User Guide for MAP for AWS Transform

AWS Migration Acceleration Program 2.0 Tagging Guide for AWS Transform



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Migration Acceleration Program 2.0 Tagging Guide for AWS Transform: User Guide for MAP for AWS Transform

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is AWS Migration Acceleration Program 2.0? 1
Getting started 2
Prerequisites 2
PRM ID length 3
Tagging Resources 3
MAP tag 4
Tagging with AWS Tools 4
Auto-tagging in MAP for AWS Transform 16
Manual tagging 21
Tagging key combinations 23
Services 25
SAP 25
Oracle 25
Amazon Bedrock MAP Tagging using the AWS CLI 25
IAM principal tagging (recommended) 26
Resource tagging (application inference profiles) 28
Projects tagging (bedrock-mantle endpoint) 35
Database and analytics tags 38
SAP workload tags 39
Oracle tags 40
Other examples 41
Example 1: Re-hosting using AWS Transform MGN (MGN) 41
Example 2: DataCenter Migration (mix of migration patterns) 41
Example 3: Migrate commercial database from EC2 to RDS 42
Example 4: Database modernization 42
Frequently Asked Questions (FAQs) 44
How can I create Amazon Bedrock inference profiles to use supported foundation models? 45
What are the supported foundation models in Amazon Bedrock? 45
Why is tagging required for MAP 2.0? 45
Should I activate the Cost Allocation Tag? 46
Should migrations occur under the management (payer) account? 46
How do I tag my migrated resources in AWS with `aws-apn-id` tag? 47
What is the process for baseline AWS services that cannot be tagged but might also be included in MAP (which can include Amazon Connect or AWS Managed Services)? 48

Will I receive MAP incentives if I tag an existing AWS resource? 48
Can I use an Amazon S3 bucket that existed before the MAP agreement as destination for my migration? 48
What services are in scope for MAP 2.0? 48
Are tags case sensitive? 49
Can I use Amazon EC2 Dedicated Hosts as part of MAP 2.0? 49
What should I do if I don't see the `aws-apn-id` tag in the Cost allocation tags screen? 49
Troubleshooting 50
Verify and Fix aws-apn-id cost-allocation tag 50
Next step 51
Document history 52

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 `aws-apn-id` 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 `aws-apn-id` 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
- PRM ID length
- Tagging Resources

Prerequisites

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

- You must have your PRM ID number. For more information, see the section called “PRM ID length”.
- You must have access to the AWS management (payer) accounts as part of the agreement.
- Ensure that Cost Explorer is enabled. To learn how to enable Cost Explorer, see Enabling Cost Explorer.
- IAM permissions to tag migrated workloads.

PRM ID length

Your project number is also known as your PRM ID. The PRM ID follows the format `ra-` followed by 13 alphanumeric characters.

For example:

- **Project ra-abcdefg123456**: PRM ID = ra-abcdefg123456

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 `aws-apn-id` 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 `aws-apn-id` 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
- Tagging with AWS Tools
- Auto-tagging in MAP for AWS Transform
- Manual tagging

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:

```
aws-apn-id
```

> **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.

Tagging with AWS Tools

We encourage you to use AWS tools when tagging your resources as much as possible. You can bulk apply tags 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 help with 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 **PRM ID** with the tag value you want to apply to the migrated workloads.

   Example:

   - If your PRM ID is *ra-abcdefg123456*, use the value *migra-abcdefg123456*.

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 PRM ID, the tag value can be any of the following:

- `mig`*PRM ID*
- `sap`*PRM ID*
- `oracle`*PRM ID*

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes. For more information about what tag values you should use, see *Tagging key combinations*. For more information about your PRM ID, see the section called "PRM 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 `migra-abcdefg123456` with the tag value needed for your migrated resource followed by your PRM ID in the following example:

```
{
    "Key" : "aws-apn-id",
    "Value" : "migra-abcdefg123456"
}
```

Note

The Migration Acceleration Program requires that you tag resources with the `aws-apn-id` 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 PRM ID, the tag value can be any of the following:

- `mig`*PRM ID*
- `sap`*PRM ID*
- `oracle`*PRM ID*

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes. For more information about what tag values you should use, see *Tagging key combinations*. For more information about your PRM ID, see the section called "PRM 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 `migra-abcdefg123456` with the tag value needed for your migrated resource followed by your PRM ID in the following example:

