ModelArts

Service Overview

Issue 01

Date 2025-02-06

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

2 What Is ModelArts?

ModelArts is a one-stop development platform provided by Huawei Cloud. With large-volume data preprocessing, semi-automated data labeling, distributed training, automated model building, and on-demand model deployment across devices, edge devices, and cloud, ModelArts helps you quickly build and deploy models and efficiently manage the AI development lifecycle.

ModelArts covers all stages of AI development, including data processing, algorithm development, and model training and deployment. The underlying technologies of ModelArts support a wide range of heterogeneous computing resources, allowing you to flexibly select and use the resources that fit your needs. 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.

Offerings

ModelArts offers several products.

Table 2-1 ModelArts offerings

Offering	Overview	Scenario
ModelArts Standard	ModelArts Standard is a one-stop platform for AI development. It offers a user-friendly console with integrated toolchains for ExeML, data management, development environments, model training, management, and deployment. This enables seamless management of the AI development lifecycle.	It is ideal for users who require an AI development platform.

Offering	Overview	Scenario
ModelArts MaaS	ModelArts MaaS offers an end-to-end toolchain for foundation model production, along with Ascend computing resources and popular open-source models. It enables data production, model fine-tuning, prompt engineering, and application orchestration.	It is designed for users who need to develop production-ready models using a MaaS platform.
ModelArts Lite Server	ModelArts Lite Server offers cloud servers running on bare metal servers, accessible via EIPs.	It is designed for users who have built their own AI development platforms and require only computing power. It provides cost-effective AI computing power, mainstream AI development suites, and Huawei's acceleration plug-ins.
ModelArts Lite Cluster	ModelArts Lite Cluster provides direct access to Kubernetes APIs, enabling you to manage nodes and clusters within resource pools.	It is designed for users who have built their own AI development platforms and require only computing power. It requires basic knowledge of Kubernetes.
ModelArts Edge	ModelArts Edge simplifies the deployment and management of edge computing. It supports various types of edge devices and offers features such as model deployment, AI application and node management, resource pooling and load balancing, and application commercial use assurance. With ModelArts Edge, you can quickly build cost-effective AI solutions that leverage both edge and cloud resources.	It is suitable for edge deployment.

Offering	Overview	Scenario
AI Gallery	AI Gallery offers a high-quality model development experience on Ascend Cloud and provides access to a wealth of community resources.	It is suitable for AI development and exploration.

Architecture

- The computing power layer offers a full range of Ascend hardware, 10,000-card cluster management, and resource scheduling and management. It supports popular AI development, debugging, training, and inference frameworks.
- The AI platform layer offers an end-to-end AI development toolchain, enabling you to develop and deploy models quickly and easily. It also enables efficient resource management and automatic fault recovery, speeding up AI model development, training, and deployment.
- The AI development toolchain layer offers an end-to-end foundation model development toolchain, including high-quality open-source models and development suites that streamline the development process and reduce time to market.

Access Methods

ModelArts provides multiple access methods for different products.

- **Console**

  ModelArts Standard can be accessed through the console, where you can perform various tasks such as ExeML, data management, model training, model management, and model deployment. The console allows for end-to-end AI development.

  ModelArts MaaS can be accessed through the console, where you can perform various tasks such as data production, model fine-tuning, prompt engineering, and application orchestration.

- **SDK**

  ModelArts Standard can be called through SDKs to be integrated into a third-party system for secondary development. ModelArts SDKs encapsulate ModelArts Standard RESTful APIs in Python to simplify development. For details about the SDKs and operations, see the **ModelArts SDK Reference**.

  Additionally, you can directly call the ModelArts SDKs when writing code in ModelArts Standard Notebook.

- **API**

  If you want to integrate ModelArts Standard into a third-party system for secondary development, use APIs to access ModelArts..

- **Cloud native**

  ModelArts Lite Server allows you to use EIPs to access your cloud servers. For details, see **ModelArts Lite Server User Guide**.

ModelArts Lite Cluster allows you to use native K8s APIs to manage your clusters. For details, see **ModelArts Lite Cluster User Guide**.

3 Advantages

ModelArts has the following advantages:

Stable and secure computing backbone, fast and simple model training

- 10,000-node compute clusters
- Large-scale distributed training for accelerated foundation model development
- Cost-effective proprietary compute
- Decades of software and hardware expertise in the optimization of AI applications
- Acceleration suites for training, inference, and data access

One-stop E2E development toolchain for a consistent development experience

- The out-of-the-box and full-lifecycle AI development platform provides one-stop data processing, and development, training, management, and deployment of models.
- Local IDE and ModelArts plug-ins are provided for seamless on-premises and in-cloud AI development and training. Distributed deployment and inference of foundation models is supported.
- E2E AI development is managed, boosting efficiency while maintaining records of the entire AI development process.

Flexible deployment for various scenarios

- Multiple production environments, including cloud and edge
- Multiple deployment types, including real-time inference, batch inference, and edge inference

AI engineering for AI lifecycle management

- MLOps, analytics on data, models, and training logs, as well as monitoring and diagnosis

Strong fault tolerance for fast fault recovery

- Awareness and detection across racks, nodes, accelerator cards, and tasks
- Recovery at the node, job, and container levels, ensuring uninterrupted 1,000-card training

Multiple resource deployment options

- Cluster mode: Kubernetes clusters come pre-configured and ready for immediate use.
- Node mode: By utilizing open-source or your own custom frameworks, you can create clusters that offer enhanced control and flexibility.

Migration without any reconstruction

- Standard Kubernetes APIs for resource utilization, ensuring smooth migration across clouds
- A consistent experience guaranteed by SSH access to nodes and containers

4 Use Cases

This section describes ModelArts use cases.

Foundation Model

Integrates third-party open-source foundation models for intelligent Q&A, chatbots, automated summarization, machine translation, and text classification.

AIGC

Provides scenario-specific solutions for generating images, text, audio, and video.

Autonomous Driving

Achieves environment perception, path planning, and control for autonomous driving. Efficient training with PB-scale data speeds up innovation and iteration. Optimized algorithms across the entire autonomous driving chain significantly boost performance.

Content Moderation

Delivers robust solutions for content moderation and facilitates fast migration (of CV workloads) to Ascend, meeting the compute and continuity needs of content platforms and service providers.

Governments

Empowers AI-driven decision-making for superior public services.

Finance

Offers efficient, intelligent, and accurate services for financial institutions.

Mining

Provides an end-to-end AI development pipeline and high-performance compute, improving inference efficiency and delivering secure, intelligent, and sustainable production solutions.

Railway

Enables intelligent train scheduling, fault prediction, and security monitoring.

Healthcare

Supports automated medical report interpretation, online diagnosis, and comprehensive health management for diverse, intelligent, and efficient services.

5 Functions

5.1 ModelArts Standard Functions

5.1.1 Standard ExeML

ModelArts helps business developers without algorithm development capabilities to develop algorithms through machine learning. It automatically generates models based on transfer learning and Neural Architecture Search (NAS). With the automatic learning function of parameter selection for model training and automatic model optimization, it enables business developers with no AI background to quickly complete model training and deployment.

ModelArts ExeML provides zero-code AI solutions for entry-level users.

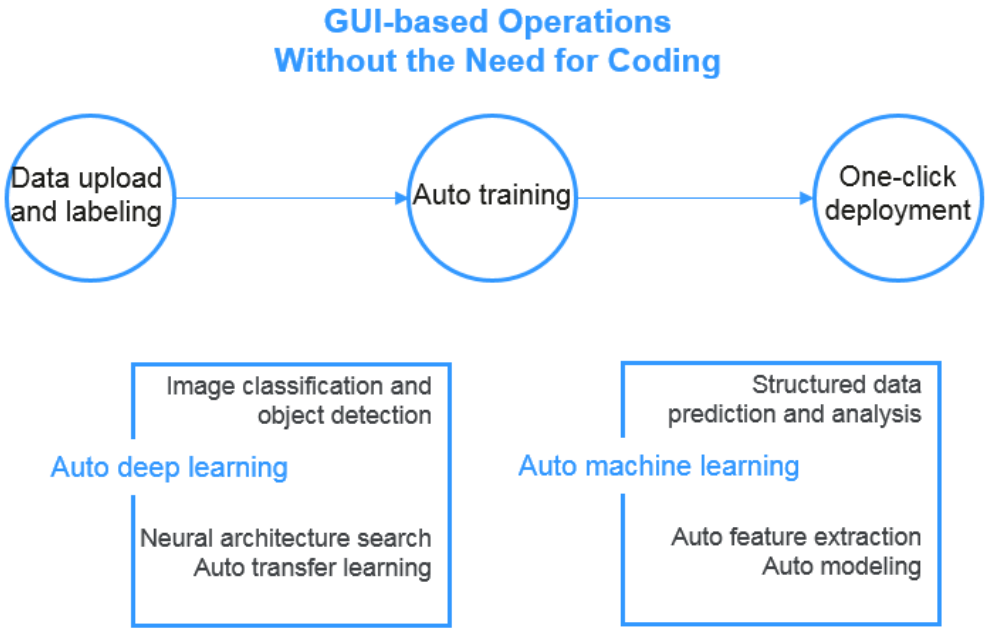
- Supports image classification, object detection, sound classification, and predictive analysis scenarios.
- Automates the end-to-end process of model development, training, tuning, and inference.
- Automatically optimizes and generates models that meet the requirements based on the final deployment environment and developer needs.

ModelArts ExeML provides template-based development capabilities for advanced users.

