

命令行工具 CLI

用户指南



用户指南

概述

命令行工具是用于统一管理阿里云服务的工具。通过安装和配置该工具，您可以在命令行方式下控制多个阿里云产品、访问和管理您账户下的云资产，包括ECS、RDS、SLB、OSS等所有类型的阿里云基础设施产品。

安装命令行工具（Windows）

安装须知

阿里云命令行工具 CLI 是基于 Python 编写的，运行 CLI 需要 Python 运行环境。pip 是基于 Python 开发的应用安装升级工具，通过 pip 您能够方便地安装、升级和删除 Python 程序包。因此，我们建议您使用 pip 工具安装命令行工具 CLI。

Python 信息详情请参见 <https://www.python.org/>，pip 信息详情请参见 <https://pip.pypa.io/>。安装 Python 时，系统默认选择安装 pip 的功能。

在线安装命令行工具和 SDK

我们推荐您在设备联网时使用 pip 安装阿里云命令行工具 CLI。另外，由于该命令行工具需要在 Python 环境中运行，为确保正确安装命令行工具和阿里云 SDK，请您依照如下步骤依次安装 Python、pip、命令行工具和阿里云 SDK。

系统要求

- Windows
- Python 2.7.x

安装 Python

若您的设备已安装 Python 2.7.x 版本，请跳过此步骤。

若您的设备没有安装 Python，请登录 Python 官方网站（<https://www.python.org/downloads/>），下载 Python 2.7.x 版本，依照安装向导进行安装即可。

注意：在安装 Python 时，请务必开启 pip 功能，如下图。否则，您将无法执行 pip 安装命令行工具的步骤。



安装命令行工具

如果系统内的 pip 版本过低，会造成 CLI 安装出错。用户可以使用如下指令先对 pip 软件进行升级后再进行相关操作。请使用 pip 7.x 或更高版本。若已是最新版本的 pip，请跳过此步骤。

在命令行窗口中执行如下命令，升级 pip。

```
cd C:\Python27
python -m pip install -U pip
```

系统显示如下类似信息，则表明升级成功。

```
Successfully uninstalled pip-7.1.2
Successfully installed pip-8.1.2
```

执行如下命令，安装阿里云命令行工具。

```
cd C:\Python27
cd Scripts
pip install aliyuncli
```

系统显示如下类似信息，则表明安装成功。

```
Successfully installed aliyuncli-2.1.2 colorama-0.3.3 jmespath-0.7.1
```

离线安装命令行工具和 SDK

若您需要在无法连网的设备上安装命令行工具 CLI，请使用命令行工具安装包进行安装。请您先在一台已连网设备上依照下面的操作步骤下载该安装包。另外，命令行工具的离线安装包内已经集成了阿里云产品的 SDK，所以在此状况下不需要再单独安装 SDK。

注意：该操作不支持命令行工具的更新。如需更新软件，请待新版本发布后，卸载掉老版本，然后下载新版本重新安装。

下载命令行工具安装包

直接点击下载或者通过云市场下载，请按照您使用的系统选择对应的版本。

Windows 64位：点击下载

Windows 32位：点击下载

命令行工具云市场下载地址：

<http://market.aliyun.com/products/53690006/cmgy000314.html?spm=5176.900004.4.2.Fbz81W>

安装命令行工具

前提条件：

已安装 Python。若没有安装，请在已连网设备上下载 Python 2.7 版本的安装包，Python 官网的地址为 <https://www.python.org/downloads/>。

双击命令行工具安装包，进入安装向导。

点击下一步，继续安装向导。

选择安装路径，点击**下一步**。我们建议您保持默认的安装路径不做更改。

点击**安装**。

点击**完成**，完成安装并退出安装向导。

安装命令行工具（Linux/UNIX/Mac OS）

安装须知

命令行工具 CLI 是基于 Python 编写的，运行 CLI 需要 Python 运行环境。pip 是基于 Python 开发的应用安装升级工具，通过 pip 您能够方便地安装、升级和删除 Python 程序包。因此，我们建议您使用 Pip 工具安装命令行工具 CLI。

Python 信息详情请参见 <https://www.python.org/>，pip 信息详情请参见 <https://pip.pypa.io/>。

在线安装命令行工具和 SDK

我们推荐您在设备联网时使用 pip 安装阿里云命令行工具 CLI。另外，由于该命令行工具需要在 Python 环境中运行，为确保正确安装命令行工具和阿里云 SDK，请您依照如下步骤依次安装 Python、pip、命令行工具和阿里云 SDK。

系统要求

Linux、UNIX 或 Mac OS

Python 2.7.x

安装 Python

若您的设备已安装 Python 2.7.x 版本，请跳过此步骤。

若您的设备没有安装 Python 2.7.x 版本，请在命令行窗口中执行下列命令，安装 Python。

注意：请确保您的设备中已安装了 `wget`。

```
$ wget https://www.python.org/ftp/python/2.7.8/Python-2.7.8.tgz (或者通过其他方式下载后放在某个路径下)
$ tar -zxvf Python-2.7.8.tgz
$ cd Python-2.7.8
$ ./configure
$ make
$ sudo make install
```

安装 pip

若您的设备已安装 `pip`，请跳过此步骤。

若您的设备没有安装 `pip`，在命令行窗口中执行如下命令，安装 `pip`。安装 `pip` 必须使用 `sudo` 权限。

```
$ curl "https://bootstrap.pypa.io/get-pip.py" -o "pip-install.py"
$ sudo python pip-install.py
```

