User Guide

AWS Migration Hub Orchestrator

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. You can get started quickly through the predefined workflow templates which are built leveraging the migration experience of AWS which also adhere to best practices. You can also use Migration Hub Orchestrator to automate the error-prone manual tasks involved in the migration process and orchestrate the related migration tools from start to finish in the Migration Hub Orchestrator console.

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.

Migration Hub Orchestrator currently supports the following use cases for migration:

- Importing virtual machine (VM) images
- Migrating SAP NetWeaver applications and HANA databases
- Rehosting applications on Amazon EC2
- Rehosting SQL Server databases to Amazon EC2 using SQL Server's native backup and restore
- Replatforming SQL Server databases to Amazon RDS
- Replatforming applications to Amazon ECS on AWS Fargate.

You can use a predefined workflow template to orchestrate the validation of the source environment for migration readiness, provision your target environment, migrate databases and applications, perform post-migration validation, and cutover to AWS.

You can access Migration Hub Orchestrator from https://console.aws.amazon.com/migrationhub/orchestrator/ or from the AWS Command Line Interface. For more information about the Migration Hub Orchestrator APIs and how to use the AWS Command Line Interface (AWS CLI), see the AWS Migration Hub Orchestrator API Reference and the AWS Migration Hub Orchestrator CLI Command Reference.

Are you a first-time user of Migration Hub Orchestrator?

If you are a first-time user of Migration Hub, we recommend that you begin by reading the following sections:

- **Setting up** – The Setting up section details what you need to set up before using Migration Hub Orchestrator.
- **Templates** – The Templates section details the templates offered by Migration Hub Orchestrator for different migration use cases.
- **Workflows** – The Migration workflows section details how workflows function in Migration Hub Orchestrator.

Related services

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

- Application Discovery Service
- AWS App2Container
- AWS Application Migration Service
- AWS Launch Wizard for SAP
- AWS Launch Wizard for SQL Server
- VM Import/Export

Pricing

Migration Hub Orchestrator is available to you at no additional cost. You only pay for the AWS resources that you provision for migrations. For more information, see AWS Migration Hub pricing.

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 AWS 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 **administrative user**, follow the steps in Create an administrative user.

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 new access key ID and secret access key in a safe and secure place.

Home Region

The data stored in the AWS Migration Hub (Migration Hub) home Region provides a single repository of discovery and migration planning information for your entire migration portfolio. The data stored in the home Region from the discovery and migration tools is used to track the progress of your migrations regardless of the migrating application's target Region. For more information, see Migration Hub home Region.

How AWS Migration Hub Orchestrator works

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

Topics

Select a template

Based on your migration requirements, select a template to begin your migration journey with Migration Hub Orchestrator. You can see the steps of a template by selecting a template card, and then choosing **Preview**.

For more information about the different templates offered by Migration Hub Orchestrator, see Templates.

Create a workflow

After selecting your template, you can start configuring your migration workflow. Ensure that you meet the prerequisites of your selected template, and that you have defined the applications you want to migrate in AWS Application Discovery Service.

Run the workflow

Once you have configured your workflow, you can run the workflow. You can now track the progress of your migration and customize your workflow. For more information, see Migration workflows.

> **Note**
>
> Before you can run the workflow, some templates require the Migration Hub Orchestrator plugin to be configured on-premises. The following table denotes which templates require the plugin setup.

Template	Plugin setup required
Migrate SAP NetWeaver applications to AWS	Yes
Rehost applications on Amazon EC2	Yes
Rehost SQL server on Amazon EC2	Yes
Replatform SQL server on Amazon RDS	Yes
Import virtual machine images to AWS	Optional
Replatform applications to Amazon ECS	No

The plugin communicates with the source and target environments to orchestrate and automate migrations. To download and setup the Migration Hub Orchestrator plugin, see Configure Migration Hub Orchestrator plugin.

Templates

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

- **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).
- **Rehost SQL Server on Amazon EC2** – A template to rehost SQL Server on Amazon EC2 using automated SQL Server backup and restore.
- **Replatform SQL Server on Amazon Relational Database Service (Amazon RDS)** – A template to replatform SQL Server on Amazon RDS using native SQL Server backup and restore.
- **Replatform applications to Amazon ECS** – A template to replatform applications to containers on Amazon Elastic Container Service (Amazon ECS).
- **Import virtual machine images to AWS** – A template to import virtual machine (VM) images to AWS as an Amazon Machine Image (AMI) for Amazon EC2.
- **Custom templates** – A template that you have created by modifying an existing AWS managed template and saving the changes.

Templates

- Migrate SAP NetWeaver based applications and SAP HANA databases to AWS
- Rehost applications on Amazon EC2
- Rehost SQL Server on Amazon EC2
- Replatform SQL on Amazon RDS
- Replatform applications to Amazon ECS
- Import virtual machine images to AWS
- Custom templates

Migrate SAP NetWeaver based applications and SAP HANA databases to AWS

With this template, you can automate the migration of your SAP NetWeaver based applications along with SAP HANA databases, or SAP HANA databases only to AWS.

Topics

Migration types

The template offers the following migration types.

- SAP NetWeaver on SAP HANA – central system installation
- SAP NetWeaver on SAP HANA – distributed system installation
- SAP NetWeaver on SAP HANA – high availability installation
- SAP NetWeaver on SAP HANA – scale-out
- SAP HANA database – single node
- SAP HANA database – high availability
- SAP HANA database – scale-out

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.

- Enable network connectivity between the source and target servers by opening the required ports on both servers.
- 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
- You must disable SAP HANA system replication before migrating SAP environments with high availability setup.

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.

- Migration Hub Orchestrator uses the same secret to connect to databases on source and target servers for validation. For your target server, ensure that you provide the same SAP HANA database sign-in credentials that you stored in AWS Secrets Manager following the steps in the section called "Prerequisites".

Create a migration workflow

1. Go to https://console.aws.amazon.com/migrationhub/orchestrator/, and 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"
   - the section called "Application"
   - the section called "Source environment configuration"

> **Note**
>
> You can customize the migration workflow once it has been created. For more information, see Migration workflows.

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. If you do not see the application in the list, you must define it in AWS Application Discovery Service.

Define applications

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

Topics

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**.

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.
- SAP application hostname: Enter the hostname of the source SAP application.
- 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

- SAP HANA replication mode: Select from *synchronous* or *asynchronous* mode for database replication.
- 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".
- 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.

SSL encryption

- If you do not want to use SSL encryption for database replication, select the box next to *I want to disable SSL encryption for database replication*.

- If you want to use SSL encryption for database replication or leave the box unchecked, a manual step – *Enable SSL on source for replication* in step group 4, must be completed to proceed with your migration workflow.

  1. Open the `global.ini` file on your source SAP HANA system.
  2. Set the replication property as follows.

     ```
     [system_replication_communication]
     enable_ssl=on
     ```

  3. Restart the database.

- **Note**

  SSL encryption is required for SAP NetWeaver on SAP HANA – scale-out and SAP HANA database – scale-out migration types.

For more information, see SAP help portal – Configure Secure Communication (TLS/SSL) Between Primary and Secondary Sites.

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 `SAPGUI`configuration 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 IAM admin 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.

Key	Value
access_key	*access_key of the credential*
secret_key	*secret_key of the credential*

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 IAM 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. Go to https://console.aws.amazon.com/migrationhub/orchestrator/, and 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”
   - the section called “Application”
   - the section called “Target environment configuration”

> **Note**
>
> You can customize the migration workflow once it has been created. For more information, see Migration workflows.

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. If you do not see the application in the list, you must define it in AWS Application Discovery Service.

Define applications

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

Topics

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**.

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".
- 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".

Rehost SQL Server on Amazon EC2

With **Rehost SQL server on Amazon EC2** template, you can rehost your SQL Server databases on an instance to Amazon EC2 using automated SQL Server backup and restore. You can also migrate databases that are encrypted with transparent data encryption (TDE). This template migrates User database items, Certificates, Logins and Agent Jobs that are associated with your SQL Server.

Topics

Prerequisites

You must set up the source environment before creating a migration workflow.