- Offers "auto learning white-boxing" capability, opening up model parameters and automatically generating models to achieve template-based development and improve development efficiency.
- Utilizes automatic deep learning technology, including transfer learning (generating high-quality models with minimal data), automatic design of model architecture in multiple dimensions (neural network search and adaptive model tuning), and faster and more accurate automatic training parameter tuning.
- Utilizes automatic machine learning technology, including tree search for optimal feature transformation based on the upper limit of information entropy approximation model and Bayesian optimization for automatic parameter tuning based on the upper limit of information entropy approximation model. It automatically learns data features and patterns from enterprise relational (structured) data, intelligently optimizes features, ML

models, and parameters, achieving accuracy comparable to that of expert developers in tuning.

Figure 5-1 Process of using ExeML



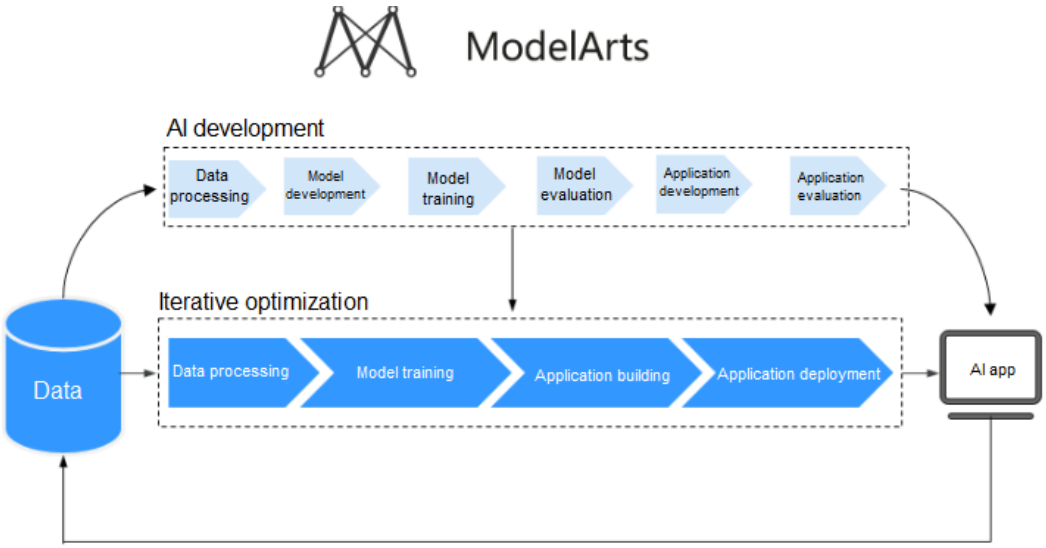
5.1.2 Standard Workflow

Workflow is a pipeline tool that developers use to deploy models or applications based on real-world business scenarios. Its core function is to break down the complete machine learning task into multiple steps of a workflow, where each step is a manageable component that can be developed, optimized, configured, and automated individually. Workflow helps standardize the process of generating machine learning models, enabling teams to execute AI tasks at a large scale and improve the efficiency of model generation.

ModelArts Workflow provides a standardized MLOps solution that reduces model training costs.

- Supports steps such as data labeling, data processing, model development/training, model evaluation, application development, and application evaluation.
- Automatically coordinates all dependencies between workflow steps, providing features like run records, monitoring, and continuous execution.
- For workflow development, Workflow offers the necessary functionality and parameter descriptions for the pipeline, allowing you to define the steps and their relationships using the SDK.
- For workflow reuse, users can save the pipeline after development, making it available for future use or for other team members, without needing to worry about the algorithms or implementation details included in the pipeline.

Figure 5-2 Workflow process



5.1.3 Standard Data Management

ModelArts Standard Data Management offers an efficient and convenient framework for managing and labeling data. It supports various data types such as images, text, audio, and video, covering multiple labeling scenarios like image classification, object detection, audio segmentation, and text classification. It is suitable for AI projects in computer vision, natural language processing, audio/video analysis, and more.

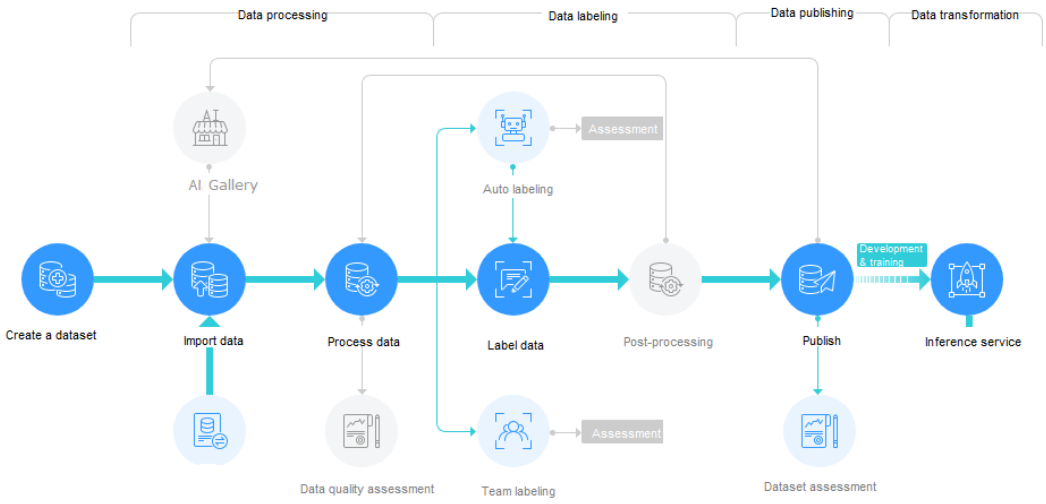
NOTE

The ModelArts Standard Data Management module is currently undergoing a reconstruction and will be upgraded with new capabilities in the era of large models. Stay tuned for updates.

ModelArts Standard Data Management supports multi-dimensional data management capabilities:

- Dataset management: Provides features for creating datasets, previewing data, and managing dataset versions.
- Data labeling: Offers online labeling capabilities for various scenarios like image classification, object detection, audio segmentation, and text triplets. It also provides intelligent image labeling solutions to enhance efficiency. Team labeling features support collaborative efforts by multiple users, as well as the review and acceptance of labeling tasks.
- Data processing: Provides analysis and processing capabilities such as data cleansing, data validation, data augmentation, and data selection.

Figure 5-3 Data labeling process



5.1.4 Standard Development Environment

Software development is a process of reducing developer costs and improving development experience. In AI development, ModelArts is also committed to improving the AI development experience and lowering the development threshold. ModelArts Standard Development Environment aims to provide a better cloud-based AI development experience for different types of AI development, exploration, and teaching users, through the integration of cloud native resource usage and development toolchains.

ModelArts Standard Notebook for seamless collaboration between the cloud and local environments

- Code development and debugging: Cloud-based JupyterLab usage, local IDE + ModelArts plugin for remote development capabilities, tailored to developers' usage habits.
- Cloud-based development environment, including AI compute resources, cloud storage, and pre-installed AI engines.
- Customizable runtime environment, allowing the development environment to be saved as an image for training and inference purposes.

Highlight Feature 1: Remote Development - Support for Local IDE Remote Access to Notebook

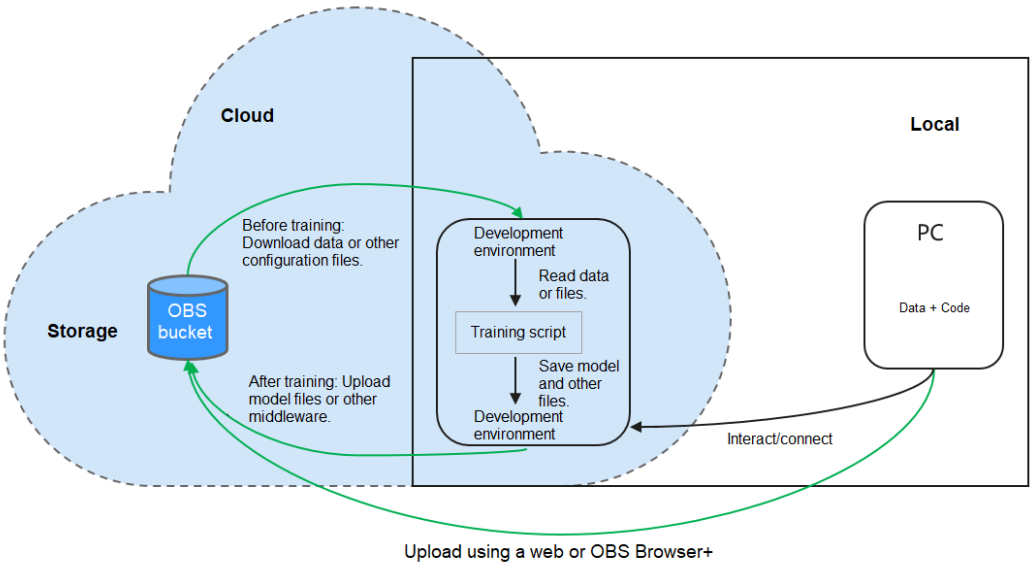
Notebook provides remote development by enabling SSH connection, allowing you to remotely connect your local IDEs to the ModelArts Notebook development environment to debug and run code.

For developers using a local IDE, due to resource limitations, the running and debugging environment is mostly shared on a team's resource server. This brings about certain environment setup and maintenance costs.

The advantage of ModelArts Notebook is that it is ready to use. It comes pre-installed with different AI engines and provides a wide range of selectable

specifications. You can have an exclusive container environment without interference from others. With simple configuration, you can connect to this environment through your local IDE for running and debugging.

Figure 5-4 Remotely accessing Notebook from a local IDE



Notebook can be seen as an extension of the local PC, both considered as local development environments. Its operations, such as reading data, training, and saving files, are consistent with regular local training.

For developers accustomed to using a local IDE, remote development does not affect their coding habits and allows them to easily and conveniently use the cloud-based Notebook development environment.

A local IDE supports VS Code, PyCharm, and SSH. There are also dedicated plugins, PyCharm Toolkit and VS Code Toolkit, which facilitate using cloud resources as local extensions.

