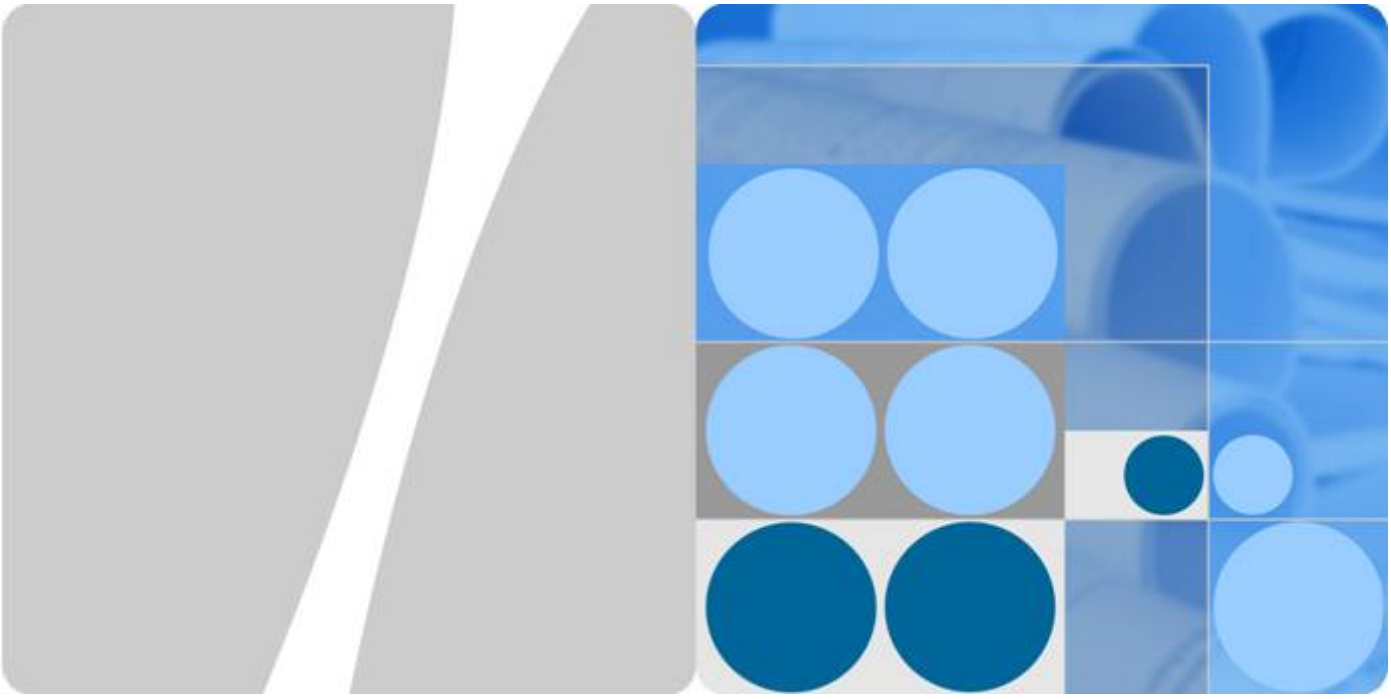


G42Cloud CLI

User Guide

Date **2022-06-30**

Contents

1 Service Overview

1.1 Privacy Statement

See **Privacy Policy**.

When using G42 Cloud Command Line Interface (G42Cloud CLI) for the first time, confirm whether to connect it to the Internet and whether to accept the *Privacy Policy*.

In some special scenarios such as G42Cloud CLI command execution with automation scripts, run the following command to agree to the privacy statement:

```
g42 configure set --cli-agree-privacy-statement=true
```

1.2 What Is G42Cloud CLI?

G42Cloud CLI is a command line tool for managing cloud service APIs released on API Explorer. With this tool, you can call open APIs of cloud services to manage and use your cloud service resources.

G42Cloud CLI provides a method for calling cloud service APIs through CLI. Before using G42Cloud CLI, you need to understand the target APIs. To get help using APIs, contact on-call personnel of the relevant cloud service.

When using G42Cloud CLI, you can search for APIs, debug them, and view their documentation on **API Explorer**.

G42Cloud CLI is flexible and easy to expand. It has the following features:

- Single executable file, which does not need installation. It can be used right after download and decompression.

- Compatibility with multiple OSs, including Linux, Windows, and macOS.
- High scalability. You can use this tool to encapsulate cloud service APIs for different functions and manage your resources with scripts.

NOTE

For details about how to download G42Cloud CLI, see **Getting Started**.

1.3 G42Cloud CLI Concepts

This topic describes common terms related to G42Cloud CLI.

- Command

  Commands are operation instructions provided by G42Cloud CLI to configure the working environment or call open APIs of cloud services.

  The format of an API calling command is as follows:

  g42 [options] <service> <operation> [--param1=paramValue1 --param2=paramValue2 ...]

  The format of a system command is as follows:

  g42 [options] <systemCommand> <operation> [--param1=paramValue1 --param2=paramValue2 ...]

  In the following command for querying Elastic Cloud Servers (ECSs), *service* is **ECS**, *operation* is **NovaListServers**, and the common information required for calling the API is obtained from the **profile** named **default**:

  ```
  g42 ECS NovaListServers --cli-profile=default
  ```

- Operation

  A unique name for a cloud service API released on API Explorer. You can query the operation list of a cloud service by using **API Explorer** or by running the **g42** *<service>* **--help** command.

- Profile

  Profiles store common information required for calling cloud service APIs. They can be defined by running G42Cloud CLI commands. All profiles constitute a configuration file, which is stored on your local host. When calling a cloud service API, you can specify a profile rather than manually inputting the common information.

  The common information that can be set in a profile includes the access key (**AK/SK**), region (**cli-region**), project ID (**cli-project-id**), and account ID (**cli-domain-id**).

- Default profile

  The profile to be used by default if no profile is specified in a command. G42Cloud CLI takes the profile that is last added or modified as the default. If the default profile is deleted, the earliest added profile among the remaining ones is used as the new default. You can run the **g42 configure set --cli-profile=***${profileName}* command to change the default profile.

- Parameter

  Parameters are classified into API parameters and G42Cloud CLI system parameters. API parameters are defined in cloud service APIs. System parameters are built-in parameters of G42Cloud CLI. The system parameters

are used in a fixed mode and have specific meanings. For details, see the **system parameter list**.

- Option

  Options are G42Cloud CLI system parameters that can be directly added to commands for calling APIs. Not all system parameters can be used as options. For details, see the **option list**.

- Metadata

  Details about cloud services and their APIs that are obtained by G42Cloud CLI during command execution to verify and parse parameters. Remotely obtained metadata is stored locally to reduce network I/O and improve command response. A file that stores metadata is called a metadata cache file. For details, see **Managing Metadata**.

  When using the **offline mode**, G42Cloud CLI downloads existing metadata, which is called an "offline metadata package".

1.4 Using G42Cloud CLI

Step 1 Download G42Cloud CLI.

G42Cloud CLI is installation-free and can be used right after download and decompression. It can work with Windows 64-bit, Linux AMD64, Linux Arm64, macOS AMD64, and macOS Arm64. **Download** the corresponding version based on the OS you use.

Step 2 Configure G42Cloud CLI.

For details, see **Initialize Configurations**.

Step 3 Obtain the commands for calling cloud service APIs.

There are two ways:

- (Recommended) From API Explorer

  View cloud service APIs on **API Explorer**, and enter the required parameters. Then obtain the example command from the **CLI Example** tab page.

- From G42Cloud CLI documentation

  To learn how to query cloud service commands, see **View and Run Cloud Service Operation Commands**. The methods of viewing commands in macOS and Linux are similar.

Step 4 Call cloud service APIs by using G42Cloud CLI.

Enter a complete API calling command and press **Enter**.

Step 5 (Optional) Integrate G42Cloud CLI commands into your custom scripts for automatic management of cloud service resources.

----End

2 Getting Started

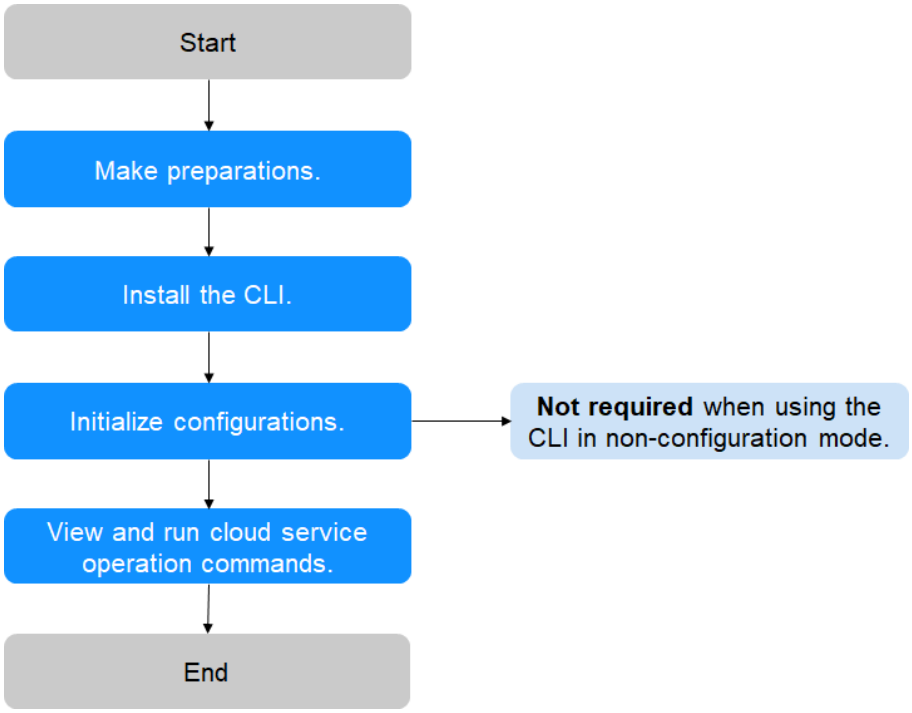
2.1 General Process

This section uses Windows as an example to describe how to use G42 Cloud Command Line Interface (G42Cloud CLI). The methods for using the CLI in Linux and macOS are similar.

The basic operations of G42Cloud CLI include installing the CLI, initializing configurations, and viewing and running cloud service operation commands.

Figure 2-1 shows the general process of using G42Cloud CLI.

Figure 2-1 General process



Step 1 Create an Identity and Access Management (IAM) user and grant them permissions. Then obtain an access key. For details, see **Step 1: Make Preparations**.

Step 2 To use G42Cloud CLI to call open APIs in **API Explorer** for managing and using cloud resources, download G42Cloud CLI for the corresponding OS version. For details, see **Overview**.

Step 3 Your identity information is obtained for authentication when you use G42Cloud CLI. If you are not **using G42Cloud CLI in non-configuration mode**, configure your authentication information. For details, see **Step 3: (Optional) Initialize Configurations**.

Step 4 Use G42Cloud CLI to manage and use your cloud service resources. For details, see **Step 4: View and Run Cloud Service Operation Commands**.

----End

2.2 Step 1: Make Preparations

Creating an IAM User and Granting Permissions

When you use G42Cloud CLI to manage and use your cloud resources, provide your IAM user identity information for authentication.

IAM users are created with an account to use cloud services. Each IAM user has their own identity credentials and uses cloud resources based on assigned permissions. IAM users cannot make payments themselves. You can use your account to pay their bills.

If you need an IAM user with specific permissions, **create a user** and **grant the user the required permissions**.

Obtaining an Access Key (AK/SK)

When you use G42Cloud CLI to manage and use your cloud resources, provide your IAM user identity information for authentication. To complete the initial configuration, **understand access keys and obtain one**.

2.3 Step 2: Install the CLI

2.3.1 Overview

- Complete installation in different environments using the following methods:
  - **Installing G42Cloud CLI in Windows**
  - **Installing G42Cloud CLI in Linux**
  - **Installing G42Cloud CLI in macOS**
  - **Configuring and Using G42Cloud CLI in Docker**
- Alternatively, download G42Cloud CLI that matches your OS from **Table 2-1** to a local directory, upload the CLI to your target location, and decompress and use it.

Table 2-1 Download addresses

OS	Download Link	Privacy Statement
Windows 64-bit	**G42Cloud-CLI-windows-amd64.zip** **G42Cloud-CLI-windows-amd64.zip_sha256**	See **Privacy Policy**.
Linux AMD64	**G42Cloud-CLI-linux-amd64.tar.gz** **G42Cloud-CLI-linux-amd64.tar.gz_sha256**	
Linux Arm64	**G42Cloud-CLI-linux-arm64.tar.gz** **G42Cloud-CLI-linux-arm64.tar.gz_sha256**	
macOS AMD64	**G42Cloud-CLI-mac-amd64.tar.gz** **G42Cloud-CLI-mac-amd64.tar.gz_sha256**	
macOS Arm64	**G42Cloud-CLI-mac-arm64.tar.gz** **G42Cloud-CLI-mac-arm64.tar.gz_sha256**	

NOTE

The download links of G42Cloud CLI do not contain **sha256**. Links that end with **sha256** are used to download software package verification files.

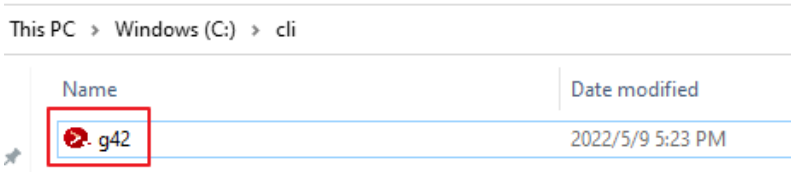
For example, the download link for Windows 64-bit is **G42Cloud-CLI-windows-amd64.zip**, and the verification file download link is **G42Cloud-CLI-windows-amd64.zip_sha256**.

2.3.2 Installing G42Cloud CLI in Windows

Step 1 **Download** G42Cloud CLI for Windows.

Step 2 Decompress the package to obtain the **g42.exe** file, as shown in **Figure 2-2**.

Figure 2-2 g42.exe file



Step 3 (Optional) Add the directory where G42Cloud CLI is located to the system environment variable **Path** so that you can run **g42** commands in any directory of Command Prompt.

1. If you use Windows 10 or Windows 8, search for and select **View advanced system settings**, and click **Environment Variables**.

   If you use Windows 7, right-click the **Computer** icon on the desktop, and choose **Properties** from the shortcut menu. Click the **Advanced system settings** link, and click **Environment Variables**.
2. On the **Environment Variables** page, select **Path** in the user variable list, and click **Edit**.
3. On the **Edit environment variable** page, click **New** and enter the path of the **g42.exe** file.
4. Click **OK** on the three pages successively to save the change.
5. (Optional) Open the Command Prompt, and run the following command to check whether the environment variable contains the directory where the **g42.exe** file is located. If yes, the configuration is successful.

   ```
   set path
   ```

Step 4 (Optional) If Step 3 is not performed, go to the directory where G42Cloud CLI is located, open the Command Prompt of Windows, and run the following command to check whether the CLI is installed successfully:

```
g42 version
```

If version information similar to the following is displayed, the installation is successful:

```
g42 version
Current G42Cloud CLI version: 3.2.8
```

----End

2.3.3 Installing G42Cloud CLI in Linux

G42Cloud CLI can be run on Linux AMD64 or Linux Arm64. You can install the CLI with one click or in several steps. If you choose step-by-step installation, use the installation command that matches your OS. Run the following command to query the OS used by your host:

```
echo $HOSTTYPE
```

If the command output is **x86_64**, use the download command for AMD64. If the command output is **aarch64**, use the download command for Arm64.

One-Click Installation

Run the following command to install G42Cloud CLI:

```
curl -sSL https://hby-g42-apixplorer-cli-obs.obs.ae-ad-1.g42cloud.com/cli/latest/g42_install.sh -o ./g42_install.sh && bash ./g42_install.sh
```

By default, G42Cloud CLI is downloaded to the **/usr/local/g42/** directory. A symbolic link is created for G42Cloud CLI in the **/usr/local/bin/** directory.

You can modify the file download directory based on the interaction information during command execution. If you do not have sufficient permissions, switch to user **root** and run the installation command again.

To use the default configuration and skip the interaction, add **-y** to the end of the command as follows:

```
curl -sSL https://hby-g42-apixplorer-cli-obs.obs.ae-ad-1.g42cloud.com/cli/latest/g42_install.sh -o ./g42_install.sh && bash ./g42_install.sh -y
```

Step-by-Step Installation

Perform the following steps:

Step 1 Run one of the following commands to download G42Cloud CLI:

- **curl** commands
  - Download command for AMD64
    ```
    curl -LO "https://hby-g42-apixplorer-cli-obs.obs.ae-ad-1.g42cloud.com/cli/latest/g42cloud-cli-linux-amd64.tar.gz"
    ```
  - Download command for Arm64
    ```
    curl -LO "https://hby-g42-apixplorer-cli-obs.obs.ae-ad-1.g42cloud.com/cli/latest/g42cloud-cli-linux-arm64.tar.gz"
    ```
- **wget** commands
  - Download command for AMD64
    ```
    wget "https://hby-g42-apixplorer-cli-obs.obs.ae-ad-1.g42cloud.com/cli/latest/g42cloud-cli-linux-amd64.tar.gz" -O g42cloud-cli-linux-amd64.tar.gz
    ```
  - Download command for Arm64
    ```
    wget "https://hby-g42-apixplorer-cli-obs.obs.ae-ad-1.g42cloud.com/cli/latest/g42cloud-cli-linux-arm64.tar.gz" -O g42cloud-cli-linux-arm64.tar.gz
    ```

Step 2 Decompress the tool package.

- Decompression command for AMD64
  ```
  tar -zxvf g42cloud-cli-linux-amd64.tar.gz
  ```
- Decompression command for Arm64
  ```
  tar -zxvf g42cloud-cli-linux-arm64.tar.gz
  ```

Step 3 (Optional) Create a symbolic link for G42Cloud CLI or add its directory to the **$PATH** variable so that you can use **g42** commands in any directory.

- Create a symbolic link for G42Cloud CLI in the **/usr/local/bin/** directory.
  ```
  ln -s $(pwd)/g42 /usr/local/bin/
  ```
  After creating a symbolic link in the **/usr/local/bin/** directory, you **do not need** to add the directory of CLI to the **$PATH** variable.
- Add the directory of G42Cloud CLI to the **$PATH** variable.
  ```
  export PATH=$PATH:$(pwd)/
  ```

Step 4 (Optional) Run the following command to turn on autocomplete:

```
g42 auto-complete on
```

If the following information is displayed, autocomplete is turned on. If the configuration does not take effect, run the **bash** command as prompted.

```
g42 auto-complete on
Autocomplete (bash only) turned on. If it does not work, run the `bash` command.
```

Step 5 (Optional) Run the following command to check whether the installation is successful:

```
g42 version
```

If version information similar to the following is displayed, the installation is successful:

```
g42 version
Current G42Cloud CLI version: 3.2.8
```

----End

2.3.4 Installing G42Cloud CLI in macOS

G42Cloud CLI can be run on macOS AMD64 or macOS Arm64. You can install the CLI with one click or in several steps. If you choose step-by-step installation, use the installation command that matches your OS. Run the following command to query the OS used by your host:

```
echo $HOSTTYPE
```

If the command output is **x86_64**, use the download command for AMD64. If the command output is empty, run the following command to check the OS used by your host:

```
uname -a
```

If the command output ends with **x86_64**, use the download command for AMD64. If the command output ends with **arm64**, use the download command for Arm64.

One-Click Installation

Run the following command to install G42Cloud CLI:

```
curl -sSL https://hby-g42-apixplorer-cli-obs.obs.ae-ad-1.g42cloud.com/cli/latest/g42_install.sh -o ./
g42_install.sh && bash ./g42_install.sh
```

By default, G42Cloud CLI is downloaded to the **/usr/local/g42/** directory. A symbolic link is created for G42Cloud CLI in the **/usr/local/bin/** directory.

You can modify the file download directory based on the interaction information during command execution. If you do not have sufficient permissions, switch to user **root** and run the installation command again.

To use the default configuration and skip the interaction, add **-y** to the end of the command as follows:

```
curl -sSL https://hby-g42-apixplorer-cli-obs.obs.ae-ad-1.g42cloud.com/cli/latest/g42_install.sh -o ./g42_install.sh && bash ./g42_install.sh -y
```

Step-by-Step Installation

Perform the following steps:

Step 1 Run one of the following commands to download G42Cloud CLI:

- **curl** commands
  - Download command for AMD64
    ```
    curl -LO "https://hby-g42-apixplorer-cli-obs.obs.ae-ad-1.g42cloud.com/cli/latest/g42cloud-cli-mac-amd64.tar.gz"
    ```
  - Download command for Arm64
    ```
    curl -LO "https://hby-g42-apixplorer-cli-obs.obs.ae-ad-1.g42cloud.com/cli/latest/g42cloud-cli-mac-arm64.tar.gz"
    ```
- **wget** commands
  - Download command for AMD64
    ```
    wget "https://hby-g42-apixplorer-cli-obs.obs.ae-ad-1.g42cloud.com/cli/latest/g42cloud-cli-mac-amd64.tar.gz" -O g42cloud-cli-mac-amd64.tar.gz
    ```
  - Download command for Arm64
    ```
    wget "https://hby-g42-apixplorer-cli-obs.obs.ae-ad-1.g42cloud.com/cli/latest/g42cloud-cli-mac-arm64.tar.gz" -O g42cloud-cli-mac-arm64.tar.gz
    ```

Step 2 Decompress the tool package.

- Decompression command for AMD64
  ```
  tar -zxvf g42cloud-cli-mac-amd64.tar.gz
  ```
- Decompression command for Arm64
  ```
  tar -zxvf g42cloud-cli-mac-arm64.tar.gz
  ```

Step 3 (Optional) Create a symbolic link for G42Cloud CLI or add its directory to the **$PATH** variable so that you can use **g42** commands in any directory.

- Create a symbolic link for G42Cloud CLI in the **/usr/local/bin/** directory.
  ```
  ln -s $(pwd)/g42 /usr/local/bin/
  ```
  After creating a symbolic link in the **/usr/local/bin/** directory, you **do not need** to add the directory of CLI to the **$PATH** variable.
- Add the directory of G42Cloud CLI to the **$PATH** variable.
  ```
  export PATH=$PATH:$(pwd)/
  ```

Step 4 (Optional) Run the following command to turn on autocomplete:

```
g42 auto-complete on
```

If the following information is displayed, autocomplete is turned on. If the configuration does not take effect, run the **bash** command as prompted.

```
g42 auto-complete on
Autocomplete (bash only) turned on. If it does not work, run the `bash` command.
```

Step 5 (Optional) Run the following command to check whether the installation is successful:

```
g42 version
```

If version information similar to the following is displayed, the installation is successful:

```
g42 version
Current G42Cloud CLI version: 3.2.8
```

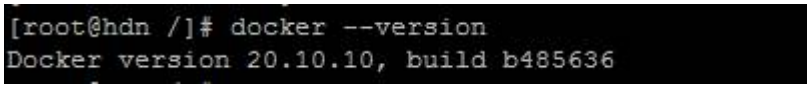
----End

2.3.5 Configuring and Using G42Cloud CLI in Docker

Configure and use G42Cloud CLI in Docker by performing the following steps. This procedure uses a Docker container running Ubuntu Linux as an example.

Before performing the following steps, ensure that you have installed Docker. For details about how to install Docker, visit the **Docker website**. To check whether Docker has been installed, run the following command:

```
docker --version
```



Step 1 Create a Dockerfile.

Create a directory and create a text file named **Dockerfile** in the directory. The file content is as follows:

```
FROM ubuntu:latest
RUN apt-get update -y && apt-get install curl -y
# Install G42Cloud CLI with one click.
RUN curl -sSL https://hby-g42-apixplorer-cli-obs.obs.ae-ad-1.g42cloud.com/cli/latest/g42_install.sh -o ./g42_install.sh && bash ./g42_install.sh -y
WORKDIR g42
```

NOTE

The name of the Dockerfile contains an uppercase letter D and does not have an extension. Only one Docker file can be saved in each directory.

Add the following content to the preceding Dockerfile to specify G42Cloud CLI as the program to run when the container is started:

```
ENTRYPOINT ["/usr/local/bin/g42"]
```

The container started by the Docker image built using this file (see **Method 2**) supports only a single G42Cloud CLI command.

Step 2 Build an image.

Run the following command in the directory to create a Docker image named **g42cli**:

```
docker build --no-cache -t g42cli .
```

NOTE

The period (.) at the end of the command cannot be omitted. It indicates that the Docker image is built in the current directory.

After the image is built successfully, run the following command to view the image:

```
docker images
```

Step 3 Use the image.

- Method 1: Create a background container using the **g42cli** image and run commands in the container.

  ```
  docker run -it -d --name g42cli g42cli
  ```

  Run the following command to view the started Docker container:

  ```
  docker ps
  ```

  Run the following command to enter the Docker container. Then you can use G42Cloud CLI in the same way as you use it on the host.

  ```
  docker exec -it g42cli /bin/bash
  ```

  Run the following command to exit the **g42cli** container:

  ```
  exit
  ```

  To stop the **g42cli** container, run the following command:

  ```
  docker stop g42cli
  ```

- Method 2: Create a temporary container using the **g42cli** image:

  a. Run the following command to create a temporary container:

  ```
  docker run --rm -it g42cli ${command}
  ```

  - If the Docker image is created using a Dockerfile that does not contain **ENTRYPOINT ["/usr/local/bin/g42"]**, specify **g42** as the program to run in the **docker run** command, for example, to query the current CLI version.

    ```
    docker run --rm -it g42cli g42 version
    ```

  - If the Docker image is created using a Dockerfile that contains **ENTRYPOINT ["/usr/local/bin/g42"]**, you do not need to specify a program to run. In this case, running the **docker run --rm -it g42cli** command is equivalent to running the **g42** command on the host. For example:

    ```
    docker run --rm -it g42cli version
    ```

    NOTE

    If the Docker image is created using a Dockerfile that contains **ENTRYPOINT ["/usr/local/bin/g42"]**, **docker run** only supports G42Cloud CLI commands.

  b. Create a temporary container and share files of a host (for example, a host running Linux) with the container.

  Mount a host directory to a container directory to share files of the host with the container.

  Example 1: Mount the **/root/.g42/** directory of the host to the **/root/.g42/** directory of the container to share the host configuration file with the container.

  ```
  docker run --rm -it -v /root/.g42/:/root/.g42/ g42cli ${command}
  ```

  For example, a Docker image has been created using a Dockerfile that contains **ENTRYPOINT ["/usr/local/bin/g42"]**. Run the following command to view the added profile:

  ```
  docker run --rm -it -v /root/.g42/:/root/.g42/ g42cli configure list
  ```

  Example 2: Mount the **/cli** directory of the host to the current directory of the container to share the files of the host with the container.

  ```
  docker run --rm -it -v /root/.g42/:/root/.g42/ -v /cli:$(pwd) g42cli ${command}
  ```

  c. Create a temporary container and share an environment variable of a host (for example, a host running Linux) with the container.

Use **-e** to mark the environment variable to be shared with the container.

```
docker run --rm -it -e ${envName} g42cli ${command}
```

NOTE

Set an alias for the command (a host running Linux is used as an example). For example, a Docker image has been created using a Dockerfile that contains **ENTRYPOINT ["/usr/local/bin/g42"]**. Run the following command to set the alias **g42** for the original command:

```
alias g42='docker run --rm -it g42cli'
```

Then you can run the original command using the alias.

Step 4 Update the image.

The G42Cloud CLI version in the image is the latest version when the image is created. To ensure that the image uses the latest version, **rebuild the image**.