Topics

- Source environment setup

Source environment setup

- 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
```

- Install the `DBA.Tools` module on your Windows machine, with the following command.

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

Creating the migration workflow

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

> **Note**
>
> You can customize the migration workflow once it has been created. For more information, see Migration workflows.

Topics

- Application
- ServerId
- Source Environment Configuration

- Target Environment Configuration

Application

Select the application you want to migrate. If you do not see the application in the list, you must define it in AWS Application Discovery Service. An Application in this context is considered as a group of servers, and does not refer to applications running on top of your SQL server.

ServerId

Within the Application you defined in the AWS Application Discovery Service, select the serverId of the server which hosts your SQL server.

Source Environment Configuration

The details here help us to identify the details of your source SQL Server.

1. **TDE** - Check this checkbox if you have TDE enabled on your Databases. If you select this option, your certificates will be migrated to the target server.
2. **Migration Mode** - This template offers 3 distinct migrations depending on your use-case.
   a. " *Use only Full backup* " - The template will only create a full backup of your databases and restore it on your target.
   b. " *Use Full backup and Differential backup for Cutover* " - A full backup of your databases will be created and restored on the target, after which you can mark the databases readonly, and a differential backup and restore will be used to migrate the remainder of the data.
   c. " *Use Full backup, Differential backup for pre-cutover and T-Log backup for cutover* " - A full backup of your databases will be created and restored on the target. When you are getting ready for cutover, a differential backup and restore will be used to migrate the remainder of the data. Lastly, after you mark your databases readonly, Tail-Log backups will be used to migrate the remainder of the data.
3. **Allow Migration Without Direct Connect** - This template uploads backup files from your source instance to S3 using the AWS CLI. The database files are transmitted over an HTTPS to AWS S3. However, if you are not comfortable with the backup files travelling over the public Internet, we recommend using AWS Direct Connect with a Public VIF setup. If you are comfortable with this, please select this checkbox. The migration workflow will not create unless you check this checkbox or have the setup mentioned above.
4. **Source SQL Server database names** - The names of the SQL Databases that you would like to migrate.

5. **AWS ADS server ID for your application** - See “ServerId” section above.
6. **Source SQL Server instance name** - The name of your SQL server instance.
7. **Backup location** - As a part of the migration, this template needs to take backups of your SQL Server. The path specified here is where the backup files will be stored. Please ensure this is an absolute path and has enough space for a Full and Differential backup of your databases.

Target Environment Configuration

The details here help us to identify the details of your migration to your target server.

1. **Restore Logins** - Select this checkbox if you would like to migrate your SQL Server Logins to your target instance.
2. **Restore Agent Jobs** - Select this checkbox if you would like to migrate your SQL Server Agent Jobs to your target instance.

Running the migration workflow

- When configuring the Migration Hub Orchestrator plugin, ensure that the username that is provided to connect to your Windows machine has the

  SYSAdmin permission on the source SQL server instance.

Create AWS Profile on Source Server

- 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 user with the above policy attached.
- Configure a named profile for AWS Command Line Interface that uses the preceding IAM user. For more information, see Using AWS credentials . The credentials stored in the profile are used to upload your backups to a S3 bucket located in your account. (You will need the name of this profile when creating the workflow)

Create Target EC2 Instance

This template does not create your EC2 instance for you. To create this instance based on your requirements, we recommend one of the following:

- (*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 .
- Connect to the target EC2 instance and note the following:
  - Name of the SQL Server
  - Path to store data for the SQL Server
  - Path to store logs for the SQL Server
  - Path to store the downloaded backup files for the restore procedure. Please ensure this is large enough to hold the backup files of your database.

Configure Target Permissions

Once your EC2 instance is configured and your target SQL server is deployed, follow these steps:

- 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 Target SQL Server User

- Create a username in your target SQL server with SYSAdmin permission.
- Provide credentials in AWS Secrets Manager for the username 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.
     a. `username` - enter your username
     b. `password` - enter your password
  4. Select **Next** and enter a name for the key pair beginning with `migrationhub-orchestrator-secretname123` .
     a. 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** .

6. Copy the name of this secret and put the value into the manual step in the workflow.

FAQ

Q. **What does this template do?**

A. This template migrates User Database Items, Certificates, Agent Jobs and Logins from a source SQL server to a target SQL Server environment on Amazon EC2 .

Q. **Do I need to create the target SQL Server?**

A. Yes. This template focuses on data migration. You need to setup the target SQL server before using this template. Based on your requirements, we recommend using AWS Launch Wizard or AWS VM Import/Export Service to accomplish this.

Q. **What kind of backups do you use for migration?**

A. Based on your input, we use either only a full backup, a combination of full and differential backups or a combination of full, differential and tail-log backups for migration.

Q. **When do I need to put my databases in 'readonly' mode?**

A. Based on the type of migration selected there are different points to do this -

1. For full backup only migrations set the databases to readonly before begging the migration workflow.
2. For full and differential backup migrations, set the databases to read only when instructed to do so on Step 4.1 in the workflow.
3. For full, differential and tail-log backups, set the databases to read only when instructed to do so on Step 4.4 in the workflow.

These different configurations help us ensure we capture all the changes in your SQL server when migrating, to create parity between your source and target.

Q. **What security measures do you take while migrating data?**

A. Our goal is to handle data securely at all times. Backup files are transferred over an HTTPS connection to AWS S3, and then to your target EC2 instance. Certificate files are handled specially,

as the workflow creates a KMS key in your account, which is used to encrypt the certificate before transport, and de-crypt the certificate in your target environment.

Q. **What are the limitations of this template?**

A. This template will not do the following:

1. This template does not migrate System Databases or SQL Server properties.
2. This template can only migrate SQL logins. Any Windows level logins are not guaranteed to be migrated.
3. This template expects that while the workflow is running, you will not initiate a full-backup of the database yourself. If a full-backup is taken, it breaks the chain of backups used to restore your databases on the target server.

Q. **I ran into an error during a database connection step. What do I do?**

A. An error here indicates a problem with connecting to your SQL Server.

1. If this occurs on the source, ensure the user that was given to the plugin to connect to the machine has SYSADMIN permissions on your source SQL server.
2. If this happens on the target, ensure that the EC2 Instance ID provided during workflow creation is correct, and ensure the SQL credentials stored in your Secrets Manager Secret are correct.

Q. **I ran into an error during a database validation step. What do I do?**

A. An error here indicates an incompatibility between the inputs provided during workflow creation and the target environment. Look at the step logs located inside the S3 bucket shown in the error message to diagnose the issue and re-create the workflow with the appropriate inputs.

Q. **I ran into an error while a backup step was running. What do I do?**

A. If you run into an error during the backup steps, look at the step logs located inside the S3 bucket shown in the error message. Once you diagnose and fix the issue, please clean the appropriate backup directory on the source machine before re-trying this step.

Q. **I ran into an error while a restore step was running. What do I do?**

A. If you run into an error during the restore steps, look at the step logs located inside the S3 bucket shown in the error message. If you have taken a full backup after the workflow started, the

backup chain is broken and hence this workflow cannot be recovered. You will have to delete the workflow, wipe your target SQL server and re-create the workflow.

Replatform SQL on Amazon RDS

With **Replatform SQL server on Amazon RDS** template, you can replatform your SQL Server databases on an instance to Amazon RDS using native backup and restore. You can also migrate databases that are encrypted with transparent data encryption. This template migrates User database items, Certificates, Logins and Agent Jobs that are associated with your SQL Server.

Topics

Prerequisites

You must set up the source environment before creating a migration workflow.

Topics

Source environment setup

- 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
```

- Install the `DBA.Tools` module on your Windows machine, with the following command.

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

Creating the migration workflow

- Go to https://console.aws.amazon.com/migrationhub/orchestrator/
- Select **Create migration workflow**.
- On Choose a workflow template page, select **Replatform SQL server on Amazon RDS** template.
- Configure and submit your workflow to begin migration.

> **Note**
>
> You can customize the migration workflow once it has been created. For more information, see Migration workflows.

Topics

Application

Select the application you want to migrate. If you do not see the application in the list, you must define it in AWS Application Discovery Service. An Application in this context is considered the unit of migration, and does not refer to applications running on top of your SQL server.

ServerId

Within the Application you defined in the AWS Application Discovery Service, select the serverId of the server which hosts your SQL server.

Source Environment Configuration

The details here help us to identify the details of your source SQL Server.

- **TDE** - Check this checkbox if you have TDE enabled on your Databases. If you select this option, your certificates will be migrated to the target server.

- **Migration Mode** - This template offers 3 distinct migrations depending on your use-case.
- • "*Use only Full backup*" - The template will only create a full backup of your databases and restore it on your target.
  - "*Use Full backup and Differential backup for Cutover*" - A full backup of your databases will be created and restored on the target, after which you can mark the databases readonly, and a differential backup and restore will be used to migrate the remainder of the data.
  - "*Use Full backup, Differential backup for pre-cutover and T-Log backup for cutover*" - A full backup of your databases will be created and restored on the target. When you are getting ready for cutover, a differential backup and restore will be used to migrate the remainder of the data. Lastly, after you mark your databases readonly, Tail-Log backups will be used to migrate the remainder of the data.
- **Allow Migration Without Direct Connect** - This template uploads backup files from your source instance to S3 using the AWS CLI. The database files are transmitted over an HTTPS to AWS S3. However, if you are not comfortable with the backup files travelling over the public Internet, we recommend using AWS Direct Connect with a Public VIF setup. If you are comfortable with this, please select this checkbox. The migration workflow will not create unless you check this checkbox or have the setup mentioned above.
- **Source SQL Server database names** - The names of the SQL Databases that you would like to migrate.
- **AWS ADS server ID for your application** - See "ServerId" section above.
- **Source SQL Server instance name** - The name of your SQL server instance.
- **Backup location** - As a part of the migration, this template needs to take backups of your SQL Server. The path specified here is where the backup files will be stored. Please ensure this is an absolute path and has enough space for a Full and Differential backup of your databases.

Running the migration workflow

- When configuring the Migration Hub Orchestrator plugin, ensure that the username that is provided to connect to your Windows machine has the `SYSAdmin` permission on the source SQL server instance.

Create AWS Profile on Source Server

- 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 user with the above policy attached.
- Configure a named profile for AWS Command Line Interface that uses the preceding IAM user. For more information, see Using AWS credentials. The credentials stored in the profile are used to upload your backups to a S3 bucket located in your account. Note the name of this profile and enter it into the step when prompted.

Create your RDS Database

This template does not create your RDS instance for you.