系统显示如下类似信息，则表明安装成功。

```
Successfully installed pip-7.1.2 setuptools-18.7 wheel-0.26.0
```

安装命令行工具

如果系统内的 `pip` 版本过低，会造成 CLI 安装出错。用户可以使用如下指令先对 `pip` 软件进行升级后再进行相关操作。请使用 `pip 7.x` 或更高版本。若已是最新版本的 `pip`，请跳过此步骤。

在命令行窗口中执行如下命令，升级 `pip`。

```
pip install -U pip
```

系统显示如下类似信息，则表明升级成功。

```
Successfully uninstalled pip-7.1.2
Successfully installed pip-8.1.2
```

执行如下命令，安装阿里云命令行工具。

```
$ sudo pip install aliyuncli
```

系统显示如下类似信息，则表明安装成功。

```
Successfully installed aliyuncli-2.1.2 colorama-0.3.3 jmespath-0.7.1
```

在线安装 SDK

命令行工具需要与阿里云各产品的 SDK 协同工作。为方便用户使用，在线安装命令行工具时，与阿里云 SDK 的安装进行了分离。您可以自由选择所需产品的 SDK，不必下载所有的阿里云 SDK。命令行工具能够根据您的系统中安装的 SDK 进行动态调整。

以 RDS 产品为例，在命令行窗口中执行如下命令，安装 SDK。

```
$ sudo pip install aliyun-python-sdk-rds
```

系统显示如下类似信息，则表明安装成功。

```
Successfully installed aliyun-python-sdk-core-2.0.35 aliyun-python-sdk-rds-2.0.3
```

更多产品对应的 SDK，请参见阿里云 Python SDK 列表。

离线安装命令行工具和 SDK

若您需要在无法连网的设备上安装命令行工具 CLI，请使用命令行工具安装包进行安装。请您先在一台已连网设备上依照下面的操作步骤下载该安装包。另外，命令行工具的离线安装包内已经集成了阿里云产品的 SDK，所以在此状况下不需要再单独安装 SDK。

注意：该操作不支持命令行工具的更新。如需更新软件，需要待新版本发布后，卸载掉老版本，然后下载新版本重新安装。

下载命令行工具安装包

点击"点击下载"或者通过云市场下载，请按照您使用的系统选择对应的版本。

Linux/UNIX & Mac OS：点击下载

命令行工具云市场下载地址：

<http://market.aliyun.com/products/53690006/cmgy000314.html?spm=5176.900004.4.2.Fbz81W>

安装命令行工具

前提条件：

已安装 Python。若没有安装，请在已连网设备上下载 Python 2.7.x 版本的安装包，Python 官网的地址为

<https://www.python.org/downloads/>。

在命令行窗口找到压缩软件包 cli.tar.gz。

在命令行窗口中执行如下命令，安装命令行工具。

```
$ tar -zxvf cli.tar.gz
$ cd cli
$ sudo sh install.sh
```

系统显示如下类似信息，则表明安装成功。

```
New aliyuncli version has been installed successfully!
```

启用命令自动补全功能

在 Linux/UNIX 和 Mac OS 环境下，命令行工具 CLI 包含一项命令提示和自动补全的功能，让您可以使用 TAB 键完成部分键入的命令并且可以产生命令的联想提示。此功能不是自动安装的，需要您手动配置。

配置命令行工具自动补全功能需要两种信息：您所使用的 Shell 类型和命令行工具自动补全脚本 aliyun_completer 脚本的位置。

识别 Shell 类型

如果您不确定所使用的 Shell 类型，可以使用以下命令之一进行识别。命令行工具目前支持 bash 和 zsh 的自动补全。

方法一：显示 Shell 安装目录

在命令行窗口执行如下命令，显示 Shell 的安装目录。这通常会与所使用的 Shell 匹配，除非您在登录后启动了不同的 Shell。

```
$ echo $SHELL
```

方法二：显示当前用户运行的进程

在命令行窗口执行如下命令，显示当前用户运行的进程，Shell 是其中之一。

```
$ ps
```


确定 aliyun_completer 的存放路径

在命令行窗口中执行如下命令，找到 aliyun_completer 的存放路径。

```
$ which aliyun_completer
```

启用命令自动补全功能

Shell 类型 - bash

在命令行窗口执行如下命令，使用complete指令启用命令行工具的命令自动补全功能。
Complete指令是bash自动补全的指令，通过将指定的补全脚本编译到可执行文件中，实现对应的可执行文件在按下TAB时出现提示和自动补全的功能。

```
$ complete -C '<aliyun_completer的存放路径, 如/usr/local/bin/aliyun_completer>' aliyuncli
```

执行如下命令，检查命令自动补全功能是否成功启用。

```
$ complete
```

系统显示如下类似信息，则说明命令自动补全功能已成功启用。

```
complete -C <aliyun_completer的存放路径, 如/usr/local/bin/aliyun_completer > aliyuncli
```

为保证每次启动 bash 时自动补全功能都有效，您需要将自动补全的命令写入 Shell 的配置文件中。
请执行如下命令，将步骤1中的命令追加到 .bash_profile 中。

```
$ vim ~/.bash_profile
将complete -C '<aliyun_completer的存放路径, 如/usr/local/bin/aliyun_completer>' aliyuncli 添加在文件最后即可。
```

Shell 类型 - zsh

在命令行窗口执行如下命令，使用 aliyun_zsh_complete.sh 去启用命令行工具的命令自动补全功能。

```
source /usr/local/bin/aliyun_zsh_complete.sh
```

