

阿里云 服务器迁移中心

SDK示例

文档版本：20200706

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告： 重启操作将导致业务中断，恢复业务时间约十分钟。
	用于警示信息、补充说明等，是用户必须了解的内容。	 注意： 权重设置为0，该服务器不会再接受新请求。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明： 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	单击 设置 > 网络 > 设置网络类型 。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面，单击 确定 。
Courier字体	命令。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid Instance_ID</code>
[]或者[a b]	表示可选项，至多选择一个。	<code>ipconfig [-all]-t</code>
{ }或者[a b]	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

法律声明	1
通用约定	1
1 SDK概述	1
2 Java示例	2
2.1 安装Java SDK.....	2
2.2 迁移源.....	2
2.2.1 查询迁移源信息.....	2
2.2.2 删除迁移源.....	4
2.2.3 修改迁移源名称和描述.....	5
2.3 迁移任务.....	6
2.3.1 创建迁移任务.....	6
2.3.2 删除迁移任务.....	8
2.3.3 查询迁移任务.....	9
2.3.4 修改迁移任务信息.....	10
2.3.5 启动迁移任务.....	12
2.3.6 停止迁移任务.....	13
2.4 一次性迁移服务器.....	14
3 Python示例	19
3.1 安装Python SDK.....	19
3.2 一次性迁移服务器.....	19

1 SDK概述

服务器迁移中心SMC支持Java SDK和Python SDK开发。

SDK列表

您可以在GitHub仓库查看SDK源码。

编程语言	GitHub仓库地址	
Java	aliyun-java-sdk-smc	
Python	aliyun-python-sdk-smc	

2 Java示例

2.1 安装Java SDK

本文介绍如何安装服务器迁移中心SMC Java SDK。

环境准备

安装SMC Java SDK必须用1.6或更高版本JDK。

安装SDK

在Maven工程中使用SMC Java SDK，只需在pom.xml中加入相应依赖即可。在<dependencies>内加入如下内容：

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <optional>true</optional>
  <version>4.3.2</version>
</dependency>
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-smc</artifactId>
  <version>1.0.1</version>
</dependency>
```

2.2 迁移源

2.2.1 查询迁移源信息

本文介绍如何通过SMC Java SDK调用DescribeSourceServers查询一个或多个迁移源信息。

背景信息

您可以调用DescribeSourceServers接口批量查询迁移源信息，包括迁移任务ID、内核版本级别、迁移源系统盘大小等详细信息，详情请参见[#unique_8](#)。

代码示例

以下代码适用于查询迁移状态为InUse（迁移中）的迁移源信息：

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.smc.model.v20190601.DescribeSourceServersRequest;
import com.aliyuncs.smc.model.v20190601.DescribeSourceServersResponse;
import com.google.gson.Gson;
```

```
import java.util.ArrayList;
import java.util.List;

public class DescribeSourceServers {
    public static void main(String[] args) {
        //初始化AcsClient实例, 设置地域ID及您的AccessKey信息
        DefaultProfile profile = DefaultProfile.getProfile("<region-Id>", "<accessKeyId>", "<
accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);

        //创建API请求, 并设置参数
        DescribeSourceServersRequest request = new DescribeSourceServersRequest();
        //迁移源ID
        List<String> sourceIds = new ArrayList<>();
        sourceIds.add("s-bp152mkfswviw7sd****");
        //sourceIds.add("s-bp152mkfswviw7sd****");
        request.setSourceIds(sourceIds);
        //返回的迁移源列表的页码
        request.setPageNumber(1);
        //分页查询时设置的每页行数
        request.setPageSize(10);
        //迁移源状态
        request.setState("InUse");
        //迁移源名称
        //request.setName("smcName");
        //迁移任务ID
        //request.setJobId("j-bp152mkfswviwbqj****");

        //发送请求获取返回值或处理异常
        try {
            DescribeSourceServersResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

预期结果

```
{
  "requestId": "766829A6-BC4D-47D2-A0C6-08870FEA785B",
  "totalCount": 1,
  "pageNumber": 1,
  "pageSize": 10,
  "sourceServers": [{
    "sourceId": "s-bp152mkfswviw7sd****",
    "jobId": "j-bp152mkfswviwbqj****",
    "agentVersion": "2.0.5",
    "name": "smcName",
    "description": "Server Source Imported By GotoAliyun. Hostname: smcName IP
Address: 192.***.***.29",
    "kernelLevel": 0,
    "platform": "CentOS",
    "architecture": "x86_64",
    "systemDiskSize": 40,
    "replicationDriver": "SMT",
    "systemInfo": "{\"agent_mode\":\"daemon\",\"agent_type\":\"aliyun\",\"client_typ
e\":\"\",\"cores\":\"2\",\"cpu_info\":\"\",\"cpu_usage\":\"1.99\",\"hostname\":\"
smcName\",\"ipV4\":\"192.***.***.29\",\"memory\":\"1.00\",\"memory_usage\":\"0.38
\"}"
  }],
}
```