Step 5 Remove the image.

Run the following command to remove the **g42cli** image:

```
docker rmi g42cli
```

----End

2.4 Step 3: (Optional) Initialize Configurations

This section uses Windows as an example to describe how to use G42Cloud CLI. The methods for using the CLI in Linux and macOS are similar.

If you want to **use G42Cloud CLI in non-configuration mode**, skip this step. To add a profile in non-interactive mode, see **Adding or Modifying a Profile**.

The G42Cloud CLI initialization command is used to store the **permanent** AK/SK and region information you often use (see the following table) in the configuration file, eliminating the need for frequently entering the information during operations.

Table 2-2 Initialized parameters

Parameter	Description
Access Key ID	(Required) Access key ID (AK) of the access key (permanent AK/SK).
Secret Access Key	(Required) Secret access key (SK) of the access key (permanent AK/SK).
Region	(Optional) Region, for example, **ae-ad-1**.

After typing in this command, press **Enter** to go to the interactive mode, and set the parameters as prompted.

g42 configure init

```
g42 configure init
Starting initialization. 'Secret Access Key' is anonymized. To obtain the parameter, see 'https://
```

```
docs.g42cloud.com/en-us/usermanual/g42cli/hcli_09.html'.
Access Key ID [required]: H9NNF********SG65MXW
Secret Access Key [required]: ****
Secret Access Key (again): ****
Region: ae-ad-1

*********************************************************
*****                                       *****
*****          Initialization successful          *****
*****                                       *****
*********************************************************
```

NOTE

- During initialization, you need to enter an SK for **Secret Access Key** twice. The value you enter is anonymized to ensure account security. The entered characters are displayed as asterisks (*) after you press **Enter**. After the configuration is complete, G42Cloud CLI **encrypts** sensitive authentication information in the profiles and **saves** the information to a local directory.
- If you run the initialization command again, the existing configuration file is deleted and a new file is generated. The path for storing the new configuration file is as follows:
  - Windows OS: **C:\Users***{Your Windows username}***\.g42\config.json**
  - Linux OS: **/home/***{Current username}***/.g42/config.json**
  - macOS: **/Users/***{Current username}***/.g42/config.json**

After initialization, you can run the following command to query the configuration information. G42Cloud CLI **anonymizes** the sensitive information of query results.

g42 configure show --cli-profile=default

```
g42 configure show --cli-profile=default
{
  "name": "default",
  "mode": "AKSK",
  "accessKeyId": "H9N****MXW",
  "secretAccessKey": "****",
  "securityToken": "",
  "xAuthToken": "",
  "expiresAt": "",
  "region": "ae-ad-1",
  "projectId": "",
  "domainId": "",
  "skipSecureVerify": "false",
  "readTimeout": 10,
  "connectTimeout": 5,
  "retryCount": 0,
  "agencyDomainId": "",
  "agencyDomainName": "",
  "agencyName": "",
  "sourceProfile": ""
}
```

2.5 Step 4: View and Run Cloud Service Operation Commands

This section uses Windows as an example to describe how to use G42Cloud CLI. The methods for using the CLI in Linux and macOS are similar.

After the initialization is complete, you can query the cloud services supported by G42Cloud CLI and run operation commands. The following uses the Elastic Cloud Server (ECS) API for querying ECS details as an example to describe how to query and run a command.

Step 1 Query the operations of a cloud service.

g42 <service> --help

As shown below, **Available Operations** lists the operations supported by ECS.

```
g42 ECS --help

G42Cloud CLI Version 3.2.8 Copyright(C) 2022-2022 www.g42cloud.com

Usage:
  g42 ECS <operation> --param1=value1 --param2=value2 ...

Service:
  ECS

Available Operations:
  AddServerGroupMember          MigrateServer                NovaShowServerAction
  AssociateServerVirtualIp      NovaAssociateSecurityGroup   NovaShowServerGroup
  AttachServerVolume            NovaAttachInterface          NovaShowServerInterface
  BatchAddServerNics            NovaAttachVolume             NovaShowServerMetadata
  ...
```

NOTE

- **During command execution, your authentication information will be used for requests, and the invocation of some APIs will incur charges.**
- Alternatively, you can query the operations supported by a cloud service on **API Explorer**.
- To query the cloud services supported by G42Cloud CLI, run the **g42 --help** command.

NOTICE

If your G42Cloud CLI uses the **offline mode**, it automatically downloads the **offline metadata package** when executing commands for the first time. Once downloaded, the package will not be automatically updated. To invoke newly launched services in the offline mode, run the **g42 meta download** command to update the offline metadata package, or run the **g42 configure set --cli-offline=false** command to switch to the **online mode**.

Step 2 Query the help information about an operation.

g42 <service> <operation> --help

Select **NovaShowServer** from the obtained ECS operation list to query the help information about the API. The API description, parameters, and parameter descriptions are displayed.

```
g42 ECS NovaShowServer --help

G42Cloud CLI Version 3.2.8 Copyright(C) 2022-2022 www.g42cloud.com

Service:
  ECS

Description:
This API is used to query details about an ECS by ECS ID.

Method:
  GET

Params:
```

```
 --cli-region
   required    string  Region where the API can be called. If no region is specified in the command, cli-region
in the current profile is used.
 --project_id
   required    string  path    Specifies the project ID. If no project ID is specified in the command, either the
parent project ID of the specified region in the authentication information or cli-project-id in the current
profile is used.
 --server_id
   required    string  path    Specifies the ECS ID.
 --OpenStack-API-Version
   optional    string  header  API with a microversion.
```

Step 3 Run the command to call the API and obtain the execution result.

g42 <service> <operation> [--param1=paramValue1 --param2=paramValue2 ...]

After you enter **cli-region** (region), **project_id** (project ID), and **server_id** (ECS ID) and press **Enter**, the ECS information is returned.

```
g42 ECS NovaShowServer --cli-region="ae-ad-1" --project_id="0dd8cb***************19b5a84546" --
server_id="4f06****-****-****-****-****04dd856a"
{
 "server": {
  "tenant_id": "0dd8cb***************19b5a84546",
  "metadata": {},
  "addresses": {
   "c865****-****-****-****-****efe7e8d8": [
    {
     "OS-EXT-IPS-MAC:mac_addr": "fa:**:**:**:**:**",
     "OS-EXT-IPS:type": "fixed",
     "addr": "192.***.*.**",
     "version": 4
    }
   ]
  },
  "OS-EXT-STS:task_state": null,
  "OS-DCF:diskConfig": "AUTO",
  "OS-EXT-AZ:availability_zone": "ae-ad-1a",
  "links": [
   {
    "rel": "self",
    "href": "https://ecs.ae-ad-1.g42cloud.com/v2.1/0dd8cb***************19b5a84546/servers/4f06****-****-****-
****-****04dd856a"
   },
   {
    "rel": "bookmark",
    "href": "https://ecs.ae-ad-1.g42cloud.com/0dd8cb***************19b5a84546/servers/4f06****-****-****-****-
****04dd856a"
   }
  ],
  "OS-EXT-STS:power_state": 4,
  "id": "4f06****-****-****-****-****04dd856a",
  "os-extended-volumes:volumes_attached": [
   {
    "id": "aed9****-****-****-****-****0e3219cf"
   }
  ],
  "OS-EXT-SRV-ATTR:host": "51f41ce46*******************************38b69b7aa4ea2a8",
  "image": {
   "links": [
    {
     "rel": "bookmark",
     "href": "https://ecs.ae-ad-1.g42cloud.com/0dd8cb***************19b5a84546/images/67f4****-****-****-****-
****38539e09"
    }
   ],
   "id": "67f4****-****-****-****-****38539e09"
  },
```

```
 "OS-SRV-USG:terminated_at": null,
 "accessIPv4": "",
 "accessIPv6": "",
 "created": "2022-05-10T12:56:36Z",
 "hostId": "51f41ce46*********************************38b69b7aa4ea2a8",
 "OS-EXT-SRV-ATTR:hypervisor_hostname": "cf199aabae*********************************bed586126e6f57",
 "flavor": {
  "links": [
   {
    "rel": "bookmark",
    "href": "https://ecs.ae-ad-1.g42cloud.com/0dd8cb***************19b5a84546/flavors/s6.medium.2"
   }
  ],
  "id": "s6.medium.2"
 },
 "key_name": null,
 "security_groups": [
  {
   "name": "Sys-**********"
  }
 ],
 "OS-EXT-STS:vm_state": "stopped",
 "user_id": "b4d561***************346deaf79e",
 "OS-EXT-SRV-ATTR:instance_name": "instance-*******",
 "name": "ecs-****",
 "OS-SRV-USG:launched_at": "2022-05-10T12:56:53.000000",
 "updated": "2022-05-13T08:05:17Z",
 "status": "SHUTOFF"
}
}
```


NOTE

- When using G42Cloud CLI to call APIs, you can **obtain example CLI commands on API Explorer**.
- G42Cloud CLI automatically obtains the account ID and project ID of an IAM user based on the user authentication information during API calling. The user does not need to specify **cli-region** in the command if it has already been specified in the configuration information.
- If a parameter value in the command is incorrect, an error message is displayed as follows:

```
g42 ECS ShowServer --project_id="0dd8cb***************19b5a84546" --cli-region="ae-ad-1" --server_id="abc"
{
  "error": {
    "message": "Instance[abc] could not be found.",
    "code": "Ecs.0114"
  }
}
For details, go to the API Error Center at 'https://console.g42cloud.com/apiexplorer/#/errorcenter?keyword=Ecs.0114&product=ECS'.
```

----End

2.6 Modifying an Initialized Profile

G42Cloud CLI supports the following authentication parameters in profiles: **access key (permanent AK/SK)**, **temporary security credentials (temporary AK/SK and SecurityToken)**, and **access token**. Temporary security credentials (temporary AK/SK and SecurityToken) and access tokens have limited validity.

During initialization, the profile name is **default**, and only a pair of permanent AK/SK can be configured. To use another authentication mode or modify parameters in the initialized profile, run the **g42 configure set --cli-**

profile=default --key1=value1... command. For details, see **Adding or Modifying a Profile**.

3 Configuration Management

3.1 Configuration Commands

G42 Cloud Command Line Interface (G42Cloud CLI) configuration commands are used to initialize configurations, and add, delete, modify, and query profiles. The following table lists the commands.

System Command	Function
g42 configure init	Initialize configurations.
g42 configure set	Add or modify a profile.
g42 configure list	Query the profile list.
g42 configure show	Query a specific profile.
g42 configure delete	Delete a specific profile.
g42 configure test	Check the configuration file format.
g42 configure clear	Delete all profiles.

3.2 Initializing Configurations

The G42Cloud CLI initialization command is used to store the **permanent** AK/SK and region information you often use in the configuration file, eliminating the need for frequently entering the information during operations. Run the following command to initialize configurations:

g42 configure init

After you type in the preceding command and press **Enter**, the system prompts you to enter the **access key ID** (required), **secret access key** (required), and default or common **region** (optional).

```
g42 configure init
Starting initialization. 'Secret Access Key' is anonymized. To obtain the parameter, see 'https://
docs.g42cloud.com/en-us/usermanual/g42cli/hcli_09.html'.
Access Key ID [required]: H9NNF********SG65MXW
Secret Access Key [required]: ****
Secret Access Key (again): ****
Region: ae-ad-1

*******************************************************
*****                                         *****
*****          Initialization successful          *****
*****                                         *****
*******************************************************
```

3.3 Adding or Modifying a Profile

3.3.1 Introduction

To manage resources in multiple regions or projects or use multiple accounts, add profiles to store fixed information in different scenarios. If you have configured multiple profiles, use **--cli-profile** to specify the name of the profile to be configured or used.

The following table lists the G42Cloud CLI profile parameters that can be modified.

Table 3-1 Profile parameters

Parameter	Description
cli-profile	Name of a profile. When you modify a profile without specifying this parameter, **the default profile** will be modified. To view the content of the default profile, run the **g42 configure show** command.

Parameter	Description
cli-mode	Authentication mode. Options: • **AKSK**: permanent AK/SK (access key) or temporary AK/SK and SecurityToken (security credentials). • **token**: A token has a validity period, and must be reconfigured when it expires. • **ecsAgency**: Temporary authentication information is automatically obtained when you use G42Cloud CLI to call cloud service APIs on an Elastic Cloud Server (ECS). • **agency**: After a trust relationship is established between two accounts, the delegated party can be authenticated using the agency to manage cloud services and resources of the delegating party.
cli-region	Region.
cli-access-key	Access key ID (AK). It must be modified together with SK.
cli-secret-key	Secret access key (SK). It must be modified together with AK.
cli-x-auth-token	The access token issued to an IAM user. It must be reconfigured when it expires. Otherwise, the authentication cannot be completed when you call cloud service APIs.
cli-security-token	Temporary security credential. It is required when a temporary AK/SK is used for identity authentication. It must be reconfigured when it expires. Otherwise, the authentication cannot be completed when you call cloud service APIs.
cli-lang	Language of G42Cloud CLI. Options: • **cn**: Chinese • **en**: English
cli-offline	Specifies whether to use G42Cloud CLI offline mode. Options: • **true** (use offline mode) • **false** (default value; use online mode)
cli-project-id	Project ID.
cli-domain-id	ID of the account to which an IAM user belongs. This parameter is required when you call APIs of global services in AK/SK mode.
cli-custom	Specifies whether to configure custom parameters. Options: • **true**: Configure the custom parameters in the profile. • **false**: Default value, which means to configure the system parameters in the profile.

Parameter	Description
cli-read-timeout	I/O timeout, in seconds. The default value is **10**, and the minimum value is **1**.
cli-connect-timeout	Request connection timeout, in seconds. The default value is **5**, and the minimum value is **1**.
cli-retry-count	Number of connection attempts. The value ranges from 0 to 5, and the default value is **0**.
cli-skip-secure-verify	Specifies whether to skip HTTPS certificate verification (not recommended). Options: • **true**: HTTPS certificate verification is skipped. This is **not recommended** for security purposes. • **false**: Default value, which means that HTTPS certificate verification is required.
cli-agency-domain-name	Account name of a delegating party. This parameter must be used together with **cli-agency-name**.
cli-agency-domain-id	Account ID of the delegating party. This parameter must be used together with **cli-agency-name**.
cli-agency-name	Agency name. It must be used together with **cli-agency-domain-id** or **cli-agency-domain-name**.
cli-source-profile	The profile that stores the authentication information of the delegated party. The value of **cli-source-profile** cannot be the current profile.
cli-agree-privacy-statement	Whether to agree to the privacy statement. Options: • **true**: Yes • **false**: No

NOTE

- The value of **--cli-custom** indicates the type of the parameter to be set in the profile. If **--cli-custom** is set to **false**, the system parameters will be set. If **--cli-custom** is set to **true**, the custom parameters will be set.
- When you set a system parameter for a profile that does not exist, the profile is created; if you do so for an existing profile, the profile is modified.
- You cannot set custom parameters for a profile that does not exist. You cannot set both system parameters and custom parameters in the same command.
- After you set system parameters or custom parameters for a profile, the profile then becomes **the default**.

Command for adding or modifying a profile:

g42 configure set --param1=paramValue1 --param2=paramValue2 ...

3.3.2 Setting System Parameters in a Profile

The following are examples of setting system parameters in a profile.

- Adding a profile

  When adding a profile, use **--cli-profile** to declare the profile name. For example:

  - Add a profile with AK/SK authentication mode. Authentication parameters: permanent AK (**cli-access-key**) and SK (**cli-secret-key**).
    ```
    g42 configure set --cli-profile=testAKSK --cli-mode=AKSK --cli-region=ae-ad-1 --cli-access-key=8NVT**********KIOV --cli-secret-key=VHMQjoC**************************lsk3cGf --cli-project-id=068119468***************af89d2e --cli-domain-id=094518e46***************cfbc4c0 --cli-read-timeout=10 --cli-connect-timeout=5
    ```
  - Add a profile with AK/SK authentication mode. Authentication parameters: temporary AK (**cli-access-key**), SK (**cli-secret-key**), and SecurityToken (**cli-security-token**).
    ```
    g42 configure set --cli-profile=testAKSKST --cli-mode=AKSK --cli-region=ae-ad-1 --cli-access-key=5FSU**********607T --cli-secret-key=VoyjgLh*************************qRc8pSq --cli-security-token=******* --cli-project-id=068119468***************af89d2e --cli-domain-id=094518e46***************cfbc4c0 --cli-read-timeout=10 --cli-connect-timeout=5
    ```

    For a profile with AK/SK authentication mode, if the **cli-security-token** parameter is configured in the profile, **cli-access-key** and **cli-secret-key** are temporary AK/SK. Otherwise, they are permanent AK/SK.
  - Add a profile with token authentication mode. Authentication parameter: access token (**cli-x-auth-token**).
    ```
    g42 configure set --cli-profile=testToken --cli-mode=token --cli-region=ae-ad-1 --cli-x-auth-token=******* --cli-project-id=068119468***************af89d2e --cli-domain-id=094518e46***************cfbc4c0 --cli-read-timeout=10 --cli-connect-timeout=5
    ```

    When you use a command prompt tool (such as **cmd.exe**) to run the preceding command, the command line may be incomplete if the value of **cli-x-auth-token** is too long. Therefore, check whether the command line content is complete before executing a command.
  - Add a profile with ecsAgency authentication mode. No authentication parameters are required.
    ```
    g42 configure set --cli-profile=testEcsAgency --cli-mode=ecsAgency --cli-region=ae-ad-1 --cli-read-timeout=10 --cli-connect-timeout=5
    ```

    When you use G42Cloud CLI to call cloud service APIs on an ECS, the CLI automatically obtains temporary authentication information based on the ECS agency.
  - Add a profile with agency authentication mode. Authentication parameters: delegating account name (**cli-agency-domain-name**) or ID (**cli-agency-domain-id**), agency name (**cli-agency-name**), and the profile (**cli-source-profile**) that stores the authentication information of the delegated account.

    i. Create a profile or use an existing one to store the authentication information of the delegated account. The profile uses AK/SK or token authentication.
    ```
    g42 configure set --cli-profile=testAKSK --cli-mode=AKSK --cli-region=ae-ad-1 --cli-access-key=8NVT**********KIOV --cli-secret-key=VHMQjoC**************************lsk3cGf
    ```

    ii. Create another profile that uses agency authentication. Use the profile configured in the previous step as the value of **cli-source-profile**.
    ```
    g42 configure set --cli-profile=testAgency --cli-mode=agency --cli-region=ae-ad-1 --cli-agency-domain-id=13534326******************5cf67b --cli-agency-name=****** --cli-source-profile=testAKSK --cli-read-timeout=10 --cli-connect-timeout=5
    ```

    When you add a profile with agency authentication mode, both **cli-agency-domain-id** (or **cli-agency-domain-name**) and **cli-agency-name**

must be configured. The value of **cli-profile** cannot be the same as that of **cli-source-profile** in the command. The profile specified by **cli-source-profile** must exist.

- Changing the **default profile**

  If you have configured multiple profiles, you can run the following command to change the default profile:

  ```
  g42 configure set --cli-profile=test
  ```

- Changing the language

  This change will take effect for all profiles. Run the following command to change the language:

  ```
  g42 configure set --cli-lang=cn
  ```

- Switching between online and offline modes

  This change will take effect for all profiles. Run the following command to switch to the offline mode:

  ```
  g42 configure set --cli-offline=true
  ```

- Agreeing to the privacy statement

  This change will take effect for all profiles. When using G42Cloud CLI for the first time, you will be prompted to read and agree to the privacy statement. In some special scenarios such as command execution with automation scripts, run the following command to agree to the privacy statement:

  ```
  g42 configure set --cli-agree-privacy-statement=true
  ```

- Changing the authentication mode of a profile

  You can configure an AK/SK, token, and **agency** in a profile. Run the following command to specify an authentication mode for a profile:

  ```
  g42 configure set --cli-profile=test --cli-mode=token
  ```

- Changing the region of a profile

  ```
  g42 configure set --cli-profile=test --cli-region=ae-ad-1
  ```

- Modifying the **AK/SK** of a profile

  AK and SK must be modified at the same time.

  ```
  g42 configure set --cli-profile=test --cli-access-key=8NVT**********KIOV --cli-secret-key=VHMQjoC**********lsk3cGf
  ```

- Changing the project ID of a profile

  A project ID must be specified when you call the APIs of certain cloud services. To call these APIs, **obtain project IDs** and add those you will often use to a profile.

  ```
  g42 configure set --cli-profile=test --cli-project-id=068119468**********af89d2e
  ```

- Modifying the request connection timeout of a profile

  ```
  g42 configure set --cli-profile=test --cli-connect-timeout=5
  ```

- Modifying the I/O timeout of a profile

  ```
  g42 configure set --cli-profile=test --cli-read-timeout=10
  ```

- Modifying the maximum number of connection attempts of a profile

  ```
  g42 configure set --cli-profile=test --cli-retry-count=3
  ```

CAUTION

Setting the retry count with **--cli-retry-count** may cause idempotence and cyclic API calls. Exercise caution when using this option for resource creation APIs.

- Changing the configuration of skipping HTTPS certificate verification in a profile
  ```
  g42 configure set --cli-profile=test --cli-skip-secure-verify=true
  ```

NOTE

You can modify multiple parameters of a profile at a time. For example, to modify **cli-project-id** and **cli-region**, run the following command:

g42 configure set --cli-profile=test --cli-project-id=0681194**********f89d2e --cli-region=ae-ad-1

3.3.3 Setting Custom Parameters in a Profile

G42Cloud CLI allows you to set custom parameters in a profile. This helps you manage parameters that are frequently used in commands, have long values, or have values to be encrypted. Run the following command to enter the interactive mode and add or modify custom parameters in a profile:

g42 configure set [--cli-profile=${profileName}] --cli-custom=true

```
g42 configure set --cli-profile=test --cli-custom=true
  1. Enter the custom parameter name: projectId
  2. Encrypt the custom parameter for storage? (y/N): n
  3. Enter the parameter value: 0681000000000000000000000f89d2e
  Custom parameter saved. To stop setting more custom parameters, press `Ctrl+C`.
  1. Enter the custom parameter name: password
  2. Encrypt the custom parameter for storage? (y/N): y
  3. Enter the parameter value: ****
Custom parameter saved. To stop setting more custom parameters, press `Ctrl+C`.
```

A custom parameter cannot exceed 128 characters. During interactive setting, determine whether to encrypt the custom parameter for storage according to your requirements. To ensure the security of your custom parameters that need to be encrypted for storage, the system **anonymizes** the parameter values you enter. The entered characters are displayed as asterisks (****) after you press **Enter**.

To use a custom parameter, replace the original value in the command with **custom.*${Name of the custom parameter}***. For example:

```
g42 ECS NovaListServers --cli-profile=test --project_id="custom.projectId"
```

When you use an encrypted custom parameter, G42Cloud CLI **anonymizes** the parameter value in places where the value may be displayed, for example, in the printed request URL of debugging information or in printed body parameters in dryrun mode.

3.4 Querying the Profile List

3.4.1 Querying System Parameters in the Profile List

G42Cloud CLI supports multiple profiles. Run the following command to query system parameters in the profile list. The query result can be output in a table or in JSON or TSV format. By default, the query result is output in JSON format.

g42 configure list

```
g42 configure list
{
  "language": "cn",
```

```
 "offline": "false",
 "agreePrivacy": "true",
 "current": "test",
 "profiles": [
  {
    "name": "default",
    "mode": "AKSK",
    "accessKeyId": "H9N****MXW",
    "secretAccessKey": "****",
    "securityToken": "",
    "xAuthToken": "",
    "expiresAt": "",
    "region": "ae-ad-1",
    "projectId": "060576**********134588f135",
    "domainId": "",
    "skipSecureVerify": "false",
    "readTimeout": 10,
    "connectTimeout": 5,
    "retryCount": 0,
    "agencyDomainId": "",
    "agencyDomainName": "",
    "agencyName": "",
    "sourceProfile": ""
  },
  {
    "name": "test",
    "mode": "AKSK",
    "accessKeyId": "8NV****IOV",
    "secretAccessKey": "****",
    "securityToken": "",
    "xAuthToken": "",
    "expiresAt": "",
    "region": "ae-ad-1",
    "projectId": "",
    "domainId": "",
    "skipSecureVerify": "false",
    "readTimeout": 10,
    "connectTimeout": 5,
    "retryCount": 0,
    "agencyDomainId": "",
    "agencyDomainName": "",
    "agencyName": "",
    "sourceProfile": ""
  }
 ]
}
```

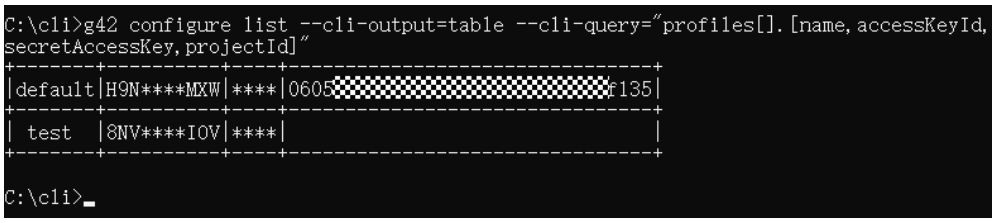

The values of sensitive parameters are **anonymized**.

Use **--cli-query** to filter the query result, and use **--cli-output** to specify the output format. The default output format is JSON. For example, output the query result in JSON format and filter the AK/SK in each profile:

```
g42 configure list --cli-output=json --cli-query="profiles[].{Name:name,AK:accessKeyId,SK:secretAccessKey}"
[
 {
   "AK": "H9N****MXW",
   "Name": "default",
   "SK": "****"
 },
 {
   "AK": "8NV****IOV",
   "Name": "test",
   "SK": "****"
 }
]
```

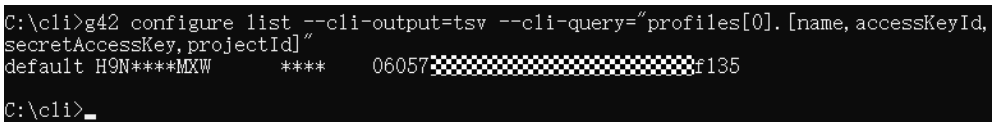
To output the query result in a table, add **--cli-output=table** in the command.

g42 configure list --cli-output=table --cli-query="profiles[].[name,accessKeyId,secretAccessKey,projectId]"



To output the query result in TSV format, add **--cli-output=tsv** in the command.

g42 configure list --cli-output=tsv --cli-query="profiles[0].[name,accessKeyId,secretAccessKey,projectId]"



3.4.2 Querying Custom Parameters in the Profile List

Run the following command to query custom parameters in the profile list of G42Cloud CLI. The query result can be output in a table or in JSON or TSV format. By default, the query result is output in JSON format.

g42 configure list --cli-custom=true

```
g42 configure list --cli-custom=true
{
  "current": "test",
  "profiles": [
    {
      "name": "default",
      "custom": null
    },
    {
      "name": "test",
      "custom": {
        "password": {
          "value": "****",
          "isEncrypted": true
        },
        "projectId": {
          "value": "0681000000000000000000000f89d2e",
          "isEncrypted": false
        }
      }
    }
  ]
}
```

Encrypted custom parameter values are **anonymized**.

