

Alibaba Cloud

E-MapReduce
SDK Reference

Document Version: 20220615

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
 Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
 Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: If the weight is set to 0, the server no longer receives new requests.
 Note	A note indicates supplemental instructions, best practices, tips, and other content.	 Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings> Network> Set network type .
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click OK .
Courier font	Courier font is used for commands	Run the <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

1.E-MapReduce SDK release notes	05
2.Java SDK	09
2.1. Download SDKs	09
2.2. Reference project	09
2.3. Initialize a client	10
2.4. Sample code	10
3.Python SDK	17
3.1. Install SDK	17
3.2. Sample code	17

1.E-MapReduce SDK release notes

E-MapReduce SDK release notes for each version.

Note

- **emr-core:** Supports the interaction between data sources in OSS and Hadoop/Spark. It exists in the cluster by default. You do not need to package emr-code as part of a job.
- **emr-tablestore:** Supports the interaction between data sources in Table Store and Hadoop/Hive/Spark. Package emr-tablestore in the JAR file.
- **emr-mns_2.10/emr-mns_2.11:** Supports Spark to read data sources in MNS. Package emr-mns_2.10/emr-mns_2.11 in the JAR file.
- **emr-ons_2.10/emr-ons_2.11:** Supports Spark to read data sources in Message Queue (MQ). Package emr-ons_2.10/emr-ons_2.11 in the JAR file.
- **emr-logservice_2.10/emr-logservice_2.11:** Supports Spark to read data sources in Log Service. Package emr-logservice_2.10/emr-logservice_2.11 in the JAR file.
- **emr-maxcompute_2.10/emr-maxcompute_2.11:** Supports Spark to read data sources in MaxCompute. Package emr-maxcompute_2.10/emr-maxcompute_2.11 in the JAR file.

```
<!--Supports interaction with data sources in OSS-->
<dependency>
  <groupId>com.aliyun.emr</groupId>
  <artifactId>emr-core</artifactId>
  <version>1.4.1</version>
</dependency>
<!--Supports interaction with data sources in Table Store
-->
<dependency>
  <groupId>com.aliyun.emr</groupId>
  <artifactId>emr-tablestore</artifactId>
  <version>1.4.1</version>
</dependency>
<!--Supports interaction with data sources in MNS, MQ, Log Service, and MaxCompute (in
the Spark 1.x environment) -->
<dependency>
  <groupId>com.aliyun.emr</groupId>
  <artifactId>emr-mns_2.10</artifactId>
  <version>1.4.1</version>
</dependency>
<dependency>
  <groupId>com.aliyun.emr</groupId>
  <artifactId>emr-logservice_2.10</artifactId>
  <version>1.4.1</version>
</dependency>
<dependency>
  <groupId>com.aliyun.emr</groupId>
  <artifactId>emr-maxcompute_2.10</artifactId>
  <version>1.4.1</version>
```

```
</dependency>
<dependency>
  <groupId>com.aliyun.emr</groupId>
  <artifactId>emr-ons_2.10</artifactId>
  <version>1.4.1</version>
</dependency>
<!--Supports interaction with data sources in MNS, MQ, Log Service, and MaxCompute (in
the Spark 2.x environment)-->
<dependency>
  <groupId>com.aliyun.emr</groupId>
  <artifactId>emr-mns_2.11</artifactId>
  <version>1.4.1</version>
</dependency>
<dependency>
  <groupId>com.aliyun.emr</groupId>
  <artifactId>emr-logservice_2.11</artifactId>
  <version>1.4.1</version>
</dependency>
<dependency>
  <groupId>com.aliyun.emr</groupId>
  <artifactId>emr-maxcompute_2.11</artifactId>
  <version>1.4.1</version>
</dependency>
<dependency>
  <groupId>com.aliyun.emr</groupId>
  <artifactId>emr-ons_2.11</artifactId>
  <version>1.4.1</version>
</dependency>
```

