User Guide

AWS Migration Acceleration Program 2.0 Tagging Guide

AWS Migration Acceleration Program 2.0 Tagging Guide: User Guide

Table of Contents

What is AWS Migration Acceleration Program 2.0?

The Migration Acceleration Program (MAP 2.0) is a comprehensive and proven cloud migration program that's the result of AWS' experience migrating hundreds of enterprise customers. MAP consists of an agile-based migration methodology, a global network of vetted partners, automation tools, a training path to up-skill staff, professional services, and financial investment to help manage migration double bubble costs.

Getting started with AWS Migration Acceleration Program 2.0

As you move your existing on-premises workloads to AWS, the migrated workloads are identified through a tagging mechanism. Tagging is required because it is used to report the migrated workloads' spend and generate appropriate incentives. MAP cannot provide these incentives if you do not activate Cost Allocation Tags or tag the migrated resources.

The MAP migration tracking mechanism uses native AWS functionality that you configure in the following steps:

1. Tagging migrated workloads.
2. Selecting the appropriate MAP 2.0 tag value for your `map-migrated` key. To learn how to select the right tag value, see *Tagging key combinations*.

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost allocation tag. Tags that are automatically activated don't count towards your cost allocation tag quota. For more information, see Quotas and restrictions.
> Ensure that Cost Explorer is enabled. To learn how to enable Cost Explorer, see Enabling Cost Explorer.

Topics

Prerequisites

Before you begin tagging your workloads, you must have the following:

- Your AWS Migration Program Engagement number (MPE ID number), also known as your project number, in your Migration Plan.
- Access to the AWS management (payer) account(s) listed in your Migration Plan.
- Ensure that Cost Explorer is enabled. To learn how to enable Cost Explorer, see Enabling Cost Explorer.
- IAM permissions to tag migrated workloads.

MPE ID length

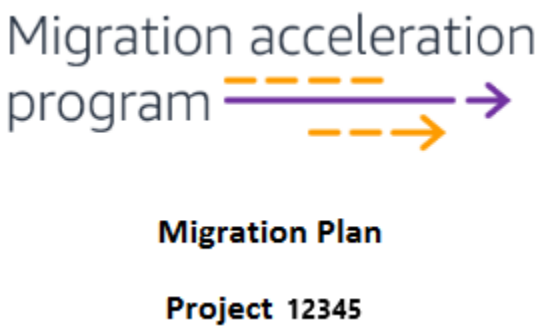
Your project number found within your migration plan is also known as your MPE ID, and consists of 5-digits (short MPE ID) or 10 alphanumeric characters (long MPE ID).

For example:

- **Project 12345**: MPE ID = 12345 (short MPE ID)
- **Project ABCDE12345**: MPE ID = ABCDE12345 (long MPE ID)

> **Note**
>
> The alphanumeric MPE ID uses uppercase letters.

Refer to the first page below the title of your migration plan to find your project number as shown in the following image:



Tagging Resources

You can begin tagging your migrated resources in order for them to be included in your MAP 2.0 incentive calculations. Tagging your migrated workloads with the `map-migrated` tag gives you the following benefits:

- You can track the migration inventory scope as it is migrated over time from your existing environment to AWS.
- You can identify specific AWS resources used in place of existing pre-migration resources.
- You can collect the cost and usage data of the migrated resources for you to report TCO and other financial data.

These tags are applied by the workload owners who are migrating their workloads and this process is repeated as workloads are moved across until the entire MAP migration scope has been migrated. While we encourage you to automate as much as possible, you have the ability to tag your migrated workloads using automated and manual tagging methods.

You can add a tag line to your templates if you are using an Infrastructure-As-Code tool such as AWS CloudFormation or Terraform to create your migrated resource on AWS. However, you can tag resources individually or bulk tag them using the AWS Tag Editor if you are creating resources directly from the AWS Management Console. For more information about tagging values, see *Tagging key combinations*.

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost allocation tag. Tags that are automatically activated don't count towards your cost allocation tag quota. For more information, see Quotas and restrictions.

Topics

MAP tag

The Migration Acceleration Program requires that you tag resources with the following tag key. Copy the following key and use it exactly as it is without any alterations:

```
map-migrated
```

> **Important**
>
> The tag key is case sensitive. You must use it exactly as it appears here. You cannot introduce spaces, change the case of any of the letters, or alter the key in any way.

Automated tagging

We encourage you to automate tagging your resources as much as possible. You can automate your tagging by adding a tag line to your AWS CloudFormation template, AWS Cloud Development Kit (AWS CDK), AWS Transform MGN (MGN), Cloud Migration Factory Solution (CMF), and AWS Serverless Application Model (SAM). You can also use the AWS Tag Editor to automate your tagging.

Topics

- AWS Transform MGN (MGN)
- CloudFormation
- AWS Cloud Development Kit (AWS CDK)
- Cloud Migration Factory Solution (CMF)
- AWS Serverless Application Model (SAM)
- AWS Tag Editor

AWS Transform MGN (MGN)

AWS Transform MGN (MGN) is a highly automated re-hosting solution that simplifies, expedites, and reduces the cost of migrating applications to AWS. It enables companies to re-host a large number of physical, virtual, or cloud servers without compatibility issues, performance disruption, or long cutover windows. AWS MGN continuously replicates source servers to your AWS account. When you're ready, it automatically converts and launches your servers on AWS so you can quickly

benefit from the cost savings, productivity, resilience, and agility of the Cloud. In addition, AWS MGN allows you to modernize launched applications by running custom or preconfigured actions.

MGN is MAP 2.0 aware and can automatically apply the required MAP 2.0 tags to your workloads migrated by MGN.

To get started

1. Set up MGN in any AWS account associated in the same AWS organization as the payer account listed in your MAP 2.0 term. For more information about setting up MGN, see Getting started with AWS Application Migration Service in the *Application Migration Service guide*.
2. During the MGN setup, choose **Settings**.
3. Choose the **Launch** template tab.
4. Choose **Edit**.
5. Choose **Add MAP tag to launched servers**.
6. Enter and replace your **MPE ID** with the tag value you want to apply to the migrated workloads.

   Example:

   - If your MPE ID is *12345*, use the value *mig12345*.
   - If your MPE ID is *ABCDE12345*, use the value *migABCDE12345*.
7. Choose **Save template**.
8. Proceed migrating your workloads with MGN as detailed in Source Servers of the Application Migration Service guide. For more information about tagging in MGN, see Tags in the *Application Migration Service guide*.

   The MAP 2.0 tags are automatically applied to all workloads migrated using MGN. Repeat Steps 2 - 7 if you are using MGN to migrate to multiple accounts or regions.

Depending on your migrated resource and MPE ID, the tag value can be any of the following:

Short MPE IDs

- `mig`*5-digit MPE ID*
- `sap`*5-digit MPE ID*

- `oracle`*`5-digit MPE ID`*

Long MPE IDs

- `mig`*`10 alphanumeric MPE ID characters`*
- `sap`*`10 alphanumeric MPE ID characters`*
- `oracle`*`10 alphanumeric MPE ID characters`*

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about what tag values you should use, see *Tagging key combinations*. For more information about your MPE ID, see the section called "MPE ID length".

CloudFormation

AWS CloudFormation

AWS CloudFormation (CloudFormation) enables you to create and provision AWS infrastructure deployments predictably and repeatedly. It helps you leverage AWS products such as Amazon EC2, Amazon Elastic Block Store (EBS), Amazon SNS, Elastic Load Balancing, and Application Auto Scaling to build highly reliable, highly scalable, cost-effective applications in the cloud without worrying about creating and configuring the underlying AWS infrastructure. CloudFormation enables you to use a template file to create and delete a collection of resources together as a single unit (a stack).