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

Tags.of(myConstruct).add('aws-apn-id', 'migra-abcdefg123456');
```

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `aws-apn-id` 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 PRM ID, the tag value can be any of the following:

- `mig`*PRM ID*
- `sap`*PRM ID*
- `oracle`*PRM ID*

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes. For more information about what tag values you should use, see *Tagging key combinations*. For more information about your PRM ID, see the section called "PRM 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 `aws-apn-id` as the Tag key.
5. Enter and replace your **PRM ID** with the tag value you want to apply to the migrated workloads.

   Example:

   - If your PRM ID is *`ra-abcdefg123456`*, use the value *`migra-abcdefg123456`*.
6. Choose **Add tags**.

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

- `mig`*PRM ID*
- `sap`*PRM ID*
- `oracle`*PRM ID*

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes. For more information about what tag values you should use, see *Tagging key combinations*. For more information about your PRM ID, see the section called "PRM 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 `aws-apn-id` 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 `migra-abcdefg123456` 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:
       aws-apn-id: "migra-abcdefg123456"

  HttpApi:
    Tags:
       aws-apn-id: "migra-abcdefg123456"
```

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `aws-apn-id` 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 PRM ID, the tag value can be any of the following:

- `mig`*PRM ID*
- `sap`*PRM ID*
- `oracle`*PRM ID*

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes. For more information about what tag values you should use, see *Tagging key combinations*. For more information about your PRM ID, see the section called “PRM 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 `aws-apn-id` in the Tag Key field.

   > **Note**
   >
   > The Migration Acceleration Program requires that you tag resources with the `aws-apn-id` 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 **PRM ID** with the tag value you want to apply to the migrated workloads.

   Example:

   - If your PRM ID is *`ra-abcdefg123456`*, use the value *`migra-abcdefg123456`*.
10. Choose **Review**.
11. Choose **Apply tag** changes.

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

- `mig`*`PRM ID`*

- `sap`*PRM ID*
- `oracle`*PRM ID*

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

Auto-tagging in MAP for AWS Transform

What is auto-tagging?

Auto-tagging is a feature of AWS Migration Acceleration Program (MAP) for AWS Transform that applies MAP tags to the AWS resources that AWS Transform agents create when you migrate or modernize workloads to AWS. The following agents support auto-tagging:

- **AWS Transform migration agent** (previously called the VMware migration agent): server and VM rehosting, and source code containerization.
- **AWS Transform Windows (SQL Server) modernization agent**: SQL Server to Amazon Aurora PostgreSQL.

These tags enable AWS to track your migration spend and apply your MAP credits.

Why it matters

MAP credits are applied only to tagged spend, and only from the moment a tag is correctly in place. There is no retroactive credit for untagged resources. In the past, you had to apply these tags yourself, which risked missed or incorrect tags. Auto-tagging applies the correct tag automatically, at the moment the resource is created, every time, so that no eligible spend is missed.

How it works

Once your engagement and workspace are configured, auto-tagging is fully managed. You do not need to manually apply tags to individual resources. The agent handles that automatically. The process works as follows:

1. **Your engagement is registered.** When your Express Migration qualifies for MAP, a unique attribution identity is generated for your migration. This identity consists of:
   - A **tag key**: `aws-apn-id`
   - A unique **tag value** tied to your engagement
2. **Agents receive tagging instructions automatically.** The agent knows which tag to apply as soon as your migration wave runs.
3. **Tags are applied at resource creation.** The agent applies the correct tag to every supported resource it creates.

Supported use cases

The following table lists the migration types that support auto-tagging, the AWS Transform agent used, the resources each type tags, and example workloads.

Migration type	AWS Transform agent	Resources tagged	Workload examples
Server or VM rehosting	Migration agent	Amazon EC2 instances and their Amazon EBS volumes; ENIs created at launch; Amazon FSx for NetApp ONTAP SVMs (when FSx storage is used); VPCs and security groups	VMware workloads migrated using the AWS Transform migration agent
Source code containerization	Migration agent	Amazon ECR repositories (one per service); Amazon S3 bucket and AWS CodeBuild CloudWatch log group created for the migration (one per connector). Amazon ECS targets: resources created by the generated Terraform (ECS cluster, services, task definitions, ALB, target groups, listeners, log groups, Route 53 private hosted zone, AWS Cloud Map service, Amazon EFS, access-log bucket).	Application source code containerized and deployed to Amazon ECS or Amazon EKS

Migration type	AWS Transform agent	Resources tagged	Workload examples
		Amazon EKS targets: the ALB (when the Helm chart includes an Ingress)	
Database modernization	Windows (SQL Server) moderniza tion agent	Amazon Aurora clusters	SQL Server to Amazon Aurora PostgreSQL