配置命令行工具

公共云版

注意：本文档内容仅适用于公共云用户配置其命令行工具，专有云、专有域用户请参见 [配置命令行工具（专有云/专有云版）](#)。

安装好命令行工具后，您需要先配置安全证书，证书是工具和阿里云基础服务之间必须的凭证，所有命令的请求都必须包含这些信息。所以您首先要在可连网的设备上创建 access key。

除证书外，您还可以配置您购买的阿里云产品的区域信息和默认的输出格式，如 text、table 或 json。

创建 access key

登录阿里云官网 <https://www.aliyun.com/>。

登录[管理控制台](#)。

点击 [AccessKeys](#)。

注意：若出现如下提示框，请点击[继续使用 Accesskey](#)。



点击[创建 Access Key](#)，然后再点击[同意并创建](#)。

输入短信校验码，点击[确定](#)。Access Key 创建成功。

配置命令行工具

在 Windows 环境下，执行如下命令，打开并填写所列参数。

```
cd C:\Python27\Scripts
aliyuncli configure
Aliyun Access Key ID [None]: <输入Access Key ID>
Aliyun Access Key Secret [None]: <输入Access Key Secret>
Default Region Id [None]: <输入实例的Region Id>
Default output format [None]: <输入您需要的输出格式>
```

在 Linux/UNIX 和 Mac OS 环境下，执行如下命令，打开并填写所列参数。

```
$ sudo aliyuncli configure
Aliyun Access Key ID [None]: <输入Access Key ID>
Aliyun Access Key Secret [None]: <输入Access Key Secret>
Default Region Id [None]: <输入您购买的阿里云产品的Region Id>
Default output format [None]: <输入您需要的输出格式>
```

修改命令行工具的配置

若需要修改命令行工具中的参数设置，请执行aliyuncli configure命令。不需修改的参数，请直接按Enter键；需要修改的参数，请输入新的参数值。请参见以下案例。

在 Windows 环境下，执行如下命令：

```
cd C:\Python27\Scripts
aliyuncli configure
Aliyun Access Key ID [*****jkf]: <Enter>
Aliyun Access Key Secret [*****OkI]: <Enter>
Default Region Id [cn-qingdao]: cn-hangzhou
Default output format [table]: <Enter>
```

在 Linux/UNIX 和 Mac OS 环境下，执行如下命令：

```
$ sudo aliyuncli configure
Aliyun Access Key ID [*****jkf]: <Enter>
Aliyun Access Key Secret [*****OkI]: <Enter>
Default Region Id [cn-qingdao]: cn-hangzhou
Default output format [table]: <Enter>
```

升级命令行工具

在命令行窗口中执行如下命令，升级现有的命令行工具。

在 Windows 环境下，执行如下命令：

```
pip install --upgrade aliyuncli
```

在 Linux/UNIX 和 Mac OS 环境下，执行如下命令：

```
$ sudo pip install --upgrade aliyuncli
```

如果出现 "already up-to-date" 的提示信息则说明已经是最新版本，无须升级。

卸载命令行工具

在 Windows 环境下，执行如下命令，卸载命令行工具。

```
cd C:\Python27
cd Scripts
pip uninstall aliyuncli
```

在 Linux/UNIX 和 Mac OS 环境下，执行如下命令，卸载命令行工具。

```
$ sudo pip install aliyuncli
```

系统显示如下类似信息，则表明卸载成功。

```
Successfully uninstalled aliyuncli-2.1.2
```

升级阿里云SDK

以 RDS 产品为例，在命令行窗口中执行如下命令，升级现有的阿里云SDK。

在 Windows 环境下，执行如下命令：

```
pip install --upgrade aliyun-python-sdk-rds
```

在 Linux/UNIX 和 Mac OS 环境下，执行如下命令：

```
$ sudo pip install --upgrade aliyun-python-sdk-rds
```

如果出现 "already up-to-date" 的提示信息则说明已经是最新版本，无须升级。

阿里云 Python SDK 列表

阿里云各产品对应的 SDK 如下：

产品	SDK
AAS	aliyun-python-sdk-aas
Alidns	aliyun-python-sdk-alidns
BatchCompute	aliyun-python-sdk-batchcompute
Bsn	aliyun-python-sdk-bsn
Bss	aliyun-python-sdk-bss
Cdn	aliyun-python-sdk-cdn
Cf	aliyun-python-sdk-cf
Cms	aliyun-python-sdk-cms
Core	aliyun-python-sdk-core
Crm	aliyun-python-sdk-crm
Cs	aliyun-python-sdk-cs
Domain	aliyun-python-sdk-domain
Drds	aliyun-python-sdk-drds
ECS	aliyun-python-sdk-ecs
ESS	aliyun-python-sdk-ess
Ft	aliyun-python-sdk-ft
Green	aliyun-python-sdk-green
hpc	aliyun-python-sdk-hpc
httpdns	aliyun-python-sdk-httpdns
Iot	aliyun-python-sdk-iot
Kms	aliyun-python-sdk-kms
Mts	aliyun-python-sdk-mts
Ocs	aliyun-python-sdk-ocs
Oms	aliyun-python-sdk-oms
OssAdmin	aliyun-python-sdk-ossadmin
Push	aliyun-python-sdk-push
Ram	aliyun-python-sdk-ram

RDS	aliyun-python-sdk-rds
Risk	aliyun-python-sdk-risk
Ros	aliyun-python-sdk-ros
Slb	aliyun-python-sdk-slb
Sts	aliyun-python-sdk-sts
Ubsms	aliyun-python-sdk-ubsms
Yundun	aliyun-python-sdk-yundun