You can use the Resource Tags property to apply tags to resources, which can help you identify and categorize those resources. You can only tag CloudFormation supported resources. For information about which resources you can tag with CloudFormation, see the AWS resources and property types reference in the *CloudFormation user guide*. For more information on how to use CloudFormation, see the Resource tag in the *CloudFormation user guide*.

Replace `mig12345` with the tag value needed for your migrated resource followed by your MAP term agreement number in following example:

```
{
    "Key" : "map-migrated",
    "Value" : "mig12345"
}
```

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost allocation tag. Tags that are automatically activated don't count towards your cost allocation tag quota. For more information, see Quotas and restrictions.

Depending on your migrated resource and MPE ID, the tag value can be any of the following:

Short MPE IDs

- `mig`*5-digit MPE ID*
- `sap`*5-digit MPE ID*
- `oracle`*5-digit MPE ID*

Long MPE IDs

- `mig`*10 alphanumeric MPE ID characters*
- `sap`*10 alphanumeric MPE ID characters*
- `oracle`*10 alphanumeric MPE ID characters*

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about what tag values you should use, see *Tagging key combinations*. For more information about your MPE ID, see the section called "MPE ID length".

AWS Cloud Development Kit (AWS CDK)

The AWS Cloud Development Kit (AWS CDK) (AWS CDK) is an open-source software development framework to define your cloud application resources using familiar programming languages. A tag in AWS CDK is applied to a given construct that also applies to all of its taggable children. These tags are included in the AWS CloudFormation template synthesized from your application and are applied to the AWS resources it deploys. For more information about AWS CDK tagging, see Tagging in the *AWS Cloud Development Kit (AWS CDK) v2 developer guide*.

Replace `mig12345` with the tag value needed for your migrated resource followed by your MAP term agreement number in following example:

```
import { Tags } from 'aws-cdk-lib';

Tags.of(myConstruct).add('map-migrated', 'mig12345');
```

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost allocation tag. Tags that are automatically activated don't count towards your cost allocation tag quota. For more information, see Quotas and restrictions.

Depending on your migrated resource and MPE ID, the tag value can be any of the following:

Short MPE IDs

- `mig`*5-digit MPE ID*
- `sap`*5-digit MPE ID*
- `oracle`*5-digit MPE ID*

Long MPE IDs

- `mig`*10 alphanumeric MPE ID characters*
- `sap`*10 alphanumeric MPE ID characters*
- `oracle`*10 alphanumeric MPE ID characters*

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about what tag values you should use, see *Tagging key combinations*. For more information about your MPE ID, see the section called "MPE ID length".

Cloud Migration Factory Solution (CMF)

Cloud Migration Factory Solution coordinates and automates large scale migrations to the AWS Cloud, involving numerous servers. Enterprises can improve performance and prevent long cutover windows by automating manual processes and integrating multiple tools efficiently. This is possible through this solution's orchestration platform options, which includes MGN(MGN). We recommend using AWS MGN to rehost servers to AWS at scale. Today, this solution is used by AWS Professional Services, AWS Partners, and other enterprises. For more information about CMF, see the Coordinate and automate large scale migrations to the AWS Cloud using the Cloud Migration Factory on AWS solution in the *Cloud Migration Factory on AWS implementation guide*.

Use the steps outlined below to add tags to all imported servers in CMF.

To get started

1. Use the following script to add tags to all imported servers in CMF:

```
# Version: 09APR2021.01

from __future__ import print_function
import sys
import argparse
import requests
import json
import csv
import boto3
import botocore.exceptions
import mfcommon

serverendpoint = mfcommon.serverendpoint
```

```
with open('FactoryEndpoints.json') as json_file:
    endpoints = json.load(json_file)

def get_reader(file):
    ordered_dict_list = []
    input_file = csv.DictReader(open(file))
    for row in input_file:
        ordered_dict_list.append(row)
    # return input_file
    return ordered_dict_list

def data_validation(data, servers):
    # Validate if Name column exist
    keys = data[0].keys()
    if "Name" not in keys:
        print ("ERROR: 'Name' column is mandatory")
        sys.exit(3)
    # check if none value exist
    for row in data:
        for key in keys:
            if key not in row:
               print("ERROR: "+ key + " tag value is missing for server " +
 row['Name'])
               sys.exit(4)
            if row[key] == None:
               print("ERROR: "+ key + " tag value is missing for server " +
 row['Name'])
               sys.exit(6)
            if row[key] == row[key].strip() == "":
               print("ERROR: "+ key + " tag for server " + row['Name'] + " is
 empty")
               sys.exit(7)
    # Validate duplicate server names in csv file
    server_list = []
    for row in data:
        if row['Name'].strip().lower() not in server_list:
            server_list.append(str(row['Name']).strip().lower())
        else:
            print("ERROR: Duplicated Server Name: " + row['Name'])
            sys.exit(2)
    # Check if server exist in the migration factory
    for server in server_list:
        match = False
        for s in servers:
```

```
            if (server.lower() == s['server_name'].lower()):
                match = True
        if (match == False):
            print("ERROR: Server " + server + " doesn't exist in the migration
 factory")
            sys.exit(1)


def uploading_data(data, token, UserHOST):
    keys = data[0].keys()
    auth = {"Authorization": token}
    servers = json.loads(requests.get(UserHOST + serverendpoint,
 headers=auth).text)
    data_validation(data, servers)
    for row in data:
        update_server_tags = {}
        tags = []
        server_id = ""
        for server in servers:
            if row['Name'].strip().lower() ==
 server['server_name'].strip().lower():
                server_id = server["server_id"]
                for key in keys:
                    tag = {}
                    tag['key'] = key
                    tag['value'] = row[key].strip()
                    tags.append(tag)
        update_server_tags['tags'] = tags
        r = requests.put(UserHOST + serverendpoint + '/' + server_id, headers=auth,
 data=json.dumps(update_server_tags))
        if r.status_code == 200:
           print("Tags for server " + row['Name'] + " updated in the migration
 factory")
        else:
           print("ERROR: updating tags for server " + row['Name'] + " failed : " +
 r.text + ".......")

def main(arguments):
    parser = argparse.ArgumentParser(
        description=__doc__,
        formatter_class=argparse.RawDescriptionHelpFormatter)
    parser.add_argument('--Intakeform', required=True)
    args = parser.parse_args(arguments)
```

```
    UserHOST = ""

    # Get MF endpoints from FactoryEndpoints.json file
    if 'UserApiUrl' in endpoints:
        UserHOST = endpoints['UserApiUrl']
    else:
        print("ERROR: Invalid FactoryEndpoints.json file, please update
 UserApiUrl")
        sys.exit()

    print("****************************")
    print("*Login to Migration factory*")
    print("****************************")
    token = mfcommon.Factorylogin()

    print("****************************")
    print("*  Reading Tags form List  *")
    print("****************************")
    data = get_reader(args.Intakeform)
    print("Tags loaded for processing....")
    print("")

    print("*********************************************")
    print("*   Updating tags in the migration factory  *")
    print("*********************************************")

    r = uploading_data(data,token,UserHOST)

if __name__ == '__main__':
    sys.exit(main(sys.argv[1:]))
```

2. Go to the migrated resources such as Amazon RDS.
3. Choose **Add tags**.
4. Enter `map-migrated` as the Tag key.
5. Enter and replace your **MPE ID** with the tag value you want to apply to the migrated workloads.

