



# SAS<sup>®</sup> Model Manager 15.3: Administrator's Guide

---

## Overview of SAS Model Manager Administration

This guide provides post-installation configuration tasks for SAS Model Manager 15.3 on SAS Viya, and explains both how to prepare SAS Model Manager for use and how to manage information that is associated with SAS Model Manager.

---

**Note:** For general post-installation tasks for SAS Viya, see the deployment guide for your environment. For more information, see [“Post-installation Tasks” in SAS Viya for Linux: Deployment Guide](#).

---

Here are the tasks that are included:

- [Manage permissions](#)
- [Manage content](#)
- [Promote content](#)
- [Configure data libraries](#)
- [Configure publishing destinations](#)
- [Configure access to analytic store model files](#)
- [Configure support for Python code files](#)

For information about deploying SAS Model Manager 15.3 on the SAS Viya platform, see the **Deployment and Administration** section of the [SAS Viya documentation page](#).

---

# Managing Permissions

---

## How to Manage Permissions

SAS Environment Manager and the sas-admin command-line interface (CLI) can be used to manage identities and authorization for SAS Model Manager. Information is available in the SAS Viya administration documentation:

- [“Authorization in SAS Viya” in SAS Viya Administration: Orientation to Authorization](#)
- [“Identity Management: Overview” in SAS Viya Administration: Identity Management](#)

The default permissions for SAS Model Manager are described in [“Default Permissions” on page 2](#). To modify the default permissions, you can do the following:

- Modify permissions for specific folders or objects. For more information, see [“General Authorization: How To \(Authorization Window\)” in SAS Viya Administration: General Authorization](#) and [“General Authorization: How To \(CLI\)” in SAS Viya Administration: General Authorization](#).
- Modify the existing groups or create new ones. For more information, see the following topics:
  - [“Manage Custom Groups” in SAS Viya Administration: Identity Management](#)
  - [“Identity Management: How To \(CLI\)” in SAS Viya Administration: Identity Management](#)
  - [Granting Access to Test Results](#)
- Modify the existing rules or create new rules. For more information, see [“General Authorization: How To \(Rules Page\)” in SAS Viya Administration: General Authorization](#).
- Modify file system directory permissions for caslibs. For more information, see [“File System Directory Permissions” in SAS Viya Administration: Models](#).

---

**Note:** If the **Rules** page or the Authorization window displays a warning icon next to a principal's name, that principal does not exist in the identities service. For more information, see [“Unrecognized Principals” in SAS Viya Administration: General Authorization](#).

---

---

## Default Permissions

---

**Note:** Only SAS Administrators and other authorized users can create, update, or delete repository folders. In addition, authenticated users cannot initially access new custom repositories. A SAS Administrator must grant access for a user or group to a new custom repository. Authorization for existing repositories is not modified during an upgrade. For more information, see [“Access to Models” in SAS Viya Administration: Models](#).

---

By default, all authenticated users have permission to do the following in the default repository and standard repositories:

- Read or view a list of all models and projects.
- Create a standard repository folder when registering models from Model Studio or SAS Visual Analytics.
- Create a model or project.
- Copy a model from another project or a folder.
- Move a model from a folder to another folder or project version.
- Update and delete any model or project.
- Publish any model, including a project champion and challenger models.
- Create a test definition for any model that they have access to read.
- Create a performance definition, run performance, and view performance results and history.

By default, only the user that created a test definition can do the following:

- View, update, or delete the test definition.
- Run the test and view the test results.

You can grant access to test definitions and test results to users other than the user that created the definition either by adding the users to the `SASScoreUsers` group or by granting the users access to specific test results. For more information, see [“Granting Access to Test Results” on page 3](#).