Resources not auto-tagged

The following resources are not auto-tagged:

- **Source code containerization:**
  - Temporary CodeBuild projects.
  - Amazon ECS tasks and ECS-managed Amazon EBS volumes.
  - On EKS: the ALB log bucket, hosted zone, EKS access entry, and EBS or EFS volumes created from PVCs.
- **Database modernization:** Amazon RDS and Amazon DynamoDB.
- **Customer-owned or pre-existing resources:** existing VPCs, IAM roles, ECS or EKS clusters, ECR repositories, S3 buckets, or log groups. Auto-tagging applies only to resources the agent creates.
- **Migration of existing ECS or EKS clusters:** not supported.

> **Note**
>
> Tagging the S3 bucket is best-effort. For resources that are not auto-tagged, apply the tag manually. For instructions, see the section called “Tagging Resources”.

How auto-tagging is triggered

Auto-tagging is not a setting you turn on. It activates automatically when the following conditions are met. You must first set up your workspace and migration job (see the section called “What you need to do”).

1. The APN opportunity is **approved**.
2. The migration details are **submitted** (not just saved).
3. The engagement is in **review** status.

Once these conditions are true, the tag appears in your engagement. If the tag is missing, verify these three conditions before you proceed.

What you need to do

Complete the following steps to set up auto-tagging for your migration:

1. Create a **workspace** in AWS Transform.
2. Attach the workspace to your engagement.
3. Create a **migration job** in the workspace. For VMware migrations, select **Migration V3**.
4. Run migration planning and accept the plan.
5. When connecting to the AWS target account, **select the APN tag**. The migration agent uses this tag for all resources it creates.
6. Run the migration job for the servers in scope.
7. Apply the MAP tag to the server inventory, using one of the following options:

   Option A – Update the inventory CSV

   a. Download the inventory of the servers to be migrated from the **Artifacts** tab of your workspace.

   b. Update the tagging fields in the CSV:

Field	Value
`map-tagging`	TRUE
`map-tag-key`	`aws-apn-id`

Field	Value
`map-tag-value`	Your tag value (for example, `migra-abcdefg123456`)

c. Import the updated CSV file back into the migration job.

Option B – Ask the migration agent in chat

In the job chat, ask the migration agent to apply the MAP tag to the servers in scope. For example:

```
Apply the MAP tag aws-apn-id with value migra-abcdefg123456 to all servers in this
 wave.
```

Validating the tag

Before you proceed with the migration

Before you start the migration, confirm with the migration agent in the job chat which tag key and tag value it will apply to the migrated resources. This lets you catch an incorrect or missing tag before any resources are created, and before any untagged spend occurs.

For example, ask the agent:

```
Which MAP tag key and value will be applied to the migrated instances?
```

Confirm that the agent returns:

Field	Expected value
Tag key	`aws-apn-id`
Tag value	Your tag value (for example, `migra-abcdefg123456`)

If the tag key or value is incorrect or missing, update it (see step 7 in the section called “What you need to do”) and confirm with the agent again before you proceed.

After the migration completes

Verify that the tag was applied correctly to the migrated resources:

1. Copy the instance ID (or resource ID) from the migration job results.
2. Open the relevant AWS console (for example, Amazon EC2) in the target account.
3. Confirm that the tag `aws-apn-id` appears on the resource with the correct value.

If the tag is missing from a migrated resource:

- Confirm that the APN tag was selected during the target account connection step.
- Confirm that the tagging fields were applied to the inventory, either by re-importing the CSV with `map-tagging = TRUE` and the correct `map-tag-key` and `map-tag-value`, or by asking the migration agent in chat. You can download the inventory from the **Artifacts** tab of your workspace.
- If the migration has already completed and the APN tag was not inherited or applied, you can apply the tag manually to the resource. This way you don't need to re-run the migration job. For instructions, see the section called “Tagging Resources”.

Related resources

- AWS Transform VMware workshop: Create a workspace and job

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 `aws-apn-id` as the Tag key.

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `aws-apn-id` 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 **PRM ID** with the tag value you want to apply to the migrated workloads.

   Example:

   - If your PRM ID is *`ra-abcdefg123456`*, use the value *`migra-abcdefg123456`*.

6. Choose **Save**.

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

- `mig`*`PRM ID`*
- `sap`*`PRM ID`*
- `oracle`*`PRM ID`*

> **Note**
>
> Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes. For more information about what tag values you should use, see *Tagging key combinations*. For more information about your PRM ID, see the section called "PRM 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*.

