ModelArts

Service Overview

Issue 01

Date 2024-06-15

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Infographics

1.1 What Is ModelArts


What Is ModelArts?
A fast inclusive AI development platform
ModelArts is a one-stop AI development platform that enables developers and data scientists of any skill level to rapidly build, train, and deploy models anywhere, from the cloud to the edge, and manage full-lifecycle AI workflows. ModelArts accelerates AI development and fosters AI innovation with key capabilities, including data preprocessing and auto labeling, distributed training, automated model building, and one-click workflow executing.
What are the biggest challenges in AI development?
Increasing data volume
Expensive and scarce acceleration resources
Increasingly time-consuming computing
Numerous tools Extensive learning period
More complex models
Extensive training duration
Insufficient resources
Advantages
1
100 times more efficient data preparation
Hard example sets
Videos
Image sets
Train
Label
Time required to process 40 TB of data: 8000 person-day > 80 person-day
2
50% faster model training
Algorithm optimization Fast
Hyperparameter tuning Simplified
Training acceleration rate for a cluster with 1000 cards: 0.8
03
One-click model deployment to the cloud, edge devices, and other devices
AI model deployment
Edge inference
Real-time inference
Batch inference
04
ExeML accelerated AI development
UI wizard
Adaptive training
05
Full-lifecycle management
Automatic, visualized development process
Resumable training
Comparable training results
06
Shared AI resources
Enterprise-wide sharing
AI sharing platform
External market
Improved efficiency
Data
Model
Applications
Open ecosystem
Application Scenarios
Video analysis
Image recognition
Product recommendation
Anomaly detection
Speech recognition

2 What Is ModelArts?

ModelArts is a one-stop AI development platform geared toward developers and data scientists of all skill levels. It enables you to rapidly build, train, and deploy models anywhere (from the cloud to the edge), and manage full-lifecycle AI workflows. ModelArts accelerates AI development and fosters AI innovation with key capabilities, including data preprocessing and auto labeling, distributed training, automated model building, and one-click workflow execution.

ModelArts covers all stages of AI development, including data processing, algorithm development, and model training and deployment. The underlying technologies of ModelArts support various heterogeneous computing resources, allowing developers to flexibly select and use resources. In addition, ModelArts supports popular open-source AI development frameworks such as TensorFlow, PyTorch, and MindSpore. ModelArts also allows you to use customized algorithm frameworks tailored to your needs.

ModelArts aims to simplify AI development.

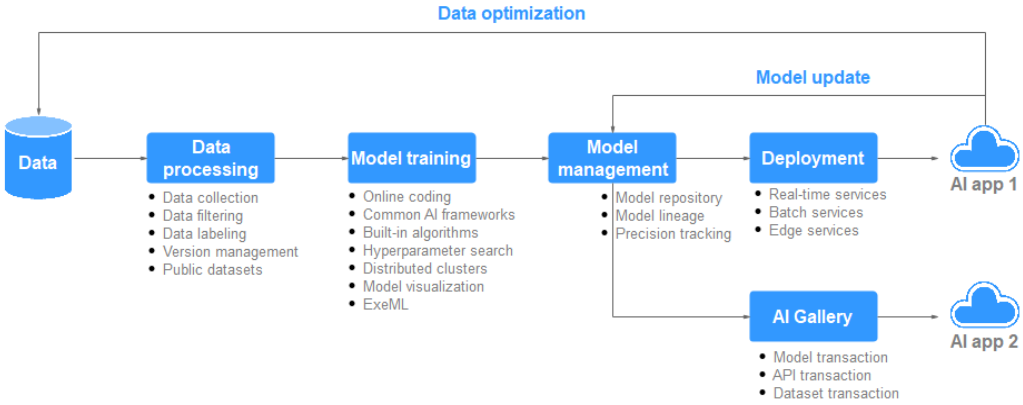
ModelArts is suitable for AI developers with varying levels of development experience. Service developers can use ExeML to quickly build AI applications without coding. Beginners can directly use built-in algorithms to build AI applications. AI engineers can use multiple development environments to quickly compile code for modeling and application development.

Product Architecture

ModelArts supports the entire development process, including data processing, and model training, management, and deployment. It also provides AI Gallery for sharing models.

ModelArts supports various AI application scenarios, such as image classification, object detection, video analysis, speech recognition, product recommendation, and exception detection.

Figure 2-1 ModelArts Standard architecture



Product Advantages

- **One-stop platform**

  The out-of-the-box and full-lifecycle AI development platform provides one-stop data processing, and development, training, management, and deployment of models.

- **Easy to use**
  - Multiple built-in models provided and free use of open-source models
  - Automatic optimization of hyperparameters
  - Code-free development and simplified operations
  - One-click deployment of models to the cloud, edge, and devices
- **High performance**
  - Resource utilization is optimized for accelerating real-time inference.
  - Models running on Ascend AI chips achieve more efficient inference.
- **Flexible**
  - Mainstream open-source frameworks such as TensorFlow, PyTorch, and MindSpore
  - Exclusive use of dedicated resources
  - Custom images for custom frameworks and operators

Using ModelArts for the First Time

If you are a first-time user, the following information will help you get familiar with ModelArts:

- **Basic concepts**

  Basic Knowledge describes the basic concepts of ModelArts, including the basic process and concepts of AI development, and specific concepts and functions of ModelArts.

- **Getting started**

  Getting Started provides samples with detailed operations, helping you get started with ModelArts Standard.

- **Best practices**

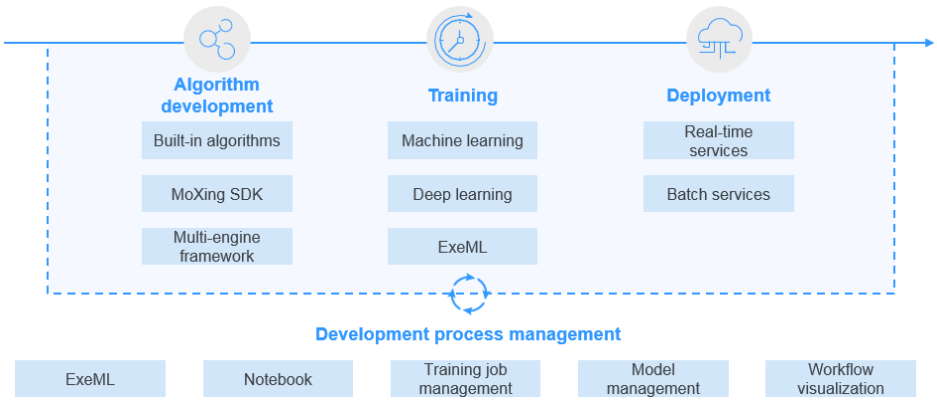
ModelArts supports multiple open-source engines and provides extensive use cases based on the engines and functions. You can build and deploy models by referring to **Best Practices**.

- **Other functions and operation guides**
  - If you are a service developer, you can use ExeML to quickly build models without coding. For details, see ***User Guide (ExeML)***.
  - If you are an AI engineer, you can use one or more functions in your AI development, including **DevEnviron**, **data preparation**, **data labeling**, **model development**, and **inference**. You can use one or more functions in your AI development.
  - If you want to use ModelArts APIs or SDKs for AI development, see ***API Reference*** or ***SDK Reference***.

3 Functions

AI engineers face challenges in the installation and configuration of various AI tools, data preparation, and model training. To address these challenges, the one-stop AI development platform ModelArts is provided. The platform integrates data preparation, algorithm development, model training, and model deployment into the production environment, allowing AI engineers to perform one-stop AI development.

Figure 3-1 Function overview



ModelArts has the following features:

- **Data governance**

  Manages data preparation, such as data filtering and labeling, and dataset versions.

- **Rapid and simplified model training**

  Enables high-performance distributed training and simplifies coding with the self-developed MoXing deep learning framework.

- **Cloud-edge-device synergy**

  Deploys models in various production environments such as devices, the edge, and the cloud, and supports real-time and batch inference.

- **Auto learning**

Enables model building without coding and supports image classification, object detection, and predictive analytics.

4 Basic Knowledge

4.1 Introduction to the AI Development Lifecycle

What Is AI

Artificial intelligence (AI) is a technology capable of simulating human cognition through machines. The core capability of AI is to make a judgment or prediction based on a given input.

What Is the Purpose of AI Development

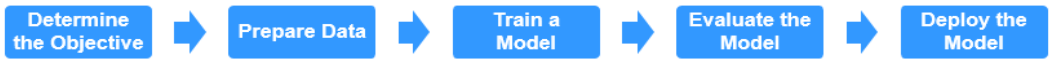
AI development aims to centrally process and extract information from volumes of data to summarize internal patterns of the study objects.

Massive volumes of collected data are computed, analyzed, summarized, and organized by using appropriate statistics, machine learning, and deep learning methods to maximize data value.

Basic Process of AI Development

The basic process of AI development includes the following steps: determining an objective, preparing data, and training, evaluating, and deploying a model.

Figure 4-1 AI development process



Step 1 Determine an objective.

Before starting AI development, determine what to analyze. What problems do you want to solve? What is the business goal? Sort out the AI development framework and ideas based on the business understanding. For example, image classification and object detection. Different projects have different requirements for data and AI development methods.

Step 2 Prepare data.

Data preparation refers to data collection and preprocessing.

Data preparation is the basis of AI development. When you collect and integrate related data based on the determined objective, the most important thing is to ensure the authenticity and reliability of the obtained data. Typically, you cannot collect all the data at the same time. In the data labeling phase, you may find that some data sources are missing and then you may need to repeatedly adjust and optimize the data.

Step 3 Train a model.

Modeling involves analyzing the prepared data to find the causality, internal relationships, and regular patterns, thereby providing references for commercial decision making. After model training, usually one or more machine learning or deep learning models are generated. These models can be applied to new data to obtain predictions and evaluation results.

A large number of developers develop and train models required by relevant services based on popular AI engines, such as TensorFlow, Spark_MLlib, MXNet, Caffe, PyTorch, XGBoost-Sklearn, and MindSpore.

Step 4 Evaluate the model.

A model generated by training needs to be evaluated. Typically, you cannot obtain a satisfactory model after the first evaluation, and may need to repeatedly adjust algorithm parameters and data to further optimize the model.

Some common metrics, such as the accuracy, recall, and area under the curve (AUC), help you effectively evaluate and obtain a satisfactory model.

Step 5 Deploy the model.

Model development and training are based on existing data (which may be test data). After a satisfactory model is obtained, the model needs to be formally applied to actual data or newly generated data for prediction, evaluation, and visualization. The findings can then be reported to decision makers in an intuitive way, helping them develop the right business strategies.

----End

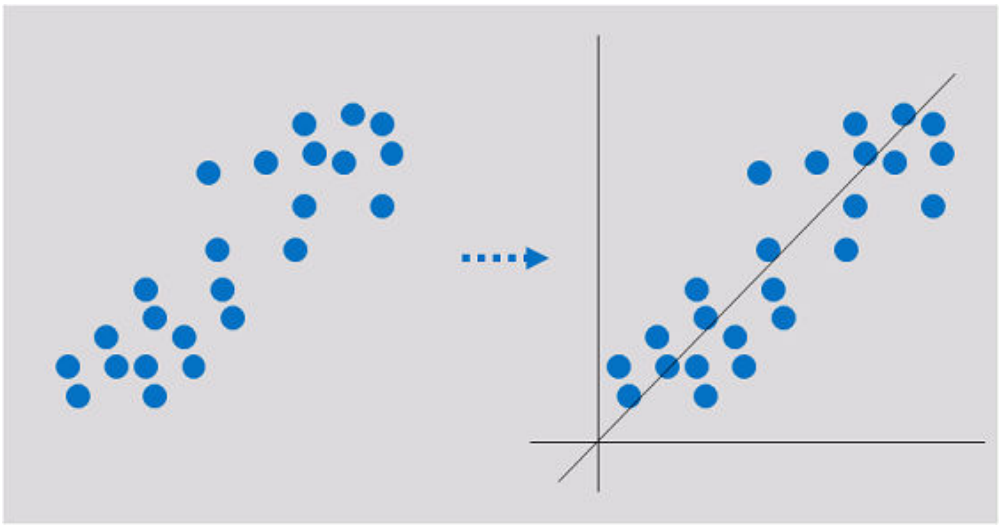
4.2 Basic Concepts of AI Development

Machine learning is classified into supervised, unsupervised, and reinforcement learning.