- v1.4.1
 - MaxCompute: Fixes the problem of truncation for the DATETIME values.
 - MaxCompute: Fixes the thread-safety problem for the SimpleDateFormat class.
- v1.4.0
 - MaxCompute: Adds implementation based on the DataSource class. Only versions 2.x and above of Spark are supported.
 - Log Service: Adds implementation based on Direct API. Only versions 2.x and above of Spark are supported.
 - OTS: Optimizes read and write operations.
 - Fixes the bug that the access key of a user is replaced by the access key of a cluster app role when reading data sources in Log Service.
- v1.3.2
 - Fixes some bugs in Table Store.
- v1.3.1
 - Fixes the problem that NullPointerExceptions are thrown in some scenarios when Spark interacts with Log Service.
 - E-MapReduce SDK supports the Spark 2.x environments since this version.
- v1.3.0

- Supports Hadoop MapReduce, Spark, SparkSQL, and Hive to read data in Table Store.
- MNS and Log Service support the MetaService service provided by E-MapReduce. Based on the MetaService service, you can access data in MNS and Log Service without AccessKeys.
- Upgrades some dependencies.
- v1.1.3.1
 - Fixes the problem of dependency conflict between MNS and Spark/Hadoop packages.
 - Fixes the problem that NullPointerExceptions are thrown in some scenarios when Spark Streaming interacts with MNS.
 - Fixes some bugs for the Python SDK.
 - Supports custom time and locations in the scenarios when Spark Streaming integrates with Loghub.
 - Fixes the problem that Hadoop does not support the Snappy native files. Currently, E-MapReduce supports processing the Snappy files that Log Service have archived to OSS.
 - Fixes the problem that Spark does not support the Snappy files.
 - Fixes the problem that OSS does not support the two algorithms of the OutputCommitter class in Apache Hadoop 2.7.2.
 - Optimizes the performance of Hadoop/Spark reading and writing data in OSS.
 - Fixes the problem that a Log4j exception is thrown when Spark prints a job.
- v1.1.2
 - Fixes the problem that the *ConnectionClosedException* is thrown when a job is reading data in OSS.
 - Fixes the problem that some Hadoop commands are not available when accessing OSS data sources.
 - Fixes the *java.text.ParseException: Unparseable date* problem.
 - Optimizes the support of emr-core for local debugging.
 - Interprets the "\$folder\$" files created in the earlier versions as directory paths instead of regular files.
 - Adds a retry mechanism for Hadoop/Spark failing to read data in OSS.
- v1.1.1
 - Fixes the imbalance of disk usage caused by writing temporary files in OSS locally.
 - Removes the "\$folder\$" tag files created during OSS directory creation in a job execution.
- v1.1.0
 - Upgrades the LogHub SDK to 0.6.2. Abandons the Client DB mode and uses the Server DB mode instead.
 - Upgrades the OSS SDK to 2.2.0. Fixes the run-time exceptions caused by the bugs of the OSS SDK.
 - Adds support for MNS.

- Compatibility.
 - For the versions 1.0.x SDKs.
 - Interface:
 - Compatible
 - Namespace:
 - Incompatible: Adjusts the package structure. Modifies the package name from com.aliyun to com.aliyun.emr.
- Modifies the groupId of the project from com.aliyun to com.aliyun.emr. The modified dependency in the POM file is as follows:

```
<dependency>
  <groupId>com.aliyun.emr</groupId>
  <artifactId>emr-sdk_2.10</artifactId>
  <version>1.1.3.1</version>
</dependency>
```

- v1.0.5
 - Optimizes the LoghubUtils interface and parameter input.
 - Optimizes the output format of data in LogStore. Adds the *topic* and the *source* fields.
 - Adds the parameter configurations for the time interval of pulling data in LogStore. Parameter name: spark.logservice.fetch.interval.millis. Default value: 200. Unit: milliseconds.
 - Upgrades the ODPS SDK to 0.20.7-public.
- v1.0.4
 - Downgrades the dependency of Guava to 11.0.2 to avoid a conflict with the dependency of Guava in Hadoop.
 - The MapReduce task supports files greater than 5 GB.
- v1.0.3
 - Adds configuration parameters related to the OSS Client.
- v1.0.2
 - Fixes the bug that the OSS URIs are resolved incorrectly.
- v1.0.1
 - Optimize the settings for OSS URIs.
 - Adds support for MQ.
 - Adds support for Log Service.
 - Supports the Append Object feature of OSS.
 - Supports uploading data in OSS using the multipart upload API.
 - Supports copying data from OSS using the upload part copy mode.