Use **--cli-query** to filter the query result, and use **--cli-output** to specify the output format. The default output format is JSON. In the following example, all custom parameters of a profile named **test** are queried, and the query result is output in JSON format:

```
g42 configure list --cli-custom=true --cli-query="profiles[?name=='test'].custom" --cli-output=json
[
  {
    "password": {
```

```
    "isEncrypted": true,
    "value": "****"
  },
  "projectId": {
    "isEncrypted": false,
    "value": "0681000000000000000000000f89d2e"
  }
 }
]
```

3.5 Querying a Profile

3.5.1 Querying System Parameters in a Profile

Use **--cli-profile** to specify the name of the profile to be queried. If no profile is specified, the **default profile** is queried. The query result can be output in a table or in JSON or TSV format. By default, the query result is output in JSON format. The command for querying system parameters in a profile is as follows:

g42 configure show [--cli-profile=${profileName}]

```
g42 configure show --cli-profile=test
{
  "name": "test",
  "mode": "AKSK",
  "accessKeyId": "8NV****IOV",
  "secretAccessKey": "****",
  "securityToken": "",
  "xAuthToken": "",
  "expiresAt": "",
  "region": "ae-ad-1",
  "projectId": "",
  "domainId": "",
  "skipSecureVerify": "false",
  "readTimeout": 10,
  "connectTimeout": 5,
  "retryCount": 0,
  "agencyDomainId": "",
  "agencyDomainName": "",
  "agencyName": "",
  "sourceProfile": ""
}
```

3.5.2 Querying Custom Parameters in a Profile

Use **--cli-profile** to specify the name of the profile to be queried. The query result can be output in a table or in JSON or TSV format. By default, the query result is output in JSON format. The command for querying custom parameters in a profile is as follows:

g42 configure show [--cli-profile=${profileName}] --cli-custom=true

```
g42 configure show --cli-profile=test --cli-custom=true
{
  "password": {
    "value": "****",
    "isEncrypted": true
  },
  "projectId": {
    "value": "0681000000000000000000000f89d2e",
    "isEncrypted": false
  }
}
```

3.6 Deleting a Profile

3.6.1 Deleting a Profile and Its Custom Parameters

Use **--cli-profile** to specify the name of the profile to be deleted. The command for deleting a profile is as follows:

g42 configure delete --cli-profile=${profileName}

```
g42 configure delete --cli-profile=test
Profile test deleted. The default profile is default.
```

Deleting a profile will also delete its system parameters and custom parameters.

3.6.2 Deleting Custom Parameters in a Profile

Use **--cli-profile** to specify the name of the profile to be deleted. The command for deleting custom parameters in a profile is as follows:

- Deleting a **specific** custom parameter in a profile

  g42 configure delete --cli-profile=${profileName} [--cli-custom=true] --cli-custom-key=${customName}

  ```
  g42 configure delete --cli-profile=test --cli-custom-key=projectId
  From profile test, custom parameter projectId is deleted.
  ```

- Deleting **all** custom parameters in a profile

  g42 configure delete --cli-profile=${profileName} --cli-custom=true

  ```
  g42 configure delete --cli-profile=test --cli-custom=true
  All custom parameters in profile test are deleted.
  ```

3.7 Checking the Configuration File Format

To check the format of your local configuration file, run the following command:

g42 configure test

```
g42 configure test
The configuration file has the correct format.
```

3.8 Deleting All Profiles

Run the following command to delete all profiles:

g42 configure clear

```
g42 configure clear
Delete all profiles? (y/N): y
Profiles deleted.
```

4 HTTP Proxy

4.1 Configuring the HTTP Proxy

When you use G42Cloud CLI to manage and access cloud resources, cloud service APIs may fail to be called due to the intranet access restriction of your company. In this case, you can use the HTTP proxy by configuring the environment variables **http_proxy** and **https_proxy**.

Configuring the Environment Variable http_proxy

Run the command that matches your OS, to configure the environment variable **http_proxy**.

- Linux or macOS
  - Temporary environment variable

    Run the following command to configure **http_proxy**:

    ```
    export http_proxy="http://username:password@proxyServer:port"
    ```

    In the preceding command, **proxyServer** indicates the domain name (if resolvable) or IP address of the proxy server, and **port** indicates the port number. Enter a username and password if required, or run the following command:

    ```
    export http_proxy="http://proxyServer:port"
    ```

  - Permanent environment variable

    To permanently configure **http_proxy** in an environment variable that takes effect only for the current user, add the following content to the **~/.bashrc** file:

    ```
    export http_proxy="http://username:password@proxyServer:port"
    ```

    In the preceding command, **proxyServer** indicates the domain name (if resolvable) or IP address of the proxy server, and **port** indicates the port number. Enter a username and password if required, or run the following command:

    ```
    export http_proxy="http://proxyServer:port"
    ```

After configuring the permanent environment variable, log out and log in again for the variable to take effect.

- Windows
  - Using Command Prompt
    - Temporary environment variable

      Run the following command to configure **http_proxy**:
      ```
      set http_proxy=http://username:password@proxyServer:port
      ```
      In the preceding command, **proxyServer** indicates the domain name (if resolvable) or IP address of the proxy server, and **port** indicates the port number. Enter a username and password if required, or run the following command:
      ```
      set http_proxy=http://proxyServer:port
      ```
    - Permanent environment variable

      Run the following command to permanently configure **http_proxy** in an environment variable that takes effect only for the current user:
      ```
      setx "http_proxy" "http://username:password@proxyServer:port"
      ```
      In the preceding command, **proxyServer** indicates the domain name (if resolvable) or IP address of the proxy server, and **port** indicates the port number. Enter a username and password if required, or run the following command:
      ```
      setx "http_proxy" "http://proxyServer:port"
      ```
      The permanent environment variable will take effect on the clients opened later.
  - Using PowerShell
    - Temporary environment variable

      Run the following command to configure **http_proxy**:
      ```
      $env:http_proxy="http://username:password@proxyServer:port"
      ```
      In the preceding command, **proxyServer** indicates the domain name (if resolvable) or IP address of the proxy server, and **port** indicates the port number. Enter a username and password if required, or run the following command:
      ```
      $env:http_proxy="http://proxyServer:port"
      ```
    - Permanent environment variable

      Run the following command to permanently configure **http_proxy** in an environment variable that takes effect only for the current user:
      ```
      [environment]::SetEnvironmentvariable("http_proxy", "http://
      username:password@proxyServer:port", "User")
      ```
      In the preceding command, **proxyServer** indicates the domain name (if resolvable) or IP address of the proxy server, and **port** indicates the port number. Enter a username and password if required, or run the following command:
      ```
      [environment]::SetEnvironmentvariable("http_proxy", "http://proxyServer:port", "User")
      ```
      The permanent environment variable will take effect on the clients opened later.

Configuring the Environment Variable https_proxy

Run the command that matches your OS, to configure the environment variable **https_proxy**.

- Linux or macOS
  - Temporary environment variable

    Run the following command to configure **https_proxy**:
    ```
    export https_proxy="https://username:password@proxyServer:port"
    ```
    In the preceding command, **proxyServer** indicates the domain name (if resolvable) or IP address of the proxy server, and **port** indicates the port number. Enter a username and password if required, or run the following command:
    ```
    export https_proxy="https://proxyServer:port"
    ```
  - Permanent environment variable

    To permanently configure **https_proxy** in an environment variable that takes effect only for the current user, add the following content to the **~/.bashrc** file:
    ```
    export https_proxy="https://username:password@proxyServer:port"
    ```
    In the preceding command, **proxyServer** indicates the domain name (if resolvable) or IP address of the proxy server, and **port** indicates the port number. Enter a username and password if required, or run the following command:
    ```
    export https_proxy="https://proxyServer:port"
    ```
    After configuring the permanent environment variable, log out and log in again for the variable to take effect.
- Windows
  - Using Command Prompt
    - Temporary environment variable

      Run the following command to configure **https_proxy**:
      ```
      set https_proxy=https://username:password@proxyServer:port
      ```
      In the preceding command, **proxyServer** indicates the domain name (if resolvable) or IP address of the proxy server, and **port** indicates the port number. Enter a username and password if required, or run the following command:
      ```
      set https_proxy=https://proxyServer:port
      ```
    - Permanent environment variable

      Run the following command to permanently configure **https_proxy** in an environment variable that takes effect only for the current user:
      ```
      setx "https_proxy" "https://username:password@proxyServer:port"
      ```
      In the preceding command, **proxyServer** indicates the domain name (if resolvable) or IP address of the proxy server, and **port** indicates the port number. Enter a username and password if required, or run the following command:
      ```
      setx "https_proxy" "https://proxyServer:port"
      ```
      The permanent environment variable will take effect on the clients opened later.
  - Using PowerShell

- Temporary environment variable

  Run the following command to configure **https_proxy**:

  ```
  $env:https_proxy="https://username:password@proxyServer:port"
  ```

  In the preceding command, **proxyServer** indicates the domain name (if resolvable) or IP address of the proxy server, and **port** indicates the port number. Enter a username and password if required, or run the following command:

  ```
  $env:https_proxy="https://proxyServer:port"
  ```

- Permanent environment variable

  Run the following command to permanently configure **https_proxy** in an environment variable that takes effect only for the current user:

  ```
  [environment]::SetEnvironmentvariable("https_proxy", "https://
  username:password@proxyServer:port", "User")
  ```

  In the preceding command, **proxyServer** indicates the domain name (if resolvable) or IP address of the proxy server, and **port** indicates the port number. Enter a username and password if required, or run the following command:

  ```
  [environment]::SetEnvironmentvariable("https_proxy", "https://proxyServer:port", "User")
  ```

  The permanent environment variable will take effect on the clients opened later.

4.2 Resolving the x509 Certificate Error

When you use G42Cloud CLI to manage and access cloud resources after configuring the HTTP proxy, the x509 error may occur when you call a cloud service API. The reason is that your proxy has modified the root certificate issued by the HTTPS website. As a result, the certificate becomes invalid when you access HTTPS resources on external networks.

To resolve this problem, import the certificate issued by your company to the trusted CA certificate of the OS. Alternatively, add **--cli-skip-secure-verify=true** to your command to **skip certificate verification** (not recommended).

5 Commonly Used Commands

5.1 Viewing Help Information

Add the **--help** parameter to a command to view help information. For example:

- g42 --help

  View the instructions for using G42Cloud CLI and the supported cloud services.
- g42 *<service>* --help

  View the operation list of a cloud service. For example, to view the operation list of ECS, run **g42 ECS --help**.
- g42 *<service>* *<operation>* --help

  View details about a cloud service API. For example, to view details about the ECS API with operation **ShowServer**, run **g42 ECS ShowServer --help**.

5.2 Turning On/Off Autocomplete

After you turn on autocomplete for G42Cloud CLI, if you press **Tab** when entering a command, a prompt will appear to help you autocomplete the command.

To turn on autocomplete, run the following command:

g42 auto-complete on

```
g42 auto-complete on
Autocomplete (bash only) turned on. If it does not work, run the `bash` command.
```

To turn off autocomplete, run the following command:

g42 auto-complete off

```
g42 auto-complete off
Autocomplete turned off. If it does not work, run the `bash` command.
```

NOTE

- Autocomplete is supported only in Bash.
- If the prompted parameter name contains **[N]**, which indicates an index, replace it with a number. If the prompted parameter name contains **{*}**, which indicates a custom parameter name, replace it with a string that does not contain periods (.).

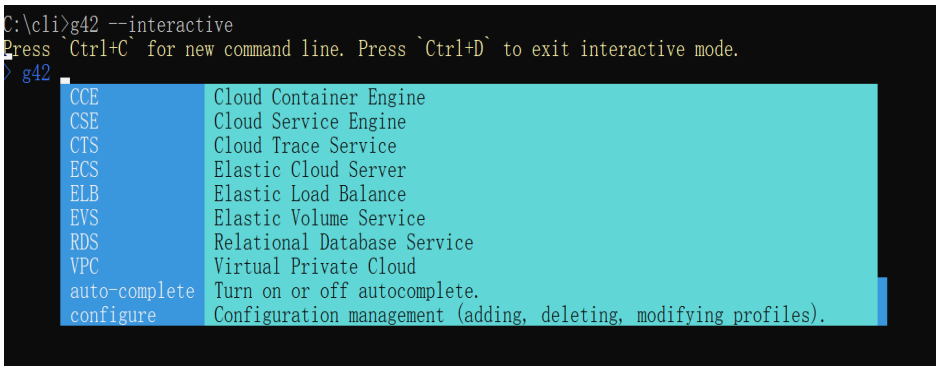
5.3 Using Interactive Prompts

G42Cloud CLI provides powerful interactive prompts to facilitate command building on different platforms. Add **--interactive** in a command to enter the interactive mode. In this mode, the system displays cloud service names, API operations, parameters, and descriptions based on the information you enter. In addition, the system autocompletes the values of some parameters.

- Use the interactive mode as follows:
  - Use the **Tab** key or the up and down arrows to select the content to be autocompleted.
  - Press the space bar to add the selected content to the current command.
  - After entering the interactive mode, press **Ctrl+C** to switch to a new command line. To exit the interactive mode, press **Ctrl+D**.
- The following contents will be prompted and autocompleted in the interactive mode:
  - Cloud services

    When you enter the interactive mode for the first time, the list of cloud services and system commands is automatically displayed.

    Figure 5-1 List of cloud services and system commands



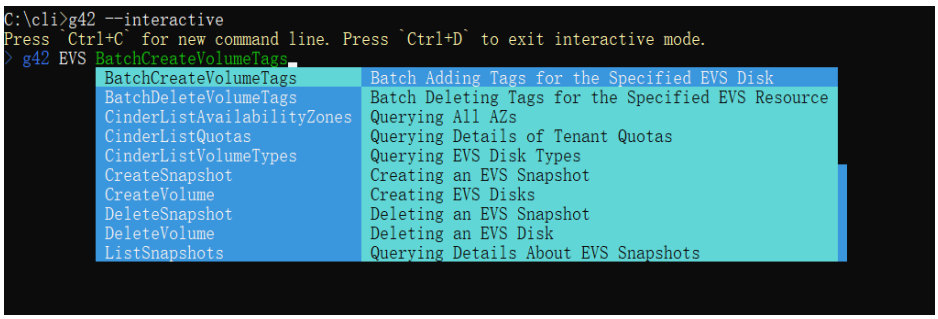
    In the cloud service list, the left column displays cloud service abbreviations or system commands, and the right column displays the full service names or system command descriptions.

- APIs

  After you enter a valid cloud service name or system command, the API list of the service or the subcommands (or parameters) of the system command are displayed.

  In the system command list, the left column displays the subcommands (or parameters) of the system command, and the right column displays the descriptions of the subcommands (or parameters).

Figure 5-2 Cloud service API list



In the cloud service API list, the left column displays API operation names, and the right column displays API descriptions.

Figure 5-3 Subcommands of a system command

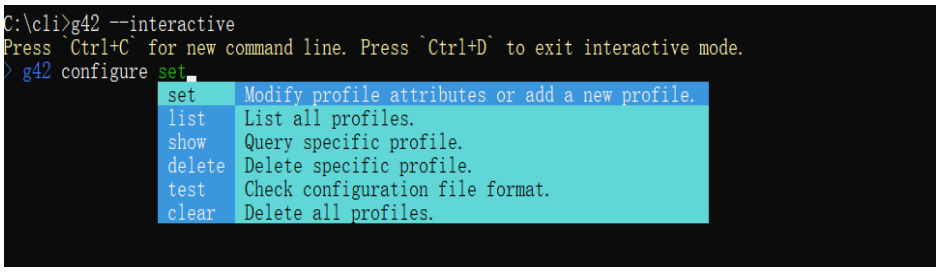
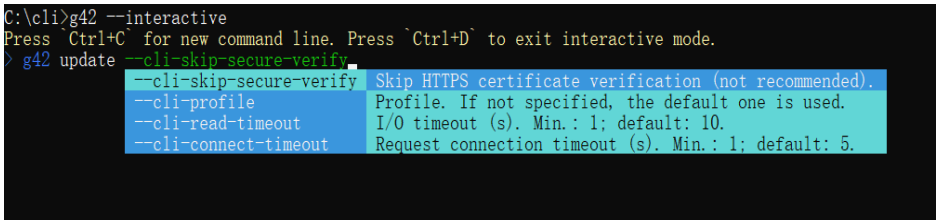


Figure 5-4 System command parameters



If the entered cloud service name or system command is invalid, no prompt is displayed.

- Parameters

  After you enter a valid cloud service name and API operation, or a valid system command and subcommand, the parameter list of the API or subcommand is displayed.

Figure 5-5 Cloud service API parameter list

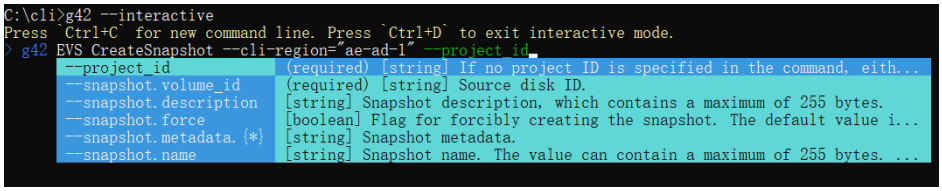
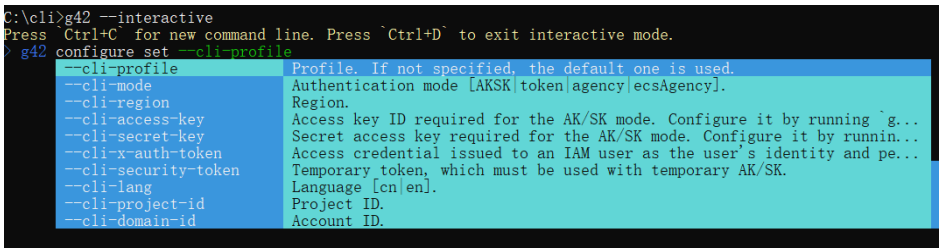


Figure 5-6 Subcommand parameter list of a system command



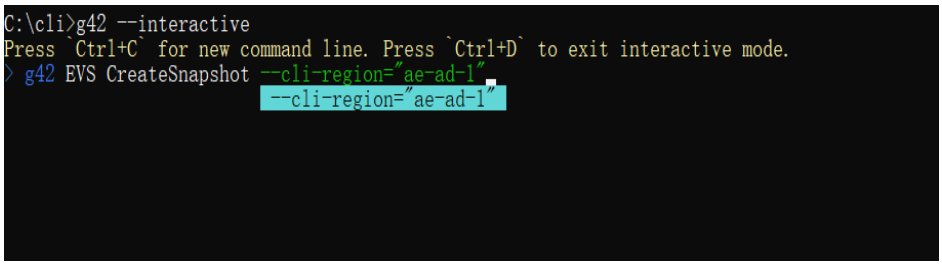
In the parameter list, the left column displays parameter names, and the right column displays their descriptions.

If the entered API operation or subcommand is invalid, or the entered parameter name is invalid, no prompt is displayed.

- Values

  In interactive mode, if you enter an equal sign (=) after a parameter name, the allowed values or the default value of the parameter is displayed. This value prompt is available only for certain parameters. For example, if you enter **--cli-region=** in an API calling command, the regions where the API is available are displayed.

  Figure 5-7 Regions where an API is available

- Shortcut keys in interactive mode
  - **Ctrl+W**: Delete the word before the cursor.
  - **Ctrl+K**: Delete the content after the cursor.
  - **Ctrl+U**: Delete the content before the cursor.
  - **Ctrl+L**: Clear the screen.

NOTE

- If no profile is added, or the region specified in the **default profile** is not supported by the API in the command, select a region from the **cli-region** list prompted for the API. Then, G42Cloud CLI continues to display the parameter list of the API.
- In interactive prompts, parameters except custom map type parameters (containing **{*}** in their names) will not be prompted again once entered. If a parameter name contains **[N]**, which indicates an index, replace the character with a number. If a parameter name contains **{*}**, which indicates a custom parameter name, replace the character with a string that does not contain periods (.).
- After switching to a new command line, you can use the up and down arrows to browse the records of executed commands.

5.4 Managing Metadata

- Clearing metadata caches

  G42Cloud CLI caches local metadata files obtained during API calling and stores the files in the following directory:

  - Online mode
    - Windows: **C:\Users***{Your Windows username}***.g42\metaRepo**
    - Linux: **/home/***{Current username}*/**.g42/metaRepo/**
    - macOS: **/Users/***{Current username}*/**.g42/metaRepo/**
  - Offline mode
    - Windows: **C:\Users***{Your Windows username}***.g42\metaOrigin**
    - Linux: **/home/***{Current username}*/**.g42/metaOrigin/**
    - macOS: **/Users/***{Current username}*/**.g42/metaOrigin/**

  Run the following commands to clear metadata cache files:

  - Online mode

    g42 meta clear

    ```
    g42 meta clear
    Cache cleared.
    ```

  - Offline mode

    Run the **g42 meta clear** command to clear the metadata cache files parsed from the downloaded offline metadata package. The package will remain. During API calling, this package will be parsed again and new metadata cache files will be written into it. To completely delete the offline metadata package and the contained metadata cache files, delete the **directory** where these files are located, according to the OS you use.

- Downloading metadata

  A downloaded offline metadata package will be stored in the preceding **directories for the offline mode**. Run the following command to download the offline metadata package:

  g42 meta download

  ```
  g42 meta download
  Download successful.
  ```

5.5 Querying the Current Version

To query the version of G42Cloud CLI, run the following command:

g42 version

```
g42 version
Current G42Cloud CLI version: 3.2.8
```

5.6 Upgrading the Version

G42Cloud CLI supports local upgrade. Run the following command to upgrade it to the latest version:

g42 update

```
g42 update
G42Cloud CLI will be upgraded to the latest version. Continue? (y/N): y
Upgrade successful.
```

To skip the confirmation for entering the interactive mode, add **-y** to the command.

```
g42 update -y
Upgrade successful.
```

5.7 Managing Logs

G42Cloud CLI provides log recording and management to cache the logs generated during API calling. Log files are stored in the following directories:

- Windows: **C:\Users***{Your Windows username}***\.g42\log**
- Linux: **/home/***{Current username}***/.g42/log/**
- macOS: **/Users/***{Current username}***/.g42/log/**

Parameters related to log management:

- **level**: log level, which can be **info**, **warning**, or **error**.
- **max-file-size**: maximum size (MB) of a single log file. Range: 1 to 100. Default value: **20**.
- **max-file-num**: the number of retained log files. The value **0** indicates that all log files are retained.
- **retention-period**: the number of days for retaining log files. The value **0** indicates that log files are retained permanently.

If none of the parameters are specified, the default log level is error, the maximum size of a single log file is 20 MB, and the log retention period is 90 days.

Run the following command to set log-related parameters:

g42 log set --key1=value1 --key2=value2 ...

```
g42 log set --level=error --max-file-size=20 --max-file-num=3 --retention-period=30
Configuration successful.
```

Run the following command to view log-related parameters:

g42 log show

```
g42 log show
{
  "maxFileSize": 20,
  "maxFileNum": 3,
  "logLevel": "error",
  "logRetentionPeriod": 30
}
```

5.8 Managing Data in OBS

G42Cloud CLI has integrated the obsutil tool of Object Storage Service (OBS). You can run **g42 obs** to manage your data in OBS.

Function Overview

With G42Cloud CLI, you can perform the following operations to manage your data in OBS.

Table 5-1 Integrated OBS functions in G42Cloud CLI

Function	Description
Basic bucket operations	Create buckets of different storage classes in specific regions, delete buckets, and obtain the bucket list and configuration information.
Basic object operations	Manage objects, including uploading, downloading, deleting, and listing objects. • Upload one or more files or folders. • Upload large files in multiple parts. • Synchronously upload, download, and copy incremental objects. • Copy a single object or copy multiple objects by name prefix. • Move a single object or move multiple objects by name prefix. • Resume failed upload, download, or copy tasks.
Logs	Configure logging on your client to record bucket and object operations for statistical analysis.

Initial Configuration

Before using G42Cloud CLI to manage data in OBS, configure OBS connection, including the OBS endpoint and access key (AK/SK). You can use G42Cloud CLI to manage OBS buckets and objects only after being authenticated by OBS.

- Permanent AK/SK:
  ```
  g42 obs config -i=ak -k=sk -e=endpoint
  ```
- Temporary AK/SK and SecurityToken:
  ```
  g42 obs config -i=ak -k=sk -t=token -e=endpoint
  ```

Checking Connectivity

After the configuration is complete, run the following command to check the connectivity:

```
g42 obs ls -s
```

Check the configuration result based on the command output:

- **Bucket number**: The configuration is correct.
- **Http status [403]**: The access key is incorrect, or you do not have permission to access the bucket list. Analyze the root cause.
- **A connection attempt failed**: OBS cannot be connected. Check the network condition.

Command Line Structure

The commands for managing OBS data with G42Cloud CLI have the following structure:

```
g42 obs command [parameters...] [options...]
```

To enter the interactive mode in Windows:

Step 1 Run the following command to enter the interactive mode:

```
g42 obs
```

Step 2 Manage your OBS data with commands in the following structure:

```
command [parameters...] [options...]
```

For example:

```
g42 obs
Enter "exit" or "quit" to logout
Enter "help" or "help command" to show help docs
Input your command:
-->ls -limit=3 -s
obs://bucket-001
obs://bucket-002
obs://bucket-003
Bucket number: 3
```

----End

NOTE

- **command** indicates the command to be executed, for example, **ls** or **cp**.
- **parameters** indicates the basic parameters (mandatory) of the command, for example, name of a bucket to be created.
- **options** indicates additional parameters (optional) of the command. Additional parameters must be preceded with a hyphen (-) when you run the command. Enter a parameter in the **-***key***=***value* or **-***key value* format, for example, **-acl=private** or **-acl private**. There is no difference between the two formats. Select either one as you like.
- The brackets ([]) are not part of the command. Do not enclose parameter values with brackets when entering a command.
- If the command contains special characters such as ampersands (&), angle brackets (<>), and spaces, escape these parameters using single quotation marks (Linux and macOS) or quotation marks (Windows).

The following table lists the OBS operation commands supported by G42Cloud CLI. The parameters of each command are the same as those in obsutil. For details about these parameters, see **Bucket Commands**, **Object Commands**, and **Auxiliary Commands**.

