# Best Practices

**Issue** 01  
**Date** 2022-01-28
ModelArts Samples

This document provides ModelArts samples concerning a variety of scenarios and AI engines to help you quickly understand the process and operations of using ModelArts for AI development.

ModelArts Permissions (Basic)

<table>
<thead>
<tr>
<th>Sample</th>
<th>Function</th>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Scenarios</td>
<td>IAM permissions and global configurations</td>
<td>Permiss</td>
<td>Assign specific ModelArts operation permissions to the IAM users under a HUAWEI CLOUD account. This prevents exceptions from occurring due to permissions when the IAM users access ModelArts.</td>
</tr>
</tbody>
</table>

ExeML Samples (Basic)

Table 1-1 ExeML sample projects

<table>
<thead>
<tr>
<th>Sample</th>
<th>Function</th>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yunbao Detection</td>
<td>ExeML</td>
<td>Object detection</td>
<td>Based on the Yunbao dataset, use the object detection algorithm of ModelArts ExeML to identify Yunbao in images.</td>
</tr>
<tr>
<td>Bank Deposit Prediction</td>
<td>ExeML</td>
<td>Predictive analytics</td>
<td>Predict whether customers would be interested in a time deposit based on their characteristics, including the age, work type, marital status, education background, housing loan, and personal loan.</td>
</tr>
</tbody>
</table>
## Development Tool Samples (Advanced)

### Table 1-2 Notebook samples

<table>
<thead>
<tr>
<th>Sample</th>
<th>Engine</th>
<th>Function</th>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using a Notebook for Handwritten Digit Recognition</td>
<td>Tensor Flow</td>
<td>● Training scripts compiled</td>
<td>Image recognition</td>
<td>Develop training scripts based on the MoXing framework and achieve the recognition of handwritten digits based on the AI development lifecycle.</td>
</tr>
</tbody>
</table>
2 Assigning ModelArts Permissions

2.1 Application Scenarios

If you already have a HUAWEI CLOUD account and need to assign the account permissions to multiple users so that they can access ModelArts, follow the instructions provided in this section.

Example Scenarios

As shown in the following figure, the administrator account is a HUAWEI CLOUD account.

- **Scenario 1**: All users in group 1 have all ModelArts operation permissions and all OBS operation permissions.
- **Scenario 2**: All users in group 2 have certain ModelArts operation permissions and the minimum permissions required by ModelArts to access OBS.
Figure 2-1 Example scenarios

Configuration Process and Precautions

Figure 2-2 shows the flowchart for assigning permissions. Before performing operations, learn about the following information:

- ModelArts uses IAM for authentication. Assign permissions on the IAM management console.
- ModelArts uses OBS to store data. Therefore, you must be authorized to access OBS before using ModelArts.
- ModelArts functions, such as data management and model management, require access to the OBS, SWR, and IEF services. Authorize the access to these services before using ModelArts.
**Figure 2-2** Process of assigning permissions

1. **Start**
2. Create an administrator account.
3. Create a user group.
4. Assign permissions to access ModelArts.
5. Assign permissions to access OBS.
6. Create a user and add the user to the user group.
7. Configure agency authorization.
8. Verify user permissions.
9. **End**

### Recommended Configurations

You can grant users permissions using roles or policies.

- **Roles**: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, assign other roles on which the permissions depend to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.

- **Policies**: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant OBS users only the permissions for managing a certain type of OBS resources.
To assign permissions based on services, use roles for authorization. To assign permissions based on operations, use policies for authorization.

In the two scenarios shown in Figure 2-1, the configuration processes are the same. The difference between the two scenarios lies in configurations. For details, see Table 2-1.

**Table 2-1 Configurations for different scenarios**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an administrator account.</td>
<td>Configuring an Administrator Account</td>
<td>Configuring an Administrator Account</td>
<td>Configuring an Administrator Account</td>
<td>The operations in the two scenarios are the same.</td>
</tr>
<tr>
<td>Create a user group.</td>
<td>Creating a User Group</td>
<td>Creating a User Group</td>
<td>Creating a User Group</td>
<td>The operations in the two scenarios are the same.</td>
</tr>
<tr>
<td>Assign permissions to access ModelArts.</td>
<td>Assigning ModelArts Operation Permissions to the User Group (Role-based authorization)</td>
<td>Assigning ModelArts Operation Permissions to the User Group (Policy-based authorization to assign ModelArts operation permissions)</td>
<td>Assigning ModelArts Operation Permissions to the User Group (Policy-based authorization to assign certain ModelArts operation permissions)</td>
<td>Configurations vary in different scenarios. Roles or policies can be used for authorization. To assign certain operation permissions, use policies for authorization.</td>
</tr>
</tbody>
</table>
### Process

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign permissions to access OBS.</td>
<td>Assigning OBS Operation Permissions to the User Group</td>
<td>Assigning OBS Operation Permissions to the User Group</td>
<td>Configurations vary in different scenarios.</td>
</tr>
<tr>
<td>(Role-based authorization)</td>
<td>(Policy-based authorization to assign minimum permissions required by ModelArts to access OBS)</td>
<td>(Policy-based authorization to assign minimum permissions required by ModelArts to access OBS)</td>
<td>Roles or policies can be used for authorization. To assign certain operation permissions, use policies for authorization.</td>
</tr>
<tr>
<td>Create a user and add the user to the user group.</td>
<td>Creating a User and Adding the User to the User Group</td>
<td>Creating a User and Adding the User to the User Group</td>
<td>The operations in the two scenarios are the same. The user group to which a user is added varies in different scenarios.</td>
</tr>
<tr>
<td>Configure agency authorization.</td>
<td>Performing Global ModelArts Configurations for All Users</td>
<td>Performing Global ModelArts Configurations for All Users</td>
<td>The operations in the two scenarios are the same.</td>
</tr>
<tr>
<td>Verify user permissions.</td>
<td>Verifying User Permissions</td>
<td>Verifying User Permissions</td>
<td>The operations in the two scenarios are the same.</td>
</tr>
</tbody>
</table>

### 2.2 Configuring an Administrator Account

In the example scenarios, a HUAWEI CLOUD account with real-name authenticated is required. It is used as an administrator account. If you already have such an account, skip this section.
Registering a HUAWEI CLOUD Account with Real-Name Authenticated

To do so, perform the following operations:

1. Register a HUAWEI CLOUD account.
   Log in to the official HUAWEI CLOUD website, click Register in the upper right corner, and set parameters as prompted to complete the registration.

2. Perform real-name authentication.
   After the registration is complete, you are automatically directed to HUAWEI CLOUD. You can use cloud services only after completing real-name authentication. For details, see Real-Name Authentication.

Globally Configuring Agency Authorization

Perform the operations described in this section to configure access authorization through an agency. For more information, see Preparations > Configuring Access Authorization > Configuring Agency Authorization.

1. Use the administrator account to log in to the ModelArts management console.
   Select your region in the upper left corner.

2. In the navigation pane on the left, choose Settings. The Global Configuration page is displayed.

3. Click Add Authorization. On the Add Authorization page, set parameters listed in Table 2-2, select I have read and agree to the ModelArts Service Statement, and click Create.

Table 2-2 Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>By default, All users is selected, which indicates that all users in the drop-down list, including the current user, will be authorized. You can also select a username from the drop-down list.</td>
</tr>
</tbody>
</table>
| Agency          | ● Use existing: If there are existing agencies in the list, select an available one to authorize the selected user. Click the drop-down arrow next to an agency name to view its permission details.  
                 ● Add agency: If there is no available agency, create one. If you use ModelArts for the first time, select Add agency. |
<p>| Add agency &gt; Agency Name | The system automatically creates an agency name that is editable. |
| Add agency &gt; Permissions &gt; Common User | You can use basic ModelArts functions, for example, accessing data and creating and managing training jobs, but not to manage resources. Select this option generally. Click View permissions to view common user permissions. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add agency &gt; Permissions &gt; Custom</td>
<td>You can flexibly assign permissions to the created agency. Select this mode for refined permissions management.</td>
</tr>
</tbody>
</table>

**Figure 2-3 Agency authorization**

2.3 Assigning All Permissions Using Roles for Authorization

This section describes how to assign all ModelArts operation permissions to user group 1. In this example scenario, roles are used for authorization.

**Creating a User Group**

1. On the management console, click your username in the upper right corner and select **Identity and Access Management** from the drop-down list to switch to the IAM management console.
2. In the navigation pane on the left, choose **User Groups**.
3. Click **Create User Group** in the upper right corner. Then, set **Name** to **UserGroup-1** and click **OK**.
Assigning ModelArts Operation Permissions to the User Group

Perform the following operations to assign all ModelArts operation permissions to user group 1. Users will inherit permissions from the group to which they are added.

1. On the IAM management console, select User Groups.
2. Locate the row containing user group 1, click Authorize in the Operation column. The Authorize User Group page is displayed.
   - Select Policy/Role: Enter ModelArts in the search box and select ModelArts FullAccess and ModelArts CommonOperations policies.
   - Select Scope: Set Scope to Region-specific projects and select the target region that is the same as the region where ModelArts is deployed, for example, CN-Hong Kong.
Figure 2-6 Selecting a scope

4. Click OK.

Assigning OBS Operation Permissions to the User Group

ModelArts uses OBS to store data, code, and models. Therefore, before using ModelArts, obtain permissions to access OBS. Perform the following operations to assign all OBS operation permissions to user group 1. For details about how to assign minimum permissions required by ModelArts to access OBS, see Assigning OBS Operation Permissions to the User Group.

1. After you perform the operations described in Assigning ModelArts Operation Permissions to the User Group, the system automatically switches back to the user group list. Click Authorize for the target user group.

2. On the Authorize User Group page, set parameters.
   - Select Policy/Role: Enter Tenant Administrator in the search box and select it.

   Figure 2-7 Selecting policies

   - Select Scope: Set Scope to Global services.
NOTE

After you assign the Tenant Administrator permission to a user group, all users in this group will inherit all operation permissions (except the IAM management permission) of all global services. Exercise caution when performing this operation.

3. Click OK. It takes about 15 to 30 minutes to complete the permission assignment.

Creating a User and Adding the User to the User Group

1. In the navigation pane on the left, select Users.
2. Click Create User in the upper right corner of the page.
3. On the displayed page, add multiple users in the User Details pane. In this example, add users A, B, and C.
   Set mandatory parameters as prompted and click Next.

4. On the Add User to Groups page, select UserGroup-1 and click Create.

The system automatically adds the three users to the target group one by one.
Performing Global ModelArts Configurations for All Users

ModelArts functions, such as data management and model management, require access to the OBS, SWR, and IEF services. Authorize the access before using ModelArts.