```
"creationTime": "2020-01-08T06:38:20Z",
"state": "InUse",
"errorCode": "SourceServer.Offline",
"heartbeatRate": 30,
"dataDisks": []
}}
}
```

2.2.2 删除迁移源

本文介绍如何通过服务器迁移工具SMC Java SDK调用DeleteSourceServer删除一个迁移源。

背景信息

您可以调用DeleteSourceServer接口删除一个迁移源，详情请参见[#unique_10](#)。



说明：

- 迁移源若关联了Running（运行中）状态的迁移任务，则无法删除。
- 迁移源若关联了除Running状态外的其他状态的迁移任务，则需将参数Force设置为true来强制删除迁移源。

代码示例

以下代码适用于强制删除迁移源：

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.smc.model.v20190601.DeleteSourceServerRequest;
import com.aliyuncs.smc.model.v20190601.DeleteSourceServerResponse;
import com.google.gson.Gson;

public class DeleteSourceServer {
    public static void main(String[] args) {
        //初始化AcsClient实例，设置地域ID及您的AccessKey信息
        DefaultProfile profile = DefaultProfile.getProfile("<region-Id>", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);

        //创建API请求，并设置参数
        DeleteSourceServerRequest request = new DeleteSourceServerRequest();
        //迁移源ID
        request.setSourceId("s-bp152mkfswviw7sd****");
        //是否强制删除迁移源
        request.setForce(true);

        //发送请求获取返回值或处理异常
        try {
            DeleteSourceServerResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```



```
}  
}  
}
```

预期结果

```
{  
  "RequestId": "0A01347D-887F-4816-85B3-2BB8A298AA27"  
}
```

2.2.3 修改迁移源名称和描述

本文介绍通过SMC Java SDK调用ModifySourceServerAttribute修改迁移源的名称和描述。

背景信息

您可以调用ModifySourceServerAttribute接口修改迁移源的名称和描述，详情请参见[#unique_12](#)。

代码示例

以下代码适用于修改迁移源的名称和描述：

```
import com.aliyuncs.DefaultAcsClient;  
import com.aliyuncs.IAcsClient;  
import com.aliyuncs.exceptions.ClientException;  
import com.aliyuncs.exceptions.ServerException;  
import com.aliyuncs.profile.DefaultProfile;  
import com.aliyuncs.smc.model.v20190601.ModifySourceServerAttributeRequest;  
import com.aliyuncs.smc.model.v20190601.ModifySourceServerAttributeResponse;  
import com.google.gson.Gson;  
  
public class ModifySourceServerAttribute {  
    public static void main(String[] args) {  
        //初始化AcsClient实例，设置地域ID及您的AccessKey信息  
        DefaultProfile profile = DefaultProfile.getProfile("<region-Id>", "<accessKeyId>", "<accessSecret>");  
        IAcsClient client = new DefaultAcsClient(profile);  
  
        //创建API请求，并设置参数  
        ModifySourceServerAttributeRequest request = new ModifySourceServerAttributeRequest();  
        //迁移源ID  
        request.setSourceId("s-bp152mkfswviw7sd****");  
        //迁移源名称  
        request.setName("MySourceServer");  
        //迁移源描述  
        request.setDescription("This is a source server.");  
  
        //发送请求获取返回值或处理异常  
        try {  
            ModifySourceServerAttributeResponse response = client.getAcsResponse(request);  
            System.out.println(new Gson().toJson(response));  
        } catch (ServerException e) {  
            e.printStackTrace();  
        } catch (ClientException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
}
```

预期结果

```
{  
  "RequestId": "78A76F74-CA05-4B34-83D2-60F30311B5AD"  
}
```

2.3 迁移任务

2.3.1 创建迁移任务

本文介绍如何通过SMC Java SDK调用CreateReplicationJob为迁移源创建一个迁移任务。

背景信息

您可以通过CreateReplicationJob接口为迁移源创建一个迁移任务，详情请参见[#unique_15](#)。



说明：

- 您只能为在线（Available）状态的迁移源创建迁移任务。
- 每个迁移源仅能关联一个未完成状态的迁移任务。未完成状态包括Ready（未开始）、Running（运行中）、Stopped（已暂停）、Waiting（等待中）、InError（出错）和Expired（已过期）。
- 每个阿里云账号可创建100个迁移任务。
- 迁移目标类型为镜像时，需指定ImageName、SystemDiskSize、DataDisk参数。
- 使用VPC内网迁移时，VSwitchId参数为必填，VpcId参数为可选。

代码示例

以下代码用于创建一个已指定VPC、目标ECS实例系统盘大小为80 GiB的迁移任务，并且为迁移任务绑定usage:production标签：

```
import com.aliyuncs.DefaultAcsClient;  
import com.aliyuncs.IAcsClient;  
import com.aliyuncs.exceptions.ClientException;  
import com.aliyuncs.exceptions.ServerException;  
import com.aliyuncs.profile.DefaultProfile;  
import com.aliyuncs.smc.model.v20190601.CreateReplicationJobRequest;  
import com.aliyuncs.smc.model.v20190601.CreateReplicationJobResponse;  
import com.google.gson.Gson;  
import java.util.ArrayList;  
import java.util.List;  
  
public class CreateReplicationJob {  
    public static void main(String[] args) {  
        //初始化AcsClient实例，设置地域ID及您的AccessKey信息  
        DefaultProfile profile = DefaultProfile.getProfile("<region-Id>", "<accessKeyId>", "<accessSecret>");  
        IAcsClient client = new DefaultAcsClient(profile);
```