全局 SDK 版本配置

阿里云 SDK 以产品为维度，您可以根据自己的需求分别安装和使用，而且您可以选择各个不同的 SDK 版本（对应于不同的 Open API 版本）。在命令行工具中，您可以配置想要的产品版本。如果您不配置，工具会默认采用最新的版本工作。

配置 SDK 版本

以配置 ECS 产品的 SDK 为例，执行如下命令，配置其 SDK 版本。

```
aliyuncli ecs ConfigVersion --version v20140526
```

查看单个产品的 SDK 版本

若您要查看当前系统中安装的某一产品的所有 SDK 的版本，如 ECS 的 SDK 版本，执行如下命令，进行查看。

```
aliyuncli ecs ShowVersions
* v20140526 ← 当前默认使用的版本。如果不配置，则默认采用最新版本。
v20150304 ← 当前系统中安装的 SDK 版本。
v20140526
```

使用在线帮助

为便于用户使用，命令行工具提供了在线帮助命令，您可以通过 help 命令查看和了解。

例如，您要查看 ECS 支持的所有操作，请在命令行窗口中执行命令 `aliyuncli ecs help`，查询结果如下图所示。

```
-bash-4.1$ aliyuncli ecs help
usage: aliyuncli <command> <operation> [options and parameters]
[ecs] valid operations as follows:

AbsConfigService          | AllocateEipAddress
AllocatePublicIpAddress   | AssociateEipAddress
AttachDisk                 | AuthorizeSecurityGroup
BindIpRange               | CreateDisk
CreateImage                | CreateInstance
CreateRouteEntry           | CreateSecurityGroup
CreateSnapshot             | CreateVSwitch
CreateVpc                  | DeleteDisk
DeleteImage                | DeleteInstance
DeleteRouteEntry           | DeleteSecurityGroup
DeleteSnapshot             | DeleteVSwitch
DeleteVpc                  | DescribeAutoSnapshotPolicy
DescribeDisks              | DescribeEipAddresses
DescribeEipMonitorData     | DescribeImages
DescribeInstanceAttribute  | DescribeInstanceMonitorData
DescribeInstanceStatus     | DescribeInstanceTypes
DescribeInstances          | DescribeRegions
DescribeRouteTables        | DescribeSecurityGroupAttribute
DescribeSecurityGroups     | DescribeSnapshots
DescribeVRouter            | DescribeVSwitches
DescribeVpcs               | DescribeZones
DetachDisk                 | JoinSecurityGroup
LeaveSecurityGroup          | ModifyAutoSnapshotPolicy
ModifyDiskAttribute        | ModifyEipAddressAttribute
ModifyInstanceAttribute    | ModifyInstanceNetworkSpec
ModifyInstanceVpcAttribute | ModifySecurityGroupAttribute
ModifyVRouterAttribute     | ModifyVSwitchAttribute
ModifyVpcAttribute         | ReInitDisk
RebootInstance             | ReleaseEipAddress
ReplaceSystemDisk          | ResetDisk
RevokeSecurityGroup        | StartInstance
StopInstance               | UnassociateEipAddress
```

若您要查看 ECS 某个操作具体对应的参数值，请在命令行窗口中执行命令 `aliyuncli ecs <操作名称> help`，如 `DescribeRegions` 所对应参数值的查询结果如下图所示。

```
-bash-4.1$ aliyuncli ecs DescribeRegions help
usage: aliyuncli <command> <operation> [options and parameters]
[ecs.DescribeRegions]: current operation can uses parameters as follow :

--AccessKeyId              | --AccessKeySecret
--Endpoint                 | --OwnerAccount
--OwnerId                  | --ResourceOwnerAccount
--output
```

命令行结构说明

阿里云命令行工具使用的命令行结构如下：

```
aliyuncli <command> <subcommand> [options and parameters]
```

aliyuncli：阿里云工具名。

command：指定一个顶级命令。1) 通常表示命令行工具中支持的阿里云 基础服务，如 ECS/RDS/MTS 等；2) 也可是命令行工具本身的命令，如 `help`、`configure` 等。

subcommand：指定要执行操作的附加子命令，即具体的某一项操作。

options and parameters：subcommand 中输入的操作所对应的参数列表。该参数列表的顺序不会对命令的使用产生影响。参数可采用各种类型的输入值，如数字、字符串、列表、映射和 JSON 结构。

例如：

```
aliyuncli rds DescribeDBInstances --PageSize 50

aliyuncli ecs DescribeRegions

aliyuncli rds DescribeDBInstanceAttribute --DBInstanceId xxxxxx
```

全局参数说明

命令行工具在执行时，支持全局参数的临时设置，您可以根据需要调整。目前支持的全局参数如下：

AccessKeyId：指定当前命令执行时，API 请求中的 Access Key Id。如果不指定或者值为空，则使用默认的全局 Access Key Id。

AccessKeySecret：指定当前命令执行时，API 请求中的 Access Key Secret。如果不指定或者值为空，则使用默认的全局 Access Key Secret。

