Elastic Compute Service

Best Practices

Best Practices

Security

Modify the default remote access port

Modify the default remote desktop port of a Windows instance

The default remote desktop port of a Windows instance is TCP Port 3389. You can follow these steps to modify the port:

Connect to the Windows instance.

Select **Start** > **All Programs** > **Accessories** > **Run** to open the **Run** dialog box, and run regedit.exe to open **Registry Editor**.

On the left-side navigation pane of the **Registry Editor**, find HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp, and then double click PortNumber.

In the **Eidt DWORD (32-bit) Value** dialog box, select **Decimal** as **Base**, and then type a number in the **Value data** field as the new remote port number, which is 3399 in this example. Click **OK**.

(Optional) If you have enabled firewall, open the new port on the firewall.

Log on to the **ECS console**, find the instance, and then select **More** > **Restart**.

After the instance is restarted, click the instance ID to go to the **Instance Details** page, and then click **Security Groups**.

Add a new security group rule to allow access to the new remote port. For more information about adding security group rules, see Add a security group rule.

Connect to the instance by accessing the IP address ending with the new port number. For example, 192.168.1.2:3399 in this example.

Note: Only the default port 3389 can be used for access by Mac remote desktop users.

Modify the default SSH port of a Linux instance

For a Linux instance, the default SSH port is 22. This section describes how to modify the SSH port of a Linux instance running CentOS 6.8.

Connect to the Linux instance.

Run vim /etc/ssh/sshd_config to edit the file:

i. Press the i key on the keyboard to enter the Edit mode.
ii. Add Port [new port number] (For example, Port 1022) under the line Port 22.
iii. Press the Esc key, and then type :wq to save and exit the file.

Run the following command to restart the sshd.

```
/etc/init.d/sshd restart
```

(Optional) Configure firewall to open the new port for Internet access.

Log on to the ECS console, find the instance, and then select **More** > **Restart**.

After the instance is restarted, click the instance ID to go to the **Instance Details** page, and then click **Security Groups**.

Add a new security group rule to allow access to the new remote port. For more information about adding security group rules, see Add a security group rule.

Connect to the instance by accessing the IP address ending with the new port number. For

example, 192.168.1.2:1022 in this example.

Run vim /etc/ssh/sshd_config again to delete the Port 22 line.

Run /etc/init.d/sshd restart to restart sshd.

> **Note**: We recommend that you do not modify the Port 22 line directly. Then if you cannot connect to the Linux instance on the new port, you can use Port 22 for debugging.

Use logs in Windows instances

Logs are records of hardware and software in the system, and system error information. They can also be used to monitor system events. When a server intrusion or system (application) error occurs, administrators can quickly locate the problems by using logs and solve the problems quickly, which improves work efficiency and server security substantially. Windows logs can be mainly divided into four categories: system logs, application logs, security logs, and applications and services logs. In this example, we use Windows Server 2008 R2 to introduce the use and analysis of the four categories of logs.

Open the Event Viewer

Follow these steps to open Event Viewer: Open the **Run** window, type eventvwr, and then click **OK** to open the **Event Viewer**.

Then, you can view the following four categories of logs in **Event Viewer**.

> **Note**: You can find the solutions to any error event ID that you can find in these logs in Microsoft knowledge base.

System logs

System logs include events recorded by Windows system components. For example, system logs record failures that occur when loading drivers or other system components during startup.

The types of events recorded by system components are predetermined by Windows.

Application logs

Application logs include events recorded by applications or programs. For example, a database application can record file errors in application logs.

The types of events recorded are determined by developers.

Security logs

Security logs include events such as valid and invalid logon attempts, and resource usage related events such as creation, opening, or deletion of files or other objects.

Administrators can specify the types of events recorded in security logs. For example, if logon has been set to be audited, logon attempts are recorded in security logs.

Applications and services logs

Applications and services logs are a new type of event logs. These logs store events from a single application or component, rather than events that may affect the global system.

Modify log path and back up logs

Logs are stored on the system disk by default. The maximum log size is 20 MB by default, and the earliest events are overwritten when 20 MB is exceeded. You can modify the maximum log size according to your needs.

Follow these steps to modify the log path and back up logs:

In the left-side navigation pane of **Event Viewer**, click **Windows Logs**.

Right click a log name, such as **Application** and click **Properties**.

In the **Log Properties** dialog box, you can modify the following settings:

- Log path
- Maximum log size
- Operations executed when maximum event log size is reached

Data recovery

Data restoration in Linux instances

When solving problems related to disks, you may frequently encounter the loss of data disk partitions. This article describes common data partition loss problems and corresponding solutions in Linux, and provides common mistakes and best practices for cloud disks to avoid possible risks of data loss.

Prerequisites

Before restoring data, you must create snapshots for data disks that lose partitions. If problems occur during the restoration process, you can roll back data disks to the status before restoration.

Introduction to disk management tools

You can select one of the following tools to fix the disk partition and restore the data in a Linux instance:

fdisk
The default partitioning tool installed in Linux instances.

testdisk
It is primarily used to restore disk partitions or data in the Linux system. The tool is not installed by default in Linux. You must install it on your own. For example, in a CentOS system, you can run the yum install -y testdisk command to install it online.

partprobe
This is the default tool installed in the Linux system. It is primarily used to enable the kernel to re-read the partition without restarting the system.

Handle data disk partition loss and data restoration in Linux

After you restart a Linux instance, you may encounter data disk partition loss or data loss issues. This may be because you have not set the partitions to be mounted automatically on startup of the instance in the etc/fstab file. In this case, you can manually mount the data disk partition first. If the

system prompts partition table loss when you manually mount the data disk, you can try to solve the problem through the following three methods.

Restore partitions by using fdisk

Restore partitions by using testdisk

Restore data by using testdisk

Restore partitions by using fdisk

Default values usually apply to the starting and ending sectors of the partition when you partition a data disk. You can then directly use fdisk to restore the partition. For more information about this tool, see Linux _ Format and mount a data disk.

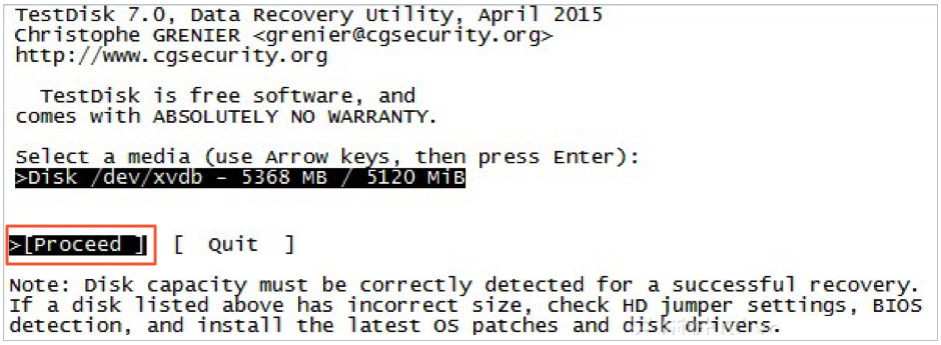
```
[root@Aliyun ~]# fdisk /dev/xvdb
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-10485759, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-10485759, default 10485759):
Using default value 10485759
Partition 1 of type Linux and of size 5 GiB is set

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@Aliyun ~]# mount /dev/xvd
xvda   xvda1  xvdb   xvdb1
[root@Aliyun ~]# mount /dev/xvdb
xvdb   xvdb1
[root@Aliyun ~]# mount /dev/xvdb1 /mnt/
[root@Aliyun ~]# ls /mnt/
123.sh  configclient  data  diamond  install_edsd.sh  install.sh  ip.qz
```

If the preceding operations do not help, you can try testdisk for the restoration.

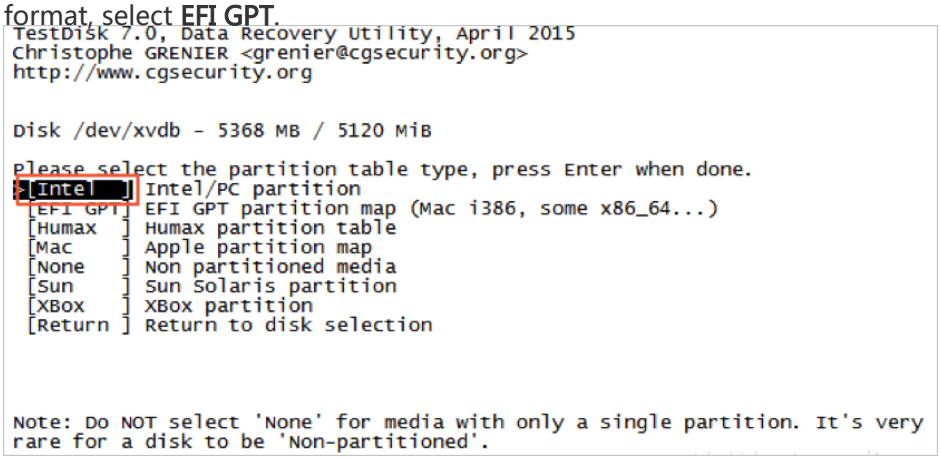
Restore partitions by using testdisk

Here we suppose the cloud disk device is named /dev/xvdb. Follow these steps to restore the partitions by using testdisk:

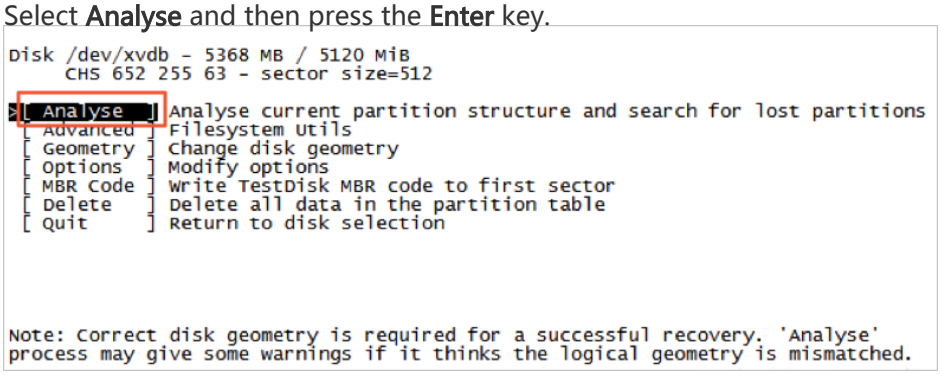
Run testdisk /dev/xvdb #Replace the device name as appropriate, and then select Proceed (default value) and press the Enter key.



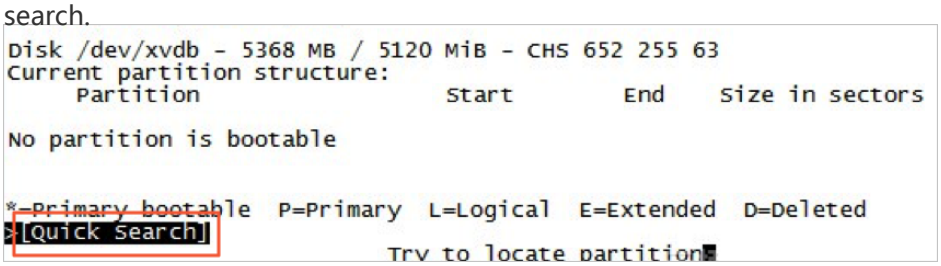
Select the partition table type for scanning: **Intel** by default. . If your data disk uses the GPT format, select **EFI GPT**.



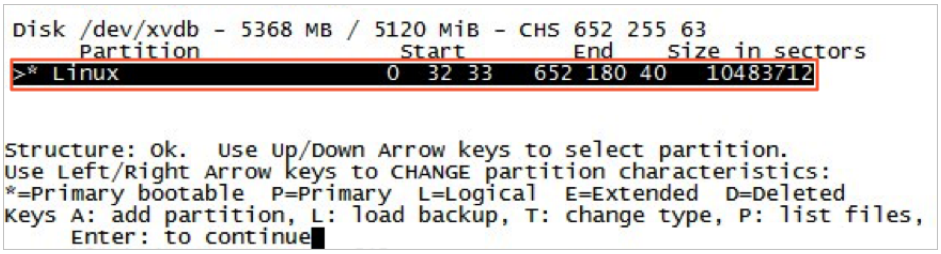
Select **Analyse** and then press the **Enter** key.



If you cannot see any partition, select **Quick Search** and then press the **Enter** key for a quick search.



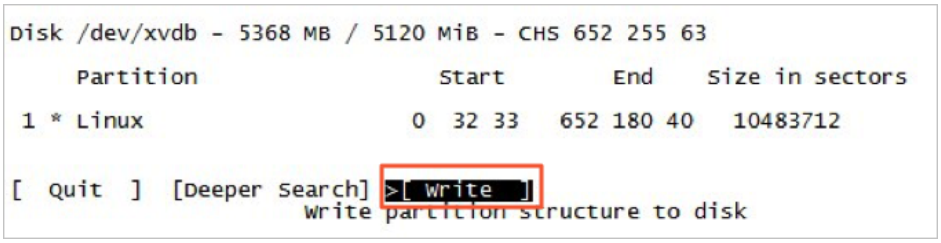
The partition information is displayed in the returned result, as shown in the following figure.



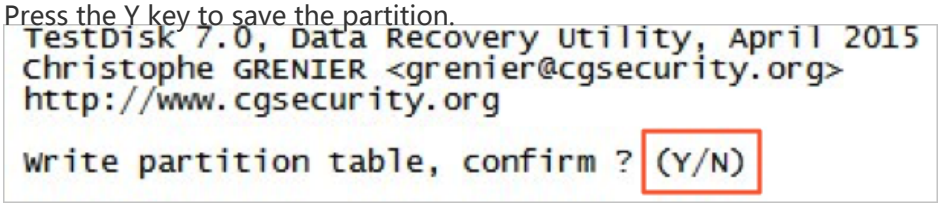
Select the partition and press the **Enter** key.

Select **Write** to save the partition.

> **Note**: Select **Deeper Search** to continue searching if the expected partition is not listed.

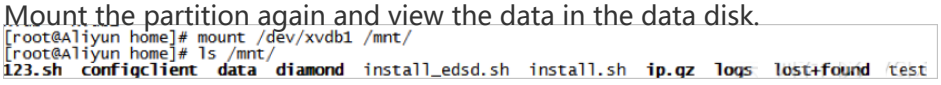


Press the Y key to save the partition.



Run partprobe /dev/xvdb #Replace the device name as appropriate to refresh the partition table manually.

Mount the partition again and view the data in the data disk.



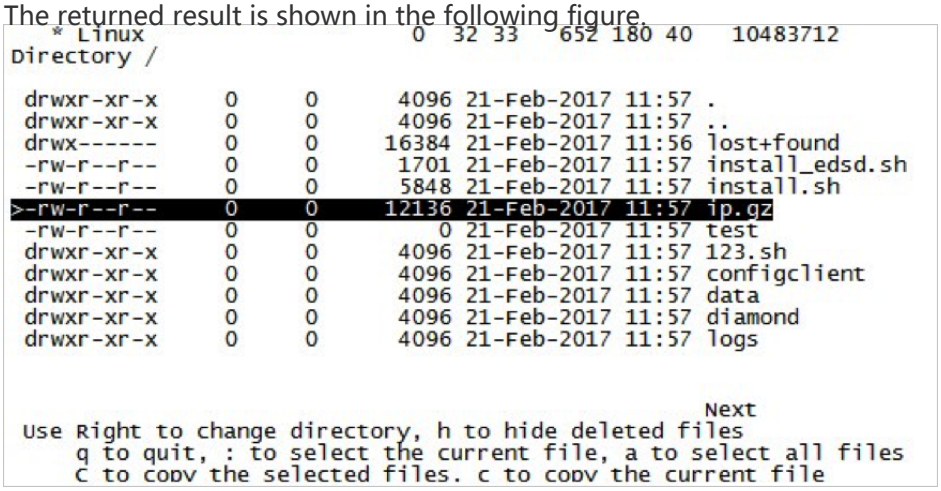
Restore data by using testdisk

In some cases, you can use testdisk to scan and locate the disk partition, but you cannot save the partition. In this case, you can try to restore files directly. Follow these steps:

Find the partition following Step 1 to Step 4 described in Restore partitions by using teskdisk.

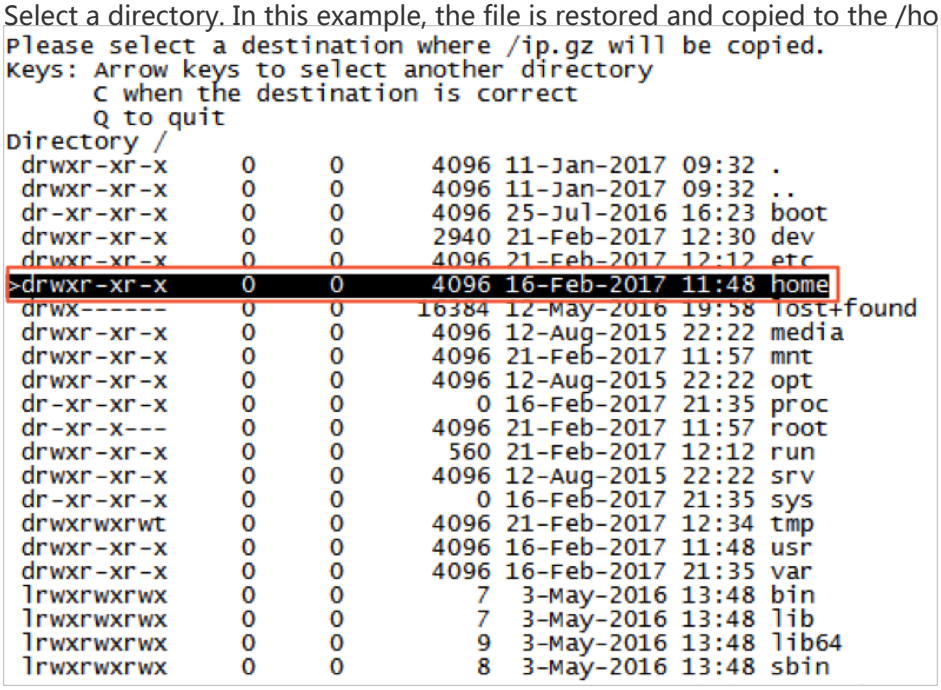
List files by pressing the P key.

The returned result is shown in the following figure.

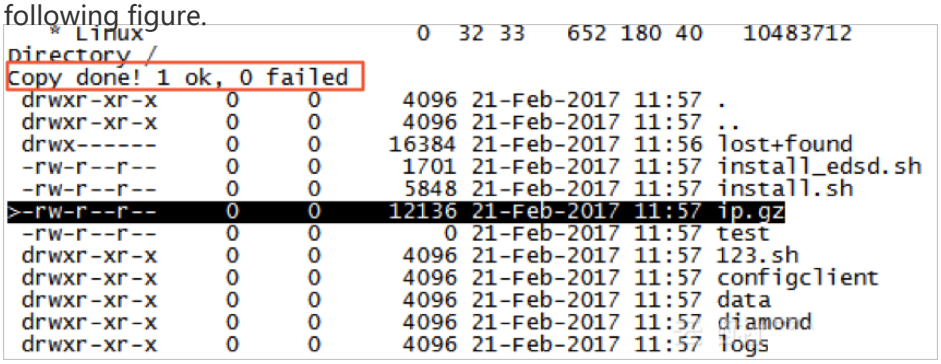


Select the files to restore, and press the C key.

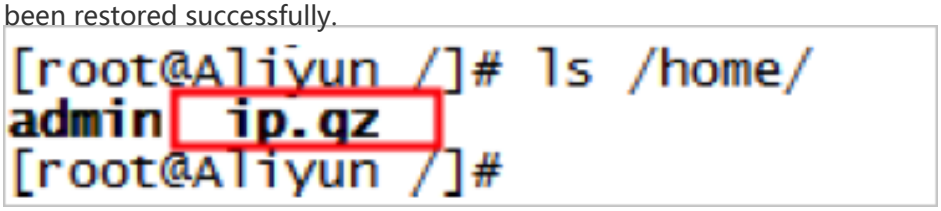
Select a directory. In this example, the file is restored and copied to the /home directory.



If you see Copy done! 1 ok, 0 failed, it indicates that copy was successful, as shown in the following figure.



Switch to the /home directory to view details. If you can see files, it indicates that files have been restored successfully.



Common mistakes and best practices

Data is users' core asset. Many users establish websites and databases (MYSQL/MongoDB/Redis) on ECS. Huge risks to the users' services may occur when data is lost. Common mistakes and best practices are summarized as follows.