---

**Note:** Specific permissions are required for the following tasks:

- To start a workflow, you must be in the Application Administrators group and have permissions to access the workflow definition. For more information, see [SAS Workflow Manager: Administrator’s Guide](#).
  - To move a model, you must have the appropriate permissions for the source folder, object, and target folder. For more information, see [“Move an Object or Folder” in SAS Viya Administration: Folders](#).
- 

## Granting Access to Test Results

By default, only the user who creates a test can view, update, or delete the test definition or run the test. Only a user who runs a test can view the test results. Other users do not have permission to access the test definition or test results unless the user is a member of the `SASScoreUsers` group.

SAS Model Manager configures the `SASScoreUsers` group automatically. Members of this group have full access to test definitions and results. These permissions enable access through the user interface, the Score Definitions service, and the Score Execution service.

Here is an example of using the `sas-admin` command-line interface (CLI) to add a user to the `SASScoreUsers` custom group:

```
sas-admin identities add-member --user-member-id user1 --group-id SASScoreUsers
```

For instructions on adding users to a group, see [“Manage Custom Groups” in SAS Viya Administration: Identity Management](#) and [“Identity Management: How To \(CLI\)” in SAS Viya Administration: Identity Management](#).

---

# Managing Content

Information that you or other users save is stored and organized in folders. A folder is a virtual container rather than a representation of a physical file system. A folder contains members that are URIs for other folders, SAS resources, or resources outside of SAS. The **Content** page in SAS Environment Manager contains objects (such as SAS content and reports) that you save and that are organized into folders.

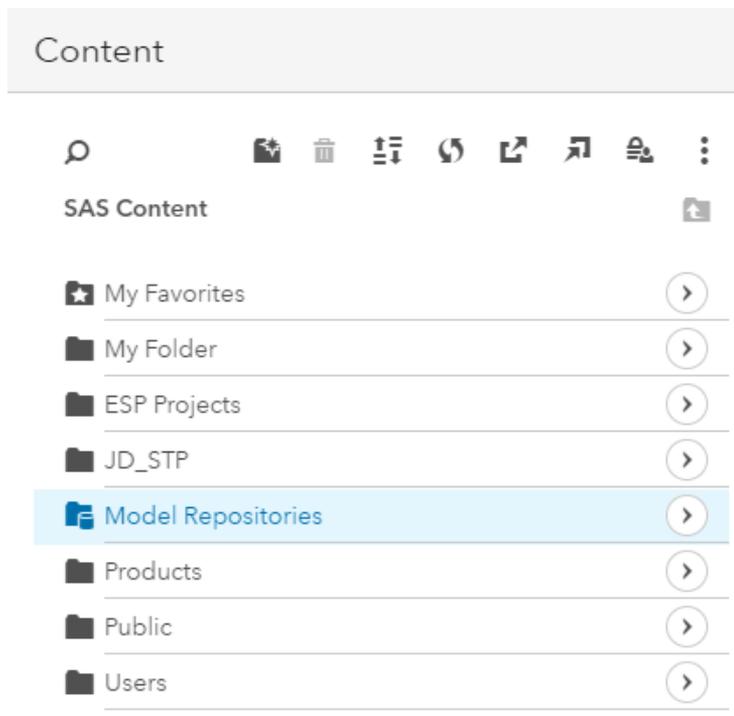
The **Model Repositories** folder is a common model repository for SAS applications. The **Model Repositories** folder can contain one or more repository folders. A repository folder can contain folders, models, and projects. A folder within a repository folder can contain models or projects. A project contains project versions, and a project version can contain one or more models. When a user creates a model or project, both a folder and an object are created. The folder and the object have the same name, and the object appears within the folder.

---

## CAUTION

**Do not rename folders or objects.** The name of the **Model Repositories** folder, as well as repository folders, project folders, project version folders, model folders, and objects within the **Model Repositories** folder should not be modified using the sas-admin CLI or SAS Environment Manager. However, user-defined folders within a repository folder can be renamed.