- Supervised learning uses labeled samples to adjust the parameters of classifiers to achieve the required performance. It can be considered as learning with a teacher. Common supervised learning includes regression and classification.
- Unsupervised learning is used to find hidden structures in unlabeled data. Clustering is a form of unsupervised learning.
- Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.

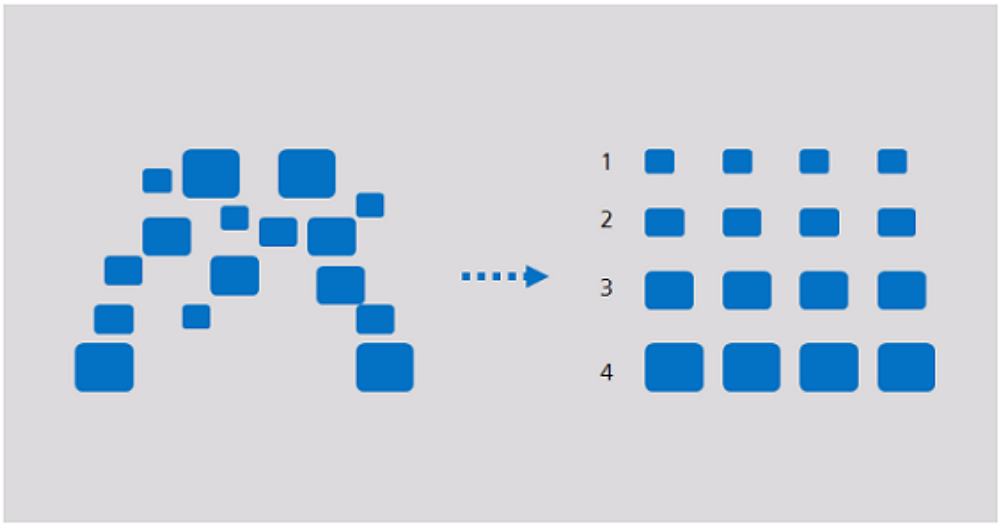
Regression

Regression reflects the time feature of data attributes and generates a function that maps one data attribute to an actual variable prediction to find the dependency between the variable and attribute. Regression mainly analyzes data and predicts data and data relationship. Regression can be used for customer development, retention, customer churn prevention, production lifecycle analysis, sales trend prediction, and targeted promotion.



Classification

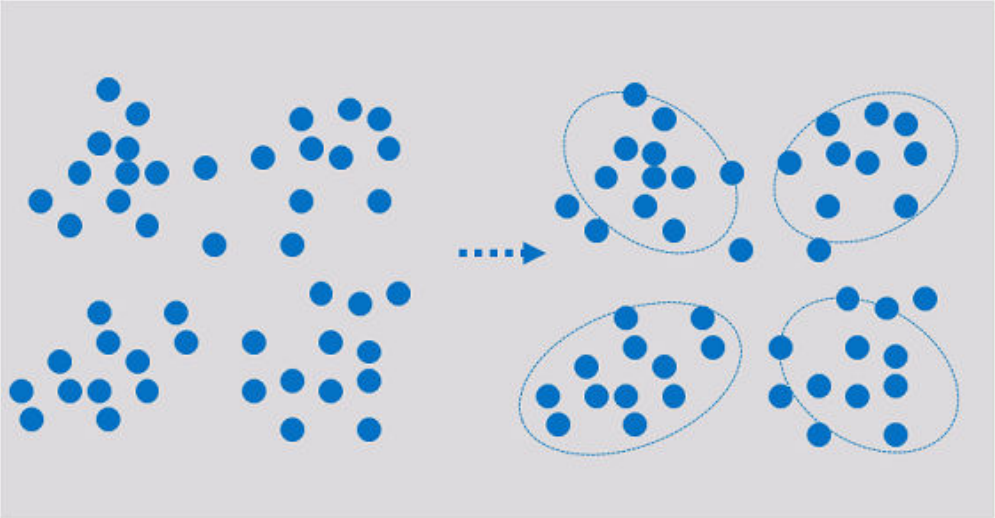
Classification involves defining a set of categories based on the common features of objects and identifying which category an object belongs to. Classification can be used for customer classification, customer properties, feature analysis, customer satisfaction analysis, and customer purchase trend prediction.



Clustering

Clustering involves grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups.

Clustering can be used for customer segmentation, customer characteristic analysis, customer purchase trend prediction, and market segmentation.



Clustering analyzes data objects and produces class labels. Objects are grouped based on the maximized and minimized similarities to form clusters. In this way, objects in the same cluster are more similar to each other than to those in other clusters.

4.3 Common Concepts of ModelArts

ExeML

ExeML is the process of automating model design, parameter tuning, and model training, model compression, and model deployment with the labeled data. The process is code-free and does not require developers to have experience in model development. A model can be built in three steps: labeling data, training a model, and deploying the model.

Device-Edge-Cloud

Device-Edge-Cloud indicates devices, intelligent edge nodes, and the public cloud.

Inference

Inference is the process of deriving a new judgment from a known judgment according to a certain strategy. In AI, machines simulate human intelligence, and complete inference based on neural networks.

Real-Time Inference

Real-time inference specifies a web service that provides an inference result for each inference request.

Batch Inference

Batch inference specifies a batch job that processes batch data for inference.

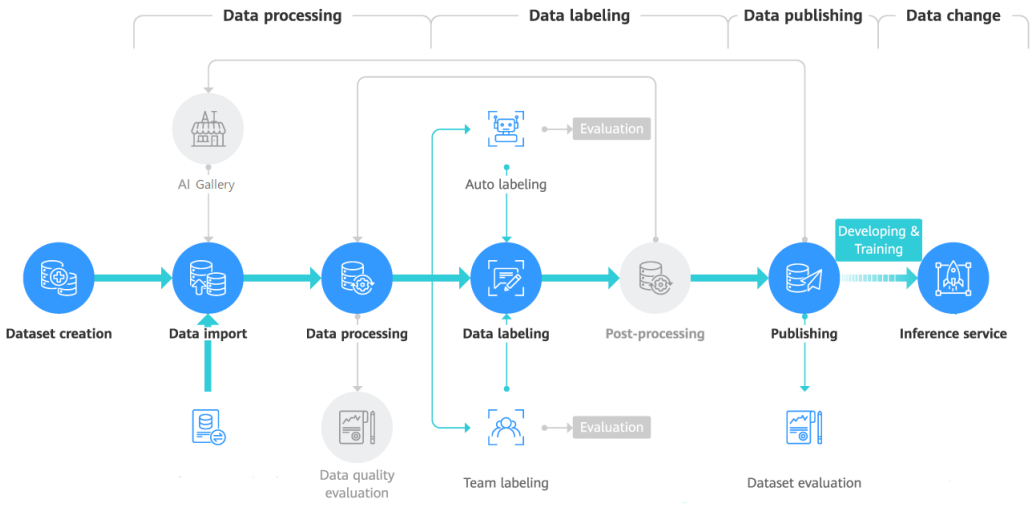
Ascend Chip

The Ascend chips are a series of Huawei-developed AI chips with high computing performance and low power consumption.

4.4 Data Management

During AI development, massive volumes of data need to be processed, and data preparing and labeling usually take more than half of the time required for the entire development process. ModelArts data management provides an efficient data management and labeling framework. It supports image, text, audio, and video data types in a range of labeling scenarios such as image classification, object detection, speech paragraph labeling, and text classification so that data management can be used in various AI projects such as computer vision, natural language processing, and audio and video analysis projects. In addition, ModelArts data management provides functions such as data filtering, data analysis, data processing, team labeling, and version management, enabling you to manage the full data labeling process. **Figure 4-2** shows the data labeling process.

Figure 4-2 Data labeling process



ModelArts data management analyzes and processes data using such functions as clustering analysis, data feature analysis, data cleansing, data verification, data augmentation, and data selection, helping you obtain high-value data that meets development or project requirements.

With data management, ModelArts allows you to label data online for image classification, object detection, speech paragraphs, text triplet, and videos. You can also use intelligent labeling to automatically label data through built-in or customized algorithms, improving the labeling efficiency.

To support large-scale collaborative labeling, data management provides team labeling with team management, personnel management, and data management for full-process project management, from project creation, data allocation, progress control, labeling, review, to acceptance. This improves labeling efficiency and minimizes project management costs.

ModelArts data management ensures the security and privacy of user data and allows data to be used only within the authorized scope.

In the new version of data management, datasets and data labeling are decoupled to facilitate your operations.

4.5 Introduction to Development Tools

NOTE

This document describes the DevEnviron notebook functions of the new version.

Software development is a process of reducing developer costs and improving development experience. In AI development, ModelArts is dedicated to improving AI development experience and simplifying the development process. ModelArts DevEnviron uses cloud native resources and integrates the development tool chain to provide better in-cloud AI development experience for AI development, exploration, and teaching.

ModelArts notebook for seamless in-cloud and on-premises collaboration

- In-cloud JupyterLab, local IDE, and ModelArts plug-ins for remote development and debugging, tailored to your needs
- In-cloud development environment with AI compute resources, cloud storage, and built-in AI engines
- Custom runtime environment saved as an image for training and inference

Feature 1: Remote development, allowing remote access to notebook from a local IDE

The notebook of the new version provides remote development. After enabling remote SSH, you can remotely access the ModelArts notebook development environment to debug and run code from a local IDE.

Due to limited local resources, developers using a local IDE run and debug code typically on a CPU or GPU server shared between team members. Building and maintaining the CPU or GPU server are costly.

ModelArts notebook instances are out of the box with various built-in engines and flavors for you to select. You can use a dedicated container environment. Only after simple configurations, you can remotely access the environment to run and debug code from your local IDE.

ModelArts notebook can be regarded as an extension of a local development environment. The operations such as data reading, training, and file saving are the same as those performed in a local environment.

ModelArts notebook allows you to use in-cloud resources while with local coding habits unchanged.

A local IDE supports Visual Studio (VS) Code, PyCharm, and SSH.

Feature 2: Preset images that are out-of-the-box with optimized configurations and supporting mainstream AI engines

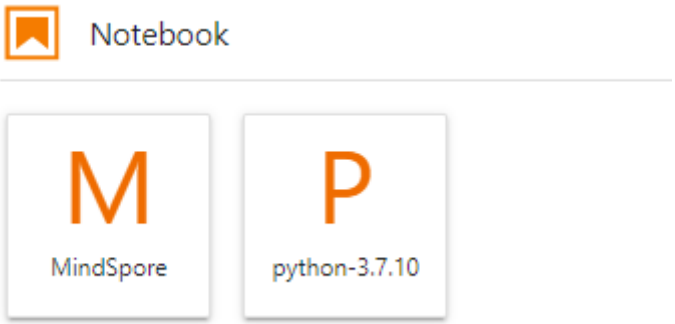
The AI engines and versions preset in each image are fixed. When creating a notebook instance, specify an AI engine and version, including the chip type.

ModelArts DevEnviron provides a group of preset images, including PyTorch, TensorFlow, and MindSpore images. You can use a preset image to start your notebook instance. After the development in the instance, submit a training job without any adaptation.