- 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.
- Configure the option group for backup/restore and transparent data encryption in Amazon RDS, 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": [
                "*"
            ]
        }
    ]
}
```

- The trust policy for this role should be:

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

```
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Create attached EC2 Instance

- 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-*/*"
            ]
        }
    ]
}
```

- Ensure that your Amazon RDS instance can be reached from the created Amazon EC2 instance.
- This instance is used to connect to your RDS instance and run restore procedures.

Create Target SQL Server User

- Provide credentials in AWS Secrets Manager for the username and password for the admin user for your RDS 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.
4. • `username` - enter your username
   • `password` - enter your password
5. Select **Next** and enter a name for the key pair beginning with `migrationhub-orchestrator-secretname123`.
6. • The Secret ID must begin with the prefix `migrationhub-orchestrator-` and must only be followed by an alphanumeric value.
7. Select **Next** and then, select **Store**.
8. Copy the name of this secret and provide it to the workflow step when prompted.

FAQ

Q. **What does this template do?**

A. This template migrates User Database Items, Certificates, Agent Jobs and Logins from a source SQL server to a target SQL Server hosted on RDS.

Q. **Do I need to create the target SQL Server?**

A. Yes. This template focuses on data migration. You need to setup the target SQL server before using this template.

Q. **What kind of backups do you use for migration?**

A. Based on your input, we use either only a full backup, a combination of full and differential backups or a combination of full, differential and tail-log backups for migration.

Q. **When do I need to put my databases in 'readonly' mode?**

A. Based on the type of migration selected there are different points to do this -

- For full backup only migrations set the databases to readonly before begging the migration workflow.
- For full and differential backup migrations, set the databases to read only when instructed to do so on Step 4.1 in the workflow.
- For full, differential and tail-log backups, set the databases to read only when instructed to do so on Step 4.4 in the workflow.

These different configurations help us ensure we capture all the changes in your SQL server when migrating, to create parity between your source and target.

Q.**What security measures do you take while migrating data?**

A. Our goal is to handle data securely at all times. Backup files are transferred over an HTTPS connection to AWS S3, and then to your EC2 instance, before being restored to RDS. Certificate files are handled specially, as the workflow creates a KMS key in your account, which is used to encrypt the certificate before transport, and de-crypt the certificate in your target environment.

Q.**Why do I need to create an EC2 instance?**

A. The EC2 instance you create is used to run the restore procedures on your RDS endpoint. It is designed this way so that your RDS endpoint does not need to be exposed to the public internet for the restore procedure.

Q.**What are the limitations of this template?**

A. This template will not do the following:

- This template does not migrate System Databases or SQL Server properties.
- This template can only migrate SQL logins. Any Windows level logins are not guaranteed to be migrated.
- This template expects that while the workflow is running, you will not initiate a full-backup of the database yourself. If a full-backup is taken, it breaks the chain of backups used to restore your databases on the target server.
- This template can only migrate databases that have the “DBO” set as a sql user or AD user. If the database is owner is a Windows level user which is not available in the RDS environment, the database will be inaccessible when restored on RDS.

Q. **I ran into an error during a database connection step. What do I do?**

A. An error here indicates a problem with connecting to your SQL Server.

- If this occurs on the source, ensure the user that was given to the plugin to connect to the machine has SYSADMIN permissions on your source SQL server.
- If this happens on the target, ensure that the EC2 Instance ID provided during workflow creation has connectivity to your RDS Endpoint, and ensure the SQL credentials stored in your Secrets Manager Secret are correct.

Q. **I ran into an error during a database validation step. What do I do?**

A. An error here indicates an incompatibility between the inputs provided during workflow creation and the target environment. Look at the step logs located inside the S3 bucket shown in the error message to diagnose the issue and re-create the workflow with the appropriate inputs.

Q. **I ran into an error while a backup step was running. What do I do?**

A. If you run into an error during the backup steps, look at the step logs located inside the S3 bucket shown in the error message. Once you diagnose and fix the issue, please clean the appropriate backup directory on the source machine before re-trying this step.

Q. **I ran into an error while a restore step was running. What do I do?**

A. If you run into an error during the restore steps, look at the step logs located inside the S3 bucket shown in the error message. If you have taken a full backup after the workflow started, the backup chain is broken and hence this workflow cannot be recovered. You will have to delete the workflow, wipe your target SQL server and re-create the workflow.

Replatform applications to Amazon ECS

You can use the *Replatform applications to Amazon ECS* template in Migration Hub Orchestrator to replatform your .NET and Java applications to containers. The applications can be sourced from EC2 instances or application artifacts that are uploaded to Amazon S3. You can deploy containerized applications on Amazon Elastic Container Service (Amazon ECS) on AWS Fargate using one application per container or with multiple applications in a single container.

Topics

Prerequisites

The prerequisites required to use this template depend on the source type that you will specify in the workflow. Your application source can be one or more Amazon EC2 instances or application artifacts that you uploaded to Amazon S3.

The following prerequisites must be met to successfully replatform your applications with this template.

Source type of Amazon EC2

The following prerequisites apply when you specify the source type of Amazon EC2 while using this template.

Topics

Application support and compatibility

Before using this template on Amazon EC2 instances, ensure that your servers and applications are supported for App2Container. For more information, see App2Container compatibility and Applications you can containerize using AWS App2Container in the *AWS App2Container User Guide*.

> **Note**
>
> You don't need to install Docker on your application server to use this template.

SSM agent

To use this template with Amazon EC2 instances, they must be managed nodes in AWS Systems Manager (Systems Manager). The SSM agent is required for your instances to become managed nodes. Some AMIs have the SSM agent preinstalled, while others require manual installation. For more information on verifying if the SSM agent is installed, and how to manually install it if required, see Amazon Machine Images (AMIs) with SSM Agent preinstalled in the *AWS Systems Manager User Guide*.

IAM instance profile for EC2 instances

This template requires that your EC2 instances have an instance profile role with the necessary permissions attached. The permissions provided by an instance profile are used by your EC2 instances. You can create a new IAM instance profile with the required permissions, or add them to an existing role used by the instance. An instance profile can only contain one IAM role. The IAM role can contain one or more policies. For more information, see Instance profiles and Work with IAM roles in the *Amazon Elastic Compute Cloud User Guide*.

To configure the required Systems Manager core functionality for your EC2 instances, you can attach the AWS managed policy `AmazonSSMManagedInstanceCore` to your instance profile. For more information about instance permissions for Systems Manager, see Step 1: Configure instance permissions for Systems Manager in the *AWS Systems Manager User Guide*.

The following permissions must also be added to the IAM role used by your instance profile. You can create a new policy with the following JSON policy document and then attach the policy to your instance profile role. For more information, see Creating IAM policies in the *AWS Identity and Access Management User Guide*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3BucketAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ]
        },
        {
            "Sid": "S3ObjectAccess",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::*/application-transformation*"
            ]
```

```
        },
        {
            "Sid": "KmsAccess",
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:*:*:key/*"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "s3.*.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "TelemetryAccess",
            "Effect": "Allow",
            "Action": [
                "application-transformation:PutMetricData",
                "application-transformation:PutLogData"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Source type of Amazon S3

The following prerequisites apply when you specify the source type of Amazon S3 while using this template.

Topics

- Amazon S3 buckets
- Application artifacts

Amazon S3 buckets

This template requires that you have an Amazon S3 bucket for the S3 input path and the Amazon S3 output path. You can create different buckets for the input and output S3 locations. The workflow requires that the application artifacts be uploaded to an Amazon S3 bucket beginning with the following prefix:

```
S3://bucket-name/application-transformation
```

For more information on creating an Amazon S3 bucket, see Creating a bucket in the *Amazon Simple Storage Service User Guide*.

Application artifacts

This template requires that you have application artifacts available in an Amazon S3 bucket in the bucket prefix mentioned previously in order to replatform the application. App2Container has the `AWSApp2Container-ReplatformApplications` AWS Systems Manager Automation runbook for use on Amazon EC2 instances which generates the required application artifacts. For more information, see App2Container Automation runbook in the *AWS App2Container User Guide*.

When using Amazon S3 as the source type, you must upload these artifacts to the S3 bucket you created with the required application artifact files. The following files are required:

- `replatform-definition.json`
- `analysis.json`
- `ContainerFiles.tar` or `ContainerFiles.zip`

The `replatform-definition.json` file should resemble the following:

```
{
    "version": "1.0",
    "workloads": [
        {
            "containers": [
                {
                    "applications": [
                        {
                            "applicationOverrideS3Uri": "s3://bucket-name/application-
transformation/path-to-application-artifacts/"
                        }
```