2.Java SDK

2.1. Download SDKs

To use Alibaba Cloud E-MapReduce (EMR) SDK for Java, you must install both the core library of Alibaba Cloud SDK for Java and EMR SDK for Java. The core library package is `aliyun-java-sdk-core`, and the EMR SDK for Java package is `aliyun-java-sdk-emr`. We recommend that you use Maven to manage your projects.

Maven dependencies

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>3.5.0</version>
</dependency>
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-emr</artifactId>
  <version>3.1.0</version>
</dependency>
```

Download links

- [aliyun-java-sdk-core-3.5.0.jar](#)
- [aliyun-java-sdk-emr-3.1.0.jar](#)

2.2. Reference project

If you are still not clear about how to construct a project through the previous steps, or you want to have a project that can be directly imported to an IDE for running, you can download the following reference project code and import it to your IDE environment.

 **Notice** You need to change the information of the AccessKey and security group of this project to yours. For more information, see comments in the sample code.

Eclipse

Eclipse version used by the reference project: Eclipse Java EE IDE for Web Developers. Version: Mars. 1 Release (4.5.1) Build id: 20150924-1200

JDK version: JDK 1.7 or later

- Maven project
[emmavensample.zip](#)
- Normal project
[emrsample.zip](#)

IntelliJ IDEA

IntelliJ IDEA version used by the reference project: IntelliJ IDEA 15.0.1

JDK version: JDK 1.7 or later

- Maven project

[emrmavensample.zip](#)

- Normal project

[emrsample.zip](#)

2.3. Initialize a client

This topic describes how to use E-MapReduce (EMR) SDK for Java to initialize a client.

Background information

Alibaba Cloud provides [OpenAPI Explorer](#) to simplify API usage. You can use OpenAPI Explorer to debug API operations and dynamically generate SDK sample code.

Before you get started

Integrate EMR OpenAPI SDK for Java into your Eclipse project.

- Create a Maven project and add Maven dependencies.

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>3.5.0</version>
</dependency>
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-emr</artifactId>
  <version>3.1.0</version>
</dependency>
```

- Download the required JAR files. In this example, Eclipse is used.
 - i. Download the SDK.

For more information, see [Download links](#).
 - ii. Copy the JAR files that you downloaded to your project folder.
 - iii. Right-click the name of your project in Eclipse and choose **Properties > Java Build Path > Add JARs**.
 - iv. Select the JAR files that you copied in [Step 2](#).

Initialize a client

```
IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<Your-AccessKeyId>", "<Your-AccessKeySecret>");
DefaultAcsClient client = new DefaultAcsClient(profile);
```

2.4. Sample code

This topic describes how to use E-MapReduce (EMR) SDK for Java to perform common operations. For example, you can create clusters, create jobs, and scale in or scale out node groups.