Table 5-2 OBS operation commands supported by G42Cloud CLI

Category	Command	Function	Description	Command Line Structure
Bucket commands	mb	Create bucket	Create a bucket with a unique name. Each account can create a maximum of 100 buckets.	`g42 obs mb obs://bucket [-fs] [-acl=xxx] [-location=xxx] [-config=xxx]`
	ls	List buckets	Obtain the bucket list. The buckets are displayed in lexicographical order of their names.	`g42 obs ls [-s] [-limit=1] [-format=default] [-config=xxx]`
	stat	Query bucket properties	Query the basic properties of a bucket, including the region, version, support for POSIX, number of objects, storage usage, and bucket quota.	`g42 obs stat obs://bucket [-acl] [-bf=xxx] [-config=xxx]`

Category	Command	Function	Description	Command Line Structure
	chattri	Set bucket properties	Set the properties of a bucket, including access policies.	g42 obs chattri obs://bucket [-acl=xxx] [-aclXml=xxx] [-config=xxx]
	rm	Delete bucket	Delete a bucket that contains no objects, historical versions, or fragments.	g42 obs rm obs://bucket [-f] [-config=xxx]
Object commands	mkdir	Create folder	Create a folder in a specified bucket or the local file system.	• Create a folder in a specified bucket. g42 obs mkdir obs://bucket/folder[/subfolder1/subfolder2] [-config=xxx] • Create a folder in the local file system. g42 obs mkdir folder_url [-config=xxx]
	cp	Upload object	Upload one or more local files or folders to a specified path in OBS. These files can be texts, images, or videos.	• Upload a file. g42 obs cp file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-u] [-vlength] [-vmd5] [-p=1] [-threshold=5248800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx] • Upload a folder. g42 obs cp folder_url obs://bucket[/key] -r [-arcDir=xxx] [-dryRun] [-link] [-f] [-flat] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx] • Upload multiple files and folders. g42 obs cp file1_url,folder1_url|filelist_url obs://bucket[/prefix] -msm=1 [-r] [-arcDir=xxx] [-dryRun] [-link] [-f] [-u] [-vlength] [-vmd5] [-flat] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]

Category	Command	Function	Description	Command Line Structure
		Copy object	Copy a single object or copy multiple objects with a specified name prefix.	• Copy a single object. g42 obs cp obs://srcbucket/key obs://dstbucket/[dest] [-dryRun][-u] [-crr] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx] • Copy multiple objects. g42 obs cp obs://srcbucket[/key] obs://dstbucket[/dest] -r [-dryRun][-f] [-flat] [-u] [-crr] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
		Download object	Download an object or download multiple objects with a specified name prefix to your local PC.	• Download a single object. g42 obs cp obs://bucket/key file_or_folder_url [-tempFileDir=xxx] [-dryRun] [-u] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-versionId=xxx] [-ps=auto] [-cpd=xxx][-fr] [-o=xxx] [-config=xxx] • Download multiple objects. g42 obs cp obs://bucket[/key] folder_url -r [-tempFileDir=xxx] [-dryRun] [-f] [-flat] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
		Resume failed upload task	Resume a failed upload task based on the task ID.	g42 obs cp -recover=xxx [-arcDir=xxx] [-dryRun] [-f] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-clear] [-config=xxx]
		Resume failed copy task	Resume a failed copy task based on the task ID.	g42 obs cp -recover=xxx [-dryRun] [-f] [-u] [-crr] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-clear] [-config=xxx]

Category	Command	Function	Description	Command Line Structure
		Resume failed download task	Resume a failed download task based on the task ID.	g42 obs cp -recover=xxx [-dryRun] [-tempFileDir=xxx] [-f] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-clear] [-config=xxx]
	stat	Query object properties	Query the basic properties of an object.	g42 obs stat obs://bucket/key [-acl][-bf=xxx] [-config=xxx]
	chattri	Set object properties	Set properties of a single object or of multiple objects with a specified name prefix.	• Set properties of a single object. g42 obs chattri obs://bucket/key [-acl=xxx] [-aclXml=xxx] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx] • Set properties of multiple objects. g42 obs chattri obs://bucket[/key] -r [-f] [-v] [-acl=xxx] [-aclXml=xxx] [-o=xxx] [-j=1] [-config=xxx]
	ls	List objects	Query objects or object versions in a bucket. All objects are displayed in lexicographical order of their names or versions.	g42 obs ls obs://bucket[/prefix] [-s] [-d] [-v] [-marker=xxx] [-versionIdMarker=xxx] [-bf=xxx] [-limit=1] [-format=default] [-config=xxx]

Category	Command	Function	Description	Command Line Structure
		List multipart upload tasks	Query multipart upload tasks in a bucket.	g42 obs ls obs://bucket[/prefix] [-s] [-d] -m [-a] [-uploadIdMarker=xxx] [-marker=xxx] [-limit=1] [-format=default] [-config=xxx]
	mv	Move object	Move a single object or move multiple objects with a specified name prefix.	• Move a single object. g42 obs mv obs://srcbucket/key obs://dstbucket/[dest] [-dryRun] [-u] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx] • Move multiple objects. g42 obs mv obs://srcbucket[/key] obs://dstbucket[/dest] -r [-dryRun] [-f] [-flat] [-u] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
	sign	Generate object download link	Generate a download link for a specified object in a bucket or generate download links for multiple objects with a specified name prefix in a bucket.	• Generate a download link for a single object. g42 obs sign obs://bucket/key [-e=300] [-config=xxx] • Generate download links for multiple objects with a specified name prefix. g42 obs sign obs://bucket[/key] -r [-e=300] [-timeRange=time1-time2] [-include=*.xxx] [-exclude=*.xxx] [-o=xxx] [-config=xxx]
	rm	Delete object	• Delete a specified object. • Delete multiple objects with a specified name prefix.	• Delete a single object. g42 obs rm obs://bucket/key [-f] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx] • Delete multiple objects. g42 obs rm obs://bucket/[key] -r [-j=1] [-f] [-v] [-o=xxx] [-config=xxx]

Category	Command	Function	Description	Command Line Structure
	sync	Synchronously upload incremental objects	Synchronize all content in a local source path to a specified target OBS bucket to keep data consistency. Incremental synchronization has the following meanings: 1. Increment: Compare the source files with the target objects and upload only the source files that have changed. 2. Synchronization: After the command is executed, ensure that the local source path is a subset of the target bucket on OBS. That is, any file in the local source path has its corresponding object in the target bucket on OBS.	• Upload a file synchronously. g42 obs sync file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-vlength] [-vmd5] [-p=1] [-threshold=5248800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-cpd=xxx] [-fr] [-config=xxx] • Upload a folder synchronously. g42 obs sync folder_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]

Category	Command	Function	Description	Command Line Structure
		Synchronously copy incremental objects	Synchronize all objects in a specified path of the source bucket to a specified path in the destination bucket to keep data consistency. Incremental synchronization has the following meanings: 1. Increment: Compare the source and target objects and copy only the source objects that have changed. 2. Synchronization: After the command is executed, ensure that the specified path of the source bucket is a subset of the specified path in the target bucket. That is, any object in the specified path of the source bucket has its corresponding object in the target bucket.	g42 obs sync obs://srcbucket[/key] obs://dstbucket[/dest] [-dryRun] [-crr] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]

Category	Command	Function	Description	Command Line Structure
		Synchronously download incremental objects	Synchronize all objects in a specified path of the source OBS bucket to a target local path to keep data consistency. Incremental synchronization has the following meanings: 1. Increment: Compare the source objects with the target files and download only the source objects that have changed. 2. Synchronization: After the command is executed, ensure that the specified path of the source bucket is a subset of the target local path. That is, any object in the specified path of the source bucket has its corresponding file in the target local path.	g42 obs sync obs://bucket[/key] folder_url [-tempFileDir=xxx] [-dryRun] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]

Category	Command	Function	Description	Command Line Structure

	abort	Delete multipart upload task	• Delete a multipart upload task of a specified ID in a specified bucket.
			• Delete multiple multipart upload tasks with a specified name prefix.
				• Delete a single multipart upload task.
				`g42 obs abort obs://bucket/key -u=xxx [-f] [-fr] [-o=xxx] [-config=xxx]`
				• Delete multiple multipart upload tasks.
				`g42 obs abort obs://bucket[/key] -r [-f] [-o=xxx] [-j=1] [-config=xxx]`

Auxiliary commands	config	Update configuration file	You can use this command to update some configurations in the **.obsutilconfig** configuration file. For details about the parameters in **.obsutilconfig**, see **Parameter Description**.	`g42 obs config -interactive [-crr] [-config=xxx]`
	clear	Delete part records	Delete part records from a specified directory.	`g42 obs clear [checkpoint_dir] [-u] [-d] [-c] [-config=xxx]`

Category	Command	Function	Description	Command Line Structure
	help	View command help information	View the supported OBS commands or the help information of a specific command.	g42 obs help [command]
	version	View version	View the version of the integrated obsutil.	g42 obs version
	archive	Archive log files	Archive log files to a local PC or to a specified bucket.	• Archive to a local PC. g42 obs archive [file_or_folder_url] [-config=xxx] • Archive to a specified bucket. g42 obs archive obs://bucket[/key] [-config=xxx]
	ls	List failure result files	List the last modified failure result files generated by the **cp** or **sync** commands in a specified folder.	g42 obs ls -failed [-limit=1000] [-o=xxx]

6 Options

6.1 Option Overview

G42Cloud CLI options are system parameters that can be added to commands for calling APIs. The following table describes the options supported by G42Cloud CLI. Any option except **help**, **debug**, **dryrun**, **cli-username**, **cli-password**, **cli-domain**, **cli-output**, **cli-query**, **cli-output-num**, **cli-jsonInput**, and **cli-endpoint** can be set

in profiles. When a command is executed, the value of a parameter in the command takes precedence over that in the profile.

Table 6-1 G42Cloud CLI options

Option	Description	Example
help	Prints help information.	g42 RDS ListCollations --cli-region="ae-ad-1" --help
debug	Prints debugging information during command invocation. For example, the steps of API calling and the complete request URL.	g42 VPC ShowVpc/v3 --cli-region="ae-ad-1" --project_id="0dd8cb**********19b5a84546" --vpc_id="0bbe****-****-****-****-****235be6e7" --debug
dryrun	Checks a command. The system prints the request message after verification. It does not call the target API.	g42 RDS CreateConfiguration --cli-region="ae-ad-1" --project_id="4ff018c**********df31948" --datastore.type="MySQL" --datastore.version="5.7" --values.max_connections="10" --name="test-001" --description="test create configuration" --dryrun
cli-region	Region where the cloud service resources to be managed are located.	g42 EVS DeleteVolume --cli-region="ae-ad-1" --volume_id="aed9****-****-****-****-****0e3219cf" --project_id="0dd8cb**********19b5a84546"
cli-username, cli-password, cli-domain	• **cli-username**: IAM user name • **cli-password**: IAM user password • **cli-domain**: the name of the account to which the IAM user belongs These options are used to call cloud service APIs using an account in non-configuration mode.	g42 EVS UpdateVolume --cli-region="ae-ad-1" --volume_id="aed9****-****-****-****-****0e3219cf" --project_id="0dd8cb**********19b5a84546" --volume.name="ecs-abcd" --volume.description="volume test" --cli-username=s******1 --cli-password=****** --cli-domain=s******1

Option	Description	Example

cli-access-key, cli-secret-key, cli-security-token	• **cli-access-key**: access key ID (AK). This parameter must be used together with SK.
	• **cli-secret-key**: secret access key (SK). This parameter must be used together with AK.
	• **cli-security-token**: temporary security credential. This parameter is required when you use a temporary AK/SK for identity authentication.
	These options are used to call cloud service APIs using an AK/SK in non-configuration mode.
		Call cloud service APIs using an AK/SK in non-configuration mode.
		• Use an access key (permanent AK/SK):
		`g42 RDS ListApiVersion --cli-region="ae-ad-1" --cli-access-key=8NVT**********KIOV --cli-secret-key=VHMQjoC**********lsk3cGf`
		• Use temporary security credentials (temporary AK/SK and SecurityToken):
		`g42 RDS ListApiVersion --cli-region="ae-ad-1" --cli-access-key=5FSU**********607T --cli-secret-key=VoyjgLh**********qRc8pSq --cli-security-token=******`

cli-x-auth-token	The access token issued to an IAM user. This option is used to call cloud service APIs using a token in non-configuration mode.	`g42 ECS ListServersDetails --cli-region="ae-ad-1" --project_id="2cc60f5**********efa5019ef" --enterprise_project_id="441d5677-****-****-****-ef7fd6336666" --cli-x-auth-token=******`

Option	Description	Example
cli-agency-domain-id/cli-agency-domain-name, cli-agency-name, cli-source-profile	• **cli-agency-domain-name**: Account name of a delegating party. This parameter must be used together with **cli-agency-name**. **cli-agency-domain-id**: Account ID of the delegating party. This parameter must be used together with **cli-agency-name**. • **cli-agency-name**: Agency name. It must be used together with **cli-agency-domain-id** or **cli-agency-domain-name**. • **cli-source-profile**: The profile that stores the authentication information of the delegated party. The value of this option cannot be the current profile. These options are used to call cloud service APIs using an agency in non-configuration mode.	g42 VPC ListAddressGroup/v3 --cli-region="ae-ad-1" --project_id="2cc60**********caefa5019ef" --cli-agency-domain-id=13534326**********5cf67b --cli-agency-name=****** --cli-source-profile=test
cli-domain-id	ID of the account to which the IAM user belongs. This option is required when global service APIs are called in AK/SK authentication mode. In this case, G42Cloud CLI automatically obtains this parameter from your authentication information.	g42 CDN ListDomains --cli-region="ae-ad-1" --cli-domain-id="08e09a6e**********1bb800"

Option	Description	Example
cli-profile	Name of a G42Cloud CLI profile. A profile stores a group of common information (such as the AK/SK, region, and project ID) required for calling cloud service APIs.	g42 EVS ListSnapshots --cli-profile=test
cli-mode	Authentication mode of a profile. Options: • AKSK • token • ecsAgency • agency If a profile contains different **authentication parameters** (such as AK/SK and token), the mode specified by **cli-mode** is used. Use **--cli-mode** in a command to specify the authentication mode for a profile.	• Set **cli-mode** to **AKSK**. g42 CCE ListNodes --cluster_id="f288****-****-****-****-****ac101534" --project_id="0dd8cb**********19b5a84546" --cli-profile=test --cli-mode=AKSK • Set **cli-mode** to **token**. g42 CCE ListNodes --cluster_id="f288****-****-****-****-****ac101534" --project_id="0dd8cb**********19b5a84546" --cli-profile=test --cli-mode=token
cli-output, cli-query, cli-output-num	Output format. • cli-output Output format of the response data. The options include: – json – table – tsv • cli-query JMESPath for filtering response data. • cli-output-num Indicates whether to print the row numbers during table output. The value can be **true** or **false**.	• When the value of **cli-output** is **json**: – Call a cloud service API g42 CCE ListClusters --cli-region="ae-ad-1" --type="VirtualMachine" --project_id="0dd8cb**********19b5a84546" --cli-query="items[0]" – Call a CLI system command g42 configure list --cli-output=json --cli-query="profiles[].{Name:name,Mode:mode,Ak:accessKeyId,SK:secretAccessKey}" • When the value of **cli-output** is **table**: g42 configure list --cli-output=table --cli-query="profiles[].{Name:name,Mode:mode,Ak:accessKeyId,SK:secretAccessKey}" • When the value of **cli-output** is **tsv**: g42 configure list --cli-output=tsv --cli-query="profiles[].{Name:name,Mode:mode,Ak:accessKeyId,SK:secretAccessKey}"

Option	Description	Example
cli-jsonInput	Specifies a JSON file to pass API parameters. If a cloud service API has too many parameters, you can define the parameters in a JSON file. G42Cloud CLI then parses the parameters in the file.	g42 ECS CreateServers --cli-region="ae-ad-1" --cli-read-timeout=60 --cli-jsonInput=C:\cli\Ecs_CreateServers.json
cli-connect-timeout and cli-read-timeout	Request timeouts. • **cli-connect-timeout**: request connection timeout, in seconds. The default value is **5**, and the minimum value is **1**. • **cli-read-timeout**: I/O timeout, in seconds. The default value is **10**, and the minimum value is **1**.	g42 ECS DeleteServerPassword --cli-region="ae-ad-1" --project_id="2cc60f5***************efa5019ef" --server_id="e6b99563-****-****-****-1820d4fd2a67" --cli-connect-timeout=10 --cli-read-timeout=15
cli-retry-count	Number of connection attempts. The system retries automatically if the connection times out. The value ranges from 0 to 5, and the default value is **0**.	g42 RDS ListInstances --cli-region="ae-ad-1" --Content-Type="application/json" --project_id="2cc60***************caefa5019ef" --cli-retry-count=3
cli-skip-secure-verify	Specifies whether to skip HTTPS certificate verification (**not recommended**). The value can be **true** or **false**. The default value is **false**. Skipping certificate verification has security risks. If you set this parameter to **true**, G42Cloud CLI will display a message asking for your confirmation.	g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="2cc6***************6caefa5019ef" --cli-skip-secure-verify=true

Option	Description	Example
cli-endpoint	Custom domain name. By default, requests are sent to the target cloud service in the relevant region. You can also specify an endpoint of this cloud service.	g42 IoTDA UpdateDevice --cli-region="ae-ad-1" --description="test update device" --device_id="testz**********0802" --cli-endpoint="iot-mqtts.ae-ad-1.g42cloud.com"

6.2 Printing Help Information

View the command help information. For example, view the help information of an RDS API whose operation is **ListCollations**:

```
g42 RDS ListCollations --cli-region="ae-ad-1" --help
```

6.3 Printing Debugging Information

Print debugging information during command execution. To do so, add **--debug** in the command:

```
g42 VPC ShowVpc/v3 --cli-region="ae-ad-1" --project_id="0dd8cb**********19b5a84546" --
vpc_id="0bbe****-****-****-****-****235be6e7" --debug
[debug info] 2022/06/21 19:59:25 Read and connection timeouts are 40s and 30s respectively.
[debug info] 2022/06/21 19:59:25 URL: https://vpc.ae-ad-1.g42cloud.com/
v3/0dd8cb**********19b5a84546/vpc/vpcs/0bbe****-****-****-****-****235be6e7
[debug info] 2022/06/21 19:59:26 API response status code is 200.
[debug info] 2022/06/21 19:59:26 API response X-Request-Id is f9fd68**********2e48ec7f88.
{
  "vpc": {
    "id": "0bbe****-****-****-****-****235be6e7",
    "name": "CCI-VPC-*******",
    "description": "",
    "cidr": "192.***.*.*/**",
    "extend_cidrs": [],
    "status": "ACTIVE",
    "project_id": "0dd8cb**********19b5a84546",
    "enterprise_project_id": "0",
    "tags": [],
    "created_at": "2022-05-10T02:53:42Z",
    "updated_at": "2022-05-10T02:53:43Z",
    "cloud_resources": [
      {
        "resource_type": "routetable",
        "resource_count": 1
      },
      {
        "resource_type": "virsubnet",
        "resource_count": 1
      }
    ]
  },
  "request_id": "f9fd68**********2e48ec7f88"
}
```

6.4 Checking Command

The **dryrun** option is used to check the correctness of a command. Add **--dryrun** in a command. The system prints the request message after verification. It does not call the target API.

```
g42 RDS CreateConfiguration --cli-region="ae-ad-1" --project_id="0dd8cb***************19b5a84546" --
datastore.type="MySQL" --datastore.version="5.7" --values.max_connections="10" --name="test-001" --
description="test create configuration" --dryrun
-------- The execution is eliminated in dry-run mode. Current request: --------
POST https://rds.ae-ad-1.g42cloud.com/v3/0dd8cb***************19b5a84546/configurations
Content-Type: application/json
X-Project-Id: 0dd8cb***************19b5a84546
X-Sdk-Date: 20220621T103331Z
Authorization: ****

{
  "datastore": {
    "type": "MySQL",
    "version": "5.7"
  },
  "description": "test create configuration",
  "name": "test-001",
  "values": {
    "max_connections": "10"
  }
}
```

6.5 Specifying Region

In addition to reading region information from the profile, G42Cloud CLI can also read the value of **cli-region** you enter in a command, as shown below:

```
g42 EVS DeleteVolume --cli-region="ae-ad-1" --volume_id="aed9****-****-****-****-****0e3219cf" --
project_id="0dd8cb***************19b5a84546"
{
    "job_id": "70a5****-****-****-****-****441e862b"
}
```

NOTE

Projects vary according to regions. Therefore, you need to specify the project ID when specifying a region.

6.6 Calling APIs with an Account in Non-configuration Mode

Add the **--cli-username**, **--cli-password**, and **--cli-domain** options in a command to call a cloud service API with an account **in non-configuration mode**.

```
g42 EVS UpdateVolume --cli-region="ae-ad-1" --volume_id="aed9****-****-****-****-****0e3219cf" --
project_id="0dd8cb***************19b5a84546" --volume.name="ecs-abcd" --volume.description="volume
test"  --cli-username=s******1 --cli-password=****** --cli-domain=s******1
{
  "id": "aed9****-****-****-****-****0e3219cf",
  "links": [
    {
      "href": "https://evs.ae-ad-1.g42cloud.com/v2/0dd8cb***************19b5a84546/volumes/aed9****-****-****-
****-****0e3219cf",
```

```
    "rel": "self"
   },
   {
    "href": "https://evs.ae-ad-1.g42cloud.com/0dd8cb***************19b5a84546/volumes/aed9****-****-****-****-****0e3219cf",
    "rel": "bookmark"
   }
  ],
  "name": "ecs-abcd",
  "status": "in-use",
  "attachments": [
   {
    "server_id": "4f06****-****-****-****-****04dd856a",
    "attachment_id": "773d****-****-****-****-****e4f3b6f0",
    "attached_at": "2022-05-11T02:27:44.453029",
    "host_name": null,
    "volume_id": "aed9****-****-****-****-****0e3219cf",
    "device": "/dev/vda",
    "id": "aed9****-****-****-****-****0e3219cf"
   }
  ],
  "description": "volume test",
  "size": 40,
  "metadata": {
   "billing": "1",
   "readonly": "False",
   "attached_mode": "rw"
  },
  "bootable": "true",
  "availability_zone": "ae-ad-1a",
  "os-vol-host-attr:host": null,
  "source_volid": null,
  "snapshot_id": null,
  "created_at": "2022-05-11T02:27:09.279069",
  "volume_type": "SAS",
  "shareable": "false",
  "multiattach": false,
  "os-vol-tenant-attr:tenant_id": null,
  "os-volume-replication:extended_status": null,
  "volume_image_metadata": null
}
```


NOTE

To call a cloud service API in this mode, **use a custom parameter** to pass the username and password in the command. This ensures account security and prevents peeping when you input your password.

6.7 Calling APIs with AK/SK in Non-configuration Mode

Add the **--cli-access-key**, **--cli-secret-key**, and **--cli-security-token** options in a command to call a cloud service API with an AK/SK **in non-configuration mode**.

By default, if only **--cli-access-key** and **--cli-secret-key** are used in a command, the AK/SK are taken as permanent ones.


```
g42 RDS ListApiVersion --cli-region="ae-ad-1" --cli-access-key=8NVT**********KIOV --cli-secret-key=VHMQjoC*************************lsk3cGf
{
  "versions": [
   {
    "id": "v3",
    "links": [],
    "status": "CURRENT",
    "updated": "2019-01-15T12:00:00Z"
   },
```

```
    {
      "id": "v1",
      "links": [],
      "status": "DEPRECATED",
      "updated": "2017-02-07T17:34:02Z"
    }
  ]
}
```

By default, if **--cli-access-key**, **--cli-secret-key**, and **--cli-security-token** are used in a command, the AK/SK are taken as temporary ones.

```
g42 RDS ListApiVersion --cli-region="ae-ad-1" --cli-access-key=5FSU**********607T --cli-secret-key=VoyjgLh**********qRc8pSq --cli-security-token=******
{
  "versions": [
    {
      "id": "v3",
      "links": [],
      "status": "CURRENT",
      "updated": "2019-01-15T12:00:00Z"
    },
    {
      "id": "v1",
      "links": [],
      "status": "DEPRECATED",
      "updated": "2017-02-07T17:34:02Z"
    }
  ]
}
```

6.8 Calling APIs with a Token in Non-configuration Mode

Add **--cli-x-auth-token** in a command to call a cloud service API with a token **in non-configuration mode**. When you use a command prompt tool (such as **cmd.exe**) to run the following command, the command line may be incomplete if the value of **cli-x-auth-token** is too long. Therefore, check whether the command line content is complete before executing a command.

```
g42 ECS ListServersDetails --cli-region="ae-ad-1" --project_id="2cc60f5**********efa5019ef" --enterprise_project_id="441d5677-****-****-****-ef7fd6336666" --cli-x-auth-token=******
{
  "count": 0,
  "servers": []
}
```

6.9 Calling APIs Using an Agency in Non-configuration Mode

After a delegating party creates an agency to delegate another account to manage its resources, the delegated party can add **--cli-agency-domain-id**/**--cli-agency-domain-name**, **--cli-agency-name**, and **--cli-source-profile** to a command to call cloud service APIs **using an agency in non-configuration mode**, and manage resources of the delegating party.

```
g42 VPC ListAddressGroup/v3 --cli-region="ae-ad-1" --project_id="2cc60**********caefa5019ef" --cli-agency-domain-id=13534326**********5cf67b --cli-agency-name=****** --cli-source-profile=test
{
  "request_id": "29ec21**********6d6b4cdd82",
```

```
  "address_groups": [],
  "page_info": {
  "current_count": 0
   }
}
```

NOTE

Among the preceding parameters, use **--cli-agency-domain-id**/**--cli-agency-domain-name** and **--cli-agency-name** at the same time. Use **--cli-source-profile** to specify the profile that stores the authentication information of the delegated party. The value of **--cli-source-profile** cannot be the current profile.

6.10 Specifying Account ID

When calling APIs of global services in AK/SK mode, an account ID (**cli-domain-id**) is required. During API calling, G42Cloud CLI automatically obtains the account ID based on the user authentication information. You can also add the **--cli-domain-id** option in the command, as shown below:

```
g42 CDN ListDomains --cli-region="ae-ad-1" --cli-domain-id="08e09a6e*****************1bb800"
{
  "total": 0,
  "domains": null
}
```

6.11 Specifying Profile

G42Cloud CLI supports multiple profiles. You can save common information (such as the AK/SK and region) in a profile and use the information by specifying the profile name through **--cli-profile**. For example:

```
g42 EVS ListSnapshots --cli-profile=test
```

6.12 Specifying Authentication Mode

G42Cloud CLI allows you to set the authentication mode to **AKSK**, **token**, **ecsAgency**, or **agency** in a profile. **AKSK** is recommended. If the profile you use contains **parameters related to different authentication modes** such as AK/SK and token, use **--cli-mode** to specify the mode that will be used.

```
g42 CCE ListNodes --cluster_id="f288****-****-****-****-****ac101534" --
project_id="0dd8cb***************19b5a84546" --cli-profile=test --cli-mode=AKSK
```

NOTE

When configuring a profile, use **--cli-profile** to specify the profile name and add authentication parameters corresponding to **--cli-mode**.

- If **--cli-mode** is **AKSK**, set both **--cli-access-key** and **--cli-secret-key**.
- If **--cli-mode** is **token**, set **--cli-x-auth-token**.
- If **--cli-mode** is **ecsAgency**, set **--cli-mode=ecsAgency**.
- If **--cli-mode** is **agency**, set **--cli-agency-domain-id**/**--cli-agency-domain-name**, **--cli-agency-name**, and **--cli-source-profile**.

6.13 Specifying Output Format

Use the **--cli-query** option in a command to pass **a JMESPath expression** so that you can extract key information from the return result. Use **--cli-output** to specify the output format of the response data and use **--cli-output-num** to specify whether to print the row numbers during table output.

Output Order

The sequence of parameters in the output result varies according to the JMESPath expression specified by **--cli-query**. The output result queried by some expressions does not contain the attribute names (parameter keys) of the output data. If you use these expressions, understand the sequence of the output data to facilitate data processing. The following table describes the output sequence of different types of JMESPath expressions.

