orchestrator` – Allows the IAM user account full access to Migration Hub Orchestrator.
- `s3` – Allows the IAM user account to create and read from the S3 buckets used by Migration Hub Orchestrator.
- `secretsmanager` – Allows the IAM user account to list secrets access in the Secrets Manager.
- `discovery` – Allows the IAM user account access to get discovery summary in Application Discovery Service.
- `iam` – Allows a service-linked role to be created for the IAM user account, which is a requirement for using Migration Hub Orchestrator.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "migrationhub-orchestrator:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:CreateBucket",
                "s3:PutEncryptionConfiguration",
                "s3:PutBucketPublicAccessBlock",
                "s3:PutBucketPolicy",
                "s3:PutBucketVersioning",
                "s3:PutLifecycleConfiguration"
            ],
            "Resource": "arn:aws:s3:::migrationhub-orchestrator-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:ListSecrets"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "discovery:GetDiscoverySummary"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "migrationhub-orchestrator.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/migrationhub-
orchestrator.amazonaws.com/AWSMigrationHubOrchestratorServiceRolePolicy*"
        }
    ]
}
```

AWS managed policy: AWSMigrationHubOrchestratorPlugin

Attach the `AWSMigrationHubOrchestratorPlugin` policy to your IAM identities.

The `AWSMigrationHubOrchestratorPlugin` policy grants an IAM user account access to the Migration Hub Orchestrator service, read/write access to the S3 buckets that are related to the service, Amazon API Gateway access to upload logs and metrics to AWS, and AWS Secrets Manager access to fetch credentials.

Permissions details

This policy includes the following permissions.

- `migrationhub-orchestrator` – Allows the IAM user account access to the Orchestrator plugin.
- `s3` – Allows the IAM user account write access to the S3 buckets used by Migration Hub Orchestrator.
- `secretsmanager` – Allows the IAM user account to access secrets in the Secrets Manager that are used by Migration Hub Orchestrator.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutObject",
                "s3:GetObject",
                "s3:GetBucketAcl"
            ],
            "Resource": "arn:aws:s3:::migrationhub-orchestrator-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "execute-api:Invoke",
                "execute-api:ManageConnections"
```

```
            ],
            "Resource": [
                "arn:aws:execute-api:*:*:*/prod/*/put-log-data",
                "arn:aws:execute-api:*:*:*/prod/*/put-metric-data"
            ]
        },
            {
                "Effect": "Allow",
                "Action": [
                    "migrationhub-orchestrator:RegisterPlugin",
                    "migrationhub-orchestrator:GetMessage",
                    "migrationhub-orchestrator:SendMessage"
                ],
                "Resource": "arn:aws:migrationhub-orchestrator:*:*:*"
            },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:migrationhub-orchestrator-*"
        }
    ]
}
```

AWS managed policy: AWSMigrationHubOrchestratorInstanceRolePolicy

Attach the `AWSMigrationHubOrchestratorInstanceRolePolicy` policy to your IAM identities.

This policy grants an IAM user account read/write access to the S3 buckets that are related to the service and AWS Secrets Manager to fetch credentials.

Permissions details

This policy includes the following permissions.

- `migrationhub-orchestrator` – Allows the IAM user account access to Migration Hub Orchestrator.
- `s3` – Allows the IAM user account write access to the S3 buckets used by Migration Hub Orchestrator.
- `secretsmanager` – Allows the IAM user account to access secrets in the Secrets Manager that are used by Migration Hub Orchestrator.

```
{
    "Effect": "Allow",
    "Action": [
        "secretsmanager:GetSecretValue"
    ],
    "Resource": "arn:aws:secretsmanager:*:*:secret:migrationhub-orchestrator-*"
}, {
    "Effect": "Allow",
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::migrationhub-orchestrator-*",
        "arn:aws:s3:::aws-migrationhub-orchestrator-*/*"
    ]
}
```

Migration Hub Orchestrator updates to AWS managed policies

View details about updates to AWS managed policies for Migration Hub Orchestrator since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Migration Hub Orchestrator Document history page.