Common mistakes

The bottom layer of Alibaba Cloud block-level storage is based on triplicate technology. Therefore, some users consider that no risk of data loss in the operating system exists. It is actually a misunderstanding. The three copies of data stored in the bottom layer provide physical layer protection for data disks. However, if problems occur to the cloud disk logic in the system, such as viruses, accidental data deletion, and file system damage, the data may still be lost. To guarantee data security, you have to use technologies such as Snapshot and backup.

Best practices

Data disk partition restoration and data restoration are the final solutions for solving data loss problems, but it is never guaranteed. We strongly recommend that you follow the best practices to perform auto or manual snapshot on data and run different backup schemes to maximize your data security.

Enable automatic snapshots

Automatic snapshots are enabled for the system disk and data disk based on actual service conditions. Note that automatic snapshot may be released when the system disk is changed, the instance is expired, or the disk is manually released.

You log on to the ECS console to change the attributes of the disks to enable or disable snapshot release with the disk.

For more information, see FAQs about automatic snapshots.

Create manual snapshots

Create snapshots manually before any important or risky operations such as:

- Upgrade the kernel
- Upgrade or change of applications
- Restoration of disk data

You must create snapshots for disks before restoring them. After the snapshots are completed, you can perform other operations.

OSS, offline, or offsite backup

You can back up important data by means of OSS, offline, or offsite backup based on actual conditions.

Data restoration in Windows instances

When solving problems related to disks, you may frequently encounter the loss of data disk partitions. This article describes common data partition loss problems and corresponding solutions in Windows, and provides common mistakes and best practices for cloud disks to avoid possible risks of data loss.

Prerequisites

Before restoring data, you must create snapshots for data disks that lose partitions. If problems occur during the restoration process, you can roll back data disks to the status before restoration.

Introduction to disk management tools

In Windows instances, you can select either of the following tools for restoring data disk data:

Disk Management
A tool provided by Windows for partitioning and formatting the disk.

Data restoration software
Generally, they are commercial software, and can be downloaded from the providers' official websites. They are mainly used for restoring data in an abnormal file system.

Status of the disk is Foreign and no partitions are

displayed

In the **Disk Management** of Windows, the disk is in the **Foreign** status and displays no partitions.

Solution: Right click the **Foreign** disk, select **Import Foreign Disks**, and then click **OK**.

Status of the disk is Offline and no partitions are displayed

In the **Disk Management** of Windows, the disk is in the **Offline** status and displays no partitions.

Solution: Right click the **Offline** disk, select **Online**.

No drive letter assigned

In the **Disk Management** of Windows, you can view data disk information, but no drive letter is allocated to the data disk.

Solution: Right click **Primary Partition** of the disk, click **Change drive letter and paths**, and then complete operations by prompt.

Error occurred during storage enumeration

In the **Disk Management** of Windows, you cannot view data disks. An error occurred during storage enumeration is reported in the system log.

> **Note**:
> Some versions may report **Error occurred during enumeration of volumes**. They are the same.

Solution: Follow these steps to fix the error.

1. Start Windows PowerShell.
2. Run winrm quickconfig for restoring. When Make these changes [y/n]? is displayed on the interface, you must type y to run the command.

After the restoration, you can have the data disks in the **Disk Management**.

Data disk is in RAW format

In some special circumstances, the disk in Windows is in RAW format. If the file system of a disk is

unrecognizable to Windows, it is displayed as a RAW disk. This usually occurs when the partition table or boot sector that records the type or location of the file system is lost or damaged. Common causes are listed as follows:

Safely remove hardware is not used when disconnecting the external disk.

Disk problems caused by power outages or unexpected shutdown.

Hardware layer failure may also cause information loss of the disk partition.

Bottom layer drivers or disk-related applications. For example, DiskProbe can be used to directly modify the disk table structure.

Computer viruses.

For more information about how to fix these problems, see Dskprobe Overview document.

Moreover, Windows also contains a large variety of free or commercial data restoration software to restore lost data. For example, you can try to use DiskGenius to scan and restore expected documents.

Common mistakes and best practices

Data is users' core asset. Many users establish websites and databases (MYSQL/MongoDB/Redis) on ECS. Huge risks to the users' services may occur when data is lost. Common mistakes and best practices are summarized as follows.

Common mistakes

The bottom layer of Alibaba Cloud block-level storage is based on triplicate technology. Therefore, some users consider that no risk of data loss in the operating system exists. It is actually a misunderstanding. The three copies of data stored in the bottom layer provide physical layer protection for data disks. However, if problems occur to the cloud disk logic in the system, such as viruses, accidental data deletion, and file system damage, the data may still be lost. To guarantee data security, you have to use technologies such as Snapshot and backup.

Best practices

Data disk partition restoration and data restoration are the final solutions for solving data loss problems, but it is never guaranteed. We strongly recommend that you follow the best practices to perform auto or manual snapshot on data and run different backup schemes to maximize your data security.

Enable automatic snapshots

Automatic snapshots are enabled for the system disk and data disk based on actual service conditions. Note that automatic snapshot may be released when the system disk is changed, the instance is expired, or the disk is manually released.

You log on to the ECS console to change the attributes of the disks to enable or disable snapshot release with the disk.

For more information, see FAQs about automatic snapshots.

Create manual snapshots

Create snapshots manually before any important or risky operations such as:

- Upgrade the kernel
- Upgrade or change of applications
- Restoration of disk data

You must create snapshots for disks before restoring them. After the snapshots are completed, you can perform other operations.

OSS, offline, or offsite backup

You can back up important data by means of OSS, offline, or offsite backup based on actual conditions.

Configurations

Synchronize clock with Internet time in a Windows ECS instance

Network Time Protocol (NTP) is a networking protocol for clock synchronization between computer systems over networks. For highly time-sensitive applications (such as those in the communication industry), clock variation between different computers may lead to serious data inconsistencies. You can use the NTP service to synchronize clocks of all servers within the network.

This article describes how use the NTP service to synchronize the clock of a Windows ECS instance

running Windows Server 2008 R2 Enterprise Edition x64. You can also use commands to complete tasks described in this article.

Enable the NTP service

Windows Time service is enabled by default on Windows Server. You must enable the NTP service in the instance to make sure that the NTP service can normally synchronize time after successful NTP service configuration.

To check and enable the NTP service, follow these steps:

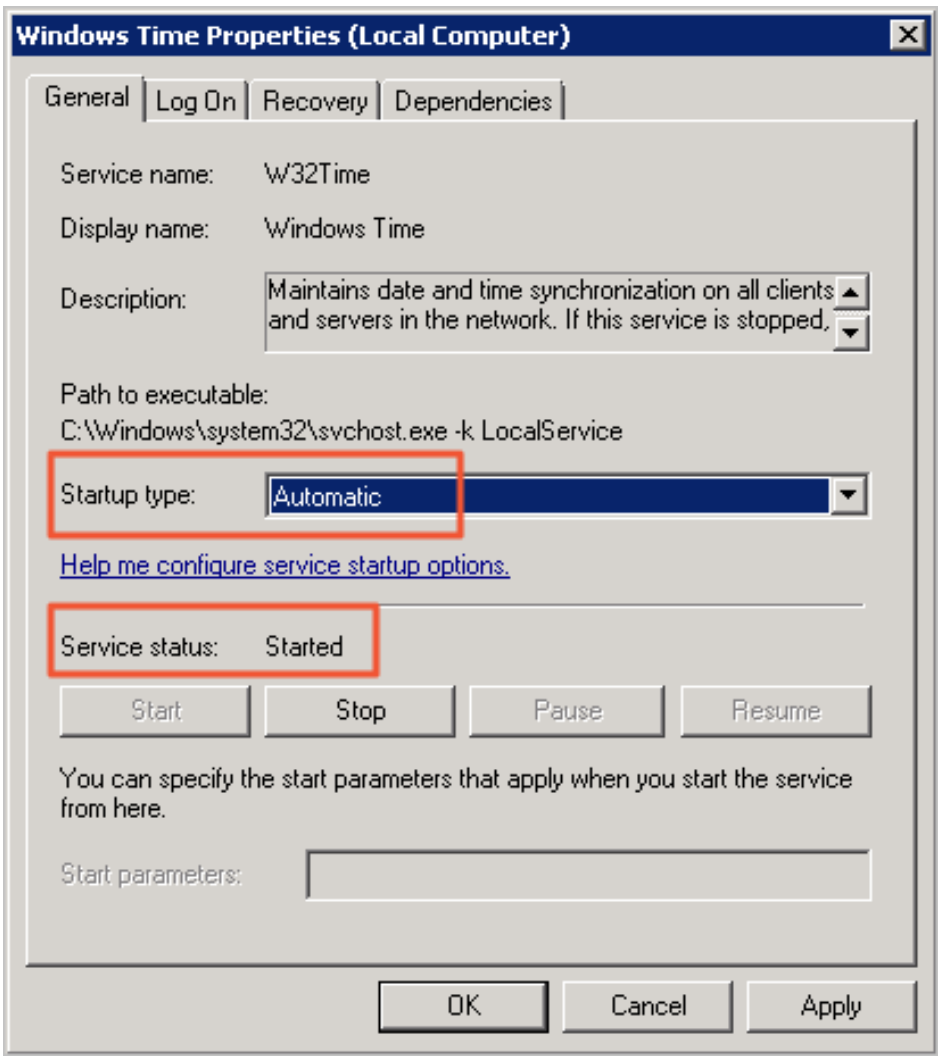
Connect to a Windows instance.

Select **Start** > **All Programs** > **Accessories** > **Run** to open the **Run** dialog box, and run services.msc.

In the **Services** window, find and double click the **Windows Time** service.

In the **Windows Time Properties (Local Computer)** dialog box, execute the following operations:

i. Set **Startup type** to **Automatic**.
ii. Check if the **Service status** is **Started**. If not, click **Start**.
After completing the settings, click **Apply**, and then click **OK**.



Modify the default NTP server address

time.windows.com is used as the default NTP server in Windows Server, but synchronization errors may frequently occur due to network issues. When using a Windows instance, you can replace the default NTP server with the intranet NTP server provided by Alibaba Cloud. For more information, see Internet and intranet NTP servers.

To modify the default NTP server address, follow these steps:

Connect to a Windows instance.

In the notification area of the task bar, click Date and Time, and then click **Change date and time settings**.

In the **Date and Time** dialog box, click the **Internet Time** tab, and then click **Change settings**.

In the **Internet Time Settings** dialog box, select **Synchronize with an Internet time server**, type an Alibaba Cloud intranet NTP server address (for detailed list, see **Internet and intranet NTP servers**), and then click **Update now**.

You are prompted if the synchronization is successful.

Modify NTP synchronization interval

The default NTP synchronization interval is 5 minutes. To modify the NTP synchronization interval, follow these steps:

Connect to a Windows instance.

Select **Start** > **All Programs** > **Accessories** > **Run** to open the **Run** dialog box, and run regedit.

On the left-side navigation pane of the **Registry Editor**, find HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/services/W32Time/TimeProviders/NtpClient, and then double click **SpecialPollInterval**.

In the **Eidt DWORD (32-bit) Value** dialog box, select **Decimal** as the **Base**, and then type the **Value data** as needed. The number you typed is the synchronization interval you need. Unit: seconds.

Build an FTP site on an ECS instance

This article describes how to build an FTP site on a Windows ECS instance or a Linux ECS instance.

> **Note**: You can harden your FTP service security. For more information, see **FTP anonymous logon and weak password vulnerabilities**.

Build an FTP site on a Windows ECS instance

This method is applicable to Windows Server 2003 and later Windows Server versions. Operations described in this article are conducted on Windows Server 2008 R2.

Procedure for building an FTP site on a Windows ECS instance is as follows:

- Step 1. Add IIS and FTP service roles
- Step 2. Create FTP username and password
- Step 3. Set permissions for shared files
- Step 4. Add and configure an FTP site
- Step 5. Configure a security group and firewall
- Step 6. Test

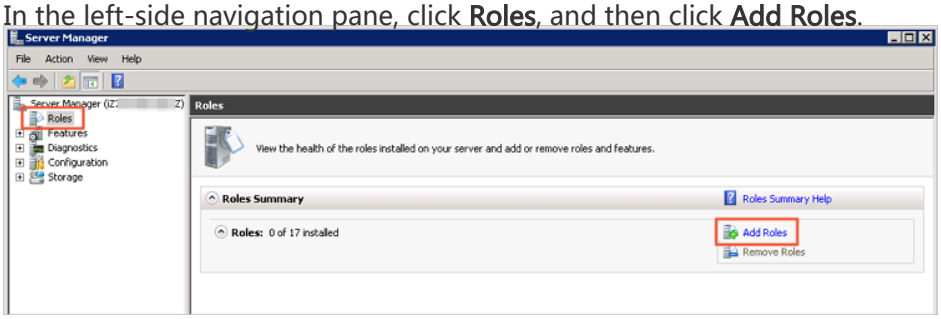
Step 1. Add IIS and FTP service roles

You must install IIS and FTP services before building an FTP site. Follow these steps to install IIS and FTP services:

Connect to a Windows instance.

Click the **Server Manager** icon.

In the left-side navigation pane, click **Roles**, and then click **Add Roles**.



In the **Add Roles Wizard** windows, click **Next**.

Select **Web Server (IIS)** and then click **Next**.

Select **IIS Management Console** and **FTP Server**, and then click **Next**.

Click **Install**.

Step 2. Create FTP username and password

Follow these steps to create a Windows username and password to be used by the FTP. If you want to allow anonymous users to access the FTP, skip this step.

Click the **Server Manager** icon.

On the left-side navigation pane, select **Configuration** > **Local Users and Groups**, and then double click **Users**.

Right click the blank space, and then select **New User**.

On the **New User** dialog box, type the new user information. For example, *ftptest* is used in this article.

Note: The password must contain a mixture of upper-case letters, lower-case letters and numbers. Otherwise, the password is invalid.

Step 3. Set permissions for shared files

You must set permissions to read, write, or execute for folders shared to users on the FTP site.

Create a folder for the FTP site, right click the folder, and then select **Properties**.

Click **Security**, select **Users**, and then click **Edit**.

Edit **Permissions for Users**. In this example, we grant all permissions.

Step 4. Add and configure an FTP site

Follow these steps to install an FTP site.

Select **Start** > **Administrative Tools** > **Internet Information Services (IIS) Manager**.

On the left-side navigation pane, click instance ID, and then right click **Sites** and click **Add FTP Site**.

In the **Add FTP Site** wizard, specify the FTP site name and the physical path of the shared folder, and then click **Next**.

Type the public IP address and port number of this instance. The default FTP port number is 21.

Select SSL settings, and click **Next**.

- **Allow SSL**: Allows the FTP site to support both non-SSL and SSL connections with

the client.
- **Require SSL**: Requires SSL encryption for communications between the FTP server and the client.
- **No SSL**: No SSL encryption is required.

Select one or more authentication methods.

- **Anonymous**: Allows any users who only enter the username “anonymous” or “ftp” to access the shared content.
- **Basic**: Requires users to enter the valid username and password before they can access the shared content. The Basic authentication method transmits the unencrypted password through the network. Therefore, use this authentication method only when you are sure that the connection between the client and the FTP server is secure, for example, when Secure Sockets Layer (SSL) is used.

Select one of the following options from the **Authorization** list, and set the permissions.

- **All users**: All users (anonymous users and identified users) can access the relevant content.
- **Anonymous users**: Anonymous users can access the relevant content.
- **Specified roles or user groups**: Only members of the specific role group or user group can access the relevant content. Enter the role group or user group in the corresponding field.
- **Specified users**: Only the specified users can access the relevant content. Enter the username in the corresponding field.

Click **Finish**.

Step 5. Configure a security group and firewall

After building the FTP site, you must add a rule to open the FTP port. For more information, see Add a security group rule.

TCP port 21 is open on the server firewall by default for FTP service. If you have entered another port number, you must add an inbound rule to open this port on the firewall.

Step 6. Test

On your local computer, access the FTP site by using ftp://public IP address:FTP port (the default port 21 is used if you do not enter the port). For example, ftp://0.0.0.0:20. You are prompted for your username and password, if the configuration was successful. After entering the username and password correctly, you can perform the relevant FTP file operations according to your permissions.

> **Note**: You must adjust the Internet Explorer settings to open FTP folders, if you use this method to access the FTP site from the client. Open Internet Explorer, and then select **Tools** > **Internet**

Options > Advanced. Select Enable folder view for FTP sites, and clear Use Passive FTP.

Build an FTP site on a Linux ECS instance

vsftpd is a light, safe, and easy-to-use FTP server for Linux. It is the most popular FTP server across all Linux versions. This article describes how to install vsftpd on a Linux ECS instance running CentOS 7.2 x64.

- Step 1. Install vsftpd
- Step 2. Configure vsftpd
- Step 3. Configure a security group
- Step 4. Test

Step 1. Install vsftpd

Connect to a Linux instance.

Run the following command to install vsftpd.

```
yum install -y vsftpd
```

Run the following command to open and view etc/vsftpd.

```
cd /etc/vsftpd
ls
```

Note:

- /etc/vsftpd/vsftpd.conf is the core configuration file.
- /etc/vsftpd/ftpusers is the blacklist, which prevents users on it from accessing the FTP server.
- /etc/vsftpd/user_list is the whitelist, which allows users on it to access the FTP server.

Run the following command to set vsftpd to automatically start on startup.

```
systemctl enable vsftpd.service
```

Run the following command to start the FTP service.

```
systemctl start vsftpd.service
```

Run the following command to view the FTP service port.

```
netstat -antup | grep ftp
```

Step 2. Configure vsftpd

After installation, vsftpd enables the Anonymous FTP function by default. Using the Anonymous FTP function, the user can log on to the FTP server without the username and password, but does not have the permission to modify or upload files.

This article describes the following vsftpd configuration methods and the corresponding parameter descriptions for your reference.

- Grant the file upload permission to anonymous users
- Configure local user logon
- Introduction to vsftpd.conf parameters

Grant the file upload permission to anonymous users

You can grant more permissions to anonymous users by modifying the options in the vsftpd.conf configuration file.

Follow these steps to modify /etc/vsftpd/vsftpd.conf:

i. Run vim /etc/vsftpd/vsftpd.conf.
ii. Press the i key to enter the Edit mode.
iii. Set write_enable=YES.
iv. Set anon_upload_enable=YES.
v. Press the Esc key and then type :wq to save and close the file.

Run the following command to change permissions of the /var/ftp/pub directory, granting write permission to FTP users, and reload the configuration file.

```
chmod o+w /var/ftp/pub/
systemctl restart vsftpd.service
```

Configure local user logon

Local user logon refers to the operation that a user logs on to the FTP server by using the username and password for the Linux operation system. After installing vsftpd, only anonymous FTP logon is supported. If you attempt to log on to the FTP server with the Linux username, your access will be denied by vsftp. However, you can adjust the vsftpd configuration to allow logon with username and

password. Follow these steps:

Run the following command to create the ftptest user.

```
useradd ftptest
```

Run the following command to modify the password for the ftptest user.

```
passwd ftptest
```

Follow these steps to modify /etc/vsftpd/vsftpd.conf:

i. Run vim /etc/vsftpd/vsftpd.conf.
ii. Press the i key to enter the Edit mode.
iii. Set anonymous enable=NO.
iv. Set local_enable=YES.
v. Press the Esc key and then type :wq to save and close the file.

Introduction to vsftpd.conf parameters

Run cat /etc/vsftpd/vsftpd.conf to view content in the configuration file.

The following table lists all the parameters related to user logon control.

Parameters	Description
anonymous_enable=YES	Allows anonymous logon
no_anon_password=YES	Anonymous users are not prompted for password when logging on
anon_root=(none)	Root directory for anonymous users
local_enable=YES	Allows local user logon
local_root=(none)	Root directory for local user

The following table lists all the parameters related to user permission control.

Parameters	Descriptions
write_enable=YES	Allows file upload (global control)
local_umask=022	Umask for the local user to upload files
file_open_mode=0666	Uses umask for file upload permission
anon_upload_enable=NO	Allows anonymous users to upload files
anon_mkdir_write_enable=NO	Allows anonymous users to create directories
anon_other_write_enable=NO	Allows anonymous users to modify and delete

	files and directories
chown_username=lightwiter	Username for anonymously uploaded files

Step 3. Configure a security group

After building the FTP site, you must add a rule to open the FTP port. For more information, see Add a security group rule.

Step 4. Test

On your local computer, access the FTP site by using ftp://public IP address:FTP port (the default port 21 is used if you do not enter the port). For example, ftp://0.0.0.0:20. You are prompted for your username and password, if the configuration was successful. After entering the username and password correctly, you can perform the relevant FTP file operations according to your permissions.