---



For more information, see the following documentation:

- [“Access to Models” in SAS Viya Administration: Models](#)
- [“Managing Permissions”](#)
- [“Folders: Overview” in SAS Viya Administration: Folders](#)

- “Folders: CLI Examples” in *SAS Viya Administration: Folders*
- “General Authorization: How To (CLI)” in *SAS Viya Administration: General Authorization*

---

# Promoting Content

---

## About Promoting Content

*Promotion* is the process of capturing content and moving it to a different location. For SAS Model Manager 15.3 on SAS Viya, promotion is performed using the transfer plug-in to the admin command-line interface (CLI) or by using SAS Environment Manager. You can promote content from SAS Model Manager 15.1 on SAS Viya 3.3 or SAS Model Manager 15.2 on SAS Viya 3.4 to SAS Model Manager 15.3 on SAS Viya 3.5.

Workflow content cannot be promoted or migrated. Promoting content from SAS Model Manager 14.2 on SAS 9.4 to SAS Model Manager 15.3 on SAS Viya is not currently supported using the SAS Viya 3.5 transfer plug-in.

**IMPORTANT** If you are promoting content from SAS Model Manager 15.1 on SAS Viya 3.3 to SAS Model Manager 15.3 on SAS Viya 3.5, you must use the `sas-admin` command-line interface.

---

**Note:** A model export tool (executable JAR file) is available externally. This tool enables you to export one model at a time from SAS Model Manager 14.2 or 14.3 on SAS 9.4 into a ZIP file format that is supported by SAS Model Manager on SAS Viya. The model can then be imported using the SAS Model Manager 15.3 web application. For more information, see the [SAS Model Manager Downloads](#) page on support.sas.com.

---

For more information, see the following documentation:

- “Promotion within SAS Viya: Tasks” in *SAS Viya Administration: Promotion (Import and Export)*
- “Import SAS Viya Resources” in *SAS Viya Administration: Promotion (Import and Export)*
- “Promotion: Import Using the Command-Line Interface” in *SAS Viya Administration: Promotion (Import and Export)*

---

## Folder and Object Types

The following folder and object types can be exported using the `sas-admin` command-line interface or from the **Content** page of SAS Environment Manager.

Here are a few points to be aware of:

- The **Model Repositories** folder, a repository folder, a folder or subfolder, and a project folder can be transferred. The transfer package contains all of the associated folders and objects within the exported folder.
- When a project object is exported, it does not contain the associated model folders and model objects in the transfer package. The model folder or model object URIs must be included in the transfer package.
- The project version folder does not contain the project object or model objects. Only the folders are transferred. Therefore, it is not recommended to promote content at the project version folder level.

**Table 1** Examples of Folder and Object Types

Folder or Object Type	Location Example	URI Example
<b>Model Repositories</b> folder	/Model Repositories	/folders/folders/ 02c3ede2-0a41-4d56-997b-703aee31b329
repository folder	/Model Repositories/ <i>Custom_Repository</i>	/folders/folders/3b709148-29c0-4be5- a1f1-4109b9b040c6
folder	/Model Repositories/ <i>Custom_Repository/sasdemo</i>	/folders/folders/870af9c9-9074-4ee0-84a6- bc3cd4954cf2
project folder	/Model Repositories/ <i>Custom_Repository/sasdemo/ QS_HMEQ</i>	/folders/folders/ de9c31c8-881f-40c0-86ef-029256c5f0a5
project version folder	/Model Repositories/ <i>Custom_Repository/sasdemo/ QS_HMEQ/Version 1</i>	/folders/folders/aa948017-7fe2-405f- abfc-1c7245f29e9f
project object	/Model Repositories/ <i>Custom_Repository/sasdemo/ QS_HMEQ/QS_HMEQ</i>	/modelRepository/projects/ a0792aac-1f20-4f70-8a0c-72baeb29523c
model folder	/Model Repositories/ <i>Custom_Repository/sasdemo/ QS_HMEQ/Version 1/ QS_Tree1</i>	/folders/folders/3a7b1268- f9a4-4e7b-86f9-64df9acd4f99
model object	/Model Repositories/ <i>Custom_Repository/sasdemo/ QS_HMEQ/Version 1/ QS_Tree1/QS_Tree1</i>	/modelRepository/models/ 79b160f3-5fc4-4b66-9a82-baa98b2b3060