Highlight Feature 2: One-Click Image Saving for Development Environment

Notebook provides the functionality to save images. It supports one-click saving of running Notebook instances as images, preserving the prepared environment for future use and easy sharing.

When saving an image, installed dependencies (pip packages) are not lost, and in the case of VS Code remote development scenarios, the plugins installed on the server are not lost.

Highlight Feature 3: Pre-Installed Images - Ready to Use, Optimized Configuration, Support for Leading AI Engines

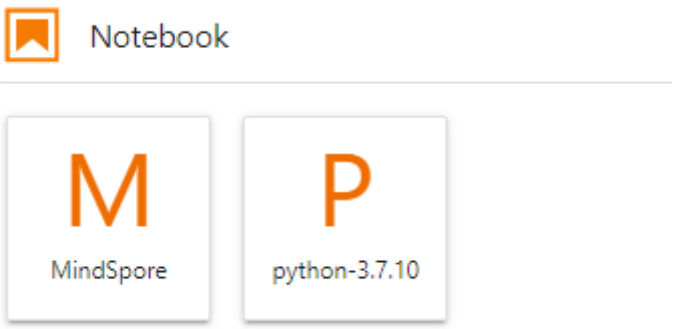
Each image comes with pre-installed AI engines and versions, and the AI engine and version, including the compatible chips, are specified when creating a Notebook instance.

The development environment provides you with a set of pre-installed images, mainly including PyTorch, TensorFlow, and MindSpore. You can directly use these pre-installed images to launch Notebook instances. After completing development in the instance, you can submit ModelArts training jobs without the need for adaptation.

The versions of pre-installed images provided in the development environment are determined based on user feedback and version stability. If the versions provided by ModelArts meet your function development requirements, you are advised to use the pre-installed images. These images have undergone thorough functional verification and come with many commonly used installation packages, saving your time on environment configuration.

The preset images provided in the development environment mainly include:

- Commonly used pre-installed packages based on standard Conda environments, including popular AI engines such as PyTorch and MindSpore; common data analysis packages such as Pandas and Numpy; and common tools such as CUDA and cuDNN to meet common AI development needs.
- Pre-installed Conda environments: Each pre-installed image creates a corresponding Conda environment and a basic Conda environment python (without any AI engines). For example, the Conda environment corresponding to the pre-installed MindSpore is as follows:



  You can choose different Conda environments based on whether they use AI engines for functional debugging.
- Notebook: a web application that allows you to write code in an interface and combine code, mathematical equations, and visual content into a document.
- JupyterLab plug-ins: Plug-ins include specification switching, sharing cases to AI Gallery for communication, and stopping instances, to enhance user experience.
- Support for remote SSH, allowing you to remotely debug a notebook instance from a local PC.
- Once the images are customized in the ModelArts development environment, they can be directly utilized in ModelArts for training jobs.

NOTE

- To simplify operations, the new version of ModelArts Notebook does not support switching between different engines within the same Notebook instance.
- Different AI engines are supported in different regions. Refer to the actual interface on the console.

Highlight Feature 4: Online Interactive Development and Debugging Tool - JupyterLab

ModelArts integrates the open source JupyterLab, which provides online interactive development and debugging. You can directly use Notebook on the ModelArts management console without worrying about installation and configuration. You can write and debug model training code in Notebook and then train models based on that code.

JupyterLab is an interactive development environment and the next-gen Jupyter Notebook. It allows you to write notebooks, operate terminals, edit Markdown text, open interactive modes, view CSV files and images, and more.

5.1.5 Standard Model Training

ModelArts Standard Model Training offers containerized services and compute resource management capabilities. It establishes and manages the infrastructure for machine learning training workloads, alleviating the burden on users and providing a flexible, stable, user-friendly, and high-performance deep learning training environment. With ModelArts Standard Model Training, you can focus on developing, training, and fine-tuning models.

ModelArts Standard Model Training supports large-scale training jobs and provides a highly available training environment.

- Supports distributed training with single-device multi-card and multi-device multi-card configurations, effectively accelerating the training process.
- Supports fault awareness, diagnosis, and recovery for training jobs, including hardware failures and job freezes. It provides process-level, container-level, and job-level recovery, ensuring the long and stable operation of your training jobs.
- Provides the ability for checkpoint-based training resumption and incremental training. Even if the training is interrupted for some reason, it can be resumed based on the checkpoint, ensuring the stability and reliability of models that require long training time and avoiding the time and computational cost of starting from scratch.
- Supports the use of SFS Turbo file system for training data mounting. Intermediate and result data generated by training jobs can be directly written to the SFS Turbo cache, and can be read and processed by downstream business processes. Result data can be asynchronously exported to associated OBS for long-term, low-cost storage, thereby accelerating data access in training scenarios in OBS.

ModelArts Standard Model Training provides convenient job management capabilities, improving the development efficiency of user model training.

- Provides algorithm asset management capabilities, supporting the creation of training jobs through algorithm assets, custom algorithms, and AI Gallery subscribed algorithms, making the creation of training jobs more flexible and user-friendly.
- Provides experiment management capabilities. You often need to adjust datasets and hyperparameters to perform multiple rounds of jobs to select the most ideal one. Model training supports the unified management of multiple training jobs, making it easier for you to choose the best model.

- Provides capabilities such as event information (key event points in the training job lifecycle), training logs (training job runtime and exception information), resource monitoring (resource utilization data), and Cloud Shell (a tool for logging in to training containers), allowing you to have a clearer understanding of the training job runtime process and more accurately troubleshoot and locate issues when encountering task exceptions.

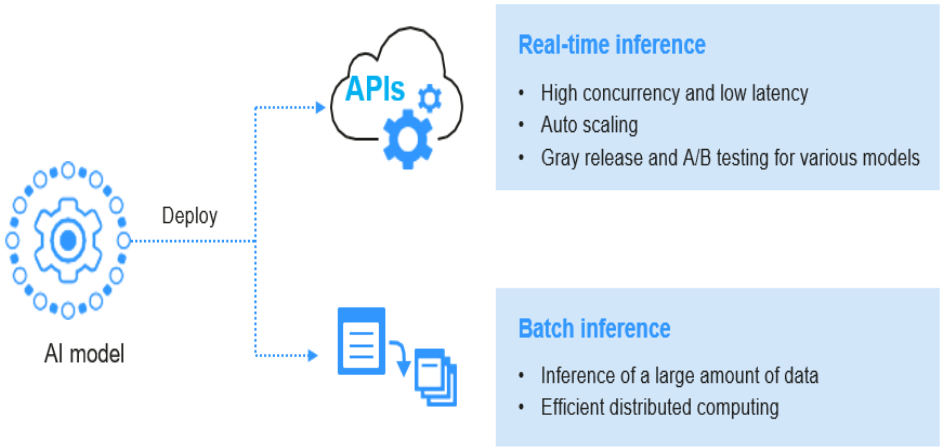
5.1.6 Standard Model Deployment

ModelArts Standard provides capabilities for managing models and services, supporting unified management of images and models with varying frameworks and functions from multiple vendors.

AI model deployment and large-scale implementation are typically complex tasks.

For example, in smart transportation projects, after obtaining a trained model, it needs to be deployed in various scenarios such as cloud, edge, and device. Deploying on the device requires deploying it to cameras of different specifications and vendors, which is a time-consuming and challenging endeavor. ModelArts supports one-click deployment of trained models to various devices and scenarios, including devices, edges, and the cloud. It also provides a comprehensive and reliable one-stop deployment solution for individual developers, enterprises, and device manufacturers.

Figure 5-5 Process of deploying a model



- Real-time inference services enable high concurrency, low latency, elastic scalability, and support multi-model grayscale release and A/B testing.
- It supports deployment in various scenarios, including real-time and batch inference services on the cloud.

5.1.7 Standard Resource Management

When using ModelArts for AI development, you have two options for resource pools:

Dedicated resource pool: A dedicated resource pool provides exclusive compute resources that are not shared with other users, offering better control over resources. You can use the compute resources of the Standard dedicated resource pool for training jobs, model deployment, and development environments on the

ModelArts Standard development platform. Before using it, you need to purchase and create a dedicated resource pool.

Public resource pool: A public resource pool provides a shared large-scale computing cluster that allocates resources based on user job parameters, ensuring job isolation. You can use the public resource pool provided by ModelArts for training jobs, model deployment, and development environment instances. It is billed based on usage and is convenient and efficient. You can directly use the public resource pool without the need to create one.

Differences between the dedicated resource pool and the public resource pool:

- A dedicated resource pool provides you with independent computing clusters and networks, with physical isolation between different users, while the public resource pool only provides logical isolation. The dedicated resource pool has higher isolation and security than the public resource pool.
- Users of a dedicated resource pool have exclusive resources, so jobs will not be queued when resources are sufficient. On the other hand, the public resource pool uses shared resources and may have queues at any time.
- A dedicated resource pool supports accessing the user's network, allowing jobs running in the dedicated resource pool to access storage and resources in the connected network. For example, when creating a training job, if you choose a dedicated resource pool with network connectivity, you can access data in SFS during training.
- A dedicated resource pool supports customizing the physical node's runtime environment, such as GPU/Ascend driver self-upgrade, which is not supported in the public resource pool.

What capabilities does a dedicated resource pool have?

The new version of the dedicated resource pool is a comprehensive improvement in technology and product, with the following main enhancements:

- Unified type of dedicated resource pool: There is no longer a distinction between training and inference dedicated resource pools. If your business allows, you can run both training and inference workloads in the same dedicated resource pool. You can also enable/disable support for specific job types in the dedicated resource pool through job type settings.
- Self-service network connectivity for dedicated pools: You can create and manage the network associated with the dedicated resource pool on the ModelArts management console. If you need to access resources in your VPC for jobs running in a dedicated resource pool, interconnect the VPC with the dedicated resource pool network.
- Improved cluster information: The redesigned dedicated resource pool details page provides more comprehensive cluster information, including job, node, and resource monitoring. This helps you understand the cluster status in a timely manner and better plan resource usage.
- Self-service management of cluster GPU/NPU drivers: Each user has different requirements for cluster drivers. In the new version of the dedicated resource pool list page, you can choose the accelerator card driver yourself and make immediate changes or smooth upgrades according to your service needs.