```
//创建API请求, 并设置参数
CreateReplicationJobRequest request = new CreateReplicationJobRequest();
//迁移源ID
request.setSourceId("s-bp152mkfswviw7sd****"); //专有网络VPC的ID
request.setVpcId("vpc-bp1vwnn14rppyicz****");
//指定VPC下的交换机ID
request.setVSwitchId("vsw-bp1ddbrxdlrcbim46****");
//目标阿里云服务器ECS的系统盘大小, 单位为GiB
request.setSystemDiskSize(80);
//创建一次性迁移任务还是增量迁移任务
request.setRunOnce(true);
//迁移任务的标签键值对
List<CreateReplicationJobRequest.Tag> tags = new ArrayList<>();
CreateReplicationJobRequest.Tag tag = new CreateReplicationJobRequest.Tag();
tag.setKey("usage");
tag.setValue("production");
tags.add(tag);
request.setTags(tags);
//迁移任务名
//request.setName("MySourceServer");
//迁移任务描述
//request.setDescription("This_is_a_migration_task");
//迁移任务的执行时间
//request.setScheduledStartTime("2019-06-04T13:35:00Z");
//迁移任务的过期时间
//request.setValidTime("2019-06-04T13:35:00Z");
//目标实例ID
//request.setInstanceId("i-bp12uyz75shn6jsp****");
//迁移任务交付的目标阿里云镜像名称
//request.setImageName("smc-image");
//保证请求幂等性, 您可以从客户端生成一个不超过64个ASCII字符的参数值, 并将值赋予
ClientToken
//request.setClientToken(UUID.randomUUID().toString());
//增量迁移任务运行的时间间隔
//request.setFrequency(12);
//目标阿里云服务器ECS的数据盘顺序和大小
//List<CreateReplicationJobRequest.DataDisk> datadisks = new ArrayList<>();
//CreateReplicationJobRequest.DataDisk datadisk = new CreateReplicationJob
Request.DataDisk();
//目标阿里云服务器ECS的数据盘顺序
//datadisk.setIndex(1);
//目标阿里云服务器ECS的数据盘大小
//datadisk.setSize(100);
//datadisks.add(datadisk);
//request.setDataDisks(datadisks);
//增量迁移任务默认保留的最大镜像数
//request.setMaxNumberOfImageToKeep(10);
//数据传输网络模式
//request.setNetMode(0);
//复制驱动器的参数信息
//request.setReplicationParameters("{\"bandwidth_limit\":0,\"compress_level\":1,\"
checksum\":true}");
//迁移任务交付的目标类型
//request.setTargetType("Image");

//发送请求获取返回值或处理异常
try {
    CreateReplicationJobResponse response = client.getAcsResponse(request);
    System.out.println(new Gson().toJson(response));
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
```

```
}  
}  
}
```

预期结果

```
{  
  "RequestId":"C8B26B44-0189-443E-9816-D951F59623A9",  
  "JobId":"j-bp152mkfswviwbqj****"  
}
```

2.3.2 删除迁移任务

本文介绍如何通过SMC Java SDK调用DeleteReplicationJob删除一个迁移任务。

背景信息

您可以通过DeleteReplicationJob接口删除一个迁移任务，详情请参见[#unique_17](#)。



说明：

- 迁移任务删除后不可恢复。
- 迁移任务删除后自动释放已创建的相关资源，如中转实例等。

代码示例

以下代码适用于删除迁移任务：

```
import com.aliyuncs.DefaultAcsClient;  
import com.aliyuncs.IAcsClient;  
import com.aliyuncs.exceptions.ClientException;  
import com.aliyuncs.exceptions.ServerException;  
import com.aliyuncs.profile.DefaultProfile;  
import com.aliyuncs.smc.model.v20190601.DeleteReplicationJobRequest;  
import com.aliyuncs.smc.model.v20190601.DeleteReplicationJobResponse;  
import com.google.gson.Gson;  
  
public class DeleteReplicationJob {  
    public static void main(String[] args) {  
        //初始化AcsClient实例，设置地域ID及您的AccessKey信息  
        DefaultProfile profile = DefaultProfile.getProfile("<region-Id>", "<accessKeyId>", "<accessSecret>");  
        IAcsClient client = new DefaultAcsClient(profile);  
  
        //创建API请求，并设置参数  
        DeleteReplicationJobRequest request = new DeleteReplicationJobRequest();  
        //迁移任务ID  
        request.setJobId("j-bp152mkfswviwbqj****");  
  
        //发送请求获取返回值或处理异常  
        try {  
            DeleteReplicationJobResponse response = client.getAcsResponse(request);  
            System.out.println(new Gson().toJson(response));  
        } catch (ServerException e) {  
            e.printStackTrace();  
        } catch (ClientException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
}  
}  
}
```

预期结果