---

## Important Considerations

Here are a few points to be aware of:

- When transferring content using the project object, model folder, or model object, you should import the projects first, followed by the models.
- When you are transferring content using the project object or project version folder, not all of the content is included in the transfer package. Therefore, it is not recommended to transfer content at these levels. You should transfer content from the project folder level instead.
- When project folders or project objects are transferred, the scoring tests, test results, performance definition, and performance results are not included. Scoring tests can be transferred using the sas-admin command-line interface with the score definition URI and the score execution URI. However, it is recommended that you re-create and rerun the test in the target environment rather than transfer the test information.
- When you are transferring analytic store models, you must manually copy the content that is stored on your file system from the source system to the target system.

---

**Note:** The analytic store files are located in the `/models/astores/viya` directory.

In a multi-tenant deployment, the analytic store files are located in the `/models/astores/tenant` directory.

For more information, see [“Configuring Access to Analytic Store Model Files” in SAS Viya Administration: Models](#).

---

- When objects are transferred, the following properties change based on who performed the transfer and when it was done:
  - Created by
  - Modified by
  - Date created
  - Date modified
- Transferring of data from the Model Publish service and Model Management service is not supported.

---

## Transfer Request Body Examples

Folders and objects can be transferred at different levels from the Model Repository service using SAS Environment Manager or the transfer plug-in to the sas-admin command-line interface. Here are some examples of the request body that would appear within the JSON files that are used to transfer folder and object content using the sas-admin command-line interface.

## Model Repositories Folder

If you specify the folder URI for the `/Model Repositories` directory, all of the repository folders, subfolders, authorization rules, projects, models, and model files are transferred as one package. Sample request body:

```
{
  "name": "ModelRepositoriesFolderID",
  "items":
  [
    "/folders/folders/0fec5575-2ee2-4b18-ac3b-5afdd32b4412"
  ]
}
```

## Repository Folder

If you specify the folder URL for a repository folder (for example, `/Model Repositories/myRepository`), all subfolders, authorization rules, projects, models, and model files are transferred as one package. Sample request body:

```
{
  "name": "RepositoryFolderID",
  "items":
  [
    "/folders/folders/b5d59f4c-2346-4de8-bb0d-e3714cdf5594"
  ]
}
```

## Folder

If you specify a folder URI for a folder within a repository folder (for example, `/Model Repositories/myRepository/myFolder`), all subfolders, authorization rules, projects, models, and model files are transferred as one package. Sample request body:

```
{
  "name": "FolderID",
  "items":
  [
    "/folders/folders/9c33b3dd-746e-4ebb-947a-c596f36d96d6"
  ]
}
```

## Project Folder

If you specify the URIs for one or more projects, the specified projects and the associated models are transferred.

```
{
  "name": "Projects",
  "items":
  [
```

```

        "/folders/folders/de9c31c8-881f-40c0-86ef-029256c5f0a5",
        "/folders/folders/18f2c85c-68e3-418a-904e-a405aff8eb50"
    ]
}

```

## Project Object

If you specify the URIs for one or more projects, the specified projects are transferred.

```

{
  "name": "Projects",
  "items":
  [
    "/modelRepository/projects/4f29e89c-bc93-42f5-8491-f338025d75e3",
    "/modelRepository/projects/15103b67-586c-4a3a-8ffd-b840c0921734"
  ]
}

```

## Model Object

If you specify the URIs for one or more models, the specified models and their content are transferred.

```

{
  "name": "Models",
  "items":
  [
    "/modelRepository/models/c8f5694d-9717-45de-98cc-14d08edf7e10",
    "/modelRepository/models/85c66c5e-58ef-4ae6-9df3-3d709daf980f",
    "/modelRepository/models/d62ea249-5bea-4fc1-911d-966a387f4758"
  ]
}