   Example:

   - If your MPE ID is *12345*, use the value *mig12345*.
   - If your MPE ID is *ABCDE12345*, use the value *migABCDE12345*.

6. Choose **Add tags**.

Depending on your migrated resource and MPE ID, the tag value can be any of the following:

Short MPE IDs

- `mig`*5-digit MPE ID*
- `sap`*5-digit MPE ID*
- `oracle`*5-digit MPE ID*

Long MPE IDs

- `mig`*10 alphanumeric MPE ID characters*
- `sap`*10 alphanumeric MPE ID characters*
- `oracle`*10 alphanumeric MPE ID characters*

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about what tag values you should use, see *Tagging key combinations*. For more information about your MPE ID, see the section called "MPE ID length".

Repeat the steps above for any other associated resources such as Snapshots. For more information about what tags you should use, see *Tagging key combinations*.

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost allocation tag. Tags that are automatically activated don't count towards your cost allocation tag quota. For more information, see Quotas and restrictions.

AWS Serverless Application Model (SAM)

The AWS Serverless Application Model (SAM) is an open-source framework for building serverless applications. It provides shorthand syntax to express functions, APIs, databases, and event source mappings. You can define the application you want and model it using YAML by using a few lines per resource. During deployment, SAM transforms and expands the SAM syntax into AWS CloudFormation syntax, enabling you to build serverless applications faster. For more information about SAM, see the Globals section of the AWS SAM template in the *AWS Serverless Application Model developer guide*.

Replace `mig12345` with the tag value needed for your migrated resource followed by your MAP term agreement number in the following example to tag resources at the Global and Resource level:

```
Globals:
  Function:
    Tags:
       map-migrated: "mig12345"

  HttpApi:
    Tags:
       map-migrated: "mig12345"
```

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost allocation tag. Tags that are automatically activated don't count towards your cost allocation tag quota. For more information, see Quotas and restrictions.

Depending on your migrated resource and MPE ID, the tag value can be any of the following:

Short MPE IDs

- `mig`*5-digit MPE ID*
- `sap`*5-digit MPE ID*
- `oracle`*5-digit MPE ID*

Long MPE IDs

- `mig`*10 alphanumeric MPE ID characters*
- `sap`*10 alphanumeric MPE ID characters*
- `oracle`*10 alphanumeric MPE ID characters*

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about what tag values you should use, see *Tagging key combinations*. For more information about your MPE ID, see the section called "MPE ID length".

AWS Tag Editor

You can use the AWS Resource Groups and Tag Editor to tag your migrated resources.

> **Note**
>
> The AWS Tag Editor only works for resources running in the account.

To get started

1. Open the AWS Management Console.
2. Go to the **Resource Groups & Tag Editor, Tagging, Tag Editor** page.
3. Specify the Region(s) your resources are located. **Example**: us-east-1.
4. Choose the type of resources you want to bulk tag. **Example**: EC2, Lambda, and S3.
5. Choose **Search resources** to view the resources that meet the conditions you have selected.
6. Select all or a few of the listed resources you want to tag.
7. Choose **Manage tags** of selected resources.
8. Enter `map-migrated` in the Tag Key field.

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost allocation tag. Tags that are automatically activated don't count towards your cost allocation tag quota. For more information, see Quotas and restrictions.

9. Enter and replace your **MPE ID** with the tag value you want to apply to the migrated workloads.

   Example:

   - If your MPE ID is *12345*, use the value *mig12345*.
   - If your MPE ID is *ABCDE12345*, use the value *migABCDE12345*.

10. Choose **Review**.
11. Choose **Apply tag** changes.

Depending on your migrated resource and MPE ID, the tag value can be any of the following:

Short MPE IDs

- `mig`*5-digit MPE ID*
- `sap`*5-digit MPE ID*
- `oracle`*5-digit MPE ID*

Long MPE IDs

- `mig`*10 alphanumeric MPE ID characters*
- `sap`*10 alphanumeric MPE ID characters*
- `oracle`*10 alphanumeric MPE ID characters*

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about what tag values you

should use, see *Tagging key combinations*. For more information about your MPE ID, see the section called "MPE ID length".

Manual tagging

You can manually tag your migrated resources using the AWS Management Console.

To get started

1. Go to your AWS Management Console.
2. Go to the migrated resources. **Example**: Amazon RDS.
3. Choose **Add tags**.
4. Enter `map-migrated` as the Tag key.

   > **Note**
   >
   > The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost allocation tag. Tags that are automatically activated don't count towards your cost allocation tag quota. For more information, see Quotas and restrictions.

5. Enter and replace your **MPE ID** with the tag value you want to apply to the migrated workloads.

   Example:

   - If your MPE ID is *12345*, use the value *mig12345*.
   - If your MPE ID is *ABCDE12345*, use the value *migABCDE12345*.
6. Choose **Save**.

Depending on your migrated resource and MPE ID, the tag value can be any of the following:

Short MPE IDs

- `mig`*5-digit MPE ID*
- `sap`*5-digit MPE ID*

- `oracle`*`5-digit MPE ID`*

Long MPE IDs

- `mig`*`10 alphanumeric characters MPE ID`*
- `sap`*`10 alphanumeric characters MPE ID`*
- `oracle`*`10 alphanumeric characters MPE ID`*

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about what tag values you should use, see *Tagging key combinations*. For more information about your MPE ID, see the section called "MPE ID length".

Repeat the steps above for all associated resources such as Snapshots. For more information about tagging resources, see the Tag your Amazon EC2 resources in the *Amazon Elastic Compute Cloud user guide for Linux instances*.