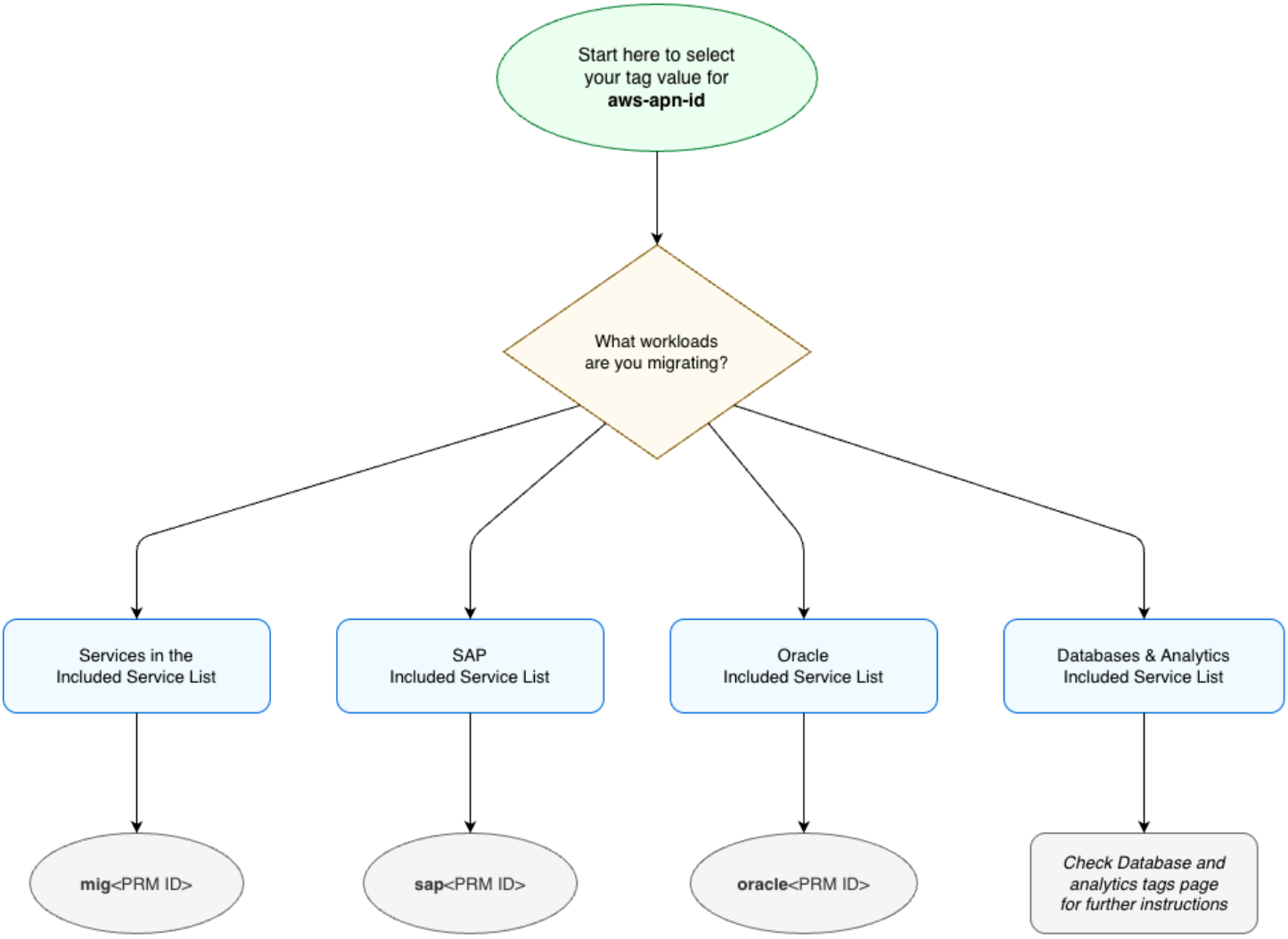
Tagging key combinations

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

- What AWS service will be the destination for your migration?
- Where is the source of your migration (on-prem or already in AWS as Amazon EC2)?
- 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**
  - **Database and analytics included in the Service list**: For more information, see the section called “Database and analytics tags”.

Note

Use lowercase letters for the `mig`, `sap`, and `oracle` prefixes. For more information about your PRM ID, see the section called “PRM ID length”.