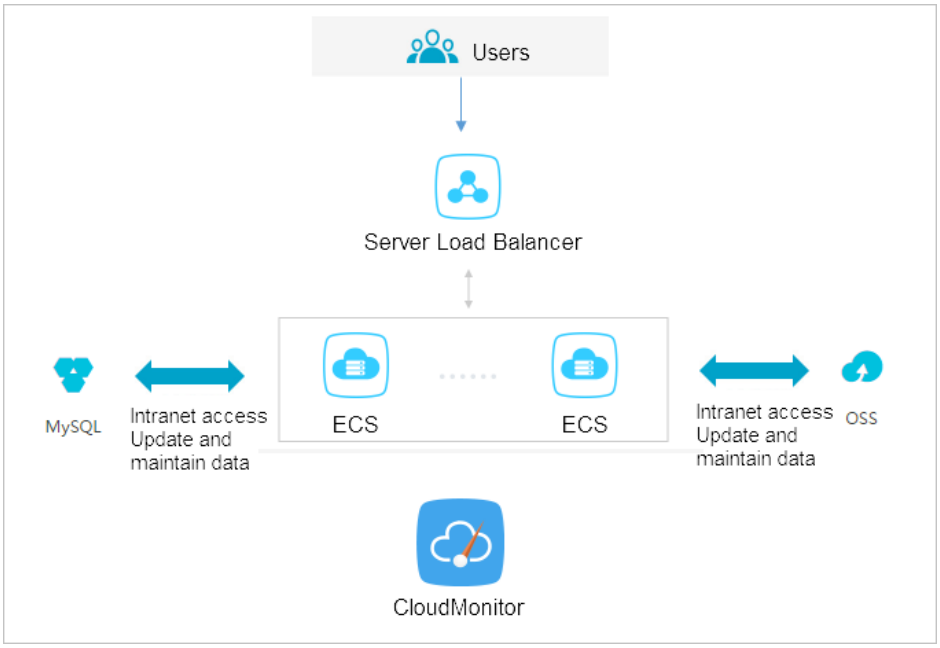
> **Note**: You must adjust the Internet Explorer settings to open FTP folders, if you use this method to access the FTP site from the client. Open Internet Explorer, and then select **Tools** > **Internet Options** > **Advanced**. Select **Enable folder view for FTP sites**, and clear **Use Passive FTP**.

Monitor

Use CloudMonitor to monitor ECS instances

Many businesses are moving to cloud computing because it is cost-effective, and saves customers of heavy lifting. This can be greatly attributed to the leverage of monitoring. Monitoring service provides real-time operation data for you to identify risks in advance, avoid potential loss, and troubleshoot as quickly as possible.

This article takes a website for example (the website architecture is shown as follows) to illustrate how to configure CloudMonitor. The example website uses Alibaba Cloud services such as ECS, RDS, OSS, and Server Load Balancer.



Prerequisites

Before you begin, you must complete the following operations:

Make sure that your ECS monitoring agents are functional to collect metric data. Otherwise, you must install the agent manually. For more information, see How to install CloudMonitor agent.

Add alarm contacts and contact groups. We recommend that you add at least two contacts to make sure real-time responses to monitoring alarms. For more information about metrics, see Cloud service overview and alarm overview.

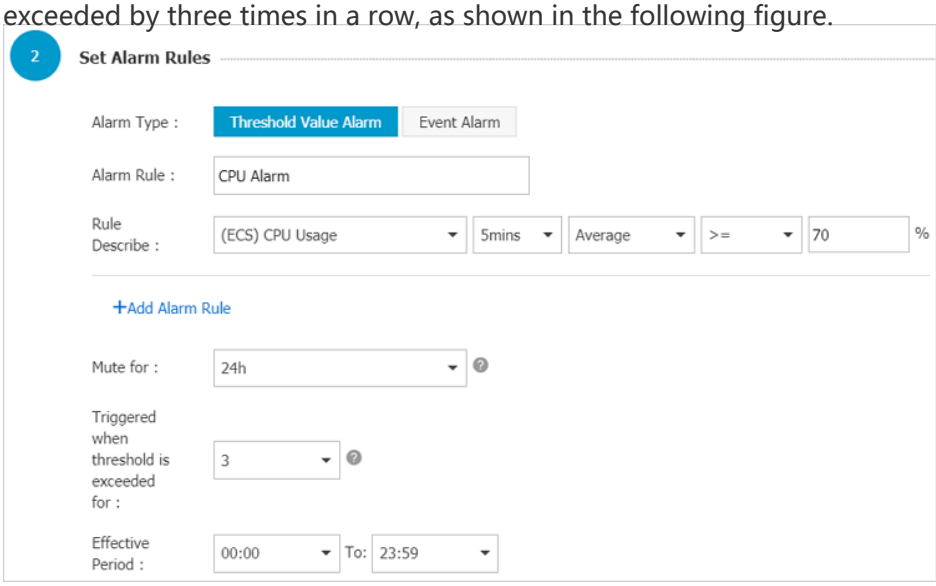
With CloudMonitor Dashboard, you can gain system-wide visibility into resource utilization and operational health. You can select a metrics dimension. You can choose per-instance metrics dimension if you only have several instances. Otherwise, you can choose ECS groups dimension or user dimension, and choose the average value.

Set alarm threshold

We recommend that you set the alarm threshold according to your business status. A much lower threshold may trigger alarm too often and render monitoring meaningless, while a much higher threshold may leave you with no time to respond to a major event.

Set alarm rules

Take CPU utilization as an example. We have to reserve some processing capacity to guarantee the normal function, so you can set the threshold to 70% and to trigger an alarm when the threshold is exceeded by three times in a row, as shown in the following figure.



If you have to set alarm rules for other metrics, click **Add Alarm Rule**.

Set process monitoring

For Web applications, you can add monitoring for process. For more information, see Process monitoring.

Set site monitoring

Site monitoring is at the network access layer to test the availability.

Set RDS monitoring

We recommend that you set the RDS CPU utilization alarm threshold to 70% and to trigger an alarm when the threshold is exceeded by three times in a row. You can set the disk utilization , IOPS utilization, total connections and other metrics as needed.

Set Server Load Balancer monitoring

Before you begin, make sure that you have enabled health check for your Server Load Balancer instance. For more information, see Health check overview.

You can use Custom monitoring metrics if the metrics you need are not covered.

Use API to run ECS

Create an instance

Use OpenAPI to Create Instance

In addition to the ECS Console or Buy Page, you can also use OpenAPI code to elastically create and manage ECS instances. This article describes how to create an ECS instance using Python.

When creating an ECS instance, pay attention to the following APIs:

- Create an ECS instance
- Query an instance list
- Start an ECS instance
- Allocate a public IP address

Create a Pay-As-You-Go ECS instance

Mandatory attributes:

- SecurityGroupId: Security group ID. A security group is used to implement the configurations of a group of instances based on firewall rules to protect the network access requests of the instances. We recommend that only necessary access rules, rather than all access rules, be enabled when you configure security group access rules. You can create a security group in the ECS console.
- InstanceType: Instance type. See the ECS Buy Page. The option “one-core 2GB n1.small” indicates that the input parameter is “ecs.n1.small”.
- ImageId: Image ID. See the image list in the ECS console. You can filter public images or custom images.

For more parameter settings, see Create an ECS instance.

Create an ECS instance

The following code shows creating an I/O optimized classic-network ECS instance with SSD as system disk and “cloud_ssd” as disk parameter.

```
# create one after pay ecs instance.
def create_after_pay_instance(image_id, instance_type, security_group_id):
request = CreateInstanceRequest();
request.set_ImageId(image_id)
request.set_SecurityGroupId(security_group_id)
request.set_InstanceType(instance_type)
request.set_IoOptimized('optimized')
request.set_SystemDiskCategory('cloud_ssd')
response = _send_request(request)
instance_id = response.get('InstanceId')
logging.info("instance %s created task submit successfully.", instance_id)
return instance_id;
```

An instance ID is returned after the ECS instance is created successfully. If creation fails, an error code is returned. Since there are many parameters, you can make adjustments by visiting the ECS Buy Page.

```
{"InstanceId":"i-***","RequestId":"006C1303-BAC5-48E5-BCDF-7FD5C2E6395D"}
```

ECS lifecycle

For more information about the operations in different ECS status, see ECS Instance Lifecycle.

Only when an instance is in the Stopped status, can the Start operation be performed, and only when it is in the Running status, can the Stop operation be performed. To query the ECS status, you can filter the instance list by inputting the parameter Instance ID. When you call DescribeInstancesRequest, input a JSON array of strings to query the resource status. When you query the status of a single instance, we suggest using DescribeInstances rather than DescribeInstanceAttribute, because the former API returns more attributes and content than the latter.

The following code is used to check the instance status. The system returns instance details only when the instance status conforms to the input parameters.

```
# output the instance owned in current region.
def get_instance_detail_by_id(instance_id, status='Stopped'):
logging.info("Check instance %s status is %s", instance_id, status)
request = DescribeInstancesRequest()
request.set_InstanceIds(json.dumps([instance_id]))
response = _send_request(request)
instance_detail = None
if response is not None:
instance_list = response.get('Instances').get('Instance')
for item in instance_list:
if item.get('Status') == status:
instance_detail = item
break;
return instance_detail;
```

Start an ECS instance

After an ECS instance is created successfully, the default instance status is Stopped. To change to the Running status, send the Start command.

```
def start_instance(instance_id):
request = StartInstanceRequest()
request.set_InstanceId(instance_id)
_send_request(request)
```

Stop an ECS instance

To stop an ECS instance, use the input instance ID.

```
def stop_instance(instance_id):
request = StopInstanceRequest()
request.set_InstanceId(instance_id)
_send_request(request)
```

Enable “ECS automatic startup” when creating an ECS instance

The ECS Start and Stop operations are asynchronous. You can perform the operation when the script is creating an ECS instance and detecting if it is in an appropriate status.

After you obtain the ID of a successfully created ECS instance, check whether the instance is in the Stopped status. If it is in the Stopped status, send the Start ECS command and wait until the ECS status changes to Running.

```
def check_instance_running(instance_id):
detail = get_instance_detail_by_id(instance_id=instance_id, status=INSTANCE_RUNNING)
index = 0
while detail is None and index < 60:
detail = get_instance_detail_by_id(instance_id=instance_id);
time.sleep(10)

if detail and detail.get('Status') == 'Stopped':
logging.info("instance %s is stopped now.")
start_instance(instance_id=instance_id)
logging.info("start instance %s job submit.")

detail = get_instance_detail_by_id(instance_id=instance_id, status=INSTANCE_RUNNING)
while detail is None and index < 60:
detail = get_instance_detail_by_id(instance_id=instance_id, status=INSTANCE_RUNNING);
time.sleep(10)

logging.info("instance %s is running now.", instance_id)
return instance_id;
```

Allocate a public IP address

If you specify the public network bandwidth when creating an ECS instance, you need to call an API to allocate a public IP address to the instance for public network access. For more information, see Allocate a public IP address.

Create an ECS instance in the Subscription mode

OpenAPI also supports creating ECS instances in the Subscription mode, in addition to Pay-As-You-Go ECS instances. The process for creating an ECS instance in the Subscription mode is different from that on Alibaba Cloud’ s website. Fees are automatically deducted for an ECS instance created in the Subscription mode. Before you create an ECS instance, make sure that you have sufficient account balance or credit amount, so that the fees can be deducted directly during creation.

When creating an ECS instance in Subscription mode, you only need to specify the payment option and duration. In the following code, the duration is set to one month.

```
request.set_Period(1)  request.set_InstanceChargeType( ‘PrePaid’ )
```

The complete code for creating an ECS instance in the Subscription mode is as follows:

```
# create one prepay ecs instance.
def create_prepay_instance(image_id, instance_type, security_group_id):
request = CreateInstanceRequest();
request.set_ImageId(image_id)
request.set_SecurityGroupId(security_group_id)
request.set_InstanceType(instance_type)
request.set_IoOptimized('optimized')
request.set_SystemDiskCategory('cloud_ssd')
request.set_Period(1)
request.set_InstanceChargeType('PrePaid')
response = _send_request(request)
instance_id = response.get('InstanceId')
logging.info("instance %s created task submit successfully.", instance_id)
return instance_id;
```

Complete code

See the complete code as follows. You can use your resource parameters for configuration.

```
#  coding=utf-8
# if the python sdk is not install using 'sudo pip install aliyun-python-sdk-ecs'
# if the python sdk is install using 'sudo pip install --upgrade aliyun-python-sdk-ecs'
# make sure the sdk version is 2.1.2, you can use command 'pip show aliyun-python-sdk-ecs' to check

import json
import logging
```

```
import time

from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.CreateInstanceRequest import CreateInstanceRequest
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.StartInstanceRequest import StartInstanceRequest

# configuration the log output formatter, if you want to save the output to file,
# append ",filename='ecs_invoke.log'" after datefmt.

logging.basicConfig(level=logging.INFO,
format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelname)s %(message)s',
datefmt='%a, %d %b %Y %H:%M:%S')

clt = client.AcsClient('Your Access Key Id', 'Your Access Key Secrect', 'cn-beijing')

IMAGE_ID = 'ubuntu1404_64_40G_cloudinit_20160727.raw'
INSTANCE_TYPE = 'ecs.s2.large' # 2c4g generation 1
SECURITY_GROUP_ID = 'sg-****'
INSTANCE_RUNNING = 'Running'

def create_instance_action():
instance_id = create_after_pay_instance(image_id=IMAGE_ID, instance_type=INSTANCE_TYPE,
security_group_id=SECURITY_GROUP_ID)
check_instance_running(instance_id=instance_id)

def create_prepay_instance_action():
instance_id = create_prepay_instance(image_id=IMAGE_ID, instance_type=INSTANCE_TYPE,
security_group_id=SECURITY_GROUP_ID)
check_instance_running(instance_id=instance_id)

# create one after pay ecs instance.
def create_after_pay_instance(image_id, instance_type, security_group_id):
request = CreateInstanceRequest();
request.set_ImageId(image_id)
request.set_SecurityGroupId(security_group_id)
request.set_InstanceType(instance_type)
request.set_IoOptimized('optimized')
request.set_SystemDiskCategory('cloud_ssd')
response = _send_request(request)
instance_id = response.get('InstanceId')
logging.info("instance %s created task submit successfully.", instance_id)
return instance_id;

# create one prepay ecs instance.
def create_prepay_instance(image_id, instance_type, security_group_id):
request = CreateInstanceRequest();
request.set_ImageId(image_id)
request.set_SecurityGroupId(security_group_id)
request.set_InstanceType(instance_type)
request.set_IoOptimized('optimized')
request.set_SystemDiskCategory('cloud_ssd')
request.set_Period(1)
request.set_InstanceChargeType('PrePaid')
response = _send_request(request)
instance_id = response.get('InstanceId')
```

```
logging.info("instance %s created task submit successfully.", instance_id)
return instance_id;

def check_instance_running(instance_id):
detail = get_instance_detail_by_id(instance_id=instance_id, status=INSTANCE_RUNNING)
index = 0
while detail is None and index < 60:
detail = get_instance_detail_by_id(instance_id=instance_id);
time.sleep(10)

if detail and detail.get('Status') == 'Stopped':
logging.info("instance %s is stopped now.")
start_instance(instance_id=instance_id)
logging.info("start instance %s job submit.")

detail = get_instance_detail_by_id(instance_id=instance_id, status=INSTANCE_RUNNING)
while detail is None and index < 60:
detail = get_instance_detail_by_id(instance_id=instance_id, status=INSTANCE_RUNNING);
time.sleep(10)

logging.info("instance %s is running now.", instance_id)
return instance_id;

def start_instance(instance_id):
request = StartInstanceRequest()
request.set_InstanceId(instance_id)
_send_request(request)

# output the instance owned in current region.
def get_instance_detail_by_id(instance_id, status='Stopped'):
logging.info("Check instance %s status is %s", instance_id, status)
request = DescribeInstancesRequest()
request.set_InstanceIds(json.dumps([instance_id]))
response = _send_request(request)
instance_detail = None
if response is not None:
instance_list = response.get('Instances').get('Instance')
for item in instance_list:
if item.get('Status') == status:
instance_detail = item
break;
return instance_detail;

# send open api request
def _send_request(request):
request.set_accept_format('json')
try:
response_str = clt.do_action(request)
logging.info(response_str)
response_detail = json.loads(response_str)
return response_detail
except Exception as e:
logging.error(e)

if __name__ == '__main__':
logging.info("Create ECS by OpenApi!")
```

```
create_instance_action()
# create_prepay_instance_action()
```

Manage instances

In addition to using Alibaba Cloud ECS Console for resource creation and daily management, you can also use API to manage and customize resources. API allows you to manage and configure ECS instances with greater flexibility.

Alibaba Cloud encapsulates API in an SDK to integrate ECS instance management into existing systems. This article describes how to manage ECS instances through API based on Python development. You can develop ECS instances easily even if you do not have Python development experience.

Get the AccessKey for a RAM user

An AccessKey (AccessKey ID and AccessKey Secret) is required when you want to use API to manage ECS instances. To keep your cloud service secure, you have to create a RAM user and generate an AccessKey for it, and authorize the RAM user to manage ECS resources only. Then, you can use the RAM user and its AccessKey to manage ECS resources by using API.

Follow these steps to get the AccessKey for a RAM user.

1. Create a RAM user and get the AccessKey.
2. Grant permissions to the RAM user directly. To manage ECS resources, you have to grant **AliyunECSFullAccess** to the RAM user.

Install the ECS Python SDK

Make sure that the Python runtime environment has been installed. This article uses Python 2.7+.

```
pip install aliyun-python-SDK-ecs
```

If you do not have the permission, switch to sudo to continue.

```
sudo pip install aliyun-python-SDK-ecs
```

The SDK version is **2.1.2**.

Hello Alibaba Cloud

Create the file **hello_ecs_api.py**. To use SDK, you have to use the AccessKey of the RAM user to instantialize an AcsClient object.

> The AccessKey allows the RAM user to access Alibaba Cloud APIs and give you full access to the user. Keep them safe.

```
from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.DescribeRegionsRequest import DescribeRegionsRequest


clt = client.AcsClient('Your Access Key Id', 'Your Access Key Secrect', 'cn-beijing')
```

You can develop your first application after the AcsClient object is instantiated. Query the list of regions that your account supports. For more information, see Query the list of available regions.

```
def hello_aliyun_regions():
request = DescribeRegionsRequest()
response = _send_request(request)
region_list = response.get('Regions').get('Region')
assert response is not None
assert region_list is not None
result = map(_print_region_id, region_list)
logging.info("region list: %s", result)

def _print_region_id(item):
region_id = item.get("RegionId")
return region_id

def _send_request(request):
request.set_accept_format('json')
try:
response_str = clt.do_action(request)
logging.info(response_str)
response_detail = json.loads(response_str)
return response_detail
except Exception as e:
logging.error(e)

hello_aliyun_regions()
```

In the command line, run python hello_ecs_api.py to obtain a list of supported regions. The output is similar to the following.

```
[u'cn-shenzhen', u'ap-southeast-1', u'cn-qingdao', u'cn-beijing', u'cn-shanghai', u'us-east-1', u'cn-hongkong', u'me-east-1', u'ap-southeast-2', u'cn-hangzhou', u'eu-central-1', u'ap-northeast-1', u'us-west-1']
```

Query the list of ECS instances in the current region

The process for querying the instance list is similar to the region list. You only need to replace the input parameter DescribeRegionsRequest with DescribeInstancesRequest. For a full list of query parameters, see Query an instance list.

```
def list_instances():
request = DescribeInstancesRequest()
response = _send_request(request)
if response is not None:
instance_list = response.get('Instances').get('Instance')
result = map(_print_instance_id, instance_list)
logging.info("current region include instance %s", result)

def _print_instance_id(item):
instance_id = item.get('InstanceId');
return instance_id
```

The output is as follows.

```
current region include instance [u'i-****', u'i-****'']
```

For a full list of APIs, see ECS API overview. If you want to query a list of disks, replace DescribeInstancesRequest with DescribeDisksRequest.