```

---

## sas-admin Command Line Interface Example

- 1 Create and initialize profiles for the source and target systems.

```

sas-admin --profile mySource profile init
sas-admin --profile myTarget profile init

```

- 2 Log in to the target and source systems.

```

sas-admin --profile mySource auth login
sas-admin --profile myTarget auth login

```

- 3 Export the source content to the transfer package.

```

/opt/sas/viya/home/bin/sas-admin --profile mySource transfer export
--request @request_myrepository.json

```

## 10

- 4 Download the transfer package from the source system. Replace the value of the *id* below with the *id* taken from the command output above in the console.

```
/opt/sas/viya/home/bin/sas-admin --profile mySource transfer download  
--id 9b613274-ccf2-48b6-9567-85f46adb1430 --file myPackage_repository.json
```

- 5 Upload the transfer package to the target system.

```
/opt/sas/viya/home/bin/sas-admin --profile myTarget transfer upload  
--file myPackage_repository.json
```

- 6 Replace the value of the *id* below with the *id* taken from the command output above in the console.

```
/opt/sas/viya/home/bin/sas-admin --profile myTarget transfer import  
--request "{\"packageUri\":\"/transfer/packages/6b62ddfc-0a00-4b30-987e-1bb7c428e9a1\"}"
```

---

## Transfer Content That Was Published to a CAS Destination

You can transfer the published module for a model by using the transfer plug-in to the sas-admin CLI. Here is the format of the URI to use in the CLI command when specifying the module that you want to transfer:

```
/modelPublish/destinations/destination_name/models/published_name
```

---

**Note:** Only the content that was included inline in the generated module is transferred.

---

To transfer a published module between SAS Viya environments:

- 1 Export the published module. For example, to export a module named QS\_Reg1 that was published to the destination CAS\_Public, specify the following CLI command:

```
sas-admin --profile source-profile transfer export  
--resource-uri "/modelPublish/destinations/CAS_Public/models/QS_Reg1"
```

If the `export` command is successful, the CLI returns the package ID:

The package with the ID `8f32140f-34e1-42dd-8d85-f8129ff5dabb` was created.

**TIP** You can export multiple items with one transfer request by specifying the items in a JSON file and specifying the JSON file as input to the CLI command. The JSON file contains two fields named `name` and `items`. For example, to export the modules for three different versions of the `QS_Reg1` object, the JSON file might contain the following data:

```
{
  "name": "Published Models",
  "items": [
    "/modelPublish/destinations/CAS_Public/models/QS_Reg1_0",
    "/modelPublish/destinations/CAS_Public/models/QS_Reg1_1",
    "/modelPublish/destinations/CAS_Public/models/QS_Reg1_2"
  ]
}
```

Specify the JSON file name with the `--request` option in the CLI command. If this JSON code is in a file named `request.json`, specify the following command to export all of the specified modules:

```
sas-admin --profile source-profile transfer export --request @request.json
```

- 2 Download the JSON package file. Specify the package ID and the JSON file name in the download command:

```
sas-admin --profile source-profile transfer download --id "8f32140f-34e1-42dd-8d85-f8129ff5dabb"
-f QS_Reg1_0.json
```

- 3 Verify that the target server has a CAS publishing destination with the same name as the source server. For the package to be imported successfully, the target server must have a CAS publishing destination with the same name as the source server. For more information, see [SAS Viya Administration: Publishing Destinations](#).

- 4 Use the `upload` command to upload the JSON file to the destination server:

```
sas-admin --profile target-profile transfer upload -f QS_Reg1_0.json
```

If the `upload` command is successful, the CLI prints a message that includes the ID of the transfer package. Here is an example:

```
{
  "description": "",
  "id": "ef4204f7-8a79-49ae-8d45-a7877f5e1e86",
  "links": [
    ...
  ],
  "name": "Export",
  "type": "TransferPackage",
  "version": 2
}
```

- 5 Import the contents of the transfer package. Specify the ID of the transfer package that was returned in the previous step:

```
sas-admin --profile target-profile transfer import --id "ef4204f7-8a79-49ae-8d45-a7877f5e1e86"
```