RegionId：指定当前命令执行时，API 请求对应的 Region。如果不指定，则会用全局的 RegionId。

output：指定当前命令执行时要显示的格式。

profile：指定当前命令执行时采用的账户信息是哪一个。如果指定的账户信息不存在，不会选择默认账户信息。另外，如果 profile 和其他全局参数同时出现时，优先级低于其他全局参数。例如 profile 和 AccessKeyId 同时出现，优先选择 AccessKeyId 作为 Access Key。

version：指定当前命令执行时采用的产品开放 API 版本信息。如果没有配置，那么执行命令时会选择当前系统安装的最新的版本去执行。

注意：若此参数直接跟在 aliyuncli 之后，那么会显示当前命令行工具的版本。

参数值输入要求

阿里云命令行工具在调用时，可能需要传递相应的值给命令行工具。请您在使用阿里云命令行工具时，参考如下注意事项处理您的输入值，避免出现错误。

大部分情况下，您会传递一个string类型或者一个数字类型的值到阿里云命令行工具，此时只需要简单的写上需要的值即可。

例如：

```
$ aliyuncli ecs DescribeInstanceAttribute --InstanceId myInstanceId
```

若您输入的值中有空格，请使用单引号 (') 括住您的值。在 Windows PowerShell、Mac OS 以及 Linux/UNIX 环境下，都可采用这种方式。

例如：

```
$ aliyuncli ecs DescribeInstanceAttribute --InstanceId 'my instance id'
```

若采用 Windows Command Processer，请使用双引号 (") 括住您的值。

例如：

```
> aliyuncli ecs DescribeInstanceAttribute --InstanceId "my instance id"
```

使用 JSON 格式作为参数。JSON 格式在阿里云命令行工具中是允许使用的，特别是当您要同时查询多个实例信息或者多个磁盘信息时，可以按照 JsonArray 的格式传入多个 id 值。使用 JSON 格式时，您需要严格按照 JSON 格式编写数据。

查询多个实例信息可以这样传值 ["my-intances-id1", "my-intances-id2"]。但在 Python 环境下，双引号 (") 会被系统默认过滤掉，因此您需要对 JSON 格式中的双引号 (") 进行特殊的处理。请根据您的系统环境，依照如下方法处理双引号 (")。

在 Linux/UNIX 和 Mac OS 环境下，请用单引号 (') 括住整个 JSON 的值。

例如：

```
$ aliyuncli ecs DescribeInstances --InstanceIds ['my-intances-id1', 'my-intances-id2']
```

在 Windows Command Processer 环境下，双引号 (") 需要用反斜杠 (\) 方式转义，同时再用双引号 (") 将整个 JSON 值括住。

例如：

```
> aliyuncli ecs DescribeInstances --InstanceIds ["\"my-intances-id1\", \"my-intances-id2\"]"
```

在 Windows PowerShell 环境下，双引号 (") 需要用反斜杠 (\) 方式转义，同时再用单引号 (') 将整个 JSON 值括住。

例如：

```
> aliyuncli ecs DescribeInstances --InstanceIds ["my-intances-id1", "my-intances-id2"]
```

命令输出格式

输出格式类型

为满足用户在实际使用中对输出格式的不同要求，阿里云命令行工具支持以下三种不同的输出格式。

JSON (json)

JSON 格式是命令行工具默认的输出格式。大多数语言有内置功能或者公开的 JSON 解析库，提供便捷解析 JSON 字符串的方法。JSON 格式主要可以用在其他脚本或者任意编程语言的联合作业中，便于开发者解析和使用。

```
-bash-4.1$ aliyuncli ecs DescribeRegions --output json
{
  "Regions": {
    "Region": [
      {
        "LocalName": "\u6df1\u5733",
        "RegionId": "cn-shenzhen"
      },
      {
        "LocalName": "\u9752\u5c9b",
        "RegionId": "cn-qingdao"
      },
      {
        "LocalName": "\u5317\u4eac",
        "RegionId": "cn-beijing"
      },
      {
        "LocalName": "\u9999\u6e2f",
        "RegionId": "cn-hongkong"
      },
      {
        "LocalName": "\u676d\u5dde",
        "RegionId": "cn-hangzhou"
      },
      {
        "LocalName": "\u7f8e\u56fd\u7845\u8c37",
        "RegionId": "us-west-1"
      }
    ]
  },
  "RequestId": "B0626530-F8F2-47E6-A786-BFF3F93EF766"
}
```

制表符分隔的文本 (text)

text 格式将命令行工具的输出组织为制表符分隔的行。此格式适合在传统UNIX文本工具（如 sed、grep 和 awk）以及 Windows PowerShell 中使用。text 输出格式遵循以下所示的基本结构。这些列根据底层 JSON 对象相应的键的名称按字母顺序排序。

```
-bash-4.1$ aliyuncli ecs DescribeRegions --output text
A612E1D4-5768-4280-9E87-03466DD4B0FF
REGION 深圳      cn-shenzhen
REGION 青岛      cn-qingdao
REGION 北京      cn-beijing
REGION 香港      cn-hongkong
REGION 杭州      cn-hangzhou
REGION 美国硅谷    us-west-1
```

ASCII 格式的表 (table)

table 格式便于用户阅读命令行工具的命令输出。示例如下：

```
REGION 美国硅谷    us-west-1
-bash-4.1$ aliyuncli ecs DescribeRegions --output table
+-----+-----+
| DescribeRegions |
+-----+-----+
| RequestId | 65E4B31A-14F2-471D-8216-819442EEAADC |
+-----+-----+
| Regions |
+-----+-----+
| Region |
+-----+-----+
| LocalName | RegionId |
+-----+-----+
| 深圳 | cn-shenzhen |
| 青岛 | cn-qingdao |
| 北京 | cn-beijing |
| 香港 | cn-hongkong |
| 杭州 | cn-hangzhou |
| 美国硅谷 | us-west-1 |
+-----+-----+
```