The image versions preset in ModelArts are determined based on user feedback and version stability. If your development can be carried out using the versions preset in ModelArts, for example, MindSpore 1.5, use preset images. These images have been fully verified and have many commonly-used installation packages built in. They are out-of-the-box, relieving you from configuring the environment.

The images preset in ModelArts DevEnviron include:

- Common preset packages: common AI engines such as PyTorch and MindSpore based on standard Conda, common data analysis software packages such as Pandas and Numpy, and common tool software such as CUDA and CUDNN, meeting common AI development requirements.
- Preset Conda environments: A Conda environment and basic Conda Python (excluding any AI engine) are created for each preset image. The following figure shows the Conda environment for the preset MindSpore.

  Select a Conda environment based on whether the AI engine is used for debugging.
- Notebook: a web application that enables you to code on the GUI and combine the code, mathematical equations, and visualized content into a document.
- JupyterLab plug-ins: enable flavor changing, case sharing to AI Gallery for communication, and instance stopping to improving user experience.
- Remote SSH: allows you to remotely debug a notebook instance from a local PC.
- After the images preset in ModelArts DevEnviron support development, the training jobs can be executed on ModelArts.

NOTE

- To simplify operations, ModelArts notebook of the new version does not support switchover between AI engines in a notebook instance.
- AI engines vary based on regions. For details about the AI engines available in a region, see the AI engines displayed on the management console.

Feature 3: JupyterLab, an online interactive development and debugging tool

ModelArts integrates open-source JupyterLab for online interactive development and debugging. You can use the notebook on the ModelArts management console to compile and debug code and train models based on the code, without concerning environment installation or configuration.

JupyterLab is an interactive development environment. It is the next-generation product of Jupyter Notebook. JupyterLab enables you to compile notebooks, operate terminals, edit Markdown text, enable interaction, and view CSV files and images.

4.6 Model Training

In addition to data and algorithms, developers spend a lot of time configuring model training parameters. Model training parameters determine the model's precision and convergence time. Parameter selection is heavily dependent on developers' experience. Improper parameter selection will affect the model's precision or significantly increase the time required for model training.

To simplify AI development and improve development efficiency and training performance, ModelArts offers visualized job management, resource management, and version management and automatically performs hyperparameter optimization based on machine learning and reinforcement learning. It provides automatic hyperparameter tuning policies such as learning rate and batch size, and integrates common models.

Currently, when most developers build models, the models usually have dozens of layers or even hundreds of layers and MB-level or GB-level parameters to meet precision requirements. As a result, the specifications of computing resources are extremely high, especially the computing power of hardware resources, memory, and ROM. The resource specifications on the device side are strictly limited. For example, the computing power on the device side is 1 TFLOPS, the memory size is about 2 GB, and the ROM space is about 2 GB, so the model size on the device side must be limited to 100 KB and the inference delay must be limited to 100 milliseconds.

Therefore, compression technologies with lossless or near-lossless model precision, such as pruning, quantization, and knowledge distillation, are used to implement automatic model compression and optimization, and automatic iteration of model compression and retraining to control the loss of model precision. The low-bit quantization technology, which eliminates the need for retraining, converts the model from a high-precision floating point to a fixed-point operation. Multiple compression and optimization technologies are used to meet the lightweight requirements of device and edge hardware resources. The model compression technology reduces the precision by less than 1% in specific scenarios.

When the training data volume is large, the training of the deep learning model is time-consuming. The acceleration of deep learning training has always been an important concern to the academia and the industry.

Distributed training acceleration needs to be considered in terms of software and hardware. A single optimization method cannot meet expectations. Therefore, optimization of distributed acceleration is a system project. The distributed training architecture needs to be considered in terms of hardware and chip design. To minimize compute and communication delays, many factors need to be considered, including overall compute specifications, network bandwidth, high-speed cache, power consumption, and heat dissipation of the system, and the relationship between compute and communication throughput.

The software design needs to combine high-performance hardware features to fully use the high-speed hardware network and implement high-bandwidth distributed communication and efficient local data caching. By using training optimization algorithms, such as hybrid parallel, gradient compression, and convolution acceleration, the software and hardware of the distributed training system can be efficiently coordinated and optimized from end to end, and training acceleration can be implemented in a distributed environment of multiple hosts and cards. ModelArts delivers an industry-leading speedup of over 0.8 for ResNet50 on the ImageNet dataset in the distributed environment with thousands of hosts and cards.

To measure the acceleration performance of distributed deep learning, the following two key indicators are used:

- Throughput, that is, the amount of data processed in a unit time
- Convergence time, that is, the time required to achieve certain precision

The throughput depends on server hardware (for example, more AI acceleration chips with higher FLOPS processing capabilities and higher communication bandwidth achieve higher throughput), data reading and caching, data preprocessing, model computing (for example, convolution algorithm selection), and communication topology optimization. Except low-bit computing and gradient (or parameter) compression, most technologies improve throughput without affecting model precision. To achieve the shortest convergence time, you need to optimize the throughput and adjust the parameters. If the parameters are not adjusted properly, the throughput cannot be optimized. If the batch size is set to a small value, the parallel performance of model training will be relatively poor. As a result, the throughput cannot be improved even if the number of compute nodes are increased.

Users are most concerned about convergence time. The MoXing framework implements full-stack optimization and significantly reduces the training convergence time. For data read and preprocessing, MoXing uses multi-level concurrent input pipelines to prevent data I/Os from becoming a bottleneck. In terms of model computing, MoXing provides hybrid precision calculation, which combines semi-precision and single-precision for the upper layer models and reduces the loss caused by precision calculation through adaptive scaling. Dynamic hyperparameter policies (such as momentum and batch size) are used to minimize the number of epochs required for model convergence. MoXing also works with underlying Huawei servers and computing libraries to further improve distributed acceleration.

ModelArts High-Performance Distributed Training Optimization

- Automatic hybrid precision to fully utilize hardware computing capabilities
- Dynamic hyperparameter adjustment technologies (dynamic batch size, image size, and momentum)
- Automatic model gradient merging and splitting
- Communication operator scheduling optimization based on BP bubble adaptive computing
- Distributed high-performance communication libraries (NStack and HCCL)
- Distributed data-model hybrid parallel
- Training data compression and multi-level caching

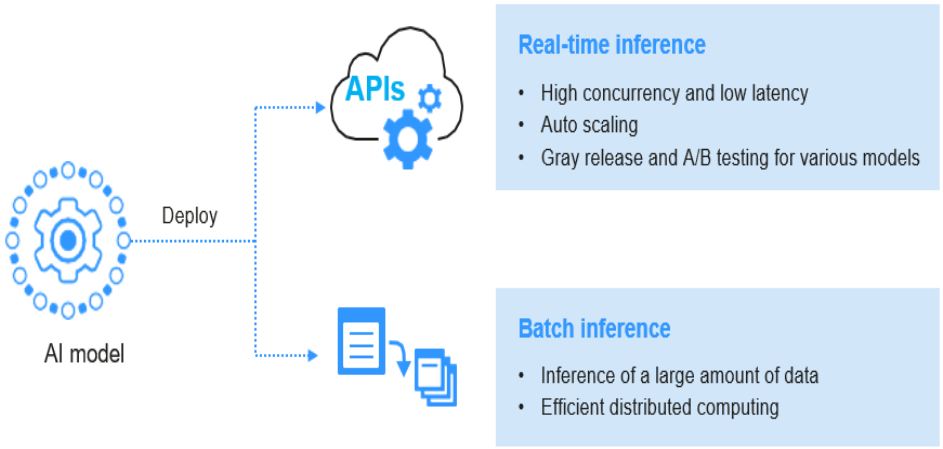
4.7 Model Deployment

ModelArts is capable of managing models and services. This allows mainstream framework images and models from multiple vendors to be managed in a unified manner.

Generally, AI model deployment and large-scale implementation are complex.

For example, in a smart transportation project, the trained model needs to be deployed to the cloud, edges, and devices. It takes time and effort to deploy the model on the devices, for example, deploying a model on cameras of different specifications and vendors. ModelArts supports one-click deployment of a trained model on various devices for different application scenarios. In addition, it provides a set of secure and reliable one-stop deployment modes for individual developers, enterprises, and device manufacturers.

Figure 4-3 Process of deploying a model



- The real-time inference service features high concurrency, low latency, and elastic scaling, and supports multi-model gray release and A/B testing.
- Models can be deployed as real-time inference services and batch inference tasks on the cloud.

5 Which AI Frameworks Does ModelArts Support?

The AI frameworks and versions supported by ModelArts vary slightly based on the development environment notebook, training jobs, and model inference (AI application management and deployment). The following describes the AI frameworks supported by each module.

Development Environment Notebook

The image and versions supported by development environment notebook instances vary based on runtime environments.

Table 5-1 Images supported by notebook of the new version

Image	Description	Supported Chip	Remote SSH	Online Jupyter Lab
pytorch1.8-cuda10.2-cudnn7-ubuntu18.04	CPU- or GPU-powered public image for general algorithm development and training, with built-in AI engine PyTorch 1.8	CPU or GPU	Yes	Yes
mindspore1.7.0-cuda10.1-py3.7-ubuntu18.04	CPU- or GPU-powered general algorithm development and training, preconfigured with AI engine MindSpore 1.7.0 and CUDA 10.1	CPU or GPU	Yes	Yes

Image	Description	Supported Chip	Remote SSH	Online Jupyter Lab
mindspore1.7.0-py3.7-ubuntu18.04	CPU-powered general algorithm development and training, preconfigured with AI engine MindSpore 1.7.0	CPU	Yes	Yes
pytorch1.10-cuda10.2-cudnn7-ubuntu18.04	CPU- or GPU-powered general algorithm development and training, preconfigured with AI engine PyTorch 1.10 and CUDA 10.2	CPU or GPU	Yes	Yes
tensorflow2.1-cuda10.1-cudnn7-ubuntu18.04	CPU- or GPU-powered public image for general algorithm development and training, with built-in AI engine TensorFlow 2.1	CPU or GPU	Yes	Yes
conda3-ubuntu18.04	Clean customized base image only includes Conda	CPU	Yes	Yes
pytorch1.4-cuda10.1-cudnn7-ubuntu18.04	CPU- or GPU-powered public image for general algorithm development and training, with built-in AI engine PyTorch 1.4	CPU or GPU	Yes	Yes
tensorflow1.13-cuda10.0-cudnn7-ubuntu18.04	GPU-powered public image for general algorithm development and training, with built-in AI engine TensorFlow 1.13.1	GPU	Yes	Yes

Image	Description	Supported Chip	Remote SSH	Online Jupyter Lab
conda3-cuda10.2-cudnn7-ubuntu18.04	Clean customized base image includes CUDA 10.2, Conda	CPU	Yes	Yes
spark2.4.5-ubuntu18.04	CPU-powered algorithm development and training, preconfigured with PySpark 2.4.5 and can be attached to preconfigured Spark clusters including MRS and DLI	CPU	No	Yes
mindspore1.2.0-cuda10.1-cudnn7-ubuntu18.04	GPU-powered public image for algorithm development and training, with built-in AI engine MindSpore-GPU	GPU	Yes	Yes
mindspore1.2.0-openmpi2.1.1-ubuntu18.04	CPU-powered public image for algorithm development and training, with built-in AI engine MindSpore-CPU	CPU	Yes	Yes

Training Jobs

The following table lists the AI engines.