- 6 Verify that the import process completed successfully. Use the `show` command to display the export and import history for the package. To display the export history:

```
sas-admin --profile source-profile transfer show
--id "8f32140f-34e1-42dd-8d85-f8129ff5dabb" --history
```

To display the import history:

```
sas-admin --profile target-profile transfer show  
  --id "ef4204f7-8a79-49ae-8d45-a7877f5e1e86" --history
```

For more information about using the transfer plug-in, see [“Promotion: How To \(CLI\)” in SAS Viya Administration: Promotion \(Import and Export\)](#).

---

## Configuring Data Libraries

---

### About Configuring Data Libraries

During the deployment of SAS Model Manager, the ModelPerformanceData and ModelStore caslibs are created on each CAS server (for example, cas-shared-default and cas-shared-mpp). The source type for the caslibs is a file system path. Users must have Read and Write permissions to the source file system directory paths. You can also create your own data libraries and then import tables or CSV formatted data.

For more information, see [“Configuring Model Data Libraries” in SAS Viya Administration: Models](#).

---

### File System Directory Permissions

When defining a caslib where the source type is a file system directory path, the appropriate permissions must be granted. By default, CAS sessions run using the **cas** account. The CASHostAccountRequired custom group is a SAS Viya reserved group name, but it is not created during deployment of SAS Viya. If you add identities or groups to the custom group with the ID CASHostAccountRequired, members of this group automatically run their CAS sessions under their own host account. Users within this group must have Read and Write permissions to caslib file system directory paths in order to register analytic store models and generate performance results. In addition, users’ ability to share analytic stores depends on their primary group permissions. An analytic store is created with group ownership by each user’s primary group. For more information, see [“File System Directory Permissions” in SAS Viya Administration: Models](#).

---

## Configuring Publishing Destinations

---

### About Publishing Destinations

You can publish models to publishing destinations on SAS Cloud Analytic Services (CAS), Apache Hadoop, SAS Micro Analytic Service, and Teradata, as well as container destinations such as

Amazon Web Services and Private Docker. By default, a SAS Micro Analytic Service destination named **maslocal** is defined for you. You must configure all other publishing destinations.

Here are the prerequisites for configuring other publishing destinations:

- Configure the Docker daemon for container destinations. For more information, see [“Configuring the Docker Daemon” on page 13](#).
- Create a caslib for a CAS publishing destination using the sas-admin command-line interface (CLI) or SAS Environment Manager. For more information, see [“Create a Global Caslib” in SAS Viya Administration: Publishing Destinations](#).

Here is an example of creating a caslib using the sas-admin CLI:

```
sas-admin cas caslibs create path --name myCaslibA
--path /opt/sas/viya/config/data/cas/default/myCaslibA
--server cas-shared-default
```

For more information, see [“Manage Caslibs” in SAS Viya Administration: Using the Command-Line Interfaces](#).

You can use SAS Environment Manager to define and manage publishing destinations for SAS Cloud Analytic Services (CAS), Apache Hadoop, SAS Micro Analytic Service, and Teradata. For more information, see [“Using SAS Environment Manager to Configure Publishing Destinations” on page 14](#).

In order to configure publishing destinations for Amazon Web Services and Private Docker, you must submit a REST API request to the Model Publish API. For more information, see [“Using the REST API to Define Publishing Destinations” on page 14](#).

## Configuring the Docker Daemon

### About Configuring the Docker Daemon

To enable publishing of Python and R models to container destinations and to validate the models within the container destinations, you must first configure the Docker daemon. Here are the requirements:

- If the Docker daemon is installed on the system where the Model Publish API service resides and the Docker host is not configured, the `sas` user must be configured to use the Docker daemon socket. See [“Add sas User to Docker Group” on page 13](#).
- If the Docker daemon is installed on a different system than the one on which the Model Publish API service resides, then the Docker host must be configured to go through the TCP port.