5.1.8 Supported AI Frameworks in ModelArts Standard

For the development environment Notebook, training jobs, and model inference (model management and deployment), ModelArts Standard supports various AI frameworks and versions. Refer to the following sections.

Unified Image List

ModelArts provides a unified image for Arm+Ascend specifications, including MindSpore and PyTorch. These images are suitable for the Standard development environment, model training, and service deployment. Refer to the table below for more details.

For the URL of the images and the included dependencies, refer to **ModelArts Unified Image List**.

Table 5-1 MindSpore

Preset Image	Supported Chip	Application Scope	Applicable Region
mindspore_2.2.0-cann_7.0.1-py_3.9-euler_2.10.7-aarch64-snt9b	Ascend snt9b	Notebook, training, and inference deployment	CN-Hong Kong

Table 5-2 PyTorch

Preset Image	Supported Chip	Application Scope	Applicable Region
pytorch_2.1.0-cann_7.0.1-py_3.9-euler_2.10.7-aarch64-snt9b	Ascend snt9b	Notebook, training, and inference deployment	CN-Hong Kong
pytorch_1.11.0-cann_7.0.1-py_3.9-euler_2.10.7-aarch64-snt9b	Ascend snt9b	Notebook, training, and inference deployment	CN-Hong Kong

Development Environment Notebook

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

Table 5-3 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/GPU	Yes	Yes
mindspore1.7.0-cuda10.1-py3.7-ubuntu18.04	CPU and GPU general algorithm development and training, preconfigured with AI engine MindSpore1.7.0 and cuda 10.1	CPU/GPU	Yes	Yes
mindspore1.7.0-py3.7-ubuntu18.04	CPU general algorithm development and training, preconfigured with AI engine MindSpore1.7.0	CPU	Yes	Yes
pytorch1.10-cuda10.2-cudnn7-ubuntu18.04	CPU and GPU general algorithm development and training, preconfigured with AI engine PyTorch1.10 and cuda10.2	CPU/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/GPU	Yes	Yes
conda3-ubuntu18.04	Clean user customized base image only include conda	CPU	Yes	Yes

Image	Description	Supported Chip	Remote SSH	Online Jupyter Lab
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/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
conda3-cuda10.2-cudnn7-ubuntu18.04	Clean user customized base image include cuda10.2, 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 supported AI engines and their corresponding versions for training are as follows when creating a training job.

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 5-4 AI engines supported by training jobs

Runtime Environment	CPU Architecture	OS Version	AI Engine and Version	Supported CUDA or Ascend Version
TensorFlow	x86_64	Ubuntu 18.04	tensorflow_2.1.0-cuda_10.1-py_3.7-ubuntu_18.04-x86_64	cuda10.1
PyTorch	x86_64	Ubuntu 18.04	pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64	CUDA 10.2
MPI	x86_64	Ubuntu 18.04	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 Inference

If you import a preset image from a template or OBS to create a model, you can select the AI engines and versions in the table below.

NOTE

- Runtime environments marked as **recommended** is sourced from unified images, which will be the mainstream inference base image in the future. The unified image contains more comprehensive installation packages. For details, see **Base Inference Images**.
- Images of the old version will be discontinued. Use unified images instead.
- The base images to be removed are no longer maintained.
- The naming convention for the unified runtime image is as follows: <AI engine name and version> - <Hardware and version: CPU or CUDA or CANN> - <Python version> - <OS version> - <CPU architecture>.

Table 5-5 Supported AI engines and their runtime environments

Engine	Runtime Environment	Remarks
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**. • The model can run on both CPUs and GPUs when using **python3.6**, **python2.7**, or **tf2.1-python3.7**. 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**. • **python 2.7** and **python 3.6** can only be used to run models applicable to CPU.
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**. • **python 2.7** and **python 3.6** can only be used to run models applicable to CPU.
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**. • **python 2.7** and **python 3.6** can only be used to run models applicable to CPU.

Engine	Runtime Environment	Remarks
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**. • The model can run on both CPUs and GPUs when using **python2.7**, **python3.6**, **python3.7**, **pytorch1.4-python3.7**, or **pytorch1.5-python3.7**. • The default runtime is **python2.7**.
MindSpore	aarch64 (recommended)	**aarch64** can only be used to run models on Snt3 chips.

5.2 Introduction to the Model-as-a-Service (MaaS) Platform

For ordinary businesses, developing large models requires not only powerful computing capabilities but also specialized knowledge of training, deployment, and parameter configuration. ModelArts Studio, the Model-as-a-Service platform (referred to as MaaS), is a customer-oriented platform that provides a user-friendly model development toolchain. It supports custom development of large models, seamlessly integrating model applications with business systems, significantly reducing the cost and difficulty of implementing AI in enterprises.

- **Integration of leading open source large models**

  MaaS integrates leading open source large models, including Llama, Baichuan, Yi, Qwen, AIGC, and more. All models are fully adapted and optimized for Ascend AI cloud services, resulting in improved accuracy and performance. Developers no longer need to build models from scratch; they can simply choose suitable pre-trained models for fine-tuning or direct application, greatly reducing the burden of model integration.

- **Zero-code, configuration-free, and tuning-free model development**

  Based on industry best practices and experience in adapting and tuning open source large models for over 100 customers, the platform provides one-click training, automatic hyperparameter tuning, and highly automated parameter configuration mechanisms. This eliminates the need for manual trial and error in the model optimization process, significantly shortening the cycle from model development to deployment. It ensures high-performance model performance in various applications, allowing customers to focus more on business logic and innovative application design.

- **Accessible resources, pay-per-use billing, scalability, fault recovery, and resumable training**

When enterprises integrate large models into their application systems, they need to consider not only the model experience but also the specific accuracy and cost of the model in practical applications.

MaaS provides flexible model development capabilities and, based on the computing capabilities of Ascend Cloud, offers several key capabilities to ensure the efficient use of large models in customer business applications.

It provides a flexible cost-effective resource allocation plan with pay-per-use billing and scalability, effectively avoiding resource idle time and waste, and reducing the barriers to entry into the AI field.

The architecture emphasizes high availability, with multiple data center deployments ensuring data and task backups. Even in the event of a failure, it seamlessly switches to a standby system, ensuring uninterrupted model training and protecting long-term projects from time and resource losses, thereby ensuring progress and returns.

- **Large model application development, helping developers build intelligent agents quickly**

  In enterprises, complex tasks at the project level often require understanding the task, breaking it down into multiple questions for decision-making, and then calling multiple subsystems to execute. MaaS, based on multiple high-quality Ascend Cloud open source large models, provides high-quality prompt templates, enabling accurate understanding of business intent, decomposition of complex tasks, and the development of multiple intelligent agents. This helps enterprises quickly and intelligently build and deploy large model applications.

5.3 Lite Cluster & Server Introduction

ModelArts Lite is a cloud-native AI computing power cluster that combines hardware and software optimization. It provides an open, compatible, cost-effective, stable, and scalable platform for AI high-performance computing and other scenarios. It has been widely used in areas such as large-scale model training and inference, autonomous driving, AIGC, and content moderation.

ModelArts Lite has two forms:

- ModelArts Lite Server offers different models of xPU bare metal servers. You can access them through EIPs and install relevant drivers and software on the given OS image. You can use SFS or OBS for data storage and retrieval operations, meeting the needs of algorithm engineers for daily training. Refer to **Elastic BMS Lite Server**.
- ModelArts Lite Cluster is tailored for users focused on Kubernetes resources. It offers a managed Kubernetes cluster with mainstream AI development plug-ins and proprietary acceleration plug-ins. This setup provides AI-native resources and task capabilities in a cloud-native manner, allowing you to directly manage nodes and Kubernetes clusters within the resource pool. See **Elastic Kubernetes Cluster**.

ModelArts Lite Cluster supports the following features:

- **Support for servers with different subscription periods in the same Ascend computing resource pool**

In the same Ascend computing resource pool, you can subscribe to different types/billing cycles of resources. This solves the following scenarios:

  - Users cannot scale short-term nodes in a long-term resource pool.
  - Users cannot add pay-per-use nodes (including AutoScaler scenarios) in a yearly/monthly resource pool.

- **Support for SFS product permission partitioning**

  Enabling SFS permission partitioning provides fine-grained access control over mounted SFS folders during training, preventing unauthorized users from accidentally deleting all data.

- **Support for selecting driver versions in the resource pool**

  By selecting the driver version in the resource pool, the issue of all nodes in the resource pool having the same driver version and new nodes not automatically upgrading to that version is resolved. This optimizes the current manual handling process and reduces O&M costs.

- **Support for enabling admission control by default for newly added nodes in the cluster to launch real GPU/NPU detection tasks**

  When the cluster is scaled out, the newly added nodes are set to enable admission control by default. This admission control can also be disabled to improve the success rate of launching real GPU/NPU detection tasks.

5.4 AI Gallery Functions

AI Gallery is an open source community for developers, offering large models to users and advancing the large model industry. It offers a wide range of third-party open source models adapted for Ascend Cloud, along with the ability to quickly experience and develop models, providing developers with an ultimate development experience and helping them quickly understand and learn about large models.

- **Build a zero-threshold online model experience, allowing beginners to use all models with just three lines of code.**

  Through the AI Gallery's online model experience, you can instantly access model services without going through the tedious process of environment configuration. You can intuitively experience the model's effects and quickly try out foundation models, achieving the goal of "instant access, instant experience".