```
                    ]
                }
            ]
        }
    ]
}
```

Required IAM resources

Multiple resources must have the required permissions in order to use this template. Ensure that you have the following required policies and roles created.

Topics

- IAM policy for users and roles
- IAM policies and roles for Amazon ECS
- (Optional) KMS key policy

IAM policy for users and roles

Your user or role must have the required permissions to use this template. You can add this policy inline, or create and add this policy to your user, group, or role. For more information, see Creating IAM policies and Choosing between managed policies and inline policies in the *AWS Identity and Access Management User Guide*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AssessmentIAMRoleAccess",
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:GetInstanceProfile",
                "iam:GetRole"
  ],
  "Resource": "*"
        },
        {
            "Sid": "ApplicationTransformationAccess",
            "Effect": "Allow",
            "Action": [
```

```
                "application-transformation:StartRuntimeAssessment",
                "application-transformation:GetRuntimeAssessment",
                "application-transformation:PutLogData",
                "application-transformation:PutMetricData",
                "application-transformation:StartContainerization",
                "application-transformation:GetContainerization",
                "application-transformation:StartDeployment",
                "application-transformation:GetDeployment"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AssessmentEc2ReadAccess",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AssessmentIAMRoleAccess",
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:GetInstanceProfile"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AsssessmentSSMSendCommandAccess",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ssm:*::document/AWS-RunRemoteScript"
            ]
        },
        {
            "Sid": "AsssessmentSSMDescribeAccess",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceInformation",
```

```
                "ssm:ListCommandInvocations",
                "ssm:GetCommandInvocation"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:*"
            ]
        },
        {
            "Sid": "S3ObjectAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::*/application-transformation*"
            ]
        },
        {
            "Sid": "S3ListAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Sid": "KmsAccess",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:*::*"
        },
        {
            "Sid": "EcrAccess",
            "Effect": "Allow",
            "Action": [
                "ecr:CreateRepository",
                "ecr:GetLifecyclePolicy",
                "ecr:GetRepositoryPolicy",
```

```
                "ecr:ListImages",
                "ecr:ListTagsForResource",
                "ecr:TagResource",
                "ecr:UntagResource"
            ],
            "Resource": "arn:*:ecr:*:*:repository/*"
        },
        {
            "Sid": "EcrPushAccess",
            "Effect": "Allow",
            "Action": [
                "ecr:InitiateLayerUpload",
                "ecr:PutImage",
                "ecr:UploadLayerPart",
                "ecr:CompleteLayerUpload",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer"
            ],
            "Resource": "arn:*:ecr:*:*:repository/*"
        },
        {
            "Sid": "EcrAuthAccess",
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ContainerizeKmsCreateGrantAccess",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant"
            ],
            "Resource": "arn:aws:kms:*::*",
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
        },
        {
            "Sid": "CloudformationExecutionAccess",
            "Effect": "Allow",
```

```
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:UpdateStack"
            ],
            "Resource": [
                "arn:*:cloudformation:*:*:stack/application-transformation-*"
            ]
        },
        {
            "Sid": "GetECSSLR",
            "Effect": "Allow",
            "Action": "iam:GetRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/ecs.amazonaws.com/
AWSServiceRoleForECS"
        },
        {
            "Sid": "CreateEcsServiceLinkedRoleAccess",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/ecs.amazonaws.com/
AWSServiceRoleForECS",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "ecs.amazonaws.com"
                }
            }
        },
         {
            "Sid": "CreateElbServiceLinkedRoleAccess",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
elasticloadbalancing.amazonaws.com/AWSServiceRoleForElasticLoadBalancing",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "elasticloadbalancing.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CreateSecurityGroupAccess",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSecurityGroup"
```

```
            ],
            "Resource": "*"
        },
        {
            "Sid": "Ec2CreateAccess",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateInternetGateway",
                "ec2:CreateKeyPair",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSubnet",
                "ec2:CreateTags",
                "ec2:CreateVpc"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Ec2ModifyAccess",
            "Effect": "Allow",
            "Action": [
                "ec2:AssociateRouteTable",
                "ec2:AttachInternetGateway",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:DeleteTags",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:RevokeSecurityGroupIngress"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IAMPassRoleAccess",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*"
        },
        {
            "Sid": "EcsCreateAccess",
            "Effect": "Allow",
            "Action": [
                "ecs:CreateCluster",
```

```
                "ecs:CreateService",
                "ecs:RegisterTaskDefinition"
            ],
            "Resource": "*"
        },
        {
            "Sid": "EcsModifyAccess",
            "Effect": "Allow",
            "Action": [
                "ecs:TagResource",
                "ecs:UntagResource",
                "ecs:UpdateService"
            ],
            "Resource": "*"
        },
        {
            "Sid": "EcsReadTaskDefinitionAccess",
            "Effect": "Allow",
            "Action": [
                "ecs:DescribeTaskDefinition"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "cloudformation.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudwatchCreateAccess",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:TagResource",
                "logs:PutRetentionPolicy"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/ecs/containerinsights/*:*",
                "arn:aws:logs:*:*:log-group:/aws/ecs/container-logs/*:*"
            ]
        },
        {
            "Sid": "CloudwatchGetAccess",
            "Effect": "Allow",
```

```
            "Action": [
                "logs:GetLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/ecs/containerinsights/*:*",
                "arn:aws:logs:*:*:log-group:/aws/ecs/container-logs/*:*"
            ]
        },
        {
            "Sid": "ReadOnlyAccess",
            "Effect": "Allow",
            "Action": [
                "cloudformation:DescribeStacks",
                "cloudformation:ListStacks",
                "clouddirectory:ListDirectories",
                "ds:DescribeDirectories",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeImages",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ecr:DescribeImages",
                "ecr:DescribeRepositories",
                "ecs:DescribeClusters",
                "ecs:DescribeServices",
                "ecs:DescribeTasks",
                "ecs:ListTagsForResource",
                "ecs:ListTasks",
                "iam:ListRoles",
                "s3:GetBucketLocation",
                "s3:GetBucketVersioning",
                "s3:ListAllMyBuckets",
                "secretsmanager:ListSecrets",
                "acm:DescribeCertificate",
                "acm:GetCertificate",
                "ssm:GetParameters"
            ],
            "Resource": "*"
        },
```

```
        {
            "Sid": "ElasticLoadBalancingCreateAccess",
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:CreateListener",
                "elasticloadbalancing:CreateLoadBalancer",
                "elasticloadbalancing:CreateTargetGroup",
                "elasticloadbalancing:CreateRule"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ElasticLoadBalancingModifyAccess",
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:AddTags",
                "elasticloadbalancing:ModifyTargetGroup",
                "elasticloadbalancing:ModifyTargetGroupAttributes"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ElasticLoadBalancingGetAccess",
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:DescribeLoadBalancerAttributes",
                "elasticloadbalancing:DescribeTags",
                "elasticloadbalancing:DescribeTargetGroups",
                "elasticloadbalancing:DescribeRules",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeLoadBalancers"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Route53CreateAccess",
            "Effect": "Allow",
            "Action": [
                "route53:CreateHostedZone"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Route53ModifyAccess",
```

```
            "Effect": "Allow",
            "Action": [
                "route53:ChangeTagsForResource",
                "route53:ChangeResourceRecordSets",
                "route53:GetChange",
                "route53:GetHostedZone",
                "route53:ListResourceRecordSets",
                "route53:CreateHostedZone",
                "route53:ListHostedZonesByVPC"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SsmMessagesAccess",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeSessions",
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ServiceDiscoveryCreateAccess",
            "Effect": "Allow",
            "Action": [
                "servicediscovery:CreateService",
                "servicediscovery:CreatePrivateDnsNamespace",
                "servicediscovery:UpdatePrivateDnsNamespace",
                "servicediscovery:TagResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ServiceDiscoveryGetAccess",
            "Effect": "Allow",
            "Action": [
                "servicediscovery:GetNamespace",
                "servicediscovery:GetOperation",
                "servicediscovery:GetService",
                "servicediscovery:ListTagsForResource"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

IAM policies and roles for Amazon ECS

To deploy your containerized applications on Amazon ECS, you must create IAM policies and roles in your Amazon ECS tasks. For more information about these IAM resources for Amazon ECS and how to create them, see Task execution IAM role and Task IAM role in the *Amazon Elastic Container Service Developer Guide*.

(Optional) KMS key policy

You can use AWS KMS to encrypt resources used by this template. If you create a KMS key to use with this template, we recommend that you use the following least-privilege permissions for your key policy. For more information, see Key policies in AWS KMS in the *AWS Key Management Service Developer Guide*.

```
{
    "Sid": "KmsAccess",
    "Effect": "Allow",
    "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:*:*:key/*"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": [
                "s3.*.amazonaws.com"
            ]
        }
    }
}
```

Configuring a workflow

You must configure the workflow for the template in order to replatform your application.

To create a workflow using the template

1. Access the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/.
2. In the left navigation pane, under **Orchestrate**, choose **Create workflow**.
3. On the **Choose a workflow template** page, choose the **Replatform applications to Amazon ECS** template.
4. On the **Configure your workflow** page, enter values for the following:

   a. For **Workflow details**, enter values for the following:

      i. For **Name**, enter a name for your migration workflow.

      ii. (Optional) For **Description**, enter a description for the workflow you are creating.

   b. For **Source environment configuration**, specify the following:

      i. For **Source Region**, choose the Region from the dropdown list in which you have EC2 instances hosting applications you want to replatform or the S3 bucket containing your application artifacts.

      ii. For **Source type**, choose **EC2 instances** if your applications you want to replatform are in EC2 instances, or **S3 location** if your application artifacts are in an S3 bucket.

         A. If you chose **EC2 instances**, under **Select from EC2 instances**, select the instances which have the applications you want to replatform.

         B. If you chose **S3 location**, under **Specify input path in *Region***, enter the path to your `replatform-definition.json` file in the S3 bucket. Your other required application artifacts should also be in this bucket. You can also choose **Browse S3** to specify the path by navigating to it in the console. The path should resemble the following:

```
S3://bucket-name/application-transformation/replatform-definition.json
```

   c. For **Specify S3 output path**, enter the path of your S3 bucket using `S3://` syntax. You can also choose **Browse S3** to specify the path by navigating to it in the console. The path should resemble the following example:

```
S3://bucket-name/application-transformation
```

d. (Optional) For **Tags**, choose **Add new tag** and enter any desired key-value pairs for your resources that are created by this workflow.

e. Choose **Next**.

f. On the Review and submit page, ensure the provided details for the workflow are correct, then choose **Create**.

Creating a migration workflow doesn't take action on your resources. You will need to run the workflow as detailed in the following section.

> **Note**
>
> You can customize the migration workflow once it has been created. For more information, see Migration workflows.

Running a workflow

With the workflow created, you can now run it to replatform your applications.

To run a workflow

1. Access the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/.
2. In the left navigation pane, under **Orchestrate**, choose **Workflows**.
3. On the **Workflows** page, choose your workflow and then choose **View details**.
4. Choose **Run** to run the workflow.

> **Important**
>
> Some steps might require additional action to complete. All steps must be completed in order to replatform your application. The following section details this process.

Combining multiple applications in one container

If you are combining multiple applications from your source server to one container, there are additional requirements for the workflow. You can specify this option when you are configuring

your workflow for the template **Combine applications in one container** in Completing the required steps.

> **Note**
>
> If you are replatforming a single application to one container, the following process is not required.

Python script

You can use the following content to create a Python script on your application server. The script helps you create the required configuration file to containerize multiple application to one container.

- This script only supports applications running on Linux.
- This script only supports Regions that are enabled by default.