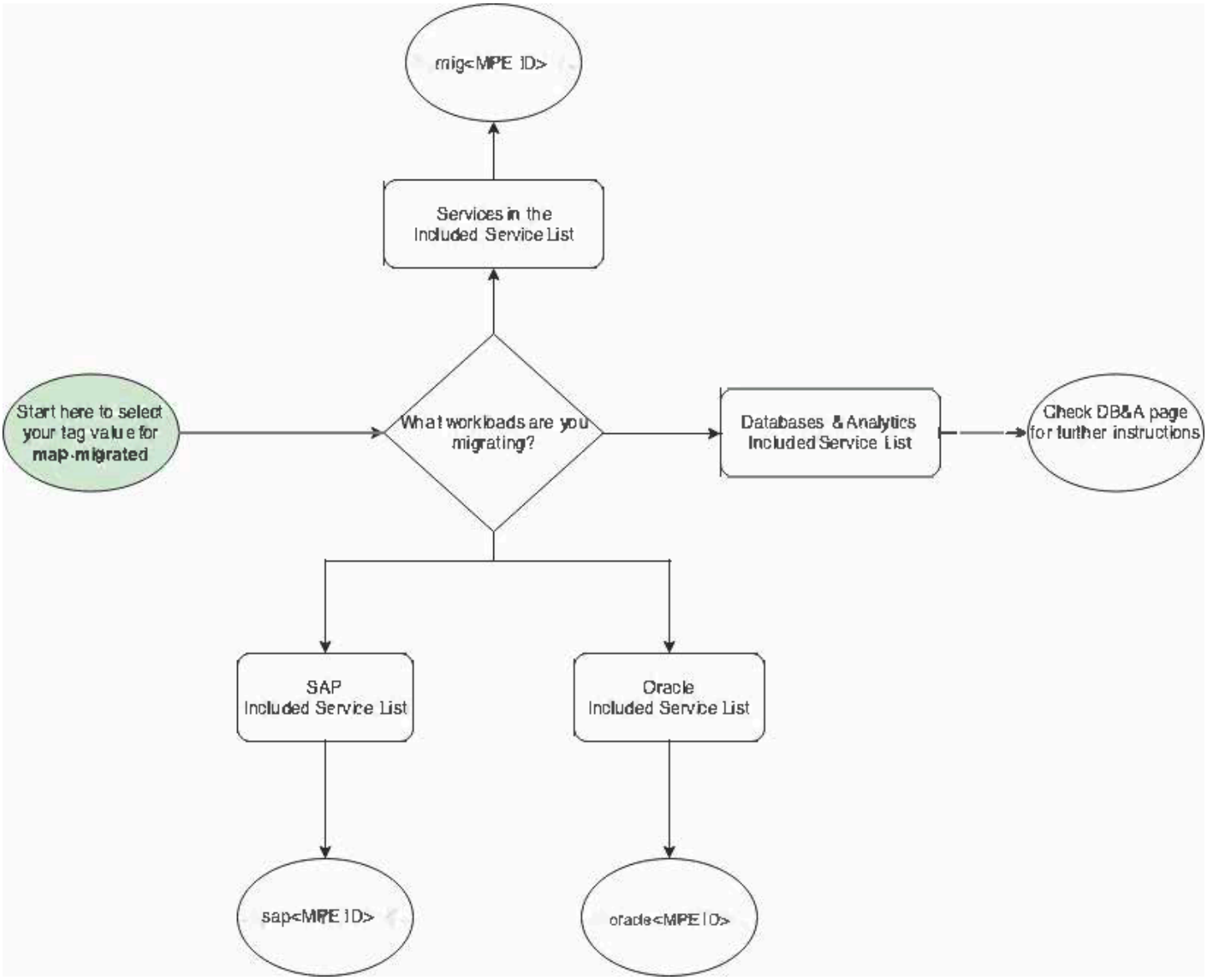
Tagging key combinations

To better help you find your tagging values, consider the following questions:

- What AWS service will be the destination for your migration?
- What is in the license agreement of your databases?
- Where is the source of your migration (on-prem or already in AWS as Amazon EC2A)?
- Are your workloads in the specialized workload list such as SAP or Oracle workloads?

Use the tagging decision tree shown in the following diagram to help you assign the tag value for your migrated workloads. For a complete list of services that are included in MAP 2.0, see the MAP 2.0 Included Services list: **https://s3-us-west-2.amazonaws.com/map-2.0-customer-documentation/included-services/MAP_Included_Services_List.pdf**.

The following diagram shows the decision tree for selecting the appropriate tag value:



To get started

- Where are you migrating to?

  - **Services included in the Service list**
  - **SAP included in the Service list**
  - **Oracle included in the Service list**
  - **DB&A included Service list**: For more information, see the section called “Database and analytic tags”.

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about your MPE ID, see the section called "MPE ID length".

Services - Short MPE IDs

`mig`*`5-digit MPE ID`*

Example: `mig12345`

Services - Long MPE IDs

`mig`*`10 alphanumeric MPE ID characters`*

Example: `migABCDE12345`

SAP - Short MPE IDs

`sap`*`5-digit MPE ID`*

Example: `sap12345`

SAP - Long MPE IDs

`sap`*`10 alphanumeric MPE ID characters`*

Example: `sapABCDE12345`

Oracle - Short MPE IDs

`oracle`*`5-digit MPE ID`*

Example: `oracle12345`

Oracle - Long MPE IDs

`oracle`*`10 alphanumeric MPE ID characters`*

Example: `oracleABCDE12345`

Amazon Bedrock MAP Tagging using the AWS CLI

This guide explains how to tag Amazon Bedrock and Amazon Bedrock AgentCore workloads to report MAP spend and generate any appropriate incentives using the AWS CLI.

There are two methods to tag your Bedrock workloads for MAP:

- **IAM principal tagging (recommended)** — Tag the IAM role used to invoke Bedrock APIs with `map-migrated`. This is the simplest approach and requires no changes to your application code or additional Bedrock resources. This method is only available effective June 8, 2026.
- **Resource tagging (application inference profiles)** — Create an application inference profile, tag it with `map-migrated`, and invoke models through the profile. This method provides per-model and per-region cost granularity but requires creating and managing additional Bedrock resources.

> **Important**
>
> If both a resource tag and an IAM principal tag are present, the resource tag takes precedence for MAP spend. Therefore, select only one method to tag Amazon Bedrock and Amazon Bedrock AgentCore workloads.

IAM principal tagging (recommended)

Tag the IAM role used for Amazon Bedrock or Amazon Bedrock AgentCore API calls with the `map-migrated` tag. This approach uses IAM principal cost allocation tags, which allows MAP spend tracking without requiring you to create and manage application inference profiles or change your application code.

Prerequisites

1. Must have AWS Migration Program Engagement number (MPE ID number), also known as your project number, in your Migration Plan.
2. Enable IAM principal cost allocation tags in your AWS Billing and Cost Management console. For more information, see IAM principal cost allocation tags in the *AWS Billing User Guide*.
3. Activate the `map-migrated` tag as a cost allocation tag (CAT) in the Billing console under **Cost allocation tags**. Cost allocation tags are available for both resource tags and IAM principal tags.
4. Your role must have permission to tag IAM roles. If your role has the IAMFullAccess AWS-managed policy attached, you can skip this step. Otherwise, ensure your role has the `iam:TagRole` and `iam:ListRoleTags` permissions.

Tagging your IAM role

Tag the IAM role that is used to invoke Amazon Bedrock or AgentCore APIs with the `map-migrated` tag and your MPE ID.

1. Identify the IAM role used for your Bedrock API calls. This is the role assumed by your application or service when invoking Bedrock models.
2. Use the AWS CLI to apply the `map-migrated` tag to the role:

   ```
   $ aws iam tag-role --role-name MyBedrockRole --tags "Key=map-
   migrated,Value=migYOUR_MPE_ID"
   ```

   Replace *MyBedrockRole* with the name of your IAM role and *YOUR_MPE_ID* with your MPE ID.
3. Verify the tag was applied correctly:

   ```
   $ aws iam list-role-tags --role-name MyBedrockRole
   ```

   The output should show the `map-migrated` tag:

   ```
   {
     "Tags": [
       {
         "Key": "map-migrated",
   ```

```
      "Value": "mig123456789"
    }
  ]
}
```

Once the IAM role is tagged, all Amazon Bedrock and AgentCore API calls made using that role will be tracked for MAP spend. The `iamPrincipal/map-migrated` tag will appear in your Cost and Usage Report (CUR) data for eligible billing lines.