Complete code

The following is the complete code of the operations described in this document.

```
#  coding=utf-8

# if the python sdk is not install using 'sudo pip install aliyun-python-sdk-ecs'
# if the python sdk is install using 'sudo pip install --upgrade aliyun-python-sdk-ecs'
# make sure the sdk version is 2.1.2, you can use command 'pip show aliyun-python-sdk-ecs' to check

import json
import logging

from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.DescribeRegionsRequest import DescribeRegionsRequest

# configuration the log output formatter, if you want to save the output to file,
# append ",filename='ecs_invoke.log'" after datefmt.
logging.basicConfig(level=logging.INFO,
format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelname)s %(message)s',
datefmt='%a, %d %b %Y %H:%M:%S')

clt = client.AcsClient('Your Access Key Id', 'Your Access Key Secrect', 'cn-beijing')
```

```
# sample api to list aliyun open api.
def hello_aliyun_regions():
request = DescribeRegionsRequest()
response = _send_request(request)
if response is not None:
region_list = response.get('Regions').get('Region')
assert response is not None
assert region_list is not None
result = map(_print_region_id, region_list)
logging.info("region list: %s", result)


# output the instance owned in current region.
def list_instances():
request = DescribeInstancesRequest()
response = _send_request(request)
if response is not None:
instance_list = response.get('Instances').get('Instance')
result = map(_print_instance_id, instance_list)
logging.info("current region include instance %s", result)


def _print_instance_id(item):
instance_id = item.get('InstanceId');
return instance_id


def _print_region_id(item):
region_id = item.get("RegionId")
return region_id


# send open api request
def _send_request(request):
request.set_accept_format('json')
try:
response_str = clt.do_action(request)
logging.info(response_str)
response_detail = json.loads(response_str)
return response_detail
except Exception as e:
logging.error(e)


if __name__ == '__main__':
logging.info("Hello Aliyun OpenApi!")
hello_aliyun_regions()
list_instances()
```

If you want to learn other API operations in ECS, see ECS API operation.

Release an instance

One important feature of ECS is on-demand resource creation. You can create custom resources elastically on demand during peak service hours, and then release those resources after service computing is completed. This document describes how to easily release ECS instances and achieve elasticity.

This document covers the following APIs:

- DeleteInstance
- ModifyInstanceAutoReleaseTime
- StopInstance
- Instance list query API

After an ECS instance is released, the physical resources used by the instance are recycled, including disks and snapshots. The data of the instance is completely lost and can never be recovered. If you want to retain the data, we recommend that you create snapshots of disks before releasing the ECS instance. The snapshots can be directly used to create a new ECS instance.

To release an ECS instance, you must stop it first. If any application is affected after the ECS instance is stopped, restart the instance.

Stop an ECS instance

Use the **StopInstance** interface to stop an ECS instance, regardless of the billing method of the instance. The stop command is as follows. When the ForceStop parameter is set to true, the ECS instance is stopped directly but data is not necessarily written to a disk, similar to power failure. Therefore, if you want to release an instance, set ForceStop to true.

```
def stop_instance(instance_id, force_stop=False):
'''
stop one ecs instance.
:param instance_id: instance id of the ecs instance, like 'i-***'.
:param force_stop: if force stop is true, it will force stop the server and not ensure the data
write to disk correctly.
:return:
'''
request = StopInstanceRequest()
request.set_InstanceId(instance_id)
request.set_ForceStop(force_stop)
logging.info("Stop %s command submit successfully.", instance_id)
_send_request(request)
```

Release an ECS instance

Use the **DeleteInstance** interface to release an ECS instance.

When the ECS instance is in the **Stopped** status, you can release it. The API has only two request parameters:

- InstanceId: Instance ID
- Force: If this parameter is set to “true”, the ECS instance is released forcibly even when it is not in the **Stopped** status. Use caution when setting this parameter. Release by mistake may affect your services.

```
def release_instance(instance_id, force=False):
'''
delete instance according instance id, only support after pay instance.
:param instance_id: instance id of the ecs instance, like 'i-***'.
:param force:
if force is false, you need to make the ecs instance stopped, you can
execute the delete action.
If force is true, you can delete the instance even the instance is running.
:return:
'''
request = DeleteInstanceRequest();
request.set_InstanceId(instance_id)
request.set_Force(force)
_send_request(request)
```

The following response is returned when an ECS instance is released successfully:

```
{"RequestId":"689E5813-D150-4664-AF6F-2A27BB4986A3"}
```

If you release an ECS instance when it is not in the **Stopped** status, an error occurs:

```
{"RequestId":"3C6DEAB4-7207-411F-9A31-6ADE54C268BE","HostId":"ecs-cn-
hangzhou.aliyuncs.com","Code":"IncorrectInstanceStatus","Message":"The current status of the resource does not
support this operation."}
```

Set the automatic release time for an ECS instance

You can set the automatic release time for an ECS instance to simplify instance management. When the set time is reached, Alibaba Cloud releases your ECS instance automatically. Use the **ModifyInstanceAutoReleaseTime** to set the automatic release time for an ECS instance.

> Note:
> The automatic release time follows the ISO8601 standard in UTC time. The format is yyyy-MM-ddTHH:mm:ssZ. If the seconds place is not 00, it is automatically set to start from the current

minute.
The automatic release time must be at least half an hour later than the current time, and must not be more than 3 years since the current time.

```
def set_instance_auto_release_time(instance_id, time_to_release = None):
'''
setting instance auto delete time
:param instance_id: instance id of the ecs instance, like 'i-***'.
:param time_to_release: if the property is setting, such as '2017-01-30T00:00:00Z'
it means setting the instance to be release at that time.
if the property is None, it means cancel the auto delete time.
:return:
'''
request = ModifyInstanceAutoReleaseTimeRequest()
request.set_InstanceId(instance_id)
if time_to_release is not None:
request.set_AutoReleaseTime(time_to_release)
_send_request(request)
```

Run the command set_instance_auto_release_time('i-1111', '2017-01-30T00:00:00Z') to set the time. Then you can use the DescribeInstances to query the automatic release time.

```
def describe_instance_detail(instance_id):
'''
describe instance detail
:param instance_id: instance id of the ecs instance, like 'i-***'.
:return:
'''
request = DescribeInstancesRequest()
request.set_InstanceIds(json.dumps([instance_id]))
response = _send_request(request)
if response is not None:
instance_list = response.get('Instances').get('Instance')
if len(instance_list) > 0:
return instance_list[0]


def check_auto_release_time_ready(instance_id):
detail = describe_instance_detail(instance_id=instance_id)
if detail is not None:
release_time = detail.get('AutoReleaseTime')
return release_time
```

If you want to cancel the automatic release due to service changes, run the set_instance_auto_release_time('i-1111') command to set the automatic release time to null.

Complete example code

Note: Proceed with caution when releasing ECS instances.

```
#  coding=utf-8

# if the python sdk is not install using 'sudo pip install aliyun-python-sdk-ecs'
# if the python sdk is install using 'sudo pip install --upgrade aliyun-python-sdk-ecs'
# make sure the sdk version is 2.1.2, you can use command 'pip show aliyun-python-sdk-ecs' to check

import json
import logging

from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DeleteInstanceRequest import DeleteInstanceRequest
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.ModifyInstanceAutoReleaseTimeRequest import \
ModifyInstanceAutoReleaseTimeRequest
from aliyunsdkecs.request.v20140526.StopInstanceRequest import StopInstanceRequest

# configuration the log output formatter, if you want to save the output to file,
# append ",filename='ecs_invoke.log'" after datefmt.
logging.basicConfig(level=logging.INFO,
format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelname)s %(message)s',
datefmt='%a, %d %b %Y %H:%M:%S')

clt = client.AcsClient('Your Access Key Id', 'Your Access Key Secrect', 'cn-beijing')

def stop_instance(instance_id, force_stop=False):
'''
stop one ecs instance.
:param instance_id: instance id of the ecs instance, like 'i-***'.
:param force_stop: if force stop is true, it will force stop the server and not ensure the data
write to disk correctly.
:return:
'''
request = StopInstanceRequest()
request.set_InstanceId(instance_id)
request.set_ForceStop(force_stop)
logging.info("Stop %s command submit successfully.", instance_id)
_send_request(request)

def describe_instance_detail(instance_id):
'''
describe instance detail
:param instance_id: instance id of the ecs instance, like 'i-***'.
:return:
'''
request = DescribeInstancesRequest()
request.set_InstanceIds(json.dumps([instance_id]))
response = _send_request(request)
if response is not None:
instance_list = response.get('Instances').get('Instance')
if len(instance_list) > 0:
return instance_list[0]

def check_auto_release_time_ready(instance_id):
detail = describe_instance_detail(instance_id=instance_id)
if detail is not None:
```

```
release_time = detail.get('AutoReleaseTime')
return release_time

def release_instance(instance_id, force=False):
'''
delete instance according instance id, only support after pay instance.
:param instance_id: instance id of the ecs instance, like 'i-***'.
:param force:
if force is false, you need to make the ecs instance stopped, you can
execute the delete action.
If force is true, you can delete the instance even the instance is running.
:return:
'''
request = DeleteInstanceRequest();
request.set_InstanceId(instance_id)
request.set_Force(force)
_send_request(request)

def set_instance_auto_release_time(instance_id, time_to_release = None):
'''
setting instance auto delete time
:param instance_id: instance id of the ecs instance, like 'i-***'.
:param time_to_release: if the property is setting, such as '2017-01-30T00:00:00Z'
it means setting the instance to be release at that time.
if the property is None, it means cancel the auto delete time.
:return:
'''
request = ModifyInstanceAutoReleaseTimeRequest()
request.set_InstanceId(instance_id)
if time_to_release is not None:
request.set_AutoReleaseTime(time_to_release)
_send_request(request)
release_time = check_auto_release_time_ready(instance_id)
logging.info("Check instance %s auto release time setting is %s. ", instance_id, release_time)

def _send_request(request):
'''
send open api request
:param request:
:return:
'''
request.set_accept_format('json')
try:
response_str = clt.do_action(request)
logging.info(response_str)
response_detail = json.loads(response_str)
return response_detail
except Exception as e:
logging.error(e)

if __name__ == '__main__':
logging.info("Release ecs instance by Aliyun OpenApi!")
set_instance_auto_release_time('i-1111', '2017-01-28T06:00:00Z')
# set_instance_auto_release_time('i-1111')
# stop_instance('i-1111')
# release_instance('i-1111')
```

```
# release_instance('i-1111', True)
```

If you want to learn other API operations in ECS, see ECS API operation.

Renew instances

Lifecycle is important to ECS instances of the Subscription billing method. If you fail to renew your ECS instance on time, the instance may be locked or even released, thus affecting your service continuity. In addition to the ECS console or the ECS purchase page, Alibaba Cloud provides you with APIs to view the resource expiration time and renew your instance.

This document covers the following APIs:

- DescribeInstances
- ModifyInstanceAutoRenewAttribute

Query ECS instances by expiration time

Use the DescribeInstances interface to query the instances that will expire within the specified time range by setting the filter parameters ExpiredStartTime and ExpiredEndTime. The time parameters follow the ISO8601 standard in UTC time, using the format yyyy-MM-ddTHH:mmZ. The system returns a list of instances that will expire within the specified time range.

> **Note**: If you want to filter by security group, add the security group ID.

```
INSTANCE_EXPIRED_START_TIME_IN_UTC_STRING = '2017-01-22T00:00Z'
INSTANCE_EXPIRE_END_TIME_IN_UTC_STRING = '2017-01-28T00:00Z'

def describe_need_renew_instance(page_size=100, page_number=1, instance_id=None,
check_need_renew=True, security_group_id=None):
request = DescribeInstancesRequest()
if check_need_renew is True:
request.set_Filter3Key("ExpiredStartTime")
request.set_Filter3Value(INSTANCE_EXPIRED_START_TIME_IN_UTC_STRING)
request.set_Filter4Key("ExpiredEndTime")
request.set_Filter4Value(INSTANCE_EXPIRE_END_TIME_IN_UTC_STRING)
if instance_id is not None:
request.set_InstanceIds(json.dumps([instance_id]))
if security_group_id:
request.set_SecurityGroupId(security_group_id)
request.set_PageNumber(page_number)
request.set_PageSize(page_size)
return _send_request(request)
```

Query and enable automatic ECS instance renewal

You can use the ModifyInstanceAutoRenewAttribute interface to query and set automatic renewal. The API supports only ECS instances of the Subscription billing method. If you use the API on a Pay-As-You-Go instance, an error will be returned.

Query automatic renewal setting

To query the automatic renewal setting, only the instance ID is required. You can query the automatic renewal status of up to 100 ECS instances of the Subscription billing method at a time. Use commas to separate multiple instance IDs.

```
# check the instances is renew or not
def describe_auto_renew(instance_ids, expected_auto_renew=True):
describe_request = DescribeInstanceAutoRenewAttributeRequest()
describe_request.set_InstanceId(instance_ids)
response_detail = _send_request(request=describe_request)
failed_instance_ids = ''
if response_detail is not None:
attributes = response_detail.get('InstanceRenewAttributes').get('InstanceRenewAttribute')
if attributes:
for item in attributes:
auto_renew_status = item.get('AutoRenewEnabled')
if auto_renew_status != expected_auto_renew:
failed_instance_ids += item.get('InstanceId') + ','

describe_auto_renew('i-1111,i-2222')
```

The following content is returned:

```
{"InstanceRenewAttributes":{"InstanceRenewAttribute":[{"Duration":0,"InstanceId":"i-
1111","AutoRenewEnabled":false},{"Duration":0,"InstanceId":"i-
2222","AutoRenewEnabled":false}]},"RequestId":"71FBB7A5-C793-4A0D-B17E-D6B426EA746A"}
```

If automatic renewal is set, the returned attribute AutoRenewEnabled is true. If automatic renewal is not set, the attribute is false.

Enable automatic renewal for ECS instances

To enable automatic renwal for ECS instances, three input parameters are required:

- InstanceId: You can set automatic renewal for up to 100 ECS instances of the Subscription billing method at a time. Use commas to separate multiple instance IDs.
- Duration: Set to 1, 2, 3, 6, or 12, in unit of Month.
- AutoRenew: Set to true to enable automatic renewal.

  Note: Set to false to disable automatic renewal.

```
def setting_instance_auto_renew(instance_ids, auto_renew = True):
logging.info('execute enable auto renew ' + instance_ids)
request = ModifyInstanceAutoRenewAttributeRequest();
request.set_Duration(1);
request.set_AutoRenew(auto_renew);
request.set_InstanceId(instance_ids)
_send_request(request)
```

When the operation is successful, the following response is returned:

```
{"RequestId":"7DAC9984-AAB4-43EF-8FC7-7D74C57BE46D"}
```

You can perform a query after successful renewal. The system will return the renewal duration and the status of automatic renewal (true/false).

```
{"InstanceRenewAttributes":{"InstanceRenewAttribute":[{"Duration":1,"InstanceId":"i-
1111","AutoRenewEnabled":true},{"Duration":1,"InstanceId":"i-
2222","AutoRenewEnabled":true}]},"RequestId":"7F4D14B0-D0D2-48C7-B310-B1DF713D4331"}
```

Complete example code

```
#  coding=utf-8

# if the python sdk is not install using 'sudo pip install aliyun-python-sdk-ecs'
# if the python sdk is install using 'sudo pip install --upgrade aliyun-python-sdk-ecs'
# make sure the sdk version is 2.1.2, you can use command 'pip show aliyun-python-sdk-ecs' to check

import json
import logging

from aliyunsdkcore import client
from aliyunsdkecs.request.v20140526.DescribeInstanceAutoRenewAttributeRequest import \
DescribeInstanceAutoRenewAttributeRequest
from aliyunsdkecs.request.v20140526.DescribeInstancesRequest import DescribeInstancesRequest
from aliyunsdkecs.request.v20140526.ModifyInstanceAutoRenewAttributeRequest import \
ModifyInstanceAutoRenewAttributeRequest
from aliyunsdkecs.request.v20140526.RenewInstanceRequest import RenewInstanceRequest

logging.basicConfig(level=logging.INFO,
format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelname)s %(message)s',
datefmt='%a, %d %b %Y %H:%M:%S')

clt = client.AcsClient('Your Access Key Id', 'Your Access Key Secrect', 'cn-beijing')

# data format in UTC, only support passed the value for minute, seconds is not support.
INSTANCE_EXPIRED_START_TIME_IN_UTC_STRING = '2017-01-22T00:00Z'
INSTANCE_EXPIRE_END_TIME_IN_UTC_STRING = '2017-01-28T00:00Z'

def renew_job(page_size=100, page_number=1, check_need_renew=True, security_group_id=None):
```

```
response = describe_need_renew_instance(page_size=page_size, page_number=page_number,
check_need_renew=check_need_renew,
security_group_id=security_group_id)
response_list = response.get('Instances').get('Instance')
logging.info("%s instances need to renew", str(response.get('TotalCount')))
if response_list > 0:
instance_ids = ''
for item in response_list:
instance_id = item.get('InstanceId')
instance_ids += instance_id + ','
renew_instance(instance_id=instance_id)
logging.info("%s execute renew action ready", instance_ids)

def describe_need_renew_instance(page_size=100, page_number=1, instance_id=None,
check_need_renew=True, security_group_id=None):
request = DescribeInstancesRequest()
if check_need_renew is True:
request.set_Filter3Key("ExpiredStartTime")
request.set_Filter3Value(INSTANCE_EXPIRED_START_TIME_IN_UTC_STRING)
request.set_Filter4Key("ExpiredEndTime")
request.set_Filter4Value(INSTANCE_EXPIRE_END_TIME_IN_UTC_STRING)
if instance_id is not None:
request.set_InstanceIds(json.dumps([instance_id]))
if security_group_id:
request.set_SecurityGroupId(security_group_id)
request.set_PageNumber(page_number)
request.set_PageSize(page_size)
return _send_request(request)

# check the instances is renew or not
def describe_instance_auto_renew_setting(instance_ids, expected_auto_renew=True):
describe_request = DescribeInstanceAutoRenewAttributeRequest()
describe_request.set_InstanceId(instance_ids)
response_detail = _send_request(request=describe_request)
failed_instance_ids = ''
if response_detail is not None:
attributes = response_detail.get('InstanceRenewAttributes').get('InstanceRenewAttribute')
if attributes:
for item in attributes:
auto_renew_status = item.get('AutoRenewEnabled')
if auto_renew_status != expected_auto_renew:
failed_instance_ids += item.get('InstanceId') + ','
if len(failed_instance_ids) > 0:
logging.error("instance %s auto renew not match expect %s.", failed_instance_ids,
expected_auto_renew)

def setting_instance_auto_renew(instance_ids, auto_renew=True):
logging.info('execute enable auto renew ' + instance_ids)
request = ModifyInstanceAutoRenewAttributeRequest();
request.set_Duration(1);
request.set_AutoRenew(auto_renew);
request.set_InstanceId(instance_ids)
_send_request(request)
describe_instance_auto_renew_setting(instance_ids, auto_renew)

# if using the instance id can be found means the instance is not renew successfully.
```

```
def check_instance_need_renew(instance_id):
response = describe_need_renew_instance(instance_id=instance_id)
if response is not None:
return response.get('TotalCount') == 1
return False

# Renew an instance for a month
def renew_instance(instance_id, period='1'):
need_renew = check_instance_need_renew(instance_id)
if need_renew:
_renew_instance_action(instance_id=instance_id, period=period)
# describe_need_renew_instance(instance_id=instance_id, check_need_renew=False)

def _renew_instance_action(instance_id, period='1'):
request = RenewInstanceRequest()
request.set_Period(period)
request.set_InstanceId(instance_id)
response = _send_request(request)
logging.info('renew %s ready, output is %s ', instance_id, response)

def _send_request(request):
request.set_accept_format('json')
try:
response_str = clt.do_action(request)
logging.info(response_str)
response_detail = json.loads(response_str)
return response_detail
except Exception as e:
logging.error(e)

if __name__ == '__main__':
logging.info("Renew ECS Instance by OpenApi!")
# Query whether there is any instance that needs to be renewed within the specified time range.
describe_need_renew_instance()
# Renew an instance by direct fee deduction
renew_instance('i-1111')
# Query the status of automatic renewal
# describe_instance_auto_renew_setting('i-1111,i-2222')
# Set automatic instance renewal
# setting_instance_auto_renew('i-1111,i-2222')
```

If you want to learn other API operations in ECS, see ECS API operation.

Use APIs to manage spot instances

This document describes how to use Alibaba Cloud ECS SDKs to quickly create and manage spot instances.

Preparation

Before you begin, make sure that:

- You know which instance types and regions meet your business requirements.
- You have a basic understanding of Alibaba Cloud ECS SDKs and calling methods. For more information, see SDK documentation.