```
import boto3
import json
import tarfile
import os
import subprocess
import shutil
from pathlib import Path
from argparse import ArgumentParser
from urllib.parse import urlparse


ANALYSIS_INFO_JSON = "analysis.json"
CONTAINER_FILES_TAR = "ContainerFiles.tar"
COMBINED_APPLICATION = "CombinedApplication"
TAR_BINARY_PATH = "/usr/bin/tar"

def get_bucket(s3path):
    o = urlparse(s3path, allow_fragments=False)
    return o.netloc

def get_key(s3path):
    o = urlparse(s3path, allow_fragments=False)
    key = o.path
```

```
    if key.startswith('/'):
        key = key[1:]

    if not key.endswith('/'):
        key += '/'

    return key

def format_path(path):
    if not path.endswith('/'):
        path += '/'
    return path

def upload_to_s3(s3_output_path, workflow_id, analysis_file, container_file):
    s3 = boto3.client('s3')

    bucket = get_bucket(s3_output_path)
    key = get_key(s3_output_path)

    analysis_object = key + workflow_id + "/" + COMBINED_APPLICATION +  "/" +
 ANALYSIS_INFO_JSON
    container_object = key + workflow_id + "/" +  COMBINED_APPLICATION + "/" +
 CONTAINER_FILES_TAR

    s3.upload_file(analysis_file, bucket, analysis_object)
    s3.upload_file(container_file, bucket, container_object)

def download_from_s3(region, s3_paths_list, workspace_s3_download_path):

    s3 = boto3.client('s3')

    dir_number=1
    workspace_s3_download_path = format_path(workspace_s3_download_path)

    for s3_path in s3_paths_list:
        download_path = workspace_s3_download_path + 'd' + str(dir_number)
        dir_number += 1
        Path(download_path).mkdir(parents=True, exist_ok=True)

        bucket = get_bucket(s3_path)
        key = get_key(s3_path)

        analysis_key = key + ANALYSIS_INFO_JSON
```

```
        container_files_key = key + CONTAINER_FILES_TAR

        download_analysis_path = download_path + '/' + ANALYSIS_INFO_JSON
        download_container_files_path = download_path + '/' + CONTAINER_FILES_TAR

        s3.download_file(bucket, analysis_key, download_analysis_path)
        s3.download_file(bucket, container_files_key, download_container_files_path)

def get_analysis_data(analysis_json):
    data = ""
    with open(analysis_json) as json_data:
        data = json.load(json_data)
        json_data.close()
    return data

def combine_container_files(workspace_path, count, output_path):
    if not workspace_path.endswith('/'):
        workspace_path += '/'

    for dir_number in range(1, count+1):
        container_files_path = workspace_path + 'd' + str(dir_number)
        container_file_tar = container_files_path + '/' + CONTAINER_FILES_TAR

        extract_tar(container_file_tar, output_path)

def tar_container_files(workspace_path, tar_dir):
    os.chdir(workspace_path)
    subprocess.call([TAR_BINARY_PATH, 'czf', "ContainerFiles.tar", "-C", tar_dir, "."])

def combine_analysis(workspace_path, count, analysis_output_path, script_output_path):
    if not workspace_path.endswith('/'):
        workspace_path += '/'

    #First analysis file is used as a template
    download_path = workspace_path + 'd' + str(1)
    analysis_json = download_path + '/' + ANALYSIS_INFO_JSON
    first_data = get_analysis_data(analysis_json)

    cmd_list = []
    ports_list = []

    for dir_number in range(1, count+1):
        download_path = workspace_path + 'd' + str(dir_number)
        analysis_json = download_path + '/' + ANALYSIS_INFO_JSON
```

```
        data = get_analysis_data(analysis_json)

        cmd = data['analysisInfo']['cmdline']
        cmd = " ".join(cmd)
        cmd_list.append(cmd)

        ports = data['analysisInfo']['ports']
        ports_list += ports

    start_script_path = create_startup_script(cmd_list, script_output_path)
    os.chmod(start_script_path, 0o754)

    start_script_filename = '/' + Path(start_script_path).name
    cmd_line_list = [start_script_filename]

    first_data['analysisInfo']['cmdline'] = cmd_line_list
    first_data['analysisInfo']['ports'] = ports_list

    analysis_output_path = format_path(analysis_output_path)
    analysis_output_file = analysis_output_path + '/' + ANALYSIS_INFO_JSON
    write_analysis_json_data(first_data, analysis_output_file)

def write_analysis_json_data(data, output_path):
    with open(output_path, 'w') as f:
        json.dump(data, f)

def create_startup_script(cmd_list, output_path):
    start_script_path = output_path + '/start_script.sh';
    with open (start_script_path, 'w') as rsh:
        rsh.write('#! /bin/bash\n')
        for cmd in cmd_list:
            rsh.write('nohup ' + cmd + ' >> /dev/null 2>&1 &\n')
        rsh.close()
    return start_script_path

def extract_tar(tarFilePath, extractTo):
    os.chdir(extractTo)
    subprocess.call([TAR_BINARY_PATH, 'xvf', tarFilePath])

def validate_args(args):
    MIN_COUNT = 2
    MAX_COUNT = 5
    s3_paths_count = len(args.s3_input_path)
    if (s3_paths_count < MIN_COUNT):
```

```
        print("ERROR: input_s3_path needs atleast " + str(MIN_COUNT) +" s3 paths")
        exit(0)

    if (s3_paths_count > MAX_COUNT):
        print("ERROR: Max input_s3_paths is " + str(MAX_COUNT))
        exit(0)


def cleanup_workspace(temp_workspace):
    yes = "YES"
    ack = input("Preparing workspace. Deleting dir and it's contents '" +
 temp_workspace + "'. Please confirm with 'yes' to procced.\n")
    if (ack.casefold() == yes.casefold()):
        if (os.path.exists(temp_workspace) and os.path.isdir(temp_workspace)):
            shutil.rmtree(temp_workspace)
    else:
        print("Please confirm with 'yes' to continue. Exiting.")
        exit(0)

def main():
    parser = ArgumentParser()
    parser.add_argument('--region', help='Region selected during A2C workflow
 creation', required=True)
    parser.add_argument('--workflow_id', help='Migration Hub Orchestrator workflowId',
 required=True)
    parser.add_argument('--s3_output_path', help='S3 output path given while creating
 the workflow', required=True)
    parser.add_argument('--s3_input_path', nargs='+', help='S3 paths which has
 application artifacts to combine', required=True)
    parser.add_argument('--temp_workspace', nargs='?', default='/tmp', type=str,
 help='Temp path for file downloads')
    args = parser.parse_args()

    validate_args(args)

    #prepare workspace
    temp_workspace = format_path(args.temp_workspace)
    temp_workspace += 'mho_workspace'

    #cleanup tmp workspace
    cleanup_workspace(temp_workspace)

    #create workspace directories
    Path(temp_workspace).mkdir(parents=True, exist_ok=True)
```

```
    apps_count = len(args.s3_input_path)
    temp_output_container_files = temp_workspace + '/outputs/containerfiles'
    os.makedirs(temp_output_container_files, exist_ok=True)
    temp_workspace_output = temp_workspace + "/outputs"

    #download files
    download_from_s3(args.region, args.s3_input_path, temp_workspace)

    #combine files
    combine_container_files(temp_workspace, apps_count, temp_output_container_files)
    combine_analysis(temp_workspace, apps_count, temp_workspace_output,
 temp_output_container_files)
    tar_container_files(temp_workspace_output, temp_output_container_files)

    #prepare upload
    analysis_json_file_to_upload = temp_workspace_output + "/" + ANALYSIS_INFO_JSON
    container_files_to_upload = temp_workspace_output + "/" + CONTAINER_FILES_TAR
    upload_to_s3(args.s3_output_path, args.workflow_id, analysis_json_file_to_upload,
 container_files_to_upload)

if __name__=="__main__":
    main()
```

To run the Python script

1. Install Python 3.8 or later on your application server. For information on how to get the latest version of Python, see the official Python documentation.
2. Install AWS SDK for Python (Boto3). For more information, see AWS SDK for Python (Boto3).
3. Configure Boto3 credentials. For more information, see Credentials.
4. Run the `combine_applications.py` script while specifying values for the following parameters:

    a. **region** – The Region where your Amazon S3 bucket is located.

    b. **workflow_id** – The workflow ID.

    c. **s3_input_path** – The S3 path that has the S3 artifacts uploaded that need to be combined.

    d. **s3_output_path** – The output path given when creating the workflow.

    e. **temp_workspace** – The workspace directory to use. The default is `/tmp/`.

The following example demonstrates running the script with the required parameters:

```
python3 combine_applications.py --region us-west-2 \
    --workflow_id mw-abc123 \
    --s3_output_path s3://bucket-name/application-transformation/mw-abc123/
CombinedApplications \
    --s3_input_path s3://bucket-name/application-transformation/appname1/ s3://bucket-
name/application-transformation/appname2/
```

Once the script has completed, the application artifacts will be uploaded to Amazon S3 with a path similar to the following:

```
s3://bucket-name/application-transformation/mw-abc123/CombinedApplications
```

Completing the required steps

The workflow will require additional input for certain steps in order to complete them. The workflow might take some time to reach this status before you can take action on the steps.

To complete steps for a workflow

1. Access the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/.
2. In the left navigation pane, under **Orchestrate**, choose **Workflows**.
3. On the **Workflows** page, choose your workflow and then choose **View details**.
4. In the **Steps** tab, choose **Expand all**. Steps with a **Status** of **User attention required** need additional input to complete the step.
5. Choose the step which requires further input, choose **Actions**, **Change status**, and then choose **Completed**.

    a. The **Analyze** step requires the following input:

        i. For **Applications**, from the dropdown list, select the applications that you want to replatform.

        ii. For **Containerization options**, choose either **One application per container** to provision one application per container, or **Combine applications in one container** to provision all applications in one container. For more information on the requirements

to combine applications in one container, see Combining multiple applications in one container.

iii. Choose **Confirm** to complete the step.

b. The **Deploy** step requires the following input:

i. For **VPC ID**, enter the ID of the VPC to use for deployment.

ii. For **ECS task execution IAM role ARN**, choose the ARN of the ECS task execution IAM role used to make AWS API calls on your behalf.

iii. (Optional) For **Task role ARN**, choose the ARN of the role to be assumed by Amazon ECS tasks.

iv. (Optional) For **Cluster name**, enter a name to use for the ECS cluster.

v. (Optional) For **CPU**, choose the number of CPU units the Amazon ECS container agent should reserve for the container.

vi. (Optional) For **Memory**, enter amount of memory to allocate to the container, specified in GB.

c. Choose **Confirm** to complete the step.

6. On the **Workflows** page, under **Migration workflows**, verify that the overall status of the workflow is **Complete**.

Import virtual machine images to AWS

You can use the *Import virtual machine images to AWS* template to convert existing images to Amazon Machine Images (AMI) for Amazon EC2.

Prerequisites

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

AWS Identity and Access Management (IAM) requirements

You need to create resources in IAM for both Migration Hub Orchestrator and VM Import/Export:

- Create an IAM user and attach the required policies to use Migration Hub Orchestrator. For more information, see Create an IAM user
- Create an IAM user and a service role, and attach required policies to use VM Import/Export. For more information, see Required permissions.

VM Import/Export requirements

You might need to perform additional tasks to prepare your AWS environment before importing your image. For more information, see VM Import/Export Requirements.

Upload images to Amazon S3

Create an Amazon S3 bucket, and add the image files you want to import into the bucket. For more information about creating an Amazon S3 bucket, see Creating a bucket.

The following considerations and limitations apply:

- The Amazon S3 bucket must be in the same Region as the AWS account in which you are using Migration Hub Orchestrator.
- You must have separate folders for each image file format you want to upload in your S3 bucket.
- Migration Hub Orchestrator supports importing the following image file formats:
  - OVA
  - RAW
  - VHD
  - VHDX
  - VMDK

Each image file type has additional requirements for the S3 bucket, file name, and workflow.

OVA files

The following considerations apply when you import OVA files:

- The folder must be named with the prefix `migrationhub-orchestrator-vmie-`*`folder-name`* and must only contain one OVA file.
- The S3 object must end with `.ova`.
- Only one OVA file can be added in one import task.
- You can add up to five import tasks in the workflow.

RAW files

The following considerations apply when you import RAW files:

- The folder must be named with the prefix `migrationhub-orchestrator-vmie-`*folder-name* and must only contain one RAW file.
- The S3 object must end with `.raw`.
- Only one RAW file can be added in one import task.
- You can add up to five import tasks in the workflow.

VMDK files

The following considerations apply when you import VMDK files:

- The folder must be named with the prefix `migrationhub-orchestrator-vmie-`*folder-name*.
- The S3 object must end with `.vmdk`.
- The folder must only contain VMDK files.
- The folder can contain up to 21 VMDK files.

VHD files

The following considerations apply when you import VHD files:

- The folder must be named with the prefix `migrationhub-orchestrator-vmie-`*folder-name*.
- The S3 object must end with `.vhd`.
- The folder must only contain VHD files.
- The folder can contain up to 21 VHD files.

VHDX files

The following considerations apply when you import VHDX files:

- The folder must be named with the prefix `migrationhub-orchestrator-vmie-`*folder-name*.
- The S3 object must end with `.vhdx`.
- The folder must only contain VHDX files.
- The folder can contain up to 21 VHDX files.

Create a workflow

1. Go to https://console.aws.amazon.com/migrationhub/orchestrator/, select **Create migration workflow**.
2. On Choose a workflow template page, select **Import virtual images to AWS** template.
3. Configure and submit your workflow to begin the VM import.

   - the section called “Details”
   - the section called “Source environment configuration”
   - the section called “Target environment configuration”

> **Note**
>
> You can customize the migration workflow once it has been created. For more information, see Migration workflows.

Details

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

Source environment configuration

You need to specify the following parameters to configure your workflow.

- **Server IP** – This is an optional parameter where you can provide the IP address of the on-premises server that needs to be migrated. You must setup the Migration Hub Orchestrator plugin on providing the IP address. This enables Migration Hub Orchestrator to run a validation and detect any failure scenarios before import.
- **Disk container** – You must specify the Amazon S3 path to your images that you set up in the section called “Prerequisites”. See the following examples for more details.

  OVA files

  You can use either of the following path style examples for the disk container parameter.

s3://*bucket-name*/migrationhub-orchestrator-vmie-*folder-name*

s3://*bucket-name*/migrationhub-orchestrator-vmie-*folder-name*/*file-name*.ova

RAW files

You can use either of the following path style examples for the disk container parameter.

s3://*bucket-name*/migrationhub-orchestrator-vmie-*folder-name*

s3://*bucket-name*/migrationhub-orchestrator-vmie-*folder-name*/*file-name*.raw

VMDK files

You can use either of the following path style examples for the disk container parameter.

s3://*bucket-name*/migrationhub-orchestrator-vmie-*folder-name*

s3://*bucket-name*/migrationhub-orchestrator-vmie-*folder-name*/*file-name*.vmdk

VHD files

You can use either of the following path style examples for the disk container parameter.

s3://*bucket-name*/migrationhub-orchestrator-vmie-*folder-name*

s3://*bucket-name*/migrationhub-orchestrator-vmie-*folder-name*/*file-name*.vhd

VHDX files

You can use either of the following path style examples for the disk container parameter.

s3://*bucket-name*/migrationhub-orchestrator-vmie-*folder-name*

s3://*bucket-name*/migrationhub-orchestrator-vmie-*folder-name*/*file-name*.vhdx

When more than one disk container is added, Migration Hub Orchestrator runs the workflow sequentially. If the first disk container fails, you must recover the failed container or create a new workflow.

- **Add new item** – You can add up to five image tasks for the workflow.

Target environment configuration

This section of the Import virtual machine images to AWS template workflow has optional parameters for licensing. For more information, refer to the following documentation.

- Licensing options
- Boot modes

Custom templates

You can customize the templates provided by AWS Migration Hub Orchestrator for your use case and save them for reuse. You must first use a template provided by Migration Hub Orchestrator to create a migration workflow. Once the migration workflow is created, you can proceed with making customizations to the workflow.

To help ensure that the custom template works as expected, we recommend that you run the customized workflow after making your changes. Once the updated migration workflow completes successfully, you can save your changes as a new custom template.

Topics

Prerequisites

The following prerequisites must be met to create custom templates:

- If the AWS managed template that you will customize requires the Migration Hub Orchestrator plugin, you must configure it before running the workflow. For more information, see Run the workflow.
- To create a migration workflow, you must have the resources required by your desired AWS managed template available.
- If you are going to create steps in the workflow of the **Automated** type, ensure that the scripts are accessible using one of the following methods:

- If the scripts for your custom template will be sourced from an Amazon Simple Storage Service (Amazon S3) location, the script files must be uploaded to an Amazon S3 bucket with the prefix `migrationhub-orchestrator-`. For more information, see Organizing objects using prefixes in the *Amazon Simple Storage Service User Guide*.
- If the scripts for your template will be uploaded to the AWS Management Console, you must be able to upload a copy of the script from the device that you are using.

Creating custom templates

You can use the following steps to create a custom template using the Migration Hub Orchestrator Console or the AWS CLI.

Console

To create a custom template using the Migration Hub Orchestrator console

1. Sign in to the AWS Management Console and open the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/?region=us-east-1#/templates.
2. Choose the template from the list that you want to customize. For more information about the available templates, see Templates.
3. Choose **Create workflow**.
4. Customize the workflow steps and step groups of the template as necessary. For more information about how to modify a workflow, see Migration workflows.
5. (Recommended) Once you have finished modifying the workflow, choose **Run** to start the workflow and ensure it completes successfully. For more information, see Running workflows.
6. Choose **Save as a template**.
7. For **Name**, enter a name for the custom template.
8. (Optional) For **Description**, enter a description for the custom template.
9. (Optional) Add tags to your custom template:
    a. Choose **Add new tag** for each tag that you'd like to associate with the custom template.
    b. Enter values in the **Key** and **Value** fields as necessary.

10. Choose **Save**.

Your custom template will now be available in the workflow templates list.

AWS CLI

You can use the CreateTemplate Migration Hub Orchestrator API operation to create a custom template using the AWS CLI.

Running custom templates

You can use the Migration Hub Orchestrator console or the AWS CLI to run custom templates.

> **Tip**
>
> Each template might have prerequisites and manual steps to run the workflow successfully. You can refer to the documentation for the predefined templates provided by Migration Hub Orchestrator your custom template is based on. For more information, see Templates.

Console

To run a custom template using the Migration Hub Orchestrator console

1. Sign in to the AWS Management Console and open the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/?region=us-east-1#/templates.
2. Choose the custom template from the list that you want to run.
3. Choose **Create workflow**.
4. Choose **Run**.
5. In the **Steps** tab, choose **Expand all**. Steps with a **Status** of **User attention required** need additional input to complete the step.
6. Choose the step which requires further input, choose **Actions**, **Change status**, **Completed**.
7. Take action on any steps which require your input for the workflow to proceed to completion.

AWS CLI

You can use the CreateWorkflow Migration Hub Orchestrator API operation to create a custom template using the AWS CLI.

Updating custom templates

You can't directly update a custom template. Instead, you can create a workflow from a custom template and make updates to the migration workflow it creates. Once you have made your updates, you can save your updates to a new custom template.

Console

To update a custom template and save them as a new template using the Migration Hub Orchestrator console

1. Sign in to the AWS Management Console and open the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/?region=us-east-1#/templates.
2. Choose the custom template from the list that you want to update.
3. Choose **Create workflow**.
4. Customize the workflow steps and step groups of the template as necessary. For more information about how to modify a workflow, see Migration workflows.
5. (Recommended) Once you have finished modifying the workflow, choose **Run** to start the workflow. For more information, see Running custom templates.
6. Choose **Save as a template**.
7. For **Name**, enter a name for the custom template.
8. (Optional) For **Description**, enter a description for the custom template.
9. (Optional) Add tags to your custom template:
    a. Choose **Add new tag** for each tag that you'd like to associate with the custom template.
    b. Enter values in the **Key** and **Value** fields as necessary.
10. Choose **Save**.

Your new custom template will be available in the workflow templates list.

AWS CLI

You can use the UpdateTemplate Migration Hub Orchestrator API operation to create a custom template using the AWS CLI.

Configure the Migration Hub Orchestrator plugin

The Migration Hub Orchestrator plugin is a virtual appliance that you can install in your on-premises VMware environment.

> **Important**
>
> 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.

Download

To deploy the plugin as a virtual machine (VM) in your VMware environment, download the plugin Open Virtualization Archive (OVA) file using the following steps.

1. Sign in to the https://console.aws.amazon.com/migrationhub/orchestrator/.
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.

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

Configure

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 username 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 username is Administrator, then the username 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.

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 NTLM 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 username 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 username 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.
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.

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.

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 NTLM 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 username 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.

Migration workflows

Migration workflows are created using templates. The predefined templates provided by Migration Hub Orchestrator 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 at least one step each. You can create a migration workflow with one of the predefined templates, or with a custom template that you can create.

Each template will have different prerequisites and configuration options. You should refer to the documentation for the template that you intend to use before creating a migration workflow. Once you create a migration workflow, you can perform various actions to customize it for your use case. For more information, see Templates.

Topics

Considerations and limitations

The following are considerations and limitations when working with migration workflows:

- You can make modifications to a migration workflow after it's created.
- You can only make modifications to step groups and steps that you have added to a migration workflow.