Create a cluster

```
public static void main(String[] args) {
    IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<Your-AccessKeyId>", "<Your-AccessKeySecret>");
    DefaultAcsClient client = new DefaultAcsClient(profile);
    final CreateClusterRequest request = new CreateClusterRequest();
    request.setName("Your-Cluster-Name");
    request.setSecurityGroupId("Your-Security-Group-Id"); // If no security group ID is specified, a new security group with a given name is created.
    request.setAutoRenew(false);
    request.setChargeType("PostPaid"); // Specify the billing method. In this example, the pay-as-you-go billing method is used.
    request.setClusterType("HADOOP"); // Specify the cluster type.
    request.setEmrVer("EMR-1.3.0"); // Specify the EMR version.
    request.setIsOpenPublicIp(true);
    request.setLogEnable(true);
    request.setLogPath("oss://Your-Bucket/Your-Folder");
    request.setMasterPwdEnable(true); // Enable password logon for the master node.
    request.setMasterPwd("Aa123456789"); // Specify the password for the master node.
    request.setZoneId("cn-hangzhou-b"); // Specify the zone ID.
    // The availability of hardware configurations, such as ECS instance types and cloud disk types, is determined based on the specified I/O optimization parameter, ECS instance series, and network type. For more information, visit the ECS buy page.
    request.setIoOptimized(true); // Set this parameter to true to enable I/O optimization.
    request.setInstanceGeneration("ecs-2"); // Set this parameter to ecs-2. Valid values: ecs-1 and ecs-2.
    request.setNetType("vpc"); // Specify the network type.
    request.setVpcId("your-vpcId");
    request.setVSwitchId("your-switchId");
    List<CreateClusterRequest.EcsOrder> ecsOrders = new ArrayList<CreateClusterRequest.EcsOrder>();
    CreateClusterRequest.EcsOrder masterOrder = new CreateClusterRequest.EcsOrder();
    masterOrder.setIndex(1);
    masterOrder.setDiskCapacity(50);
    masterOrder.setDiskCount(2);
    masterOrder.setDiskType("CLOUD_EFFICIENCY"); // Specify the disk type.
    masterOrder.setInstanceType("ecs.n1.large"); // Specify the ECS instance type.
    masterOrder.setNodeCount(1);
    masterOrder.setNodeType("MASTER"); // Specify the node type. In this example, the master node is used.
    ecsOrders.add(masterOrder);
    CreateClusterRequest.EcsOrder coreOrder = new CreateClusterRequest.EcsOrder();
    coreOrder.setIndex(2);
    coreOrder.setDiskCapacity(50);
    coreOrder.setDiskCount(4);
    coreOrder.setDiskType("CLOUD_EFFICIENCY");
    coreOrder.setInstanceType("ecs.n1.large");
    coreOrder.setNodeCount(3);
    coreOrder.setNodeType("CORE");
}
```

```
ecsOrders.add(coreOrder);
request.setEcsOrders(ecsOrders);
try {
    CreateClusterResponse response = client.getAcsResponse(request);
    String clusterId = response.getClusterId(); // cluster id
    // Perform operations on the cluster.
} catch (Exception e) {
}
}
```

- When you create a cluster, specify a security group for the cluster. If you do not specify the ID of a security group, you must specify a security group name. This way, a security group is automatically created during cluster creation.
- Specify whether to enable high availability. For more information, see hardware configurations in [Create a cluster](#).

```
request.setHighAvailabilityEnable(true);
```

- Configure the settings of optional software components. For more information, see software configurations in [Create a cluster](#).

```
List<String> soft = new ArrayList<String>();
soft.add("presto");
soft.add("oozie");
request.setOptionSoftwareLists(soft);
```

- Specify other configuration items. For more information, see [Customize software configurations](#).

```
request.setConfigurations("oss://your-bucket/your-conf.json");
```

- Configure bootstrap actions. For more information, see [Bootstrap actions](#).

```
List<CreateClusterRequest.BootstrapAction> bootstrapActionLists = new ArrayList<CreateClusterRequest.BootstrapAction>();
CreateClusterRequest.BootstrapAction bootstrapActionList = new CreateClusterRequest.BootstrapAction();
bootstrapActionList.setName("bootstrapName");
bootstrapActionList.setPath("oss://emr-agent-pack/bootstrap/run-if.py");
bootstrapActionList.setArg("instance.isMaster=true mkdir -p /tmp/abc");
bootstrapActionLists.add(bootstrapActionList);
request.setBootstrapActions(bootstrapActionLists);
```

Query the details of a cluster

```
public static void main(String[] args) {
    IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<Your-AccessKeyId>", "<Your-AccessKeySecret>");
    DefaultAcsClient client = new DefaultAcsClient(profile);
    final DescribeClusterRequest request = new DescribeClusterRequest();
    request.setId("C-XXXXXXXXXXXXXXXX"); // Specify the ID of the cluster.
    try {
        DescribeClusterResponse response = client.getAcsResponse(request);
        DescribeClusterResponse.ClusterInfo clusterInfo = response.getClusterInfo();
        // Perform operations on the cluster.
    } catch (Exception e) {
    }
}
```