```
{  
  "RequestId": "473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E"  
}
```

2.3.3 查询迁移任务

本文介绍如何通过SMC Java SDK调用DescribeReplicationJobs查询一个或多个迁移源信息。

背景信息

您可以调用DescribeReplicationJobs接口批量查询迁移源信息，包括迁移时使用的网络类型、迁移任务总进度、迁移中转实例ID等详细信息，详情请参见[#unique_19](#)。

代码示例

以下代码适用于查询迁移源信息：

```
import com.aliyuncs.DefaultAcsClient;  
import com.aliyuncs.IAcsClient;  
import com.aliyuncs.exceptions.ClientException;  
import com.aliyuncs.exceptions.ServerException;  
import com.aliyuncs.profile.DefaultProfile;  
import com.aliyuncs.smc.model.v20190601.DescribeReplicationJobsRequest;  
import com.aliyuncs.smc.model.v20190601.DescribeReplicationJobsResponse;  
import com.google.gson.Gson;  
import java.util.ArrayList;  
import java.util.List;  
  
public class DescribeReplicationJobs {  
    public static void main(String[] args) {  
        //初始化AcsClient实例，设置地域ID及您的AccessKey信息  
        DefaultProfile profile = DefaultProfile.getProfile("<region-Id>", "<accessKeyId>", "<accessSecret>");  
        IAcsClient client = new DefaultAcsClient(profile);  
  
        //创建API请求，并设置参数  
        DescribeReplicationJobsRequest request = new DescribeReplicationJobsRequest();  
        //设置迁移任务的页码  
        request.setPageNumber(1);  
        //设置每页行数  
        request.setPageSize(10);  
        //设置迁移源ID列表  
        List<String> sourceIds = new ArrayList<>();  
        sourceIds.add("s-bp152mkfswviw7sd****");  
        request.setSourceIds(sourceIds);  
        //设置迁移任务的名称  
        //request.setName("MigrationTask");  
        //设置迁移任务的业务状态  
        //request.setBusinessStatus("Cleaning");  
        //设置迁移任务ID列表  
        //List<String> jobIds = new ArrayList<>();  
        //jobIds.add("j-bp152mkfswviwbqj****");
```

```
//request.setJobIds(jobIds);
//设置迁移任务的主状态
//request.setStatus("InError");

//发送请求获取返回值或处理异常
try {
    DescribeReplicationJobsResponse response = client.getAcResponse(request);
    System.out.println(new Gson().toJson(response));
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

预期结果

```
{
  "requestId": "55E622AC-8E2A-432A-93FB-ABA049CA3E9B",
  "totalCount": 1,
  "pageNumber": 1,
  "pageSize": 10,
  "replicationJobs": [{
    "jobId": "j-bp152mkfswviwbqj****",
    "sourceId": "s-bp152mkfswviw7sd****",
    "name": "MigrationTask",
    "description": "smcDescription",
    "regionId": "cn-hangzhou",
    "targetType": "Image",
    "imageId": "m-bp18h4rkl1g4as4d****",
    "status": "Finished",
    "businessStatus": "Cleaning",
    "errorCode": "",
    "progress": 100.0,
    "creationTime": "2020-01-08T06:44:04Z",
    "validTime": "2020-02-07T06:44:05Z",
    "startTime": "2020-01-08T06:44:05Z",
    "endTime": "2020-01-08T08:33:54Z",
    "netMode": 0,
    "systemDiskSize": 80,
    "vpcId": "vpc-bp1vwnn14rqpyiczj****",
    "vSwitchId": "vsw-bp1ddbrxdlrcbim46****",
    "transitionInstanceId": "i-bp1a1ww07yiz0vzn****",
    "runOnce": true,
    "dataDisks": [],
    "replicationJobRuns": [{
      "imageId": "m-bp18h4rkl1g4as4d****",
      "type": "Manual",
      "startTime": "2020-01-08T06:44:05Z",
      "endTime": "2020-01-08T08:32:25Z"
    }
  ]
}
```

2.3.4 修改迁移任务信息

本文介绍如何通过SMC Java SDK调用ModifyReplicationJobAttribute修改迁移任务信息。

背景信息

您可以调用ModifyReplicationJobAttribute接口修改迁移任务信息，包括修改迁移任务交付的目标镜像名称、目标实例ID、增量迁移任务默认保留的最大镜像数等信息，详情请参见[#unique_21](#)。



说明：

- 参数Name和Description在迁移任务的整个生命周期内均可以修改。
- 参数Frequency和MaxNumberOfImageToKeep只能在迁移任务执行前或任务状态为等待中时修改。
- 其他参数只能在迁移任务执行前修改。

代码示例

以下代码适用于修改迁移任务名称、描述及目标ECS实例系统盘大小的信息：

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.smc.model.v20190601.ModifyReplicationJobAttributeRequest;
import com.aliyuncs.smc.model.v20190601.ModifyReplicationJobAttributeResponse;
import com.google.gson.Gson;

public class ModifyReplicationJobAttribute {
    public static void main(String[] args) {
        //初始化AcsClient实例，设置地域ID及您的AccessKey信息
        DefaultProfile profile = DefaultProfile.getProfile("<region-Id>", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);