Services

`mig`*`PRM ID`*

Example: `migra-abcdefg123456`

SAP

`sap`*`PRM ID`*

Example: `sapra-abcdefg123456`

Oracle

`oracle`*`PRM ID`*

Example: `oraclera-abcdefg123456`

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.

For more information about Amazon Bedrock cost management and tagging methods, see Track usage and costs in Amazon Bedrock in the *Amazon Bedrock User Guide*.

Use the following guidance to choose a tagging method. There are three methods to tag your Amazon Bedrock workloads for MAP:

- **IAM principal tagging (recommended)** — Tag the IAM role used to invoke Bedrock APIs with `aws-apn-id`. 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 `aws-apn-id`, and invoke models through the profile. This method provides per-model and per-region cost granularity but requires creating and managing additional Bedrock resources.
- **Projects tagging (bedrock-mantle endpoint)** — Create an Amazon Bedrock project, tag it with `aws-apn-id`, and pass the project ID in your API calls to the bedrock-mantle endpoint. This method is available for the Responses API, Chat Completions API, and Messages API.

> **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 principal (role or user) used for Amazon Bedrock or Amazon Bedrock AgentCore API calls with the `aws-apn-id` 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. You must have your PRM ID. For more information about PRM ID format and length, see the section called “PRM ID length”.
2. Create a new IAM principal (user or role) dedicated to your Amazon Bedrock or Amazon Bedrock AgentCore workloads initiated and deployed during the MAP Credit Period.
3. 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*.
4. Activate the `aws-apn-id` 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.
5. Your role must have permission to tag IAM principals. If your role has the IAMFullAccess AWS-managed policy attached, you can skip this step. Otherwise, ensure your role has

`iam:TagRole` and `iam:ListRoleTags` permissions (for tagging roles) or `iam:TagUser` and `iam:ListUserTags` permissions (for tagging users).

Tagging your IAM role

Tag the IAM role that is used to invoke Amazon Bedrock or AgentCore APIs with the `aws-apn-id` tag and your PRM 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 `aws-apn-id` tag to the role:

   ```
   $ aws iam tag-role --role-name MyBedrockRole --tags "Key=aws-apn-
   id,Value=migYOUR_PRM_ID"
   ```

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

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

   The output should show the `aws-apn-id` tag:

   ```
   {
     "Tags": [
       {
         "Key": "aws-apn-id",
         "Value": "migra-abcdefg123456"
       }
     ]
   }
   ```

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/aws-apn-id` tag will appear in your Cost and Usage Report (CUR) data for eligible billing lines.

Tagging IAM Users

If your application calls Amazon Bedrock using an IAM user rather than an IAM role, tag the user instead:

```
$ aws iam tag-user --user-name MyBedrockUser --tags "Key=aws-apn-
id,Value=migYOUR_PRM_ID"
```

Verify with:

```
$ aws iam list-user-tags --user-name MyBedrockUser
```

MAP spend tracking recognizes the `aws-apn-id` tag on any IAM principal that appears on an eligible billing line.