Agency authorization is recommended. In this mode, only the administrator is allowed to assign permissions for IAM users. Therefore, in this example, the administrator needs to configure access authorization for all the three users. For more information, see Preparations > Configuring Access Authorization > Configuring Agency Authorization.

1. Use the administrator account to log in to the ModelArts management console.

   Select your region in the upper left corner, for example, CN-Hong Kong.

2. In the navigation pane on the left, choose Settings. The Global Configuration page is displayed.

3. Click Add Authorization. On the Add Authorization page, set parameters listed in Table 2-3, select I have read and agree to the ModelArts Service Statement, and click Create.

Table 2-3 Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>By default, All users is selected, which indicates that all users in the drop-down list, including the current user, will be authorized. You can also select a username from the drop-down list.</td>
</tr>
</tbody>
</table>
| Agency                        | • Use existing: If there are existing agencies in the list, select an available one to authorize the selected user. Click the drop-down arrow next to an agency name to view its permission details.  
                                • Add agency: If there is no available agency, create one. If you use ModelArts for the first time, select Add agency. |
| Add agency > Agency Name      | The system automatically creates an agency name that is editable.             |
| Add agency > Permissions > Common User | You can use basic ModelArts functions, for example, accessing data and creating and managing training jobs, but not to manage resources. Select this option generally. Click View permissions to view common user permissions. |
| Add agency > Permissions > Custom | You can flexibly assign permissions to the created agency. Select this mode for refined permissions management. |
Verifying User Permissions

It takes 15 to 30 minutes for the OBS permissions to take effect. Therefore, wait for 30 minutes after the configuration and then verify the configuration.

1. Log in to the ModelArts management console as user A, B, or C. On the login page, ensure that IAM User Login is selected.
   Change the password as prompted upon the first login.
2. Check ModelArts permissions.
   a. Select the target region in the upper left corner, which must be the same as that in the authorization configuration. In this example, CN-Hong Kong is used.
   b. In the navigation pane on the left of the ModelArts management console, choose Data Management > Datasets and click Create. If this operation is successful, you have obtained ModelArts operation permissions.
      Alternatively, you can try other functions, such as Training Management > Training Jobs or DevEnviron > Notebook. If the operation is successful, you can use ModelArts properly.
3. Verify OBS permissions.
   a. In the service list in the upper left corner, select OBS. The OBS management console is displayed.
   b. Click Create Bucket in the upper right corner. If this operation is successful, you have obtained OBS operation permissions.

The verification shows that you have operation permissions on both ModelArts and OBS. Then, you can use ModelArts properly.
2.4 Assigning All Permissions Using Policies for Authorization

This section describes how to assign all ModelArts operation permissions to user group 1. In this example scenario, policies are used for authorization.

Creating a User Group

1. On the management console, click your username in the upper right corner and select Identity and Access Management from the drop-down list to switch to the IAM management console.
2. In the navigation pane on the left, choose User Groups.
3. Click Create User Group in the upper right corner. Then, set Name to UserGroup-1 and click OK.

Figure 2-12 Creating a user group

After the user group is created, the system automatically switches to the user group list.

Assigning ModelArts Operation Permissions to the User Group

Perform the following operations to assign all ModelArts operation permissions to user group 1. Users will inherit permissions from the group to which they are added.

To do so, create a custom policy with specific ModelArts operation permissions and associate the policy with the user group.

1. Create a custom policy. For more information, see Creating a Custom Policy.
   a. On the IAM management console, choose Permissions > Policies/Roles.
   b. On the Policies/Roles page, click Create Custom Policy in the upper right corner.
c. Set policy parameters.
   **Policy Name**: Customize a policy name.
   **Policy View**: Select JSON.
   **Policy Content**: Configure the policy by referring to the following examples. Two policy examples are provided. Select one based on your service needs.

**Figure 2-13** Configuring a custom policy for ModelArts

Configure a policy that assigns all ModelArts operation permissions.

```json
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": ["modelarts:*:*"],
            "Effect": "Allow"
        }
    ]
}
```

Configure a policy that assigns common user operation permissions. A common user has all operation permissions except adding, deleting, or modifying a dedicated resource pool.

```json
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": ["modelarts:*:*"
        }
    ]
}
```
d. After configuring the policy, click OK.

2. Associate the custom policy with the target user group.
   a. On the IAM management console, select User Groups.
   b. Locate the row containing user group 1, click Authorize in the Operation column. The Authorize User Group page is displayed.
   c. On the Authorize User Group page, set parameters.
      - **Select Policy/Role**: Enter the policy keyword in the search box and select the custom policy created in 1.
      - **Select Scope**: Set Scope to Region-specific projects and select the target region that is the same as the region where ModelArts is deployed, for example, CN-Hong Kong.
   d. Click OK.

Assigning OBS Operation Permissions to the User Group

ModelArts uses OBS to store data, code, and models. Therefore, before using ModelArts, obtain permissions to access OBS.
In this example, policies are used to assign OBS permissions to users. To do so, create a custom policy with minimum operation permissions required by ModelArts to access OBS and associate the policy with the target user group.

1. Create a custom policy. For more information, see OBS Custom Policies.
   a. On the IAM management console, choose Permissions > Policies/Roles.
   b. On the Policies/Roles page, click Create Custom Policy in the upper right corner.
   c. Set policy parameters.
      Policy Name: Customize a policy name.
      Policy View: Select JSON.
      Policy Content: Configure the policy by referring to the following example. In this example, the minimum permissions required by ModelArts to access OBS are assigned, including the permissions for OBS buckets and objects. After obtaining these permissions, you can access OBS in ModelArts.

Figure 2-16 Configuring a custom policy for OBS

```json
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "obs:object:PutObjectAcl",
        "obs:bucket:PutBucketAcl",
        "obs:bucket:PutBucketPolicy",
        "obs:bucket:DeleteObjectVersion",
        "obs:object:AbortMultiUploads",
        "obs:object:DeleteObjectVersion",
        "obs:object:PutObject",
        "obs:object:CreateBucket",
        "obs:object:GetObject",
        "obs:bucket:GetBucketLocation",
        "obs:object:GetObjectVersionAcl",
        "obs:bucket:GetBucketAcl",
        "obs:object:ListMultipleUploadParts",
        "obs:bucket:ListBucketVersions",
        "obs:object:GetObjectAcl"
      ],
      "Resource": "*"
    }
  ]
}
```
a. After configuring the policy, click **OK**.

2. Associate the custom policy with the target user group.
   a. On the IAM management console, select **User Groups**.
   b. Locate the row containing the target user group, click **Authorize** in the **Operation** column. The **Authorize User Group** page is displayed.
   c. On the **Authorize User Group** page, set parameters.
      - **Select Policy/Role**: Enter the policy keyword in the search box and select the custom policy created in 1.
      - **Select Scope**: Set **Scope** to **Global services**.
   d. Click **OK**.
Creating a User and Adding the User to the User Group

1. In the navigation pane on the left, select **Users**.
2. Click **Create User** in the upper right corner of the page.
3. On the displayed page, add multiple users in the **User Details** pane. In this example, add users A, B, and C.
   Set mandatory parameters as prompted and click **Next**.

   **Figure 2-19** Adding multiple users

![Adding multiple users](image)

4. On the **Add User to Groups** page, select **UserGroup-1** and click **Create**.

   **Figure 2-20** Adding users to the target user group

![Adding users to the target user group](image)

The system automatically adds the three users to the target group one by one.

Performing Global ModelArts Configurations for All Users

ModelArts functions, such as data management and model management, require access to the OBS, SWR, and IEF services. Authorize the access before using ModelArts.

Agency authorization is recommended. In this mode, only the administrator is allowed to assign permissions for IAM users. Therefore, in this example, the administrator needs to configure access authorization for all the three users. For more information, see **Preparations > Configuring Access Authorization > Configuring Agency Authorization**.

1. Use the administrator account to log in to the ModelArts management console.
   Select your region in the upper left corner, for example, **CN-Hong Kong**.
2. In the navigation pane on the left, choose **Settings**. The **Global Configuration** page is displayed.
3. Click **Add Authorization**. On the **Add Authorization** page, set parameters listed in **Table 2-4**, select I have read and agree to the ModelArts Service Statement, and click **Create**.
### Table 2-4 Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>By default, All users is selected, which indicates that all users in the drop-down list, including the current user, will be authorized. You can also select a username from the drop-down list.</td>
</tr>
<tr>
<td>Agency</td>
<td><strong>Use existing</strong>: If there are existing agencies in the list, select an available one to authorize the selected user. Click the drop-down arrow next to an agency name to view its permission details.</td>
</tr>
<tr>
<td></td>
<td><strong>Add agency</strong>: If there is no available agency, create one. If you use ModelArts for the first time, select Add agency.</td>
</tr>
<tr>
<td>Add agency &gt; Agency Name</td>
<td>The system automatically creates an agency name that is editable.</td>
</tr>
<tr>
<td>Add agency &gt; Permissions &gt; Common User</td>
<td>You can use basic ModelArts functions, for example, accessing data and creating and managing training jobs, but not to manage resources. Select this option generally. Click View permissions to view common user permissions.</td>
</tr>
<tr>
<td>Add agency &gt; Permissions &gt; Custom</td>
<td>You can flexibly assign permissions to the created agency. Select this mode for refined permissions management.</td>
</tr>
</tbody>
</table>

### Figure 2-21 Agency authorization

![Agency authorization](image)

### Verifying User Permissions

It takes 15 to 30 minutes for the OBS permissions to take effect. Therefore, wait for 30 minutes after the configuration and then verify the configuration.
1. Log in to the ModelArts management console as user A, B, or C. On the login page, ensure that IAM User Login is selected. Change the password as prompted upon the first login.

2. Check ModelArts permissions.
   a. Select the target region in the upper left corner, which must be the same as that in the authorization configuration. In this example, CN-Hong Kong is used.
   b. In the navigation pane on the left of the ModelArts management console, choose Data Management > Datasets and click Create. If this operation is successful, you have obtained ModelArts operation permissions.
      Alternatively, you can try other functions, such as Training Management > Training Jobs or DevEnviron > Notebook. If the operation is successful, you can use ModelArts properly.

3. Verify OBS permissions.
   a. In the service list in the upper left corner, select OBS. The OBS management console is displayed.
   b. Click Create Bucket in the upper right corner. If this operation is successful, you have obtained OBS operation permissions.

The verification shows that you have operation permissions on both ModelArts and OBS. Then, you can use ModelArts properly.

### 2.5 Assigning Certain Permissions Using Policies for Authorization

This section describes how to use a custom policy for authorization to assign certain ModelArts operation permissions to user group 2.