NOTE:
The ECS SDK version for spot instances is 4.2.0 and later. Here is an example of how to change the pom dependency.

```
<dependency>
<groupId>com.aliyun</groupId>
<artifactId>aliyun-java-sdk-core</artifactId>
<version>3.2.8</version>
</dependency>
<dependency>
<groupId>com.aliyun</groupId>
<artifactId>aliyun-java-sdk-ecs</artifactId>
<version>4.2.0</version>
</dependency>
```

Query regions and available instance types

Use the DescribeZones interface to query the regions where you can create spot instances and the available instance types. The sample code is as follows.

OpenApiCaller.java

```
public class OpenApiCaller {
IClientProfile profile;
IAcsClient client;
public OpenApiCaller() {
profile = DefaultProfile.getProfile("cn-hangzhou", AKSUtil.accessKeyId, AKSUtil.accessKeySecret);
client = new DefaultAcsClient(profile);
}
public <T extends AcsResponse> T doAction(AcsRequest<T> var1) {
try {
return client.getAcsResponse(var1);
} catch (Throwable e) {
e.printStackTrace();
return null;
}
}
}
```

DescribeZonesSample.java

```
public class DescribeZonesSample {
public static void main(String[] args) {
OpenApiCaller caller = new OpenApiCaller();
DescribeZonesRequest request = new DescribeZonesRequest();
request.setRegionId("cn-zhangjiakou");//You can use DescribeRegionsRequest to get the RegionId of each region
request.setSpotStrategy("SpotWithPriceLimit");//This field must be entered to query the availability of instance types
request.setInstanceChargeType("PostPaid");//Post-paid mode, spot instances must be post-paid
DescribeZonesResponse response = caller.doAction(request);
System.out.println(JSON.toJSONString(response));
}
}
```

In the following output result, you can see the instance types, disk types, and network types available in each region.

```
{
"requestId": "388D6321-E587-470C-8CFA-8985E2963DAE",
"zones": [
{
"localName": "China North 3 Zone A",
"zoneId": "cn-zhangjiakou-a",
"availableDiskCategories": [
"cloud_ssd",
"cloud_efficiency"
],
"availableInstanceTypes": [
"ecs.e4.large",
"ecs.n4.4xlarge",
"ecs.sn2.medium",
"ecs.i1.2xlarge",
"ecs.se1.2xlarge",
"ecs.n4.xlarge",
"ecs.se1ne.2xlarge",
"ecs.se1.large",
"ecs.sn2.xlarge",
"ecs.se1ne.xlarge",
"ecs.xn4.small",
"ecs.sn2ne.4xlarge",
"ecs.se1ne.4xlarge",
"ecs.sn1.medium",
"ecs.n4.8xlarge",
"ecs.mn4.large",
"ecs.e4.2xlarge",
"ecs.mn4.2xlarge",
"ecs.mn4.8xlarge",
"ecs.n4.2xlarge",
"ecs.e4.xlarge",
"ecs.sn2ne.large",
"ecs.sn2ne.xlarge",
```

```
"ecs.sn1ne.large",
"ecs.n4.large",
"ecs.sn1.3xlarge",
"ecs.e4.4xlarge",
"ecs.sn1ne.2xlarge",
"ecs.e4.small",
"ecs.i1.4xlarge",
"ecs.se1.4xlarge",
"ecs.sn2ne.2xlarge",
"ecs.sn2.3xlarge",
"ecs.i1.xlarge",
"ecs.n4.small",
"ecs.sn1ne.4xlarge",
"ecs.mn4.4xlarge",
"ecs.sn1ne.xlarge",
"ecs.se1ne.large",
"ecs.sn2.large",
"ecs.i1-c5d1.4xlarge",
"ecs.sn1.xlarge",
"ecs.sn1.large",
"ecs.mn4.small",
"ecs.mn4.xlarge",
"ecs.se1.xlarge"
],
"availableResourceCreation": [
"VSwitch",
"IoOptimized",
"Instance",
"Disk"
],
"availableResources": [
{
"dataDiskCategories": [
"cloud_ssd",
"cloud_efficiency"
],
"instanceGenerations": [
"ecs-3",
"ecs-2"
],
"instanceTypeFamilies": [
"ecs.mn4",
"ecs.sn1",
"ecs.sn2",
"ecs.sn1ne",
"ecs.xn4",
"ecs.i1",
"ecs.se1",
"ecs.e4",
"ecs.n4",
"ecs.se1ne",
"ecs.sn2ne"
],
"instanceTypes": [
"ecs.n4.4xlarge",
"ecs.sn2.medium",
```

```
"ecs.i1.2xlarge",
"ecs.se1.2xlarge",
"ecs.n4.xlarge",
"ecs.se1ne.2xlarge",
"ecs.se1.large",
"ecs.sn2.xlarge",
"ecs.se1ne.xlarge",
"ecs.xn4.small",
"ecs.sn2ne.4xlarge",
"ecs.se1ne.4xlarge",
"ecs.sn1.medium",
"ecs.n4.8xlarge",
"ecs.mn4.large",
"ecs.mn4.2xlarge",
"ecs.mn4.8xlarge",
"ecs.n4.2xlarge",
"ecs.sn2ne.large",
"ecs.sn2ne.xlarge",
"ecs.sn1ne.large",
"ecs.n4.large",
"ecs.sn1.3xlarge",
"ecs.sn1ne.2xlarge",
"ecs.e4.small",
"ecs.i1.4xlarge",
"ecs.se1.4xlarge",
"ecs.sn2ne.2xlarge",
"ecs.sn2.3xlarge",
"ecs.i1.xlarge",
"ecs.n4.small",
"ecs.sn1ne.4xlarge",
"ecs.mn4.4xlarge",
"ecs.sn1ne.xlarge",
"ecs.se1ne.large",
"ecs.sn2.large",
"ecs.i1-c5d1.4xlarge",
"ecs.sn1.xlarge",
"ecs.sn1.large",
"ecs.mn4.small",
"ecs.mn4.xlarge",
"ecs.se1.xlarge"
],
"ioOptimized": true,
"networkTypes": [
"vpc"
],
"systemDiskCategories": [
"cloud_ssd",
"cloud_efficiency"
]
}
],
"availableVolumeCategories": [
"san_ssd",
"san_efficiency"
]
}
```

```
]
}
```

Query spot instance price history

Use the DescribeSpotPriceHistory interface to query the price changes of a spot instance type over the last 30 days, so you can find the most cost efficient regions and instance types. The sample code (DescribeSpotPriceHistorySample.java) is as follows.

```
public class DescribeSpotPriceHistorySample {
public static void main(String[] args) {
OpenApiCaller caller = new OpenApiCaller();
List<DescribeSpotPriceHistoryResponse.SpotPriceType> result = new
ArrayList<DescribeSpotPriceHistoryResponse.SpotPriceType>();
int offset = 0;
while (true) {
DescribeSpotPriceHistoryRequest request = new DescribeSpotPriceHistoryRequest();
request.setRegionId("cn-hangzhou");//You can use DescribeRegionsRequest to get the RegionId of each region
where spot instances are available
request.setZoneId("cn-hangzhou-b");//You must enter the zone
request.setInstanceType("ecs.sn2.medium");//See the instance types returned by DescribeZones, this field is
mandatory
request.setNetworkType("vpc");//See the network types returned by DescribeZones, this field is mandatory
// request.setIoOptimized("optimized");//Determines if the instance is I/O optimized, see IoOptimized returned by
DescribeZones, this field is optional
// request.setStartTime("2017-09-20T08:45:08Z");//The price start time, optional, default value: within 3 days
// request.setEndTime("2017-09-28T08:45:08Z");//Price end time, optional
request.setOffset(offset);
DescribeSpotPriceHistoryResponse response = caller.doAction(request);
if (response != null && response.getSpotPrices() != null) {
result.addAll(response.getSpotPrices());
}
if (response.getNextOffset() == null || response.getNextOffset() == 0) {
break;
} else {
offset = response.getNextOffset();
}
}
if (!result.isEmpty()) {
for (DescribeSpotPriceHistoryResponse.SpotPriceType spotPriceType : result) {
System.out.println(spotPriceType.getTimestamp() + "--->spotPrice:" + spotPriceType.getSpotPrice() + "----
>originPrice:" + spotPriceType.getOriginPrice());
}
System.out.println(result.size());
} else {
}
}
}
```

Returned results sample.

```
2017-09-26T06:28:55Z--->spotPrice:0.24---->originPrice:1.2
```

```
2017-09-26T14:00:00Z--->spotPrice:0.36---->originPrice:1.2
2017-09-26T15:00:00Z--->spotPrice:0.24---->originPrice:1.2
2017-09-27T14:00:00Z--->spotPrice:0.36---->originPrice:1.2
2017-09-27T15:00:00Z--->spotPrice:0.24---->originPrice:1.2
2017-09-28T14:00:00Z--->spotPrice:0.36---->originPrice:1.2
2017-09-28T15:00:00Z--->spotPrice:0.24---->originPrice:1.2
2017-09-29T06:28:55Z--->spotPrice:0.24---->originPrice:1.2
```

Repeat this process to find the price trends and recent prices of the instance type in each zones.

> Note:
> You can use average price or maximum price to determine if you can afford this spot instance. You can also use more rational data models to analyze historical price data and adjust your instance types and zones at will for maximum cost effectiveness.

Create a spot instance

You must complete the following work before creating a spot instance:

- To use a custom image to create a spot instance, you must have already created the custom image.
- Create a security group in the console or use the CreateSecurityGroup to create a security group. Then, retrieve the security group ID (SecurityGroupId).
- In the console, create a VPC and VSwitch, or use the CreateVpc and CreateVSwitch interfaces to do so. Then, retrieve the VSwitch ID (VSwitchId).

Use the CreateInstance to create a spot instance. The sample code (CreateInstaneSample.java) is as follows.

```
public class CreateInstaneSample {
public static void main(String[] args) {
OpenApiCaller caller = new OpenApiCaller();
CreateInstanceRequest request = new CreateInstanceRequest();
request.setRegionId("cn-hangzhou");//The region ID
request.setZoneId("cn-hangzhou-b"); //The zone ID
request.setSecurityGroupId("sg-bp11nhf94ivkdxwb2gd4");//The ID of the security group
request.setImageId("centos_7_03_64_20G_alibase_20170818.vhd");//We recommend that you select a custom image you have prepared in this region
request.setVSwitchId("vsw-bp164cyonthfudn9kj5br");//For VPC, the VSwitch ID is required
request.setInstanceType("ecs.sn2.medium"); //Enter the instance type you want to purchase
request.setIoOptimized("optimized");//See the parameters returned by DescirbeZones
request.setSystemDiskCategory("cloud_ssd");//See the parameters returned by DescirbeZones, select one: cloud_ssd, cloud_efficiency, or cloud
request.setSystemDiskSize(40);
request.setInstanceChargeType("PostPaid");//Post-paid is required for spot instances
request.setSpotStrategy("SpotWithPriceLimit");//SpotWithPriceLimit: bid mode, SpotAsPriceGo: no bids, the maximum Pay-As-You-Go price
request.setSpotPriceLimit(0.25F);//This applies only to SpotWithPriceLimit. Set the maximum price you are willing to pay, units: USD/hour. An instance is created when this price is higher than the current market price
```

```
CreateInstanceResponse response = caller.doAction(request);
System.out.println(response.getInstanceId());
}
}
```

Recover a spot instance

Mandatory recovery can be imposed on spot instances due to changes in price or supply and demand. At such a time, the operation of the spot instance is suspended. Before being released, the spot instance enters the locked status and a prompt notifies you that the instance will be automatically recovered. You can design a withdrawal logic to automatically process instances in the recovery status.

Now, you can use the following methods to determine the suspension and lock statuses of spot instances:

Obtain this information from the instance metadata. Run the following command.

```
curl 'http://100.100.100.200/latest/meta-data/instance/spot/termination-time'
```

If no result is returned, it indicates the instance can continue to be used. If the returned result contains UTC time stamp information in the format of YYYY-MM-DDTHH:mm:ssZ (for example 2015-01-05T18:02:00Z), it indicates the instance will be released at the specified time.

You can use the OperationLocks information returned by the DescribeInstances to see if an instance is in the **Awaiting Recovery** status. The sample code (DescribeInstancesSample.java) is as follows.

```
public class DescribeInstancesSample {
public static void main(String[] args) throws InterruptedException {
OpenApiCaller caller = new OpenApiCaller();
JSONArray allInstances = new JSONArray();
allInstances.addAll(Arrays.asList("i-bp18hgfai8ekoqwo0y2n", "i-bp1ecbyds24ij63w146c"));
while (!allInstances.isEmpty()) {
DescribeInstancesRequest request = new DescribeInstancesRequest();
request.setRegionId("cn-hangzhou");
request.setInstanceIds(allInstances.toJSONString());//Specify the instance ID, maximum efficiency
DescribeInstancesResponse response = caller.doAction(request);
List<DescribeInstancesResponse.Instance> instanceList = response.getInstances();
if (instanceList != null && !instanceList.isEmpty()) {
for (DescribeInstancesResponse.Instance instance : instanceList) {
System.out.println("result:instance:" + instance.getInstanceId() + ",az:" + instance.getZoneId());
if (instance.getOperationLocks() != null) {
for (DescribeInstancesResponse.Instance.LockReason lockReason : instance.getOperationLocks()) {
System.out.println("instance:" + instance.getInstanceId() + "-->lockReason:" +
lockReason.getLockReason() + ",vmStatus:" + instance.getStatus());
```

```
if ("Recycling".equals(lockReason.getLockReason())) {
//do your action
System.out.println("spot instance will be recycled immediately, instance id:" + instance.getInstanceId());
allInstances.remove(instance.getInstanceId());
}
}
}
}
System.out.print("try describeInstances again later ...");
Thread.sleep(2 * 60 * 1000);
} else {
break;
}
}
}
}
```

The output result when recovery is triggered is as follows.

```
instance:i-bp1ecbyds24ij63w146c-->lockReason:Recycling,vmStatus:Stopped
spot instance will be recycled immediately, instance id:i-bp1ecbyds24ij63w146c
```

Other operations

You can start, stop, and release spot instances. These operations are the same as for Pay-As-You-Go instances. For more information, see the API documentation:

- Start an instance: StartInstance
- Stop an instance: StopInstance
- Release an instance: DeleteInstance

User-defined data

Customize yum repository, NTP service, and DNS service

User-defined scripts are a type of script provided by Alibaba Cloud for users to customize the startup behaviors of ECS instances. For details, see User-defined data.

This example uses a Linux instance to demonstrate how to use a user-defined script to configure your own yum repository, NTP service, and DNS service when creating a Linux instance. User-defined scripts also enable you to configure NTP service and DNS service for a Windows instance.

Scenarios

When a Linux instance is started, Alibaba Cloud automatically configures a pre-defined yum repository, NTP service, and DNS service for the instance. However, if you want to have your own yum repository, NTP service, and DNS service, use user-defined scripts to implement them.

Note:

- If you are using a custom yum repository, Alibaba Cloud does not provide support for it.
- If you are using a custom NTP service, Alibaba Cloud does not provide time service.

Procedure

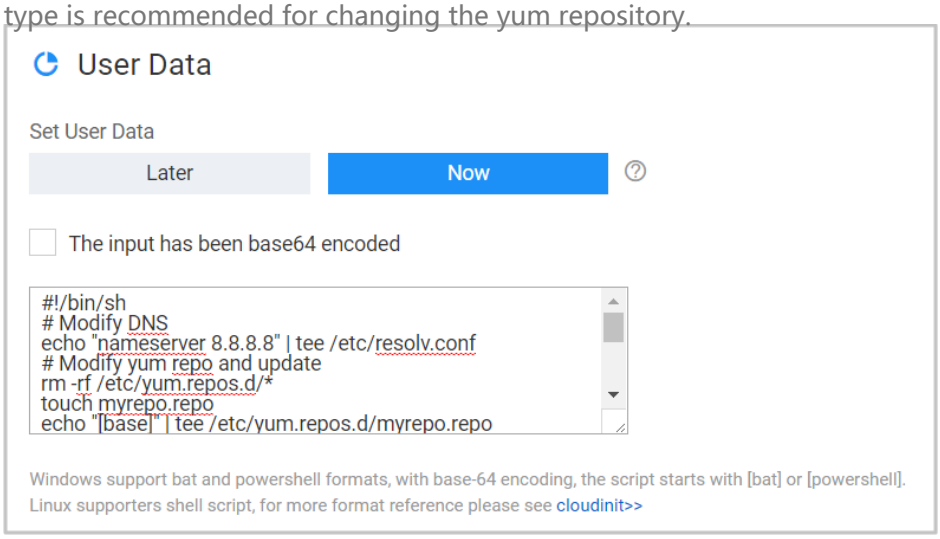
To customize your yum repository, NTP service, and DNS service for a Linux instance when creating it, follow these steps:

1. Log on to the ECS console and create an instance. Configure the instance as follows:
   - **Network Type**: Select **VPC**.
   - **Instance Type**: Select an I/O-optimized instance.
   - **Operating System**: Select CentOS 7.2 in **Public Image** tab.
   - **Security Setup**: Select one to meet your requirements.
2. Enter the following script in the **User Data** box on the instance creation page.

```
#!/bin/sh
# Modify DNS
echo "nameserver 8.8.8.8" | tee /etc/resolv.conf
# Modify yum repo and update
rm -rf /etc/yum.repos.d/*
touch myrepo.repo
echo "[base]" | tee /etc/yum.repos.d/myrepo.repo
echo "name=myrepo" | tee -a /etc/yum.repos.d/myrepo.repo
echo "baseurl=http://mirror.centos.org/centos" | tee -a /etc/yum.repos.d/myrepo.repo
echo "gpgcheck=0" | tee -a /etc/yum.repos.d/myrepo.repo
echo "enabled=1" | tee -a /etc/yum.repos.d/myrepo.repo
yum update -y
# Modify NTP Server
echo "server ntp1.aliyun.com" | tee /etc/ntp.conf
systemctl restart ntpd.service
```

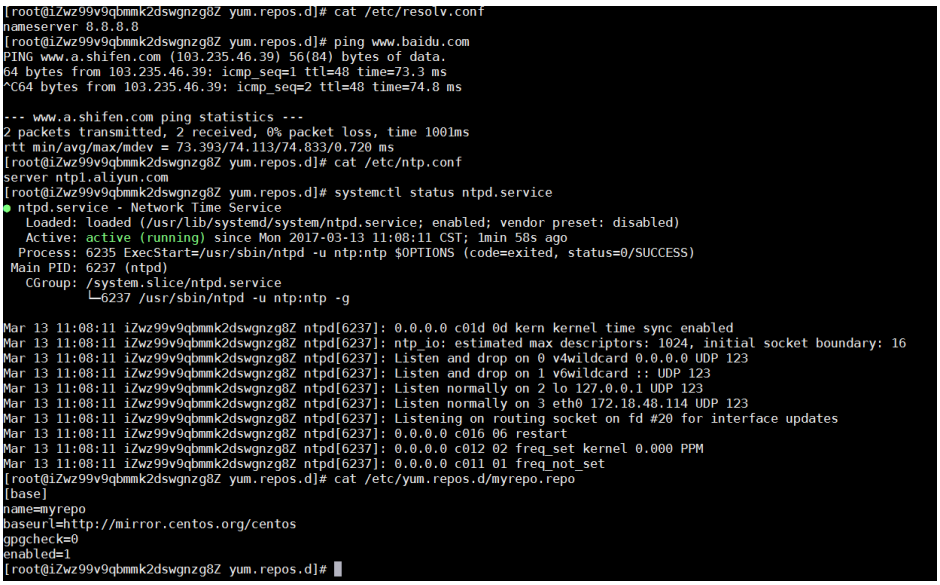
Note:

- The first line must be #!/bin/sh, with no leading space.
- Do not add unnecessary spaces or carriage return characters in the full text.
- You can customize URLs of your own DNS server, NTP Server, and yum repository based on the instance situations.
- The preceding content applies to CentOS 7.2. If you are using other images, modify the scripts as needed.
- You can also define the yum repository in the scripts of the Cloud Config type, but it is not recommended because it is not flexible enough to get adapted to Alibaba Cloud that may pre-configure some yum repository. Scripts of script type is recommended for changing the yum repository.



After you complete the configuration, click **Buy Now** and activate the instance following the instructions on the page.

After the instance is created, you can connect to the instance to view the implementation details, as shown in the following figure.



The preceding figure shows that you have successfully customized the DNS service, the NTP service, and the yum repository.

Create a new account with the root user privilege

User-defined scripts are a type of script provided by Alibaba Cloud to enable users to customize the startup behavior of ECS instances. For details, see User-defined data.