- 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.

- For the **Import virtual machine images to AWS** and **Rehost applications on Amazon EC2** predefined templates provided by Migration Hub Orchestrator, only steps added to the first step group are retained when you save a new custom template.

Creating step groups

You can add step groups using the Migration Hub Orchestrator console or the AWS CLI. Each step group must contain a step.

Console

To add step groups using the Migration Hub Orchestrator console

1. Sign in to the AWS Management Console and access the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/.
2. In the left navigation pane, under **Orchestrate**, choose **Workflows**.
3. On the **Workflows** page, select the workflow and choose **View details**.
4. Under **Steps**, select an existing step group.
5. Choose **Add**, **Add step group above** or **Add step group below**.
6. For **Step group name**, enter a name.
7. (Optional) for **Step Group description**, enter a description.
8. For **Name**, enter a name for the step.
9. (Optional) For **Description**, enter a description for the step.
10. (Optional) For **Script location**, choose **Amazon S3 URI** or **Upload a file**.
    a. If you chose **Amazon S3 URI**, provide the following details:
        i. For **Script run command**, enter a command to run the script.
        ii. For **Script run environment**, choose **On premises** or **AWS**.

A. If you chose **On premises**, for **Server**, choose a server from the dropdown menu. The resources listed are based on what you configured in AWS Application Discovery Service. For more information, refer to the documentation on the template your workflow was created from in the Templates section.

B. If you chose **AWS**, for **EC2 instance**, select the instances to run the script on.

b. If you chose **Upload a file**, provide the following details:

i. For **Script file**, choose **Choose file** and select a shell or PowerShell script file to use.

ii. For **Script run command**, enter a command to run the script.

iii. For Script run environment, choose On premises or AWS. The servers listed are the application servers that you configured in AWS Application Discovery Service. For more information, refer to the documentation on the template your workflow was created from in the Templates section.

A. If you chose **On premises**, for **Server**, choose a server from the dropdown menu. The resources listed are based on what you configured in AWS Application Discovery Service. For more information, refer to the documentation on the template your workflow was created from in the Templates section.

B. If you chose **AWS**, for **EC2 instance**, select the instances to run the script on.

> **Note**
>
> If you don't make a selection for **Script location**, the step's type will be set to **Manual** which requires user intervention. After completing the manual procedure for the step, you must update the status to complete for the migration workflow to continue.

AWS CLI

You can use the CreateWorkflowStepGroup API operation to add step groups to a workflow.

Creating steps in an existing step group

You can add steps using the Migration Hub Orchestrator console or the AWS CLI.

Console

To add steps using the Migration Hub Orchestrator console

1. Sign in to the AWS Management Console and access the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/.
2. In the left navigation pane, under **Orchestrate**, choose **Workflows**.
3. On the **Workflows** page, select the workflow that you want to customize and choose **View details**.
4. Select the step group in which you want to add steps.
5. Under **Steps**, select **Add**, **Add step to group**.
6. For **Name**, enter a name for the step.
7. (Optional) For **Description**, enter a description for the step.
8. (Optional) For **Script location**, choose **Amazon S3 URI** or **Upload a file**.
    a. If you chose **Amazon S3 URI**, provide the following details:
        i. For **Script run command**, enter a command to run the script.
        ii. For **Script run environment**, choose **On premises** or **AWS**.
            A. If you chose **On premises**, for **Server**, choose a server from the dropdown menu. The resources listed are based on what you configured in AWS Application Discovery Service. For more information, refer to the documentation on the template your workflow was created from in the Templates section.
            B. If you chose **AWS**, for **EC2 instance**, select the instances to run the script on.
    b. If you chose **Upload a file**, provide the following details:
        i. For **Script file**, choose **Choose file** and select a shell or PowerShell script file to use.
        ii. For **Script run command**, enter a command to run the script.

iii. For Script run environment, choose On premises or AWS. The servers listed are the application servers that you configured in AWS Application Discovery Service. For more information, refer to the documentation on the template your workflow was created from in the Templates section.

A. If you chose **On premises**, for **Server**, choose a server from the dropdown menu. The resources listed are based on what you configured in AWS Application Discovery Service. For more information, refer to the documentation on the template your workflow was created from in the Templates section.

B. If you chose **AWS**, for **EC2 instance**, select the instances to run the script on.

> **Note**
>
> If you don't make a selection for **Script location**, the step's type will be set to **Manual** which requires user intervention. After completing the manual procedure for the step, you must update the status to complete for the migration workflow to continue.

AWS CLI

You can use the CreateWorkflowStep API operation to add steps to a step group.

Updating step groups

You can update step groups using the Migration Hub Orchestrator console or the AWS CLI.

Console

To update step groups

1. Sign in to the AWS Management Console and access the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/.
2. In the left navigation pane, under **Orchestrate**, choose **Workflows**.
3. On the **Workflows** page, select the workflow that you want to customize and choose **View details**.
4. If you need to update the order of step groups in the workflow:

a. In the **Steps** pane, select an existing step group.

b. Choose **Move up** or **Move down** to reorder the step groups as needed.

5. If you need to update information about a step group:

a. In the **Steps** pane, select an existing step group, then choose **Actions**, **Edit step group**.

b. Update the **Step group name** and **Step group description** as required.

c. Choose **Save**.

AWS CLI

You can use the UpdateWorkflowStepGroup API operation to update the step groups in a workflow.

> **Important**
>
> You can only update step groups that you have added to a migration workflow that isn't running or paused.

Updating steps

You can update steps using the Migration Hub Orchestrator console or the AWS CLI.

Console

To update steps

1. Sign in to the AWS Management Console and access the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/.
2. In the left navigation pane, under **Orchestrate**, choose **Workflows**.
3. On the **Workflows** page, select the workflow that you want to customize and choose **View details**.
4. In the **Steps** pane, select the step group in which you need to update.
5. Choose **Expand all** to view the steps within the step group.
6. If you need to update the order of step groups in the workflow:

a. In the **Steps** pane, select the step you need to move within the step group.
b. Choose **Move up** or **Move down** to reorder the step as needed.

7. If you need to update information about a step group:

a. In the **Steps** pane, select an existing step group, then choose **Actions**, **Edit step**.
b. Update the **Step name** and **Step group description**, and **Script location** configuration as required.
c. Choose **Save**.

AWS CLI

You can use the UpdateWorkflowStep API operation to update the steps in a step group.

> **Important**
>
> You can only update steps that you have added to a migration workflow that isn't running or paused.

Deleting step groups

You can delete step groups using the Migration Hub Orchestrator console or the AWS CLI.

Console

You can only delete step groups when the following conditions are met:

- The workflow has a **Status** of **Not started** or **Paused**.
- The step group has a **Status** of **Awaiting dependencies**.
- The step group has a **Managed by** value of **Custom**.

To delete step groups

1. Sign in to the AWS Management Console and access the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/.
2. In the left navigation pane, under **Orchestrate**, choose **Workflows**.

3. On the **Workflows** page, select the workflow that you want to customize and choose **View details**.
4. In the **Steps** pane, select the step group to delete.
5. Choose **Actions**, **Delete step group**.
6. Enter **`delete`** and choose **Delete** to proceed.

AWS CLI

You can use the DeleteWorkflowStepGroup API operation to delete step groups in a workflow.

Deleting steps

You can delete steps using the Migration Hub Orchestrator console or the AWS CLI.

Console

You can only delete steps when the following conditions are met:

- The workflow has a **Status** of **Not started** or **Paused**.
- The step has a **Status** of **Awaiting dependencies**.
- The step has a **Managed by** value of **Custom**.

To delete steps

1. Sign in to the AWS Management Console and access the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/.
2. In the left navigation pane, under **Orchestrate**, choose **Workflows**.
3. On the **Workflows** page, select the workflow that you want to customize and choose **View details**.
4. In the **Steps** pane, select the step group that contains the step to delete.
5. Choose **Expand all** to view the steps within the step group.
6. Select the step to delete.
7. Choose **Actions**, **Delete step**.
8. Enter **`delete`** and choose **Delete** to proceed.

AWS CLI

You can use the DeleteWorkflowStep API operation to delete steps in a step group.

Running workflows

For guidance on running a migration workflow, refer to the documentation for the template that was used. For more information on the available templates, see Templates.

Pausing workflows

You can pause a running migration workflow using the Migration Hub Orchestrator console or the AWS CLI. You can only pause a running workflow.

Console

To pause a running workflow

1. Sign in to the AWS Management Console and access the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/.
2. In the left navigation pane, under **Orchestrate**, choose **Workflows**.
3. On the **Workflows** page, select the workflow that you want to pause.
4. Choose **Actions**, **Pause**.

AWS CLI

You can use the UpdateWorkflow API operation to pause a workflow.

Deleting workflows

You can delete migration workflows using the Migration Hub Orchestrator console or the AWS CLI. The workflow must have a **Status** of **Not started**, **Paused**, or **Failed**.

Console

To delete a migration workflow

1. Sign in to the AWS Management Console and access the Migration Hub Orchestrator console at https://console.aws.amazon.com/migrationhub/orchestrator/.
2. In the left navigation pane, under **Orchestrate**, choose **Workflows**.
3. On the **Workflows** page, select the workflow that you want to delete.
4. Choose **Actions**, **Delete**.
5. Enter **`delete`** and choose **Delete** to proceed.

AWS CLI

You can use the DeleteWorkflow API operation to delete a paused or failed workflow.

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. You are also responsible for 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 users with AWS IAM Identity Center or 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 require TLS 1.2 and recommend TLS 1.3.
- 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 sensitive 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 text 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 text 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.

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 AWS account full access to the Migration Hub Orchestrator service through the AWS Management Console.

Permissions details

This policy includes the following permissions.

- `account` – Grants permissions that allow listing AWS Regions.

- `discovery` – Grants permissions that allow access to Application Discovery Service.
- `ec2` – Grants permissions that allow describing EC2 instances and VPCs.
- `iam` – Allows a service-linked role to be created for the AWS account, which is a requirement for using Migration Hub Orchestrator. This policy also grants permissions that allow listing instance profiles and IAM roles.
- `kms` – Grants permissions that allow listing AWS KMS keys and aliases.
- `migrationhub-orchestrator` – Grants full access to Migration Hub Orchestrator.
- `s3` – Grants permissions that allow creating and reading from the S3 buckets used by Migration Hub Orchestrator.
- `secretsmanager` – Grants permissions that allow access to AWS Secrets Manager.

To view the permissions for this policy, see AWSMigrationHubOrchestratorConsoleFullAccess in the *AWS Managed Policy Reference Guide*.

AWS managed policy: AWSMigrationHubOrchestratorPlugin

Attach the `AWSMigrationHubOrchestratorPlugin` policy to your IAM identities.

The `AWSMigrationHubOrchestratorPlugin` policy grants an AWS 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` – Grants permissions that allow access to the Orchestrator plugin.
- `s3` – Grants permissions that allow write access to the S3 buckets used by Migration Hub Orchestrator.
- `secretsmanager` – Grants permissions that allow access to AWS Secrets Manager.

To view the permissions for this policy, see AWSMigrationHubOrchestratorPlugin in the *AWS Managed Policy Reference Guide*.

AWS managed policy: AWSMigrationHubOrchestratorInstanceRolePolicy

Attach the `AWSMigrationHubOrchestratorInstanceRolePolicy` policy to your IAM identities.

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

Permissions details

This policy includes the following permissions.

- `migrationhub-orchestrator` – Grants permissions that allow access to Migration Hub Orchestrator.
- `s3` – Grants permissions that allow read/write access to Amazon S3 buckets used by Migration Hub Orchestrator.
- `secretsmanager` – Grants permissions that allow access to AWS Secrets Manager.

To view the permissions for this policy, see AWSMigrationHubOrchestratorInstanceRolePolicy in the *AWS Managed Policy Reference Guide*.

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
AWSMigrationHubOrchestratorServiceRolePolicy – Updated policy	`launchwizard:ListDeployments` and `launchwizard:GetDeployment` actions added to the policy.	March 4, 2024

Change	Description	Date
AWSMigrationHubOrchestrator ConsoleFullAccess – Updated policy	`ec2:DescribeVpcs` , `kms:ListKeys` , `kms:ListAliases` , `iam:ListInstancePr ofiles` , `iam:ListRoles` , `ecs:ListClusters` , and `account:ListRegions` actions added to the policy.	December 5, 2023
AWSMigrationHubOrchestratorServiceRo lePolicy – Updated policy	`ec2:DescribeLaunch Templates` action added to the policy.	February 24, 2023
AWSMigrationHubOrchestratorServiceRo lePolicy – Updated policy	`ec2:DescribeImport ImageTasks` , `s3:ListBu cket` , and `events:Re moveTargets` actions added to the policy.	December 21, 2022
AWSMigrationHubOrchestrator ConsoleFullAccess – New policy made available at launch	`AWSMigrationHubOrc hestratorConsoleFu llAccess` grants an AWS account full access to the Migration Hub Orchestra tor service through the AWS Management Console.	April 20, 2022

Change	Description	Date
AWSMigrationHubOrchestratorPlugin – New policy made available at launch	AWSMigrationHubOrchestratorPlugin grants an AWS account access to the Migration Hub Orchestrator service and read/write access to Amazon 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 – 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 AWS account read/write access to Amazon S3 buckets that are related to the service and to AWS Secrets Manager to fetch credentials.	April 20, 2022

Change	Description	Date
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:DescribeProvisionedApp
```

```
launchwizard:GetDeployment
```

```
launchwizard:ListDeployments
```

```
launchwizard:ListProvisionedApps
```