The built-in training engines are named in the following format:

```
<Training engine name_version>-[cpu | <cuda_version | cann_version >]-<py_version>-<OS name_version>-<x86_64 | aarch64>
```


Table 5-2 AI engines supported by training jobs

Runtime Environment	System Architecture	System Version	AI Engine and Version	Supported CUDA or Ascend Version
TensorFlow	x86_64	Ubuntu18.04	tensorflow_2.1.0-cuda_10.1-py_3.7-ubuntu_18.04-x86_64	cuda10.1
PyTorch	x86_64	Ubuntu18.04	pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64	cuda10.2
MPI	x86_64	Ubuntu18.04	mindspore_1.3.0-cuda_10.1-py_3.7-ubuntu_1804-x86_64	cuda_10.1
Horovod	x86_64	ubuntu_18.04	horovod_0.20.0-tensorflow_2.1.0-cuda_10.1-py_3.7-ubuntu_18.04-x86_64	cuda_10.1
			horovod_0.22.1-pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64	cuda_10.2

NOTE

Supported AI engines vary depending on regions.

Supported AI Engines for ModelArts Inference

If you import a model from a template or OBS to create an AI application, the following AI engines and versions are supported.

NOTE

- Runtime environments marked with **recommended** are unified runtime images, which will be used as mainstream base inference images. The installation packages of unified images are richer. For details, see **Base Inference Images**.
- Images of the old version will be discontinued. Use unified images.
- The base images to be removed are no longer maintained.
- Naming a unified runtime image: <*AI engine name and version*> - <*Hardware and version*: CPU, CUDA, or CANN> - <*Python version*> - <*OS version*> - <*CPU architecture*>

Table 5-3 Supported AI engines and their runtime

Engine	Runtime	Note
TensorFlow	python3.6 python2.7 (unavailable soon) tf1.13-python3.6-gpu tf1.13-python3.6-cpu tf1.13-python3.7-cpu tf1.13-python3.7-gpu tf2.1-python3.7 (unavailable soon) tensorflow_2.1.0-cuda_10.1-py_3.7-ubuntu_18.04-x86_64 (recommended)	• TensorFlow 1.8.0 is used in **python2.7** and **python3.6**. • **python3.6**, **python2.7**, and **tf2.1-python3.7** indicate that the model can run on both CPUs and GPUs. For other runtime values, if the suffix contains **cpu** or **gpu**, the model can run only on CPUs or GPUs. • The default runtime is **python2.7**.
Spark_MLlib	python2.7 (unavailable soon) python3.6 (unavailable soon)	• Spark_MLlib 2.3.2 is used in **python2.7** and **python3.6**. • The default runtime is **python2.7**. • **python2.7** and **python3.6** can only be used to run models on CPUs.
Scikit_Learn	python2.7 (unavailable soon) python3.6 (unavailable soon)	• Scikit_Learn 0.18.1 is used in **python2.7** and **python3.6**. • The default runtime is **python2.7**. • **python2.7** and **python3.6** can only be used to run models on CPUs.
XGBoost	python2.7 (unavailable soon) python3.6 (unavailable soon)	• XGBoost 0.80 is used in **python2.7** and **python3.6**. • The default runtime is **python2.7**. • **python2.7** and **python3.6** can only be used to run models on CPUs.

Engine	Runtime	Note
PyTorch	python2.7 (unavailable soon) python3.6 python3.7 pytorch1.4-python3.7 pytorch1.5-python3.7 (unavailable soon) pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64 (recommended)	• PyTorch 1.0 is used in **python2.7**, **python3.6**, and **python3.7**. • **python2.7**, **python3.6**, **python3.7**, **pytorch1.4-python3.7**, and **pytorch1.5-python3.7** indicate that the model can run on both CPUs and GPUs. • The default runtime is **python2.7**.
MindSpore	aarch64 (recommended)	AArch64 can run only on Snt3 chips.

6 Related Services

IAM

ModelArts uses Identity and Access Management (IAM) for authentication and authorization. For more information about IAM, see ***Identity and Access Management User Guide***.

OBS

ModelArts uses Object Storage Service (OBS) to securely and reliably store data and models at low costs. For more details, see ***Object Storage Service Console Operation Guide***.

Table 6-1 Relationship between ModelArts and OBS

Function	Sub Task	Relationship
ExeML	Data labeling	The data labeled on ModelArts is stored in OBS.
	Auto training	After a training job is completed, the generated model is stored in OBS.
	Model deployment	ModelArts deploys models stored in OBS as real-time services.
AI development lifecycle	Data management	• Datasets are stored in OBS. • The dataset labeling information is stored in OBS. • Data can be imported from OBS.
	Development environment	Data or code files in a notebook instance are stored in OBS.

Function	Sub Task	Relationship
	Model training	• The datasets used by training jobs are stored in OBS. • The running scripts for training jobs are stored in OBS. • The models generated by training jobs are stored in the specified OBS paths. • The run logs of training jobs are stored in the specified OBS paths.
	AI application management	After a training job is completed, the generated model is stored in OBS. You can import the model from OBS.
	Service deployment	The models stored in OBS can be deployed as services.
Settings	-	Authorizes ModelArts to access OBS (using an agency or access key) so that ModelArts can use OBS to store data and create notebook instances.

EVS

ModelArts uses Elastic Volume Service (EVS) to store created notebook instances. For more details, see ***Elastic Volume Service User Guide***.

CCE

ModelArts uses Cloud Container Engine (CCE) to deploy models as real-time services. CCE enables high concurrency and provides elastic scaling. For more information about CCE, see ***Cloud Container Engine User Guide***.

SWR

To use an AI framework that is not supported by ModelArts, use Software Repository for Container (SWR) to customize an image and import the image to ModelArts for training or inference. For details about SWR, see ***Software Repository for Container User Guide***.

Cloud Eye

ModelArts uses Cloud Eye to monitor online services and model loads in real time and send alarms and notifications automatically. For details about Cloud Eye, see ***Cloud Eye User Guide***.

7 How Do I Access ModelArts?

You can access ModelArts through the web-based management console or by using HTTPS-based application programming interfaces (APIs).

- **Using the Management Console**

  ModelArts features a simple and easy-to-use management console, and provides a host of functions including ExeML, data management, development environment, model training, AI application management, AI Gallery, and service deployment. You can complete end-to-end AI development on the management console.

  To use the ModelArts management console, you need to register with HUAWEI CLOUD first. If you have created a Huawei Cloud account, choose **AI** > **ModelArts** on the official website and log in to the management console.

- **Using SDKs**

  If you want to integrate ModelArts into a third-party system for secondary development, call SDKs to complete the development. ModelArts SDKs encapsulate RESTful APIs provided by ModelArts to simplify secondary development. For details about the SDKs and operations, see ***ModelArts SDK Reference***.

  In addition, you can directly call the ModelArts SDKs when writing code in a notebook on the management console.

- **Using APIs**

  If you want to integrate ModelArts into a third-party system for secondary development, use APIs to access ModelArts. For details about the APIs and operations, see ***ModelArts API Reference***.

8 Billing Description

8.1 Overview

ModelArts is a one-stop AI development platform geared toward developers and data scientists of all skill levels. It enables you to rapidly build, train, and deploy models anywhere (from the cloud to the edge), and manage full-lifecycle AI workflows. ModelArts accelerates AI development and fosters AI innovation with key capabilities, including data preprocessing and auto labeling, distributed training, automated model building, and one-click workflow execution.

ModelArts can be billed either pay-as-you-go or on a more economical yearly/monthly basis. For more details, see **Product Pricing Details**.

After reading this section, you can quickly learn about ModelArts billing information, such as **billing items** and **billing modes**.

8.2 Billing Items

ModelArts billing items during AI development include storage resources and compute resources.

- **Storage resources**: Fees are generated for the used OBS storage and EVS storage (applicable only to notebook instances).
- **Compute resources**: Fees are generated for the used ModelArts compute resources.

Storage Resources

Table 8-1 Billing for storage resources

Billing Item	Description
OBS	ModelArts uses OBS to store data and models, which incurs fees. For details, see **OBS Pricing Details**.

Compute Resources

ModelArts allows you to select proper compute resources for AI development based on your region and required service type. For details, see **ModelArts Pricing Details**.

For details about how to use compute resources for ModelArts, see **Pay-Per-Use**.

Table 8-2 Billing for compute resources

Billing Item	Description
AI development lifecycle	Machine learning and deep learning algorithm development and deployment for developers with AI development experience, including data processing, model development, model training, as well as AI application management and deployment. The following items are billed: • Development environment (notebook instances) • Model training (training jobs) • Service deployment (real-time services)
ExeML	Automatic design, tuning, and training of models, as well as service deployment based on labeled data for developers with few AI development experience, enabling code-free AI development. Resources are billed for ExeML-powered job training and AI application deployment. The following items are billed: • ExeML-powered training jobs • ExeML-powered AI application deployment **NOTE** Only pay-per-use billing is supported.

8.3 Billing Modes

The compute resources used in ModelArts can be billed on a pay-per-use or yearly/monthly basis.

- **Pay-per-use**: allows you to make a subscription or unsubscription at any time. This billing mode can be used when you select resources for creating a development environment, creating a training job, or deploying a model as a service.
- **Yearly/Monthly**: ModelArts allows you to purchase resources on a yearly or monthly basis. This mode provides a larger discount than pay-per-use.

NOTE

- Public resource pools can only be billed in pay-per-use mode.
- Only dedicated resource pools can be billed on a yearly or monthly basis. The dedicated resource pool functions and purchase methods vary depending on regions. For details, see the management console.

  To purchase a dedicated resource pool, log in to the ModelArts management console, click **Dedicated Resource Pools** in the navigation pane, and click **Create** on the **Dedicated Resource Pools** page. If **Dedicated Resource Pools** is unavailable on the ModelArts management console or the yearly/monthly billing mode is unavailable on the page for purchasing a dedicated resource pool, the current region does not support the yearly/monthly billing mode.

Table 8-3 Billing modes

Billing Mode	Yearly/Monthly	Pay-per-use
Payment Method	Prepaid Billed by the purchase period specified in your order	Postpaid Billed by usage duration of resources
Billing Period	Purchase duration specified in your order	Billed by the second. A bill is generated on the hour.
Changing a Billing Mode	Yearly/Monthly billing can be changed to pay-per-use, which is only supported by the old-version dedicated resource pools for development or training. The pay-per-use mode takes effect only after a yearly/monthly period expires. For details about how to change the billing mode from yearly/monthly to pay-per-use, see **Switching the Dedicated Resource Pool Billing Mode**.	Pay-per-use billing can be changed to yearly/monthly, which is only supported by the old-version dedicated resource pools for development or training. The billing mode can be changed to yearly/monthly only if there are pay-per-use consumption records. For details about how to change the billing mode from pay-per-use to yearly/monthly, see **Switching the Dedicated Resource Pool Billing Mode**.
Application Scenarios	This cost-effective mode is ideal when the duration of resource usage is predictable. This billing mode is recommended for long-term usage.	This mode applies if resource requirements fluctuate. You only need to pay for what you have used.

8.4 Modifying Configurations

When using ModelArts, you can select compute resources as required. ModelArts allows you to modify configurations after a job is started. If your requirements still

cannot be met after you use the methods provided by ModelArts to modify configurations, you can create a new job and migrate data to it.

Changing a Billing Mode

ModelArts allows you to change the billing mode of a dedicated resource pool. For details, see **Resource Pools**.

NOTE

The billing mode of only old-version dedicated resource pools for development or training can be changed between pay-per-use and yearly/monthly.

Restrictions on changing a billing mode are as follows:

- Pay-per-use can be changed to yearly/monthly only if there are pay-per-use consumption records. The yearly/monthly billing takes effect immediately after the change.
- Yearly/Monthly changed from pay-per-us takes effect only after the original yearly/monthly period has expired.

Resizing a Dedicated Resource Pool

ModelArts allows you to resize a running dedicated resource pool. For details, see **Adjusting the Capacity of a Resource Pool**.