        //创建API请求，并设置参数
        ModifyReplicationJobAttributeRequest request = new ModifyReplicationJobAttributeRequest();
        //迁移任务ID
        request.setJobId("j-bp152mkfswviwbqj****");
        //迁移任务名称
        request.setName("MyMigrationTask");
        //迁移任务描述
        request.setDescription("This is my migration task");
        //目标阿里云服务器ECS的系统盘大小
        request.setSystemDiskSize(50);
        //迁移任务的执行时间
        //request.setScheduledStartTime("2019-06-04T13:35:00Z");
        //目标实例ID
        //request.setInstanceId("i-bp12uyz75shn6jsp****");
        //迁移任务交付的目标阿里云镜像名称
        //request.setImageName("smc-image");
        //增量迁移任务运行的时间间隔
        //request.setFrequency(12);
        //目标阿里云服务器ECS的数据盘顺序和大小
        //List<ModifyReplicationJobAttributeRequest.DataDisk> datadisks = new ArrayList<>();
        //ModifyReplicationJobAttributeRequest.DataDisk datadisk = new ModifyReplicationJobAttributeRequest.DataDisk();
        //目标阿里云服务器ECS的数据盘顺序
        //datadisk.setIndex(1);
        //目标阿里云服务器ECS的数据盘大小
```

```
//datadisk.setSize(100);
//datadisks.add(datadisk);
//request.setDataDisks(datadisks);
//增量迁移任务默认保留的最大镜像数
//request.setMaxNumberOfImageToKeep(10);
//迁移任务交付的目标类型
//request.setTargetType("Image");
//中转实例的实例规格
//request.setInstanceType("ecs.c5.large");

//发送请求获取返回值或处理异常
try {
    ModifyReplicationJobAttributeResponse response = client.getAcsResponse(request
);
    System.out.println(new Gson().toJson(response));
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

执行结果

实际返回结果为：

```
{
  "RequestId": "1C488B66-B819-4D14-8711-C4EAAA13AC01"
}
```

2.3.5 启动迁移任务

本文介绍如何通过SMC Java SDK调用StartReplicationJob启动一个迁移任务。

背景信息

您可以调用StartReplicationJob接口启动一个迁移任务，详情请参见[#unique_23](#)。



说明：

您只能使用该API启动Ready（未开始）、Stopped（已暂停）或InError（出错）状态的迁移任务。

代码示例

以下代码适用于启动迁移任务：

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.smc.model.v20190601.StartReplicationJobRequest;
import com.aliyuncs.smc.model.v20190601.StartReplicationJobResponse;
import com.google.gson.Gson;
```



```
public class StartReplicationJob {
    public static void main(String[] args) {
        //初始化AcsClient实例, 设置地域ID及您的AccessKey信息
        DefaultProfile profile = DefaultProfile.getProfile("<region-Id>", "<accessKeyId>", "<
accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);

        //创建API请求, 并设置参数
        StartReplicationJobRequest request = new StartReplicationJobRequest();
        //迁移任务ID
        request.setJobId("j-bp152mkfswviwbqj****");

        //发送请求获取返回值或处理异常
        try {
            StartReplicationJobResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

预期结果

```
{
  "RequestId": "473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E"
}
```

2.3.6 停止迁移任务

本文介绍如何通过SMC Java SDK调用StopReplicationJob停止一个迁移任务。

背景信息

您可以调用StopReplicationJob接口停止一个迁移任务, 详情请参见[#unique_25](#)。

代码示例

以下代码适用于停止一个迁移任务:

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.smc.model.v20190601.StartReplicationJobResponse;
import com.aliyuncs.smc.model.v20190601.StopReplicationJobRequest;
import com.aliyuncs.smc.model.v20190601.StopReplicationJobResponse;
import com.google.gson.Gson;

public class StopReplicationJob {
    public static void main(String[] args) {
        //初始化AcsClient实例, 设置地域ID及您的AccessKey信息
        DefaultProfile profile = DefaultProfile.getProfile("<region-Id>", "<accessKeyId>", "<
accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);

        //创建API请求, 并设置参数
```

```
StopReplicationJobRequest request = new StopReplicationJobRequest();
//迁移任务ID
request.setJobId("j-bp152mkfswviwbqj****");

//发送请求获取返回值或处理异常
try {
    StopReplicationJobResponse response = client.getAcsResponse(request);
    System.out.println(new Gson().toJson(response));
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

预期结果

```
{
  "RequestId": "473469C7-AA6F-4DC5-B3DB-A3DCODE3C83E"
}
```

2.4 一次性迁移服务器

本文介绍如何通过服务器迁移中心SMC Java SDK调用CreateReplicationJob和StartReplicationJob创建并执行一次性迁移任务，再用DescribeReplicationJobs查询迁移任务的信息。