This example uses a Linux instance to demonstrate how to use a user-defined script to create a new account, with the root user privilege, when creating a Linux instance. User-defined scripts can also be used to create a new account with the administrator privilege for a Windows instance.

Scenarios

Use user-defined scripts of instances if you want to achieve the following results when creating a Linux ECS instance:

- Disable the default **root** account that comes with a Linux ECS instance. You can use the script to customize how to disable the root user and how many root user privileges are disabled.
- Create a new account with the root user privilege and customize the account name.
- Use only SSH key pairs, but not user passwords, for remote logon to manage the instance by using the new account with the root user privilege.
- If this new account is required to perform operations that can only be done by a user with root user privilege, the sudo command can be used without a password for privilege

escalation.

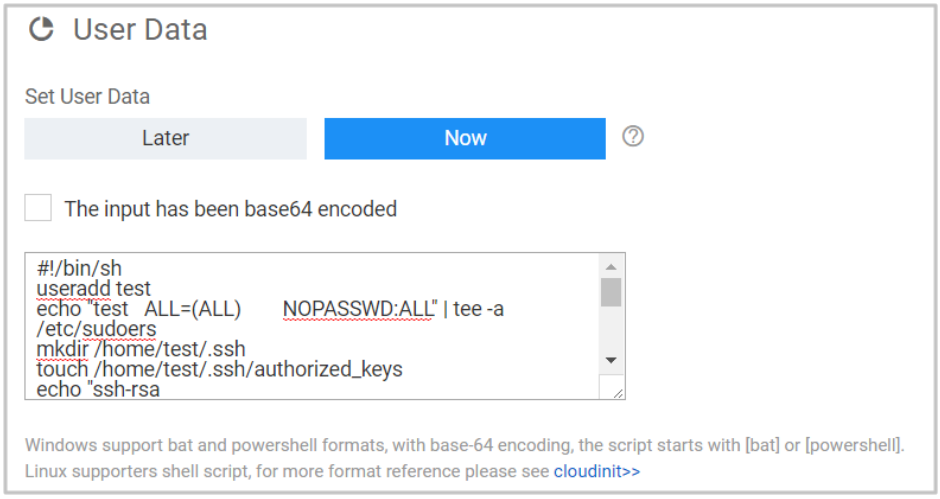
Procedure

To create a new account with the root user privilege, follow these steps:

1. Create a Linux instance. Configure the instance as follows:
   - **Network Type**: Select **VPC**.
   - **Instance Type**: Select an I/O-optimized instance.
   - **Operating System**: Select CentOS 7.2 in **Public Image** tab.
   - **Security Setup**: select **Later**.
2. Enter the following script in the **User Data** box on the instance creation page:

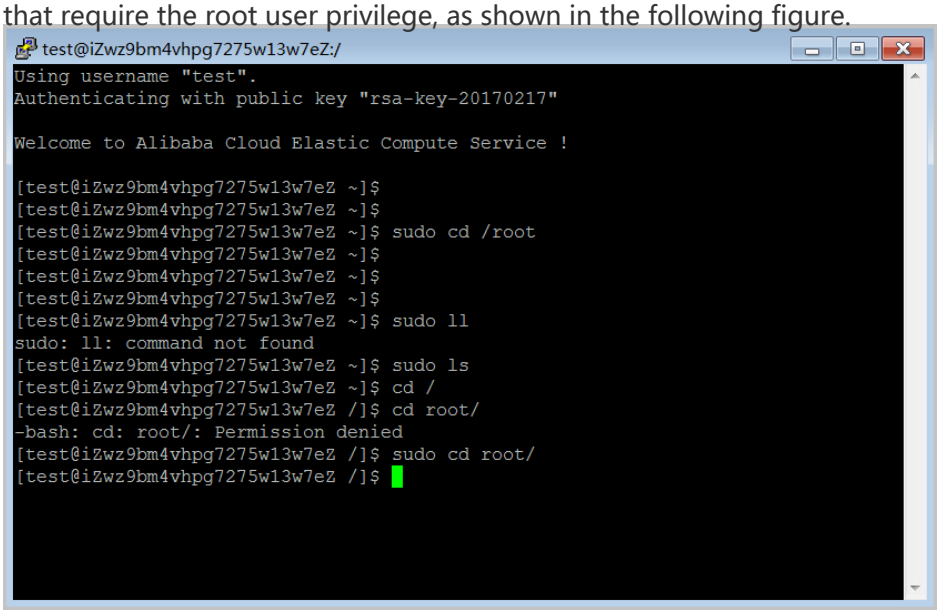
```
#!/bin/sh
useradd test
echo "test ALL=(ALL) NOPASSWD:ALL" | tee -a /etc/sudoers
mkdir /home/test/.ssh
touch /home/test/.ssh/authorized_keys
echo "ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAhGqhEh/rGbIMCGItFVtYpsXPQrCaunGJKZVIWtINrGZwusLc290qDZ
93KCeb8o6X1Iby1Wm+psZY8THE+/BsXq0M0HzfkQZD2vXuhRb4xi1z98JHskX+0jnbjqYGY+Brgai9BvKDX
TTSyJtCYUnEKxvcK+d1ZwxbNuk2QZ0ryHESDbSaczlNFgFQEDxhCrvko+zWLjTVnomVUDhdMP2g6fZ0tgF
VwkJFV0bE7oob3NOVcrx2TyhfcAjA4M2/Ry7U2MFADDC+EVkpoVDm0SOT/hYJgaVM1xMDlSeE7kzX7yZ
bJLR1XAWV1xzZkNclY5w1kPnW8qMYuSwhpXzt4gsF0w== rsa-key-20170217" | tee -a
/home/test/.ssh/authorized_keys
```

   Note:
   - The first line must be #!/bin/sh with no leading space.
   - Do not enter unnecessary spaces or carriage return characters in the text.
   - The last line is your public key. You can define it.
   - You can add other configuration in the script, as you need.
   - The example script only applies to CentOS 7.2. If you are using other images, customize the script according to the operating system types.



After you finish the configuration, click **Buy Now** and activate the instance by following instructions on the page.

After the instance is created, you can use the new **test** user to connect to the instance using an SSH private key. You can also escalate the permission level using the sudo command and run operations that require the root user privilege, as shown in the following figure.



Access APIs of other cloud products by using the instance RAM role

Overview

Previously, applications deployed on an ECS Instance usually needed to use AccessKey ID and AccessKey Secret (AK) to access APIs of other Alibaba Cloud products. AK is the key to accessing Alibaba Cloud APIs and has all of the permissions of the corresponding accounts. To help applications manage the AK, you have to save AK in the configuration files of the application or save it in an ECS instance by using other methods, which makes it more complicated to manage the AK and reduces its confidentiality. What' s more, if you need concurrent deployment across regions, the AK is diffused along with the images or instances created by the image, which makes you have to update and re-deploy the instances and images one by one when changing the AK.

Now with the help of the instance RAM role, you can assign a RAM role to an ECS instance. The applications on the instance can then access APIs of other cloud products with the STS credential. The STS credential is automatically generated and updated by the system, and the applications can use the specified meta data URL to obtain the STS credential without special management. Meanwhile, you can modify the RAM role and the authorization policy to grant different or identical access permissions to an instance to different Alibaba Cloud products.

This article introduces how to create an ECS instance that plays a RAM role and how to enable applications on the ECS instance to access other Alibaba Cloud products with the STS credential. In this section, using Python on an ECS instance to access an OSS bucket is used as the example.

Procedure

To enable python on an instance to access an OSS bucket under the same account by using the instance RAM role, follow these steps:

Step 1. Create a RAM role and attach it to an authorization policy.
Step 2. Create an ECS instance playing the RAM role to create.
Step 3. Within the instance, access the metadata URL to obtain the STS credential.
Step 4. Use Python to access OSS using the STS credential.

Step 1. Create a RAM role and attach it to an authorization policy

Use the CreateRole API to create a RAM role. The required request parameters are:

- **RoleName**: Specify a name for the role. *EcsRamRoleTest* is used in this example.
- **AssumeRolePolicyDocument**: Specify a policy as follows, which indicates that the role to be created is a service role and an Alibaba Cloud product (ECS in this example) is assigned to play this role.

```
{
"Statement": [
```

```
{
"Action": "sts:AssumeRole",
"Effect": "Allow",
"Principal": {
"Service": [
"ecs.aliyuncs.com"
]
}
}
],
"Version": "1"
}
```

Use the CreatePolicy API to create an authorization policy. The required request parameters are:

- **PolicyName**: Specify a name for the authorization policy. *EcsRamRolePolicyTest* is used in this example.
- **PolicyDocument**: Specify a policy as follows, which indicates that the role has OSS read-only permission.

```
{
"Statement": [
{
"Action": [
"oss:Get*",
"oss:List*"
],
"Effect": "Allow",
"Resource": "*"
}
],
"Version": "1"
}
```

Use the AttachPolicyToRole API to attach the authorization policy to the role. The required request parameters are:

- **PolicyType**: Set it to *Custom*.
- **PolicyName**: Use the policy name specified in step 2. Use *EcsRamRolePolicyTest* in this example.
- **RoleName**: Use the role name specified in step 1. Use *EcsRamRoleTest* in this example.

Step 2. Create an ECS instance playing the RAM role

You can use either method to create an ECS instance playing the RAM role:

- Attach a RAM role to an existing VPC instance

- Create a VPC instance with the RAM role

Attach a RAM role to an existing VPC instance

Use the AttachInstanceRamRole API to attach a RAM role to an existing VPC instance. The parameters are as follows:

- **RegionId**: The ID of the region where the instance is located.
- **RamRoleName**: The name of a RAM role. In this example, *EcsRamRoleTest* is used.
- **InstanceIds**: The IDs of VPC instances that you want to attach the RAM role to, in the format of ["i-bXXXXXXXX"] for one instance, or ["i-bXXXXX" , "i-cXXXXX" , "i-dXXXXX" ...] for multiple instances.

Create a VPC ECS instance with the RAM role

You must have a VPC network before creating an ECS instance with the RAM role.

To create a VPC instance with the RAM role, follow these steps:

Use the CreateInstance API to create an ECS instance. The required request parameters are:

- **RegionId**: The region of the instance. In this example, *cn-hangzhou* is used.
- **ImageId**: The image of the instance. In this example, *centos_7_03_64_40G_alibase_20170503.vhd* is used.
- **InstanceType**: The type of the instance. In this example, *ecs.xn4.small* is used.
- **VSwitchId**: The virtual switch of the VPC network where the instance is located.

  > Because the instance RAM role only supports VPC network, VSwitchId is required.

- **RamRoleName**: The name of RAM Role. In this example, *EcsRamRoleTest* is used.

  > If you want to authorize a sub account to create an ECS instance playing the specified RAM role, besides the permission to create an ECS instance, the sub account must have the PassRole permission. Therefore, you must customize an authorization policy as follows and attach it to the sub account. If the action is creating an ECS instance only, set *[ECS RAM Action]* to ecs:CreateInstance. You can grant more permissions to meet your needs. For more information, see Actions in RAM that can be authorized to an ECS instance. If you want to grant all ECS action permissions to the sub account, set *[ECS RAM Action]* to ecs:*.

```
{
"Statement": [
{
"Action": "[ECS RAM Action]",
"Resource": "*",
"Effect": "Allow"
```

```
},
{
"Action": "ram:PassRole",
"Resource": "*",
"Effect": "Allow"
],
"Version": "1"
}
```

Set the password and start the instance.

Set the ECS instance to access the Internet by using API or in the ECS console.

> To set an ECS instance in a VPC network to access the Internet on a console, see Bind an Elastic IP address (EIP) in the *Quick Start* of Virtual Private Cloud.

Step 3. Access the metadata URL within the instance to obtain the STS credential

To obtain the STS credential of the instance, follow these steps:

Connect to the instance.

Access the following URL to obtain the STS credential.
http://100.100.100.200/latest/meta-data/ram/security-credentials/EcsRamRoleTest
The last part of the URL is the RAM role name, which must be replaced with the one you create.

> In this example, we run the curl command to access the URL. If you are using a Windows ECS instance, see Use metadata of an instance in ECS the *User Guide* to obtain the STS credential.

The return parameters are as follows.

```
[root@local ~]# curl http://100.100.100.200/latest/meta-data/ram/security-credentials/EcsRamRoleTest
{
"AccessKeyId" : "STS.J8XXXXXXXXXX4",
"AccessKeySecret" : "9PjfXXXXXXXXXBf2XAW",
"Expiration" : "2017-06-09T09:17:19Z",
"SecurityToken" : "CAIXXXXXXXXXXXwmBkleCTkyI+",
"LastUpdated" : "2017-06-09T03:17:18Z",
"Code" : "Success"
}
```

Step 4. Use Python SDK to access OSS with the STS credential

In this example, with the STS credential, we use Python to list 10 files in an OSS bucket that is in the same region with the instance.

Prerequisites

- You have remotely connected to the ECS instance.
- Python has been installed on the ECS instance. If you are using a Linux ECS instance, pip must be installed.
- A bucket has been created in the region of the instance, and the bucket name and the Endpoint have been acquired. In this example, the bucket name is *ramroletest*, and the endpoint is *oss-cn-hangzhou.aliyuncs.com*.

Procedure

To use Python to access the OSS bucket, follow these steps:

Run the command pip install oss2 to install OSS Python SDK.

If you are using a Windows ECS instance, see Installation in the *Python-SDK* Reference of Object Storage Service.

Run the following commands to test, of which:

- The three parameters in oss2.StsAuth must be set to the values of the return parameters: AccessKeyId, AccessKeySecret, and SecurityToken.
- The last two parameters in oss2.Bucket are the bucket name and the endpoint.

```
import oss2
from itertools import islice
auth = oss2.StsAuth(<AccessKeyId>, <AccessKeySecret>, <SecurityToken>)
bucket = oss2.Bucket(auth, <your Endpoint>, <your Bucket name>)
for b in islice(oss2.ObjectIterator(bucket), 10):
print(b.key)
```

The output result is displayed as follows.

```
[root@local ~]# python
Python 2.7.5 (default, Nov 6 2016, 00:28:07)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-11)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import oss2
```

```
>>> from itertools import islice
>>> auth = oss2.StsAuth("STS.J8XXXXXXXXXX4", "9PjfXXXXXXXXXBf2XAW",
"CAIXXXXXXXXXXXwmBkleCTkyI+")
>>> bucket = oss2.Bucket(auth, "oss-cn-hangzhou.aliyuncs.com", "ramroletest")
>>> for b in islice(oss2.ObjectIterator(bucket), 10):
... print(b.key)
...
ramroletest.txt
test.sh
```

FaaS instances best practices

Use OpenCL on an f1 instance

This article introduces how to use Open Computing Language (OpenCL) to create an image file, and then download the image to an FPGA chip.

> **Note**:
> We strongly recommend that you use an f1 instance as a RAM user. To avoid unwanted operations, you must authorize the RAM user to perform essential actions only. You must create a role for the RAM user and grant temporary permissions to the role to access the OSS buckets. If you want to encrypt the IP address, grant the RAM user to use Key Management Service (KMS). If you want the RAM user to check permissions, authorize the RAM user to view the resources of an account.

Prerequisites

Before use OpenCL to create an image file, you must finish the following operations:

- Create an f1 instance.
- If you operate an f1 instance as a RAM user, you must do the following operations:
  - Create a RAM user and grant permissions.
  - Create a RAM role and grant permissions.
  - Create an AccessKey.
- Log on to the ECS console, and in the instance detail page of the f1 instance, obtain the instance ID.

Procedure

To configure the environment of FPGA Server Example, follow these steps.

Step 1. Connect to your f1 instance

Connect to the Linux instance.

Step 2. Install the basic environment

Run the script to install the basic environment.

```
source /opt/dcp1_0/script/f1_env_set.sh
```

Step 3. Download the OpenCL Example

To download the OpenCL Example, follow these steps:

Create the /opt/tmp directory, and change the current directory to it.

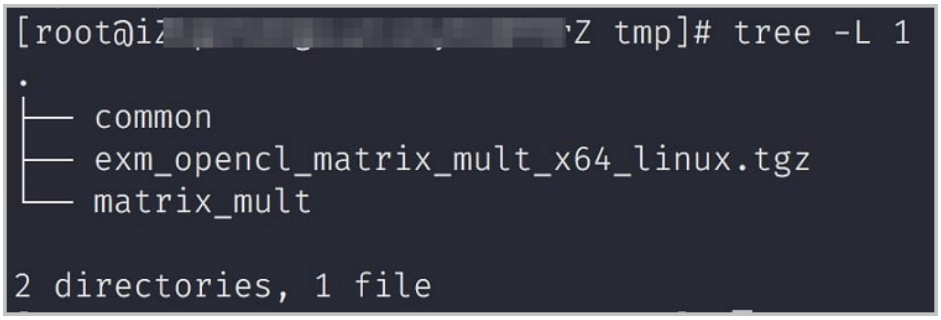
```
mkdir -p /opt/tmp
cd /opt/tmp
```

Now, you are at the /opt/tmp directory.

Run the commands one by one to download and decompress the OpenCL Example file.

```
wget https://www.altera.com/content/dam/altera-
www/global/en_US/others/support/examples/download/exm_opencl_matrix_mult_x64_linux.tgz
tar -zxvf exm_opencl_matrix_mult_x64_linux.tgz
```

The following figure displays the directory after decompression.



Change the current directory to the matrix_mult directory and run the command for compilation.

```
cd matrix_mult
aoc -v -g --report ./device/matrix_mult.cl
```

The process of compilation takes several hours. You can open a new console, and run the top command to monitor processes and system resource usage on the instance and view the status of the compilation process.

Step 4. Upload the configuration file to the OSS bucket

Run the commands to initialize the faascmd.

```
# If needed, add the environment variable and grant the permission to run the commands
export PATH=$PATH:/opt/dcp1_0/script/
chmod +x /opt/dcp1_0/script/faascmd
# Replace hereIsMySecretId with your AccessKey ID. Replace hereIsMySecretKey with your AccessKey Secret
faascmd config --id=hereIsMySecretId --key=hereIsMySecretKey
# Replace hereIsMyBucket with the bucket name of your OSS in the Region China East 1.
faascmd auth --bucket=hereIsMyBucket
```

Change the current directory to the matrix_mult/output_files directory, and upload the configuration file.

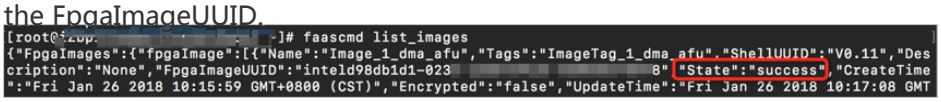
```
cd matrix_mult/output_files # Now you are accessing /opt/tmp/matrix_mult/matrix_mult/output_files
faascmd upload_object --object=afu_fit.gbs --file=afu_fit.gbs
```

Use gbs to create an FPGA image.

```
# Replace hereIsFPGAImageName with your image name. Replace hereIsFPGAImageTag with the tag of your image.
faascmd create_image --object=afu_fit.gbs --fpgatype=intel --name=hereIsFPGAImageName --
```

```
tags=hereIsFPGAImageTag --encrypted=false --shell=V0.11
```

Run the faascmd list_images command to check whether the image is created.
In the returned result, if "State":"success" is displayed, it means the image is created. Record the FpgaImageUUID.



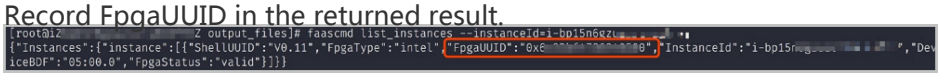
Step 5. Download the image to your f1 instance

To download the image to your f1 instance, follow these steps:

Run the command to obtain FPGA ID.

```
# Replace hereIsYourInstanceId with your f1 instance ID.
faascmd list_instances --instanceId=hereIsYourInstanceId
```

Record FpgaUUID in the returned result.



Run the command to download the image to your f1 instance.

```
# Replace hereIsYourInstanceID with your f1 instance ID. Replace hereIsFpgaUUID with your FpgaUUID.
Replace hereIsImageUUID with your FpgaImageUUID.
faascmd download_image --instanceId=hereIsYourInstanceID --fpgauuid=hereIsFpgaUUID --
fpgatype=intel --imageuuid=hereIsImageUUID --imagetype=afu --shell=V0.11
```

Run the command to check whether the image is downloaded.

```
# Replace hereIsYourInstanceID with your f1 instance ID. Replace hereIsFpgaUUID with your FpgaUUID.
faascmd fpga_status --fpgauuid=hereIsFpgaUUID --instanceId=hereIsYourInstanceID
```

If "TaskStatus": "operating" exists in the returned result, it means the image is downloaded.