NOTE

For details about how to resize an old-version dedicated resource pool, see **Resource Pools**. Restrictions on resizing a dedicated resource pool are as follows:

- The capacity of a dedicated resource pool billed on a yearly/monthly basis can only be increased. After the yearly/monthly period expires, its billing mode can be changed to pay-per-use.
- The capacity of a pay-per-use dedicated resource pool can be manually adjusted, and the adjusted resource pool will be billed based on the new number of nodes.

8.5 Renewal

ModelArts resources can be billed either on a pay-per-use or yearly/monthly basis. In pay-per-use billing, fees are deducted every hour. If the balance is insufficient, your account will be in arrears. After a yearly/monthly period expires, you will be automatically billed on a pay-per-use basis. Your services will not be interrupted as long as your account balance is sufficient. If your subscription is not renewed, your resources will enter a retention period, during which your services still run. After the retention period expires, your services will stop but data will be retained.

- The retention period varies depending on your level. For details, see **Service Suspension and Resource Release**.
- To renew your subscription, go to the **Renewals** page.

8.6 Expiration and Overdue Payment

The following describes arrears and expiration of ModelArts dedicated resource pools:

- Resources billed on a pay-per-use basis will not expire. Such resources are billed by the hour. If the balance is insufficient, your account will be in arrears. Then, resources will be frozen. Top up your account to unfreeze the resources. After your account is in arrears, the resources enter the grace period and then retention period.
- After a yearly/monthly resource expires, it enters the grace period and then retention period.

You can access and use the resource pool during the grace period. If you do not renew the resource pool within the grace period, the resource enters the retention period and the resource status changes to **Frozen**. You cannot perform any operation on the resource in the retention period. If the resource pool is not renewed after the retention period ends, the resource pool will be automatically deleted.

CAUTION

If resources are renewed within a retention period, pay for the period from the time the resources enter the retention period to the time the resources are renewed.

Outstanding Balance

If your account is in arrears, some operations will be restricted. Top up your account as soon as possible. **Table 8-4** describes the restricted operations.

Table 8-4 Restricted operations due to arrears

Function	Restricted Operation
Workflow	Workflow subscription, model training, and model deployment
ExeML	Model training and deployment
DevEnviron > Notebook	Creating and starting notebook instances
Training Management > Training Jobs	Creating training jobs
Service Deployment > Real-Time Services, Batch Services, or Edge Services	Deploying real-time, batch, or edge services
Dedicated Resource Pools	Creating dedicated resource pools

9 Permissions Management

ModelArts allows you to configure fine-grained permissions for refined management of resources and permissions. This is commonly used by large enterprises, but it is complex for individual users. It is recommended that individual users configure permissions for using ModelArts by referring to **Assigning Permissions to Individual Users for Using ModelArts**.

NOTE

If you meet any of the following conditions, read this document.

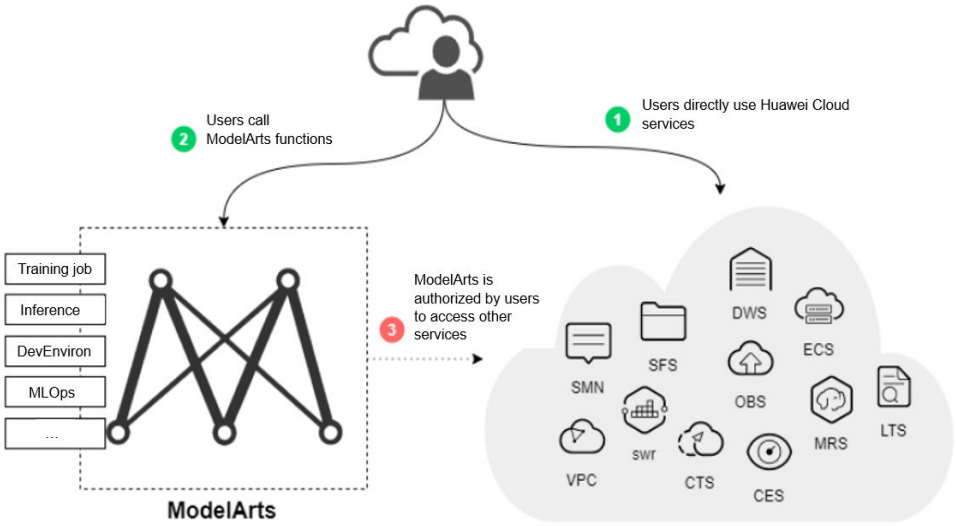
- You are an enterprise user, and
  - There are multiple departments in your enterprise, and you need to control users' permissions so that users in different departments can access only their dedicated resources and functions.
  - There are multiple roles in your enterprise, including administrators, algorithm developers, and application O&M engineers. You need them to use only specific functions.
  - Logically, there are multiple isolated environments, including the development environment, pre-production environment, and production environment. You need to control users' permissions on different environments.
  - You need to control permissions of specific IAM user or user group.
- You are an individual user, and you have created multiple IAM users. You need to assign different ModelArts permissions to different IAM users.
- You need to understand the concepts and operations of ModelArts permissions management.

ModelArts uses Identity and Access Management (IAM) for most permissions management functions. Before reading below, learn about **Basic Concepts**. This helps you better understand this document.

To implement fine-grained permissions management, ModelArts provides permission control, agency authorization, and workspace. The following describes the details.

ModelArts Permissions and Agencies

Figure 9-1 Permissions management



Exposed ModelArts functions are controlled through IAM permissions. For example, if you as an IAM user need to create a training job on ModelArts, you must have the **modelarts:trainJob:create** permission. For details about how to assign permissions to a user (you need to add the user to a user group and then assign permissions to the user group), see **Permissions Management**.

ModelArts must access other services for AI computing. For example, ModelArts must access OBS to read your data for training. For security purposes, ModelArts must be authorized to access other cloud services. This is agency authorization.

The following summarizes permissions management:

- Your access to any cloud service is controlled through IAM. You must have the permissions of the cloud service. (The required service permissions vary depending on the functions you use.)
- To use ModelArts functions, you need to grant permissions through IAM.
- ModelArts must be authorized by you to access other cloud services for AI computing.

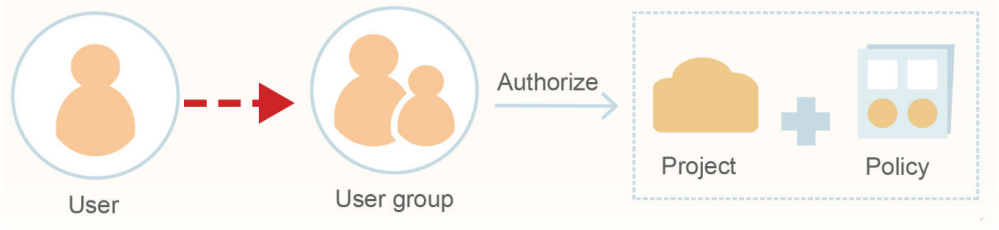
ModelArts Permissions Management

By default, new IAM users do not have any permissions assigned. You need to add the user to a user group and grant the user group with policies, so that the users in the group can inherit the permissions. After authorization, users can perform operations on ModelArts based on permissions.

CAUTION

ModelArts is a project-level service deployed and accessed in specific physical regions. When you authorize an agency, you can set the scope for the permissions you select to all resources, enterprises projects, or region-specific projects. If you specify region-specific projects, the selected permissions will be applied to resources in these projects.

For details, see **Creating a User Group and Assigning Permissions**.



When assigning permissions to a user group, IAM does not directly assign specific permissions to the user group. Instead, IAM needs to add the permissions to a policy and then assign the policy to the user group. To facilitate user permissions management, each cloud service provides some preset policies for you to directly use. If the preset policies cannot meet your requirements of fine-grained permissions management, you can customize policies.

Table 9-1 lists all the preset system-defined policies supported by ModelArts.

Table 9-1 System-defined policies supported by ModelArts

Policy	Description	Type
ModelArts FullAccess	Administrator permissions for ModelArts. Users granted these permissions can operate and use ModelArts.	System-defined policy
ModelArts CommonOperations	Common user permissions for ModelArts. Users granted these permissions can operate and use ModelArts, but cannot manage dedicated resource pools.	System-defined policy
ModelArts Dependency Access	Permissions on dependent services for ModelArts	System-defined policy

Generally, ModelArts FullAccess is assigned only to administrators. If fine-grained management is not required, assigning ModelArts CommonOperations to all users will meet the development requirements of most small teams. If you want to customize policies for fine-grained permissions management, see **IAM**.

NOTE

When you assign ModelArts permissions to a user, the system does not automatically assign the permissions of other services to the user. This ensures security and prevents unexpected unauthorized operations. In this case, however, you must separately assign permissions of different services to users so that they can perform some ModelArts operations.

For example, if an IAM user needs to use OBS data for training and the ModelArts training permission has been configured for the IAM user, the IAM user still needs to be assigned with the OBS read, write, and list permissions. The OBS list permission allows you to select the training data path on ModelArts. The read permission is used to preview data and read data for training. The write permission is used to save training results and logs.

- For individual users or small organizations, it is a good practice to configure the **Tenant Administrator** policy that applies to global services for IAM users. In this way, IAM users can obtain all user permissions except IAM. However, this may cause security issues. (For an individual user, its default IAM user belongs to the **admin** user group and has the **Tenant Administrator** permission.)
- If you want to restrict user operations, configure the minimum permissions of OBS for ModelArts users. For details, see **OBS Permissions Management**. For details about fine-grained permissions management of other cloud services, see the corresponding cloud service documents.

ModelArts Agency Authorization

ModelArts must be authorized by users to access other cloud services for AI computing. In the IAM permission system, such authorization is performed through agencies.

For details about the basic concepts and operations of agencies, see **Cloud Service Delegation**.

To simplify agency authorization, ModelArts supports automatic agency authorization configuration. You only need to configure an agency for yourself or specified users on the **Global Configuration** page of the ModelArts console.

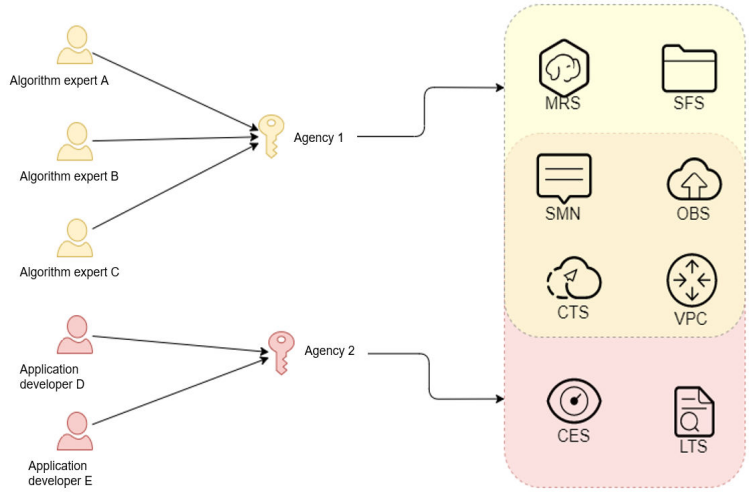
NOTE

- Only users with the IAM agency management permission can perform this operation. Generally, members in the IAM admin user group have this permission.
- ModelArts agency authorization is region-specific, which means that you must perform agency authorization in each region you use.

On the **Global Configuration** page of the ModelArts console, after you click **Add Authorization**, you can configure an agency for a specific user or all users. Generally, an agency named **modelarts_agency_**<*Username*>*_Random ID* is created by default. In the **Permissions** area, you can select the preset permission configuration or select the required policies. If both options cannot meet your requirements, you can create an agency on the IAM management page (you need to delegate ModelArts to access your resources), and then use an existing agency instead of adding an agency on the **Add Authorization** page.

ModelArts associates multiple users with one agency. This means that if two users need to configure the same agency, you do not need to create an agency for each user. Instead, you only need to configure the same agency for the two users.