前提条件

- 已导入迁移源，详情请参见[#unique_27](#)。
- 已将Maven依赖添加至pom.xml文件中，详情请参见[安装Java SDK](#)。

完整代码示例

```
package com.aliyun.smc;

import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.smc.model.v20190601.*;
import com.google.gson.Gson;

import java.util.ArrayList;
import java.util.List;
import java.util.UUID;

public class Demo {
    public static void main(String[] args) {
        //初始化AcsClient实例，设置地域ID及您的AccessKey信息
        DefaultProfile profile = DefaultProfile.getProfile("<region-Id>", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        //创建迁移任务
```

```
String jobId = createReplicationJob(client);
//启动迁移任务
startReplicationJob(client,jobId);
//查询迁移任务状态
while(true){
    String result = describeReplicationJobs(client,jobId);
    System.out.println("迁移正在进行.....");
    if("Finished".equals(result)){
        System.out.println("迁移完成! ");
        break;
    }
    try {
        Thread.sleep(600000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

private static String createReplicationJob (IAcsClient client){
    //创建API请求, 并设置参数
    CreateReplicationJobRequest request = new CreateReplicationJobRequest();
    //迁移源ID
    request.setSourceId("s-bp152mkfswviw7sd****");
    //目标阿里云服务器ECS的系统盘大小, 单位为GiB
    request.setSystemDiskSize(80);
    //保证请求幂等性, 您可以从客户端生成一个不超过64个ASCII字符的参数值, 并将值赋予
    ClientToken
    //request.setClientToken(UUID.randomUUID().toString());
    //迁移任务描述
    request.setDescription("smcDescription");
    //迁移任务名
    request.setName("MigrationTask");
    //创建一次性迁移任务还是增量迁移任务
    request.setRunOnce(true);
    //迁移任务的执行时间
    //request.setScheduledStartTime("2019-06-04T13:35:00Z");
    //迁移任务的标签键值对
    List<CreateReplicationJobRequest.Tag> tags = new ArrayList<>();
    CreateReplicationJobRequest.Tag tag = new CreateReplicationJobRequest.Tag();
    tag.setKey("usage");
    tag.setValue("production");
    tags.add(tag);
    request.setTags(tags);
    //迁移任务交付的目标类型
    //request.setTargetType("Image");
    //指定VPC下的交换机ID
    request.setVSwitchId("vsw-bp1ddbrxdlrcbim46****");
    //迁移任务的过期时间
    //request.setValidTime("2019-06-04T13:35:00Z");
    //已配置高速通道服务或者VPN网关的VPC的ID
    request.setVpcId("vpc-bp1vwnn14rqpyiczj****");

    //发送请求获取返回值或处理异常
    CreateReplicationJobResponse response = null;
    try {
        response = client.getAcsResponse(request);
        System.out.println(new Gson().toJson(response));
        System.out.println("*****成功创建迁移任务*****");
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
}
```

```
        return response.getJobId();
    }

    private static void startReplicationJob(IAcsClient client,String jobId){
        //创建API请求, 并设置参数
        StartReplicationJobRequest request = new StartReplicationJobRequest();
        //迁移任务ID
        request.setJobId(jobId);

        //发送请求获取返回值或处理异常
        try {
            StartReplicationJobResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
            System.out.println("*****启动迁移任务*****");
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }

    private static String describeReplicationJobs(IAcsClient client,String jobId){
        //创建API请求, 并设置参数
        DescribeReplicationJobsRequest request = new DescribeReplicationJobsRequest();
        //设置迁移任务的页码
        request.setPageNumber(1);
        //设置每页行数
        request.setPageSize(10);
        //设置迁移任务的名称
        request.setName("MigrationTask");
        //设置迁移任务ID列表
        List<String> jobIds = new ArrayList<>();
        jobIds.add(jobId);
        request.setJobIds(jobIds);
        //设置迁移源ID列表
        List<String> sourceIds = new ArrayList<>();
        sourceIds.add("s-bp152mkfswviw7sd****");
        request.setSourceIds(sourceIds);

        //发送请求获取返回值或处理异常
        DescribeReplicationJobsResponse response = null;
        try {
            response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
            System.out.println("*****查询迁移任务状态*****");
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
        String status = null;
        List<DescribeReplicationJobsResponse.ReplicationJob> replicationJobList = response.
        getReplicationJobs();
        for(DescribeReplicationJobsResponse.ReplicationJob rl : replicationJobList){
            status = rl.getStatus();
        }
        return status;
    }
}
```