**Creating a User Group**

1. On the management console, click your username in the upper right corner and select Identity and Access Management from the drop-down list to switch to the IAM management console.
2. In the navigation pane on the left, select User Groups.
3. Click Create User Group in the upper right corner. Then, set Name to UserGroup-2 and click OK.
After the user group is created, the system automatically switches to the user group list.

**Assigning ModelArts Operation Permissions to the User Group**

Perform the following operations to create a custom policy so that the users in group 2 can obtain ModelArts operation permissions except the permission to use the development environment.

To do so, create a custom policy with specific ModelArts operation permissions and associate the policy with the user group.

1. Create a custom policy. For more information, see [Creating a Custom Policy](#).
   a. On the IAM management console, choose Permissions > Policies/Roles.
   b. On the Policies/Roles page, click Create Custom Policy in the upper right corner.
   c. Set policy parameters.
      - **Policy Name**: Customize a policy name.
      - **Policy View**: Select JSON.
      - **Policy Content**: Configure the policy by referring to the following example.
Figure 2-23 Configuring a custom policy for ModelArts

**Configure** a policy to prohibit the users from using the development environment.

**NOTE**

Newly created users have the ModelArts operation permissions by default. To prohibit such a user from using certain ModelArts functions, deny specific operation permissions for the user.

```json
{  
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "modelarts:notebook:list",
        "modelarts:notebook:create",
        "modelarts:notebook:get",
        "modelarts:notebook:update",
        "modelarts:notebook:delete",
        "modelarts:notebook:action",
        "modelarts:notebook:access"
      ]
    }
  ]
}
```

d. After configuring the policy, click **OK**.

2. Associate the custom policy with the target user group.
a. On the IAM management console, select User Groups.
b. Locate the row containing user group 2, click Authorize in the Operation column. The Authorize User Group page is displayed.
c. On the Authorize User Group page, set parameters.
   ▪ **Select Policy/Role:** Enter the policy keyword in the search box and select the custom policy created in 1.

   ![Figure 2-24 Selecting policies](image)

   ![Figure 2-25 Selecting a scope](image)

   ▪ **Select Scope:** Set Scope to Region-specific projects and select the target region that is the same as the region where ModelArts is deployed, for example, **CN-Hong Kong**.

d. Click **OK**.

**Assigning OBS Operation Permissions to the User Group**

ModelArts uses OBS to store data, code, and models. Therefore, before using ModelArts, obtain permissions to access OBS.

In this example, policies are used to assign OBS permissions to users. To do so, create a custom policy with minimum operation permissions required by ModelArts to access OBS and associate the policy with the target user group.

1. Create a custom policy. For more information, see **OBS Custom Policies**.
   a. On the IAM management console, choose Permissions > Policies/Roles.
   b. On the Policies/Roles page, click Create Custom Policy in the upper right corner.
   c. Set policy parameters.
      **Policy Name:** Customize a policy name.
      **Policy View:** Select JSON.
      **Policy Content:** Configure the policy by referring to the following example. In this example, the minimum permissions required by
ModelArts to access OBS are assigned, including the permissions for OBS buckets and objects. After obtaining these permissions, you can access OBS in ModelArts.

**Figure 2-26 Configuring a custom policy for OBS**

```json

"Version": "1.1",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "obs:object:PutObjectAcl",
            "obs:bucket:PutBucketAcl",
            "obs:bucket:PutBucketPolicy",
            "obs:bucket:HeadBucket",
            "obs:bucket:ListAllMyBuckets",
            "obs:bucket:ListBucket",
            "obs:object:DeleteObjectVersion",
            "obs:object:AbortMultipartUpload",
            "obs:object:DeleteObject",
            "obs:object:PutObject",
            "obs:bucket:CreateBucket",
            "obs:object:GetObject",
            "obs:bucket:GetBucketLocation",
            "obs:object:GetObjectVersionAcl",
            "obs:object:GetObjectAcl",
            "obs:object:GetObjectVersion",
            "obs:object:GetObjectVersionAcl",
            "obs:bucket:GetBucketPolicy"
        ],
        "Resource": [
            "*"
        ]
    }
]
```


a. After configuring the policy, click **OK**.

2. **Associate the custom policy with the target user group.**
   a. On the IAM management console, select **User Groups**.
   b. Locate the row containing the target user group, click **Authorize** in the **Operation** column. The **Authorize User Group** page is displayed.
   c. On the **Authorize User Group** page, set parameters.
   - **Select Policy/Role**: Enter the policy keyword in the search box and select the custom policy created in **1**.
   - **Select Scope**: Set Scope to **Global services**.

![Selecting policies](image)

*Figure 2-27 Selecting policies*

![Selecting a scope](image)

*Figure 2-28 Selecting a scope*

   d. Click **OK**.

**Creating a User and Adding the User to the User Group**

1. In the navigation pane on the left, select **Users**.
2. Click **Create User** in the upper right corner of the page.
3. On the displayed page, add multiple users in the **User Details** pane. In this example, add users D, E, and F.
   Set mandatory parameters as prompted and click **Next**.
4. On the **Add User to Groups** page, select **UserGroup-2** and click **Create**.

**Figure 2-30 Adding users to the target user group**

The system automatically adds the three users to the target group one by one.

**Performing Global ModelArts Configurations for All Users**

ModelArts functions, such as data management and model management, require access to the OBS, SWR, and IEF services. Authorize the access before using ModelArts.

Agency authorization is recommended. In this mode, only the administrator is allowed to assign permissions for IAM users. Therefore, in this example, the administrator needs to configure access authorization for all the three users. For more information, see **Preparations > Configuring Access Authorization > Configuring Agency Authorization**.

1. Use the administrator account to log in to the ModelArts management console.
   
   Select your region in the upper left corner, for example, **CN-Hong Kong**.

2. In the navigation pane on the left, choose **Settings**. The **Global Configuration** page is displayed.

3. Click **Add Authorization**. On the **Add Authorization** page, set parameters listed in **Table 2-5**, select **I have read and agree to the ModelArts Service Statement**, and click **Create**.
Table 2-5 Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>By default, All users is selected, which indicates that all users in the drop-down list, including the current user, will be authorized. You can also select a username from the drop-down list.</td>
</tr>
</tbody>
</table>
| Agency    | - **Use existing**: If there are existing agencies in the list, select an available one to authorize the selected user. Click the drop-down arrow next to an agency name to view its permission details.  
  - **Add agency**: If there is no available agency, create one. If you use ModelArts for the first time, select **Add agency**. |
| Add agency > Agency Name | The system automatically creates an agency name that is editable. |
| Add agency > Permissions > Common User | You can use basic ModelArts functions, for example, accessing data and creating and managing training jobs, but not to manage resources. Select this option generally. Click **View permissions** to view common user permissions. |
| Add agency > Permissions > Custom | You can flexibly assign permissions to the created agency. Select this mode for refined permissions management. |

Figure 2-31 Agency authorization

Verifying User Permissions

It takes 15 to 30 minutes for the OBS permissions to take effect. Therefore, wait for 30 minutes after the configuration and then verify the configuration.
1. Log in to the ModelArts management console as user D, E, or F. On the login page, ensure that IAM User Login is selected. Change the password as prompted upon the first login.

2. Check ModelArts permissions.
   a. Select the target region in the upper left corner, which must be the same as that in the authorization configuration. In this example, CN-Hong Kong is used.
   b. In the navigation pane on the left of the ModelArts management console, choose DevEnviron > Notebook. If a message indicating insufficient permissions is displayed in the upper right corner, you are not allowed to perform operations on notebooks, and the custom policy has taken effect.

   Figure 2-32 Insufficient permissions

   ![Image of insufficient permissions message]

   c. In the navigation pane on the left of the ModelArts management console, choose Data Management > Datasets and click Create. If this operation is successful, you have obtained ModelArts operation permissions.

      Alternatively, you can try other functions, such as Training Management > Training Jobs. If the operation is successful, you can use ModelArts properly.

3. Verify OBS permissions.
   a. In the service list in the upper left corner, select OBS. The OBS management console is displayed.
   b. Click Create Bucket in the upper right corner. If this operation is successful, you have obtained OBS operation permissions.

   The verification shows that you have certain ModelArts operation permissions. Then, you can use ModelArts properly.

2.6 Assigning Permissions to IAM Users for Using Certain Cloud Services

Certain ModelArts functions require the permission to access other HUAWEI CLOUD services. This section describes how to assign specific permissions to IAM users when they use ModelArts.

Context

The permissions of IAM users are controlled by their tenant user. Logging in as a tenant user, you can assign permissions to the target user group through IAM. Then, the permissions are assigned to all members in the user group.
### Table 2-6 Service authorization

<table>
<thead>
<tr>
<th>Target Service</th>
<th>Description</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModelArts</td>
<td>Assign permissions to IAM users for using ModelArts.</td>
<td>Yes</td>
</tr>
<tr>
<td>Object Storage Service (OBS)</td>
<td>Assign permissions to IAM users for using OBS. ModelArts data management, training jobs, and model deployment require OBS for forwarding data.</td>
<td>Yes</td>
</tr>
<tr>
<td>SoftWare Repository for Container (SWR)</td>
<td>Assign permissions to IAM users for using SWR. ModelArts custom images require the SWR FullAccess permission.</td>
<td>Yes</td>
</tr>
<tr>
<td>Key Management Service (KMS)</td>
<td>To use remote SSH of ModelArts notebook, IAM users require KMS authorization.</td>
<td>No</td>
</tr>
<tr>
<td>Intelligent EdgeFabric (IEF)</td>
<td>Assign permissions to IAM users for using IEF. Tenant administrator permissions are required so that ModelArts edge services depending on IEF can be used.</td>
<td>No</td>
</tr>
<tr>
<td>Cloud Eye</td>
<td>Assign permissions to IAM users for using Cloud Eye. Using Cloud Eye, you can view the running statuses of ModelArts real-time services and AI application loads, and set monitoring alarms.</td>
<td>No</td>
</tr>
<tr>
<td>Simple Message Notification (SMN)</td>
<td>Assign permissions to IAM users for using SMN. SMN is used with Cloud Eye.</td>
<td>No</td>
</tr>
<tr>
<td>Virtual Private Cloud (VPC)</td>
<td>During the creation of a dedicated resource pool for ModelArts, IAM users require VPC permissions so that they can customize networks.</td>
<td>No</td>
</tr>
</tbody>
</table>

### Step 1 Create a User Group and Add Users to the User Group

Multiple IAM users can be created under a tenant user, and the permissions of the IAM users are managed by group. This section describes how to create a user group and IAM users and add the IAM users to the user group.

1. Log in to the management console as a tenant user, click your username in the upper right corner, and choose **Identity and Access Management** from the drop-down list to switch to the IAM management console.
2. Create a user group. In the navigation pane on the left, choose **User Groups**. Click **Create User Group** in the upper right corner. Then, set **Name** to **UserGroup-2** and click **OK**.