  When you want to develop and train models, AI Gallery provides zero-code development tools for beginners, enabling you to quickly infer and deploy models. For developers with basic coding skills, AI Gallery integrates complex models, data, and algorithm policies to create an efficient collaborative model experience environment, allowing you to call any model with just a few lines of code, significantly simplifying model development.

- **Abundant and powerful compute resources, recommended computing solutions for best practices, improving efficiency and cost-effectiveness.**

  AI Gallery understands the practical difficulties that developers face in the process of advancing AI projects, especially the high costs of model training and deployment, which often hinder the implementation of creative ideas. Through extensive developer practices, AI Gallery has developed the best combination of compute resources for mainstream Ascend Cloud open source

models. It provides developers with the best practice computing solutions, practical guides, and documentation for the final step of model development, saving developers learning and trial-and-error costs, and improving learning and development efficiency.

6 AI Development Basics

6.1 Introduction to the AI Development Lifecycle

What Is AI Development

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

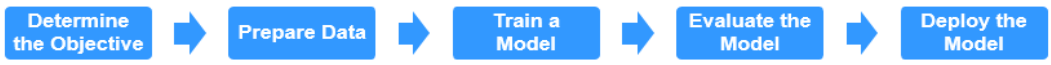
AI development aims to centrally process and extract information from volumes of data to summarize internal patterns of the study objects.

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 6-1 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.

Developers typically build and train models for their services using popular AI frameworks like TensorFlow, PyTorch, and MindSpore.

Step 4 Evaluate the model.

A model generated by training needs to be evaluated. To achieve a good model, initial results often require refinement through repeated adjustments of algorithm settings and data.

Key metrics like accuracy, recall, and AUC enable effective evaluation and optimization.

Step 5 Deploy the model.

Models are trained using existing data, which can include test data. Once a reliable model is obtained, it is applied to real-world data to make predictions and evaluation, and visualize the results. This information helps decision-makers create effective business strategies by presenting complex insights in an easy-to-understand format.

----End

6.2 Basic Concepts of AI Development

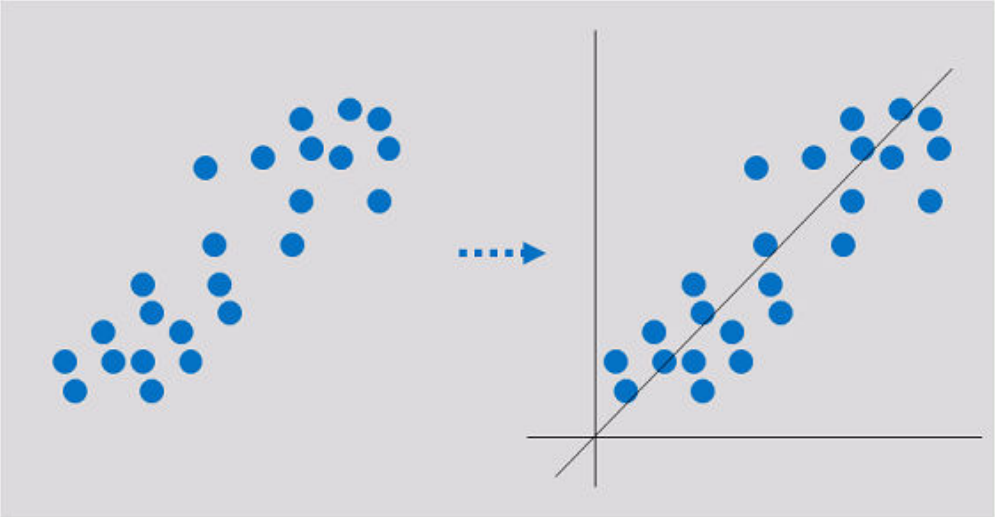
Machine learning is classified into supervised, unsupervised, and reinforcement learning.

- 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

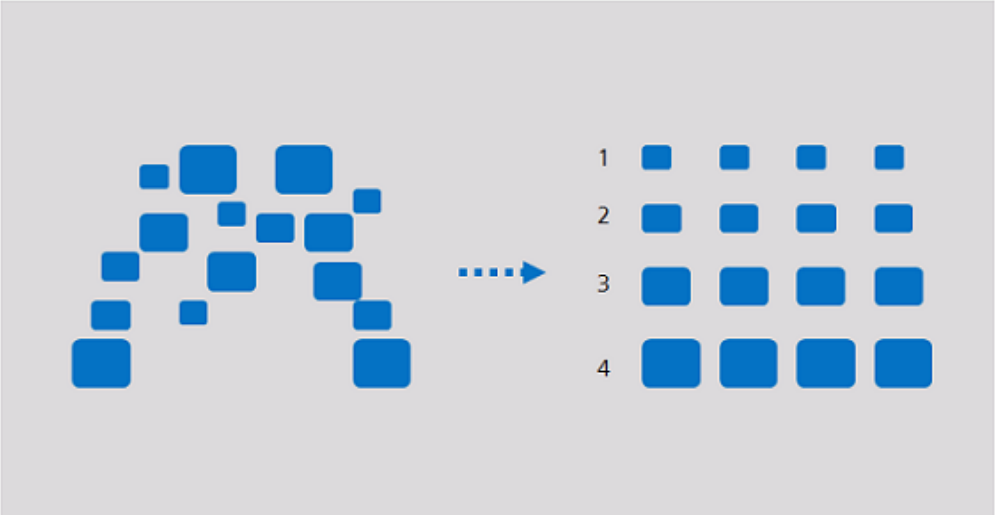
Regression reflects the time feature of data attributes and generates a function that maps one data attribute to an actual variable prediction to find the

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.



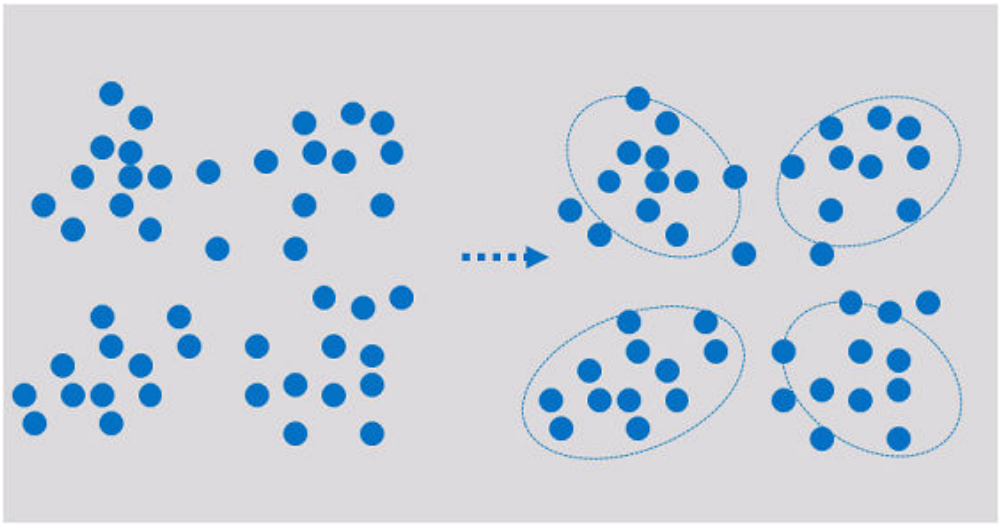
Classification

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.

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

Real-Time Inference

Real-time inference is a web service that synchronously provides inference results for each inference request.

Batch Inference

Batch inference processes batch data for inference.

Ascend Chip

Ascend chips are Huawei-developed AI chips that offer high computing performance while consuming low power.

Resource pool

ModelArts provides large-scale compute clusters for model development, training, and deployment. Both public resource pool and dedicated resource pool are available for you to select.

ModelArts Standard provides public resource pools by default. ModelArts Standard dedicated resource pools are created separately and used exclusively.

Both ModelArts Lite Server and ModelArts Lite Cluster use dedicated resource pools.

MoXing

A lightweight distributed framework developed by the ModelArts team and built on deep learning engines such as TensorFlow, PyTorch, MXNet, and MindSpore. It improves performance of these engines and makes them easier to use. MoXing contains many components. MoXing Framework is a basic common component that can be used to access OBS. It is decoupled from AI engines and can be used in all ModelArts-supported AI engines such as TensorFlow, PyTorch, MXNet, and MindSpore.

MoXing Framework provides common data file operations in OBS, such as reading, writing, listing, creating folders for, querying, moving, copying, and deleting data files.

You can call MoXing APIs in ModelArts notebook instances without downloading or installing the SDKs. Therefore, MoXing is more convenient than ModelArts and OBS SDKs.