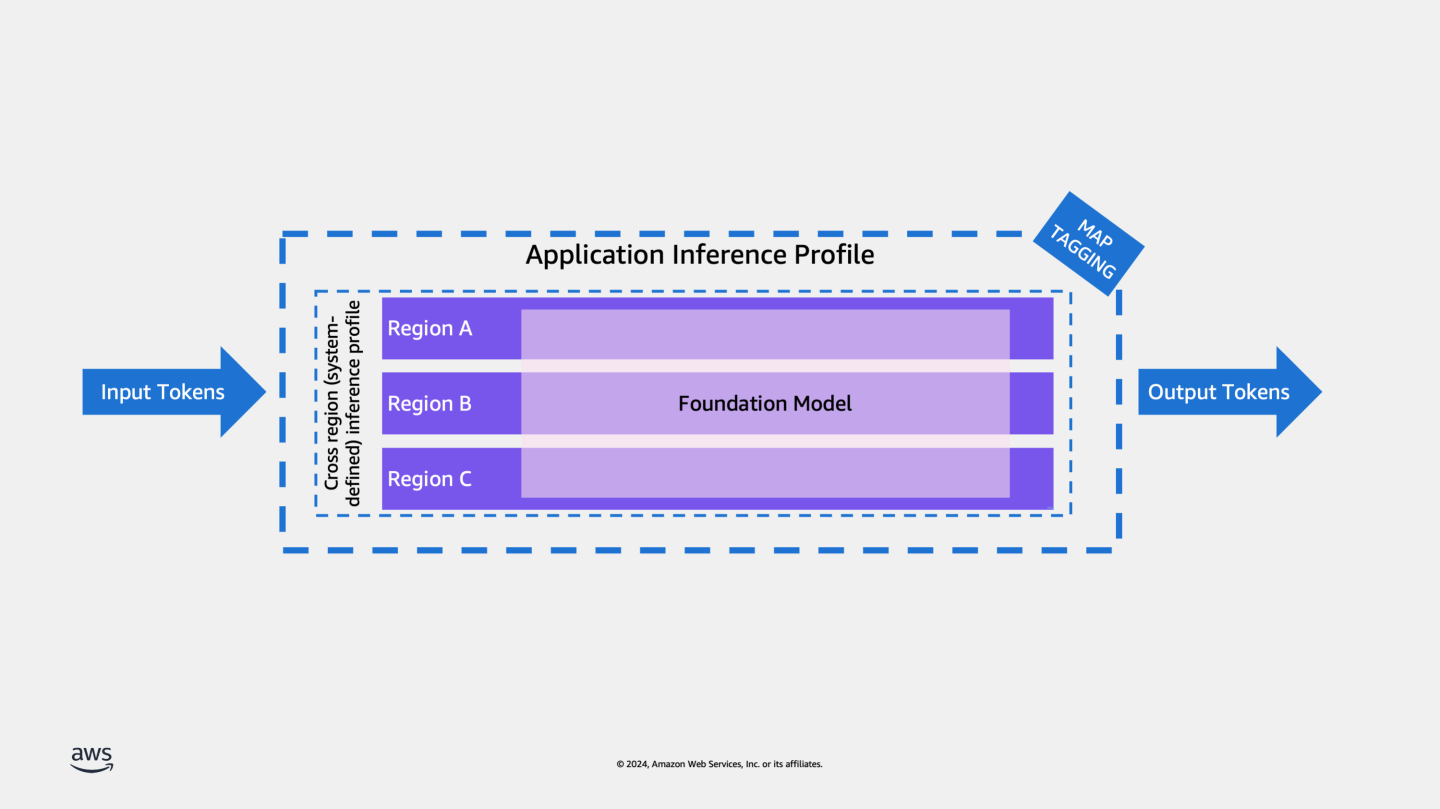
Important considerations

- **Supported principal types**: Both IAM roles and IAM users are supported. Tag whichever principal your application uses to invoke Amazon Bedrock APIs.
- **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 `aws-apn-id` tag (for example, on an application inference profile) and an IAM principal `aws-apn-id` 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 PRM ID (for example, `migra-abcdefg123456`).

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 `aws-apn-id`. 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. You must have your PRM ID. For more information about PRM ID format and length, see the section called “PRM ID length”.
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. Create new inference profiles dedicated to your Amazon Bedrock or Amazon Bedrock AgentCore workloads initiated and deployed during the MAP Credit Period.

5. 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 "aws-apn-id" 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 aws-apn-id tag and the value of your PRM ID following the tagging guidelines.

```
$ aws bedrock tag-resource --resource-arn "arn:aws:bedrock:us-
east-1:123456789123:application-inference-profile/k1c3lwu20lem" --tags "key=aws-
apn-id,value=migra-abcdefg123456"
```

   For the –resource-arn, use the generated ARN from the previous step. For the -–tags parameter, replace the value with your PRM ID prefixed by `mig` (for example, `migra-abcdefg123456`).

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": "aws-apn-id",
      "value": "migra-abcdefg123456"
    }
  ]
}
```

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 "aws-apn-id" 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.

Projects tagging (bedrock-mantle endpoint)

Amazon Bedrock Projects lets you attribute MAP spend to specific workloads. Use Projects tagging when you call the APIs on the bedrock-mantle endpoint — the Responses API, Chat Completions API, or Messages API. A project is a logical boundary that represents a workload, such as an application, environment, or experiment. You tag the project with `aws-apn-id` and pass the project ID in your API calls. AWS Cost Explorer and AWS Cost and Usage Reports (CUR 2.0) then track these costs.

> **Note**
>
> Projects tagging is available for the APIs on the bedrock-mantle endpoint (Responses API, Chat Completions API, and Messages API). If you use the native Amazon Bedrock Runtime APIs (`InvokeModel` / `Converse`), use the IAM principal tagging or resource tagging (application inference profiles) methods described earlier in this guide.

Prerequisites

- You must have your PRM ID. For more information about PRM ID format and length, see the section called “PRM ID length”.
- You must have access to Amazon Bedrock with an API key for the bedrock-mantle endpoint. For more information, see Getting started with Amazon Bedrock.
- You must have IAM permissions for Amazon Bedrock Projects, inference, and tagging. You can attach the AWS managed policy `AmazonBedrockMantleFullAccess`. For production, implement least privilege policies.
- Activate the `aws-apn-id` tag as a cost allocation tag (CAT) in the Billing console under **Cost allocation tags**.

Step 1: Create a project with the aws-apn-id tag

Create an Amazon Bedrock project and include the `aws-apn-id` tag with your PRM ID in the project's tag set. You can add additional cost allocation tags (such as `Application`, `Environment`, or `Team`) to further segment your costs.

Use `curl` to call the Projects API on the bedrock-mantle endpoint:

```
$ curl -X POST "$OPENAI_BASE_URL/organization/projects" \
    -H "Authorization: Bearer $OPENAI_API_KEY" \
    -H "Content-Type: application/json" \
    -d '{
      "name": "MyMigratedApp-Prod",
      "tags": {
        "aws-apn-id": "migra-YOUR_PRM_ID",
        "Application": "MyMigratedApp",
        "Environment": "Production",
        "Team": "PlatformEngineering",
```

```
        "CostCenter": "CC-2002"
      }
    }'
```

Replace *YOUR_PRM_ID* with your PRM ID. Set the environment variables before running:

```
$ export OPENAI_BASE_URL="https://bedrock-mantle.us-east-1.api.aws/openai/v1"
$ export OPENAI_API_KEY="YOUR-BEDROCK-API-KEY"
```

The command returns the project details, including the project ID and ARN:

```
{
  "id": "proj_abc123",
  "arn": "arn:aws:bedrock-mantle:us-east-1:YOUR-ACCOUNT-ID:project/proj_abc123"
}
```

Save the project ID — you use it to associate inference requests in the next step.

Step 2: Associate inference requests with the project

Pass the project ID when making inference calls. Amazon Bedrock attributes all costs incurred through the project to your MAP migration.

Using curl (Responses API)

```
$ curl -X POST "$OPENAI_BASE_URL/responses" \
    -H "Authorization: Bearer $OPENAI_API_KEY" \
    -H "Content-Type: application/json" \
    -H "OpenAI-Project: proj_abc123" \
    -d '{
      "model": "openai.gpt-5.6-luna",
      "input": "Summarize the key findings from our migration assessment."
    }'
```

Using curl (Chat Completions API)

```
$ curl -X POST "$OPENAI_BASE_URL/chat/completions" \
    -H "Authorization: Bearer $OPENAI_API_KEY" \
    -H "Content-Type: application/json" \
    -H "OpenAI-Project: proj_abc123" \
    -d '{
```

```
      "model": "openai.gpt-5.6-luna",
      "messages": [{"role": "user", "content": "Summarize the key findings from our
 migration assessment."}]
    }'
```

Using the OpenAI Python SDK

```
from openai import OpenAI

client = OpenAI(
    base_url="https://bedrock-mantle.us-east-1.api.aws/openai/v1",
    project="proj_abc123",  # ID returned when you created the project
)