After the user group is created, the system automatically switches to the user group list. Then, you can add existing IAM users to the user group through user group management. If there is no existing IAM user, create users and add them to the user group.
3. Create IAM users and add them to the user group. In the navigation pane on the left, choose **Users**. On the displayed page, click **Create User** in the upper right corner. On the **Create User** page, add multiple users. Set parameters as prompted and click **Next**.

**Figure 2-35 Creating multiple users**

4. On the **Add User to Group** page, select **UserGroup-2** and click **Create**.

**Figure 2-36 Adding users to the target user group**

The system will automatically add the two users to the target group one by one.

### Step 2 Configure IAM Permissions for the User Group

An IAM user can use cloud services such as ModelArts and OBS only after they are assigned with permissions from the tenant user. This section describes how to assign the permissions to use cloud services to all IAM users in a user group.

1. On the user group list page of IAM, click **Authorize** of the target user group. The **Authorize User Group** page is displayed.
2. Before assigning permissions, learn about minimum permissions requirements of each ModelArts module, as shown in Table 2-6.

3. Assign permissions for using ModelArts. Search for ModelArts. Select either ModelArts FullAccess or ModelArts CommonOperations.

   The differences between the options are as follows:
   - The users with the ModelArts CommonOperations permission can only use resources, but cannot create, update, or delete any dedicated resource pool. You are advised to assign this permission to IAM users.
   - The users with the ModelArts FullAccess permission have all access permissions, including creating, updating, and deleting dedicated resource pools. Exercise caution when selecting this option.

4. Assign permissions for using OBS. Search for OBS and select OBS Administrator. ModelArts training jobs use OBS for forwarding data. Therefore, the permissions for using OBS are required.
5. Assign permissions for using SWR. Search for **SWR FullAccess**. ModelArts custom images require the SWR FullAccess permission.

![Figure 2-40 Assigning permissions for using OBS](image)

**Figure 2-40** Assigning permissions for using OBS

6. (Optional) Assign the key management permission. Remote SSH of ModelArts notebook requires the key management permission. Search for **DEW KeypairFullAccess**.

DEW key management permission is configured in the following regions: **CN North-Beijing1**, **CN North-Beijing4**, **CN East-Shanghai1**, **CN East-Shanghai2**, **CN South-Guangzhou**, **CN Southwest-Guiyang1**, **CN-Hong Kong**, and **AP-Singapore**. In other regions, the KMS key management permission is configured. In this example, the **CN-Hong Kong** region is used. Therefore, the DEW key management permission is to be configured.

![Figure 2-41 Assigning permissions for using SWR](image)

**Figure 2-41** Assigning permissions for using SWR

![Figure 2-42 DEW key management permission](image)

**Figure 2-42** DEW key management permission
7. (Optional) Assign permissions for using IEF. ModelArts requires the Tenant Administrator permission so that edge services depending on IEF can be used. Tenant Administrator has the permission to manage all cloud services, not only the ModelArts service. Exercise caution when assigning the Tenant Administrator permission.

8. (Optional) Assign permissions for using Cloud Eye and SMN. On the details page of a ModelArts real-time service deployed for inference, the number of calls is available. Click View Details to obtain more information. If you want to view the overall running status of ModelArts real-time services and AI application loads on Cloud Eye, assign Cloud Eye permissions to IAM users. To view monitoring data only, select CES ReadOnlyAccess.
To set alarm monitoring on Cloud Eye, you also need to add **CES FullAccess** and SMN permissions.

**Figure 2-46 Assigning alarm monitoring permissions**

![Assigning alarm monitoring permissions](image)

**Figure 2-47 Assigning permissions for using SMN**

![Assigning permissions for using SMN](image)

9. (Optional) Assign permissions for using VPC. To enable custom network configuration when creating a dedicated resource pool, assign permissions for using VPC.

**Figure 2-48 Assigning permissions for using VPC**

![Assigning permissions for using VPC](image)

10. Click **View Selected** in the upper left corner and confirm the selected permissions.
11. Click **Next** and set the minimum authorization scope. Click **Show More**, select **Region-specific projects**, select **ap-southeast-1 [CN-Hong Kong]**, and click **OK**.

In this example, you are allowed to use services in the **CN-Hong Kong** region only.

12. A message is displayed, indicating that the authorization is successful. View the authorization information and click **Finish**. It takes 15 to 30 minutes for the authorization to take effect.

**Step 3 Configure Agent-based ModelArts Access Authorization for the User Group**

After assigning IAM permissions, configure ModelArts access authorization for IAM users on the ModelArts page so that ModelArts can access dependent services such as OBS, SWR, and IEF.

In agent-based ModelArts access authorization, only tenant users are allowed to configure for their IAM users. Therefore, in this example, the administrator needs to configure access authorization for all the IAM users.

1. Use the tenant account to log in to the ModelArts management console. Select your region in the upper left corner, for example, **CN-Hong Kong**.
2. In the navigation pane on the left, choose Settings. The Global Configuration page is displayed.

3. Click Add Authorization. On the Add Authorization page, set Username to All users and select Add agency to configure the agency-based authorization for all IAM users under the tenant account.
   - Common User: You can use basic ModelArts functions, for example, accessing data and creating and managing training jobs, but not to manage resources. Select this mode generally.
   - Custom: You can select permissions from the permission list as required to flexibly assign permissions to the created agency. Select this mode for refined permissions management.
4. Select **I have read and agree to the ModelArts Service Statement** and click **Create**.

**Figure 2-54 Configured agency authorization**

### Step 4 Verify User Permissions

It takes 15 to 30 minutes for the permissions configured in **Step 2 Configure IAM Permissions for the User Group** to take effect. Therefore, wait for 30 minutes after the configuration and then verify the configuration.

1. Log in to the ModelArts management console as an IAM in **UserGroup-2**. On the login page, ensure that **IAM User Login** is selected. Change the password as prompted upon the first login.

2. Check ModelArts permissions.
   a. Select the target region in the upper left corner, which must be the same as that in the authorization configuration. **CN-Hong Kong** is used as an example.
b. In the navigation pane on the left of the ModelArts management console, choose DevEnviron > Notebook. If there is no message indicating insufficient permissions, the ModelArts permissions and agency authorization have been configured.

If the information shown in the following figure is displayed, the ModelArts agency authorization has not been configured. In this case, follow the instructions provided in Step 3 Configure Agent-based ModelArts Access Authorization for the User Group to configure the authorization.

Figure 2-55 Insufficient permissions

![Insufficient permissions](image)

3. Verify OBS permissions.
   a. In the service list in the upper left corner, select OBS. The OBS management console is displayed.
   b. Click Create Bucket in the upper right corner. If this operation is successful, you have obtained OBS operation permissions.

4. Verify SWR permissions.
   a. In the service list in the upper left corner, select SWR. The SWR management console is displayed.
   b. If an SWR page can be properly displayed, you have obtained SWR operation permissions.
5. Verify other optional permissions.
6. Experience ModelArts.
3.1 HUAWEI CLOUD Mascot Detection (Using ExeML for Object Detection)

ModelArts provides ExeML for service developers, freeing you from model development and parameter tuning. With ExeML, you can finish an AI development project in just three steps, including data labeling, auto training, and service deployment.

As an example of object detection, this section describes how to detect Yunbao, the mascot of HUAWEI CLOUD, to help you quickly get started with ModelArts. By using the built-in Yunbao image dataset, the system automatically trains and generates a detection model, and deploys the generated model as a real-time service. After the deployment is completed, you can use the real-time service to identify whether an input image contains Yunbao.

Before you start, carefully complete the preparations described in Preparations. To use ExeML to build a model, perform the following steps:

- **Step 1: Prepare Data**
- **Step 2: Create an Object Detection Project**
- **Step 3: Label Data**
- **Step 4: Generate a Model with ExeML**
- **Step 5: Deploy the Model as a Real-Time Service**
- **Step 6: Test the Service**

**Preparations**

- Your HUAWEI CLOUD account is not in arrears or frozen.
- Access authorization has been configured for your account. For details, see Configuring Agency Authorization. If you have been authorized using access keys, clear the authorization and configure agency authorization.
- OBS buckets and folders are ready for model data storage. For details about how to create OBS buckets and folders, see Creating a Bucket and Creating a Folder. For normal data access, ensure that the created OBS bucket and ModelArts are in the same region.
Data management is required for creating ExeML projects. Therefore, obtain the permission to access OBS from the Data Management module before using ExeML.

On the ModelArts management console, choose Data Management > Datasets in the left navigation pane. On the page that is displayed, click Service Authorization to apply for permission authorization. If you log in using an account, a dialog box is displayed, asking you to accept the authorization. If you log in as an IAM user (member account), your master account or a user with admin permissions grants authorization to you.

**Step 1: Prepare Data**

ModelArts provides a sample dataset of Yunbao named Yunbao-Data-Custom. This example uses this dataset to build a model. Perform the following operations to upload the dataset to the OBS directory test-modelarts/dataset-yunbao created in preparation.

If you want to use your own dataset, skip this step, upload the dataset to the OBS folder, and select this directory in **Step 2: Create an Object Detection Project**.

1. Download the Yunbao-Data-Custom dataset to the local PC.
2. Decompress the Yunbao-Data-Custom.zip file to the Yunbao-Data-Custom directory on the local PC.
3. Batch upload all files from the Yunbao-Data-Custom folder to the test-modelarts/dataset-yunbao directory on OBS. For details, see Uploading a File.

   The obtained dataset has two directories: eval and train. The data stored in train is used for model training, and the data stored in eval is used for model prediction.

**Step 2: Create an Object Detection Project**

1. On the ModelArts management console, click ExeML in the left navigation pane.
2. In the Object Detection box, click Create Project. On the Create Object Detection Project page that is displayed, enter a project name and a dataset name, and select an input dataset path. The OBS path of the Yunbao dataset is /test-modelarts/dataset-yunbao/train/. Select an empty directory in Output Dataset Path.

   **NOTE**

   The Yunbao dataset has two directories: eval and train. Select the data in the train directory for training. If the upper-layer directory of train is selected, an error message is displayed, indicating that OBS has invalid data. As a result, the project will fail to be created.
3. Click **Create Project**. The object detection project is created. After the project is created, the **ExeML > Label Data** page is displayed and data source synchronization is automatically performed.

**Step 3: Label Data**

For an object detection project, labeling data is to locate an object in an image and assign a label to the object. The labeled data is used for model training. In the Yunbao dataset, part of data has been labeled. You can label the unlabeled data for trial use.