Table 6-2 Data output sequence of different JMESPath expressions

JMESPath Expression Type	Example JMESPath Expression	Attribute Names Contained in JSON or Table Output	Attribute Names Contained in TSV Output	Data Output Sequence	Example Output
Object-level expression	--cli-query="items[0]"	Yes	No	In alphabetical order of the attribute names of the object	**Example 1**
Attribute-level expression without changing the attribute names	--cli-query="items[0].items[0].[spec.flavor,metadata.uid]"	No	No	In the attribute name order specified in the JMESPath expression	**Example 2**
Attribute-level expression with the attribute names changed	--cli-query="items[0].{Flavor:spec.flavor,ClusterID:metadata.uid}"	Yes	No	In alphabetical order of the new attribute names	**Example 3**

The following examples show the sequence of data output in JSON format through different JMESPath expressions:

- Example 1

When an object is specified, G42Cloud CLI outputs the attribute values in alphabetical order of the attribute names of the object. In this example, the object **items[0]** is specified. The attributes of the object are sorted in alphabetical order as follows: **apiVersion**, **kind**, **metadata**, **spec**, **status**. The output result is as follows:

```
g42 CCE ListClusters --cli-region="ae-ad-1" --type="VirtualMachine" --
project_id="0dd8cb***************19b5a84546" --cli-query="items[0]"
{
  "apiVersion": "v3",
  "kind": "Cluster",
  "metadata": {
    "creationTimestamp": "2022-05-13 08:51:58.252509 +0000 UTC",
    "labels": {
      "FeatureGates": "elbv3,"
    },
    "name": "github-*****-****",
    "uid": "f288****-****-****-****-****ac101534",
    "updateTimestamp": "2022-05-13 09:10:06.395875 +0000 UTC"
  },
  "spec": {
    "authentication": {
      "authenticatingProxy": {},
      "mode": "rbac"
    },
    "az": "multi_az",
    "billingMode": 0,
    "category": "CCE",
    "containerNetwork": {
      "cidr": "10.*.*.*/**",
      "mode": "vpc-router"
    },
    "eniNetwork": {},
    "extendParam": {
      "alpha.cce/fixPoolMask": "25",
      "kubernetes.io/cpuManagerPolicy": "",
      "upgradefrom": ""
    },
    "flavor": "cce.s2.small",
    "hostNetwork": {
      "SecurityGroup": "653e****-****-****-****-****6a23eb7e",
      "subnet": "d5df****-****-****-****-****4955c724",
      "vpc": "c865****-****-****-****-****efe7e8d8"
    },
    "kubeProxyMode": "iptables",
    "kubernetesSvcIpRange": "10.***.*.*/**",
    "masters": [
      {
        "availabilityZone": "ae-ad-1b"
      },
      {
        "availabilityZone": "ae-ad-1a"
      }
    ],
    "supportIstio": true,
    "type": "VirtualMachine",
    "version": "v1.19.10-r0"
  },
  "status": {
    "endpoints": [
      {
        "type": "Internal",
        "url": "https://192.***.*.***:5443"
      },
      {
        "type": "External",
        "url": "https://121.**.**.***:5443"
      }
    ],
```

```
    "phase": "Available"
  }
}
```

- Example 2

  When specific attributes of an object are specified without changing the attribute names, G42Cloud CLI outputs the attribute values in the order that the attribute names are specified. In this example, the expression **items[0].[spec.flavor,metadata.uid]** specifies the child attribute **flavor** of **spec** and **uid** of **metadata** under the object **items[0]**. In the output result, the value of **spec.flavor** is displayed prior to that of **metadata.uid**.

```
g42 CCE ListClusters --cli-region="ae-ad-1" --type="VirtualMachine" --
project_id="0dd8cb****************19b5a84546" --cli-query="items[0].[spec.flavor,metadata.uid]"
[
  "cce.s2.small",
  "f288****-****-****-****-****ac101534"
]
```

- Example 3

  When specific attributes of an object are specified with the attribute names changed, G42Cloud CLI outputs the attribute values in alphabetical order of the new attribute names. In this example, the child attributes of two attributes under the object **items[0]** are specified. The expression **items[0].{Flavor:spec.flavor,ClusterID:metadata.uid}** renames the **flavor** attribute of **spec Flavor** and renames the **uid** attribute of **metadata ClusterID**. The new attributes are sorted in alphabetical order as follows: **ClusterID**, **Flavor**. Therefore, the output result is as follows:

```
g42 CCE ListClusters --cli-region="ae-ad-1" --type="VirtualMachine" --
project_id="0dd8cb****************19b5a84546" --cli-query="items[0].
{Flavor:spec.flavor,ClusterID:metadata.uid}"
{
  "ClusterID": "f288****-****-****-****-****ac101534",
  "Flavor": "cce.s2.small"
}
```

Output Format

Use **--cli-output** to specify the output format. The value can be **json**, **table**, or **tsv**.

- When the value of **--cli-output** is **json**:

  The result is output in JSON format as follows:

```
g42 configure list --cli-output=json --cli-query="profiles[].
{Name:name,Mode:mode,Ak:accessKeyId,SK:secretAccessKey}"
[
  {
    "Ak": "8NV****IOV",
    "Mode": "AKSK",
    "Name": "test",
    "SK": "****"
  },
  {
    "Ak": "H9N****MXW",
    "Mode": "AKSK",
    "Name": "default",
    "SK": "****"
  }
]
```

- When the value of **--cli-output** is **table**:

  The result is output in a table as follows:

```
C:\cli>g42 configure list --cli-output=table --cli-query="profiles[].{Name:name,Mode:mode,
AK:accessKeyId,SK:secretAccessKey}"
+---+----------+----+-------+----+
|No.|    AK    |Mode| Name  | SK |
+---+----------+----+-------+----+
| 1 |H9N****MXW|AKSK|default|****|
+---+----------+----+-------+----+
| 2 |8NV****IOV|AKSK| test  |****|
+---+----------+----+-------+----+

C:\cli>
```

In the meantime, you can use **--cli-output-num** to specify whether to print the row numbers.

```
C:\cli>g42 configure list --cli-output=table --cli-query="profiles[].{Name:name,Mode:mode,
AK:accessKeyId,SK:secretAccessKey}" --cli-output-num=false
+----------+----+-------+----+
|    AK    |Mode| Name  | SK |
+----------+----+-------+----+
|H9N****MXW|AKSK|default|****|
+----------+----+-------+----+
|8NV****IOV|AKSK| test  |****|
+----------+----+-------+----+

C:\cli>
```

- When the value of **--cli-output** is **tsv**:

  The result is output in TSV format as follows:

```
C:\cli>g42 configure list --cli-output=tsv --cli-query="profiles[].{Name:name,Mode:mode,
AK:accessKeyId,SK:secretAccessKey}"
H9N****MXW      AKSK    default ****
8NV****IOV      AKSK    test    ****

C:\cli>
```

  In TSV format output, data values are separated by tab characters and line breaks, and no extra symbols are included. You can use the output result for other commands. The TSV output result does not contain table headers. To prevent confusion when using the data for other commands, you need to understand the data output sequence of different types of JMESPath expressions. For details, see **Output Order**.

  If multiple attributes are specified but not renamed in a JMESPath expression for TSV format output, attributes enclosed in square brackets ([]) will be displayed in new lines. The following is an example:

```
C:\cli>g42 configure list --cli-output=tsv --cli-query="profiles[].[[name],mode,accessKeyId,secretAccessKey]"
AKSK    H9N****MXW      ****
default
AKSK    8NV****IOV      ****
test

C:\cli>
```

  In this example, the attributes **name**, **mode**, **accessKeyId**, and **secretAccessKey** are specified. Attribute **name** is output in a new line. The attributes are output in the order that they appear in the command. The first line displays the values of **mode**, **accessKeyId**, and **secretAccessKey**, and the second line displays the value of **name**.

NOTE

For details about other precautions for using **--cli-query**, **--cli-output**, and **--cli-output-num**, see **FAQs**.

6.14 Passing API Parameters with JSON File

When calling a cloud service API that has too many parameters, use **--cli-jsonInput** to input some or all parameters of the API through a JSON file rather than directly entering the parameters in the command. Enter other parameters, such as **G42Cloud CLI system parameters** and API parameters that are not included in the JSON file, in the command. Example:

```
g42 ECS CreateServers --cli-region="ae-ad-1" --cli-read-timeout=60 --cli-jsonInput=C:\cli
\Ecs_CreateServers.json
{
   "job_id": "ff808082**********************ae0646",
   "serverIds": [
      "dd86****-****-****-****-****91527651"
   ]
}
```

To pass API parameters through **--cli-jsonInput**, compile a JSON file and place each API parameter in the corresponding key based on its location in the request.

To construct a JSON file, perform the following steps:

Step 1 Add **--help** to the end of the original command and run the command. Then view the location of each API parameter in **Params** of the command output.

Step 2 Create a JSON file and name it in the format *${Service name}_${API name}*.**json**. Add the following content to the file:

```
{
   "header": {},
   "path": {},
   "query": {},
   "formData": {},
   "cookie": {},
   "body": {}
}
```

Step 3 Set parameters based on the key order in the JSON file.

- For **non-body** parameters, add **parameter-value** pairs in the relevant curly brackets ({}) and separate them with commas (,). Do not add any commas between the last pair and the right bracket (}).
- For **body** parameters, set them in **API Explorer** and then click **Edit Code**. Copy the JSON format parameters to the **body** key in the JSON file while **keeping only one pair of brackets**.

Step 4 If there is no parameter next to a key, **delete the entire row where the key locates**. When deleting the last key, delete the comma (,) between the outermost and previous right brackets (}).

Step 5 In your G42Cloud CLI command, use **--cli-jsonInput=***${JSON file location}* to pass API parameters, and run the command.

----End

For more precautions, see **FAQs**.

In the preceding example, the **Ecs_CreateServers.json** file passed by **--cli-jsonInput=C:\cli\Ecs_CreateServers.json** is as follows:

```
{
   "path": {
```

```
    "project_id": "0dd8cb41******************a84546"
  },
  "body": {
    "server": {
      "adminPass": "wh**********",
      "auto_terminate_time": "2022-01-19T03:30:52Z",
      "availability_zone": "ae-ad-1a",
      "data_volumes": [
        {
          "multiattach": true,
          "shareable": true,
          "size": 100,
          "volumetype": "SATA"
        }
      ],
      "flavorRef": "2d53****-****-****-****-****257bf163",
      "imageRef": "7059****-****-****-****-****0b5e9e4c",
      "name": "ecs_server_01",
      "nics": [
        {
          "ipv6_enable": true,
          "subnet_id": "4eb2****-****-****-****-****ff9a042d"
        }
      ],
      "publicip": {
        "eip": {
          "bandwidth": {
            "sharetype": "PER",
            "size": 30
          },
          "iptype": "5_sbgp"
        }
      },
      "root_volume": {
        "volumetype": "SATA"
      },
      "server_tags": [
        {
          "key": "date",
          "value": "211102"
        }
      ],
      "vpcid": "5aa5****-****-****-****-****1df05a3a"
    }
  }
}
```


6.15 Specifying Request Timeouts

The **cli-connect-timeout** and **cli-read-timeout** options are used to set request timeouts. For the request connection timeout (**--cli-connect-timeout**), the default value is 5s. For the I/O timeout (**--cli-read-timeout**), the default value is 10s.

```
g42 ECS DeleteServerPassword --cli-region="ae-ad-1" --project_id="2cc60f5***************efa5019ef" --
server_id="e6b99563-****-****-****-1820d4fd2a67" --cli-connect-timeout=10 --cli-read-timeout=15
```

You can use either **--cli-connect-timeout** or **--cli-read-timeout** or use both options in a command.

6.16 Specifying Retry Count

The **cli-retry-count** option specifies the number of connection attempts. The system retries automatically if a request times out due to a network connection problem. The value range of **--cli-retry-count** is 0 to 5, and the default value is **0**.

- If a request fails due to abnormal network connection, the following information is displayed on G42Cloud CLI:

```
g42 RDS ListInstances --cli-region="ae-ad-1" --Content-Type="application/json" --
project_id="2cc60***************caefa5019ef" --cli-retry-count=3
[NETWORK_ERROR] Connection timed out 4 consecutive times (reconnection attempts: 3). Check
network connectivity.
```

- If the network connection is normal, the following information is displayed:

```
g42 RDS ListInstances --cli-region="ae-ad-1" --Content-Type="application/json" --
project_id="2cc60***************caefa5019ef" --cli-retry-count=3
{
  "instances": [],
  "total_count": 0
}
```

CAUTION

Setting the retry count with **--cli-retry-count** may cause idempotence and cyclic API calls. Exercise caution when using this option for resource creation APIs.

6.17 Skipping HTTPS Request Certificate Verification

The **cli-skip-secure-verify** option specifies whether to skip HTTPS certificate verification (**skipping is not recommended**). When you use G42Cloud CLI to call a cloud service API after **configuring the HTTP proxy**, the error x509 may occur due to a certificate verification failure. To prevent this from happening, add **--cli-skip-secure-verify=true** to a command and then run the command. During the command execution, you will be prompted to confirm whether to skip HTTPS certificate verification.

```
g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="2cc6***************6caefa5019ef" --cli-skip-
secure-verify=true
Using `--cli-skip-secure-verify=true` will expose your private data to the public network and cause
interception risks. Continue? (y/N): y
{
  "servers": []
}
```

NOTE

However, using **--cli-skip-secure-verify=true** to skip HTTPS certificate verification will cause your private data to be exposed to the public network and a risk of theft. Therefore, you are **not advised** to use this option. To resolve this problem, **you are advised to** import the certificate issued by your company to the trusted CA certificate of the OS.

6.18 Specifying a Domain Name

Use **cli-endpoint** to specify a domain name of the target cloud service. By default, G42Cloud CLI sends requests to the target cloud service in the relevant region. You can also specify an endpoint of this cloud service.

```
g42 IoTDA UpdateDevice --cli-region="ae-ad-1" --description="test update device" --
device_id="testz********************0802" --cli-endpoint="iot-mqtts.ae-ad-1.g42cloud.com"
{
   "app_id" : "103b***********************6202",
   "app_name" : "DefaultApp_********",
   "device_id" : "testz********************0802",
```

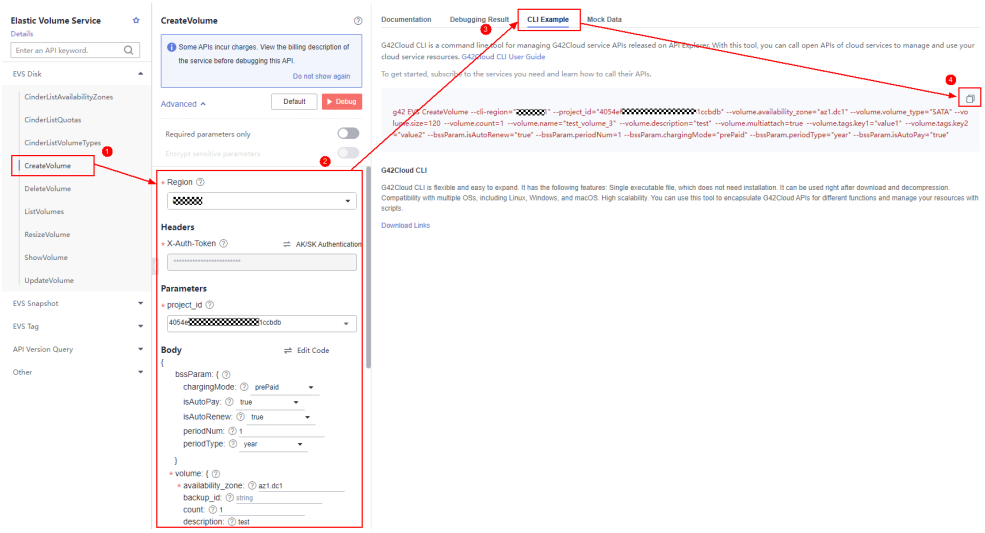
```
  "node_id" : "testz******************0802",
  "gateway_id" : "testz******************0802",
  "device_name" : "test**********************************************0802",
  "node_type" : "CATEWAY",
  "description" : "test update device",
  "fw_version" : null,
  "sw_version" : null,
  "device_sdk_version" : null,
  "auth_info" : {
     "auth_type" : "CERTIFICATES"
     "secret" : null,
     "fingerprint" : "e30db21**********************b1772929c",
     "secure_access" : true,
     "timeout" : 0
  }
  "product_id" : "50c1**********************588e",
  "product_name" : "50c1**********************588e",
  "status" : "INACTIVE",
  "create time" : "20220801T085024z",
  "tags" : [],
  "extension_info" : {
     "sha1_fingerprint" : "82cd23e2**********************e089d59d",
     "sha256_fingerprint" : "396ldb****************************************************4a5f88ccb7"
  }
}
```

7 Obtaining CLI Examples on API Explorer

API Explorer supports online API debugging and generates G42Cloud CLI examples for cloud service APIs. After configuring the parameters of an API on API Explorer, you can copy and use the relevant **CLI Example**.

The following figure shows the EVS API for creating an EVS disk on API Explorer. After you set the request parameters, the CLI example will be updated accordingly. You can save the command with the set request parameters. Later, you can directly invoke this command in G42Cloud CLI or integrate it into a script.

Figure 7-1 Obtaining a CLI example on API Explorer



NOTE

- On the **API Explorer** console, only the parameters with values specified are displayed in the CLI example.
- A CLI example carries information such as a project ID and region. To use the example in another project or region, replace the project ID and region with the corresponding values.

8 Using G42Cloud CLI in Non-configuration Mode

8.1 Introduction

In G42Cloud CLI, you can call cloud service APIs using a profile or do so in non-configuration mode. When using G42Cloud CLI in non-configuration mode, you do not need to pass your authentication information through a profile. Instead, directly pass your authentication parameters in commands without adding any profiles. You can call cloud service APIs using any of the following authentication modes:

- **AK/SK Authentication**
- **Account Authentication**
- **Token Authentication**
- **ecsAgency Authentication**
- **Agency Authentication**

When using G42Cloud CLI in non-configuration mode, note the **precautions** and understand the **priority of each authentication mode**.

8.2 AK/SK Authentication

- Access key (permanent AK/SK)

Enter a permanent AK (**cli-access-key**) and SK (**cli-secret-key**) in a command to call a cloud service API.

```
g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="4ff018c3**********f31948" --cli-access-key=8NVT**********KIOV --cli-secret-key=VHMQjoC**********lsk3cGf
```

- Temporary security credentials (temporary AK/SK and SecurityToken)

  Enter a temporary AK (**cli-access-key**), SK (**cli-secret-key**), and SecurityToken (**cli-security-token**) in a command to call a cloud service API.

```
g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="4ff018c3**********f31948" --cli-access-key=5FSU**********607T --cli-secret-key=VoyjgLh**********qRc8pSq --cli-security-token=******
```

8.3 Account Authentication

Enter an IAM user name (**cli-username**), password (**cli-password**), and **account name** (**cli-domain**) in a command to call a cloud service API.

```
g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="4ff018c3**********f31948" --cli-username=s******1 --cli-password=****** --cli-domain=s******1
```

NOTE

To call a cloud service API in this mode, **use a custom parameter** to pass the username and password in the command. This ensures account security and prevents password leakage.

8.4 Token Authentication

Enter a token (**cli-x-auth-token**) in a command to call a cloud service API.

```
g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="4ff018c3**********f31948" --cli-x-auth-token=******
```

8.5 ecsAgency Authentication

Assume that you have created an ECS agency. When using G42Cloud CLI on an ECS, add **--cli-mode=ecsAgency** in a command so that G42Cloud CLI can use the ECS agency to automatically obtain a temporary AK/SK and SecurityToken for authentication.

To use this authentication mode, ensure that you have created an ECS agency. If no ECS agency is available, create one in IAM. For details, see **Cloud Service Delegation**. Then add the agency in the **Management Information** > **Agency** area of the ECS details page.

8.6 Agency Authentication

By creating an agency, you can delegate another account to manage your resources based on assigned permissions. After a trust relationship is established, the delegated party can use the delegating account name (**cli-agency-domain-name**)/ID (**cli-agency-domain-id**), agency name (**cli-agency-name**), and the delegated party's authentication information (token or AK/SK) to get authenticated when calling APIs to manage and use resources of the delegating party.

To use this authentication mode, the delegating party must create an agency for the delegated party. If you are the delegating party, create an agency on the IAM console by referring to **Account Delegation**. If you are the delegated party, only you and users in the **admin** group can manage the delegated resources. To assign a common IAM user to manage the resources, authorize the user by referring to **Assigning Permissions to an IAM User (by a Delegated Party)**.

As the delegated party, when calling APIs to manage and use the cloud services and resources of the delegating party, use an AK/SK (access key or temporary security credentials), an account, or a token for authentication.

- AK/SK (delegated party's) authentication
  - Access key (permanent AK/SK)
    ```
    g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="4ff018c3**********f31948" --cli-agency-domain-id=13534326**********5cf67b --cli-agency-name=****** --cli-access-key=8NVT**********KIOV --cli-secret-key=VHMQjoC**********lsk3cGf
    ```
  - Temporary security credentials (temporary AK/SK and SecurityToken)
    ```
    g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="4ff018c3**********f31948" --cli-agency-domain-id=13534326**********5cf67b --cli-agency-name=****** --cli-access-key=5FSU**********607T --cli-secret-key=VoyjgLh**********qRc8pSq --cli-security-token=******
    ```
- Token (**cli-x-auth-token**) authentication
  ```
  g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="4ff018c3**********f31948" --cli-agency-domain-id=13534326**********5cf67b --cli-agency-name=****** --cli-x-auth-token=******
  ```
- Account (**cli-username**, **cli-password**, and **cli-domain**) authentication
  ```
  g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="4ff018c3**********f31948" --cli-agency-domain-name=q******2 --cli-agency-name=****** --cli-username=s******1 --cli-password=****** --cli-domain=s******1
  ```

9 Obtaining Authentication Information

9.1 Obtaining a Permanent AK/SK

Access keys (AK/SK) are an authentication mechanism of IAM. They are used to encrypt the signature of a request, ensuring that the request is secure and integral, and that identities of the request sender and receiver are correct.

- AK: a unique ID associated with an SK. It is used together with the SK to sign requests.
- SK: a key used together with the AK to sign requests. The AK and SK identify senders and prevent requests from being altered.

Constraint

You can create up to **two** access keys, which are permanently valid.

Viewing a Downloaded Access Key

If you have generated and downloaded an access key (AK/SK), find the local AK/SK file, which is generally named **credentials.csv**.

As shown in the following figure, the file contains a username, AK, and SK.

Figure 9-1 Content of the credential.csv file

	A	B	C
1	User Name	Access Key Id	Secret Access Key
2	[redacted]	CIA[redacted]PI	zr17[redacted]5uCy

Creating an Access Key

If no AK/SK file has been generated or can be found locally, create another access key.

Step 1 Log in to the console.

Step 2 On the top navigation menu, hover over the username and choose **My Credentials**.

Step 3 On the **My Credentials** page, choose **Access Keys** > **Create Access Key**.

Step 4 In the **Create Access Key** dialog box that is displayed, enter the password and verification code.

NOTE

- If no email address and mobile number have been bound to your account, enter only the login password.
- If you have bound an email address or mobile number to your account, verify your identity using either of them

Step 5 Click **OK**.

Step 6 Save the access key as prompted. The access key is saved in the default download folder of the browser.

NOTE

- Keep the access key secure. If you click **Cancel** in the download dialog box, the access keys will not be downloaded and cannot be downloaded later. You can create a new one if required.
- Rotate the access key (AK/SK) periodically.

Step 7 Open the downloaded **credentials.csv** file to obtain the access key (AK/SK).

----End

9.2 Obtaining an Account Name, Account ID, and Project ID

A project ID is required in most cases when you call a cloud service API. To obtain a project ID, perform the following operations:

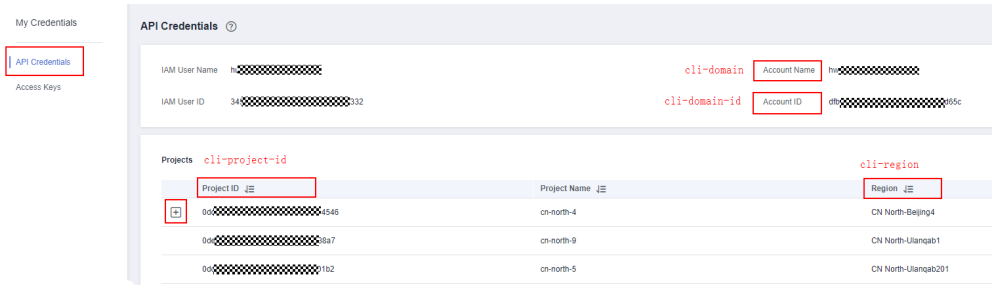
Step 1 Log in to the management console.

Step 2 Click the username in the upper right corner, choose **My Credentials** from the drop-down list, and view **Account Name** (**cli-domain**), **Account ID** (**cli-domain-id**), and **Project ID** (**cli-project-id**) on the displayed page.

Projects physically isolate cloud server resources by region, and multiple projects can be created in the same region for more fine-grained isolation. As shown in the following figure, find the region where your server locates, obtain the

corresponding project ID in the **Project ID** column, and click  on the left to obtain a subproject ID.

Figure 9-2 Viewing the account name, account ID, and project ID



NOTE

During API calling, G42Cloud CLI automatically obtains the account ID and project ID from the request header based on the authentication information of the current user. Therefore, you do not need to enter them in a command.

----End

9.3 Obtaining a Region

See **Regions and Endpoints**.

9.4 Obtaining a Token

A token is an access credential issued to an IAM user to bear the user's identity and permissions. The validity period of a token is 24 hours. Cache the token to prevent frequent API calling. The original token will be valid till it expires regardless of whether a new token has been obtained. Ensure that the token is valid when you use it. Using a token that will soon expire may cause API calling failures. You can obtain a token using any of the following methods:

Debugging an API in API Explorer

Set parameters on the page of the IAM API **KeystoneCreateUserTokenByPassword** on API Explorer, and click **Debug**. The value of **X-Subject-Token** in **Response** is a token.

Using Postman

Use Postman to obtain a token. If **201** is returned after you send a request, click **Header**. The value of **X-Subject-Token** is a token.

Using G42Cloud CLI

```
g42 IAM KeystoneCreateUserTokenByPassword --cli-region=${regionName} --auth.identity.methods.1="password" --auth.identity.password.user.name=${IAM user name} --auth.identity.password.user.password=${IAM user password} --auth.identity.password.user.domain.name=${Name of the account to which the IAM user belongs} --auth.scope.domain.name=${Name of the account to which the IAM user belongs} --cli-output=tsv --cli-query="response_header.X-Subject-Token[0]"
```

If the invocation is successful, the returned result is a token.

9.5 Obtaining a Temporary AK/SK and SecurityToken

A temporary AK/SK and SecurityToken are issued by the system to IAM users and are valid for 15 minutes to 24 hours. The temporary AK/SK and SecurityToken follow the principle of least privilege. For details, see **Obtaining a Temporary Access Key and SecurityToken Through a Token**.

10 FAQs

10.1 Overview

G42 Cloud Command Line Interface (G42Cloud CLI) classifies command errors into five types, each of which has a different identifier at the start of the error message. Troubleshoot the errors as follows:

- [NETWORK_ERROR]: HTTP request exceptions. Check the network connection.
- [CLI_ERROR]: G42Cloud CLI exceptions that occur during command processing. Contact the on-call personnel of G42Cloud CLI.
- [USE_ERROR]: Errors caused by an incorrect parameter in a command. Modify the parameter according to the error message.
- [OPENAPI_ERROR]: Errors that occur when calling a cloud service API. Contact the on-call personnel of the cloud service.

- [APIE_ERROR]: Errors that occur when calling API Explorer to obtain metadata. Contact the on-call personnel of API Explorer.

Alternatively, you can look through **the following FAQs** to find required information.

Table 10-1 FAQs

Category	Link
Authentication	**How Do I Obtain a Permanent AK/SK?**
	How Do I Obtain an Account Name, Account ID, and Project ID?
	How Do I Obtain a Region?
	How Do I Obtain a Token?
	How Do I Obtain a Temporary AK/SK and SecurityToken?
	Authentication Mode Priority
Profiles	**Which Profile Will Be Used by Default If No Profile Is Specified in a Command?**
Metadata cache	**Where Are Metadata Cache Files Stored? How Do I Clear Them?**
Logs	**Where Are Log Files Stored?**
Network connections	**What Can I Do If the Network Connection Times Out?**
Cloud services	**Why Am I Seeing a Message Indicating an Unsupported Service?**
Cloud service APIs	**Why Am I Seeing a Message Indicating an Unsupported Operation?**
	How Do I Specify a Cloud Service API and Its Version?
Regions	**Why Am I Seeing a Message Indicating that the cli-region Parameter Is Missing?**
	Why Am I Seeing a Message Indicating Unsupported cli-region?
Parameters	**What Are G42Cloud CLI System Parameters?**
	Why Am I Seeing a Message Indicating an Invalid Parameter?