预期结果

```
{
```

```
"requestId": "667E03A3-7D16-4822-8B8D-1F3883B25903",
"jobId": "j-bp152mkfswviwbqj****"
}
*****成功创建迁移任务*****
{
  "requestId": "3A236693-5FF9-4F76-8F7D-A405113A6B55"
}
*****启动迁移任务*****
{
  "requestId": "0E9F3604-D010-4EDE-995C-F123ED9CEB5B",
  "totalCount": 1,
  "pageNumber": 1,
  "pageSize": 10,
  "replicationJobs": [{
    "jobId": "j-bp152mkfswviwbqj****",
    "sourceId": "s-bp152mkfswviw7sd****",
    "name": "MigrationTask",
    "description": "smcDescription",
    "regionId": "cn-hangzhou",
    "targetType": "Image",
    "status": "Running",
    "businessStatus": "Preparing",
    "progress": 0.0,
    "creationTime": "2020-01-08T06:44:04Z",
    "validTime": "2020-02-07T06:44:05Z",
    "startTime": "2020-01-08T06:44:05Z",
    "netMode": 0,
    "systemDiskSize": 80,
    "vpId": "vpc-bp1vwnn14rqpyiczj****",
    "vSwitchId": "vsw-bp1ddbrxdlrcbim4****j",
    "runOnce": true,
    "dataDisks": [],
    "replicationJobRuns": []
  ]
}
*****查询迁移任务状态*****
迁移正在进行.....
{
  "requestId": "0E9F3604-D010-4EDE-995C-F123ED9CEB5B",
  "totalCount": 1,
  "pageNumber": 1,
  "pageSize": 10,
  "replicationJobs": [{
    "jobId": "j-bp152mkfswviwbqj****",
    "sourceId": "s-bp152mkfswviw7sd****",
    "name": "MigrationTask",
    "description": "smcDescription",
    "regionId": "cn-hangzhou",
    "targetType": "Image",
    "status": "Running",
    "businessStatus": "Preparing",
    "progress": 0.0,
    "creationTime": "2020-01-08T06:44:04Z",
    "validTime": "2020-02-07T06:44:05Z",
    "startTime": "2020-01-08T06:44:05Z",
    "netMode": 0,
    "systemDiskSize": 80,
    "vpId": "vpc-bp1vwnn14rqpyiczj****",
    "vSwitchId": "vsw-bp1ddbrxdlrcbim46****",
    "runOnce": true,
    "dataDisks": [],
    "replicationJobRuns": []
  ]
}
}
```

```
*****查询迁移任务状态*****
迁移正在进行.....
{
  "requestId": "0E9F3604-D010-4EDE-995C-F123ED9CEB5B",
  "totalCount": 1,
  "pageNumber": 1,
  "pageSize": 10,
  "replicationJobs": [{
    "jobId": "j-bp152mkfswviwbqj****",
    "sourceId": "s-bp152mkfswviw7sd****",
    "name": "MigrationTask",
    "description": "smcDescription",
    "regionId": "cn-hangzhou",
    "targetType": "Image",
    "status": "Finished",
    "businessStatus": "Preparing",
    "progress": 0.0,
    "creationTime": "2020-01-08T06:44:04Z",
    "validTime": "2020-02-07T06:44:05Z",
    "startTime": "2020-01-08T06:44:05Z",
    "netMode": 0,
    "systemDiskSize": 80,
    "vpId": "vpc-bp1vwnn14rqpyiczj****",
    "vSwitchId": "vsw-bp1ddbrxdlrcbim46****",
    "runOnce": true,
    "dataDisks": [],
    "replicationJobRuns": []
  }]
}
*****查询迁移任务状态*****
迁移完成!
```

3 Python示例

3.1 安装Python SDK

本文介绍如何安装服务器迁移中心SMC Python SDK。

环境准备

首先确保您已经具备Python的Runtime，建议使用的Python版本为2.7或更高版本。

安装SDK

- 通过pip安装，执行如下命令：

```
pip install aliyun-python-sdk-smc
```



说明：

如果您的环境尚未安装pip，请参见[pip官网](#)安装。

- 通过源码安装，从[GitHub](#)下载SDK包，解压后进入目录，进入aliyun-python-sdk-smc子目录下，确认目录下有setup.py文件，然后执行命令如下：

```
python setup.py install
```