Data source synchronization is automatically performed when you create an ExeML project. Data source synchronization takes a certain period of time. If the synchronization fails, you can click **Synchronize Data Source** to manually execute the synchronization.

1. On the **ExeML > Label Data** page, click the **Unlabeled** tab. All unlabeled images are displayed. Click an image to go to the labeling page.

2. Left-click and drag the mouse to select the area where Yunbao is located. In the dialog box that is displayed, enter the label name, for example, **yunbao**, and press **Enter**. After the labeling is completed, the status of the image changes to **Labeled** in the left **Image Catalog** pane.

You can select another image from the image catalog in the lower part of the page and repeat the preceding steps to label the image. If an image contains more than one Yunbao, you can label all. You are advised to label all images in the dataset to train a model with better precision.
3. After all images in the image directory are labeled, click the project name in the upper left corner. In the dialog box that is displayed, click OK to save the labeling information. On the Labeled tab page, you can view the labeled images and view the label names and quantity in the right pane.

Step 4: Generate a Model with ExeML

1. After data labeling is completed, click Train in the upper right corner of the data labeling page. In the Training Configuration dialog box that is displayed, set related parameters. For details, see Figure 3-3.

Figure 3-3 Setting training parameters

**Training Configuration**

- Dataset Version: V004
- Training and Validation Ratios: Training Set Ratio: 0.8, Validation Set Ratio: 0.2
- Incremental Training Version: None
- max training time (minute): 60
- training preference: balance
- Instance Flavor: Compute-intensive 1 instance (GPU)
2. Click **Next**. On the configuration page that is displayed, confirm the specifications and click **Submit** to start auto model training. The training takes a certain period of time. If you close or exit the page, the system continues training until it is completed.

After the training is completed, you can view the training details on the page, such as the accuracy, evaluation result, training parameters, and classification statistics.

**Figure 3-4 Model training**

---

**Step 5: Deploy the Model as a Real-Time Service**

1. On the **Train Model** tab page, wait until the training status changes to **Completed**. Click **Deploy** in the **Version Manager** pane.

**Figure 3-5 Deploying a service**

---

2. In the displayed **Deploy** dialog box, set **Specifications** and **Auto Stop**, and click **OK** to deploy the object detection model as a real-time service.

If you select free specifications, you do not need to set **Auto Stop**, because the node will be stopped one hour later.
3. After the deployment is started, the system automatically switches to the **Deploy Service** tab page. This page displays the deployment progress and status.

The deployment takes a certain period of time. After the deployment is completed, the status in the **Version Manager** pane changes to **Running**.

**Figure 3-7 Successful deployment**

---

**Step 6: Test the Service**

After the model is deployed, you can test the service using an image.

1. On the **Deployment Online** tab page, select a running service version, and click **Upload** to upload a local image.

**Figure 3-8 Uploading an image**

---

2. Select an image from a local environment. The image must contain Yunbao. Click **Predict** to perform the test.

After the prediction is completed, the label name **yunbao**, location coordinates, and confidence score are displayed in the prediction result pane.
on the right. In the prediction result, `detection_boxes` indicates the location of the object, `detection_scores` indicates the detection score of `yunbao`.

If the model accuracy does not meet your expectation, add images on the Label Data tab page, label the images, and train and deploy the model again.

**NOTE**

A running real-time service keeps consuming the resources. If you do not need to use the real-time service, click **Stop** in the Version Manager pane to stop the service and avoid unnecessary billing. If you want to use the service again, click **Start**.

**Figure 3-9 Test result**

![Service Test](image)

---

### 3.2 Bank Deposit Prediction (Using ExeML for Predictive Analytics)

Banks often predict whether customers would be interested in a time deposit based on their characteristics, including the age, work type, marital status, education background, housing loan, and personal loan.

Now, you can use the ExeML function on ModelArts to predict whether a customer would be interested in the time deposit. The process of using ExeML is as follows:

1. **Preparing Data**: Download a dataset and upload it to Object Storage Service (OBS) on HUAWEI CLOUD.
2. **Creating a Predictive Analytics Project**: Create a predictive analytics project based on the existing dataset.
3. **Training a Model**: Preview the data and select the training objective, and then start the model training.
4. **Deploying the Model**: Deploy the trained model as a real-time service and test the prediction result.

**Preparing Data**

In this example, the dataset is from the Machine Learning Repository of UCI. For details about the dataset, see [Bank Marketing Data Set](#) and [Table 3-1](#) and [Table 3-2](#).
describe the parameters and sample data of the dataset. You can obtain the
dataset from GitHub and upload it to OBS.

**Table 3-1** Parameters and meanings of data sources

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attr_1</td>
<td>Age</td>
<td>Int</td>
<td>Age of the customer</td>
</tr>
<tr>
<td>attr_2</td>
<td>Occupation</td>
<td>String</td>
<td>Occupation of the customer</td>
</tr>
<tr>
<td>attr_3</td>
<td>Marital status</td>
<td>String</td>
<td>Marital status of the customer</td>
</tr>
<tr>
<td>attr_4</td>
<td>Education background</td>
<td>String</td>
<td>Education background of the customer</td>
</tr>
<tr>
<td>attr_5</td>
<td>Real estate</td>
<td>String</td>
<td>Real estate of the customer</td>
</tr>
<tr>
<td>attr_6</td>
<td>Loan</td>
<td>String</td>
<td>Loan of the customer</td>
</tr>
<tr>
<td>attr_7</td>
<td>Deposit</td>
<td>String</td>
<td>Deposit of the customer</td>
</tr>
</tbody>
</table>

**Table 3-2** Sample data of the dataset

<table>
<thead>
<tr>
<th>attr_1</th>
<th>attr_2</th>
<th>attr_3</th>
<th>attr_4</th>
<th>attr_5</th>
<th>attr_6</th>
<th>attr_7</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>blue-collar</td>
<td>married</td>
<td>secondary</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>41</td>
<td>management</td>
<td>married</td>
<td>tertiary</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>38</td>
<td>technician</td>
<td>single</td>
<td>secondary</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>39</td>
<td>technician</td>
<td>single</td>
<td>secondary</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>39</td>
<td>blue-collar</td>
<td>married</td>
<td>secondary</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>39</td>
<td>services</td>
<td>single</td>
<td>unknown</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

1. Download the **ModelArts-Lab** project from GitHub and obtain the **train.csv**
   training data file from the `\ModelArts-Lab-master\official_examples\Using_ModelArts_to_Create_a_Bank_Marketing_Application\data`
directory of the project.

2. Upload the **train.csv** file to OBS, for example, to the **test-modelarts/bank-
   marketing** directory. For details about how to upload files to OBS, see
   **Uploading a File**.
Creating a Predictive Analytics Project

1. On the ModelArts management console, click ExeML in the left navigation pane.

   Figure 3-10 ExeML

2. On the ExeML page, click Create Project in the Predictive Analytics area.

3. On the Create Predictive Analytics Project page, set the project name and select the OBS path where the training data is stored. In this example, the dataset path is test-modelarts/bank-marketing/train.csv. Click Create Now. The data labeling page is displayed, as shown in Figure 3-11.

   Figure 3-11 Creating a predictive analytics project

Training a Model

1. On the data labeling page, preview the data and select the training objective on the page displayed. The training objective here is to determine whether the customer will apply for a deposit (specified in attr_7). Set Label Column Data Type to Discrete value. After the training objective is specified, click Train.
2. In the displayed Training Configuration dialog box, select an instance flavor used for training, click Next to check the training configuration, and click Submit to start model training.

The training takes a certain period of time. If you close or exit the page, the system continues training until it is complete.

3. In the upper left corner of the model training page, if the status of the training job changes to Completed, the training job is complete. The Training Details area on the right shows the details about the training job.

Deploying the Model

1. On the Train Model page, click Deploy in the Version Manager area. The system starts to deploy the service and automatically switches to the Deploy Service page.

2. In the Version Manager area, when the status changes to Running, the service has been deployed. You can test the service in the prediction area. The following shows the test code. As shown in Figure 3-15, the prediction result is "predict": "no", indicating that the customer will not apply for a deposit.
"data":
{
"count": 1,
"req_data":
[
{
"attr_1": "34",
"attr_2": "blue-collar",
"attr_3": "single",
"attr_4": "tertiary",
"attr_5": "no",
"attr_6": "no"
}
]
}

Figure 3-15 Testing the model
4.1 Using a Notebook for Handwritten Digit Recognition

ModelArts provides notebook instances for AI engineers. The engineers can prepare data, train models, and perform prediction in notebook instances.

This section describes how to use MoXing to develop a handwritten digit recognition application, helping you quickly learn about AI development with notebook instances on ModelArts.

MNIST is a dataset containing handwritten digits, and is often used as an introductory example of deep learning. In this example, model training and prediction code (provided by ModelArts by default) for the MNIST dataset are compiled using the MoXing interface. You can use this example to complete model training in a notebook instance and upload images for prediction.

Before you start, carefully complete the preparations described in Preparations. To use a notebook instance to build a model, perform the following steps:

- **Step 1: Prepare Data**
- **Step 2: Use a Notebook Instance to Train a Model and Perform Prediction**
- **Step 3: Delete Related Resources to Avoid Unnecessary Billing**

**Preparations**

- Your HUAWEI CLOUD account is not in arrears or frozen.
- Access authorization has been configured for your account. For details, see Configuring Agency Authorization. If you have been authorized using access keys, clear the authorization and configure agency authorization.
- You have created a bucket and folders in OBS for storing the sample dataset and model. In this example, create a bucket named `test-modelarts` and folders listed in Table 4-1.
  
  For details about how to create OBS buckets and folders, see Creating a Bucket and Creating a Folder. Ensure that the OBS directory you use and ModelArts are in the same region.
Table 4-1 Folder list

<table>
<thead>
<tr>
<th>Folder</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataset-mnist</td>
<td>Stores the dataset.</td>
</tr>
<tr>
<td>mnist-MoXing-code</td>
<td>Stores the compiled model code file mnist_example.ipynb.</td>
</tr>
<tr>
<td>train-log</td>
<td>Stores images for prediction.</td>
</tr>
</tbody>
</table>

- In this example, ModelArts provides a compiled model code file mnist_example.ipynb. You need to obtain this file from GitHub. After the model training is completed, you need to upload the file to the specified location.
  a. Go to the ModelArts-Lab project on GitHub, click Clone or download, and then click Download ZIP to download the project.

  ![Figure 4-1 Downloading ModelArts-Lab](image)

  b. After the project is downloaded, decompress the ModelArts-Lab-master.zip file and obtain sample code file mnist_example.ipynb from the |
  \ModelArts-Lab-master\official_examples \Using_Notebook_to_Create_a_MNIST_Dataset_Recognition_Application\code directory.
  c. Upload the mnist_example.ipynb file to the mnist-MoXing-code folder of the test-modelarts bucket. For details, see Uploading a File.