Category	Link
	Why Are Old and New System Parameters (Such as region and cli-region) Available? Which Parameters Are Recommended?
	Why Am I Seeing a Message Indicating a Duplicate Parameter?
	How Do I Use cli-jsonInput?
	When Can I Use cli-jsonInput?
	Why Am I Seeing a Message Indicating an Unsupported Parameter Position or Type?
	How Do I Leave a Body Parameter Empty for Cloud Service APIs?
Interactive mode and autocomplete	How Do I Use Interactive Mode and Autocomplete?
Output formats	What Output Formats Are Supported by G42Cloud CLI?
	How Do I Use JMESPath Expressions?
	Which Built-in Functions Are Supported by JMESPath?
	Which G42Cloud CLI System Parameters Are Related to Data Output? Which Ones Are Recommended?
	How Do I Use cli-output, cli-query, and cli-output-num?
	How Do I Use cli-output-rows, cli-output-cols, and cli-output-num?
	What Are the Precautions for Using cli-output-rows, cli-output-cols, and cli-output-num?
	How Do I Use cli-json-filter?
	What Are the Precautions for Using cli-json-filter?
Other	How Do I Use G42Cloud CLI in Non-configuration Mode?
	Should I Enclose a Service Name, Operation, and Parameter Value in Quotation Marks in a Command?
	What Are the Application Scenarios of Online/Offline Modes?

Category	Link
	How Do I Uninstall G42Cloud CLI?

10.2 Authentication

10.2.1 Obtaining Authentication Information

How Do I Obtain a Permanent AK/SK?

See **Obtaining a Permanent AK/SK**.

How Do I Obtain an Account Name, Account ID, and Project ID?

See **Obtaining an Account Name, Account ID, and Project ID**.

How Do I Obtain a Region?

For details, see **Regions and Endpoints**.

How Do I Obtain a Token?

See **Obtaining a Token**.

How Do I Obtain a Temporary AK/SK and SecurityToken?

See **Obtaining a Temporary Access Key and SecurityToken Through a Token**.

10.2.2 Authentication Mode Priority

During command parsing, G42Cloud CLI performs authentication for API calling according to the following authentication mode priority:

1. Delegating account ID (**cli-agency-domain-id**) or name (**cli-agency-domain-name**), agency name (**cli-agency-name**), and profile (**cli-source-profile**) that stores the authentication information of the delegated party
2. AK/SK, account, or token authentication in non-configuration mode
   a. Access key (permanent AK/SK, that is, **cli-access-key** and **cli-secret-key**) or temporary security credentials (temporary AK/SK and SecurityToken, that is, **cli-access-key**, **cli-secret-key**, and **cli-security-token**)
   b. IAM user name (**cli-username**), IAM user password (**cli-password**), and **account name** (**cli-domain**)
   c. Token (**cli-x-auth-token**)
3. Profile specified in the command or the default profile
4. Cloud service agency (applicable only when G42Cloud CLI is used on an ECS)

Parameters of different authentication modes mentioned in **2** cannot be used interchangeably in the same command. If an exception occurs during

authentication mode parsing, authentication modes with a lower priority will not be used.

10.3 Profiles

10.3.1 Which Profile Will Be Used by Default If No Profile Is Specified in a Command?

Background

When you use G42Cloud CLI to manage and use your cloud service resources, it preferentially uses the profile specified by the **--cli-profile** option in the command during an API call.

If no profile is specified in the command, the **default profile** is used to call the target API.

If the profile does not match the target API or a parameter required for access to the target API is missing, an error message is displayed, for example:

- [USE_ERROR] Specify cli-region.
- [USE_ERROR] The value of cli-region is not supported. Supported regions: *
- [USE_ERROR] The following parameters are required: project_id
- [USE_ERROR] cli-domain-id is required for access to global services using AK/SK. Add this parameter or run `g42 configure set` to configure it.

Solution

- If you specify a profile with **--cli-profile** in the command, run the **g42 configure show --cli-profile=***${profileName}* command to view the profile, and check whether the specified profile is proper.
- If no profile is specified in the command, the **default profile** is used to call the target API. Run the **g42 configure show** command to query details about the default profile.
- To use other profiles, run the **g42 configure list** command to view all the configured profiles, use **--cli-profile=***${profileName}* in the API calling command to specify the name of the target profile, and try again.

10.4 Metadata Cache

10.4.1 Where Are Metadata Cache Files Stored? How Do I Clear Them?

When you use G42Cloud CLI to manage and use your cloud service resources, it remotely obtains details about the target cloud services and APIs in the commands. To reduce the remote calling times and accelerate response, G42Cloud CLI introduces the cache mechanism to store the information about cloud services and their APIs in local cache files, known as metadata cache files. Before the files

expire, the included information is used to verify and assemble parameters in commands.

- Storage directory of metadata cache files
  - Online mode
    - Windows: **C:\Users***{Your Windows username}***.g42\metaRepo**
    - Linux: **/home/***{Current username}***/.g42/metaRepo/**
    - macOS: **/Users/***{Current username}***/.g42/metaRepo/**
  - Offline mode
    - Windows: **C:\Users***{Your Windows username}***.g42\metaOrigin**
    - Linux: **/home/***{Current username}***/.g42/metaOrigin/**
    - macOS: **/Users/***{Current username}***/.g42/metaOrigin/**
- Procedure for clearing metadata cache files
  - Online mode

    To clear metadata cache files, run **g42 meta clear**. After they are cleared, data is obtained and written to new cache files during API calling.
  - Offline mode

    Run the **g42 meta clear** command to clear the metadata cache files parsed from the downloaded offline metadata package. The package will remain. During API calling, this package will be parsed again and new metadata cache files will be written into it.

10.5 Logs

10.5.1 Where Are Log Files Stored?

G42Cloud CLI records the logs generated for running commands. Currently, log recording cannot be disabled.

Log files are named **g42.log** and stored in the following directories:

- Linux: **/home/***{Current username}***/.g42/log/**
- Windows: **C:\Users***{Current username}***.g42\log**
- macOS: **/Users/***{Current username}***/.g42/log/**

10.6 Network Connections

10.6.1 What Can I Do If the Network Connection Times Out?

Background

During a cloud service API call, G42Cloud CLI verifies the validity of your input parameters. Specifically, it first remotely obtains the details about the cloud

service and API, and then remotely calls the target API. If the request fails due to a network connection error, an error message is displayed. For example:

- [NETWORK_ERROR] Connection timed out. Check network connectivity.
- [USE_ERROR] API calling timed out. Check the value of readTimeout in the profile or command.
- [NETWORK_ERROR] Connection timed out * consecutive times (reconnection attempts: *). Check network connectivity.

Solution

1. Check whether your network connection is normal.
2. If the network connection is normal but the error message indicates that the connection timed out, the value of **cli-connect-timeout** in the profile or command may be too small. Change the value as follows:
   - If the **--cli-connect-timeout** option is used in the command, increase the value and try again.
   - If the **--cli-connect-timeout** option is not used in the command, the value of this parameter in the current profile is used during command execution. Add **--cli-connect-timeout=***${connectTimeout}* to the current command to temporarily overwrite the parameter value in the profile, and try again. To change the parameter value in the profile, run the **g42 configure set --cli-profile=***${profileName}* **--cli-connect-timeout=***${connectTimeout}* command.
3. If the network connection is normal but the error message prompts you to check **readTimeout**, the value of **cli-read-timeout** in the profile or command may be too small. Change the value as follows:
   - If the **--cli-read-timeout** option is used in the command, increase the value and try again.
   - If the **--cli-read-timeout** option is not used in the command, the value of this parameter in the current profile is used during command execution. Add **--cli-read-timeout=***${readTimeout}* to the current command to temporarily overwrite the parameter value in the profile, and try again. To change the parameter value in the profile, run the **g42 configure set --cli-profile=***${profileName}* **--cli-read-timeout=***${readTimeout}* command.

10.7 Cloud Services

10.7.1 Why Am I Seeing a Message Indicating an Unsupported Service?

Background

During a cloud service API call, G42Cloud CLI verifies the validity of your input parameters. If the cloud service name in the command is incorrect, or the cloud service is not released on G42Cloud CLI, an error message is displayed:

[USE_ERROR] Unsupported service: *

Solution

1. Run the **g42 --help** command to view all supported cloud services and check whether the service name is correct.
2. If the service name is correct but the service is not in the **Available services** list output by the **g42 --help** command, check either of the following reasons:
   a. Using the **online mode**: The cloud service has not been launched on G42Cloud CLI.
   b. Using the **offline mode**: The offline data package you used may not be the latest version, resulting in a parameter verification failure. In this case, run the **g42 meta download** command to update the offline data package, and then run the command in **1** again. If the cloud service still does not exist in the result, it is unavailable for G42Cloud CLI offline mode. Try again when the offline data package is updated, or **use the online mode** instead.
3. Determine your current language environment based on the language setting on G42Cloud CLI or your system time zone. Currently, the cloud services released in the English environment are different from those released in the Chinese environment. To switch the language, run the **g42 configure set --cli-lang=en** command.

10.8 Cloud Service APIs

10.8.1 Why Am I Seeing a Message Indicating an Unsupported Operation?

Background

During a cloud service API call, G42Cloud CLI verifies the validity of your input parameters. If the API operation in the command is incorrect or the API is not released on G42Cloud CLI, the following error message is displayed:

[USE_ERROR] Unsupported operation: *

Solution

1. Run the **g42 <service> --help** command to view the list of supported operations and check whether the operation is correct.
2. If the operation is correct but not included in the **Available Operations** list output by the **g42 <service> --help** command, check either of the following reasons:
   a. Using the **online mode**: The corresponding API has not been launched on G42Cloud CLI.
   b. Using the **offline mode**: The offline data package you used may not be the latest version, resulting in a parameter verification failure. In this case, run the **g42 meta download** command to update the offline data package, and then run the command in **1** again. If the operation still does not exist in the result, the corresponding API is unavailable for G42Cloud

CLI offline mode. Try again when the offline data package is updated, or **use the online mode** instead.

3. Determine your current language environment based on the language setting on G42Cloud CLI or your system time zone. Currently, the cloud services and APIs released in the English environment are different from those in the Chinese environment. To switch the language, run the **g42 configure set --cli-lang=en** command.

10.8.2 How Do I Specify a Cloud Service API and Its Version?

Background

During a cloud service API call, G42Cloud CLI verifies the validity of your input parameters. If the service has multiple versions, some or all APIs of the service also have multiple versions. The parameters and application scenarios of an API may vary with versions. The CLI needs to obtain the version information of an API that belongs to a service with multiple versions.

Querying and Specifying the Version of a Cloud Service API

- Querying version information

  Run **g42 <service> --help** to view the operation list of the cloud service. If an operation appears multiple times in the **Available Operations** list and different version numbers are concatenated with a slash (/), specify a version when calling the API. For details, see **Specifying a version**. For other operations that appear only once in the operation list, their version numbers do not need to be concatenated. G42Cloud CLI calls their only version.

- Specifying a version
  - Manually specifying a version

    When calling an API that belongs to a service with multiple versions, use a slash (/) to manually add the version number after **operation** of the original API. For example, if **operation** is **showLogs** in the original command and **showLogs/v1** and **showLogs/v2** exist in the **Available Operations** list, you can set **operation** to **showLogs/v1** or **showLogs/v2**.
  - Using the default version

    When calling an API that belongs to a service with multiple versions, if you do not specify a version in the command, G42Cloud CLI obtains all available versions of the API and sorts them in alphabetical order. The last version of the API in the list is called by default. (Generally, this version is the latest version of the API.)

Extended Question and Solution

- Question

  When you call an API that belongs to a service with multiple versions by **using the API's default version**, if the metadata files cached locally are modified, G42Cloud CLI may not be able to correctly parse the API version based on the cache files. In this case, the following error message is displayed:

  [USE_ERROR] The API has multiple versions. Specify one.

- Solution

Run **g42 meta clear** to clear the cache files, and try again.

10.9 Regions

10.9.1 Why Am I Seeing a Message Indicating that the cli-region Parameter Is Missing?

Background

When you use G42Cloud CLI to call cloud service APIs, a proper value of **cli-region** is required. If you have neither specified the value of **cli-region** in the command, nor configured it in the current profile, the following error message is displayed:

[USE_ERROR] Specify cli-region.

Solution

1. If the value of **cli-region** is not specified in the command, add **--cli-region=***${regionValue}* to the command and try again.
2. If there is a region you often use, run the **g42 configure set --cli-profile=***${profileName}* **--cli-region=***${regionValue}* command to add it to the target profile. When you use the profile again to call an API, you do not need to enter the value of **cli-region** in the command. However, if the target API is not available in this region, enter **--cli-region=***${regionValue}* in the command to specify a supported region.

10.9.2 Why Am I Seeing a Message Indicating Unsupported cli-region?

Background

When you use G42Cloud CLI to call cloud service APIs, a proper value of **cli-region** is required. If you encounter any of the following situations:

- The value of **cli-region** in the command is incorrect.
- The value of **cli-region** is specified in the command, but the target API does not support the region.
- The value of **cli-region** is not specified in the command, and the target API does not support the region obtained from the current profile.

Either of the following error messages is displayed:

[USE_ERROR] The value of cli-region is not supported. Supported regions: *

[USE_ERROR] Value of cli-region in the current profile is not supported. Supported regions: *

Solution

1. View the supported regions in the prompt and check whether the **cli-region** value you specified is correct.

2. If the value is correct but one of the preceding error messages is displayed during command execution, check the following possible causes:
   a. Using the **online mode**: The target API does not support the **cli-region**. Modify the parameter as follows:
      i. If you have specified the value of **cli-region** in the command, change this value based on the list of supported regions in the prompt message, and try again.
      ii. If the value of **cli-region** is not specified in the command, G42Cloud CLI parses and uses the value in the current profile. Add **--cli-region=***${regionValue}* to the current command based on the list of supported regions in the error message, and try again. To change the value of **cli-region** in the profile, run the **g42 configure set --cli-profile=***${profileName}* **--cli-region=***${regionValue}* command.
   b. Using the **offline mode**: The offline data package you used may not be the latest version, resulting in a parameter verification failure. In this case, run the **g42 meta download** command to update the offline data package, and run the original command again. If the error persists, the **cli-region** value is unavailable for G42Cloud CLI offline mode. Try again when the offline data package is updated, or **use the online mode** instead.
3. If the preceding error messages are not displayed during the command execution, but the returned value of the called API indicates that the region is incorrect, the local cache files containing the **cli-region** list may be modified. As a result, the parameter verification of G42Cloud CLI is inaccurate. In this case, run the **g42 meta clear** command to clear the local cache files, and try again.
4. Determine your current language environment based on the language setting on G42Cloud CLI or your system time zone. Currently, the regions (**cli-region**) supported by cloud service APIs in the English environment are different from those in the Chinese environment. To switch the language, run the **g42 configure set --cli-lang=en** command.

10.10 Parameters

10.10.1 What Are G42Cloud CLI System Parameters?

System Parameters

G42Cloud CLI system parameters are internal parameters. **Table 1** describes the system parameters and their usage.

Table 10-2 G42Cloud CLI's new system parameters

Parameter	Description	How to Use
help	Prints help information.	Use it directly in a command.

Parameter	Description	How to Use
debug	Prints debugging information.	Use it directly in a command.
dryrun	Prints the request message after verification, without execution.	Use it directly in a command.
interactive	Puts you into the interactive mode.	Use it directly in a command.
cli-region	Region.	Configure it in a profile or use it directly in a command.
cli-access-key	Access key ID required in AK/SK mode.	Configure it in a profile or use it directly in a command.
cli-secret-key	Secret access key required in AK/SK mode.	Configure it in a profile or use it directly in a command.
cli-username	Username.	Use it directly in a command.
cli-password	Password.	Use it directly in a command.
cli-domain	Account name.	Use it directly in a command.
cli-domain-id	Account ID.	Configure it in a profile or use it directly in a command.
cli-project-id	Project ID	Configure it in a profile or use it directly in a command.
cli-profile	Profile. If not specified, the default one is used.	Configure it in a profile or use it directly in a command.
cli-mode	Authentication mode (**AKSK**, **token**, **ecsAgency**, or **agency**).	Configure it in a profile or use it directly in a command.

Parameter	Description	How to Use
cli-jsonInput	Passes API parameters using a JSON file.	Use it directly in a command.
cli-output	Response data output format (**json**, **table**, or **tsv**).	Use it directly in a command.
cli-query	JMESPath for filtering response data.	Use it directly in a command.
cli-output-num	Whether to print the row numbers during table output. The value can be **true** (default) or **false**.	Use it directly in a command.
cli-endpoint	Custom domain name.	Use it directly in a command.
cli-connect-timeout	Request connection timeout, in seconds. The default value is 5s, and the minimum value is 1s.	Configure it in a profile or use it directly in a command.
cli-read-timeout	I/O timeout, in seconds. The default value is 10s, and the minimum value is 1s.	Configure it in a profile or use it directly in a command.
cli-retry-count	Number of connection attempts. The value ranges from 0 to 5. The default value is **0**.	Configure it in a profile or use it directly in a command.
cli-x-auth-token	Specifies the access token issued to an IAM user to carry its identity and permissions.	Configure it in a profile or use it directly in a command.
cli-security-token	Specifies a temporary token, which must be used together with a temporary AK/SK.	Configure it in a profile or use it directly in a command.
cli-lang	Language, which can be **cn** or **en**.	Configure it in a profile.
cli-offline	Specifies whether to use offline mode. The value can be **true** or **false**. The default value is **false**.	Configure it in a profile.
cli-skip-secure-verify	Specifies whether to skip HTTPS certificate verification (not recommended). The value can be **true** or **false** (default).	Configure it in a profile or use it directly in a command.

Parameter	Description	How to Use
cli-agency-domain-name	Account name of a delegating party. This parameter must be used together with **cli-agency-name**.	Configure it in a profile or use it directly in a command.
cli-agency-domain-id	Account ID of the delegating party. This parameter must be used together with **cli-agency-name**.	Configure it in a profile or use it directly in a command.
cli-agency-name	Agency name. It must be used together with **cli-agency-domain-id** or **cli-agency-domain-name**.	Configure it in a profile or use it directly in a command.
cli-source-profile	The profile that stores the authentication information of the delegated party. The value of **cli-source-profile** cannot be the current profile.	Configure it in a profile or use it directly in a command.
cli-agree-privacy-statement	Whether to agree to the privacy statement. The value can be **true** or **false** (default).	Configure it in a profile.

The ways parameters listed in **Table 1** can be used are described as follows:

- Configure it in a profile.

  The parameter can be used only after being configured in a profile by running **g42 configure set --key1=value1 --key2=value2 ...** The profile name can be specified using **--cli-profile=***${profileName}*. G42Cloud CLI parses and uses the parameter values configured in the profile during the running process.

  If you use the parameter directly in a command, the following error message is displayed:

  [USE_ERROR] Invalid parameter: *
- Use it directly in a command.

  The parameter can be used only in a command in the format of **--key1=value1 --key2=value2 ...**.

  If you configure the parameter in a profile, the following error message is displayed:

  [USE_ERROR] Invalid parameter: *
- Configure it in a profile or use it directly in a command.

  The parameter can be used after being configured in a profile by running **g42 configure set --key1=value1 --key2=value2 ...**. Alternatively, it can be directly used in a command in the format of **--key1=value1 --key2=value2 ...** G42Cloud CLI preferentially uses the parameter value specified in the command. If the parameter is not specified in the command, its value in the current profile is used.

For details about the old G42Cloud CLI system parameters, see **Table 10-3**.

10.10.2 Why Am I Seeing a Message Indicating an Invalid Parameter?

Background

During a cloud service API call to manage and use your cloud service resources, G42Cloud CLI verifies the validity of your input parameters. If you enter an unsupported parameter in a command or pass a parameter that can be used only in a profile into a command, the following error message is displayed:

[USE_ERROR] Invalid parameter: *

Solution

1. Run the **g42 <service> <operation> --help** command and check the parameters in **Params** of the command output. If the parameter does not exist, check either of the following reasons:
   a. Using the **online mode**: The API does not support this parameter. Modify the parameter based on the value of **Params** in the command output.
   b. Using the **offline mode**: The offline data package you used may not be the latest version, resulting in a parameter verification failure. In this case, run the **g42 meta download** command to update the offline data package, and then run the command in **1** again. If the parameter still does not exist in the result, the API data is unavailable for G42Cloud CLI offline mode. Try again when the offline data package is updated, or **use the online mode** instead.
2. If you have entered system parameters (such as **cli-lang**) in the command for calling a cloud service API, an error message is displayed. The parameters can only be configured in a profile by running **g42 configure set --key1=value1 --key2=value2 ...**
3. If the preceding error message is not displayed during the command execution, but the returned value of the called API indicates a parameter error, the local cache files containing the API details may be modified. As a result, the parameter verification is inaccurate. In this case, run the **g42 meta clear** command to clear the cache files and run the **g42 <service> <operation> --help** command again to check whether the parameter is included in the parameter list of the target API.

10.10.3 Why Are Old and New G42Cloud CLI System Parameters (Such as region and cli-region) Available? Which Parameters Are Recommended?

Background

Among the G42Cloud CLI system parameters, some have two forms, for example, **--region** and **--cli-region**. The parameters without the prefix **cli-** are **old system parameters**. Those with the prefix **cli-** are **new system parameters**. This happens because some parameters of cloud service APIs may have the same names as old

system parameters. Two parameters with the same name in a command may be used for different purposes. That is, one is used as a parameter of the target API, and the other is used as a system parameter. When checking the validity of parameters, the following error message may be displayed:

[USE_ERROR] Duplicate *. Change the key of the G42Cloud CLI system parameter to cli-*.

In addition, if the cloud service API contains a parameter (or a custom one) with the same name as an old system parameter, G42Cloud CLI cannot determine the function if the API parameter appears in a command. Therefore, when parsing the command, the CLI confirms the actual function of the parameter with you through interactive information to prevent errors. Example:

- The target API contains a parameter with the same name as the G42Cloud CLI system parameter %s (unknown location %s). Confirm whether this parameter is a G42Cloud CLI system parameter (a), a target API parameter (b), or both (c):
- You can define a parameter for the target API using the same name as the G42Cloud CLI system parameter %s (unknown location %s). Confirm whether this parameter is a G42Cloud CLI system parameter (a), a target API parameter (b), or both (c):

When you construct G42Cloud CLI commands, use the new system parameters to prevent errors or interactions caused by parameter conflicts.

NOTE

The new system parameters have been added to the new parameter list. The old system parameters can still be used but will not be upgraded.

Old System Parameters

The **following table** describes G42Cloud CLI's old system parameters and the corresponding new system parameters.

Table 10-3 G42Cloud CLI's old system parameters

Old Parameter	Description	New Parameter
region	Region.	cli-region
access-key	Access key ID required in AK/SK mode.	cli-access-key
secret-key	Secret access key required in AK/SK mode.	cli-secret-key
username	Username.	cli-username
password	Password.	cli-password
domain	Account name.	cli-domain
domain-id	Account ID	cli-domain-id
project-id	Project ID.	cli-project-id

Old Parameter	Description	New Parameter
profile	Profile.	cli-profile
mode	Authentication mode (**AKSK**, **token**, **ecsAgency**, or **agency**).	cli-mode
jsonInput	Passes API parameters using a JSON file.	cli-jsonInput
output-cols	Specifies the fields to print during table output.	cli-output-cols
output-rows	Specifies the levels to print during table output.	cli-output-rows
output-num	Whether to print the row numbers during table output. The value can be **true** (default) or **false**.	cli-output-num
json-filter	Performs a JMESPath query on the output JSON result.	cli-json-filter
connect-timeout	Request connection timeout, in seconds. The default value is 5s, and the minimum value is 1s.	cli-connect-timeout
read-timeout	I/O timeout, in seconds. The default value is 10s, and the minimum value is 1s.	cli-read-timeout
retry-count	Specifies the number of connection attempts. The value ranges from 0 to 5. The default value is **0**.	cli-retry-count
X-Auth-Token	Specifies the access token issued to an IAM user to carry its identity and permissions.	cli-x-auth-token
security-token	Specifies a temporary token, which must be used together with a temporary AK/SK.	cli-security-token
lang	Language, which can be **cn** or **en**.	cli-lang

10.10.4 Why Am I Seeing a Message Indicating a Duplicate Parameter?

Background

G42Cloud CLI checks the validity of parameters during command execution. If a command contains duplicate parameters, different error messages will be displayed accordingly. For example:

1. [USE_ERROR] Duplicate parameter: *
2. [USE_ERROR] Duplicate *. Change the key of the G42Cloud CLI system parameter to cli-*.
3. [USE_ERROR] Duplicate *. Enter the API parameter using '--cli-jsonInput=xx'. For details, see...

Among the G42Cloud CLI system parameters, some have two forms, for example, **--region** and **--cli-region**. This happens because some parameters of cloud service APIs may have the same names as system parameters.

Solution

1. If the first error message is displayed, duplicate non-system parameters may exist in the command. Check whether the parameters are entered correctly. This error may also occur when the system parses the command content. If the parameter value contains special characters, enclose the value with double quotation marks ("") to prevent parsing errors.
2. If the second error message is displayed, a duplicate **old system parameter** exists in the command. Run the **g42 <service> <operation> --help** command and compare the value of **Params** in the command output (that is, the parameter list of the current API) to check whether the parameter exists in the target API or whether a parameter with a custom name (that is, the parameter whose name is {*}) exists. If not, check whether the input is correct. If yes, use the two parameters with the same name for different purposes, one as a parameter of the target API, and the other as a system parameter. Replace the old system parameter in the command with the **new one** as prompted. If a command contains both the old and new forms of a system parameter, for example, **--cli-region=${regionValue1} --region=${regionValue2}**, G42Cloud CLI automatically identifies the usage of each parameter based on the parameter list of the current API.
   - If the target API has the **region** parameter or a parameter with a custom name, G42Cloud CLI automatically identifies **--cli-region** as a system parameter and uses its value to obtain details about the target API. G42Cloud CLI identifies **--region** as a parameter of the target API and uses its value to call the API.
   - If the target API does not have the **region** parameter or a parameter with a custom name, G42Cloud CLI automatically identifies **--cli-region** as a system parameter and uses its value to obtain details about the target API. G42Cloud CLI ignores **--region** passed to the command.

   When you construct G42Cloud CLI commands, **use the new system parameters** to prevent errors or interactions caused by parameter conflicts.

3. If the third error message is displayed, a duplicate new system parameter exists in the command. Run the **g42 <service> <operation> --help** command and compare the value of **Params** in the command output (that is, the parameter list of the current API) to check whether the parameter exists in the target API or whether a parameter with a custom name (that is, the parameter whose name is {*}) exists. If not, check whether the input is correct. If yes (this conflict rarely occurs), write the API parameters in the command into the **cli-jsonInput** file and **call the command using this JSON file**.