Step 6. Download the FPGA image to an FPGA chip

To download the FPGA image to an FPGA chip, follow these steps:

Open the console in Step 1. If it is closed, repeat Step 1.

Run the following command to configure the runtime environment for OpenCL.

```
sh /opt/dcp1_0/opencl/dcp_opencl_bsp/linux64/libexec/setup_permissions.sh
```

Run the command to go back to the parent directory.

```
cd ../.. # Now, you are at the /opt/tmp/matrix_mult directory
```

Run the command to compile.

```
make
# Output the environment configuration
export CL_CONTEXT_COMPILER_MODE_ALTERA=3
cp matrix_mult.aocx ./bin/matrix_mult.aocx
cd bin
host matrix_mult.aocx
```

If the following result is returned, it means the configuration is successful. Note that the last line must be Verification: PASS.

```
[root@iZbpXXXXXZ bin]# ./host matrix_mult.aocx
Matrix sizes:
A: 2048 x 1024
B: 1024 x 1024
C: 2048 x 1024
Initializing OpenCL
Platform: Intel(R) FPGA SDK for OpenCL(TM)
Using 1 device(s)
skx_fpga_dcp_ddr : SKX DCP FPGA OpenCL BSP (acl0)
Using AOCX: matrix_mult.aocx
Generating input matrices
Launching for device 0 (global size: 1024, 2048)
Time: 40.415 ms
Kernel time (device 0): 40.355 ms
Throughput: 106.27 GFLOPS
Computing reference output
Verifying
Verification: PASS
```

Download a bitstream file to an FPGA chip

This document describes how to generate and download a custom bitstream file to a specified FPGA chip of an f1 instance.

> **Note**:
> We strongly recommend that you use an f1 instance as a RAM user. To avoid unwanted operations, the RAM user must be authorized to perform essential actions only. You must create a role for the RAM user and grant temporary permissions to the role to access the OSS buckets. If you want to encrypt the IP addresses, authorize the RAM user to use Key Management Service (KMS). If you want the RAM user to check permissions, authorize the RAM user to view the resources.

Prerequisites

Before you start, finish the following operations:

- Create an f1 instance.
- Activate OSS to upload your custom bitstream files.
- If you want to encrypt your bitstream, activate Key Management Service (KMS).
- If you operate an f1 instance as a RAM user, you must do the following operations:
  - Create a RAM user and grant permissions.
  - Create a RAM role and grant permissions.
  - Create an AccessKey.

Procedure

To generate and download a bitstream file, follow these steps.

Step 1. Generate a bitstream file

To generate a bistream file, follow these steps:

1. Upload the project file to the specified OSS bucket. The RAM user must have the permission to access the OSS bucket.
   - If you are using an Intel FPGA chip, you must upload the final gbs file to your OSS bucket.
   - If you are using a Xilinx FPGA chip, you must upload the DCP file to your OSS bucket.
2. Call the CreateFpgaImageTask interface in the Python SDK to create a bitstream file. See the following example code.

```
from aliyunsdkcore import client
clt = client.AcsClient(<Your AccessKey ID>,<Your AccessKey Secret>,'cn-hangzhou')
from aliyunsdkfaas.request.v20170824 import CreateFpgaImageTaskRequest
request = CreateFpgaImageTaskRequest.CreateFpgaImageTaskRequest()
request.set_Bucket(<The OSS bucket for the DCP or bitstream file>)
request.set_Object(<The object name of the DCP or bitstream file>)
request.set_FpgaType(<Fpga type>)
request.set_ShellUUID(<shell type>)
request.set_Name(<Specify a name for the image to ease management>)
request.set_RoleArn(<Create a role for faas-admin>)
request.set_Encrypted(<Encrypted or not, True/False>)
request.set_KeyId(<If encrypted, specify the ID of the key in the KMS>)
result = clt.do_action_with_exception(request)
print result
```

Call the DescribeFpgaImages interface in the Python SDK to check whether the bitstream is generated.

> **Note**:
> CreateFpgaImageTask is an asynchronous action. After you send the request, the back-end server performs the security check. If you are using a Xilinx project, the back-end server has to generate a bitstream file from the DCP project, which takes some time.

See the following example code.

```
from aliyunsdkcore import client
clt = client.AcsClient(<Your AccessKey ID>,<Your AccessKeySecret>,'cn-hangzhou')
from aliyunsdkfaas.request.v20170824 import DescribeFpgaImagesRequest
request = DescribeFpgaImagesRequest.DescribeFpgaImagesRequest()
result = clt.do_action_with_exception(request)
print result
```

Step 2. Download the bitstream file

To download the bitstream file to the specified FPGA chip, follow these steps:

Call the DescribeFpgaInstances interface in the Python SDK to check the FpgaUUID associated with your f1 instance. FpgaUUID is the unique identifier of an FPGA chip. See the following example code.

```
from aliyunsdkcore import client
clt = client.AcsClient(<Your AccessKey ID>,<Your AccessKey Secret>,'cn-hangzhou')
from aliyunsdkfaas.request.v20170824 import DescribeFpgaInstancesRequest
request = DescribeFpgaInstancesRequest.DescribeFpgaInstancesRequest()
request.set_InstanceId(<Specify the instance ID>)
request.set_RoleArn(<Create a role for faas-admin>)
```

```
result = clt.do_action_with_exception(request)
print result
```

Call the DescribeFpgaImages interface to view the information about the bitstream file under your account. The unique identifier of your bitstream file is FpgaImageUUID. See the following example code.

```
from aliyunsdkcore import client
clt = client.AcsClient(<Your AccessKey ID>,<Your AccessKey Secret>,'cn-hangzhou')
from aliyunsdkfaas.request.v20170824 import DescribeFpgaImagesRequest
request = DescribeFpgaImagesRequest.DescribeFpgaImagesRequest()
result = clt.do_action_with_exception(request)
print result
```

Call the LoadFpgaImageTask interface in the Python SDK to download the specified bitstream file to the specified FPGA chip. See the following example code.

```
from aliyunsdkcore import client
clt = client.AcsClient(<Your AccessKey ID>,<Your AccessKey Secret>,'cn-hangzhou')
from aliyunsdkfaas.request.v20170824 import LoadFpgaImageTaskRequest
request = LoadFpgaImageTaskRequest.LoadFpgaImageTaskRequest()
request.set_InstanceId(<Specify the instance ID>)
request.set_FpgaUUID(<Specify the FPGA chip>)
request.set_FpgaType(<Fpga type>)
request.set_FpgaImageUUID(<The UUID of the image to be downloaded>)
request.set_FpgaImageType(<Image type>)
request.set_ShellUUID(<Specify shell>)
request.set_RoleArn(<Create a role for faas-admin>)
result = clt.do_action_with_exception(request)
print result
```

Call the DescribeLoadTaskStatus interface in the Python SDK to check whether the bitstream file is downloaded. See the following example code.

```
from aliyunsdkcore import client
clt = client.AcsClient(<Your AccessKey ID>,<Your AccessKey Secret>,'cn-hangzhou')
from aliyunsdkfaas.request.v20170824 import DescribeLoadTaskStatusRequest
request = DescribeLoadTaskStatusRequest.DescribeLoadTaskStatusRequest()
request.set_FpgaUUID(<Specify the FPGA chip>)
request.set_InstanceId(<Specify the instance ID>)
request.set_RoleArn(<Create a role for faas-admin>)
result = clt.do_action_with_exception(request)
print result
```

Use RTL compiler on an f1 instance

This article describes how to use Register Transfer Level (RTL) compiler on an f1 instance.

> **Note**:
> We strongly recommend that you use an f1 instance as a RAM user. To avoid unwanted operations, you must authorize the RAM user to perform essential actions only. You must create a role for the RAM user and grant temporary permissions to the role to access the OSS buckets. If you want to encrypt the IP address, grant the RAM user to use Key Management Service (KMS). If you want the RAM user to check permissions, authorize the RAM user to view the resources of an account.

Prerequisites

Before you start, you must finish the following operations:

- Create an f1 instance.
- If you operate an f1 instance as a RAM user, you must do the following operations:
  - Create a RAM user and grant permissions.
  - Create a RAM role and grant permissions.
  - Create an AccessKey.
- Log on to the ECS console, and in the instance detail page of the f1 instance, obtain the instance ID.

Procedure

To use RTL compiler on an f1 instance, follow these steps.

Step 1. Connect to the f1 instance

Connect to your f1 instance.

Step 2. Configure the basic environment

Run the script to configure the basic environment.

```
source /opt/dcp1_0/script/f1_env_set.sh
```

Step 3. Compile the project

Run the following commands to compile the project.

```
cd /opt/dcp1_0/hw/green_bits/dma_afu/src
run.sh
```

Note:
It takes a long time to compile the project.

Step 4. Create an image

To create an image, follow these steps:

1. Run the following commands to initialize faascmd.

```
# If needed, add the environment variable and grant permission to run the commands.
export PATH=$PATH:/opt/dcp1_0/script/
chmod +x /opt/dcp1_0/script/faascmd
# Replace hereIsMySecretId with your AccessKey ID. Replace hereIsMySecretKey with your AccessKey Secret.
faascmd config --id=hereIsMySecretId --key=hereIsMySecretKey
# Replace hereIsMyBucket with the OSS bucket name in the China East 1 region.
faascmd auth --bucket=hereIsMyBucket
```

2. Make sure you are at the /opt/dcp1_0/hw/green_bits/dma_afu/src directory, and run the command to upload the gbs file.

```
faascmd upload_object --object=dma_afu.gbs --file=dma_afu.gbs
```

3. Run the command to create an image.

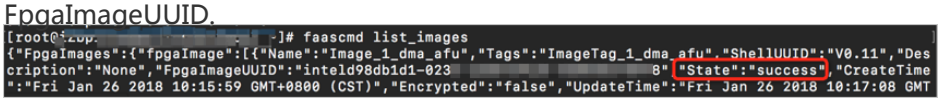
```
#  Replace hereIsYourImageName with your image name.
faascmd create_image --object=dma_afu.gbs --fpgatype=intel --name=hereIsYourImageName --tags=hereIsYourImageTag --encrypted=false --shell=V0.11
```

Step 5. Download the image

To download the image, follow these steps:

1. Run the faascmd list_images command to check whether the image is created.

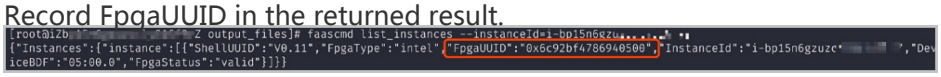
If "State":"success" exists in the returned result, it means the image is created. Record the FpgaImageUUID.



Run the command to obtain FPGA ID.

```
# Replace hereIsYourInstanceId with your f1 instance ID.
faascmd list_instances --instanceId=hereIsYourInstanceId
```

Record FpgaUUID in the returned result.



Run the command to download the image to your f1 instance.

```
# Replace hereIsYourInstanceID with your f1 instance ID. Replace hereIsFpgaUUID with your FpgaUUID.
Replace hereIsImageUUID with your FpgaImageUUID.
faascmd download_image --instanceId=hereIsYourInstanceID --fpgauuid=hereIsFpgaUUID --
fpgatype=intel --imageuuid=hereIsImageUUID --imagetype=afu --shell=V0.11
```

Run the command to check whether the image is downloaded.

```
# Replace hereIsYourInstanceID with your f1 instance ID. Replace hereIsFpgaUUID with your FpgaUUID.
faascmd fpga_status --instanceId=hereIsYourInstanceID --fpgauuid=hereIsFpgaUUID
```

If "TaskStatus":"operating" exists in the returned result, and the displayed FpgaImageUUID is identical with your recorded FpgaImageUUID, the image is downloaded.

Step 6. Test

Run the commands one by one for test.

```
cd /opt/dcp1_0/hw/green_bits/dma_afu/src/sw
make
./fpga_dma_test use_ase=0
```

If the following result is returned, the test is completed.

```
[root@iZ[redacted]Z sw]# ./fpga_dma_test use_ase=0
Running test in HW mode
Buffer Verification Success!
Buffer Verification Success!
Running DDR sweep test
Allocated test buffer
Fill test buffer
DDR Sweep Host to FPGA
Measured bandwidth = 5726.623061 Megabytes/sec
Clear buffer
DDR Sweep FPGA to Host
Measured bandwidth = 4473.924267 Megabytes/sec
Verifying buffer..
Buffer Verification Success!
```

Note:
If the Huge pages feature is not enabled, run the following command to enable it.

```
sudo bash -c "echo 20 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages"
```

Use OpenCL on an f2 instance

This document introduces how to use Open Computing Language (OpenCL) to create an image file, and then download image to an FPGA chip.

Note:
We strongly recommend that you use an f2 instance as a RAM user. To avoid unwanted operations, you must authorize the RAM user to run essential actions only. You must create a role for the RAM user and grant temporary permissions to the role to access the OSS buckets.

Prerequisites

Before you begin the procedure, you must:

- Create an f2 instance.
- Create a RAM user with granted permissions to play a RAM role with granted permissions, if you are using an f2 instance as a RAM user. You must create a pair of AccessKeyID and AccessKeySecret for the RAM user.
- Activate OSS service and create a bucket.
- Log on to the ECS console and obtain the instance ID of your f2 instance.

Procedure

To create an image by using OpenCL on an f2 instance and download it to an FPGA chip, follow these steps.

Step 1. Set up the environment

To set up the environment on an f2 instance, follow these steps.

1. Connect to an f2 instance.
2. Run the vim /root/xbinst_oem/setup.sh command to modify the file: comment out the unset XILINX_SDX line (Line 5), and save the file.

```
export XILINX_OPENCL=/root/2pf_acs_4ddr_normal_0906/xbinst_oem
export LD_LIBRARY_PATH=$XILINX_OPENCL/runtime/lib/x86_64:$LD_LIBRARY_PATH
export PATH=$XILINX_OPENCL/runtime/bin:$PATH
unset XILINX_SDACCEL
#unset XILINX_SDX
unset XCL_EMULATION_MODE
```

3. Run the command to install Screen to keep the terminal session persistent.

```
yum install screen -y
```

4. Run the command to open a new screen.

```
screen -S f2opencl
```

5. Run the command to set up a secure environment to download a file.

```
source /root/xbinst_oem/F2_env_setup.sh
```

Step 2. Compile a binary file

To compile a binary file, follow these steps.

1. Run the command to change the directory.

```
cd /opt/Xilinx/SDx/2017.2/examples/vadd
```

Run the cat sdaccel.mk | grep "XDEVICE" command to check the configuration of the XDEVICE parameter. Make sure it is set to xilinx:aliyun-ku115-f2:4ddr-xpr:4.2.

Run the vim ../common/common.mk command to modify the common.mk file: Replace Line 63

```
CLCC_OPT += $(CLCC_OPT_LEVEL) ${DEVICE_REPO_OPT} --platform ${XDEVICE} -o ${XCLBIN}
${KERNEL_DEFS} ${KERNEL_INCS}
```

with

```
CLCC_OPT += $(CLCC_OPT_LEVEL) ${DEVICE_REPO_OPT} --platform ${XDEVICE} -o ${XCLBIN}
${KERNEL_DEFS} ${KERNEL_INCS} --xp param:compiler.acceleratorBinaryContent=dcp
```

> **Note**:
> The parameters may be in lines between Line 60 and Line 62, which is determined by your file.

Run the command to compile the program.

```
export XILINX_SDX=/opt/Xilinx/SDx/2017.2
make -f sdaccel.mk xbin_hw
```

If the following information is displayed, it means the compilation of the binary file is in progress. The process may take several hours. Please be patient.

```
[root@iZ[redacted]'Z vadd]# make -f sdaccel.mk xbin_hw
make SDA_FLOW=hw xbin -f sdaccel.mk
make[1]: Entering directory `/opt/Xilinx/SDx/2017.2/examples/vadd'
xocc -t hw   --platform xilinx:kcu1500:4ddr-xpr:4.0 -o bin_vadd_hw.xclbin   --xp param
:compiler.acceleratorBinaryContent=dcp -s --kernel krnl_vadd krnl_vadd.cl

****** xocc v2017.2_sdx (64-bit)
  **** SW Build 1972098 on Wed Aug 23 11:34:38 MDT 2017
    ** Copyright 1986-2017 Xilinx, Inc. All Rights Reserved.

INFO: [XOCC 60-585] Compiling for hardware target
INFO: [XOCC 60-895]    Target platform: /opt/Xilinx/SDx/2017.2/platforms/xilinx_kcu150
0_4ddr-xpr_4_0/xilinx_kcu1500_4ddr-xpr_4_0.xpfm
```

Step 3. Check the packaging script

Run the command to check whether the packaging script exists or not.

```
file /root/xbinst_oem/sdaccel_package.sh
```

> **Note**:

A returned message containing cannot open (No such file or directory) means no packaging script exists. Then, download the script by running the following command.

```
wget http://fpga-tools.oss-cn-shanghai.aliyuncs.com/sdaccel_package.sh
```

Step 4. Create an image

To create an image, follow these steps.

1. Run the command to set up the OSS environment.

```
# Replace hereIsMySecretId, hereIsMySecretKey, and hereIsMyBucket with your own AccessKeyID,
AccessKeySecret, and OSS Bucket name
faascmd config --id=hereIsMySecretId --key=hereIsMySecretKey
faascmd auth --bucket=hereIsMyBucket
```

Run the ls command to obtain the file name.

```
[root@iZbp18o21m55wsf2k0obb7Z vadd]# ls
bin_vadd_hw.xclbin          krnl_vadd.cl  vadd.cpp
description.json            README.md     vadd.h
Export_Compliance_Notice.md sdaccel.mk    _xocc_krnl_vadd_bin_vadd_hw.dir
```

Run the command to package the binary file.

```
/root/xbinst_oem/sdaccel_package.sh -
xclbin=/opt/Xilinx/SDx/2017.2/examples/vadd/bin_vadd_hw.xclbin
```

After the packaging is completed, you have a packaged file in the same directory, such as 17_10_28-021904_SDAccel_Kernel.tar.gz in this example.

```
[root@iZbp18o21m55wsf2k0obb7Z vadd]# ls
17_10_28-021904-primary.bit          krnl_vadd.cl
17_10_28-021904_SDAccel_Kernel.tar.gz README.md
17_10_28-021904-xclbin.xml           sdaccel.mk
bin_vadd_hw.xclbin                   to_aliyun
description.json                     vadd.cpp
Export_Compliance_Notice.md          vadd.h
header.bin                           _xocc_krnl_vadd_bin_vadd_hw.dir
```

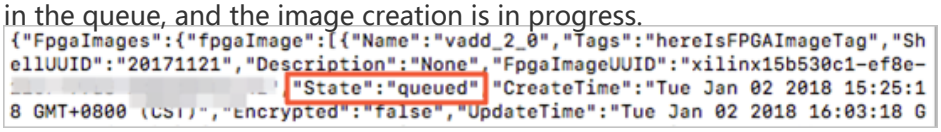
Run the command to upload the packaged file to your own OSS bucket.

```
# Replace the file name with the name of the packaged file
faascmd upload_object --object=bit.tar.gz --file=bit.tar.gz
```

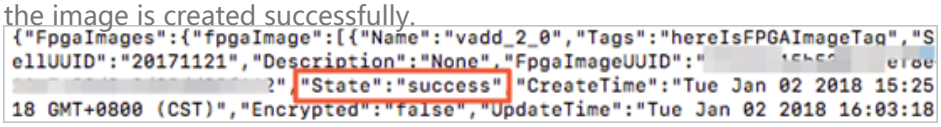
Run the command to create an image.

```
# Replace bit.tar.gz, hereIsFPGAImageName, and hereIsFPGAImageTag with the name of the packaged file, the image name, and the image tag
faascmd create_image --object=bit.tar.gz --fpgatype=xilinx --name=hereIsFPGAImageName --tags=hereIsFPGAImageTag --encrypted=false --shell=20171121
```

The example returned result is displayed as follows. The "State":"queued" means the task is in the queue, and the image creation is in progress.



> **Note:**
> Creation of an image consumes time. You can run the faascmd list_images command to check the image status. In the returned result, if "State":"success" displays, it means the image is created successfully.



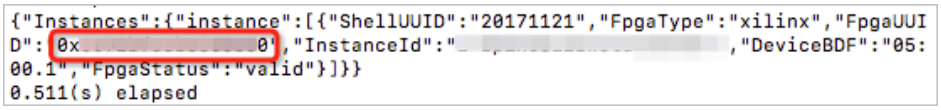
Record the FPGAImageUUID.