- Prepare an image of 28 x 28 pixels with a white handwritten digit on the black background. For example, prepare an image named 7.jpg. The image contains a handwritten digit 7. Upload the prepared image to the train-log folder of the test-modelarts bucket for prediction.

Step 1: Prepare Data

ModelArts provides a sample MNIST dataset named Mnist-Data-Set. This example uses this dataset to build a model. Perform the following operations to upload the dataset to the OBS directory test-modelarts/dataset-mnist created in preparation.

1. Download the Mnist-Data-Set dataset to the local PC.
2. Decompress the **Mnist-Data-Set.zip** file to the **Mnist-Data-Set** directory on the local PC.

3. Batch upload all files from the **Mnist-Data-Set** folder to the **test-modelarts/dataset-mnist** directory on OBS. For details, see **Uploading a File**.

The following provides content of the **Mnist-Data-Set** dataset. .gz is the compressed package.

- **t10k-images-idx3-ubyte**: validation set, which contains 10,000 samples
- **t10k-images-idx3-ubyte.gz**: compressed package file of the validation set
- **t10k-labels-idx1-ubyte**: labels of the validation set, which contain the labels of the 10,000 samples
- **t10k-labels-idx1-ubyte.gz**: compressed package file of the validation set label
- **train-images-idx3-ubyte**: training set, which contains 60,000 samples
- **train-images-idx3-ubyte.gz**: compressed package file of the training set
- **train-labels-idx1-ubyte**: labels of the training set, which contain the labels of the 60,000 samples
- **train-labels-idx1-ubyte.gz**: compressed package file of the training set label

### Step 2: Use a Notebook Instance to Train a Model and Perform Prediction

After data preparation is completed, use a notebook instance to compile code for modeling. ModelArts provides a sample code file **mnist_example.ipynb** for handwritten digit image training and prediction based on MoXing.

1. Obtain the **mnist_example.ipynb** file and upload it to OBS, for example, **test-modelarts/mnist-MoXing-code**. For details, see **Preparations**.

2. On the ModelArts management console, choose **DevEnviron > Notebooks** and click **Create** in the upper left corner.

3. On the **Create Notebook** page, set required parameters by referring to **Table 4-2**, and click **Next**.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Notebook instance name, which can be customized</td>
</tr>
<tr>
<td>Auto Stop</td>
<td>To avoid resource waste, enable auto stop and select 1 hour later</td>
</tr>
<tr>
<td>Work Environment</td>
<td>Select <strong>Multi-Engine 1.0 (Python 3, Recommended)</strong>.</td>
</tr>
<tr>
<td>Resource Pool</td>
<td>Select <strong>Public resource pools</strong>.</td>
</tr>
<tr>
<td>Type</td>
<td>Select <strong>GPU</strong>.</td>
</tr>
<tr>
<td>Instance Flavor</td>
<td>Select **GPU: 1 x v100NV32 CPU: 8 vCPUs</td>
</tr>
</tbody>
</table>
Parameter | Description
--- | ---
Storage | Select OBS and set **Storage Path** to the OBS path for storing sample files, for example, `test-modelarts/mnist-MoXing-code`.

4. On the **Confirm** tab page, check the configurations and click **Submit** to create a notebook instance.

5. On the **Notebook** management page, when the status of the created notebook instance changes to **Running**, the notebook instance has been successfully created. In the **Operation** column, click **Open**. The Jupyter page is displayed.

6. On the **Files** tab page of the Jupyter page, you can see the sample code file uploaded in 1. Click the file name. The notebook details page is displayed. On the notebook details page, select the kernel environment that matches the code environment as prompted, for example, **TensorFlow-1.8**, and click **Set Kernel**. If the following prompt is not displayed, skip the kernel environment configuration, and go to the next step.

**Figure 4-2** Setting the kernel

```html
Kernel not found
```

7. On the notebook instance details page, check the details about the sample code file, including data preparation, training model, and prediction.

a. **Data preparation**: Data has been prepared according to [Step 1: Prepare Data](https://www.example.com/step1). The dataset file is stored in `test-modelarts/dataset-mnist/`. The sample code provides the description of the dataset.

b. **Model training**

In the model training area, change **data_url** to the OBS path where the dataset is located in [Step 1: Prepare Data](https://www.example.com/step1). You can copy the OBS path from OBS Console, and change the OBS path to a path starting with `obs://`. The following gives an example:

```python
data_url = 'obs://test-modelarts/dataset-mnist/
```

After the code is modified, click from the first cell to run all the cells in the model training area. The run log is displayed at the end of the model training area. If information similar to the following is displayed, the model training is successful. The following log information indicates that the model training is successful and the model file is successfully generated:

```text
INFO:tensorflow:No assets to write.
INFO:tensorflow:Restoring parameters from ./cache/log/model.ckpt-1000
INFO:tensorflow:SavedModel written to: b'/cache/log/model/saved_model.pb'
```
An exception has occurred, use %tb to see the full traceback.

c. **Prediction**

After the model training is completed, upload an image and use the generated model for prediction. As described in [Preparations](#), upload the 7.jpg file to the test-modelarts/train-log path for prediction.

On the notebook instance, change the value of `src_path` in the prediction area to the actual OBS path of the image. Use the OBS path starting with `obs://`.

```python
src_path = 'obs://test-modelarts/train-log/7.jpg'
```

After the code is modified, click ![Run](run.png) from the first cell to run all the cells in the prediction area. The run log is displayed at the end of the prediction area. If information similar to the following is displayed, the prediction result is displayed. For example, the predicted handwritten digit in this example is 7. Compare the digits in the image and prediction result to check whether the prediction result is correct.

```
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Done running local_init_op.
The result: [7]
INFO:tensorflow: [1 examples]
INFO:tensorflow: [1 examples]
```

An exception has occurred, use %tb to see the full traceback.

---

**Step 3: Delete Related Resources to Avoid Unnecessary Billing**

To avoid unnecessary billing, delete related resources after the trial use. In this example, delete the data and notebook instance.

- To delete a notebook, choose [DevEnviron > Notebooks](#) and then click [Delete](#) in the [Operation](#) column.
- To delete data, access OBS, delete the uploaded data, and delete the folder and OBS bucket.
5 Model Development (Custom Algorithms in Training Jobs of the New Version)

5.1 Using a Custom Algorithm to Build a Handwritten Digit Recognition Model

This section describes how to modify a local custom algorithm to train and deploy models on ModelArts.

Scenarios

This case describes how to use TensorFlow 1.8 to recognize handwritten digit images. An official MNIST dataset is used in this case.

Through this case, you can learn how to use your own algorithms to train jobs, deploy an inference model, and perform prediction on ModelArts.

Process

Before performing the following operations, complete necessary operations. For details, see Prerequisites.

1. **Step 1 Prepare the Training Data**: Obtain an MNIST dataset and upload it to OBS.
2. **Step 2 Prepare Training Files and Inference Code Files**: Obtain the training script, inference code file, and configuration file from GitHub/Gitee. Modify the local algorithm code to adapt to model training on ModelArts.
3. **Step 3 Create an OBS Bucket and Upload Files to OBS**: Create an OBS bucket and folder, and upload the training data and script file to OBS.
4. **Step 4 Create an Algorithm**: Upload a locally developed algorithm from OBS to ModelArts.
5. **Step 5 Train a Model**: Use the uploaded algorithm to train a model.
6. **Step 6 Deploy the Model for Inference**: Import the trained model to ModelArts, create an AI application, and deploy the AI application as a real-time service.

7. **Step 7 Perform Prediction**: Upload a handwritten digit image and send an inference request to obtain the inference result.

8. **Step 8 Release Resources**: Stop the service and delete the data in OBS to stop billing.

**Prerequisites**

You have registered with HUAWEI CLOUD and checked the account status before using ModelArts. The account cannot be in arrears or frozen.

**Step 1 Prepare the Training Data**

An MNIST dataset downloaded from the MNIST official website is used in this case. Ensure that the four files are all downloaded.

![Figure 5-1 MNIST dataset](image)

Four files are available on this site:

- `train-images-idx3-ubyte.gz`: training set images (9912422 bytes)
- `train-labels-idx1-ubyte.gz`: training set labels (28881 bytes)
- `t10k-images-idx3-ubyte.gz`: test set images (1648877 bytes)
- `t10k-labels-idx1-ubyte.gz`: test set labels (4542 bytes)

Decompress the downloaded file in a local directory. The decompressed dataset contains the following files, where .gz indicates a compressed package:

- `t10k-images-idx3-ubyte`: validation set, which contains 10,000 samples
- `t10k-images-idx3-ubyte.gz`: compressed package of the validation set
- `t10k-labels-idx1-ubyte`: validation set labels of the 10,000 samples
- `t10k-labels-idx1-ubyte.gz`: compressed package of the validation set labels
- `train-images-idx3-ubyte`: training set, which contains 60,000 samples
- `train-images-idx3-ubyte.gz`: compressed package of the training set
- `train-labels-idx1-ubyte`: training set labels of the 60,000 samples
- `train-labels-idx1-ubyte.gz`: compressed package of the training set labels

**Step 2 Prepare Training Files and Inference Code Files**

In this case, ModelArts provides the training script, inference code, and configuration file for service deployment. Perform the following operations to obtain the sample files:

Go to the ModelArts-Lab project on GitHub, click Clone or download, and then click Download ZIP to download the project.
After the project is downloaded, decompress **ModelArts-Lab-master.zip** and obtain training script **train_mnist_tf.py**, inference code **customize_service.py**, and configuration file **config.json** in \ModelArts-Lab-master\official_examples\Using Custom Algorithms to Build Models on ModelArts\codes.

The preceding code files have been adapted and can be directly used for training on ModelArts. If you use a custom training script, adapt the local code. For details, see **Developing a Custom Script**.

**Step 3 Create an OBS Bucket and Upload Files to OBS**

The data and code files used for training must be uploaded to an OBS bucket. When running a training job on ModelArts, read data and code files from the OBS bucket.

1. Log in to OBS management console and create an OBS bucket and folder. For details, see **Creating a Bucket** and **Creating a Folder**.

```bash
# OBS bucket name, which is customizable, for example, test-modelarts-xx
(OBS bucket)

# OBS bucket name, which is customizable, for example, mnist
-(OBS folder)

# OBS folder, which is used to store the training dataset. The folder name is customizable, for example, mnist-data.
-mnist-data

# OBS folder, which is used to store training script train_mnist_tf.py. The folder name is customizable, for example, mnist-code.
-mnist-code

# OBS folder, which is used to store trained models. The folder name is customizable, for example, output.
-output

# Model folder automatically generated after a training job is complete. It is used to store trained models, inference code, and configuration file. The inference code customize_service.py and configuration file config.json need to be uploaded to this directory after the training is complete.
```

**Figure 5-2 Downloading ModelArts-Lab**

**Figure 5-3 Training code files**
- The region where the created OBS bucket resides must be the same as that where ModelArts is used. Otherwise, the OBS bucket will be unavailable for training. For details, see How Do I Check Whether ModelArts and an OBS Bucket Are in the Same Region?
- When creating an OBS bucket, do not set the archive storage class. Otherwise, training models will fail.