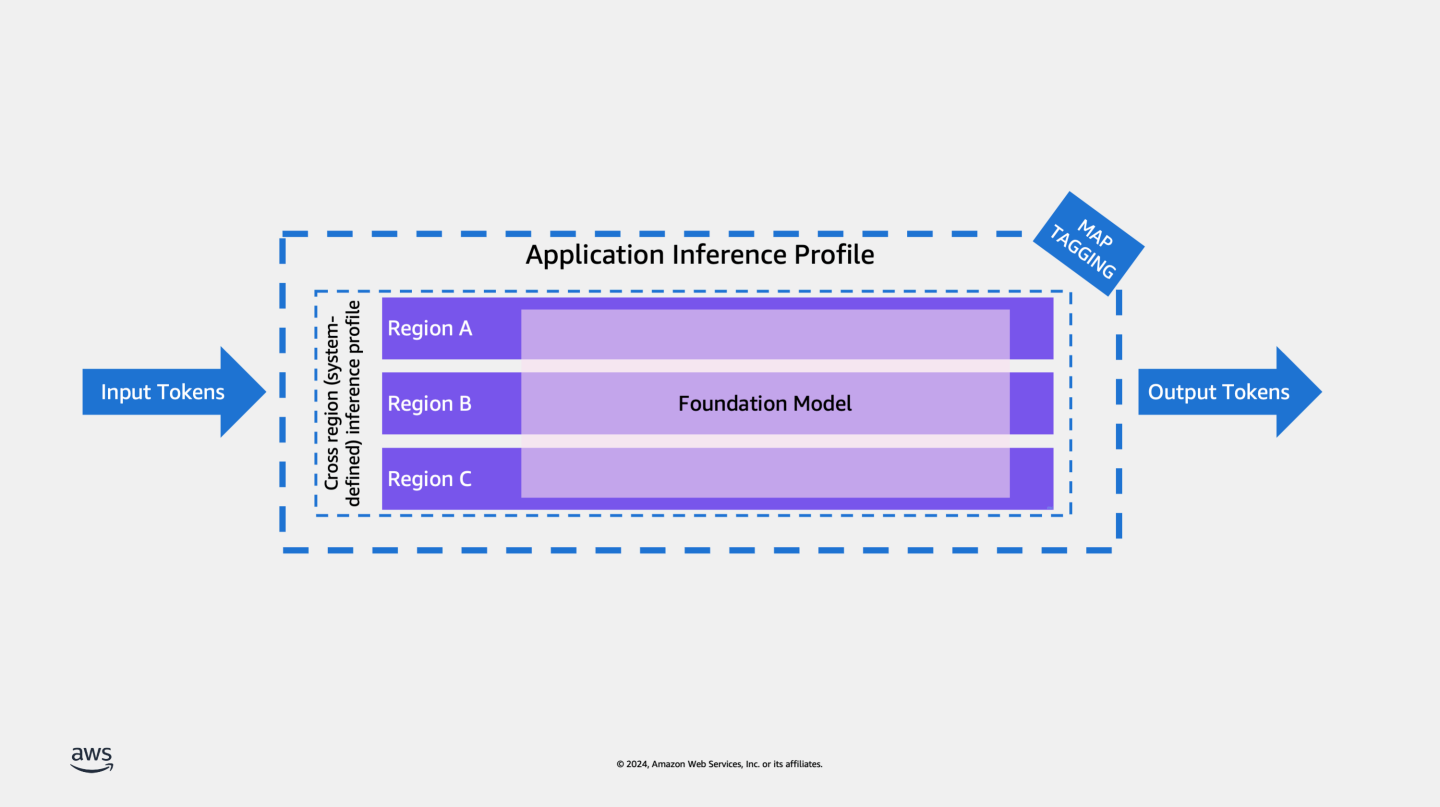
Important considerations

- **Supported services**: IAM principal tagging for MAP is only recognized for Amazon Bedrock and Amazon Bedrock AgentCore billing lines. Other AWS services are not eligible for MAP credit via IAM principal tags.
- **Tag precedence**: If both a resource-level `map-migrated` tag (for example, on an application inference profile) and an IAM principal `map-migrated` tag are present, the resource tag takes precedence for MAP spend reporting.
- **Resource exclusion**: If the IAM role was in use before start of your MAP migration, the associated spend will be excluded from MAP spend.
- **Tag value format**: Use the same tag value format as resource tags: `mig` followed by your MPE ID (for example, `mig12345` or `migABCDE12345`).

Resource tagging (application inference profiles)

If you need per-model or per-region cost granularity, you can create application inference profiles and tag them with `map-migrated`. This approach requires creating Bedrock resources and updating your application to invoke models through the tagged profile.

Introducing inference profiles



Inference profiles are a resource of Amazon Bedrock that enable model invocation and cost management.

An inference profile serves as a configuration container that specifies both a foundation model and its associated AWS regions for model invocation. Using inference profiles organizations can effectively manage their model invocations across multiple regions, helping to distribute workload and prevent performance bottlenecks. They play a crucial role in cost management and resource tracking, as they can be tagged with cost allocation tags to monitor usage and expenses across different regions.

Amazon Bedrock offers the following types of inference profiles:

Cross-region (system-defined) inference profiles: Inference profiles that are predefined in Amazon Bedrock and include multiple Regions to which requests for a model can be routed.

Application inference profiles: Inference profiles that a user creates to track costs and model usage. You can create an inference profile that routes model invocation requests to one Region or to multiple Regions.

Currently you can only create an inference profile using the Amazon Bedrock API.

A key feature of inference profiles is their integration with cost allocation tags - organizations can apply cost allocation tags to application inference profiles to track and manage expenses. When invoking models through Bedrock, users must specify a profile which contains the region and model specifications. This profile-based approach represents an evolution from the previous direct model ARN invocation method, providing better resource management and cost tracking capabilities. Specifically, application inference profiles support MAP tagging functionality, enabling organizations to track migration progress and resource utilization within the MAP program framework.

Amazon Bedrock offers significant cost advantages for customers with predictable foundation model inference patterns. When used appropriately, provisioned throughput can deliver substantial cost savings compared to on-demand pricing, making it an economical choice for consistent workload volumes. This pricing model is particularly beneficial for customers who can forecast their model inference needs and maintain steady utilization levels. For example, enterprises running regular batch processing jobs, customer service applications with predictable chat volumes, or content generation workflows with consistent throughput requirements can optimize their costs by choosing provisioned capacity. While on-demand mode provides flexibility for variable workloads, provisioned capacity in Bedrock enables customers to optimize their AI/ML spending by committing to a specific throughput level. This approach not only helps reduce operational costs but also ensures reliable performance for applications that require consistent model access, though customers need to carefully plan their capacity requirements to maximize the cost benefits.

Prerequisites

To implement MAP tagging with Bedrock inference profiles, you'll need to follow a structured process that involves creating and tagging application inference profiles. Here's how it works:

1. Must have AWS Migration Program Engagement number (MPE ID number), also known as your project number, in your Migration Plan.
2. Customers must gain access to supported foundation models only by using the Amazon Bedrock Console or API.
3. If using volume-based discounts (such as Provisioned Throughput for supported foundation models), these must be purchased only using the Amazon Bedrock Console or API.
4. Your role must have access to the inference profile API actions. If your role has the AmazonBedrockFullAccess AWS-managed policy attached, you can skip this step. Otherwise, follow the steps at Creating IAM policies and create the following policy, which allows a role to

do inference profile-related actions and run model inference using all foundation models and inference profiles.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "bedrock:InvokeModel*",
        "bedrock:CreateInferenceProfile"
      ],
      "Resource": [
        "arn:aws:bedrock:*::foundation-model/*",
        "arn:aws:bedrock:*:*:inference-profile/*",
        "arn:aws:bedrock:*:*:application-inference-profile/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "bedrock:GetInferenceProfile",
        "bedrock:ListInferenceProfiles",
        "bedrock:DeleteInferenceProfile",
        "bedrock:TagResource",
        "bedrock:UntagResource",
        "bedrock:ListTagsForResource"
      ],
      "Resource": [
        "arn:aws:bedrock:*:*:inference-profile/*",
        "arn:aws:bedrock:*:*:application-inference-profile/*"
      ]
    }
  ]
}
```

Using the CLI

Now create an application inference profile using the AWS CLI. This profile will serve as the foundation for your model invocations and cost tracking. The process typically involves the following steps:

1. Download and install the latest version of the AWS CLI.
2. Create the application inference profile with specific parameters.
3. Create an application inference profile: Open a new terminal on a machine that has the AWS CLI installed and configured and run the following CLI command:

```
$ aws bedrock create-inference-profile \
  --inference-profile-name "myappinferenceprofile" \
  --model-source "copyFrom=arn:aws:bedrock:us-east-1:123456789123:inference-
profile/us.amazon.nova-pro-v1:0"
```

   Enter a new name for the application inference profile and define a model source. The profile is created using a copy of the system inference profile using the "copyFrom" parameter. In this case I am using the Amazon Nova Pro system inference profile using its ARN. The model source is composed using the format:

```
arn:aws:bedrock:region:accountID:inference-profile/inferenceProfileId
```

   Replace the placeholders for your selected region where the new application inference profile will be created in and add your account ID. Select your preferred system inference profile ID from the list of available system inference profiles. The CLI output shows the created inference profile ARN and status as active.