response = client.responses.create(
    model="openai.gpt-5.6-luna",
    input="Summarize the key findings from our migration assessment."
)
print(response.output_text)
```

To maintain clean cost attribution, always specify a project ID in your API calls rather than relying on the default project.

Database and analytics tags

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

Database and analytics tags

Source	Destination	Tag key	Tag value
On-premises Commercial DB&A	Any AWS DB&A Service	`aws-apn-id`	`comm`*`PRM ID`*
On-premises non-commercial DB&A	Any AWS DB&A Service	`aws-apn-id`	`mig`*`PRM ID`*
On-premises Commercial DB&A	EC2	`aws-apn-id`	`comm`*`PRM ID`*

Source	Destination	Tag key	Tag value
On-premises non-commercial DB&A	EC2	`aws-apn-id`	`mig`*`PRM ID`*
EC2 Commercial DB&A	Any AWS DB&A Service	`aws-apn-id`	`comm_ec2_` *`PRM ID`*
EC2 non-commercial DB&A	Any AWS DB&A Service	`aws-apn-id`	`mig_ec2_`*`PRM ID`*

> **Note**
>
> Use lowercase letters for the `mig`, `mig_ec2_`, `comm`, and `comm_ec2_` prefixes. For more information about your PRM ID, see the section called “PRM 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**.

SAP workload tags

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

SAP workload tags

Source	Destination	Tag key	Tag value
On-premises	AWS	`aws-apn-id`	`sap`*`PRM ID`*

> **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. For more information about your PRM ID, see the section called "PRM 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.

Oracle workload tags

Source	Destination	Tag key	Tag value
On-premises	AWS	`aws-apn-id`	`oracle`*`PRM ID`*

> **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. For more information about your PRM ID, see the section called "PRM 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.

Re-hosting using MGN example

Tag key (automated)	Tag value (automated)
`aws-apn-id`	`mig`*`PRM ID`*

> **Note**
>
> Use lowercase letters for the `mig` prefix . For more information about your PRM ID, see the section called “PRM 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.

DataCenter migration (mix of migration patterns) example

Tag key	Tag value
`aws-apn-id`	`mig`*`PRM ID`*

Note

Use lowercase letters for the `mig` prefix . For more information about your PRM ID, see the section called "PRM ID length".

Example 3: Migrate commercial database from EC2 to RDS

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

Migrate commercial database from EC2 to RDS example

Tag key	Tag value
`aws-apn-id`	`comm_ec2_` *`PRM ID`*

Note

Use lowercase letters for the `comm_ec2_` prefix. For more information about your PRM ID, see the section called "PRM ID length".

Example 4: Database modernization

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

Database modernization example

Tag key	Tag value
`aws-apn-id`	`comm`*`PRM ID`*

Note

Use lowercase letters for the `comm` prefix. For more information about your PRM ID, see the section called “PRM ID length”.

Frequently Asked Questions (FAQs)

This section answers the most common questions about MAP 2.0.

Topics

- How can I create Amazon Bedrock inference profiles to use supported foundation models?
- What are the supported foundation models in Amazon Bedrock?
- Why is tagging required for MAP 2.0?
- Should I activate the Cost Allocation Tag?
- Should migrations occur under the management (payer) account?
- How do I tag my migrated resources in AWS with aws-apn-id tag?
- What is the process for baseline AWS services that cannot be tagged but might also be included in MAP (which can include Amazon Connect or AWS Managed Services)?
- Will I receive MAP incentives if I tag an existing AWS resource?
- Can I use an Amazon S3 bucket that existed before the MAP agreement as destination for my migration?
- What services are in scope for MAP 2.0?
- Are tags case sensitive?
- Can I use Amazon EC2 Dedicated Hosts as part of MAP 2.0?
- What should I do if I don't see the aws-apn-id tag in the Cost allocation tags screen?

- the section called “How can I create Amazon Bedrock inference profiles to use supported foundation models?”
- the section called “What are the supported foundation models in Amazon Bedrock?”
- the section called “Why is tagging required for MAP 2.0?”
- the section called “Should I activate the Cost Allocation Tag?”
- the section called “Should migrations occur under the management (payer) account?”
- the section called “How do I tag my migrated resources in AWS with `aws-apn-id` tag?”
- the section called “What is the process for baseline AWS services that cannot be tagged but might also be included in MAP (which can include Amazon Connect or AWS Managed Services)?”
- the section called “Will I receive MAP incentives if I tag an existing AWS resource?”

- the section called “Can I use an Amazon S3 bucket that existed before the MAP agreement as destination for my migration?”
- the section called “What services are in scope for MAP 2.0?”
- the section called “Are tags case sensitive?”
- the section called “Can I use Amazon EC2 Dedicated Hosts as part of MAP 2.0?”
- the section called “What should I do if I don't see the `aws-apn-id` tag in the Cost allocation tags screen?”

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 `aws-apn-id` 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 `aws-apn-id` 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 `aws-apn-id` 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 `aws-apn-id` 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.

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 `aws-apn-id` as defined in this guide.

Important

You must use the tag key `aws-apn-id` 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 `aws-apn-id` 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.

How do I tag my migrated resources in AWS with `aws-apn-id` 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 `aws-apn-id` 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 `aws-apn-id` 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.

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

The process is automated for Amazon Connect 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 `aws-apn-id` tag in the Cost allocation tags screen?

The Migration Acceleration Program requires that you tag resources with the `aws-apn-id` 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 aws-apn-id cost-allocation tag”.

> **Note**
>
> The Migration Acceleration Program requires that you tag resources with the `aws-apn-id` 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 aws-apn-id cost-allocation tag

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

Verify that the `aws-apn-id` 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.

   ```
   aws-apn-id
   ```

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

Manually set up the `aws-apn-id` 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.

   ```
   aws-apn-id
   ```

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
Initial release	Initial release of the AWS Migration Acceleration Program 2.0 User Guide for MAP for AWS Transform	November 17, 2026