**Figure 5-4 OBS file directory**

2. Upload data to OBS, for example, `obs://test-modelarts-lvy/mnist/mnist-data`. For details, see Uploading a File.

**CAUTION**

When uploading data to OBS, do not encrypt the data. Otherwise, the training will fail.
3. Upload training script `train_mnist_tf` to OBS, for example, `obs://test-modelarts-lvy/mnist/mnist-code`.

**Step 4 Create an Algorithm**

1. Log in to the ModelArts management console and select the same region as the OBS bucket.
2. In the navigation pane on the left, choose **Settings** and check whether access authorization has been configured for the current account. For details, see **Configuring Access Authorization**. If you have been authorized using access keys, clear the authorization and configure agency authorization.
3. In the navigation pane on the left, choose **Algorithm Management**, click the **My algorithms** tab, and click **Create**. Configure parameters for creating an algorithm.
   - **Startup Mode**: Select **Preset images** and then **Show Old Images**. Then, choose **TensorFlow** and **TF-1.8.0-python3.6** from the drop-down list.
   - **Boot File**: Select the training script `train_mnist_tf.py` uploaded to the code directory.
   - **Input** in **Configure Pipeline**: Add one input and set its **Parameter Name** to `data_url`.
   - **Output** in **Configure Pipeline**: Add one output and set its **Parameter Name** to `train_url`.
   - Retain default values for other parameters.
Figure 5-6 Parameters for creating an algorithm

4. Check the algorithm parameters and click Submit. On My algorithms page, click the algorithm name to view its detailed information.

Step 5 Train a Model

After an algorithm is created, perform the following operations to train a model:

1. On the ModelArts management console, choose Training Management > Training Jobs, and click Create Training Job in the upper right corner.

2. On the Create Training Job page, configure parameters.
   - Created By > My algorithms: Select the algorithm created in Step 4 Create an Algorithm. The latest algorithm is selected by default.
   - Training Input: Select the path where the data is stored and then the OBS path test-modelarts-lvy/mnist/mnist-data.
   - Training Output: Select the OBS path /test-modelarts-lvy/mnist/ output/ created in Step 3 Create an OBS Bucket and Upload Files to OBS.
   - Resource Type: Select GPU and then GPU: 1*NVIDIA-V100(32GB) | CPU: 8 vCPUs 64GB.

Figure 5-7 Selecting an algorithm
When configuring training input and output parameters, click in the lower right corner of the page and view the mapping between the training input and output paths in the training container, facilitating training code debugging.

3. Click **Submit**, verify parameter settings for the training job, and click **Yes**.

4. The system automatically switches back to the **Training Jobs** page. When the training job status changes to **Completed**, the model training is complete.

5. Click the training job name. On the job details page that is displayed, check whether there are error messages in logs. If so, the training failed. Identify the cause and locate the fault based on the logs.

6. In the lower left corner of the training details page, click the training output path to go to OBS. Then, check whether the **model** folder is available and whether there are any trained models in the folder. If there is no **model** folder or trained model, the training input may be incomplete. In this case, completely upload the training data and train the model again.
Step 6 Deploy the Model for Inference

After the model training is complete, create an AI application and deploy it as a real-time prediction service.

1. Upload the inference code `customize_service.py` and configuration file `config.json` obtained in Step 2 Prepare Training Files and Inference Code Files to `obs://test-modelarts-lvy/mnist/output/model`.

   After the training job is complete, the `model` folder will be automatically generated in the training output path `obs://test-modelarts-lvy/mnist/output/test-modelarts-cz/mnist-model` to store trained models.
Figure 5-12 Uploading the inference code and configuration file

If there is no model folder or trained model in the training output path, the training input may be incomplete. In this case, completely upload the training data and train the model again.

**NOTE**

If you want to use your own inference code and configuration file, ensure that these files comply with the Model Package Specifications.

2. Log in to the ModelArts management console. In the navigation pane on the left, choose AI Application Management > AI Applications. On the My AI Applications page, click Create.

3. On the Create page, configure parameters and click Create now. Choose Training Job > Training Jobs (New) for Meta Model Source. The system automatically selects the latest training job. The values of AI Engine will be automatically configured.
On the AI Applications page, if the application status changes to Normal, the AI application has been created.

4. Click the down arrow next to an AI application name to view all versions of the application. In the row of a version, choose Deploy > Real-Time Services in the Operation column to deploy the version of the AI application as real-time services.

5. On the Deploy page, configure parameters and create a real-time service as prompted.
After you submit the service deployment request, the system automatically switches to the **Real-Time Services** page. When the service status changes to **Running**, the service has been deployed.

**Figure 5-16 Deployed**

![Deploy, Delete, Authorize buttons]

<table>
<thead>
<tr>
<th>Name/ID</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>service-4727</td>
<td>Running (5 minutes until stop)</td>
</tr>
</tbody>
</table>

**Step 7 Perform Prediction**

1. On the **Real-Time Services** page, click the name of the real-time service. The real-time service details page is displayed.

2. Click the **Prediction** tab. On the right of **Image File**, click **Upload** to upload a sample image and click **Predict**.

After the prediction is complete, the prediction result is displayed in the **Test Result** pane. According to the prediction result, the digit on the image is **2**.

**NOTE**

The MNIST used in this case is a simple dataset used for demonstration, and its algorithms are also simple neural network algorithms used for teaching. The models generated using such data and algorithms are applicable only to teaching but not to complex prediction scenarios. The prediction is accurate only if the image used for prediction is similar to the image in the training dataset.

ModelArts AI Gallery provides commonly used high-precision algorithms and their training datasets. Obtain them in **the asset market of AI Gallery**.

**Figure 5-17 Example**

![Image of digit 2]

**Figure 5-18 Prediction results**

![Prediction result image]
Step 8 Release Resources

If you do not need to use this model and real-time service anymore, release the resources to stop billing.

- On the Real-Time Services page, locate the row containing the target service and click Stop or Delete in the Operation column.
- On the AI Applications page in AI Application Management, locate the row containing the target service and click Delete in the Operation column.
- On the Training Jobs (New) page in Training Management, locate the row containing the target service and click Delete in the Operation column.
- Go to OBS and delete the OBS bucket, folders, and files used in this example.
6 Custom Images

6.1 Creating a Training Job Using a Custom Image (GPU)

This section provides an example of using a custom image to create a training job on the ModelArts platform, to help you quickly get familiar with the platform.

Before using a custom image to create a training job, you need to be familiar with the software and have development experience. The detailed procedure is as follows:

1. Preparations
2. Creating a Custom Image
3. Uploading an Image to SWR
4. Creating a Training Job

Preparations

- You have registered with HUAWEI CLOUD and checked the account status before using ModelArts. The account cannot be in arrears or frozen.
- Your account has been authorized to access ModelArts. For details, see Configuring Agency Authorization. If you have been authorized using access keys, clear the authorization and configure agency authorization.
- You have created a bucket and folder in OBS for storing the sample dataset and training code. In this example, create a bucket named test-modelarts and folders listed in Table 6-1.

For details about how to create OBS buckets and folders, see Creating a Bucket and Creating a Folder. Ensure that the OBS directory you use and ModelArts are in the same region.
Table 6-1 Folder list

<table>
<thead>
<tr>
<th>Folder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>obs://test-modelarts/deep-learning/mnist/</td>
<td>Stores the training script.</td>
</tr>
<tr>
<td>obs://test-modelarts/deep-learning/mnist/dataset-mnist</td>
<td>Stores the dataset.</td>
</tr>
<tr>
<td>obs://test-modelarts/deep-learning/mnist/output</td>
<td>Stores training output files.</td>
</tr>
<tr>
<td>obs://test-modelarts/deep-learning/mnist/log</td>
<td>Stores log files.</td>
</tr>
</tbody>
</table>

- In this example, the MNIST dataset is used. Upload it to the corresponding folder.
- For details about the training code, see mnist_softmax.py file in the GitHub repository. Upload the file to the corresponding folder.

Creating a Custom Image

In this example, the Dockerfile file is used to customize an image.

A Linux x86_x64 host is used here. You can purchase an ECS of the same specifications or use an existing local host to create a custom image.

1. Install Docker. For details, see Docker Documentation.

   The following uses the Linux x86_64 OS as an example to describe how to obtain the Docker installation package. You can run the following command to install Docker:
   ```bash
   curl -fsSL get.docker.com -o get-docker.sh
   sh get-docker.sh
   ``

   If the docker images command is successfully executed, Docker has been installed. In this case, skip this step.

2. Obtain a basic image.

   To facilitate code download, training log output, and log file upload to OBS, ModelArts provides basic image packages for creating custom images. For details about basic images, see Specifications for Custom Images Used for Training Jobs. In this example, the following basic image is obtained:
   ```bash
   docker pull swr.cn-north-4.myhuaweicloud.com/modelarts-job-dev-image/custom-base-cuda10.0-cp36-ubuntu18.04-x86:1.1
   ```

   In addition, you can run the docker images command to view the local image list.

3. Compile a Dockerfile for building a custom TensorFlow 1.13.2 image.

   The file name is tf-1.13.2.dockerfile. Run the vi tf-1.13.2.dockerfile command to switch to the Dockerfile.

   For details about how to compile Dockerfile, see Dockerfile Reference.

   ```bash
   FROM swr.cn-north-4.myhuaweicloud.com/modelarts-job-dev-image/custom-base-cuda10.0-cp36-ubuntu18.04-x86:1.1
   # Configure the HUAWEI CLOUD source and install TensorFlow.
   RUN cp -a /etc/apt/sources.list /etc/apt/sources.list.bak &
   ```
4. Press Esc to exit the editing mode and run the `wq` command to save the `tf-1.13.2.dockerfile` file and exit.

5. Run the following command in the directory where the `tf-1.13.2.dockerfile` file is located to build the custom image `tf-1.13.2:latest`:

   ```bash
docker build -f tf-1.13.2.dockerfile . -t tf-1.13.2:latest
   ```

---

**Figure 6-1 Creating a custom image**

![Image showing the creation of a custom image](image_url)

**Figure 6-2 SWR console**

![Image showing the SWR console](image_url)

---

**Uploading an Image to SWR**

1. Log in to the SWR console and select **CN North-Beijing4**.

2. Click **Create Organization** in the upper right corner and enter an organization name to create an organization. You can customize the organization name. In this example, **deep-learning** is used.
Figure 6-3 Creating an organization

Create Organization

- Each organization name must be globally unique.
- You can create 5 organizations.
- For centralized management of images, limit each organization to one company, department, or individual.
- Examples
  - Company or department: cloud-hangzhou or cloud-develop
  - Person: john

Organization Name: deep-learning

3. Click **Create Login Command** in the upper right corner to obtain a login command.

Figure 6-4 Login command

Login Command

```
docker login -u cn-north-4+ai-appengine-1-0:0000
c2740677b6f0b57461a9051f8f336f1149a7143e9b9a32d525852515e441730
264 swr.cn-north-4.myhuaweicloud.com
```

4. Log in to the local environment as the **root** user and enter the login command.

5. Upload the image to the SWR image repository.
   a. Run the following **docker tag** command to add a tag to the uploaded image:
      ```
```
   b. Run the following **docker push** command to upload the image:
      ```
sudo docker push swr.cn-north-4.myhuaweicloud.com/deep-learning/tf-1.13.2:latest
```
6. After the image is uploaded, you can click My Images on the left navigation pane of the SWR console to view the uploaded custom images.