7 Security

7.1 Shared Responsibilities

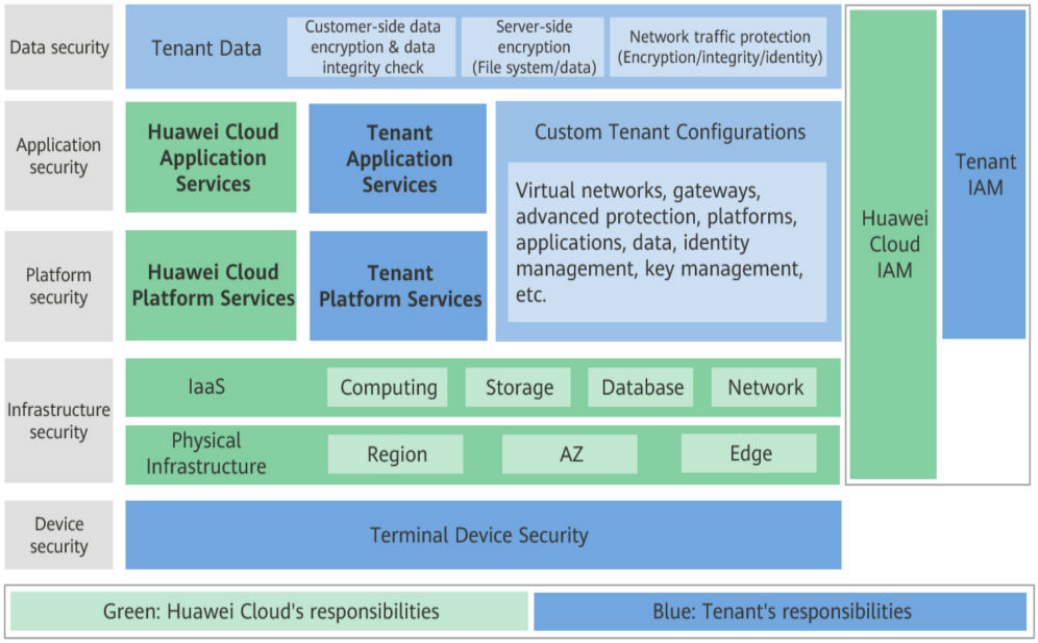
Huawei guarantees that its commitment to cyber security will never be outweighed by the consideration of commercial interests. To cope with emerging cloud security challenges and pervasive cloud security threats and attacks, Huawei Cloud builds a comprehensive cloud service security assurance system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

Figure 7-1 illustrates the responsibilities shared by Huawei Cloud and users.

- **Huawei Cloud**: ensures 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**: uses the cloud securely. Tenants of Huawei Cloud are responsible for the secure and effective management of the internal security as well as the tenant-customized configurations of cloud services, including IaaS, PaaS, and SaaS. This includes but is not limited to operating systems of virtual networks, virtual machine (VM) hosts and guest VMs, virtual firewalls, API Gateway and 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 7-1 Huawei Cloud shared security responsibility model



7.2 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**.

7.3 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 Agency Authorization for ModelArts with One Click**.

- 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.

7.4 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.
Protection for data in transit	ModelArts supports importing models via both HTTP and HTTPS protocols, 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.

7.5 Auditing and Logs

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.

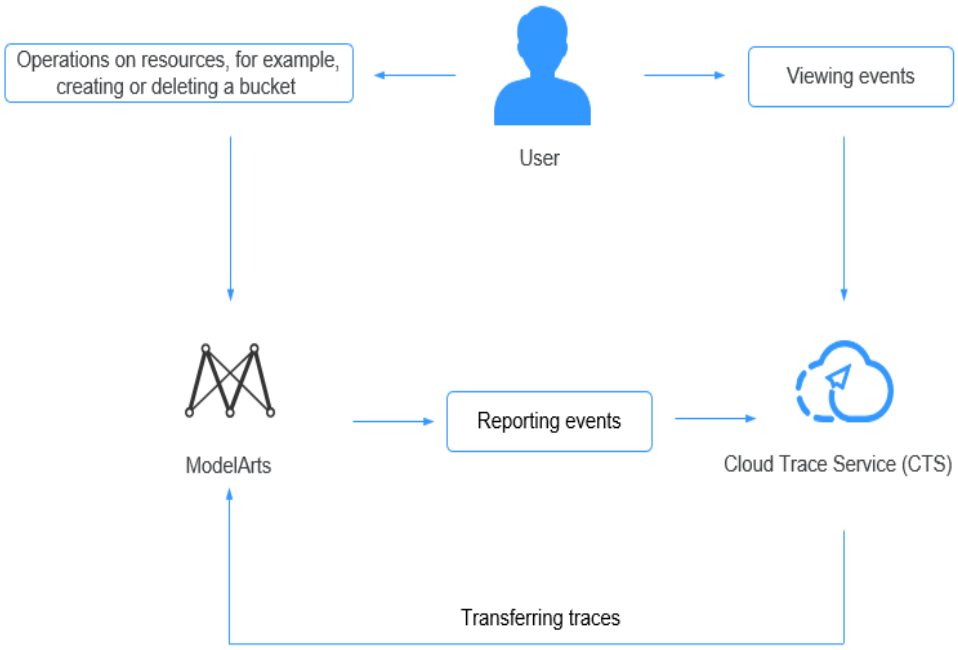
After you enable CTS and configure a trace task, CTS can record management and data traces of ModelArts for auditing.

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 Development**

Environment Operations Traced by CTS, **Key Training Job Operations Traced by CTS**, **Key Model Management Operations Traced by CTS**, and **Key Service Management Operations Traced by CTS**.

Figure 7-2 CTS



Key Data Management Operations Traced by CTS

Table 7-1 Key data management operations traced 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

Operation	Resource Type	Trace
Creating an auto grouping task	dataset	createAutoGroupingTask
Creating an auto deployment task	dataset	createAutoDeployTask
Importing samples to a dataset	dataset	importSamplesToDataset
Creating a dataset label	dataset	createLabel
Updating a dataset label	dataset	updateLabel
Deleting a dataset label	dataset	deleteLabel
Deleting a dataset label and its samples	dataset	deleteLabelWithSamples
Adding samples	dataset	uploadSamples
Deleting samples	dataset	deleteSamples
Stopping an auto labeling task	dataset	stopTask
Creating a team labeling task	dataset	createWorkforceTask
Deleting a team labeling task	dataset	deleteWorkforceTask
Starting the acceptance of a team labeling task	dataset	startWorkforceSampling-Task
Approving, rejecting, or canceling the acceptance of a team labeling task	dataset	updateWorkforceSam-plingTask
Submitting sample review comments for an acceptance task	dataset	acceptSamples
Adding a label to a sample	dataset	updateSamples
Sending an email to team labeling members	dataset	sendEmails
Starting a team labeling task as the team manager	dataset	startWorkforceTask
Updating a team labeling task	dataset	updateWorkforceTask

Operation	Resource Type	Trace
Adding a label to a team-labeled sample	dataset	updateWorkforceTask-Samples
Reviewing team labeling results	dataset	reviewSamples
Creating a labeling team member	workforce	createWorker
Updating labeling team members	workforce	updateWorker
Deleting a labeling team member	workforce	deleteWorker
Deleting labeling team members in batches	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 team labeling member	labelConsoleWorker	workerLoginLabelCon-sole
Logging out of the labeling console as a team labeling member	labelConsoleWorker	workerLogOutLabelCon-sole
Changing the password for logging in to the labeling console as a team labeling member	labelConsoleWorker	workerChangePassword
Handling the issue that the password for logging in to the labeling console as a team labeling member is lost	labelConsoleWorker	workerForgetPassword
Resetting the password for logging in to the labeling console through the URL as a team labeling member	labelConsoleWorker	workerResetPassword

Key Development Environment Operations Traced by CTS

Table 7-2 Key development environment operations traced by CTS

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

Key Training Job Operations Traced by CTS

Table 7-3 Key training job operations traced 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
Configuring a training job	ModelArtsTrainConfig	createModelArtsTrainConfig

Operation	Resource Type	Trace
Modifying training job configurations	ModelArtsTrainConfig	updateModelArtsTrain-Config
Deleting training job configurations	ModelArtsTrainConfig	deleteModelArtsTrain-Config
Creating a visualization job	ModelArtsTensorboard-Job	createModelArtsTensor-boardJob
Deleting a visualization job	ModelArtsTensorboard-Job	deleteModelArtsTensor-boardJob
Modifying the description of a visualization job	ModelArtsTensorboard-Job	updateModelArtsTensor-boardDesc
Stopping a visualization job	ModelArtsTensorboard-Job	stopModelArtsTensor-boardJob
Restarting a visualization job	ModelArtsTensorboard-Job	restartModelArtsgTensor-boardJob

Key Model Management Operations Traced by CTS

Table 7-4 Key model management operations traced by CTS

Operation	Resource Type	Trace
Creating a model	model	addModel
Updating a model	model	updateModel
Deleting a model	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 Traced by CTS

Table 7-5 Key service management operations traced by CTS

Operation	Resource Type	Trace
Deploying a service	service	addService

Operation	Resource Type	Trace
Deleting a service	service	deleteService
Updating a service	service	updateService
Starting or stopping a service	service	startOrStopService
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
Deleting a node from a dedicated resource pool	cluster	deleteClusterNode
Obtaining the result of creating a dedicated resource pool	cluster	createClusterResult

Key AI Gallery Operations Traced by CTS

Table 7-6 Key AI Gallery operations traced 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

Operation	Resource Type	Trace
Modifying user information	ModelArts_Market	update_user

Log

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 seven days on the CTS console.

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

7.6 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

The security suites ModelArts deployed at the host, application, network, and data layers can timely detect security 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.

7.7 Security 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 **Viewing Performance Metrics of a Real-Time Service on Cloud Eye**.

7.8 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.

7.9 Update Management

ModelArts Real-Time Service Upgrade

For a deployed service, you can change the model 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 7-3**.