Query the list of clusters

```
public static void main(String[] args) {
    IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<Your-AccessKeyId>", "<Your-AccessKeySecret>");
    DefaultAcsClient client = new DefaultAcsClient(profile);
    final ListClustersRequest request = new ListClustersRequest();
    request.setPageNumber(1);
    request.setIsDesc(true);
    request.setPageSize(20);
    try {
        ListClustersResponse response = client.getAcsResponse(request);
        List<ListClustersResponse.ClusterInfo> clusterInfos = response.getClusters();
        for (ListClustersResponse.ClusterInfo clusterInfo : clusterInfos) {
            // Perform operations on the clusters.
        }
    } catch (Exception e) {
    }
}
```

Release a cluster

```
public static void main(String[] args) {
    IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<Your AccessKeyId>", "<Your AccessKeySecret>");
    DefaultAcsClient client = new DefaultAcsClient(profile);
    ReleaseClusterRequest request = new ReleaseClusterRequest();
    request.setId("C-XXXXXXXXXXXXXXXX"); // Specify the ID of the cluster.
    try {
        ReleaseClusterResponse response = client.getAcsResponse(request);
    } catch (Exception e) {
    }
}
```

Create a job

```
public static void main(String[] args) {
    IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<Your-AccessKeyId>", "<Your-AccessKeySecret>");
    DefaultAcsClient client = new DefaultAcsClient(profile);
    final CreateJobRequest request = new CreateJobRequest();
    request.setName("Your-Job-Name");
    request.setRunParameter("--master yarn-client --driver-memory 4g --executor-memory 4g --executor-cores 2 --num-executors 4 --class com.test.RemoteDebug ossref://Your-Bucket/Resource.jar 1000\");
    request.setFailAct("CONTINUE"); // If the job fails, skip the job and proceed to subsequent operations.
    request.setType("SPARK"); // Specify the job type.
```

```
try {
    CreateJobResponse response = client.getAcsResponse(request);
    String jobId = response.getId();
} catch (Exception e) {
}
}
```

Delete a job

 **Notice** If you want to delete a job that is used for a workflow, you must delete or modify the workflow first.

```
public static void main(String[] args) {
    IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<Your-AccessKeyId>", "<Your-AccessKeySecret>");
    DefaultAcsClient client = new DefaultAcsClient(profile);
    final DeleteJobRequest request = new DeleteJobRequest();
    request.setId("J-XXXXXXXXXXXXXXXXX"); // Specify the ID of the job.
    try {
        DeleteJobResponse response = client.getAcsResponse(request);
    } catch (Exception e) {
    }
}
```

Scale out a node group

You can increase the number of nodes in a node group to scale out the node group.

```
IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "LTAITY****", "2Sfp*****");
DefaultAcsClient client = new DefaultAcsClient(profile);
ResizeClusterV2Request request = new ResizeClusterV2Request();
request.setClusterId("C-0E4B90219*****");
List<ResizeClusterV2Request.HostGroup> hostGroups = new ArrayList<>();
ResizeClusterV2Request.HostGroup hostGroup = new ResizeClusterV2Request.HostGroup();
hostGroups.add(hostGroup);
hostGroup.setHostGroupId("G-F0D0661E0A6E*****");
// Specify the number of nodes that you want to add to the node group.
hostGroup.setNodeCount(1);
request.setHostGroups(hostGroups);
System.out.println(JSON.toJSONString(client.getAcsResponse(request)));
```

Scale in a node group

You can decrease the number of nodes in a node group to scale in the node group. You can also specify the IDs of nodes that you want to remove to scale in a node group.

Decrease the number of nodes in a node group to scale in the node group

 **Notice** Before you scale in a node group by using this method, update the version of EMR SDK for Java to 3.3.8.

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-emr</artifactId>
  <version>3.3.8</version>
</dependency>
```

```
IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "LTAITY****", "2Sfp*****");
DefaultAcsClient client = new DefaultAcsClient(profile);
ReleaseClusterHostGroupRequest request = new ReleaseClusterHostGroupRequest();
# Specify the ID of the cluster.
request.setClusterId("C-01A1F4A*****");
# Specify the ID of the node group. You can call the ListClusterHostGroup operation to obtain the ID of the node group.
request.setHostGroupId("G-D11D3E*****");
// Specify the number of nodes that you want to remove.
request.setReleaseNumber(3);
request.setEnableGracefulDecommission(true);
// Specify the timeout period for graceful decommissioning of nodes. Unit: seconds. If you do not configure this parameter, the default timeout period 3600s is used.
request.setDecommissionTimeout(60);
System.out.println(JSON.toJSONString(client.getAcsResponse(request)));
```

If YARN decommissioning is enabled, change the value of the `yarn.resourcemanager.nodes.exclude-path` parameter to `/etc/ecm/hadoop-conf/yarn-exclude.xml` on the Configure tab of the YARN service in the EMR console. Then, save and deploy the configuration on the Configure tab. You can also modify the following code to change the value of the parameter:

```
IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "LTAITY****", "2Sfp*****");
DefaultAcsClient client = new DefaultAcsClient(profile);
ModifyClusterServiceConfigRequest modifyClusterServiceConfigRequest = new ModifyClusterServiceConfigRequest();
modifyClusterServiceConfigRequest.setClusterId("C-01A1F4A*****");
modifyClusterServiceConfigRequest.setRegionId("cn-hangzhou");
modifyClusterServiceConfigRequest.setServiceName("YARN");
modifyClusterServiceConfigRequest.setConfigParams("{\"yarn-site\":{\"yarn.resourcemanager.nodes.exclude-path\":\"/etc/ecm/hadoop-conf/yarn-exclude.xml\"}}");
modifyClusterServiceConfigRequest.setCustomConfigParams("{}");
modifyClusterServiceConfigRequest.setComment("for decommission gracefully");
modifyClusterServiceConfigRequest.setRefreshHostConfig(Boolean.TRUE);
System.out.println(JSON.toJSONString(client.getAcsResponse(modifyClusterServiceConfigRequest)));
```

Specify the IDs of the nodes that you want to remove to scale in a node group

```
IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", "LTAITY****", "2Sfp*****");
DefaultAcsClient client = new DefaultAcsClient(profile);
ReleaseClusterHostGroupRequest request = new ReleaseClusterHostGroupRequest();
# Specify the ID of the cluster.
request.setClusterId("C-C52CF4246D10****");
# Specify the ID of the node group. You can call the ListClusterHostGroup operation to obtain the ID of the node group.
request.setHostGroupId("G-A24651D939AD****");
// Specify the IDs of the nodes that you want to remove.
List<String> instanceIds = new ArrayList<>();
instanceIds.add("i-xxxxxxx");
instanceIds.add("i-xxxxxxy");
request.setInstanceIdList(JSON.toJSONString(instanceIds));
request.setEnableGracefulDecommission(true);
// Specify the timeout period for graceful decommissioning of nodes. Unit: seconds. If you do not configure this parameter, the default timeout period 3600s is used.
request.setDecommissionTimeout(60);
System.out.println(JSON.toJSONString(client.getAcsResponse(request)));
```

3. Python SDK

3.1. Install SDK

This section describes the installation process of Aliyun E-MapReduce Python 2.7+ SDK.

Install SDK

- You can install the SDK by using pip as follows.

```
sudo pip install aliyun-python-sdk-emr
```

- You can update the E-MapReduce Python SDK as follows.

```
sudo pip install aliyun-python-sdk-emr --upgrade
```

- You can uninstall the SDK as follows.

```
sudo pip uninstall aliyun-python-sdk-emr
```

SDK source code

[Download link](#)

3.2. Sample code

This topic describes how to use E-MapReduce (EMR) SDK for Python to perform common operations. For example, you can create clusters, query clusters, and scale in or scale out node groups.

 **Note** This topic provides sample code for Python 2. If you use Python 3, you can use the sample code in [Common API instructions](#).

Create a cluster