```
```

swr.cn-north-4.myhuaweicloud.com/deep-learning-diy/tf-1.13.2:latest is the SWR URL of the custom image.

**Creating a Training Job**

1. Log in to the ModelArts management console. In the left navigation pane, choose **Training Management** > **Training Jobs**. By default, the system switches to the **Training Jobs** page.

2. On the **Create Training Job** page, set required parameters and click **Next**.

**Figure 6-7 Name and Description**

```
Name: trainjob-testimage
Version: V0001 (System-defined version number)
Description:
```

Copyright © Huawei Technologies Co., Ltd.
Figure 6-8 Job parameter setting

- Image path: swr.cn-north-4.myhuaweicloud.com/deep-learning/tf-1.13.2:latest
- Code directory: obs://test-modelarts/deep-learning/mnist/
- Command: bash /home/work/run_train.sh python /home/work/user-job-dir/mnist/mnist_softmax.py --data_url /home/work/user-job-dir/mnist/dataset-mnist

Figure 6-9 Configuring resources

3. On the Confirm tab page, check the parameters of the training job and click Submit.
4. After the training job is created, the code directory is downloaded, the custom image is reviewed, and the training job is complete in the background. Generally, training jobs run for a period of time, which may be several minutes or tens of minutes depending on the amount of data and resources you select. After the program is executed successfully, the log similar to the following is output.
### Figure 6-10 Run logs

![Run logs](image-url)

`Log File`:
- `job-20190772.0`

```
[ModelArts Service Log]modelarts-pipes: will write log files: test-dls-task-1.log
[ModelArts Service Log]modelarts-pipes: param for max log length: 1073741824
[ModelArts Service Log]modelarts-pipes: param for whether exit on overflow: 0
[ModelArts Service Log]modelarts-pipes: total length: 8921
[ModelArts Service Log]write cluster ip to ./cluster_ip.json and local ip to ./local_ip.json
[ModelArts Service Log]user: uid=root gid=root groups=root
[ModelArts Service Log]app_url: /home/work/test-modelarts-loc/deep-learning/mnist
[ModelArts Service Log]app_url: /model/dls-task-1.log
```
A

Obtaining the Sample Dataset of ModelArts

ModelArts provides multiple samples based on various AI engines for beginners. For details about the samples, see the ModelArts Best Practices. For each sample, ModelArts has stored the sample dataset in the public OBS bucket. You can select the OBS path based on your region to obtain the sample dataset.

For details about the storage information about each sample dataset, see Sample Dataset Storage Path. You can use different methods to copy the sample dataset to your OBS bucket based on the region where your OBS bucket resides. For details about the methods, see Figure A-1. If the dataset is large, you are advised to use method 2 or 3 to copy it. If the dataset is small, method 1 is recommended.

- **Method 1: Download and Upload Files**
- **Method 2: Use the MoXing API to Copy a Dataset from the Public Bucket to Your OBS Bucket**
- **Method 3: Use the obsutil Tool of OBS to Copy Files**

**Figure A-1** Copying the sample dataset to your OBS bucket

**Sample Dataset Storage Path**

When the sample dataset is stored, two formats are available: compressed and decompressed files. The data of both formats is the same.

- Compressed file: After downloading the compressed file, upload it to your OBS bucket. You need to decompress the file before using it. However, it is convenient to download the file.
Decompressed file: You can directly copy the decompressed file to your OBS bucket. That is, you can copy an OBS bucket to another OBS bucket. The prerequisite is that your OBS bucket and the OBS bucket of the sample dataset belong to the same region.

Table A-1 Sample dataset storage path

<table>
<thead>
<tr>
<th>Sample Name</th>
<th>Dataset Format</th>
<th>Region</th>
<th>OBS Path</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yunbao Detection</td>
<td>Compressed file</td>
<td>CN-Hong Kong</td>
<td><a href="https://modelarts-apsoutheast1-market-dataset.obs.ap-southeast-1.myhuaweicloud.com/dataset-market/Yunbao-Data-Custom/archiver/Yunbao-Data-Custom.zip">https://modelarts-apsoutheast1-market-dataset.obs.ap-southeast-1.myhuaweicloud.com/dataset-market/Yunbao-Data-Custom/archiver/Yunbao-Data-Custom.zip</a></td>
<td>HUAWEI CLOUD Mascot Detection (Using ExeML for Object Detection)</td>
</tr>
<tr>
<td>Decompressed file</td>
<td>CN-Hong Kong</td>
<td>obs://modelarts-apsoutheast1-market-dataset/dataset-market/Yunbao-Data-Custom/unarchiver</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Handwritten Digit Recognition</td>
<td>Compressed file</td>
<td>CN-Hong Kong</td>
<td><a href="https://modelarts-apsoutheast1-market-dataset.obs.ap-southeast-1.myhuaweicloud.com/dataset-market/Mnist-Data-Set/archiver/Mnist-Data-Set.zip">https://modelarts-apsoutheast1-market-dataset.obs.ap-southeast-1.myhuaweicloud.com/dataset-market/Mnist-Data-Set/archiver/Mnist-Data-Set.zip</a></td>
<td>Using a Notebook for Handwritten Digit Recognition</td>
</tr>
<tr>
<td>Decompressed file</td>
<td>CN-Hong Kong</td>
<td>obs://modelarts-apsoutheast1-market-dataset/dataset-market/Mnist-Data-Set/unarchiver</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Method 1: Download and Upload Files

There is no specific restriction on the region. You can select any region to download the dataset. To improve operation efficiency, you are advised to download the compressed dataset. However, the download and upload speeds depend on your local network conditions.

Figure A-2 Operations of method 1
1. Select the storage path of the target sample dataset. You can select a dataset in an OBS bucket in any region. You are advised to download the compressed dataset file. Click the link to download the sample dataset to the local PC.

For example, if you click the link for downloading the Yunbao-Data-Custom.zip dataset for Yunbao Detection in the CN North-Beijing1 region, the Yunbao-Data-Custom.zip file is downloaded to the local PC.

2. Decompress the obtained file and upload all folders of the dataset to the OBS directory.

   a. First, create an OBS bucket and a folder for storing the sample dataset. For example, create an OBS bucket named test-modelarts and a folder named dataset-yunbao.

   b. Decompress the Yunbao-Data-Custom.zip file to the Yunbao-Data-Custom directory on the local PC.

   c. Upload all files in the Yunbao-Data-Custom directory to the test-modelarts/dataset-yunbao directory on OBS. For details about how to upload files to OBS, see Uploading a File.

**Method 2: Use the MoXing API to Copy a Dataset from the Public Bucket to Your OBS Bucket**

The sample dataset must be in the same region as your OBS bucket and you are familiar with notebook operations and ModelArts MoXing. You can copy the sample dataset from the public bucket to your OBS bucket.

You are advised to obtain the OBS path (in OBS format) of the desired decompressed dataset listed in Table A-1, create a notebook instance in ModelArts, and copy the dataset to your OBS bucket.

1. Access the ModelArts management console, create a notebook instance, and create a file on the Jupyter page.

2. Click the new file to access the development environment.

3. Check whether the public bucket where the sample dataset resides is accessible.

   For example, obtain the sample dataset of Yunbao Detection in the CN North-Beijing1 region from Table A-1. The OBS path is obs://modelarts-cnnorth1-market-dataset/dataset-market/Yunbao-Data-Custom/unarchiver. Run the following command to check whether the public bucket is accessible:

   ```python
   import moxing as mox
   mox.file.exists('obs://modelarts-cnnorth1-market-dataset/dataset-market/Yunbao-Data-Custom/unarchiver')
   ```

   If True is returned, the OBS bucket is normal.

4. Check whether your OBS bucket can be accessed.

   For example, create an OBS bucket named test-modelarts and a folder named dataset-yunbao. Run the following command to check whether your bucket is accessible:

   ```python
   import moxing as mox
   mox.file.exists('obs://test-modelarts/dataset-yunbao')
   ```

   If True is returned, the OBS bucket is normal.

5. Check whether you have the write permission on the OBS bucket.
For example, the path of the target OBS bucket is `obs://test-modelarts/dataset-yunbao`. Run the following command to check the permission:

```python
import moxing as mox
mox.file.write('obs://test-modelarts/dataset-yunbao/obs_file.txt', 'Hello, OBS Bucket!')
mox.file.remove('obs://test-modelarts/dataset-yunbao/obs_file.txt', recursive=False)
```

6. Run the following command to copy the sample dataset from the public bucket to your OBS bucket:

```python
import moxing as mox
mox.file.copy_parallel('obs://modelarts-cnnorth1-market-dataset/dataset-market/Yunbao-Data-Custom/unarchiver', 'obs://test-modelarts/dataset-yunbao')
print ('Copy procedure is completed')
```

When **Copy procedure is completed** and the execution time are returned, the dataset is copied. Information similar to the following is displayed:

```
Copy procedure is completed
CPU times: user 117 ms, sys: 92.3 ms, total: 209 ms
Wall time: 58.3 s
```

**Method 3: Use the obsutil Tool of OBS to Copy Files**

The sample dataset must be in the same region as your OBS bucket. You can use the obsutil tool provided by OBS to copy the sample dataset. You are advised to obtain the OBS path (in OBS format) of the decompressed dataset file in **Table A-1** and copy the file to your OBS bucket by running the object copy command in **Copying an Object**.

For details about how to use obsutil, see obsutil in the *Object Storage Service Tools Guide*. 