设置输出格式

您可以通过以下两种方式设置命令行工具的命令输出格式。

方法一：修改配置文件

在配置文件中设置 output 选项。以下示例将输出设置为 text：

```
[default] output=text
```

方法二：使用命令行

在命令行上使用 output 选项。以下示例将输出设置为 table：

```
$ aliyuncli ecs DescribeInstanceAttribute --InstanceId i-23rjh06vf --output table
```

脚本使用示例

命令行工具的简单 Shell 脚本

阿里云命令行工具是统一管理和配置阿里云资产的工具。您只需要简单的安装和配置，即可通过命令行方式同时管理多个阿里云产品和服务，简单快捷，是您上云的好帮手。但是用户需要记住很多 API 接口及参数，为此我们写了一个小脚本，集成了常见的几个操作方式，包括单个执行及批量执行。以 Linux 系统为例，该脚本的使用方法如下：

安装并配置命令行工具。操作步骤请参见[在线安装命令行工具（Linux/UNIX/Mac OS）](#)。

点击下载文件：[ecs.tar.gz](#)

执行如下命令：

```
tar zxvf ecs.tar.gz
sh ecs.sh
```

一级目录（单个以及批量执行启动、停止、重启、更换系统、重置、释放、重置密码），如下图所示：

```
bash-3.2# sh ecs.sh
Select the type of operation
 1 Start          8 Batch Start
 2 Stop           9 Batch Stop
 3 Restart        10 Batch Restart
 4 Replace System 11 Batch Replace System
 5 Reset System   12 Batch Reset System
 6 Release        13 Batch Release
 7 Reset PassWord 14 Batch Reset PassWord
15 Other Select   16 Exit

Please Input Select ID: 
```

二级目录（单个以及批量查询磁盘 ID、镜像 ID，批量可导出文件），如下图所示：

```
Select the type of operation
17 Query Disk ID    21 Create Snapshot
18 Query Image ID   22 Batch Create Snapshot
19 Batch Query Disk ID 51 Return to top
20 Query All Image ID 51 Exit
```

查询已订阅的镜像市场镜像信息

您可以利用命令行工具，通过如下脚本，查询已订阅的镜像市场镜像信息（镜像ID和镜像名称）。您可选择下载脚本或编辑脚本内容。

点击下载脚本：querySubscribedImageId.zip

脚本内容如下：

```
#!/bin/bash
tcount=`aliyuncli ecs DescribeImages --ImageOwnerAlias marketplace --output json --filter TotalCount`
pageNum=1
cat /dev/null >/tmp/imageids.txt
while ((tcount>0))
do
aliyuncli ecs DescribeImages --ImageOwnerAlias marketplace --filter Images.Image[*].ImageId --PageSize 100 --
PageNumber $pageNum --output json --filter Images.Image[*].ImageId | sed '1d' | sed '$d' | sed 's/ , //g' | sed
's/"//g' | sed 's/ //g' >>/tmp/imageids.txt
let pageNum++
let tcount-=100
done

cat /tmp/imageids.txt | while read line
do
isSubscribed=`aliyuncli ecs DescribeImages --ImageOwnerAlias marketplace --ImageId $line --filter
ImageIds.Image[*] --filter Images.Image[*].IsSubscribed --output json | sed '1d' | sed '$d' | sed 's/ //g'`
if [[ $isSubscribed = "true" ]];then
echo $line `aliyuncli ecs DescribeImages --ImageOwnerAlias marketplace --ImageId $line --filter ImageIds.Image[*]
--filter Images.Image[*].ImageName --output json | sed '1d' | sed '$d' | sed 's/ //g'` >>imagesInfo.txt
fi
done
native2ascii -encoding UTF-8 -reverse imagesInfo.txt imagesInfoCN.txt
rm -rf imagesInfo.txt
cat imagesInfoCN.txt`
```

编辑或下载脚本并赋权（chmod +x）后，您可以通过如下格式直接执行脚本，即会在脚本所在目录下生成包含已订阅镜像的信息文件 imagesInfoCN.txt。

用法示例：

```
./querySubscribedImageId.sh
```

输出示例：

```
# ./querySubscribedImageId.sh
m-23917oqoi "ASP/.NET运行环境 ( Windows200864位|IIS7.0 ) V1.0"
m-23n2589vc "Java运行环境 ( Centos64位|OpenJDK1.7 ) V1.0"
m-23u9mjtk "PW建站系统 ( Centos64位 ) V1.0"
```

查询指定区域下所有实例 ID 并输出到文件

您可以利用命令行工具，通过如下脚本，查询指定区域下所有实例 ID 并输出到文件。您可选择下载脚本或编辑脚本内容。

点击下载脚本：getVmList.zip

脚本内容如下：

```
#!/bin/bash

tcount=`aliyuncli ecs DescribeInstances --RegionId $1 --output json --filter TotalCount`
pageNum=1
cat /dev/null >vmList.txt
while ((tcount>0))
do
aliyuncli ecs DescribeInstances --RegionId $1 --PageSize 100 --PageNumber $pageNum --output json --filter
Instances.Instance[*].InstanceId | sed '1d' | sed '$d' | sed 's/ , //g' | sed 's/"//g' | sed 's/ //g'>>vmList.txt
let pageNum++
let tcount-=100
done
cat vmList.txt
```

编辑或下载脚本并赋权（chmod +x）后，您通过如下格式执行，即会在脚本所在目录下生成包含所需实例 ID 的文件 vmList.txt。

使用格式说明：

```
./getVmList.sh <区域id>
```

注意：区域ID可以通过DescribeRegions接口获取。

用法示例：

```
./getVmList.sh cn-hangzhou
```

输出示例：

```
#!/getVmList.sh cn-hangzhou
cn-hangzhou sg-227f611ss
cn-hangzhou sg-22esa2s7s
```