Figure 9-2 Mapping between users and agencies



NOTE

Each user can use ModelArts only after being associated with an agency. However, even if the permissions assigned to the agency are insufficient, no error is reported when the API is called. An error occurs only when the system uses unauthorized functions. For example, you enable message notification when creating a training job. Message notification requires SMN authorization. However, an error occurs only when messages need to be sent for the training job. The system ignores some errors, and other errors may cause job failures. When you implement permission minimization, ensure that you will still have sufficient permissions for the required operations on ModelArts.

Strict Authorization

In strict authorization mode, explicit authorization by the account administrator is required for IAM users to access ModelArts. The administrator can add the required ModelArts permissions to common users through authorization policies.

In non-strict authorization mode, IAM users can use ModelArts without explicit authorization. The administrator needs to configure the deny policy for IAM users to prevent them from using some ModelArts functions.

The administrator can change the authorization mode on the **Global Configuration** page.

NOTICE

The strict authorization mode is recommended. In this mode, IAM users must be authorized to use ModelArts functions. In this way, the permission scope of IAM users can be accurately controlled, minimizing permissions granted to IAM users.

Managing Resource Access Using Workspaces

Workspace enables enterprise customers to split their resources into multiple spaces that are logically isolated and to manage access to different spaces. As an enterprise user, you can submit the request for enabling the workspace function to your technical support manager.

After workspace is enabled, a default workspace is created. All resources you have created are in this workspace. A workspace is like a ModelArts twin. You can switch between workspaces in the upper left corner of the ModelArts console. Jobs in different workspaces do not affect each other.

When creating a workspace, you must bind it to an enterprise project. Multiple workspaces can be bound to the same enterprise project, but one workspace cannot be bound to multiple enterprise projects. You can use workspaces for refined restrictions on resource access and permissions of different users. The restrictions are as follows:

- Users must be authorized to access specific workspaces (this must be configured on the pages for creating and managing workspaces). This means that access to AI assets such as datasets and algorithms can be managed using workspaces.
- In the preceding permission authorization operations, if you set the scope to enterprise projects, the authorization takes effect only for workspaces bound to the selected projects.

NOTE

- Restrictions on workspaces and permission authorization take effect at the same time. That is, a user must have both the permission to access the workspace and the permission to create training jobs (the permission applies to this workspace) so that the user can submit training jobs in this workspace.
- If you have enabled an enterprise project but have not enabled a workspace, all operations are performed in the default enterprise project. Ensure that the permissions on the required operations apply to the default enterprise project.
- The preceding restrictions do not apply to users who have not enabled any enterprise project.

Summary

Key features of ModelArts permissions management:

- If you are an individual user, you do not need to consider fine-grained permissions management. Your account has all permissions to use ModelArts by default.
- All functions of ModelArts are controlled by IAM. You can use IAM authorization to implement fine-grained permissions management for specific users.
- All users (including individual users) can use specific functions only after agency authorization on ModelArts (**Settings** > **Add Authorization**). Otherwise, unexpected errors may occur.
- If you have enabled the enterprise project function, you can also enable ModelArts workspace and use both basic authorization and workspace for refined permissions management.

10 Security

10.1 Shared Responsibilities

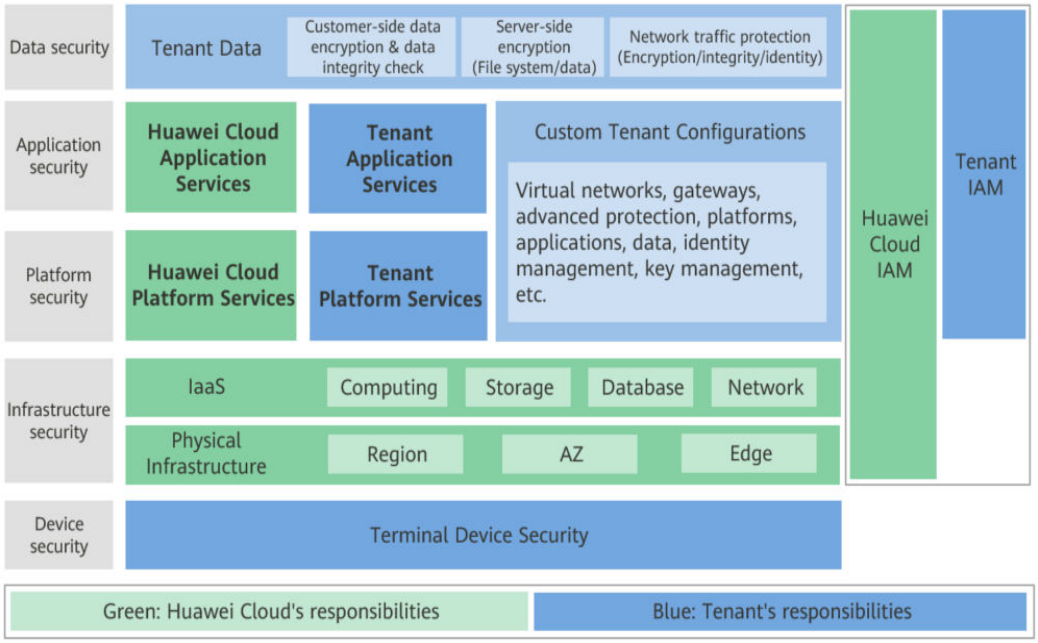
Huawei guarantees that its commitment to cyber security will never be outweighed by the consideration of commercial interests. To cope with emerging cloud security challenges and pervasive cloud security threats and attacks, Huawei Cloud builds a comprehensive cloud service security assurance system for different regions and industries based on Huawei's unique software and hardware advantages, laws, regulations, industry standards, and security ecosystem.

Figure 10-1 illustrates the responsibilities shared by Huawei Cloud and users.

- **Huawei Cloud**: Ensure the security of cloud services and provide secure clouds. Huawei Cloud's security responsibilities include ensuring the security of our IaaS, PaaS, and SaaS services, as well as the physical environments of the Huawei Cloud data centers where our IaaS, PaaS, and SaaS services operate. Huawei Cloud is responsible for not only the security functions and performance of our infrastructure, cloud services, and technologies, but also for the overall cloud O&M security and, in the broader sense, the security compliance of our infrastructure and services.
- **Tenant**: Use the cloud securely. Tenants of Huawei Cloud are responsible for the secure and effective management of the tenant-customized configurations of cloud services including IaaS, PaaS, and SaaS. This includes but is not limited to virtual networks, the OS of virtual machine hosts and guests, virtual firewalls, API Gateway, advanced security services, all types of cloud services, tenant data, identity accounts, and key management.

Huawei Cloud Security White Paper elaborates on the ideas and measures for building Huawei Cloud security, including cloud security strategies, the shared responsibility model, compliance and privacy, security organizations and personnel, infrastructure security, tenant service and security, engineering security, O&M security, and ecosystem security.

Figure 10-1 Huawei Cloud shared security responsibility model



10.2 Asset Identification and Management

Asset Identification

Your assets in AI Gallery include your published AI assets and your personal information.

AI assets include but are not limited to texts, graphics, data, articles, photos, images, illustrations, code, AI algorithms, and AI models.

Your personal information includes:

- Nickname, profile photo, and email for account registration
- Name, mobile number, and email for participating in practices
- Enterprise information for becoming a partner
- Contact name, mobile number, and email for publishing assets

Asset Management

AI Gallery centrally manages assets published by users.

- AI Gallery stores file assets in official OBS buckets.
- AI Gallery stores image assets in official SWR repositories.

AI Gallery stores personal information of users in databases. AI Gallery encrypts sensitive personal information, such as mobile numbers and emails, in databases.

For more information about AI Gallery, see **AI Gallery**.

10.3 Identity Authentication and Access Control

Identity Authentication

You can use ModelArts services through the console, APIs, or SDKs. Essentially, access requests are sent through ModelArts REST APIs.

ModelArts APIs can be accessed upon successful authentication. Requests sent through the console can be authenticated using tokens, and requests for calling APIs can be authenticated using tokens or AK/SK. For details, see **Authentication**.

Access Control

ModelArts allows you to configure fine-grained permissions for refined management of resources and permissions. To do so, ModelArts provides IAM permission control, agency authorization, and workspace.

- IAM permission control

  To use ModelArts functions, you need to grant permissions through IAM. For example, if you need to create a training job on ModelArts, you must have the **modelarts:trainJob:create** permission.

  If no fine-grained authorization policy is configured for a user created by the administrator, the user has all permissions of ModelArts by default. To control user permissions, the administrator needs to add the user to a user group on IAM and configure fine-grained authorization policies for the user group. In this way, the user obtains the permissions defined in the policies before performing operations on cloud service resources. During policy-based authorization, the administrator can select the authorization scope based on ModelArts resource types. For details about resource permissions, see **Permissions Policies and Supported Actions**.

- Agency authorization

  ModelArts needs to access other services for AI computing. For example, ModelArts needs to access OBS to read your data for training. For security purposes, ModelArts must be authorized to access other cloud services. This is agency authorization.

  ModelArts does not save your token authentication credentials. Before performing operations on your resources (such as OBS buckets) in a backend job, you are required to explicitly authorize ModelArts through an IAM agency. ModelArts will use the agency to obtain a temporary authentication credential for performing operations on your resources. For details, see **Configuring Access Authorization (Global Configuration)**.

- Workspace

  Workspace allows customers who have enabled **enterprise projects** to divide their resources into multiple logically isolated spaces and control access to different spaces.

  After workspace is enabled, a default workspace is created. All resources you have created are in this workspace. A workspace is like a ModelArts twin. You can switch between workspaces in the upper left corner of the navigation

pane. Jobs in different workspaces do not affect each other. ModelArts allows you to create multiple workspaces to develop algorithms and manage and deploy models for different service objectives. In this way, the development outputs of different applications are managed in different workspaces for use.

Remote Access Management

When you use a local IDE to remotely access the ModelArts notebook development environment through SSH, the key pair is required for authentication. You can also add the IP addresses for remotely accessing the notebook instance to the whitelist.

10.4 Data Protection

ModelArts takes different measures to keep data stored in ModelArts secure and reliable.

Measure	Description
Static data protection	AI Gallery encrypts sensitive personal information, such as mobile numbers and emails, in databases. The AES encryption algorithm is used.
Data transmission protection	When you import AI applications on ModelArts, it supports HTTP and HTTPS, but HTTPS is recommended for more secure data transmission.
Data integrity check	When you upload model files or AI Gallery assets for inference deployment, data may become inconsistent due to network hijacking, caching, and other reasons. ModelArts verifies data consistency by calculating the SHA256 value when data is uploaded or downloaded.
Data isolation mechanism	When a notebook instance is created, data storage of different tenants is isolated, so that different tenants cannot view data of other tenants.

10.5 Auditing and Logging

Auditing

Cloud Trace Service (CTS) records operations on the cloud resources in your account. You can use the logs generated by CTS to perform security analysis, trace resource changes, audit compliance, and locate faults.