Change	Description	Date
AWSMigrationHubOrchestratorConsoleFullAccess (p. 41) – New policy made available at launch	AWSMigrationHubOrchestratorConsoleFullAccess grants an IAM user account full access to the Migration Hub Orchestrator service through the AWS Management Console.	April 20, 2022
AWSMigrationHubOrchestratorPlugin (p. 43) – New policy made available at launch	AWSMigrationHubOrchestratorPlugin grants an IAM user account access to the Migration Hub Orchestrator service and read/write access to the S3 buckets that are related to the service. It also grants Amazon API Gateway access to upload logs and metrics to AWS, and AWS Secrets Manager access to fetch credentials.	April 20, 2022
AWSMigrationHubOrchestratorServiceRolePolicy (p. 46) – New policy made available at launch	The AWSMigrationHubOrchestratorServiceRolePolicy service-linked role policy provides access to AWS Migration Hub and AWS Application Discovery Service. This policy also grants permissions for storing reports in Amazon Simple Storage Service (Amazon S3).	April 20, 2022
AWSMigrationHubOrchestratorInstanceRolePolicy – New policy	AWSMigrationHubOrchestratorInstanceRolePolicy grants an IAM user account read/write access to the S3 buckets that are related to the service and AWS Secrets Manager to fetch credentials.	April 20, 2022
Migration Hub Orchestrator started tracking changes	Migration Hub Orchestrator started tracking changes for its AWS managed policies.	April 20, 2022

Using service-linked roles for Migration Hub Orchestrator

Migration Hub Orchestrator uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to Migration Hub Orchestrator.

Service-linked roles are predefined by Migration Hub Orchestrator and include all of the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up Migration Hub Orchestrator easier because you don't have to manually add the necessary permissions. Migration Hub Orchestrator defines the permissions of its service-linked roles, and unless you make changes to the configuration, only Migration Hub Orchestrator can assume its roles. Configurable permissions include the trust policy and the permissions policy. You can't attach the permissions policy to any other IAM entity.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Follow the **Yes** link to view the service-linked role documentation for that service, if applicable.

Service-linked role permissions for Migration Hub Orchestrator

Migration Hub Orchestrator uses the service-linked role named **AWSServiceRoleForMigrationHubOrchestrator** and associates it with the **AWSMigrationHubOrchestratorServiceRolePolicy** IAM policy – Provides access to AWS Migration Hub and AWS Application Discovery Service. This policy also grants permissions for storing reports in Amazon Simple Storage Service (Amazon S3).

The **AWSServiceRoleForMigrationHubOrchestrator** service-linked role trusts the following services to assume the role:

- `migrationhub-orchestrator.amazonaws.com`

The role permissions policy allows Migration Hub Orchestrator to complete the following actions.

AWS Application Discovery Service actions

```
discovery:ListConfigurations

discovery:DescribeConfigurations
```

AWS Launch Wizard actions

```
launchwizard:ListProvisionedApps

launchwizard:DescribeProvisionedApp
```

Amazon Elastic Compute Cloud actions

```
ec2:DescribeInstances

ec2:CreateLaunchTemplateVersion

ec2:ModifyLaunchTemplate
```

AWS Migration Hub actions

```
mgh:GetHomeRegion
```

Amazon EC2 Systems Manager actions

```
ssm:SendCommand

ssm:GetCommandInvocation

ssm:CancelCommand

ssm:DescribeInstanceInformation
```

`ssm:GetCommandInvocatio`

Amazon S3 actions

`s3:GetObject`

Amazon EventBridge actions

`events:PutTargets`

`events:DescribeRule`

`events:DeleteRule`

`events:PutRule`

AWS Application Migration Service actions

`mgn:GetReplicationConfiguration`

`mgn:GetLaunchConfiguration`

`mgn:StartCutover`

`mgn:FinalizeCutover`

`mgn:StartTest`

`mgn:UpdateReplicationConfiguration`

`mgn:DescribeSourceServers`

`mgn:MarkAsArchived`

`mgn:ChangeServerLifeCycleState`

`mgn:StartReplication`

The following is the full policy showing which resources the above actions apply to:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "discovery:DescribeConfigurations",
                "discovery:ListConfigurations"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "launchwizard:ListProvisionedApps",
                "launchwizard:DescribeProvisionedApp"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateLaunchTemplateVersion",
                "ec2:ModifyLaunchTemplate"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/AWSApplicationMigrationServiceManaged":
"mgn.amazonaws.com"
                }
            }
        },
        {
            "Action": [
                "mgh:GetHomeRegion"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:GetCommandInvocation",
                "ssm:CancelCommand"
            ],
            "Resource": [
                "arn:aws:ssm:*::document/AWS-RunRemoteScript",
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:s3:::aws-migrationhub-orchestrator-*",
                "arn:aws:s3:::migrationhub-orchestrator-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceInformation",
                "ssm:GetCommandInvocation"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::migrationhub-orchestrator-*",
                "arn:aws:s3:::migrationhub-orchestrator-*/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "events:PutTargets",
                "events:DescribeRule",
                "events:DeleteRule",
                "events:PutRule"
            ],
```