查询所有区域下未创建任何 VSwitch 的 VPC 列表

您可以利用命令行工具，通过如下脚本，查询所有区域下未创建任何虚拟交换机 VSwitch 的 VPC 实例 ID 列表。您可选择下载脚本或编辑脚本内容。

点击下载脚本：[getNoVSWitchVpcIds.zip](#)

脚本内容如下：

```
#!/bin/bash
for RegionId in `aliyuncli ecs DescribeRegions --filter Regions.Region[*].RegionId --output json | sed '1d' | sed '$d' | sed 's/ , //g' | sed 's/"//g' | sed 's/ //g'`
do
cat /dev/null >/tmp/vpcIDs.txt
pageNum=1
tcount=`aliyuncli ecs DescribeVpcs --RegionId $RegionId --output json --filter TotalCount`
while ((tcount>0))
do
aliyuncli ecs DescribeVpcs --RegionId $RegionId --filter Vpcs.Vpc[*].VpcId --PageSize 50 --PageNumber $pageNum --output json | sed '1d' | sed '$d' | sed 's/ , //g' | sed 's/"//g' | sed 's/ //g'>>/tmp/vpcIDs.txt
let pageNum++
let tcount-=50
done

cat /tmp/vpcIDs.txt | while read line
do
vSwitchCount=`aliyuncli ecs DescribeVSwitches --VpcId $line --output json --filter TotalCount`
if [[ $vSwitchCount -eq 0 ]];then
echo $RegionId $line
fi
done
done
```

编辑或下载脚本并赋权（`chmod +x`）后，您可以通过如下格式直接执行脚本，即会按区域 ID VpcId 格式成对输出所有地域下，未创建任何虚拟交换机 VSwitch 的 VPC 实例 ID。

使用格式说明：

```
./getNoVSWitchVpcIds.sh
```

输出示例：

```
[root@AliyunTest]# ./getNoVSWitchVpcIds.sh
cn-hangzhou vpc-23vyarrss
```

查询所有地域下未关联任何实例的安全组 ID

您可以利用命令行工具，通过如下脚本，查询所有区域下未关联任何实例的安全组，然后进行删除等处理。您可选择下载脚本或编辑脚本内容。

点击下载脚本：getUnUsedSecurityGroupId.zip

脚本内容如下：

```
#!/bin/bash
for RegionId in `aliyuncli ecs DescribeRegions --filter Regions.Region[*].RegionId --output json | sed '1d' | sed '$d' | sed 's/ , //g' | sed 's/"//g' | sed 's/ //g'`
do
cat /dev/null >/tmp/UnUsedSecurityGroupIds.txt
pageNum=1
tcount=`aliyuncli ecs DescribeSecurityGroups --RegionId $RegionId --output json --filter TotalCount`
while ((tcount>0))
do
aliyuncli ecs DescribeSecurityGroups --RegionId $RegionId --filter SecurityGroups.SecurityGroup[*].SecurityGroupId --PageSize 100 --PageNumber $pageNum --output json | sed '1d' | sed '$d' | sed 's/ , //g' | sed 's/"//g' | sed 's/ //g'>>/tmp/UnUsedSecurityGroupIds.txt
let pageNum++
let tcount-=100
done

cat /tmp/UnUsedSecurityGroupIds.txt | while read line
do
usedVMCount=`aliyuncli ecs DescribeInstances --RegionId $RegionId --SecurityGroupId $line --output json --filter TotalCount`
if [[ $usedVMCount -eq 0 ]];then
echo $RegionId " " $line
fi
done
done
```

编辑或下载脚本并赋权（chmod +x）后，您可以通过如下格式直接执行脚本，即会按地域和安全组成对输出所有地域下未使用的安全组 ID。

使用格式说明：

```
./getUnUsedSGInfo.sh
```

输出示例：

```
[root@AliyunTest]# ./getUnUsedSGInfo.sh
cn-qingdao sg-227f61lts
cn-shenzhen sg-22esa0f7s
```


多账户使用

注意：此功能适合需要管理很多账号和很多设备的用户使用。若您只需管理一个账号，请忽略此功能。

阿里云命令行工具支持多账户系统的管理操作，您可以根据需要配置多个 Access Key、Access Key Secret、Region Id 和 output format，在一个工具中管理多个不同账户的资源。关于如何配置账户，请参见配置命令行工具(公共云版)或配置命令行工具（专有云和专有域版）。

基本命令结构

多账户管理的基本命令结构如下：

```
aliyuncli configure [set/get/list] --profile profilename --key value --key1 value1
```

configure：表示要进行配置管理。

set: 可选操作，表示要设置一个配置的值。

get: 可选操作，表示要显示一个配置的值。

list: 可选操作，表示要列出一个 profile 所有的值。

profile: 全局参数，表示当前操作中采用的 profile 是什么。具体可以参考全局参数说明中对 profile 的说明，如果不带此选项，那么表示用 default 的账户。

key: 配置 profile 时，具体的 key。

value: 跟在 key 后面，配置中的 value。

账户分类

在配置文件中，我们分为两类账户：[default] 账户和 [profile profilename] 账户。在使用命令行工具时，不带 -profile 就采用 [default] 账户，带了 -profile profilename，就用 [profile profilename] 账户。

profile 示例如下：

```
aliyuncli configure ← 快速配置[default]账户
```

```
aliyuncli configure --profile test ← 快速配置[profile test]账户
```

set 示例如下:

```
aliyuncli configure set --output table --region cn-qingdao ← 设置[default]账户， output=table， region=cn-qingdao
```

```
aliyuncli configure set --output json --region cn-hangzhou --profile test1 ←设置[profile test1]账户， output = json， region = cn-hangzhou
```

get 示例如下:

```
aliyuncli configure get region ← 获取[default]账户的region值
```

输出：region = cn-hangzhou

```
aliyuncli configure get output region ← 获取[default]账户的output和region值
```