```
{
"inferenceProfileArn": "arn:aws:bedrock:us-east-1:123456789123:application-
inference-profile/k1c3lwu20lem",
"status": "ACTIVE"
}
```

   The inferenceProfileArn that can be used in other inference profile-related actions and that can be used with model invocation and Amazon Bedrock resources.
4. Once the profile is created, obtain its ARN and use it to apply the "map-migrated" tag to it. This is done using the AWS tagging API.
5. Use the bedrock tag-resource CLI command to tag the created inference profile with the map-migrated tag and the value of your MAP ID following the tagging guidelines.

```
$ aws bedrock tag-resource --resource-arn "arn:aws:bedrock:us-
east-1:123456789123:application-inference-profile/k1c3lwu20lem" --tags "key=map-
migrated,value=mig123456789"
```

For the –resource-arn, Use the generated ARN from previous step. Add the -–tags parameter using the key value pair as shown above replace the value to your corresponding MAP ID with the prefix of "mig".

6. You can list and update inference profiles and their review their tags. To confirm that the application inference profile is correctly tagged, you can list application profiles using the following CLI command:

```
$ aws bedrock list-inference-profiles --type-equals "APPLICATION"
```

This will list all application inference profiles showing the ARN, profile name and ID for each.

```
{
  "inferenceProfileSummaries": [
    {
      "inferenceProfileName": "myappinferenceprofile",
      "createdAt": "2025-01-14T21:21:54.414544+00:00",
      "updatedAt": "2025-01-14T21:21:54.414544+00:00",
      "inferenceProfileArn": "arn:aws:bedrock:us-east-1:123456789123:application-
inference-profile/k1c3lwu20lem",
      "models": [
        {
          "modelArn": "arn:aws:bedrock:us-east-1::foundation-model/amazon.nova-pro-
v1:0"
        },
        {
          "modelArn": "arn:aws:bedrock:us-west-2::foundation-model/amazon.nova-pro-
v1:0"
        },
        {
          "modelArn": "arn:aws:bedrock:us-east-2::foundation-model/amazon.nova-pro-
v1:0"
        }
      ],
      "inferenceProfileId": "k1c3lwu20lem",
      "status": "ACTIVE",
      "type": "APPLICATION"
```

```
    }
  ]
}
```

If you know the inference Profile ID you can get the specific profile details using the CLI command:

```
$ aws bedrock get-inference-profile --inference-profile-identifier "k1c3lwu20lem"
```

```
{
  "inferenceProfileName": "myappinferenceprofile",
  "createdAt": "2025-01-14T21:21:54.414544+00:00",
  "updatedAt": "2025-01-14T21:21:54.414544+00:00",
  "inferenceProfileArn": "arn:aws:bedrock:us-east-1:123456789123:application-
inference-profile/k1c3lwu20lem",
  "models": [
    {
      "modelArn": "arn:aws:bedrock:us-east-1::foundation-model/amazon.nova-pro-
v1:0"
    },
    {
      "modelArn": "arn:aws:bedrock:us-west-2::foundation-model/amazon.nova-pro-
v1:0"
    },
    {
      "modelArn": "arn:aws:bedrock:us-east-2::foundation-model/amazon.nova-pro-
v1:0"
    }
  ],
  "inferenceProfileId": "k1c3lwu20lem",
  "status": "ACTIVE",
  "type": "APPLICATION"
}
```

Using the ARN of the corresponding profile, use the following CLI command to list the resource tags:

```
$ aws bedrock list-tags-for-resource \
  --resource-arn "arn:aws:bedrock:us-east-1:123456789123:application-inference-
profile/k1c3lwu20lem"
```

The output will list tags associated with the profile.