```
            "Resource": "arn:aws:events:*:*:rule/MigrationHubOrchestratorManagedRule*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "mgn:GetReplicationConfiguration",
                "mgn:GetLaunchConfiguration",
                "mgn:StartCutover",
                "mgn:FinalizeCutover",
                "mgn:StartTest",
                "mgn:UpdateReplicationConfiguration",
                "mgn:DescribeSourceServers",
                "mgn:MarkAsArchived",
                "mgn:ChangeServerLifeCycleState",
                "mgn:StartReplication"
            ],
            "Resource": "*"
        }
    ]
}
```

To view the update history of this policy, see Migration Hub Orchestrator updates to AWS managed policies (p. 45).

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Creating a service-linked role for Migration Hub Orchestrator

You don't need to manually create a service-linked role. When you agree to allow Migration Hub to create a service-linked role (SLR) in your account in the AWS Management Console, Migration Hub Orchestrator creates the service-linked role for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you agree to allow Migration Hub to create a service-linked role (SLR) in your account, Migration Hub Orchestrator creates the service-linked role for you again.

Editing a service-linked role for Migration Hub Orchestrator

Migration Hub Orchestrator does not allow you to edit the **AWSServiceRoleForMigrationHubOrchestrator** service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using the Migration Hub Orchestrator console, CLI, or API.

Deleting a service-linked role for Migration Hub Orchestrator

To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the **AWSServiceRoleForMigrationHubOrchestrator** service-linked role. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

When deleting Migration Hub Orchestrator resources used by the **AWSServiceRoleForMigrationHubOrchestrator** SLR, you cannot have any running assessments (tasks for generating recommendations). No background assessments can be running, either. If assessments are running, the SLR deletion fails in the IAM console. If the SLR deletion fails, you can retry the deletion after all background tasks have completed. You don’t need to clean up any created resources before you delete the SLR.

Supported Regions for Migration Hub Orchestrator service-linked roles

Migration Hub Orchestrator supports using service-linked roles in all of the regions where the service is available. For more information, see AWS Regions and Endpoints.

Migration Hub Orchestrator and interface VPC endpoints (AWS PrivateLink)

You can establish a private connection between your VPC and Migration Hub Orchestrator by creating an *interface VPC endpoint*. Interface endpoints are powered by AWS PrivateLink. With AWS PrivateLink, you can privately access Migration Hub Orchestrator API operations without an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to communicate with Migration Hub Orchestrator API operations. Traffic between your VPC and Migration Hub Orchestrator stays within the Amazon network.

Each interface endpoint is represented by one or more Elastic Network Interfaces in your subnets.

For more information, see Interface VPC endpoints (AWS PrivateLink) in the *Amazon VPC User Guide*.

Considerations for Migration Hub Orchestrator VPC endpoints

Before you set up an interface VPC endpoint for Migration Hub Orchestrator, ensure that you review Interface endpoint properties and limitations and AWS PrivateLink quotas in the *Amazon VPC User Guide*.

Migration Hub Orchestrator supports making calls to all of its API actions from your VPC. To use all of Migration Hub Orchestrator, you must create a VPC endpoint.

Creating an interface VPC endpoint for Migration Hub Orchestrator

You can create a VPC endpoint for Migration Hub Orchestrator using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Creating an interface endpoint in the *Amazon VPC User Guide*.

Create a VPC endpoint for Migration Hub Orchestrator using the following service name:

- `com.amazonaws.region.migrationhub-orchestrator`

If you use private DNS for the endpoint, you can make API requests to Migration Hub Orchestrator using its default DNS name for the Region. For example, you can use the name `migrationhub-orchestrator.us-east-1.amazonaws.com`.

For more information, see Accessing a service through an interface endpoint in the *Amazon VPC User Guide*.

Creating a VPC endpoint policy for Migration Hub Orchestrator

You can attach an endpoint policy to your VPC endpoint. The VPC endpoint policy controls access to Migration Hub Orchestrator. The policy specifies the following information:

- The principal that can perform actions
- The actions that can be performed
- The resources on which these actions can be performed

For more information, see Controlling access to services with VPC endpoints in the *Amazon VPC User Guide*.

Example: VPC endpoint policy for Migration Hub Orchestrator actions

The following is an example of an endpoint policy for Migration Hub Orchestrator. When attached to an endpoint, this policy grants access to the listed Migration Hub Orchestrator actions for all principals on all resources.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "migrationhub-orchestrator:ListMigrationWorkflowTemplates",
         ],
         "Resource":"*"
      }
   ]
}
```