3.2 一次性迁移服务器

本文介绍如何通过服务器迁移中心SMC Java SDK调

用CreateReplicationJob和StartReplicationJob创建并执行一次性迁移任务，再调

用DescribeReplicationJobs查询迁移任务的信息。

前提条件

- 已导入迁移源，详情请参见[#unique_27](#)。
- 已安装SMC Python SDK，详情请参见[安装Python SDK](#)。

完整代码示例

```
from aliyunsdkcore.client import AcsClient
from aliyunsdksmc.request.v20190601 import CreateReplicationJobRequest
from aliyunsdksmc.request.v20190601 import StartReplicationJobRequest
from aliyunsdksmc.request.v20190601 import DescribeReplicationJobsRequest
import json
import time
import uuid

# 初始化AcsClient实例，设置地域ID及您的AccessKey信息
```

```
client = AcsClient('<accessKeyId>', '<accessSecret>', '<region-Id>')

# 创建迁移任务
def create_replication_job():
    # 创建API请求, 并设置参数
    request = CreateReplicationJobRequest.CreateReplicationJobRequest()
    # 迁移源ID
    request.set_SourceId('s-bp152mkfswwidyr3****')
    # 目标阿里云服务器ECS的系统盘大小, 单位为GiB
    request.set_SystemDiskSize(80)
    # 保证请求幂等性, 您可以从客户端生成一个不超过64个ASCII字符的参数值, 并将值赋予
    ClientToken
    # request.set_ClientToken(uuid.uuid4())
    # 迁移任务名
    request.set_Name('smcName')
    # 迁移任务描述
    request.set_Description('setDescription')
    # 已配置高速通道服务或者VPN网关的VPC的ID
    request.set_VpcId('vpc-bp1vwenn14rqpyiczj****')
    # 指定VPC下的交换机ID
    request.set_VSwitchId('vsw-bp1ddbrxdlrcbim46****')

    # 发送请求获取返回值
    response = client.do_action_with_exception(request)
    # python2: print(response)
    print(str(response, encoding='utf-8'))
    print('*****成功创建迁移任务*****')
    return json.loads(response)['JobId']

# 启动迁移任务
def start_replication_job(job_id):
    # 创建API请求, 并设置参数
    request = StartReplicationJobRequest.StartReplicationJobRequest()
    # 迁移任务ID
    request.set_JobId(job_id)

    # 发送请求获取返回值
    response = client.do_action_with_exception(request)
    # python2: print(response)
    print(str(response, encoding='utf-8'))
    print('*****启动迁移任务*****')

# 查询迁移任务状态
def describe_replication_jobs(job_ids):
    # 创建API请求, 并设置参数
    request = DescribeReplicationJobsRequest.DescribeReplicationJobsRequest()
    # 设置迁移任务ID列表
    jobIds = [job_ids]
    request.set_JobIds(jobIds)
    # 设置迁移任务的名称
    request.set_Name('smcName')
    # 设置迁移任务的页码
    request.set_PageNumber(1)
    # 设置每页行数
    request.set_PageSize(10)
    # 设置迁移源ID列表
    sourceIds = ['s-bp152mkfswwidyr3****']
    request.set_SourceIds(sourceIds)

    # 发送请求获取返回值
    response = client.do_action_with_exception(request)
    # python2: print(response)
    print(str(response, encoding='utf-8'))
    print('*****查看迁移任务状态*****')
```



```

joblist = json.loads(response)['ReplicationJobs']['ReplicationJob']
for i in joblist:
    return (i['Status'])

# 执行创建迁移任务方法
job_id = create_replication_job()
# 执行启动迁移任务方法
start_replication_job(job_id)
# 每隔10分钟执行一次查询迁移任务状态方法，判断是否迁移完成。
while True:
    status = describe_replication_jobs(job_id)
    print('迁移任务正在执行.....')
    time.sleep(600)
    if status == 'Finished':
        print('迁移完成! ')
        break

```

预期结果

```

{
  "JobId": "j-bp15dpodofnvm1s****",
  "RequestId": "1042DD41-FEF1-4267-8733-C910B765B9CF"
}
*****成功创建迁移任务*****
{
  "RequestId": "2A7930A1-56B2-4B5D-9874-4CB48F5C5095"
}
*****启动迁移任务*****
{
  "PageNumber": 1,
  "TotalCount": 1,
  "PageSize": 10,
  "ReplicationJobs": {
    "ReplicationJob": [{
      "Description": "setDescription",
      "SystemDiskSize": 80,
      "TargetType": "Image",
      "VSwitchId": "vsw-bp1ddbrxdlrcbim46****",
      "SourceId": "s-bp152mkfswvidyr3****",
      "NetMode": 0,
      "VpcId": "vpc-bp1vwnn14rqpyicz****",
      "Progress": 0.0,
      "Name": "smcName",
      "CreationTime": "2020-01-08T08:27:45Z",
      "Status": "Running",
      "BusinessStatus": "Preparing",
      "JobId": "j-bp15dpodofnvm1s****",
      "RegionId": "cn-hangzhou",
      "RunOnce": true,
      "StartTime": "2020-01-08T08:27:46Z",
      "ValidTime": "2020-02-07T08:27:45Z"
    }
  ]
},
  "RequestId": "B62A8AEE-95AA-400F-8A6C-4822E3F4B690"
}
*****查看迁移任务状态*****
迁移任务正在执行.....
{
  "PageNumber": 1,
  "TotalCount": 1,
  "PageSize": 10,
  "ReplicationJobs": {
    "ReplicationJob": [{

```

```
"Description": "setDescription",
"SystemDiskSize": 80,
"TargetType": "Image",
"VSwitchId": "vsw-bp1ddbrxdlrcbim46****",
"SourceId": "s-bp152mkfswvidyr3****",
"NetMode": 0,
"VpcId": "vpc-bp1vwonn14rqpyiczj****",
"Progress": 0.0,
"Name": "smcName",
"CreationTime": "2020-01-08T08:27:45Z",
"Status": "Running",
"BusinessStatus": "Preparing",
"JobId": "j-bp15dpodoftnvm1s****",
"RegionId": "cn-hangzhou",
"RunOnce": true,
"StartTime": "2020-01-08T08:27:46Z",
"ValidTime": "2020-02-07T08:27:45Z"
  }}
},
"RequestId": "B62A8AEE-95AA-400F-8A6C-4822E3F4B690"
}
*****查看迁移任务状态*****
迁移完成!
```