```
#!/usr/bin/python
from aliyunsdkcore import client
from aliyunsdkemr.request.v20160408 import CreateClusterRequest
clt = client.AcsClient('SFAW*****', 'Nc2nZ6dQoiqck0*****', 'cn-hangzhou') # set accessId and accessKey
request = CreateClusterRequest.CreateClusterRequest()
request.set_Name("pydemo")
request.set_ZoneId("cn-hangzhou-b")
request.set_LogEnable(False)
request.set_SecurityGroupId("sg-*****")
request.set_IsOpenPublicIp(True)
request.set_ChargeType("PostPaid")
request.set_EmrVer("EMR-1.3.0")
request.set_ClusterType("HADOOP")
request.set_IoOptimized(True)
request.set_InstanceGeneration("ecs-2")
# set EcsOrder
request.add_query_param('EcsOrder.1.NodeCount', '1')
request.add_query_param('EcsOrder.1.NodeType', 'MASTER')
request.add_query_param('EcsOrder.1.InstanceType', 'ecs.n1.large')
request.add_query_param('EcsOrder.1.DiskType', 'CLOUD_EFFICIENCY')
request.add_query_param('EcsOrder.1.DiskCapacity', '80')
request.add_query_param('EcsOrder.1.DiskCount', '1')
request.add_query_param('EcsOrder.1.Index', '1')
request.add_query_param('EcsOrder.2.NodeCount', '3')
request.add_query_param('EcsOrder.2.NodeType', 'CORE')
request.add_query_param('EcsOrder.2.InstanceType', 'ecs.n1.large')
request.add_query_param('EcsOrder.2.DiskType', 'CLOUD_EFFICIENCY')
request.add_query_param('EcsOrder.2.DiskCapacity', '80')
request.add_query_param('EcsOrder.2.DiskCount', '4')
request.add_query_param('EcsOrder.2.Index', '2')
request.set_accept_format('json')
result = clt.do_action(request)
print result
```



Notice Specific SDKs of Alibaba Cloud services are inconvenient to use because the SDKs of all Alibaba Cloud services are automatically generated. In most scenarios, the SDK for Python does not support input parameters of the LIST type. If you use input parameters of the LIST type, special data processing is required.

- If you set an input parameter to a list that contains values of basic data types, you can view the configurations of the StatusList parameter in the sample code in the [Query clusters](#) section.
- If you set an input parameter to a list that contains values of complex data types, you can view the configurations of the EcsOrder parameter in the sample code in the [Create a cluster](#) section.
- In other scenarios, we recommend that you use the [SDK for Java](#) to process input parameters of the LIST type, such as the BootstrapAction parameter, because SDKs for Java are easier to use.

The methods that are used to call other API operations are similar to the methods that are used to call the preceding operation. For more information, see [List of operations by function](#).

Query clusters

```
#!/usr/bin/python
from aliyunsdkcore import client
from aliyunskemr.request.v20160408 import ListClustersRequest
clt = client.AcsClient('SFAW*****', 'Nc2nZ6dQoiqck0*****', 'cn-hangzhou') # set accessId and accessKey
request = ListClustersRequest.ListClustersRequest()
request.set_accept_format('xml') # xml or json
# Specify filter conditions to obtain the clusters that are in the RUNNING state and the clusters that are in the IDLE state. The StatusList parameter is optional.
request.add_query_param('StatusList.1', 'RUNNING')
request.add_query_param('StatusList.2', 'IDLE')
result = clt.do_action(request)
print result
```

Scale out a node group

You can increase the number of nodes in the node group to scale out a node group.

```
#!/usr/bin/python
from aliyunsdkcore import client
from aliyunskemr.request.v20160408 import ResizeClusterV2Request
clt = client.AcsClient('LTAIT*****', '2SfpK*****', 'cn-hangzhou')
request = ResizeClusterV2Request.ResizeClusterV2Request()
request.set_accept_format('json')
# Specify the ID of the cluster.
request.set_ClusterId("C-01A1F4A*****")
# Specify information about the node group that you want to scale out. You can scale out multiple node groups at the same time. The node groups are counted from 1.
request.add_query_param('HostGroup.1.HostGroupId', 'G-F0D0661E0A6****')
request.add_query_param('HostGroup.1.NodeCount', 1)
#request.add_query_param('HostGroup.2.HostGroupId', 'G-F0D0661E0A6****')
#request.add_query_param('HostGroup.2.NodeCount', 3)
result = clt.do_action(request)
print(result)
```

Scale in a node group

You can decrease the number of nodes in a node group to scale in the node group. You can also specify the IDs of nodes that you want to remove to scale in a node group.

Reduce the number of nodes in a node group to scale the node group



Notice To scale in a node by using this method, run the following command to upgrade the version of EMR SDK:

```
sudo pip install aliyun-python-sdk-emr --upgrade
```

```
#!/usr/bin/python
from aliyunsdkcore import client
from aliyunskemr.request.v20160408 import ReleaseClusterHostGroupRequest
clt = client.AcsClient('LTAIT*****', '2SfpK*****', 'cn-hangzhou')
request = ReleaseClusterHostGroupRequest.ReleaseClusterHostGroupRequest()
request.set_accept_format('json')
# Specify the ID of the cluster.
request.set_ClusterId("C-01A1F4A*****")
# Specify the ID of the node group. You can call the ListClusterHostGroup operation to obtain the ID of the node group.
request.set_HostGroupId("G-D11D3E*****")
# Specify the number of nodes that you want to remove.
request.set_ReleaseNumber(1)
# Enable YARN decommissioning. You can enable YARN decommissioning only for the task node group of a Hadoop cluster.
request.set_EnableGracefulDecommission(True)
request.set_DecommissionTimeout(60)
result = clt.do_action(request)
print(result)
```

If the error message shown in the following figure is returned after you run the preceding code, change the value of the `yarn.resourcemanager.nodes.exclude-path` parameter to `/etc/ecm/hadoop-conf/yarn-exclude.xml` by using one of the following methods:

```
(base) → python sdk python release.py
b"<?xml version='1.0' encoding='UTF-8'?><Error><RequestId>DCAC406A-8CA8-555F-86CB-95EFC80F289</RequestId><HostId>emr.aliyuncs.com</HostId><Code>RequestFailed</Code><Message>FailedReason: InternalError:[Service 'YARN' config 'yarn.resourcemanager.nodes.exclude-path' should equals to '/etc/ecm/hadoop-conf/yarn-exclude.xml' in GRACEFUL scaling in mode</Message><Recommend><![CDATA[https://troubleshoot.api.aliyun.com?q=RequestFailed&product=Emr]]></Recommend></Error>"
```

- Method 1: In the EMR console, search for the `yarn.resourcemanager.nodes.exclude-path` parameter on the Configure tab of the YARN service and change the value to `/etc/ecm/hadoop-conf/yarn-exclude.xml`. Then, save and deploy the configuration on the Configure tab.
- Method 2: Modify the following code to change the value of the `yarn.resourcemanager.nodes.exclude-path` parameter:

```
request = ModifyClusterServiceConfigRequest.ModifyClusterServiceConfigRequest()
request.set_accept_format('json') # The returned result is in JSON format.
request.set_ClusterId("C-01A1F4A*****")
request.set_ServiceName("YARN")
request.set_ConfigParams('{"yarn-site":{"yarn.resourcemanager.nodes.exclude-path":"/etc/ecm/hadoop-conf/yarn-exclude.xml"}}')
request.set_Comment('for decommission gracefully')
request.set_RefreshHostConfig(True)
result = clt.do_action(request)
print(result)
```

Specify the IDs of the nodes that you want to remove to scale in a node group

```
#!/usr/bin/python
from aliynsdcore import client
from aliynsdkemr.request.v20160408 import ReleaseClusterHostGroupRequest
clt = client.AcsClient('LTAIT*****', '2SfpK*****', 'cn-hangzhou')
request = ReleaseClusterHostGroupRequest.ReleaseClusterHostGroupRequest()
request.set_accept_format('json')
# Specify the ID of the cluster.
request.set_ClusterId("C-01A1F4A*****")
# Specify the ID of the node group. You can call the ListClusterHostGroup operation to obtain the ID of the node group.
request.set_HostGroupId("G-D11D3E*****")
# Specify the IDs of nodes that you want to remove. If you configure both the InstanceIdList parameter and the ReleaseNumber parameter, the configuration of the InstanceIdList parameter prevails.
request.set_InstanceIdList(["i-1**", "i-2**"])
# Enable YARN decommissioning. You can enable YARN decommissioning only for the task node group of a Hadoop cluster.
request.set_EnableGracefulDecommission(True)
# Specify the timeout period for graceful decommissioning of nodes. Unit: seconds. If you do not configure this parameter, the default timeout period 3600s is used.
request.set_DecommissionTimeout(60)
result = clt.do_action(request)
print(result)
```