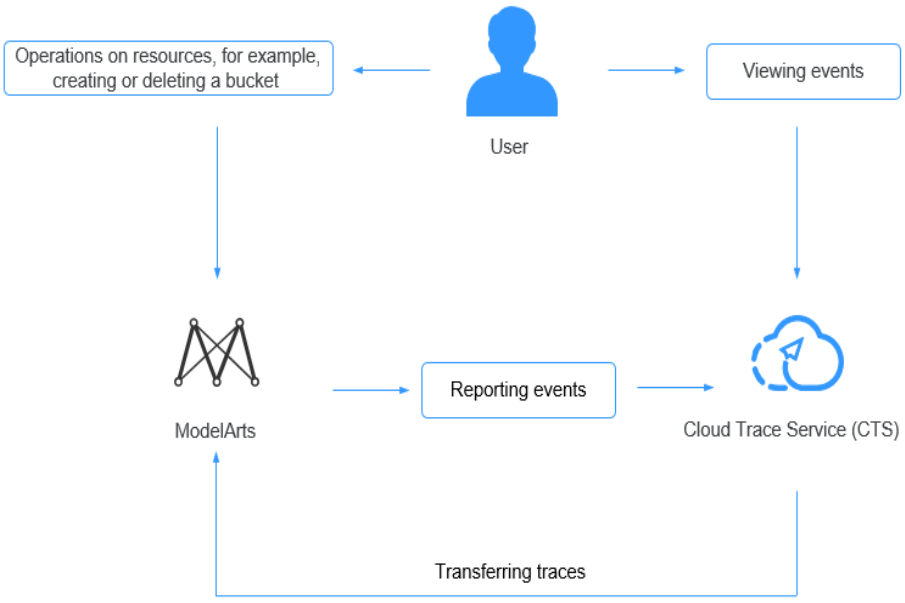
After you enable CTS and configure a tracker, CTS can record management and data traces of ModelArts for auditing.

For details about how to enable and configure CTS, see **Enabling CTS**.

For details about ModelArts management and data traces that can be tracked by CTS, see **Key Operations Recorded for Data Management**, **Key DevEnviron**

Operations Recorded by CTS, **Key Training Job Operations Recorded by CTS**, **Key AI Application Management Operations Recorded by CTS**, and **Key Service Management Operations Recorded by CTS**.

Figure 10-2 CTS



Key Data Management Operations Recorded by CTS

Table 10-1 Key data management operations recorded by CTS

Operation	Resource Type	Trace
Creating a dataset	dataset	createDataset
Deleting a dataset	dataset	deleteDataset
Updating a dataset	dataset	updateDataset
Publishing a dataset version	dataset	publishDatasetVersion
Deleting a dataset version	dataset	deleteDatasetVersion
Synchronizing the data source	dataset	syncDataSource
Exporting a dataset	dataset	exportDataFromDataset
Creating an auto labeling task	dataset	createAutoLabelingTask
Creating an auto grouping task	dataset	createAutoGroupingTask

Operation	Resource Type	Trace
Creating an automatic deployment task	dataset	createAutoDeployTask
Importing samples to a dataset	dataset	importSamplesToDataset
Creating a dataset label	dataset	createLabel
Modifying a dataset label	dataset	updateLabel
Deleting a dataset label	dataset	deleteLabel
Deleting a dataset label and the corresponding samples	dataset	deleteLabelWithSamples
Adding samples	dataset	uploadSamples
Deleting samples	dataset	deleteSamples
Stopping an auto labeling task	dataset	stopTask
Creating a team labeling job	dataset	createWorkforceTask
Deleting a team labeling job	dataset	deleteWorkforceTask
Starting the acceptance of team labeling	dataset	startWorkforceSamplingTask
Approving/rejecting/canceling acceptance	dataset	updateWorkforceSamplingTask
Submitting sample review comments for acceptance	dataset	acceptSamples
Adding a label to a sample	dataset	updateSamples
Sending an email to labeling team members	dataset	sendEmails
Starting the team labeling job as the contact person	dataset	startWorkforceTask
Updating a team labeling job	dataset	updateWorkforceTask
Adding a label to a team-labeled sample	dataset	updateWorkforceTaskSamples
Reviewing team labeling results	dataset	reviewSamples
Creating a labeling team member	workforce	createWorker
Updating a labeling team member	workforce	updateWorker
Deleting a labeling team member	workforce	deleteWorker

Operation	Resource Type	Trace
Batch deleting labeling team members	workforce	batchDeleteWorker
Creating a labeling team	workforce	createWorkforce
Updating a labeling team	workforce	updateWorkforce
Deleting a labeling team	workforce	deleteWorkforce
Automatically creating an IAM agency	IAM	createAgency
Logging in to the labeling console as a labeling team member	labelConsoleWorker	workerLoginLabelConsole
Logging out of the labeling console as a labeling team member	labelConsoleWorker	workerLogOutLabelConsole
Changing the password of the labeling console as a labeling team member	labelConsoleWorker	workerChangePassword
Forgetting the password of the labeling console as a labeling team member	labelConsoleWorker	workerForgetPassword
Resetting the password of the labeling console through the URL as a labeling team member	labelConsoleWorker	workerResetPassword

Key DevEnviron Operations Recorded by CTS

Table 10-2 Key DevEnviron operations recorded by CTS

Operation	Resource Type	Trace Name
Creating a notebook instance	Notebook	createNotebook
Deleting a notebook instance	Notebook	deleteNotebook
Opening a notebook instance	Notebook	openNotebook
Starting a notebook instance	Notebook	startNotebook
Stopping a notebook instance	Notebook	stopNotebook

Operation	Resource Type	Trace Name
Updating a notebook instance	Notebook	updateNotebook
Deleting a NotebookApp	NotebookApp	deleteNotebookApp
Switching CodeLab specifications	NotebookApp	updateNotebookApp

Key Training Job Operations Recorded by CTS

Table 10-3 Key training job operations recorded by CTS

Operation	Resource Type	Trace
Creating a training job	ModelArtsTrainJob	createModelArtsTrainJob
Creating a training job version	ModelArtsTrainJob	createModelArtsTrainVersion
Stopping a training job	ModelArtsTrainJob	stopModelArtsTrainVersion
Modifying the description of a training job	ModelArtsTrainJob	updateModelArtsTrainDesc
Deleting a training job version	ModelArtsTrainJob	deleteModelArtsTrainVersion
Deleting a training job	ModelArtsTrainJob	deleteModelArtsTrainJob
Creating a training job configuration	ModelArtsTrainConfig	createModelArtsTrainConfig
Modifying a training job configuration	ModelArtsTrainConfig	updateModelArtsTrainConfig
Deleting a training job configuration	ModelArtsTrainConfig	deleteModelArtsTrainConfig
Creating a visualization job	ModelArtsTensorboardJob	createModelArtsTensorboardJob
Deleting a visualization job	ModelArtsTensorboardJob	deleteModelArtsTensorboardJob
Modifying the description of a visualization job	ModelArtsTensorboardJob	updateModelArtsTensorboardDesc
Stopping a visualization job	ModelArtsTensorboardJob	stopModelArtsTensorboardJob

Operation	Resource Type	Trace
Restarting a visualization job	ModelArtsTensor-boardJob	restartModelArtsgTen-sorboardJob

Key AI Application Management Operations Recorded by CTS

Table 10-4 Key AI application management operations recorded by CTS

Operation	Resource Type	Trace
Creating an AI application	model	addModel
Updating an AI application	model	updateModel
Deleting an AI application	model	deleteModel
Creating a model conversion task	convert	addConvert
Updating a model conversion task	convert	updateConvert
Deleting a model conversion task	convert	deleteConvert

Key Service Management Operations Recorded by CTS

Table 10-5 Key service management operations recorded by CTS

Operation	Resource Type	Trace
Deploying a service	service	addService
Deleting a service	service	deleteService
Updating a service	service	updateService
Starting or stopping a service	service	startOrStopServic e
Adding an access key	service	addAkSk
Deleting an access key	service	deleteAkSk
Creating a dedicated resource pool	cluster	createCluster
Deleting a dedicated resource pool	cluster	deleteCluster
Adding a node to a dedicated resource pool	cluster	addClusterNode

Operation	Resource Type	Trace
Deleting a node from a dedicated resource pool	cluster	deleteClusterNode
Obtaining a result from the dedicated resource pool creation	cluster	createClusterResult

Key AI Gallery Operations Recorded by CTS

Table 10-6 Key AI Gallery operations recorded by CTS

Operation	Resource Type	Trace
Publishing an asset	ModelArts_Market	create_content
Modifying asset information	ModelArts_Market	modify_content
Publishing an asset version	ModelArts_Market	add_version
Subscribing to an asset	ModelArts_Market	subscription_content
Removing an asset from favorites	ModelArts_Market	cancel_star_content
Liking an asset	ModelArts_Market	like_content
Unliking an asset	ModelArts_Market	cancel_like_content
Publishing an activity	ModelArts_Market	publish_activity
Signing up an activity	ModelArts_Market	regist_activity
Modifying user information	ModelArts_Market	update_user

Logging

You can enable ModelArts logging for analysis or audit. After CTS is enabled, CTS starts recording operations on ModelArts. The CTS management console stores the last seven days of operation records. This section describes how to view operation records of the last 7 days on the CTS management console.

For details about how to view audit logs on CTS, see **Viewing Audit Logs**.

10.6 Service Resilience

Resilience refers to security resilience of cloud services after attacks, excluding reliability and availability. This chapter describes ModelArts capabilities of defense

and detection against intrusions, defense against jitter, proper use of domain names, and content security detection.

Security Suite and Cloud Bastion Host for Enhanced Defense and Detection Against Intrusions

Security suites have been deployed on ModelArts at the host, application, network, and data layers to promptly detect intrusions.

- ModelArts uses web secure components to prevent web security risks from web applications deployed on it and uses WAF for security protection.
- Host Security Service (HSS) products have been deployed on all hosts that carry ModelArts services. These products include but not limited to Huawei-developed HSS and Compute Security Platform (CSP).
- Vulnerability Scan Service (VSS) has been deployed on ModelArts and performs routine scanning to quickly detect and fix vulnerabilities.
- ModelArts performs security O&M on cloud resources through a security management platform.
- Situation Awareness (SA) has been deployed on ModelArts to understand security situation, query attack histories, and promptly detect compliance risks and respond to threat alarms.
- Advanced Anti-DDoS (AAD) has been deployed on the EIPs that carry key ModelArts services to prevent traffic storms.
- Database Security Service (DBSS) has been deployed on ModelArts databases that store important data.

Jitter Prevention and Emergency Response and Restoration Policies Against Attacks

ModelArts isolates resources of different tenants, so that attacks on a tenant's resources will not affect others' resources.

- ModelArts provides dedicated resource pools that are physically isolated, so that attacks on a tenant's resources will not affect others' resources.
- ModelArts defines and maintains its performance specifications to defend attacks, for example, by configuring traffic control on API access.
- ModelArts provides alarm reporting and self-protection against attacks.
- ModelArts detects abnormal service behavior, for example, by detecting abnormal operations platform data and integrating security logs.
- ModelArts provides risk control and emergency response against attacks. For example, ModelArts quickly identifies malicious tenants and malicious IP addresses.
- ModelArts quickly restores services after traffic attacks stop.

Domain Name Usage Specifications and Tenant Content Security Policies of Cloud Services

ModelArts domain names meet certain security requirements to avoid compliance risks and phishing attacks.

Domain names visible to tenants: domain names accessible to tenants, which require more attention to security and compliance.

Domain names invisible to tenants: domain names used by Huawei Cloud services to call each other on the intranet, in which case external users are not able to access the authoritative DNS servers; or domain names that can only be accessed by Huawei employees, partner staff, and outsourced personnel in yellow and green zones through Huawei's office network (namely these domain names cannot be accessed over the Internet).

- Huawei Cloud basic domain names are not directly allocated to tenants but securely used.
- External domain names that have been licensed are not used by Huawei Cloud services to call each other on the intranet.

10.7 Risk Monitoring

ModelArts automatically monitors your real-time services and models in real time and manages alarms and notifications, so that you can keep track of performance metrics of services and models. For details, see **ModelArts Metrics**.

10.8 Fault Recovery

ModelArts global infrastructure is built for Huawei Cloud regions and AZs. A Huawei Cloud region provides multiple physically independent and isolated AZs that are connected through networks with low latency, high throughput, and high redundancy. You can design and operate faulty applications and databases automatically migrated between AZs without interrupting services. Compared with the traditional infrastructure of a single data center or multiple data centers, AZs provide higher availability, fault tolerance, and scalability.