For more information, see [Configure and troubleshoot the Docker daemon](#).

### Add sas User to Docker Group

- 1 Use the following command to add the `sas` user to the `docker` group:

```
sudo usermod -aG docker sas
```

For more information, see [Manage Docker as a non-root user](#).

- 2 Use the following command to restart the Model Publish API service:

```
sudo service sas-viya-modelpublish-default restart
```

For more information, see “General Servers and Services: Operate (Linux)” in *SAS Viya Administration: General Servers and Services*.

---

## Using SAS Environment Manager to Configure Publishing Destinations

You can use SAS Environment Manager to define and manage publishing destinations for SAS Cloud Analytic Services (CAS), Apache Hadoop, SAS Micro Analytic Service, and Teradata. For more information, see *SAS Viya Administration: Publishing Destinations*.

You can use the SAS Model Manager tutorial to verify that your publishing destinations have been configured properly. For more information, see *SAS Model Manager: Quick Start Tutorial*.

---

## Creating Container Base Images for Scoring Models

After you create a container publishing destination for AWS or Private Docker, you must also create a container base image. Base image creation is currently supported for Python 3 and R models. The base images are used for scoring and validating Python models and R models at run time that have been published from SAS Model Manager to a container destination.

---

**Note:** If you want to publish both Python and R models to the same container publishing destination, you must create two base images, one for R models and one for Python models.

---

You must specify the following information when creating base images:

- publishing destination name
- host server name and port
- SAS user account that is authorized to connect to the host server

Sample Python scripts and additional information about creating the container base images are available in the `addons` directory of the [Open Model Manager Resources GitHub repository](#). The Python scripts are used to submit REST API requests to the Model Publish API when you are creating the base images.

For more information about creating base images, see the Model Containerization section of the [README.md](#) file in the `addons` directory.

---

## Using the REST API to Define Publishing Destinations

You can define publishing destinations by submitting REST API requests to the Model Publish API.

---

**Note:** When defining a publishing destination for Amazon Web Services (AWS), you must create a credential domain in the SAS Credentials service and store the AWS access key information in the credentials. You must also specify the identity (user or group) that can use the credentials to score or validate a model within the container publishing destination.

---

Python code examples for defining CAS, AWS, and Private Docker publishing destinations, as well as the mmAuthorization module, are available at <https://github.com/sassoftware/open-model-manager-resources/tree/master/addons>.

For more information, see the [Model Publish API](#) documentation and the [Model Publish API examples on GitHub](#).

---

## Configuring Access to Analytic Store Model Files

In order to publish analytic store models to a SAS Micro Analytic Service destination, you must configure access to the location where the analytic store (ASTORE) files are located. Also, users who work with analytic store models must have Read and Write access to analytic store directories. The ASTORE file system directory path is `/models/astores/viya`.

For more information, see “[Configuring Access to Analytic Store Model Files](#)” in *SAS Viya Administration: Models*.

---

## Configuring Support for Python Code Files

Here are the requirements for configuring support for Python code files:

- To support models that contain custom Python code files, you must enable PyMAS package support.

For more information, see “[Enabling PyMAS Package Support](#)” in *SAS Micro Analytic Service: Programming and Administration Guide*.

- Users who are developing and testing Python code files must be added to the CASHostAccountRequired custom group. For more information, see “[The CASHostAccountRequired Custom Group](#)” in *SAS Viya Administration: Identity Management* and “[Add or Remove Custom Group Members](#)” in *SAS Viya Administration: Identity Management*.
- Python 3.6.9 or later must be installed with the following packages:
  - joblib
  - numpy
  - pandas
  - patsy

- python-dateutil
- pytz
- scikit-learn
- scipy
- six
- statsmodels

For more information, see [Python Package Index \(PyPI\)](#).

- The resources directory must be granted Read and Write permissions. Here is an example of the command that you can use:

```
sudo chmod 777 /opt/sas/viya/config/data/modelsvr/resources/
```