```
{
  "tags": [
    {
      "key": "map-migrated",
      "value": "mig123456789"
    }
  ]
}
```

This tagging structure enables several important capabilities including proper cost allocation tracking for MAP program requirements, clear identification of migrated workloads, and simplified resource management and monitoring.

The "map-migrated" tag serves as an identifier that helps track your migration progress and ensures proper attribution of resource usage within the MAP program framework. All subsequent model invocations using this tagged profile will be properly tracked and accounted for in your MAP metrics.

To create an application inference profile for one Region, specify a foundation model. Usage and costs for requests made to that Region with that model will be tracked.

To create an application inference profile for multiple Regions, specify a cross region (system-defined) inference profile. The inference profile will route requests to the Regions defined in the cross region (system-defined) inference profile that you choose. Usage and costs for requests made to the Regions in the inference profile will be tracked.

Database and analytic tags

Use the following tables for migration plans whose Migration Tracking and Incentive Guide includes database and analytic MAP credits.

Short MPE IDs

Database and analytic tags with short IDs

Source	Destination	Tag key	Tag value
On-premises Commercial DB&A	Any AWS DB&A Service	`map-migrated`	`comm`*`5-digit MPE ID*
On-premises non-commercial DB&A	Any AWS DB&A Service	`map-migrated`	`mig`*`5-digit MPE ID*
On-premises Commercial DB&A	EC2	`map-migrated`	`mig`*`5-digit MPE ID*
On-premises non-commercial DB&A	EC2	`map-migrated`	`mig`*`5-digit MPE ID*
EC2 Commercial DB&A	Any AWS DB&A Service	`map-migrated`	`comm_ec2_` *`5-digit MPE ID`*
EC2 non-commercial DB&A	Any AWS DB&A Service	`map-migrated`	`mig_ec2_`*`5-digit MPE ID`*

Long MPE IDs

Database and analytic tags with long IDs

Source	Destination	Tag key	Tag value
On-premises Commercial DB&A	Any AWS DB&A Service	`map-migrated`	`comm`*`10 alphanumeric MPE ID characters`*
On-premises non-commercial DB&A	Any AWS DB&A Service	`map-migrated`	`mig`*`10 alphanumeric MPE ID characters`*
On-premises Commercial DB&A	EC2	`map-migrated`	`mig`*`10 alphanumeric MPE ID characters`*

Source	Destination	Tag key	Tag value
On-premises non-commercial DB&A	EC2	`map-migrated`	`mig`*`10 alphanumeric MPE ID characters`*
EC2 Commercial DB&A	Any AWS DB&A Service	`map-migrated`	`comm_ec2_` *`10 alphanumeric MPE ID characters`*
EC2 non-commercial DB&A	Any AWS DB&A Service	`map-migrated`	`mig_ec2_`*`10 alphanumeric MPE ID characters`*

Note

Use lowercase letters for the `comm`, `mig`, `comm_ec2_`, and `mig_ec2_` prefixes and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about your MPE ID, see the section called “MPE ID length”.

A commercial database is defined as any third-party database engine, data warehouse, or analytics offering for which you have paid a fee for use rights, enhancements, maintenance, or support for that third party offering. A non-commercial database is defined as any third-party database engine, data warehouse, or analytics offering for which you have not paid a fee for use rights, enhancements, maintenance, or support for that third party offering. For a complete list of services that are included in MAP – Database & Analytics (DB&A), see the MAP 2.0 Included Services list: **https://s3-us-west-2.amazonaws.com/map-2.0-customer-documentation/included-services/MAP_Included_Services_List.pdf**.

Special scenario

If your destination database is changed from EC2 to any AWS DB&A service after your initial migration, then use the following tables to tag your DB&A service.

Short MPE IDs

EC2 to database and analytic tags with short IDs after migration

Source	Destination	Tag key	Tag value
EC2 Commercial DB&A	Any AWS DB&A Service	`map-migrated`	`comm`*`5-digit MPE ID*
EC2 non-commercial DB&A	Any AWS DB&A Service	`map-migrated`	`mig`*`5-digit MPE ID*

Long MPE IDs

EC2 to database and analytic tags with long IDs after migration

Source	Destination	Tag key	Tag value
EC2 Commercial DB&A	Any AWS DB&A Service	`map-migrated`	`comm`*`10 alphanumeric MPE ID characters`*
EC2 non-commercial DB&A	Any AWS DB&A Service	`map-migrated`	`mig`*`10 alphanumeric MPE ID characters`*

Note

Use lowercase letters for the `comm` and `mig` prefixes and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about your MPE ID, see the section called “MPE ID length”.

SAP workload tags

Use the following tables for migration plans using MAP 2.0 for SAP Migration.

Short MPE IDs

SAP workload tags with short ID

Source	Destination	Tag key	Tag value
On-premises	AWS	`map-migrated`	`sap`*`5-digit MPE ID`*

Long MPE IDs

SAP workload tags with long ID

Source	Destination	Tag key	Tag value
On-premises	AWS	`map-migrated`	`sap`*`10 alphanumeric MPE ID characters`*

> **Note**
>
> The prefix for SAP workload tags is `sap`. Do not use this tag for migration plans that are not part of MAP 2.0 for SAP. Use uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about your MPE ID, see the section called “MPE ID length”.

Ensure you define the MAP tags as part of your infrastructure definition when using the AWS Launch Wizard for SAP to deploy your SAP workloads. For more information about the AWS Launch Wizard for SAP, see Deploy an SAP application with AWS Launch Wizard in the *AWS Launch Wizard user guide*. For a complete list of services included in MAP 2.0 for SAP, see the MAP 2.0 Included Services list: **https://s3-us-west-2.amazonaws.com/map-2.0-customer-documentation/included-services/MAP_Included_Services_List.pdf**.

Oracle tags

Use the following tables for migration plans using MAP 2.0 for Oracle Migration.

Short MPE IDs

Oracle workload tags with short ID

Source	Destination	Tag key	Tag value
On-premises	AWS	`map-migrated`	`oracle`*`5-digit MPE ID`*

Long MPE IDs

Oracle workload tags with long ID

Source	Destination	Tag key	Tag value
On-premises	AWS	`map-migrated`	`oracle`*`10 alphanumeric MPE ID characters`*

> **Note**
>
> The prefix for Oracle workload tags is `oracle`. Do not use this tag for migration plans that are not part of MAP 2.0 for Oracle. Use uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about your MPE ID, see the section called “MPE ID length”.

Ensure you define the MAP tags as part of your infrastructure definition when using the AWS Launch Wizard to deploy your Oracle workloads. For more information about the AWS Launch Wizard, see Get started with AWS Launch Wizard in the *AWS Launch Wizard user guide*. For a complete list of services included in MAP 2.0 for Oracle, see the MAP 2.0 Included Services list: **https://s3-us-west-2.amazonaws.com/map-2.0-customer-documentation/included-services/MAP_Included_Services_List.pdf**.

Other examples

Use the following examples to further help you tag your migrated workloads.

Example 1: Re-hosting using AWS Transform MGN (MGN)

Use this example if you are moving from on-premises to AWS using a lift-and-shift (re-hosting) migration pattern, and decided to use MGN for the migration.

Short MPE IDs

Re-hosting using MGN with short ID example

Tag key (automated)	Tag value (automated)
`map-migrated`	`mig`*`5-digit MPE ID`*

Long MPE IDs

Re-hosting using MGN with long ID example

Tag key (automated)	Tag value (automated)
`map-migrated`	`mig`*`10 alphanumeric MPE ID characters`*

> **Note**
>
> Use lowercase letters for the `mig` prefix and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about your MPE ID, see the section called “MPE ID length”.

Example 2: DataCenter Migration (mix of migration patterns)

Use this example if you are moving different workloads from on-premises to AWS using various migration patterns (re-hosting, re-architecting, etc.) as part of general MAP.

Short MPE IDs

DataCenter migration (mix of migration patterns) with short ID example

Tag key	Tag value
`map-migrated`	`mig`*`5-digit MPE ID`*

Long MPE IDs

DataCenter migration (mix of migration patterns) with long ID example

Tag key	Tag value
`map-migrated`	`mig`*`10 alphanumeric MPE ID characters`*

> **Note**
>
> Use lowercase letters for the `mig` prefix and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about your MPE ID, see the section called “MPE ID length”.

Example 3: Migrate commercial database from EC2 to RDS

Use this example if you are moving a commercial databases from Amazon EC2 instances on AWS to Amazon RDS as part of MAP for Database and Analytics.

Short MPE IDs

Migrate commercial database from EC2 to RDS with short ID example

Tag key	Tag value
`map-migrated`	`comm_ec2_` *`5-digit MPE ID`*

Long MPE IDs

Migrate commercial database from EC2 to RDS with long ID example

Tag key	Tag value
map-migrated	comm_ec2_ *10 alphanumeric MPE ID characters*

> **Note**
>
> Use lowercase letters for the comm_ec2_ prefix and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about your MPE ID, see the section called "MPE ID length".

Example 4: Database modernization

Use this example if you are moving from on-premises commercial database server to Amazon DynamoDB. This example is for a Migration Plan that is eligible for Database & Analytics MAP Credits.

Short MPE IDs

Database modernization with short ID example

Tag key	Tag value
map-migrated	comm*5-digit MPE ID*

Long MPE IDs

Database modernization with long ID example

Tag key	Tag value
map-migrated	comm*10 alphanumeric MPE ID characters*

Note

Use lowercase letters for the `comm` prefix and uppercase letters for the alphanumeric MPE IDs (long MPE IDs). For more information about your MPE ID, see the section called "MPE ID length".

Frequently Asked Questions (FAQs)

Below you'll find answers to the most common questions you may have on MAP 2.0.

Topics

How can I create Amazon Bedrock inference profiles to use supported foundation models?