10.10.5 How Do I Use cli-jsonInput?

Background

Tools such as the command prompt (**cmd.exe**) have restrictions on the allowed maximum string length. If there are too many parameters in a command or the parameter values are too long, the parameters entered may be incomplete due to the length limit. In this case, use **--cli-jsonInput=***${jsonFileName}* to pass cloud service API parameters through a JSON file. G42Cloud CLI parses and uses the parameters in the JSON file to call the target API.

Usage Description

For details about how to use **cli-jsonInput**, see **Passing API Parameters with JSON File**.

Precautions

- The JSON file passed by using the **--cli-jsonInput** option supports only API parameters of cloud services and does not support system parameters. If a parameter of the target API has the same name as **a new system parameter** or **an old system parameter**, the one written into the **jsonInput** file is identified as an API parameter by default.
- In the JSON file passed by using the **--cli-jsonInput** option, G42Cloud CLI obtains and parses the parameter value based on the key at the outermost layer of JSON. Currently, the supported keys include **path**, **query**, **body**, **formData**, **header**, and **cookie**. The content under other keys at the outermost layer of JSON will be ignored. If none of the keys at this layer is supported, the following error message is displayed:

  [USE_ERROR] The cli-jsonInput file contains invalid content. For details, see...
- When you use the **--cli-jsonInput** option to pass cloud service API parameters, all API parameters in the same position must be written into a JSON file or directly passed using a command. Otherwise, the parameters may not be completely parsed, and the following error message is displayed:

  [USE_ERROR] The following parameters are required: *
- The **--cli-jsonInput** option supports only JSON files with the .json extension. The maximum file size is 5 MB. When **--cli-jsonInput** is used, the JSON file format and the parameter types in the file are verified. If the JSON file has incorrect format, the following error message is displayed:

  [USE_ERROR]Failed to parse file for the cli-jsonInput parameter. Cause: File contains invalid parameters.

If the JSON file contains an unsupported parameter type, the following error message is displayed:

[USE_ERROR] Value type of * is not supported.

- When you use the **--cli-jsonInput** option to pass cloud service API parameters, parameter values cannot be custom parameters.

10.10.6 When Can I Use cli-jsonInput?

Use **cli-jsonInput** to pass the following API parameters so that they can be correctly parsed by G42Cloud CLI:

- API parameters with periods (.)
- API parameters with the same name in different positions
- API parameters with spaces or special characters
- API parameters that are too long

10.10.7 Why Am I Seeing a Message Indicating an Unsupported Parameter Position or Type?

Background

During a cloud service API call to manage and use your cloud service resources, G42Cloud CLI verifies the validity of your input parameters. G42Cloud CLI obtains details about all parameters of the API, including the parameter type and position in the request. If the locally cached metadata files are modified, G42Cloud CLI may not be able to correctly parse the API parameter details during the running process. The following error message will be displayed during verification:

- [CLI_ERROR] * is in an incorrect position: *
- [USE_ERROR] Unsupported parameter type: key=*, type=*

Solution

Run **g42 meta clear** to clear the cached metadata files, and try again.

10.10.8 How Do I Leave a Body Parameter Empty for Cloud Service APIs?

G42Cloud CLI allows you to leave the body parameters of cloud service APIs empty at any level.

- A map parameter can be left empty as **{}** at the corresponding level.
- An array parameter can be left empty as **[]** at the corresponding level.

Take the **BatchStopServers** operation of ECS as an example. There are two parameters in the body: **os-stop.servers.[N].id** and **os-stop.type**.

```
g42 ECS BatchStopServers --cli-region=ae-ad-1 --help

G42Cloud CLI Version 3.3.6 Copyright(C) 2022-2022 www.g42cloud.com

Service:
  ECS
```

```
Description:
This API is used to stop ECSs in a batch based on the specified ECS ID list. A maximum of 1000 ECSs can be stopped at a time.

Method:
  POST

Params:
  --cli-region
    required    string  Region where the API can be called. If no region is specified in the command, cli-region in the current profile is used.
  --os-stop.servers.[N].id
    required    string  body    ECS Instance ID. Format: --os-stop.servers.1.id=value1 ...
  --project_id
    required    string  path    Specifies the project ID. If no project ID is specified in the command, either the parent project ID of the specified region in the authentication information or cli-project-id in the current profile is used.
  --os-stop.type
    optional    string  body    Specifies an ECS stop type. The default value is SOFT. [SOFT|HARD]
    - SOFT: normal ECS stop (default)
    - HARD: forcible ECS stop
```

- No parameters left empty

  Pass the values of **os-stop.servers.[N].id** and **os-stop.type**, and run **--dryrun** to view the request body:

  ```
  g42 ECS BatchStopServers --cli-region=ae-ad-1 --os-stop.servers.1.id="test" --os-stop.type="SOFT" --dryrun
  -------- The execution is eliminated in dry-run mode. Current request: --------
  POST https://ecs.ae-ad-1.g42cloud.com/v1/0a152ab***************262d035e8/cloudservers/action
  Content-Type: application/json;charset=UTF-8
  X-Project-Id: 0a152ab***************262d035e8
  X-Sdk-Date: 20221116T121721Z
  Authorization: ****

  {
    "os-stop": {
      "servers": [
        {
          "id": "test"
        }
      ],
      "type": "SOFT"
    }
  }
  ```

- Array parameter left empty

  Pass **--os-stop.servers="[]"** for the array parameter **os-stop.servers** of **os-stop.servers.[N].id** to leave **[N]** and the remaining part empty. Then run **--dryrun** to view the request body:

  ```
  g42 ECS BatchStopServers --cli-region=ae-ad-1 --os-stop.servers="[]" --os-stop.type="SOFT" --dryrun
  -------- The execution is eliminated in dry-run mode. Current request: --------
  POST https://ecs.ae-ad-1.g42cloud.com/v1/0a152ab***************262d035e8/cloudservers/action
  X-Project-Id: 0a152ab***************262d035e8
  X-Sdk-Date: 20221116T122841Z
  Authorization: ****
  Content-Type: application/json;charset=UTF-8

  {
    "os-stop": {
      "servers": [],
      "type": "SOFT"
    }
  }
  ```

- Map parameter left empty

Pass **--os-stop="{}"** for the common parent **os-stop** of **os-stop.servers.[N].id** and **os-stop.type** to leave the map parameter empty. Then run **--dryrun** to view the request body:

```
g42 ECS BatchStopServers --cli-region=ae-ad-1 --os-stop="{}" --dryrun
-------- The execution is eliminated in dry-run mode. Current request: --------
POST https://ecs.ae-ad-1.g42cloud.com/v1/0a152ab***************262d035e8/cloudservers/action
Content-Type: application/json;charset=UTF-8
X-Project-Id: 0a152ab***************262d035e8
X-Sdk-Date: 20221117T013616Z
Authorization: ****

{
  "os-stop": {}
}
```

G42Cloud CLI checks the parameter values during command execution. It displays an error message if an empty value does not match the parameter type. For example, if the empty value **[]** for an array parameter is assigned to the map parameter **os-stop**, the following error message is displayed:

[USE_ERROR] The value of map parameter os-stop is invalid.

10.11 Interactive Mode and Autocomplete

10.11.1 How Do I Use Interactive Mode and Autocomplete?

To enable autocomplete in the bash environment, run the **g42 auto-complete on** command. Pay attention to the following:

- If the prompted parameter name contains **[N]**, which indicates an index, replace it with a number. If the prompted parameter name contains **{*}**, which indicates a custom parameter name, replace it with a string that does not contain a period (.).
- If multiple directories of the same user contain G42Cloud CLI on the same server, the autocomplete function enabled for G42Cloud CLI in one of the directories will also take effect for the CLI in the other directories.

Note the following when using the interactive mode:

If the prompted parameter name contains **[N]**, which indicates an index, replace it with a number. If the prompted parameter name contains **{*}**, which indicates a custom parameter name, replace it with a string that does not contain a period (.).

10.12 Output Formats

10.12.1 What Output Formats Are Supported by G42Cloud CLI?

G42Cloud CLI supports three output formats: **json**, **table**, and **tsv**. By default, the output is in JSON format. Use the **--cli-output** parameter to specify any of the preceding output formats. Use the **--cli-query** parameter to pass a JMESPath expression and perform a JMESPath query on the JSON result to filter the required information. For details about the output, see **Specifying Output Format**. Learn how to **define a JMESPath expression**.

10.12.2 How Do I Define a JMESPath Expression?

10.12.2.1 How Do I Use JMESPath Expressions?

You can use JMESPath expressions as follows:

- Basic expression
  - Identifier

    The simplest JMESPath expression is an identifier, which specifies a key in a JSON object:

    ```
    {"a": "foo", "b": "bar", "c": "baz"}
    ```

    For the preceding JSON content, if the expression is **a**, the result **foo** is obtained.

    If you specify a key that does not exist, G42Cloud CLI displays an error message and outputs the original JSON result.
  - Subexpression

    Use a subexpression to return the nested value in a JSON object.

    ```
    {"a": {"b": {"c": {"d": "value"}}}}
    ```

    For the preceding JSON content, if the expression is **a.b.c.d**, the result **value** is obtained.

    If you specify a key that does not exist, G42Cloud CLI displays an error message and outputs the original JSON result.
  - Index expression

    Index expressions allow you to select a specific element in a list. Indexing is zero-based.

    ```
    ["a", "b", "c", "d", "e", "f"]
    ```

    For the preceding JSON content, if the expression is **[1]**, the result **b** is obtained.

    If you specify an index that is larger than the list, G42Cloud CLI displays an error message and outputs the original JSON result. You can also use negative indexing to index from the end of the list. **[-1]** indicates the last element in the list, and **[-2]** indicates the last but one element.
  - You can combine identifiers, subexpressions, and index expressions to access JSON elements.

    ```
    {"a": {
      "b": {
        "c": [
          {"d": [0, [1, 2]]},
          {"d": [3, 4]}
        ]
      }
    }}
    ```

    For the preceding JSON content, if the expression is **a.b.c[0].d[1][0]**, the result **1** is obtained.
- Slice

  The general form of a slice is *[start:stop:step]*. By default, the step value is **1**. Therefore, the form can be *[start:stop]*. Slices allow you to select a contiguous subset of an array. In its simplest form, you can specify the starting index and the ending index. The ending index will not be included in the slice.

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

For the preceding JSON content, if the expression is **[0:5]**, the following result is obtained:

```
[
  0,
  1,
  2,
  3,
  4
]
```

This slice result contains the elements **0**, **1**, **2**, **3**, and **4**. The element at index **5** is not included.

If the expression is **[5:10]**, the following result is obtained:

```
[
  5,
  6,
  7,
  8,
  9
]
```

The two examples above can be shortened. If the **start** or **stop** value is omitted, the array starts from the first element or stops from the last element by default. For example:

If the expression is **[:5]**, the following result is obtained:

```
[
  0,
  1,
  2,
  3,
  4
]
```

By default, the step value is **1**, which means to include every element in the range specified by the **start** and **stop** value. You can use the **step** value to skip over elements. For example, to select only the even elements from an array, use the expression **[::2]**. The following result is obtained:

```
[
  0,
  2,
  4,
  6,
  8
]
```

Also note that in this example, the **start** and **stop** values are omitted, meaning that **0** is used for the **start** value and **10** is used for the **stop** value. In this example, the expression **[::2]** is equivalent to **[0:10:2]**.

If the **step** value is negative, the slice is created in reverse order. For example, if the expression is **[::-1]**, the following result is obtained:

```
[
  9,
  8,
  7,
  6,
  5,
  4,
  3,
  2,
  1,
  0
]
```

- Projection

  Projections are one of the key features of JMESPath. It allows you to apply an expression to a collection of elements. There are five types of projection: list, slice, object, flatten, and filter projections.

  – List projection

    A wildcard expression creates a list projection, which is a projection over a JSON array.

    ```
    {
      "people": [
        {"first": "James", "last": "d"},
        {"first": "Jacob", "last": "e"},
        {"first": "Jayden", "last": "f"},
        {"missing": "different"}
      ],
      "foo": {"bar": "baz"}
    }
    ```

    For the preceding JSON content, if the expression is **people[*].first**, the following result is obtained:

    ```
    [
      "James",
      "Jacob",
      "Jayden"
    ]
    ```

    In the above example, the **first** expression is applied to each element in the **people** array. The results are collected into a JSON array and returned as the result of the expression. For example, the expression **foo[*].bar.baz[0]** projects the **bar.baz[0]** expression to each element in the **foo** array.

    Pay attention to the following when using projections:

    - Projections are evaluated as two steps. The left hand side (LHS) creates a JSON array of initial values. The right hand side (RHS) of a projection is the expression to project for each element in the JSON array created by the LHS. Each projection type has slightly different semantics when evaluating the LHS and RHS.
    - If the result of the expression projected onto an individual array element is **null**, then that value is omitted from the collected set of results.
    - You can stop the projection using a pipe expression (discussed later).
    - A list projection is only valid for a JSON array. If the LHS cannot create a JSON array of initial values, G42Cloud CLI displays an error message and outputs the original JSON result.

    Note that **people[*].first** only included three elements, even though the **people** array has four elements. This is because the last element **{"missing": "different"}** evaluates to **null** when the expression **first** is applied, and **null** values are not added to the collected result array. If you try the expression **foo[*].bar**, G42Cloud CLI displays an error message and outputs the original JSON result, because the value associated with the **foo** key is a JSON object, not an array.

  – Slice projection

Slice projections are almost identical to list projections, with the exception that the LHS is the result of evaluating the slice, which may not include all the elements in the original list:

```
{
  "people": [
    {"first": "James", "last": "d"},
    {"first": "Jacob", "last": "e"},
    {"first": "Jayden", "last": "f"},
    {"missing": "different"}
  ],
  "foo": {"bar": "baz"}
}
```

For the preceding JSON content, if the expression is **people[:2].first**, the following result is obtained:

```
[
  "James",
  "Jacob"
]
```

- Object projection

  Whereas a list projection is defined for a JSON array, an object projection is defined for a JSON object. You can create an object projection using the ***** syntax. This will create a list of the values of the JSON object, and project the RHS of the projection onto the list of values.

  ```
  {
    "ops": {
      "functionA": {"numArgs": 2},
      "functionB": {"numArgs": 3},
      "functionC": {"variadic": true}
    }
  }
  ```

  For the preceding JSON content, if the expression is **ops.*.numArgs**, the following result is obtained:

  ```
  [
    2,
    3
  ]
  ```

  The object projection can be divided into two parts. The LHS is **ops** and the RHS is **numArgs**. In the preceding example, ***** creates a JSON array of the values associated with the **ops** JSON object. The RHS of the projection, **numArgs**, is then applied to the JSON array, resulting in the final array of **[2, 3]**.

  The following describes how to project an object:

  i. The LHS is evaluated to create the initial array to be projected:

     evaluate(ops, inputData) -> [{"numArgs": 2}, {"numArgs": 3}, {"variadic": True}]

  ii. Then the RHS is applied to each element in the array:

     evaluate(numArgs, {numArgs: 2}) -> 2

     evaluate(numArgs, {numArgs: 3}) -> 3

     evaluate(numArgs, {variadic: true}) -> null

  iii. Any **null** values are not included in the final result, so the result of the entire expression is **[2, 3]**.

- Flatten projection

More than one projection can be used in a JMESPath expression. In the case of a list/object projection, the original data structure is preserved when a projection is created within another projection.

```
{
  "reservations": [
    {
      "instances": [
        {"state": "running"},
        {"state": "stopped"}
      ]
    },
    {
      "instances": [
        {"state": "terminated"},
        {"state": "running"}
      ]
    }
  ]
}
```

In the preceding JSON content, the expression **reservations[*].instances[*].state** indicates that the value of the top-level key **reservations** is an array. For each element in the **reservations** array, project the **instances[*].state** expression. Within each element in the **reservations** array, there is an **instances** key which itself is a value, and a subprojection is created for each element in the **instances** array. The following result is obtained:

```
[
  [
    "running",
    "stopped"
  ],
  [
    "terminated",
    "running"
  ]
]
```

The result is a nested list. The outer list is the projection of **reservations[*]**, and the inner list is the projection of **state** created from **instances[*]**.

What if you do not care which **reservations** the **instances** belongs to and you want a list of all the **state** values? That is, your expected result is as follows:

```
[
  "running",
  "stopped",
  "terminated",
  "running"
]
```

This is what a flatten projection solves. To get the expected result, you can use **[]** instead of **[*]** to flatten a list, that is, use **reservations[].instances[].state**.

Rules of thumb to use for the flatten operator **[]** are as follows:

- It flattens sublists into the parent list (not recursively, just one level).
- It creates a projection, so anything on the RHS of the flatten projection is projected onto the newly created flattened list.

You can also use **[]** on its own to flatten a list.

```
[
  [0, 1],
  2,
  [3],
  4,
  [5, [6, 7]]
]
```

For the preceding JSON content, if the expression is **[]**, the following result is obtained:

```
[
  0,
  1,
  2,
  3,
  4,
  5,
  [
    6,
    7
  ]
]
```

If you use **[][]** to flatten the result of the expression again, the result of **[0, 1, 2, 3, 4, 5, 6, 7]** is obtained.

– Filter projection

Filter projections allow you to filter the LHS of the projection before evaluating the RHS of a projection.

```
{
  "machines": [
    {"name": "a", "state": "running"},
    {"name": "b", "state": "stopped"},
    {"name": "b", "state": "running"}
  ]
}
```

For the preceding JSON content, if the expression is **machines[?state=='running'].name**, the following result is obtained:

```
[
  "a",
  "b"
]
```

A filter expression is defined for an array and has the general form *LHS[?<Expression><Comparator><Expression>]RHS*. The following comparators are supported: ==, !=, <, <=, >, >=.

- Pipe expression

Projection is an important concept in JMESPath. However, sometimes projection semantics are not what you want. A common scenario is when you want to operate the result of a projection rather than projecting an expression onto each element in the array. For example:

```
{
  "people": [
    {"first": "James", "last": "d"},
    {"first": "Jacob", "last": "e"},
    {"first": "Jayden", "last": "f"},
    {"missing": "different"}
  ],
  "foo": {"bar": "baz"}
}
```

The expression **people[*].first** will give you an array containing the first names of everyone in the **people** array. What if you wanted the first element

in that array? If you tried **people[*].first[0]** that you evaluate **first[0]** for each element in the **people** array, and because indexing is not defined for strings, the final result would be an empty array, **[]**. To obtain the desired result, you can use a pipe expression, *<Expression>* **|** *<expression>*. For the preceding JSON content, if the expression is **people[*].first | [0]**, the result is **James**.

In the pipe expression above, the RHS of the list projection is **first**. When a pipe is encountered, the result up to that point is passed to the RHS of the pipe expression. The pipe expression is evaluated as:

a. evaluate(people[*].first, inputData) -> ["James", "Jacob", "Jayden"]

b. evaluate([0], ["James", "Jacob", "Jayden"]) -> "James"

- MultiSelect

  Multiselect expressions are classified into multiselect lists and multiselect hashes. Multiselect expressions allow you to create elements that do not exist in the JSON data. A multiselect list creates a list and a multiselect hash creates a JSON object.

  – Multiselect list

  ```
  {
    "people": [
      {
        "name": "a",
        "state": {"name": "up"}
      },
      {
        "name": "b",
        "state": {"name": "down"}
      },
      {
        "name": "c",
        "state": {"name": "up"}
      }
    ]
  }
  ```

  For the preceding JSON content, if the expression is **people[].[name,state.name]**, the following result is obtained:

  ```
  [
    [
      "a",
      "up"
    ],
    [
      "b",
      "down"
    ],
    [
      "c",
      "up"
    ]
  ]
  ```

  In the expression above, **[name,state.name]** is a multiselect list. It indicates that a list of two elements is created. The first element is the result of evaluating the **name** expression against the list element, and the second element is the result of evaluating **state.name**. Each list element will therefore create a two-element list, and the final result of the entire expression is a list of two-element lists.

  Unlike a projection, the result of the expression is always included, even if the result is a null. If you change the above expression to **people[].[foo,bar]**, each two-element list will be **[null, null]**.

```
[
  [
    null,
    null
  ],
  [
    null,
    null
  ],
  [
    null,
    null
  ]
]
```

- Multiselect hash

  A multiselect hash has the same basic idea of a multiselect list. The only difference is that a multiselect hash creates a hash instead of an array.

  ```
  {
    "people": [
      {
        "name": "a",
        "state": {"name": "up"}
      },
      {
        "name": "b",
        "state": {"name": "down"}
      },
      {
        "name": "c",
        "state": {"name": "up"}
      }
    ]
  }
  ```

  For the preceding JSON content, if the expression is **people[].{Name:name,State:state.name}**, the following result is obtained:

  ```
  [
    {
      "Name": "a",
      "State": "up"
    },
    {
      "Name": "b",
      "State": "down"
    },
    {
      "Name": "c",
      "State": "up"
    }
  ]
  ```

- Function

  JMESPath supports function expressions, for example:

  ```
  {
    "people": [
      {
        "name": "b",
        "age": 30,
        "state": {"name": "up"}
      },
      {
        "name": "a",
        "age": 50,
        "state": {"name": "down"}
      },
      {
  ```

```
    "name": "c",
    "age": 40,
    "state": {"name": "up"}
   }
  ]
}
```

For the preceding JSON content, if the expression is **length(people)**, the result is **3**.

Functions can be used to transform and filter data in a powerful way. For details about the built-in functions, see **Which Built-in Functions Are Supported by JMESPath?**

The following are some example functions.

This example prints the name of the oldest person in the **people** array:

```
{
  "people": [
    {
      "name": "b",
      "age": 30
    },
    {
      "name": "a",
      "age": 50
    },
    {
      "name": "c",
      "age": 40
    }
  ]
}
```

For the preceding JSON content, if the expression is **max_by(people,&age).name**, the result is **a**.

Functions can also be combined with filter expressions. In the following example, the JMESPath expression finds all elements in **myarray** that contains the string **foo**.

```
{
  "myarray": [
    "foo",
    "foobar",
    "barfoo",
    "bar",
    "baz",
    "barbaz",
    "barfoobaz"
  ]
}
```

For the preceding JSON content, if the expression is **myarray[?contains(@,'foo')==`true`]**, the following result is obtained:

```
[
  "foo",
  "foobar",
  "barfoo",
  "barfoobaz"
]
```

The **@** character in the above example refers to the current element being evaluated in **myarray**. The expression **contains(@, `foo`)** will return **true** if the current element in the **myarray** array contains the string **foo**.

Pay attention to the following when using functions:

- Function arguments have types. If an argument for a function has the wrong type, G42Cloud CLI displays an error message and outputs the

original JSON result. There are functions that can convert arguments (**to_string**, **to_number**) to their proper type.

- The number of function parameters is limited. If a function is called with the wrong number of arguments, G42Cloud CLI displays an error message and outputs the original JSON result.

10.12.2.2 Which Built-in Functions Are Supported by JMESPath?

The built-in functions of JMESPath support the following data types:

- number (integer and double-precision floating-point format in JSON)
- string
- boolean (true or false)
- array (an ordered sequence of values)
- object (an unordered collection of key-value pairs)
- expression (denoted by &expression)
- null

Built-in functions support different data types, as described in the following table. The special character **@** in function arguments indicates that the current result is passed to the function as an input parameter.

Table 10-4 Built-in functions supported by JMESPath expressions

Built-in Function	Input Data Type	Output Data Type	Description	Example
abs	number	number	Returns the absolute value of the provided argument.	• Expression: **abs(1)** Final result: **1** • Expression: **abs(-1)** Final result: **1**
avg	array[number]	number	Returns the average of the elements in the provided array.	Current result: **[10, 15, 20]** Expression: **avg(@)** Final result: **15**
ceil	number	number	Returns the next highest integer value by rounding up if necessary.	Expression: **ceil(`1.001`)** Final result: **2**

Built-in Function	Input Data Type	Output Data Type	Description	Example
contains	array|string, any	boolean	Returns **true** if the first given argument contains the second one, or otherwise returns **false**.	• Expression: contains('foobar','foo') Final result: **true** • Current result: **["a", "b"]** Expression: **contains(@, 'a')** Final result: **true**
ends_with	string, string	boolean	Returns **true** if the first character string ends with the second one, or otherwise returns **false**.	Current result: **foobarbaz** Expression: **ends_with(@,'baz')** Final result: **true**
floor	number	number	Returns the next lowest integer value by rounding down if necessary.	Expression: **floor(`1.001`)** Final result: **1**
join	string, array[string]	string	Returns all of the elements from the provided character string array joined using the given string argument as a separator.	Current result: **["a", "b"]** Expression: **join(',', @)** Final result: **"a, b"**

Built-in Function	Input Data Type	Output Data Type	Description	Example
keys	object	array	Returns an array containing the keys of the provided JSON object. Because JSON hashes are inherited and unordered, the keys associated with the provided object are also inherited and unordered. Implementations are not required to return keys in any specific order.	• Current result: **{"foo": "baz", "bar": "bam"}** Expression: **keys(@)** The final result could be as follows: – ["foo", "bar"] – ["bar", "foo"] • Current result: **{}** Expression: **keys(@)** Final result: **[]**

Built-in Function	Input Data Type	Output Data Type	Description	Example
length	string| array| object	number	Returns the length of the given argument using the following type rules: 1. **string**: Returns the number of characters in the string. 2. **array**: Returns the number of elements in the array. 3. **object**: Returns the number of key-value pairs in the object.	• Current result: **current** Expression: **length(@)** Final result: **7** • Current result: **["a", "b", "c"]** Expression: **length(@)** Final result: **3** • Current result: **{"foo": "bar", "baz": "bam"}** Expression: **length(@)** Final result: **2**

Built-in Function	Input Data Type	Output Data Type	Description	Example
map	expression->any->any, array[any]	array[any]	Applies the expression in the input argument to every element in the array, and returns the array of results. An element of length N will produce a return array of length N. Unlike a projection, **map()** will include the result of applying the expression for every element in the elements array, even if the result is null.	• Current result: **{"array": [{"foo": "a"}, {"foo": "b"}, {}, [], {"foo": "f"}]}** Expression: **map(&foo, array)** Final result: **["a", "b", null, null, "f"]** • Current result: **[[1, 2, 3, [4]], [5, 6, 7, [8, 9]]]** Expression: **map(&[], @)** Final result: **[[1, 2, 3, 4], [5, 6, 7, 8, 9]]**
max	array[number]|array[string]	number	Returns the largest number in the provided array argument.	• Current result: **[10, 15]** Expression: **max(@)** Final result: **15** • Current result: **["abc", "drb"]** Expression: **max(@)** Final result: **drb**

Built-in Function	Input Data Type	Output Data Type	Description	Example
max_by	array, expression->number|expression->string	any	Returns the maximum element in an array by using the expression as a comparison key.	Current result: **[{"name": "b", "age": 30, "age_str": "30"}, {"name": "a", "age": 50, "age_str": "50"}, {"name": "c", "age": 40, "age_str": "40"}]** For the preceding current result: • Expression: **max_by(@, &age)** Final result: **{"age": 50, "age_str": "50", "name": "a"}** • Expression: **max_by(@, &age).age** Final result: **50** • Expression: **max_by(@, &to_number(age_str))** Final result: **{"age": 50, "age_str": "50", "name": "a"}**

Built-in Function	Input Data Type	Output Data Type	Description	Example
merge	[object [, object ...]]	object	Accepts one or more objects as arguments, and returns a single object with subsequent objects merged. The key-value pairs of each subsequent object are added to the preceding object. This function is used to combine multiple objects into one. You can think of this as the first object being the base object, and each subsequent argument being the overrides that are applied to the base object.	• Expression: **merge(`{"a": "b"}`, `{"c": "d"}`)** Final result: **{"a": "b", "c": "d"}** • Expression: **merge(`{"a": "b"}`, `{"a": "override"}`)** Final result: **{"a": "override"}** • Expression: **merge(`{"a": "x", "b": "y"}`, `{"b": "override", "c": "z"}`)** Final result: **{"a": "x", "b": "override", "c": "z"}**
min	array[number]|array[string]	number	Returns the smallest number in the provided array argument.	• Current result: **[10, 15]** Expression: **min(@)** Final result: **10** • Current result: **["a", "b"]** Expression: **min(@)** Final result: **"a"**