- Full upgrade

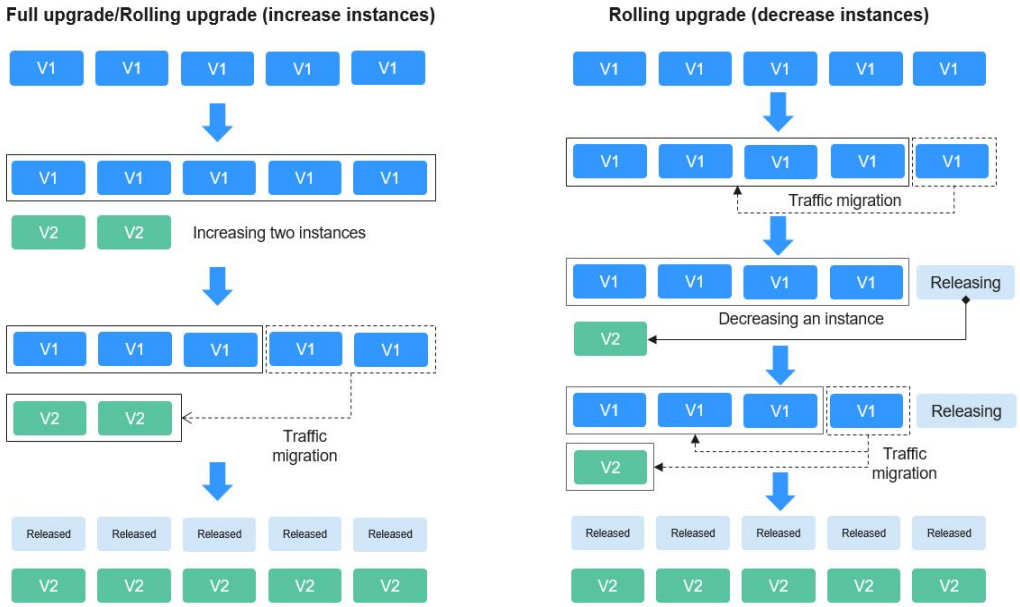
  Twice the number of resources required by the service will be used to create new-version instances in full mode.
- Rolling upgrade (increase instances)

Extra resources will be used for a rolling upgrade. The more the instances, the faster the upgrade.

- Rolling upgrade (decrease instances)

  Certain nodes that were reserved to run services will be used for a rolling upgrade. The more the instances for upgrade, the faster the upgrade, but with a higher probability of service interruption.

Figure 7-3 Service upgrade process



For details about how to upgrade an inference service, see **Modifying a Real-Time Service**.

Image Upgrade

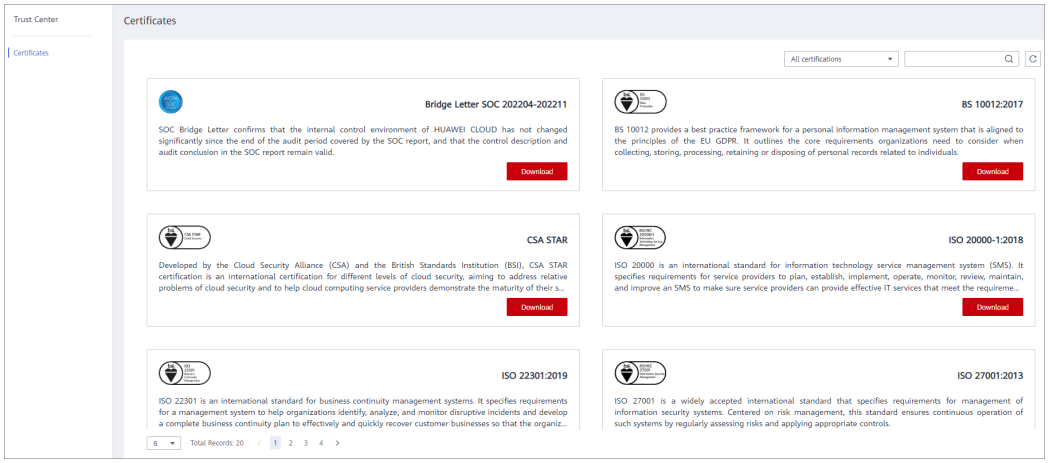
ModelArts provides three function modules: DevEnviron, training management, and inference deployment. The three modules provide base images by the same process. These images are upgraded irregularly to fix vulnerabilities.

7.10 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.

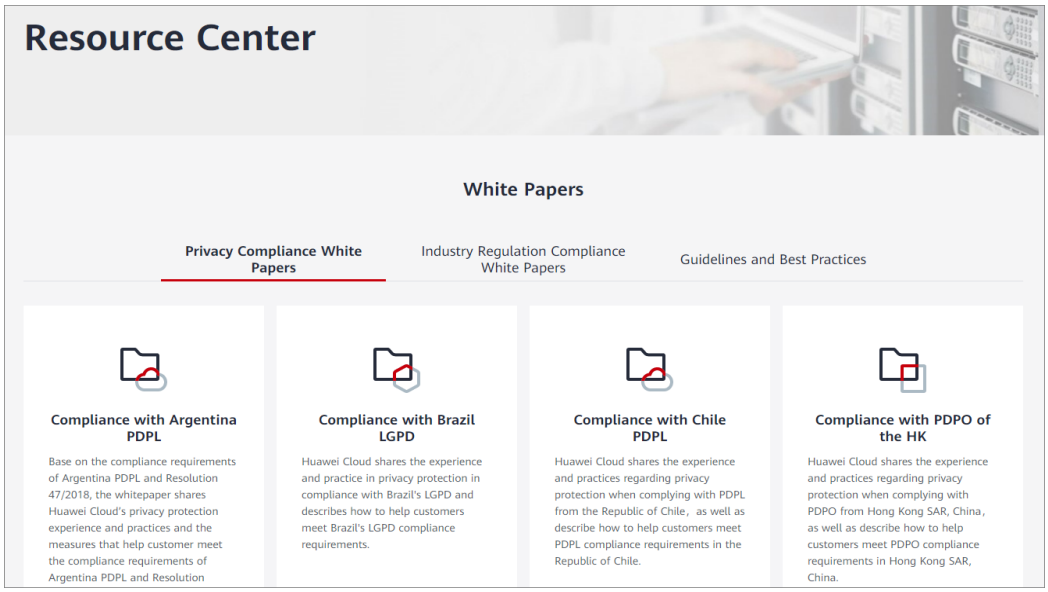
Figure 7-4 Downloading compliance certificates



Resource Center

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

Figure 7-5 Resource center



7.11 Secure Boundaries

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 model 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 containerized 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 models (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.

- Model management

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

  You can select open trusted images when creating models using a container image, for example, images from OpenEuler and Ubuntu. 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. Ensure that it is not hardcoded in the Dockerfile directly. 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.

8 Notes and Constraints

This section describes some limitations and constraints on using ModelArts.

Specifications Restrictions

Table 8-1 Specifications description

Resource Type	Specifications	Description
Compute resources	All compute resource specifications in pay-per-use, yearly/monthly, and package modes, including CPU, GPU, and NPU	All types of compute resources cannot be used across regions.
Compute resources	Package	Packages are used only for public resource pools and cannot be used for dedicated resource pools.

Quota Limits

You can log in to the console to view default quotas. For details, see **Viewing Quotas**.

Table 8-2 Quota

Resource Type	Default Quota	Adjustable	Description
ModelArts Standard notebook instance	A maximum of 10 notebook instances can be created under one account.	No	For more information, see **Creating a Notebook Instance**.

Resource Type	Default Quota	Adjustable	Description
ModelArts Standard real-time service	A maximum of 20 real-time services can be created under one account.	Yes **Submit a service ticket**.	For more information, see **Deploying a Model as a Real-Time Service**.
ModelArts Standard batch service	A maximum of 1,000 batch services can be created under one account.	No	For more information, see **Deploying an AI Application as a Batch Inference Service**.
ModelArts Standard edge service	A maximum of 1,000 edge services can be created under one account.	No	None
ModelArts Standard dedicated resource pool	A maximum of 50 dedicated resource pools can be created under one account.	Yes **Submit a service ticket**.	For more information, see **Creating a Standard Dedicated Resource Pool**.
ModelArts Standard tag	A maximum of 20 tags can be added to a training job, notebook instance, or real-time service.	No	For more information, see **How Does ModelArts Use Tags to Manage Resources by Group?**

Constraints

Table 8-3 Function constraints

Item	Constraints
ModelArts Standard dedicated resource pool	• It is good practice to create no more than 30 nodes at a time. Otherwise, the creation may fail due to traffic limiting. For more information, see **Creating a Standard Dedicated Resource Pool**. • Only pools in the **Running** status can be resized. The number of instances cannot be decreased to 0. • A pool's job types can only be modified while it is running. • A pool's driver can only be upgraded while it is running and there are GPU or Ascend resources in its nodes. • For a logical resource pool, the driver can be upgraded only after node binding is enabled. To enable node binding, submit a service ticket to contact Huawei engineers.

Item	Constraints
ModelArts Standard notebook instance	• Deleted notebook instances cannot be recovered. After a notebook instance is deleted, the data stored in the mounted directory will be deleted. • You can only change an image on a stopped notebook instance. • You can modify a notebook instance's specifications while it is stopped, running, or failed to start. • The target notebook instance must use EVS for storage. If the original capacity of an EVS disk is 4096 GB, the disk capacity cannot be expanded. A maximum of 100 GB can be added at a time. • After the notebook instance is stopped, the expanded EVS capacity still takes effect. The EVS billing is based on the expanded capacity. An EVS disk is billed as long as it is used. To stop billing an EVS disk, delete data from the EVS disk and release the disk. • Images stored in a notebook instance cannot be larger than 35 GB and there cannot be more than 125 image layers. Otherwise, the image cannot be saved.

Item	Constraints
ModelArts Standard training job	• Training logs are retained for only 30 days. To permanently store logs, enable persistent log saving and set a job log path for dumping when creating a training job. For Ascend training, you need to configure the OBS path for storing training logs by default. You need to manually enable **Persistent Log Saving** for training jobs using other resources. • Only dedicated resource pools allow logging in to training containers using Cloud Shell. The training job must be running. • Algorithms subscribed from AI Gallery cannot be saved as new algorithms. • Suspension can be detected only for training jobs that run on GPUs. • The priority can be set for a training job only when a new-version dedicated resource pool is used. The job priority cannot be set for training jobs using a public resource pool or old-version dedicated resource pool. • Only the PyTorch and MindSpore frameworks can be used for distributed training and debugging. If you want to use MindSpore, each node must be equipped with eight cards. • When using a custom image to create a training job, ensure that the custom image size is under 15 GB and does not exceed half of the container engine space in the resource pool. An oversized image affects the startup of a training job. The container engine space of a ModelArts public resource pool is 50 GB. By default, the container engine space of a dedicated resource pool is also 50 GB. You can customize the container engine space when creating a dedicated resource pool. • The **uid** of the default user of a custom training image must be **1000**.
Model Standard inference model	• The maximum size of a model file from OBS is 20 GB. For more information, see **Creating an AI Application**. • If the size of your file exceeds the container engine space, a message will be displayed, indicating that the image space is insufficient. The maximum container engine space in a public resource pool is 50 GB, and that for a dedicated resource pool is 50 GB by default. You can set the container engine space for a dedicated resource pool when you create it, which does not increase costs. For more information, see **Restrictions on the Size of an Image for Importing an AI Application**. • After deploying a model in an ExeML project, it is automatically added to the model list. ExeML-generated models can only be deployed, not downloaded.