输出:

output = json

region = cn-hangzhou

```
aliyuncli configure get region --profile profile1 ← 获取[profile profile1]账户下的region值
```

list 示例如下:

```
aliyuncli configure list ← 列出[default]账户下的信息
```

```
aliyuncli configure list --profile profile1 ← 列出[profile profile1]账户下的信息
```

显示如下:

```
-bash-4.1$ aliyuncli configure list
Name      Value      Type      Location
----      -
Profile    None       None      None
Access_Key *****wQ7v credentials /home/xixi.xxx/.aliyuncli/credentials
Secret_Key *****fxGu credentials /home/xixi.xxx/.aliyuncli/credentials
Region     cn-qingdao configuration /home/xixi.xxx/.aliyuncli/configuration
Output     json       configuration /home/xixi.xxx/.aliyuncli/configuration
-bash-4.1$
```

高级过滤功能

注意：此功能适合有多个实例的用户使用。若您只有一个实例，请忽略此功能。

虽然阿里云开放 API 调用返回的数据能够以各种格式呈现出来，但依旧过于繁杂。所以阿里云命令行工具进一步做了过滤数据的功能，您可以使用 filter 功能从结果中直接过滤出想要的值，不论是使用还是二次开发都能够对结果进行处理，方便快捷。

从开放API调用过来的数据默认采用 "JSON" 格式，因此您在使用命令行工具时，可以根据 JSON 的特点，使用 filter 功能直接获取您想要的结果。

以 ECS DescribeRegions 为例，在命令行窗口执行如下命令，输出原始的 json 格式，如下图所示。

```
aliyuncli ecs DescribeRegions --output json
```



```
-bash-4.1$ aliyuncli ecs DescribeRegions --output json
{
  "Regions": {
    "Region": [
      {
        "LocalName": "\u6df1\u5733",
        "RegionId": "cn-shenzhen"
      },
      {
        "LocalName": "\u9752\u5c9b",
        "RegionId": "cn-qingdao"
      },
      {
        "LocalName": "\u5317\u4eac",
        "RegionId": "cn-beijing"
      },
      {
        "LocalName": "\u9999\u6e2f",
        "RegionId": "cn-hongkong"
      },
      {
        "LocalName": "\u676d\u5dde",
        "RegionId": "cn-hangzhou"
      },
      {
        "LocalName": "\u7f8e\u56fd\u7845\u837d",
        "RegionId": "us-west-1"
      }
    ]
  },
  "RequestId": "B0E4042E-A543-4DEB-88C4-7D769C460D54"
}
```

过滤1

您可以直接输入一个 key 值进行过滤。请执行如下命令进行过滤。过滤结果如下图所示。

```
aliyuncli ecs DescribeRegions -output json -filter Regions
```

```

}
-bash-4.1$ aliyuncli ecs DescribeRegions --output json --filter Regions
{
  "Region": [
    {
      "LocalName": "\u6df1\u5733",
      "RegionId": "cn-shenzhen"
    },
    {
      "LocalName": "\u9752\u5c9b",
      "RegionId": "cn-qingdao"
    },
    {
      "LocalName": "\u5317\u4eac",
      "RegionId": "cn-beijing"
    },
    {
      "LocalName": "\u9999\u6e2f",
      "RegionId": "cn-hongkong"
    },
    {
      "LocalName": "\u676d\u5dde",
      "RegionId": "cn-hangzhou"
    },
    {
      "LocalName": "\u7f8e\u56fd\u7845\u8c37",
      "RegionId": "us-west-1"
    }
  ]
}

```

过滤2

若 json 的值是一个 array 的情况，那么命令行工具支持数组下标格式。请执行如下命令进行过滤。过滤结果如下图所示。

```
aliyuncli ecs DescribeRegions -output json -filter Regions.Region[0]
```

```

}
-bash-4.1$ aliyuncli ecs DescribeRegions --output json --filter Regions.Region[0]
{
  "LocalName": "\u6df1\u5733",
  "RegionId": "cn-shenzhen"
}

```

另外，命令行工具还支持 '*' 的表达，表示所有结果的集合。Filter 的结果是多个时，会以 array 形式返回。请执行如下命令进行过滤。过滤结果如下图所示。

```
aliyuncli ecs DescribeRegions -output json -filter Regions.Region[*].RegionId
```

```

}
-bash-4.1$ aliyuncli ecs DescribeRegions --output json --filter Regions.Region[*].RegionId
[
  "cn-shenzhen",
  "cn-qingdao",
  "cn-beijing",
  "cn-hongkong",
  "cn-hangzhou",
  "us-west-1"
]
-bash-4.1$

```

过滤3

命令行工具还可以对显示结果精确到一个值进行过滤。请执行如下命令进行过滤。过滤结果如下图所示。

```
aliyuncli ecs DescribeRegions -output json -filter Regions.Region[3].RegionId
```

```
]
-bash-4.1$ aliyuncli ecs DescribeRegions --output json --filter Regions.Region[3].RegionId
"cn-hongkong"
-bash-4.1$
```