Built-in Function	Input Data Type	Output Data Type	Description	Example
min_by	array, expression->number|expression->string	any	Returns the smallest element in an array using the expression as a comparison key.	Current result: **{"people": [{"name": "b", "age": 30, "age_str": "30"}, {"name": "a", "age": 50, "age_str": "50"}, {"name": "c", "age": 40, "age_str": "40"}]}** For the preceding current result: • Expression: **min_by(people, &age)** Final result: **{"age": 30, "age_str": "30", "name": "b"}** • Expression: **min_by(people, &age).age** Final result: **30** • Expression: **min_by(people, &to_number(age_str))** Final result: **{"age": 30, "age_str": "30", "name": "b"}**
not_null	[any [, any ...]]	any	This function accepts one or more arguments, and evaluates them in order until a non-null argument is encountered. If the values of all arguments as resolved as null, G42Cloud CLI displays an error message and outputs the original JSON result.	• Current result: **{"a": null, "b": null, "c": [], "d": "foo"}** Expression: **not_null(no_exist, a, b, c, d)** Final result: **[]** • Current result: **{"a": null, "b": null, "c": [], "d": "foo"}** Expression: **not_null(a, b, `null`, d, c)** Final result: **"foo"** • Current result: **{"a": null, "b": null, "c": [], "d": "foo"}** Expression: **not_null(a, b)** Final result: **null**

Built-in Function	Input Data Type	Output Data Type	Description	Example
reverse	string|array	string|array	Reverses the order of the input argument.	• Current result: **[0, 1, 2, 3, 4]** Expression: **reverse(@)** Final result: **[4, 3, 2, 1, 0]** • Current result: **[]** Expression: **reverse(@)** Final result: **[]** • Current result: **["a", "b", "c"]** Expression: **reverse(@)** Final result: **["c", "b", "a"]** • Current result: **"abcd"** Expression: **reverse(@)** Final result: **"dcba"**
sort	array[number]|array[string]	array	Accepts an array argument and returns the sorted elements as an array. The array must be a list of strings or numbers. Strings are sorted based on a dictionary.	• Current result: **["b", "a", "c"]** Expression: **sort(@)** Final result: **["a", "b", "c"]** • Current result: **[1, 4, 2]** Expression: **sort(@)** Final result: **[1, 2, 4]**

Built-in Function	Input Data Type	Output Data Type	Description	Example
sort_by	array, expression->number|expression->string	-	Sorts an array using an expression as the sort key. For each element in the array of elements, the expression is applied and the result value is used as the key for sorting the elements. sort_by follows the same sorting logic as the sort function.	Current result: **{"people": [{"name": "b", "age": 30, "age_str": "30"}, {"name": "a", "age": 50, "age_str": "50"}, {"name": "c", "age": 40, "age_str": "40"}]}** For the preceding current result: • Expression: **sort_by(people, &age)[].age** Final result: **[30, 40, 50]** • Expression: **sort_by(people, &age)[0]** Final result: **{"age": 30, "age_str": "30", "name": "b"}** • Expression: **sort_by(people, &to_number(age_str))[1]** Final result: **{"age": 40, "age_str": "40", "name": c}**
starts_with	string, string	boolean	Returns **true** if the first argument starts with the second one, or otherwise returns **false**.	• Current result: **foobarbaz** Expression: **starts_with(@, 'foo')** Final result: **true** • Current result: **foobarbaz** Expression: **starts_with(@,'baz')** Final result: **false** • Current result: **foobarbaz** Expression: **starts_with(@, 'f')** Final result: **true**
sum	array[number]	number	Returns the sum of the provided array argument. An empty array will produce a return value of **0**.	• Current result: **[10, 15]** Expression: **sum(@)** Final result: **25** • Current result: **[]** Expression: **sum(@)** Final result: **0**

Built-in Function	Input Data Type	Output Data Type	Description	Example
to_array	any	array	**array**: Returns the passed value. **number|string|object|boolean**: Returns a one-element array containing the passed argument.	• Expression: **to_array(`[1, 2]`)** Final result: **[1, 2]** • Expression: **to_array('string')** Final result: **["string"]** • Expression: **to_array(`0`)** Final result: **[0]** • Expression: **to_array(`true`)** Final result: **[true]** • Expression: **to_array(`{"foo": "bar"}`)** Final result: **[{"foo": "bar"}]**
to_string	any	string	**string**: Returns the passed value. **number|array|object|boolean**: The JSON encoded value of the object.	• Expression: **to_string(`2`)** Final result: **2** • Expression: **to_string(`[]`)** Final result: **"[]"** • Expression: **to_string(false)** Final result: **"null"**
to_number	any	number	**string**: Returns the parsed number. **number**: Returns the passed value. **array|object|boolean|null**: G42Cloud CLI displays an error message and outputs the original JSON result.	• Expression: **to_number(`2.3`)** Final result: **2.3** • Expression: **to_number(`2`)** Final result: **2**

Built-in Function	Input Data Type	Output Data Type	Description	Example
type	array| object| string| number| boolean| null	string	Returns the data type of the given argument as a string value. The return value must be one of the following: • "number" • "string" • "boolean" • "array" • "object" • "null"	• Expression: **type('foo')** Final result: **"string"** • Expression: **type(`true`)** Final result: **"boolean"** • Expression: **type(`null`)** Final result: **"null"** • Expression: **type(`123`)** Final result: **number** • Expression: **type(`123.05`)** Final result: **number** • Expression: **type(`[1,2]`)** Final result: **"array"** • Current result: **{"abc": "123"}** Expression: **type(@)** Final result: **"object"**
values	object	array	Returns an array of values of the provided JSON object. Because JSON hashes are inherited and unordered, the values associated with the provided object are also inherited and unordered. Implementations are not required to return values of the JSON object in any specific order.	Current result: **{"a": "first", "b": "second", "c": "third"}** Expression: **values(@)** The final result could be as follows: • ["first", "second", "third"] • ["first", "third", "second"] • ["second", "first", "third"] • ["second", "third", "first"] • ["third", "first", "second"] • ["third", "second", "first"]

10.12.3 Which G42Cloud CLI System Parameters Are Related to Data Output? Which Ones Are Recommended?

The following table lists the G42Cloud CLI system parameters related to data output.

Table 10-5 G42Cloud CLI system parameters related to data output

Category	Parameter	Description
New output parameters	cli-output, cli-query, cli-output-num	• cli-output Response data output format. The options are as follows: – json – table – tsv • cli-query JMESPath for filtering response data. • cli-output-num Whether to print the row numbers during table output. The value can be **true** or **false**.
Old output parameters	cli-output-rows, cli-output-cols, cli-output-num, cli-json-filter	• cli-output-rows Levels to print during table output. • cli-output-cols Fields to print during table output. • cli-output-num Whether to print the row numbers during table output. The value can be **true** or **false**. • cli-json-filter Performs a JMESPath query on the output JSON result.

Compared with the old output parameters, the new output parameters support TSV format in addition to table and JSON and are unified to facilitate user operations.

Functions related to output formats will be continuously optimized based on the new output parameters. The old output parameters can still be used but will not

be upgraded. **You are advised to use the new output parameters** when constructing commands.

10.12.4 How Do I Use cli-output, cli-query, and cli-output-num?

For details about how to use the new output parameters, see **Specifying Output Format**.

In a command, use **--cli-query** to pass a **JMESPath** expression and perform a **JMESPath** query on the result to extract key information in the original returned result. Use **--cli-output** to specify the response data output format. Use **--cli-output-num** to specify whether to print the row numbers during table output.

When you use the preceding parameters, note that:

- Use only **--cli-output** in a command to specify the output format. If you use only **--cli-query** in a command, the default output format is JSON.
- When using **--cli-query**, enclose the value with double quotation marks ("") to avoid parsing errors when the system processes commands.
- When using **--cli-output-num** to specify whether to print the row numbers, set **--cli-output** to **table**.
- In the same command, if **--cli-output** has been used and old system parameters such as **--cli-output-rows** and **--cli-json-filter** are specified, the value of **--cli-output** is preferentially used as the target output format.

10.12.5 How Do I Use cli-output-rows, cli-output-cols, and cli-output-num? What Are the Precautions?

10.12.5.1 How Do I Use cli-output-rows, cli-output-cols, and cli-output-num?

By default, results are returned in JSON format when you call cloud service APIs using G42Cloud CLI. G42Cloud CLI supports the **--cli-output-rows**, **--cli-output-cols**, and **--cli-output-num** parameters to output data in table format. These parameters facilitate the extraction of key information in the calling result.

By default, the original calling result is output in JSON format:

```
g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="0dd8cb***************19b5a84546"
{
  "servers": [
    {
      "name": "ecs-a6b4",
      "links": [
        {
          "rel": "self",
          "href": "https://ecs.ae-ad-1.g42cloud.com/v2.1/0dd8cb***************19b5a84546/servers/4f06****-****-****-****-****04dd856a"
        },
        {
          "rel": "bookmark",
          "href": "https://ecs.ae-ad-1.g42cloud.com/0dd8cb***************19b5a84546/servers/4f06****-****-****-****-****04dd856a"
        }
      ],
      "id": "4f06****-****-****-****-****04dd856a"
    },
```

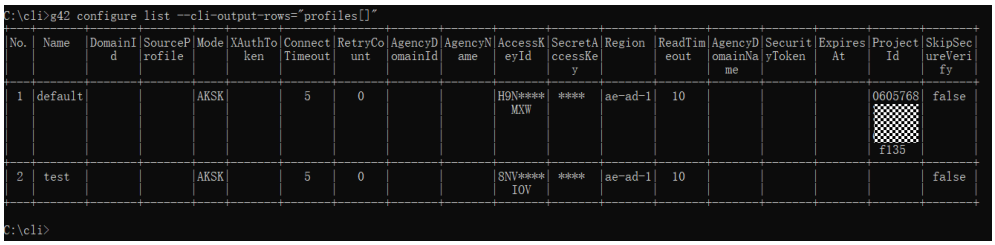
```
  {
    "name": "hdn-docker",
    "links": [
      {
        "rel": "self",
        "href": "https://ecs.ae-ad-1.g42cloud.com/v2.1/0dd8cb***************19b5a84546/servers/6731****-****-****-****-****0bc463f0"
      },
      {
        "rel": "bookmark",
        "href": "https://ecs.ae-ad-1.g42cloud.com/0dd8cb***************19b5a84546/servers/6731****-****-****-****-****0bc463f0"
      }
    ],
    "id": "6731****-****-****-****-****0bc463f0"
  },
  {
    "name": "ecs-8f88",
    "links": [
      {
        "rel": "self",
        "href": "https://ecs.ae-ad-1.g42cloud.com/v2.1/0dd8cb***************19b5a84546/servers/06a2****-****-****-****-****c79a1a26"
      },
      {
        "rel": "bookmark",
        "href": "https://ecs.ae-ad-1.g42cloud.com/0dd8cb***************19b5a84546/servers/06a2****-****-****-****-****c79a1a26"
      }
    ],
    "id": "06a2****-****-****-****-****c79a1a26"
  }
 ]
}
```


During table output, **--cli-output-rows** specifies a JSON structure level, that is, the data source of the table, **--cli-output-cols** specifies the column names of the table, which must correspond to the fields in the JSON structure, and **--cli-output-num** specifies whether to print the row numbers of the table (the default value is **true**).

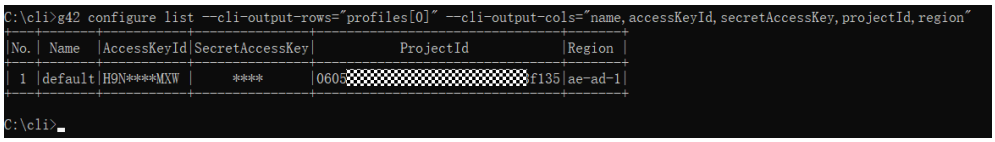
g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="0dd8cb*****************b5a84546" --cli-output-rows="servers" --cli-output-cols="name,id"

The **--cli-output-rows**, **--cli-output-cols**, and **--cli-output-num** parameters can also be used in system commands. For example:

g42 configure list --cli-output-rows="profiles[]"



g42 configure list --cli-output-rows="profiles[0]" --cli-output-cols="name,accessKeyId,secretAccessKey,projectId,region"



For details about how to use **--cli-output-rows**, **--cli-output-cols**, and **--cli-output-num**, see **What Are the Precautions for Using cli-output-rows, cli-output-cols, and cli-output-num?**

10.12.5.2 What Are the Precautions for Using cli-output-rows, cli-output-cols, and cli-output-num?

If **--cli-output-rows**, **--cli-output-cols**, and **--cli-output-num** are used in a command, the command output is in table format. Table output helps you extract key information from return values. The parameters are described as follows:

- **--cli-output-cols**: the fields to print during table output.
- **--cli-output-rows**: the levels to print during table output. For example, if you want to convert a JSON structure to a table, set this parameter to the name of the structure.
- **--cli-output-num**: whether to print the row numbers during table output. The value can be **true** (default) or **false**.

For details, see **How Do I Use cli-output-rows, cli-output-cols, and cli-output-num?**

Note the following when using the preceding parameters for table output:

- **--cli-output-cols** and **--cli-output-rows** can be used separately or together.
  - Use only **--cli-output-rows**.

    The **--cli-output-rows** option is separately used in a command to pass the name of a JSON structure in the calling result. The levels must be separated by periods (.). The content of the target JSON structure must be an array. G42Cloud CLI will display the structure content in a table. For example, if you run the **g42 configure list --cli-output-rows=profiles** command, all profile information will be displayed in a table. If the JSON structure specified in **--cli-output-rows** is not an array, the following error message is displayed:

    [CLI_ERROR] Table output error. The cli-output-cols parameter is required.
  - Use only **--cli-output-cols**.

    The **--cli-output-cols** option is used to pass the root element fields of the JSON structure of the calling result. The fields must be separated by

commas (,). For example, if you run the **g42 configure show --cli-profile=*${profileName}* --cli-output-cols=accessKeyId** command, the access key ID in the specified profile will be displayed in a table. When **--cli-output-cols** is used separately, only the root element fields of the JSON structure can be specified. Otherwise, the following error message is displayed:

[USE_ERROR] In parameter cli-output-cols, * is null.

- Use both **--cli-output-cols** and **--cli-output-rows**.

  The **--cli-output-rows** option is used to specify the levels to print, and the **--cli-output-cols** option is used to specify the fields to print at the levels. For example, if you run the **g42 configure list --cli-output-rows=profiles --cli-output-cols=accessKeyId** command, the access key IDs of all profiles will be displayed in a table.

  You can set **--cli-output-rows** to **[n]** or **[m:n]** to specify the index of the array element to be printed. Specify **[n]** to print the value of index n, and specify **[m:n]** to print the values of indexes m to (n – 1). For example, if you run the **g42 configure list --cli-output-rows=profiles[0:2] --cli-output-cols=accessKeyId** command, the access key IDs of the profiles whose indexes are 0 and 1 in the profile array are displayed in a table. Pay attention to the following:

  - If the index value of the array in **--cli-output-rows** is **[m:n]** and the value of **n** exceeds the array length limit, data within the maximum index will be printed.
  - If the index value of the array in **--cli-output-rows** is **[n]** and the value of **n** exceeds the array length limit, an error message is displayed indicating that the array index is out of range:

    [USE_ERROR] The cli-output-rows parameter contains an incorrect field (*). The array index is out of range and the array length is *. Current index: *.
  - When both **--cli-output-cols** and **--cli-output-rows** are used, the parameters in **--cli-output-rows** do not need to be of the array type. You only need to specify the level.

- When **--cli-output-num** is used separately, the output will not be displayed in a table.
- In the same command, **--cli-output-rows**, **--cli-output-cols**, and **--cli-output-num** cannot be used together with **--cli-json-filter**. Otherwise, an error occurs because the output format cannot be determined.

10.12.6 How Do I Use cli-json-filter? What Are the Precautions?

10.12.6.1 How Do I Use cli-json-filter?

By default, results are returned in JSON format when you call cloud service APIs using G42Cloud CLI. G42Cloud CLI supports the **--cli-json-filter** parameter to perform JMESPath query on JSON results. This option facilitates the extraction of key information in the results. For example:

By default, the original calling result is output in JSON format:

```
g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="0dd8cb***************19b5a84546"
{
  "servers": [
    {
      "name": "ecs-a6b4",
      "links": [
        {
          "rel": "self",
          "href": "https://ecs.ae-ad-1.g42cloud.com/v2.1/0dd8cb***************19b5a84546/servers/4f06****-****-
****-****-****04dd856a"
        },
        {
          "rel": "bookmark",
          "href": "https://ecs.ae-ad-1.g42cloud.com/0dd8cb***************19b5a84546/servers/4f06****-****-****-
****-****04dd856a"
        }
      ],
      "id": "4f06****-****-****-****-****04dd856a"
    },
    {
      "name": "hdn-docker",
      "links": [
        {
          "rel": "self",
          "href": "https://ecs.ae-ad-1.g42cloud.com/v2.1/0dd8cb***************19b5a84546/servers/6731****-****-
****-****-****0bc463f0"
        },
        {
          "rel": "bookmark",
          "href": "https://ecs.ae-ad-1.g42cloud.com/0dd8cb***************19b5a84546/servers/6731****-****-****-
****-****0bc463f0"
        }
      ],
      "id": "6731****-****-****-****-****0bc463f0"
    },
    {
      "name": "ecs-8f88",
      "links": [
        {
          "rel": "self",
          "href": "https://ecs.ae-ad-1.g42cloud.com/v2.1/0dd8cb***************19b5a84546/servers/06a2****-****-
****-****-****c79a1a26"
        },
        {
          "rel": "bookmark",
          "href": "https://ecs.ae-ad-1.g42cloud.com/0dd8cb***************19b5a84546/servers/06a2****-****-****-
****-****c79a1a26"
        }
      ],
      "id": "06a2****-****-****-****-****c79a1a26"
    }
  ]
}
```

Use **--cli-json-filter** to perform JMESPath query on the original JSON result, obtain the **id** and **name** of each **servers** element, and rename them **EcsID** and **EcsName**, respectively. The following is an example:

```
g42 ECS NovaListServers --cli-region="ae-ad-1" --project_id="0dd8cb***************19b5a84546" --cli-json-
filter="servers[].{EcsID:id,EcsName:name}"
[
  {
    "EcsID": "4f06****-****-****-****-****04dd856a",
    "EcsName": "ecs-a6b4"
  },
  {
    "EcsID": "6731****-****-****-****-****0bc463f0",
    "EcsName": "hdn-docker"
```

```
  },
  {
    "EcsID": "06a2****-****-****-****-****c79a1a26",
    "EcsName": "ecs-8f88"
  }
]
```

--cli-json-filter can also be used in system commands. For example, to query all **custom** parameters of the profile **test**, run the following commands:

```
g42 configure list --cli-custom=true --cli-json-filter="profiles[?name=='test'].custom"
[
  {
    "password": {
      "isEncrypted": true,
      "value": "****"
    },
    "projectId": {
      "isEncrypted": false,
      "value": "0681000000000000000000000f89d2e"
    }
  }
]
```

For more information about how to use **--cli-json-filter**, see **What Are the Precautions for Using cli-json-filter?**

10.12.6.2 What Are the Precautions for Using cli-json-filter?

G42Cloud CLI supports the **--cli-json-filter** option to pass a JMESPath expression and perform a JMESPath query on the JSON result. This option facilitates the extraction of key information in the results. Note the following when using **--cli-json-filter**:

- If **--cli-json-filter** is used in the command, the calling result will be output in JSON format.
- In the same command, **--cli-json-filter** cannot be used together with **--cli-output-rows**, **--cli-output-cols**, or **--cli-output-num**. Otherwise, an error occurs because the output format cannot be determined.

For details about using **cli-json-filter** to define a JMESPath expression, see the following:

- **How Do I Use JMESPath Expressions?**
- **Which Built-in Functions Are Supported by JMESPath?**

10.13 Other

10.13.1 How Do I Use G42Cloud CLI in Non-configuration Mode?

When using G42Cloud CLI in non-configuration mode, you do not need to pass your authentication information through a profile. Instead, directly pass your authentication parameters in commands. This mode enables you to use the CLI conveniently without adding any profiles. For details, see **Using G42Cloud CLI in Non-configuration Mode**.

Note the following when using the G42Cloud CLI in this mode:

- Using an AK/SK in non-configuration mode
  - Access key (permanent AK/SK)
    - When cloud service APIs are called using a permanent AK/SK, pass both the access key ID (**cli-access-key**) and secret access key (**cli-secret-key**) in commands for authentication.
    - If a global service is to be accessed, the ID (**cli-domain-id**) of the account used to create the IAM user is also required for authentication. If the ID is not passed to the command, G42Cloud CLI automatically obtains it based on the user authentication information. However, if the **cli-access-key** or **cli-secret-key** parameter is missing or the **cli-domain-id** parameter fails to be automatically obtained, the following error message is displayed:
      - [USE_ERROR] Parameters cli-access-key,cli-secret-key must be specified at the same time.
      - [USE_ERROR] cli-domain-id is required for access to global services using AK/SK. Add this parameter or run `g42 configure set` to configure it.
    - If both an AK/SK and a profile (**cli-profile**) are specified in a command, the AK/SK is preferentially used for authentication.
  - Temporary security credentials (temporary AK/SK and SecurityToken)
    - Using a temporary AK/SK and SecurityToken to call cloud service APIs through G42Cloud CLI is similar to using a permanent AK/SK. If both an AK/SK (**cli-access-key**/**cli-secret-key**) and SecurityToken (**cli-security-token**) are passed in the command, the AK/SK is considered temporary.
    - If a global service is to be accessed, the ID (**cli-domain-id**) of the account used to create the IAM user is also required for authentication. If the ID is not passed to the command, G42Cloud CLI automatically obtains it based on the user authentication information. However, if the **cli-access-key** or **cli-secret-key** parameter is missing or the **cli-domain-id** parameter fails to be automatically obtained, the following error message is displayed:
      - [USE_ERROR] Parameters cli-access-key,cli-secret-key must be specified at the same time.
      - [USE_ERROR] cli-domain-id is required for access to global services using AK/SK. Add this parameter or run `g42 configure set` to configure it.
    - If a temporary AK/SK, SecurityToken, and profile (**cli-profile**) are specified in a command, the AK/SK and SecurityToken are preferentially used for authentication.
- Using an account in non-configuration mode
  - When cloud service APIs are called using a username and password, enter the username (**cli-username**), password (**cli-password**), and account name (**cli-domain**) of the IAM user in commands for authentication. Otherwise, the following error message is displayed:

[USE_ERROR]Parameters cli-username,cli-password,cli-domain must be specified at the same time.

  - If both a username/password and a profile (**cli-profile**) are specified in a command, the username and password are preferentially used for authentication.

- Using a token in non-configuration mode
  - When you use a token to call a cloud service API with a command prompt tool (such as **cmd.exe**), the command line may be incomplete if the value of **cli-x-auth-token** is too long. Therefore, check whether the command line content is complete before executing the command.
  - If both **cli-x-auth-token** and **cli-profile** are specified in a command, **cli-x-auth-token** is preferentially used for authentication.
- Using an ECS agency in non-configuration mode
  - This authentication mode applies only when you use G42Cloud CLI on an ECS.
  - To use this mode, create a cloud service agency to delegate ECS to use the CLI on the IAM console, and add the agency in the **Management Information** > **Agency** area of the ECS details page. For details, see **Cloud Service Delegation**.
- Using an agency in non-configuration mode
  - In this authentication mode, the AK/SK, account, or token of the delegated party can be used for authentication.
  - To use this authentication mode, the delegating party must create an agency for the delegated party. If you are the delegating party, create an agency on the IAM console by referring to **Account Delegation**.

In addition to the preceding precautions, avoid using AK/SK, account, and token authentication parameters interchangeably in the same command. Otherwise, the following error message is displayed:

[USE_ERROR] Specify authentication parameters for only one of [AKSK|Account|Token] in the command.

In the preceding error message, **AKSK** indicates that the command contains **cli-access-key**, **cli-secret-key**, and **cli-security-token**; **Account** indicates that the command contains **cli-username**, **cli-password**, and **cli-domain**; **Token** indicates that the command contains **cli-x-auth-token**.

10.13.2 Should I Enclose a Service Name, Operation, and Parameter Value in Quotation Marks in a Command?

This depends on the value.

Generally, the service name, operation, and parameter value in a command do not need to be enclosed in double quotation marks. If a value contains special characters, spaces, or symbols that need to be escaped, use double quotation marks to enclose the value.

Obtain CLI examples on API Explorer rather than manually entering parameters in commands.

10.13.3 What Are the Application Scenarios of Online/Offline Modes?

- Viewing/Switching the current mode

  G42Cloud CLI can stay in online/offline modes. By default, it is online. After configuring a profile, you can run **g42 configure list --cli-query=offline** to check whether the offline mode is used.

  - To switch to offline mode, run the **g42 configure set --cli-offline=true** command.
  - To switch to online mode, run the **g42 configure set --cli-offline=false** command.

- Offline mode scenario

  Download the latest offline metadata of G42Cloud CLI to a local directory. The metadata cache files have unlimited validity. When you run G42Cloud CLI commands, the file content will be read for command verification and parsing. In this mode, local metadata cache files will not be automatically updated, and parameter verification for existing commands will not be affected. This ensures that the built G42Cloud CLI commands are always available. Use this mode if you build scripts with G42Cloud CLI commands and periodically execute the scripts to manage cloud services and resources.

- Online mode scenario

  G42Cloud CLI obtains metadata and caches it locally during command execution. The metadata cache files have limited validity. If a file has expired, G42Cloud CLI updates it and then uses it for command verification and parsing. In this mode, only the metadata related to your executed commands is saved, and you can call APIs of new cloud services through G42Cloud CLI. Use this mode if you need to run any G42Cloud CLI commands immediately to manage cloud services and resources.

10.13.4 How Do I Uninstall G42Cloud CLI?

G42Cloud CLI can be used without installation. To uninstall the CLI, delete it and the related local cache files by performing the following steps:

1. Run the **g42 auto-complete off** command to turn off autocomplete.
2. Delete the cache files, configuration files, and log files.
   - Linux: **/home/***{Current username}***/.g42/**
   - Windows: **C:\Users***{Current username}***\.g42**
   - macOS: **/Users/***{Current username}***/.g42/**
3. Delete G42Cloud CLI.

11 Change History

Date	Description
2023-04-30	This issue incorporates the following change: New feature: In some special scenarios such as command execution with automation scripts, you can run **g42 configure set --cli-agree-privacy-statement=true** to **agree to the privacy statement**.
2023-01-30	This issue incorporates the following change: Added support for **custom domain names**.
2022-11-30	This issue incorporates the following changes: New feature: OBS's obsutil tool is integrated. You can run the **g42 obs** command to **manage data in OBS**. Experience optimization: You can **leave the body parameters of cloud service APIs empty at any level**.
2022-09-30	This issue incorporates the following change: Added the description of configuring a permanent environment variable in section **Configuring the HTTP Proxy**.
2022-08-30	This issue incorporates the following change: Modified some examples in **Options**.
2022-06-30	This issue is the first official release.