Compliance validation for Migration Hub Orchestrator

Third-party auditors assess the security and compliance of Migration Hub Orchestrator as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using Migration Hub Orchestrator is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA Security and Compliance Whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – AWS Config; assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in Migration Hub Orchestrator

The AWS global infrastructure is built around AWS Regions and Availability Zones. Regions provide multiple physically separated and isolated Availability Zones, which are connected through low-latency,

high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in Migration Hub Orchestrator

As a managed service, Migration Hub Orchestrator is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access Migration Hub Orchestrator through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Logging Migration Hub Orchestrator API calls using AWS CloudTrail

Migration Hub Orchestrator integrates with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Migration Hub Orchestrator. CloudTrail captures all API calls for Migration Hub Orchestrator as events. The calls that are captured include calls from the Migration Hub Orchestrator console and code calls to Migration Hub Orchestrator API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Migration Hub Orchestrator. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Migration Hub Orchestrator, the IP address from which the request was made, who made the request, when it was made, and other details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Migration Hub Orchestrator information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When there is activity in Migration Hub Orchestrator, it's recorded in a CloudTrail event along with other AWS service events in the **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for Migration Hub Orchestrator, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for creating a trail
- CloudTrail supported services and integrations
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple Regions
- Receiving CloudTrail log files from multiple accounts

Migration Hub Orchestrator supports logging the following actions as events in CloudTrail log files:

- CreateMigrationWorkflow
- UpdateMigrationWorkflow
- DeleteMigrationWorkflow
- StartMigrationWorkflow
- StopMigrationWorkflow

- TagResource
- UntagResource
- CreateWorkflowStep
- UpdateWorkflowStep
- DeleteWorkflowStep
- RetryWorkflowStep
- CreateWorkflowStepGroup
- UpdateWorkflowStepGroup
- DeleteWorkflowStepGroup
- GetMigrationWorkflow
- ListMigrationWorkflows
- GetMigrationWorkflowTemplate
- ListMigrationWorkflowTemplates
- ListTemplateStepGroups
- GetTemplateStepGroup
- ListTemplateSteps
- GetTemplateStep
- ListTagsForResource
- GetWorkflowStep
- ListWorkflowSteps
- GetWorkflowStepGroup
- ListWorkflowStepGroups

- ListPlugins

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity element.

Understanding Migration Hub Orchestrator log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `GetWorkflowStep` action.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        type": "AssumedRole",
        "principalId": "777777777777",
        "arn": "arn:aws:sts::111122223333:assumed-role/myUserName/...",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "777777777777",
                "arn": "arn:aws:iam::111122223333:role/myUserName",
                "accountId": "111122223333",
                "userName": "myUserName"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-03-22T23:29:22Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-03-23T03:16:55Z",
    "eventSource": "migrationhub-orchestrator.amazonaws.com",
    "eventName": "GetWorkflowStep",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "99.99.999.999",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:91.0) Gecko/20100101
 Firefox/91.0",
    "requestParameters": {
        "stepGroupId": "act-1",
        "id": "step-11111",
        "workflowId": "mw-1111111"
```

```
    },
    "responseElements": null,
    "requestID": "068e87d1",
    "eventID": "e699238c",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Quotas for Migration Hub Orchestrator

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, and other quotas cannot be increased.

To view a list of the quotas for Migration Hub Orchestrator, see Orchestrator service quotas.

To view the quotas for Migration Hub Orchestrator, open the Service Quotas console. In the navigation pane, choose **AWS services** and select **Migration Hub Orchestrator**.

To request a quota increase, see Requesting a Quota Increase in the *Service Quotas User Guide*. If the quota is not yet available in Service Quotas, use the limit increase form.

Document history

Change	Description	Date
New feature	Added Replatform SQL server on Amazon RDS template.	November 01, 2022
New feature	Added Rehost SQL server on Amazon EC2 template.	November 01, 2022
Initial release	Initial release of the Migration Hub Orchestrator User Guide.	April 20, 2022