ModelArts backs up its database data for recovery in case of a service failure or original data damage.

Fault Environment Recovery

If a compute node used by a notebook instance is faulty, the instance will be automatically migrated to another available node. Then, the instance is restored. ModelArts enables you to mount an EVS disk to an instance. Huawei Cloud EVS provides scalable block storage that features high reliability, high performance, and a variety of specifications for servers. Data durability reaches 99.9999999%.

Automatic Recovery from a Training Fault

During model training, a training failure may occur due to a hardware fault. For hardware faults, ModelArts provides fault tolerance check to isolate faulty nodes to improve user experience in training.

The fault tolerance check involves environment pre-check and periodic hardware check. If any fault is detected during either of the checks, ModelArts automatically isolates the faulty hardware and issues the training job again. In distributed

training, the fault tolerance check will be performed on all compute nodes used by the training job.

Recovery from an Inference Deployment Fault

During the service running, if an inference instance is faulty due to a hardware fault, ModelArts automatically detects the fault and migrates the faulty instance to another available node. After the instance is restarted, it will be restored. The faulty node is automatically isolated and not be scheduled for running inference instances.

10.9 Upgrade Management

ModelArts Real-Time Service Upgrade

For a deployed service, you can change the AI application version to upgrade it.

Services can be upgraded in three modes: full upgrade, rolling upgrade (increase instances), and rolling upgrade (decrease instances). For details about the three upgrade modes, see **Figure 10-3**.

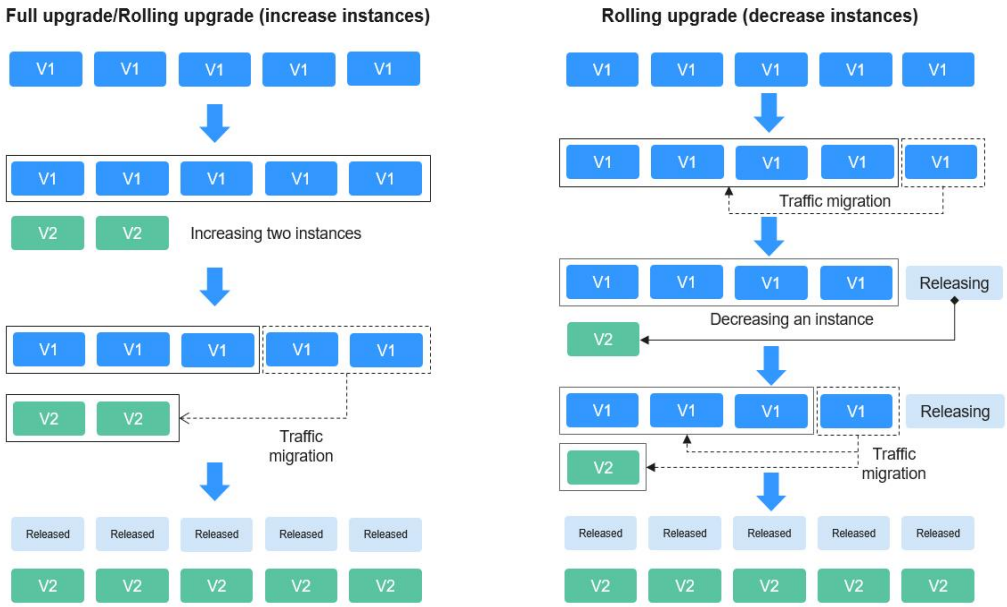
- Full upgrade

  Resources that are twice as many as those used by the service will be used to create new-version instances in full mode.
- Rolling upgrade (increase instances)

  Extra resources than those used by the service will be used for a rolling upgrade. A larger number of instances to be increased will lead to a faster upgrade.
- Rolling upgrade (decrease instances)

  Certain nodes that were intended to run services will be used for a rolling upgrade. A larger number of instances to be reduced will lead to a faster upgrade but a higher probability of service interruption.

Figure 10-3 Service upgrade process



For details about how to upgrade an inference service, see **Upgrading a Service**.

Image Upgrade

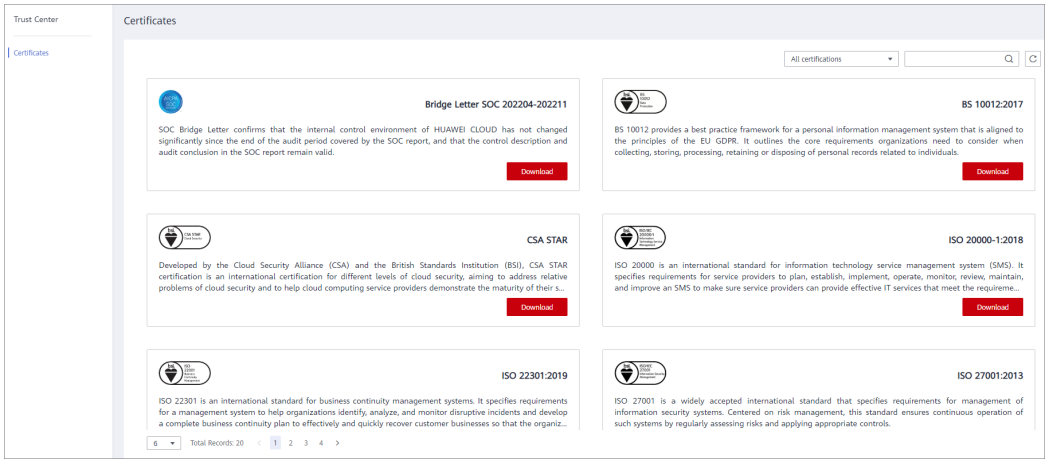
ModelArts provides three function modules: DevEnviron, training management, and inference deployment. The three modules provide base images by the same process. These images are upgraded irregularly to fix vulnerabilities.

10.10 Certificates

Compliance Certificates

Huawei Cloud services and platforms have obtained various security and compliance certifications from authoritative organizations, such as International Organization for Standardization (ISO). You can **download** them from the console.

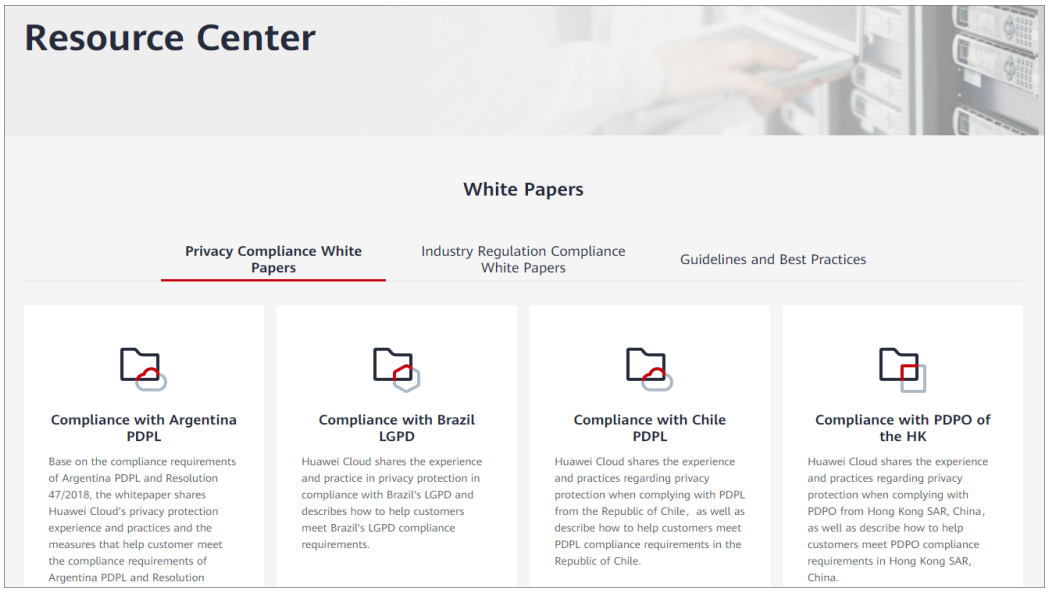
Figure 10-4 Downloading compliance certificates



Resource Center

Huawei Cloud also provides the following resources to help users meet compliance requirements. For details, see **Resource Center**.

Figure 10-5 Resource center



10.11 Security Boundary

The shared responsibility model is a cooperation mode where both providers and customers take security and compliance responsibilities of cloud services.

The providers manage the cloud infrastructure and provide secure hardware and software to ensure the service availability. The customers protect their data and applications, while complying with related compliance requirements.

The providers are responsible for the services and functions and should:

- Establish and maintain secure infrastructure, including networks, servers, and storage devices.
- Provide reliable underlying platforms to ensure runtime security for the environment.
- Provide identity authentication and access control to ensure that only authorized users can access the cloud services and tenants are isolated from each other.
- Provide reliable backup and disaster recovery to prevent data loss due to hardware faults or natural disasters.
- Provide transparent monitoring and incident response services, security updates, and vulnerability patches.

The customers should:

- Encrypt data and applications for confidentiality and integrity.
- Ensure that the AI application software is securely updated and vulnerabilities are fixed.
- Comply with related regulations, such as GDPR, HIPAA, and PCI DSS.
- Control access to ensure that only authorized users can access and manage resources such as online services.
- Monitor and report any abnormal activity and take actions in a timely manner.

Inference Deployment Security Responsibilities

- Providers
  - Fix the patches related to underlying ECSs.
  - Upgrade the K8S and fix vulnerabilities.
  - Operate VM OS lifecycle maintenance.
  - Ensure the security and compliance of the ModelArts inference platform.
  - Improve the security of container application services.
  - Upgrade the model runtime environment and fix vulnerabilities periodically.
- Customers
  - Authorize resource use and control access.
  - Ensure the security of applications, its supply chain, and dependencies by security scanning, auditing, and access verification.
  - Minimize permissions and limit credential delivery.
  - Ensure the security of AI applications (custom images, OBS models, and dependencies) during runtime.
  - Update and fix vulnerabilities in a timely manner.
  - Securely store sensitive data such as credentials.

Best Practices for Inference Deployment Security

- External service authorization

  ModelArts inference requires authorization from other cloud services. You can grant only the required permissions based on your needs. For example, you

can grant access permission on an OBS bucket to a tenant for model management.

- Internal resource authorization

  ModelArts inference supports fine-grained permission control. You can configure the permissions for users based on the actual needs to restrict the permissions on some resources.

- AI application management

  To decouple models from images and protect model assets, you can dynamically import AI applications from trainings or OBS. You need to upgrade the dependency packages of AI applications, and fix vulnerabilities in open-source or third-party packages. Sensitive information related to AI applications needs to be decoupled and configured during deployment. Select the runtime environment recommended by ModelArts. The earlier environments may have security vulnerabilities.

  You can select open trusted images when creating AI applications from a container image, for example, images from OpenEuler, Ubuntu, and NVIDIA. Create non-root users rather than root users to run an image. Only the security package required during the runtime is installed in the image. Downsize the image and upgrade the installation package to the latest vulnerability-free version. Decouple sensitive information from images during service deployment. Do not directly use the information in Dockerfile. Perform security scanning on images periodically and install patches to fix vulnerabilities. To facilitate alarm reporting and fault rectification, add health check interface and ensure that the service status can be returned properly. To ensure the service data security, use HTTPS transmission streams and reliable encryption suites for containers.

- Model deployment

  To prevent services from being overloaded or wasted, set proper compute node specifications during deployment. Do not listen to other ports in the container. If other ports need to be accessed locally, listen to them on localhost. Do not directly transfer sensitive information through environment variables. Encrypt sensitive information with encryption component before data transmission.

  App authentication key is an access credential for real-time services. You must keep the app key properly.

11 Quotas

ModelArts uses the following infrastructure resources:

- ECS
- EVS
- VPC

For details about how to view and modify the quota, see **Quotas**.