Item	Constraints
ModelArts Standard inference service	• Cloud Shell can only access a container when the associated real-time service is deployed within a dedicated resource pool and running. • Batch services can only be deployed in public resource pools. • For models in synchronous request mode, if the prediction request latency exceeds 60 seconds, the request will fail, and there is a possibility that the service may be interrupted. In this case, create an image in asynchronous request mode.
ModelArts Lite Server	• If ModelArts Lite Server runs on BMSs, upgrading or changing the OS kernel or driver can render the driver and kernel incompatible, preventing the OS from starting or making basic functions unavailable. To upgrade or change the OS kernel or driver, **contact Huawei Cloud technical support**. • ModelArts Lite Server running on ECSs does not support OS reinstallation. Similarly, some BMSs in specific regions have this limitation. To reinstall the OS, switch to a different one. • When you reinstall or change the OS of ModelArts Lite Server, the EVS system disk ID changes, which may differ from the original ID in your purchase order. This prevents you from expanding the EVS system disk capacity, resulting in an error message: "The order is expired. The capacity cannot be expanded. Renew the order." You are advised to expand the storage capacity by attaching EVS or SFS disks.
ModelArts Lite Cluster	• Only pools in the **Running** status can be resized. The number of instances cannot be decreased to 0. • You can specify the container engine space size when creating a resource pool. • To change the container engine space for new nodes in an existing Lite Cluster resource pool, set the desired size. The container engine space of existing nodes cannot be modified, as this would create inconsistencies in the dockerBaseSize across nodes with the same flavor within the pool, potentially disrupting task execution on different nodes. • A pool's driver can only be upgraded while it is running and there are GPU or Ascend resources in its nodes. • For a logical resource pool, the driver can be upgraded only after node binding is enabled. To enable node binding, submit a service ticket to contact Huawei engineers.

Item	Constraints
Interaction between ModelArts and OBS	• ModelArts does not support encrypted OBS buckets. When creating an OBS bucket, do not enable bucket encryption. • ModelArts does not support cross-region access to OBS buckets. Ensure that OBS and ModelArts are in the same region.

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 **Configuring Agency Authorization**.

NOTE

Do I need to read this document?

Read this document if any of the following descriptions matches your situation.

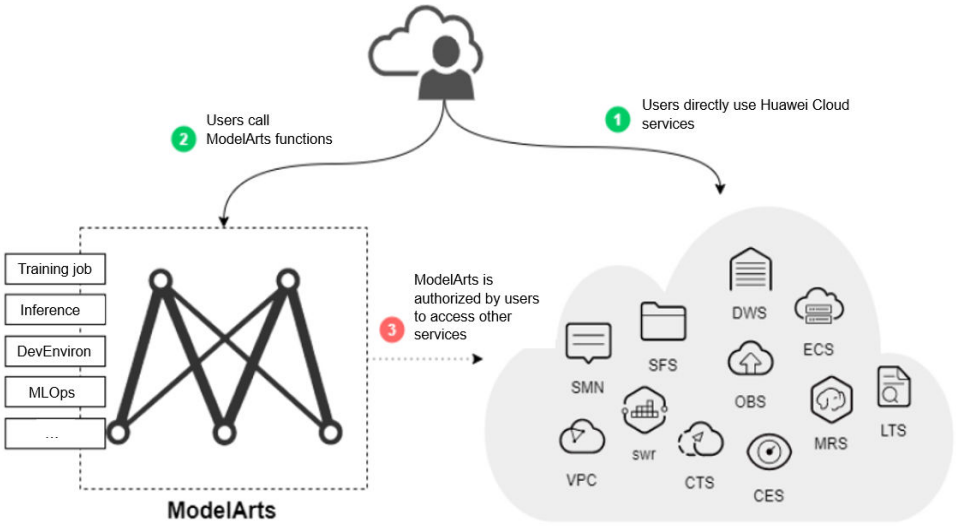
- You are an enterprise user, and
  - There are multiple departments in your enterprise, and you need to control users' permissions so that users in different departments can access only their dedicated resources and functions.
  - There are multiple roles (such as administrators, algorithm developers, and application O&M personnel) in your enterprise. You need them to use only specific functions.
  - There are logically multiple environments (such as the development environment, pre-production environment, and production environment) and are isolated from each other. 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



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.

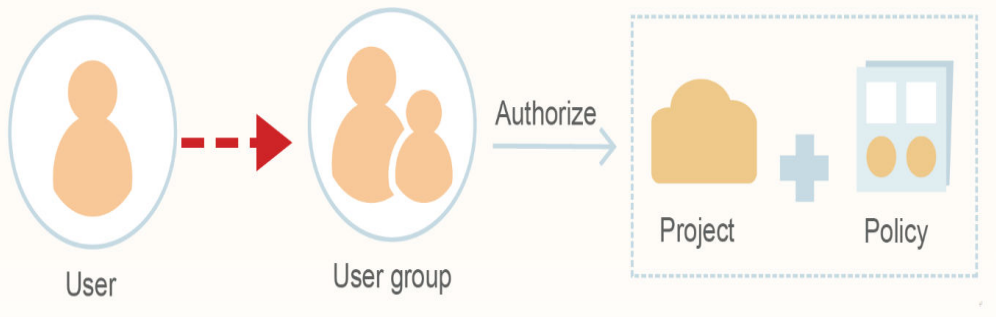
ModelArts Permissions Management

By default, new IAM users do not have any permissions assigned. You need to add the users 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.

ModelArts supports enterprise projects. You can specify an enterprise project when selecting the authorization scope. 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 adds the permissions to a policy and then assigns 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 for common dependent services of 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

As described above, ModelArts must be authorized by users to access other cloud services for AI computing. This authorization is achieved through agencies in the IAM permission system.

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.

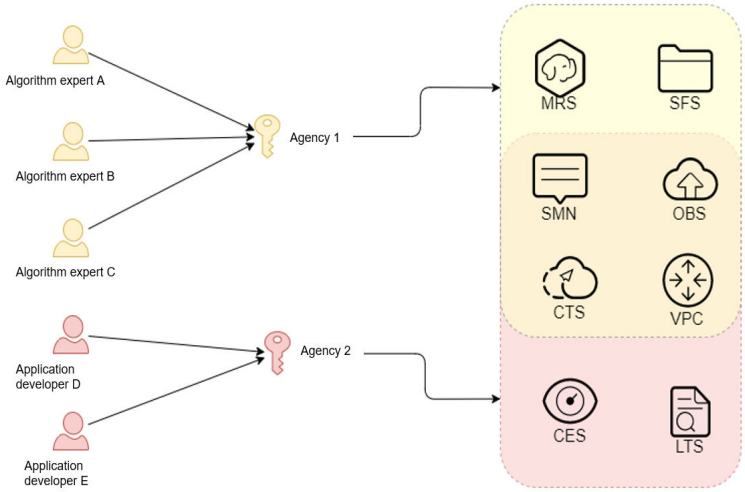
NOTE

- 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

A 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 navigation pane. 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 Billing Description

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 executing.

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

For more information, see **ModelArts Billing Modes**.

11 Quotas

This section describes the quota limits of cloud services related to ModelArts, helping you view and manage your quotas.

What Are Quotas?

A quota defines the maximum number of resources of a certain type that can be created in a region.

To help you save quotas, Huawei Cloud sets limit on the maximum number of cloud resources that you can create in each region.

If the existing resource quotas cannot meet your needs, you can request higher quotas.

Viewing Quotas

Log in to the console to view default quotas. For details, see **Quotas**.

Increasing Quotas

To increase the resource quota, see **How Do I Apply for a Higher Quota?**

Quota Items

To use ModelArts Lite Cluster or Lite Server, you will need more resources than Huawei Cloud's default quotas provided. This includes more ECS instances, memory, CPU cores, and EVS disk space. You will need to request a higher quota to meet these needs. The following table lists quota items.

Table 11-1 Resource quotas involved in ModelArts Lite

Service	Resource Type
ECS resource type	ECS instances
	CPU cores
	RAM capacity (MB)

Service	Resource Type
EIP resources	Bandwidth scaling policies
EVS and SFS resources	Disks
	Disk capacity (GB)
	Snapshots
SFS resources	Capacity limit

12 ModelArts and Other Services

IAM

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

OBS

ModelArts uses Object Storage Service (OBS) to securely and reliably store data and models at low costs. For more information about OBS, see **Object Storage Service Documentation**.

Table 12-1 Relationship between ModelArts and OBS

Function	Task	Relationship
ExeML	Data labeling	The data labeled on ModelArts is stored in OBS.
	Auto training	After a training job is complete, 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. • 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	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.
	Model management	After a training job is completed, the generated model is stored in OBS. You can import the model from OBS.
	Model deployment	Models stored in OBS can be deployed as services.
System management	N/A	ModelArts is authorized 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 details, see **Elastic Volume Service User Guide**.

CCE

ModelArts deploys models as real-time services using Cloud Container Engine (CCE), which supports high concurrency and auto 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 real-time services and model loads in real time and send alarms and notifications automatically. For details about Cloud Eye, see **Cloud Eye User Guide**.