See Track, allocate, and manage your generative AI cost and usage with Amazon Bedrock. You must tag the inference profile as described in the section called "Tagging Resources".

What are the supported foundation models in Amazon Bedrock?

For a list of supported foundation models that are eligible for MAP credits, see Supported foundation models in Amazon Bedrock. To gain access to these supported foundation models and purchase any provisioned throughput, you must use the Amazon Bedrock Console or API.

Why is tagging required for MAP 2.0?

Tagging is a way to assign a label to an AWS resource. Tags enable you to categorize your AWS resources in different ways. For example, you can categorize by purpose, owner, department,or environment. Each tag consists of a key and an optional value (for example, Department: Finance), both of which you define. For the purposes of MAP, you must tag migrated workloads with the `map-migrated` tag to enable the following:

- Tracking of the migration inventory scope, as it is migrated over time from your existing environment to AWS.
- Identification of the specific AWS resources being used in place of existing pre-migration resources.

> **Important**
>
> You must use the tag key `map-migrated` exactly as it appears here. You cannot introduce spaces, change the case of any of the letters, or alter the key in any way.

The tagging exercise is done when the workload is migrated to AWS. Typically, tags are applied by the workload owners who will be migrating their workloads. This process is repeated as workloads are moved across until the entire MAP migration scope has been migrated. Tagging your AWS resources is a best practice even outside of MAP. For more information, see the Tagging Best Practices Whitepaper.

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost allocation tag. Tags that are automatically activated don't count towards your cost allocation tag quota. For more information, see Quotas and restrictions.

Should I activate the Cost Allocation Tag?

The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost allocation tag in the management (payer) account that is mentioned in the MAP project plan. For the MAP 2.0 terms signed between November 01, 2022 and November 18, 2024, the cost-allocation tag setup is not automatically activated. See the section called “Verify and Fix map-migrated cost-allocation tag”.

Should migrations occur under the management (payer) account?

Migrations must occur under the management (payer) account and any accounts linked to the management (payer) account. All migrated resources that are part of the migration plan must be tagged with `map-migrated` as defined in this guide.

> **Important**
>
> You must use the tag key `map-migrated` exactly as it appears here. You cannot introduce spaces, change the case of any of the letters, or alter the key in any way.

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost allocation tag. Tags that are automatically activated don't count towards your cost allocation tag quota. For more information, see Quotas and restrictions.

Where do I find my MPE ID?

Your project number found within your migration plan is also known as your MPE ID. For more information about your MPE ID, see the section called “MPE ID length”.

How do I tag my migrated resources in AWS with `map-migrated` tag?

While we encourage you to automate tagging your resources, you can tag your resources in AWS in the following ways:

- If you are rehosting servers, use AWS Application Migration Service (MGN) for your migration.
- MGN supported MAP auto-tagging.
- You can go to each resource in AWS console and create tags.
- You can write custom scripts to bulk tag your resources during or after the creation.

> **Important**
>
> You must use the tag key `map-migrated` exactly as it appears here. You cannot introduce spaces, change the case of any of the letters, or alter the key in any way.

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost allocation tag. Tags that are automatically activated don't count towards your cost allocation tag quota. For more information, see Quotas and restrictions.

I've already onboarded to MAP 2.0 and have been following the tagging process that required CUR and a Server ID from Migration Hub. How should I proceed going forward?

For the previously existing MAP 2.0 projects, keep tagging as it was defined at the time the MAP 2.0 term was signed. For the MAP 2.0 terms signed after November 01, 2022, the related workloads should follow this tagging guide.

What is the process for baseline AWS services that cannot be tagged but may also be included in MAP (which can include VMware Cloud on AWS, Amazon Connect or AWS Managed Services)?

The process is automated for Amazon Connect, VMware Cloud on AWS, and AWS Managed Services (AMS).

Will I receive MAP incentives if I tag an existing AWS resource?

The Program Terms clearly state that only MAP resources that are launched after the agreement acceptance date are eligible for MAP incentives.

Can I use an Amazon S3 bucket that existed before the MAP agreement as destination for my migration?

You are recommended to use new S3 buckets for migration workloads. If, for any reason you have to use existing S3 buckets for migration workloads, you need to provide to your account team with the ARNs for each of the existing S3 buckets be used for migration and each bucket's current baseline spend. This information is added to the MAP agreement. You will then need to tag these buckets during migration. For more information on how to baseline the existing S3 buckets, see the How do I find the cost of my Amazon S3 buckets guide.

What services are in scope for MAP 2.0?

You can find all eligible services for MAP in the Included Services list: **https://s3-us-west-2.amazonaws.com/map-2.0-customer-documentation/included-services/MAP_Included_Services_List.pdf**. Certain Specialized Services may also be included in your Migration Plan or MAP Migration Tracking and Incentive Guide. Eligible services for MAP for SAP are provided in the MAP for SAP Migration Tracking and Incentive Guide.

Are tags case sensitive?

Yes. See the section called “MAP tag”.

Can I use Amazon EC2 Dedicated Hosts as part of MAP 2.0?

Yes, it is possible for you to use Amazon EC2 Dedicated Hosts as part of MAP 2.0.

What should I do if I don't see the `map-migrated` tag in the Cost allocation tags screen?

The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost-allocation tag in the management (payer) account that is mentioned in the MAP project plan. For the MAP 2.0 terms signed between November 01, 2022 and November 18, 2024, the cost-allocation tag setup is not automatically activated. See the section called “Verify and Fix map-migrated cost-allocation tag”.

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. This tag is automatically activated for you as a cost allocation tag. Tags that are automatically activated don't count towards your cost allocation tag quota. For more information, see Quotas and restrictions.

Troubleshooting

Verify and Fix map-migrated cost-allocation tag

The Migration Acceleration Program requires that you tag resources with the `map-migrated` tag. The `map-migrated` tag is automatically activated for you as a cost-allocation tag. For a very small set of dated migration project plans, the `map-migrated` tag is not automatically activated and requires a manual setup. Following provides you steps on how to verify that the `map-migrated` tag is activated and if it isn't activated, how to manually set it up.

Verify that the `map-migrated` tag is activated as a cost-allocation tag

1. Sign in to the AWS Management Console and open the AWS Billing and Cost Management console at https://console.aws.amazon.com/costmanagement/.
2. In the navigation pane, choose **Cost allocation tags**.
3. To filter, copy the following tag key and enter it in the search box.

   ```
   map-migrated
   ```

4. The status of the tag must appear as **Active**.
5. If the `map-migrated` tag is not found, or tag status is not **Active**, perform the following manual setup.

Manually set up the `map-migrated` tag and activate it as a cost-allocation tag

1. In the AWS Billing and Cost Management console, log in to the management (payer) account(s) listed in your Migration Plan.
2. Create an empty Amazon S3 bucket.
3. Copy the following tag key and tag the resource with it. The tag value can be empty.

   ```
   map-migrated
   ```

4. Wait 24 hours.

> **Note**
>
> It might take up to 24 hours for the cost allocations to appear available in the system. Therefore, if you don't see the the section called “MAP tag”, wait for 24 hours, and then refresh the cost allocation tag screen.

5. After waiting 24 hours, log in to the management (payer) account(s) listed in your Migration Plan to activate the cost allocation tags that apply to your workload.
6. In the AWS Management Console, choose **Services**.
7. Choose **Billing** from the Services menu.
8. Choose **Cost allocation tags** from the navigation panel.
9. To filter for MAP-migrated resources, enter the the section called “MAP tag” key in the search box.
10. Choose the **check boxes** for the tags created for the the section called “MAP tag”.
11. Choose **Activate**.

    The status of the tags should now appear as **Active**.

Next step

Tagging Resources

Document history for the AWS Migration Acceleration Program 2.0 User Guide

The following table describes the documentation releases for AWS Migration Acceleration Program 2.0.

Change	Description	Date
Added IAM principal tagging for Amazon Bedrock	Added documentation for IAM principal tagging as the recommended method to tag Amazon Bedrock and Amazon Bedrock AgentCore workloads for MAP spend tracking.	June 5, 2026
Initial release	Initial release of the AWS Migration Acceleration Program 2.0 User Guide	July 1, 2022