Amazon Elastic Compute Cloud actions

```
ec2:DescribeInstances
```

```
ec2:CreateLaunchTemplateVersion
```

```
ec2:ModifyLaunchTemplate
```

```
ec2:DescribeImportImageTasks
```

```
ec2:DescribeLaunchTemplates
```

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
```

```
s3:ListBucket
```

Amazon EventBridge actions

```
events:PutTargets

events:DescribeRule

events:DeleteRule

events:PutRule

events:RemoveTargets
```

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
```

To view the permissions for this policy, see AWSMigrationHubOrchestratorServiceRolePolicy in the *AWS Managed Policy Reference Guide*.

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

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.
- 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) 1.2 or later. We recommend TLS 1.3. 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.

Version history of AWS Migration Hub Orchestrator plugin

The following table provides a version history of the AWS Migration Hub Orchestrator plugin.

Version	Details	Release date
1.0.3	Bug fix: reduced redundant file creation	April 18, 2023
1.0.1	Bug fix: improved mechanisms for plugin tasks	March 03, 2023
1.0	Initial release	April 20, 2022

Document history

Change	Description	Date
Updated policy	Updated the `AWSMigrationHubStrategyServiceRolePolicy` policy.	March 4, 2024
New feature	Added the ability to create Custom templates.	February 29, 2024
New feature	Added the Replatform applications to Amazon ECS template.	December 6, 2023
Updated policy	Updated the `AWSMigrationHubOrchestratorConsoleFullAccess` policy.	December 5, 2023
New section	Added the How it works section.	May 22, 2023
Updated section	Updated the Configure plugin section.	May 22, 2023
Updated feature	Updated the Migrate SAP template.	April 04, 2023
Updated policy	Updated the `AWSMigrationHubOrchestratorServiceRolePolicy`.	February 24, 2023
New feature	Added the Import virtual machine images to AWS template.	December 21, 2022

Updated policy	Updated the `AWSMigrationHubOrchestratorServiceRolePolicy`.	December 21, 2022
New feature	Added the Replatform SQL server on Amazon RDS template.	November 01, 2022
New feature	Added the Rehost SQL server on Amazon EC2 template.	November 01, 2022
Initial release	Initial release of the Migration Hub Orchestrator User Guide.	April 20, 2022