Step 5. Download the image

To download the image to an FPGA chip, follow these steps.

Run the command to obtain the FpgaUUID.

```
# Replace hereIsYourInstanceId with your f2 instance ID
faascmd list_instances --instanceId=hereIsYourInstanceId
```

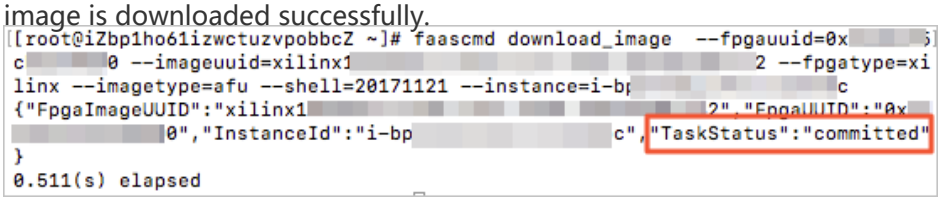


> **Note:**
> Record the FpgaUUID.

Run the command to download the image.

```
# Replace hereIsYourInstanceID with your f2 instance ID. Replace hereIsFpgaUUID with the recorded FpgaUUID. Replace hereIsImageUUID with the recorded FpgaImageUUID
faascmd download_image --instanceId=hereIsYourInstanceId --fpgauuid=hereIsFpgaUUID --fpgatype=xilinx --imageuuid=hereIsImageUUID --imagetype=afu --shell=20171121
```

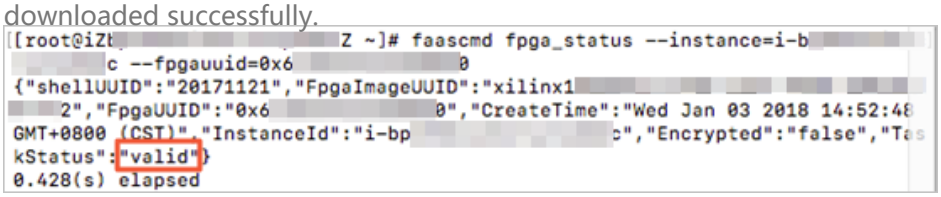
The example returned result is displayed as follows. The "State":"committed" means the image is downloaded successfully.



Note:
You can run the command to check whether the image is downloaded successfully or not.

```
# Replace hereIsYourInstanceID with your f2 instance ID. Replace hereIsFpgaUUID with your recorded FpgaUUID
faascmd fpga_status --instanceId=hereIsYourInstanceID --fpgauuid=hereIsFpgaUUID
```

The example returned result is displayed as follows. If "TaskStatus":"valid" exists and the displayed FpgaImageUUID is your image FpgaImageUUID, the image is downloaded successfully.



Step 6. Run the Host program

Run the commands to run the Host program.

```
unset XILINX_SDX
make -f sdaccel.mk host
./vadd bin_vadd_hw.xclbin
```

If Test Passed is returned, the test is successful.

Other common commands

In this section, some common commands for an FPGA instance are introduced.

Task	Command
View the help document	make -f ./sdaccel.mk help
Run software simulation	make -f ./sdaccel.mk run_cpu_em
Run hardware simulation	ake -f ./sdaccel.mk run_hw_em
Compile the host code only	make -f ./sdaccel.mk host

Compile and generate files for downloading	make -f sdaccel.mk xbin_hw
Clear a job directory	make -f sdaccel.mk clean
Force clear a job directory	make -f sdaccel.mk cleanall

Note:
During simulation of sdx2017.2, the device must be xilinx_aliyun-ku115-f2_4ddr-xpr_4_2.

Cloud Migration Tool

What is Alibaba Cloud Migration Tool

Alibaba Cloud Migration Tool, or **Cloud Migration Tool** for short, is a proprietary resource migration tool of Alibaba Cloud. It supports resource migration, such as the operating system, applications, and application data in a computer disk, of physical machines, VMs (virtual machines), or cloud hosts to the image list in the ECS console.

Being lightweight and agile, Alibaba Cloud Migration Tool helps you balance the workload between your local and cloud hosts, or cloud hosts from different cloud platforms.

Scenarios

Alibaba Cloud Migration Tool is applicable in the following scenarios:

Migrate physical servers to Alibaba Cloud ECS console.

Migrate virtual machines to Alibaba Cloud ECS console.

Migrate the cloud hosts from other cloud platforms, such as Amazon Web Services (AWS), Microsoft Azure, Tencent Cloud, or Huawei Cloud, to Alibaba Cloud ECS console.

Applicable operating systems

Alibaba Cloud Migration Tool supports the following 64-bit Windows and Linux operating systems:

- Windows server
  - Windows Server 2003
  - Windows Server 2008
  - Windows Server 2012
  - Windows Server 2016
- Linux server
  - CentOS 5/6/7
  - Ubuntu 12/14/16
  - Debian 7/8/9
  - Red Hat 5/6/7
  - SUSE 11.4/12.1/12.2
  - OpenSUSE 13.1
  - Gentoo 13.0

Before migrating an operating system that is not listed previously, you must exercise caution. In addition, see topic Migrate to Alibaba Cloud by using Cloud Migration Tool thoroughly.

Billing details

Alibaba Cloud Migration Tool is free of charge.

However, during the migration, an ECS instance is created by default under your Alibaba Cloud account to act as an intermediate station. Billing method of the intermediate ECS instance is Pay-As-You-Go. In case you have put a limit on your credit card, you must delimit your credit card before the payment is attempted.

Limits

Before migration, you must make sure that:

The on-premises server can access the Internet to uninterruptedly transfer data to the Alibaba Cloud ECS console.

The on-premises server is synchronized with the real time. Otherwise, an error indicating abnormal TimeStamp is recorded in the migration log file.

For more information, see Migrate to Alibaba Cloud by using Cloud Migration Tool.

Use Alibaba Cloud Migration Tool

For instructions on how to use Alibaba Cloud Migration Tool, see Migrate to Alibaba Cloud by using Cloud Migration Tool.

References

Currently, the following methods are available for migrating servers to Alibaba Cloud:

- Using the Alibaba Cloud Migration Tool
- Importing images in the ECS console
- Create and import on-premise images by using Packer

Change log

The following table shows the updated information about Alibaba Cloud Migration Tool:

Time	Release	Description
February 08, 2018	1.2.1	- Simplifies the user interaction during the migration process. - Capable of disabling the SELinux feature of a Linux on-premises server temporarily.
January 18, 2018	1.2.0	- Extends the range of data resource to be migrated, and more types of resource can be migrated. - Enhances the efficiency and stability of image creation.
January 01, 2018	1.1.8	- Supports SUSE 12 SP2 server. - Boosts the connection speed. - Optimizes the layout of the migration logs.

		- Fixes the possible network issue of the NetworkManager.
December 21, 2017	1.1.7	- Supports SUSE 12 SP1 server. - Capable of specifying the maximum bandwidth of data transmission.
December 14, 2017	1.1.6	- Scans the latest release of Alibaba Cloud Migration Tool. - Fixes the 6144 error of data transmission. - Proofreads the request parameter specified in the user_config.json configuration file.
December 08, 2017	1.1.5	- Fixes the issue of Linux data disk directory. - Highlights the error message in the migration logs.
December 01, 2017	1.1.3	Supports Debian server.

Migrate to Alibaba Cloud by using Cloud Migration Tool

Attention

To use Alibaba Cloud Migration Tool, consider the following:

Alibaba Cloud Migration Tool does not support migration of incremental data yet. You may select an off-peak period to suspend your on-premises server in which services that require data integrity are running.

During migration, an ECS instance named INSTANCE_FOR_GOTOALIYUN is created by default under your Alibaba Cloud account. It acts as an intermediate station.

Note:

- To avoid migration failure, do not stop, restart, or release the intermediate ECS instance. Moreover, the intermediate ECS instance is automatically released once the migration completes.
- If the migration fails, the intermediate ECS instance is retained in the ECS console for the next migration attempt. You can log on to the ECS console and manually release the instance to avoid unnecessary charges.

Billing method of the intermediate ECS instance is Pay-As-You-Go, you must make sure that no credit limit is set to your credit card and it allows the payment to go through.

After each successful migration, information about the intermediate ECS instance in the ECS console is auto recorded in the client_data configuration file. For the next migration, you must use the raw client configuration file that you have initially downloaded.

Note:
To avoid migration failure, in most cases, no manual modification for file client_data is required.

Migrating servers by using Cloud Migration Tool requires your AccessKeyID and AccessKeySecret, which are important credentials, and you must keep them confidential and secured.

Note:
If the AccessKey that you create belongs to a RAM user, you must make sure that the specified RAM user is authorized to operate the ECS resources. For more information, see *RAM* document Authorization policies.

If shared memory is used in your on-premises server:

Default action:

- For Windows servers: By default, Cloud Migration Tool recognizes and uploads the data on a shared memory that is attached to the C drive as one part of the system disk.
- For Linux servers: By default, Cloud Migration Tool recognizes and uploads the data on a shared memory as one part of the system disk.

Custom action:

- You can set the mount point directory of the shared memory as a data disk, and migrate it as an independent data disk.
- Alternatively, you can filter out the directory of the shared memory from migration and the data on the shared memory will not be migrated.

Prerequisites

Before you use the Alibaba Cloud Migration Tool, consider the following:

The on-premises server can access the Internet for uninterruptedly transferring data to Alibaba Cloud ECS console.

The system time of the on-premises server is synchronized with the real time. Otherwise, an error indicating abnormal TimeStamp is recorded in the migration log file.

To enable all the server configuration successfully after the migration, we recommend that you install cloud-init for your on-premises servers.

For on-premises servers running Windows OS

The go2aliyun_client.exe and Rsync\bin\rsync.exe programs are not restricted by firewall on the server.

The system start loader is normal.

Run Alibaba Cloud Migration Tool as an administrator.

For on-premises servers running Linux OS

The go2aliyun_client program is not restricted by firewall on the server.

The Rsync library has been installed.

- CentOS: Run yum install rsync –y.
- Ubuntu: Run apt-get install rsync –y.
- Debian: Run apt-get install rsync –y.
- Other distributions: See the installation documents of the distributions on their official website.

The Xen or Kernel-based Virtual Machine (KVM) driver is installed. For more information about how to install a KVM driver, see Install virtio driver.

SELinux must has been deactivated. You can temporarily deactivate SELinux by running setenforce 0.

Run Alibaba Cloud Migration Tool as a root user.

If the kernel of your on-premises Linux servers is too old and the version of GRUB (GRand Unified Bootloader) is earlier than 1.9. You may update the boot loader GRUB to a version later than 1.9.

Step 1. Download Alibaba Cloud Migration Tool

Log on to the ECS console to apply for migration.

After the approval, download and decompress the Alibaba Cloud Migration Tool package accordingly. The list of files is as follows:

Windows server

File or file folder	Description
Rsync file folder	This folder contains all the applications required for a migration. Do not modify the files manually except for the filter file Rsync\etc\rsync_excludes_win.txt.
client_data	Records of the transmission data during a migration.
user_config.json	Configuration file of the on-premises server
go2aliyun_client.exe	Main program of Cloud Migration Tool

Linux server

File or file folder	Description
client_check	Auxiliary program
client_data	Records of the transmission data during a migration.
user_config.json	The configuration file of the on-premises server
rsync_excludes_linux.txt	This file filters out the directories from the migration.
go2aliyun_client	Main program of Cloud Migration Tool

Step 2. Use Alibaba Cloud Migration Tool

Log on to the server, virtual machine, or cloud host to be migrated.

Decompress the Cloud Migration Tool package to a specified directory.

In the console, create an AccessKey, which is used in file user_config.json.

Configure file user_config.json as needed.

Filter out directories from migration as needed.

Run Cloud Migration Tool:

- **Windows** server: Right-click go2aliyun_client.exe and select **Run as administrator**.
- **Linux** server:
  a. Run chmod +x go2aliyun_client to make go2aliyun_client executable.
  b. Run ./ go2aliyun_client to migrate.

Wait for the results.

- If Goto Aliyun Finished! is displayed, go to the image details page of the ECS console to check the results.
- If Goto Aliyun Not Finished! is displayed, check the log files in the logs folder for troubleshooting. After the problem is rectified, run go2aliyun_client again, and it continues to proceed from where it was suspended during the preceding execution.

Customize user_config.json

The user_config.json configuration file is edited in JSON. The file contains necessary configuration information when you migrate the on-premises server to be migrated, including your AccessKey and target custom image. You must manually configure a few parameters. Make sure that the configuration complies with the JSON syntax. For more information about JSON syntax, see RFC 7159 .

Template of user_config.json

The template for user_config.json is as follows:

```
{
"access_id": "",
"secret_key": "",
"region_id": "",
"image_name": "",
"system_disk_size": 40,
"platform": "",
"architecture": "",
"data_disks": [],
"bandwidth_limit":0
}
```

Parameters in template

Table 1. Parameters for server configuration

Parameter name	Type	Required	Description
access_id	String	Yes	Your AccessKeyID for accessing Alibaba Cloud API. For more information, see Create AccessKey.
secret_key	String	Yes	Your AccessKeySecret for accessing Alibaba Cloud API. For more information, see Create AccessKey.
region_id	String	Yes	The ID of an Alibaba Cloud region to which your server is migrated, for example, cn-hangzhou (China East 1). For more information, see Regions and zones.
image_name	String	Yes	Set a name for your

			server image, which must be different from the name of existing images in the same region of Alibaba Cloud. - The image name is a string of [2, 128] English letters. - It must start with an uppercase or lowercase English letter. It can contain numbers, periods (.), underscores (_), and hyphens (-). - The image name is displayed in the ECS console. - It cannot start with http:// or https://.
system_disk_size	int	Yes	Specify the system disk size in the unit of GB. Value range: - [40, 500] - The value must be greater than the actually used space of the system disk on the on-premises server. For example, if

			the original system disk size is 500 GB and the used space is 100 GB, set this value greater than 100 GB.
platform	String	No	Operating system of the on-premises server. Optional values: - CentOS - Ubuntu - SUSE - OpenSUSE - Debian - RedHat - Others Linux - Windows Server 2003 - Windows Server 2008 - Windows Server 2012 - Windows Server 2016 The platform parameter is case sensitive.
architecture	String	No	Computer architecture. Optional values: - i386: 32-bit system architecture - x86_64: 64-bit system architecture
bandwidth_limit	int	No	The maximum bandwidth of data transmission, in the

			units of measurement KB/s. The default value is 0, and 0 indicates no limit for the bandwidth.
data_disks	Array	No	List of data disks in your on-premises server. A maximum of 16 data disks are supported. For more information about specific parameters, see Parameters for data disk configuration.

Table 2. Parameters for data disk configuration

Parameter name	Type	Manually set	Description
data_disk_index	int	Yes	Data disk number. Value range: [1, 16] Default value: 1
data_disk_size	int	Yes	Data disk size. Unit: GB. Value range: - [5, 2000] - The value must be greater than the actually used space of the data disk on the on-premises server. For example, if the original data disk size is 500 GB and the used space is 100 GB, set this value to be greater than 100 GB.
src_path	String	Yes	The directory of a data

			disk. Optional values: - In Windows, specify a drive letter, for example, D:, E:, or F:. - In Linux, specify a path, for example, /mnt/disk1, mnt/disk2, or /mnt/disk3. After you migrate an on-premises Linux server, the data disks are not mounted by default. You can run the command ls /dev/vd* to view the data disk devices. You may mount the data disks manually as needed, and edit configuration file /etc/fstab to configure the mounting file systems. For more information, see Linux_Format and mount a data disk.

Examples of custom configuration

The following describes how to customize user_config.json based on the configuration file templates in four scenarios:

Scenario 1. Migrate a Windows server without data disk

- Assume that the configuration of your server is as follows:
  - Operating system: Windows Server 2008
  - Used space of system disk: 30 GB
  - Computer architecture: 64-bit
- Migration destination:
  - Target migration region: Alibaba Cloud China East 1 region (cn-hangzhou)
  - Image name: CLIENT_IMAGE_WIN08_01
  - Size of system disk: 50 GB

You can configure the file user_config.json based on the following information:

```
{
"access_id": "YourAccessKeyID",
"secret_key": "YourAccessKeySecret",
"region_id": "cn-hangzhou",
"image_name": "CLIENT_IMAGE_WIN08_01",
"system_disk_size": 50,
"platform": "Windows Server 2008",
"architecture": "x86_64",
"data_disks": [],
"bandwidth_limit":0
}
```

Scenario 2. Migrate a Windows server with data disks

Assume that the three data disks are attached to the Windows server in Scenario 1. The drive letter and sizes of the data disks are as follows:

- D: 100 GB
- E: 150 GB
- F: 200 GB

You can configure the user_config.json file based on the following information:

```
{
"access_id": "YourAccessKeyID",
"secret_key": "YourAccessKeySecret",
"region_id": "cn-hangzhou",
"image_name": "CLIENT_IMAGE_WIN08_01",
"system_disk_size": 50,
"platform": "Windows Server 2008",
"architecture": "x86_64",
"data_disks": [ {
"data_disk_index": 1,
"data_disk_size": 100,
"src_path": "D:"
}, {
"data_disk_index": 2,
"data_disk_size": 150,
"src_path": "E:"
}, {
"data_disk_index": 3,
"data_disk_size": 200,
"src_path": "F:"
}
],
"bandwidth_limit":0
}
```

Scenario 3. Migrate a Linux server without data disk

- Assume that the configuration of your server is as follows:
  - Version: CentOS 7.2
  - Used space of system disk: 30 GB
  - Computer architecture: 64-bit
- Migration destination:
  - Target migration region: Alibaba Cloud China East 1 region (cn-hangzhou)
  - Image name: CLIENT_IMAGE_CENTOS72_01
  - Size of system disk: 50 GB

You can configure the user_config.json file based on the following information:

```
{
"access_id": "YourAccessKeyID",
"secret_key": "YourAccessKeySecret",
"region_id": "cn-hangzhou",
"image_name": "CLIENT_IMAGE_CENTOS72_01",
"system_disk_size": 50,
"platform": "CentOS",
"architecture": "x86_64",
"data_disks": [],
"bandwidth_limit":0
}
```

Scenario 4. Migrate a Linux server with data disks

Assume that the three data disks are attached to the Linux server in Scenario 3. The drive letter and sizes of the data disks are as follows:

- /mnt/disk1: 100 GB
- /mnt/disk2: 150 GB
- /mnt/disk3: 300 GB

You can configure the user_config.json file based on the following information:

```
{
"access_id": "YourAccessKeyID",
"secret_key": "YourAccessKeySecret",
"region_id": "cn-hangzhou",
"image_name": "CLIENT_IMAGE_CENTOS72_01",
"system_disk_size": 50,
"platform": "CentOS",
"architecture": "x86_64",
"data_disks": [ {
"data_disk_index": 1,
"data_disk_size": 100,
"src_path": "/mnt/disk1"
}, {
"data_disk_index": 2,
"data_disk_size": 150,
"src_path": "/mnt/disk2"
```

```
}, {
"data_disk_index": 3,
"data_disk_size": 200,
"src_path": "/mnt/disk3"
}
],
"bandwidth_limit":0
}
```

Filter out directories from migration

Alibaba Cloud Migration Tool can be used to filter out files or directories that are not migrated to Alibaba Cloud.

You can configure the rsync to filter out files and directories from migrating to Alibaba Cloud. On a Linux server, the filtering function is implemented by configuring the **rsync_excludes_linux.txt** text file. On a Windows server, the filtering function is implemented by configuring the **Rsync/etc/rsync_excludes_win.txt** text file. For how to add files and directories to be filtered out in the .txt file, see Documents Related to rsync.

Samples: For example, you do want to migrate the folder C:\MyDirs\Docs\Words and file C:\MyDirs\Docs\Excels\Report1.xlsx on a Windows server. You can write the configuration in the file **Rsync/etc/rsync_excludes_win.txt** as follows:

```
/MyDirs/Docs/Words/
/MyDirs/Docs/Excels/Report1.xlsx
```

Default files to be filtered out are as follows:

- Windows servers:
  - pagefile.sys
  - $RECYCLE.BIN
  - System Volume Information
- Linux servers:
  - /dev/*
  - /sys/*
  - /proc/*
  - /media/*
  - /lost+found/*
  - /mnt/*
  - /var/lib/lxcfs/*

  ...

  Note:

  - Directory /var/lib/lxcfs/* is only applicable to some versions. For example, when you do not have the access permission on the cache

directory for Linux containers of Ubuntu, /var/lib/lxcfs/* must be filtered out before migration.
- Cloud migration is a time-consuming task, and we recommend that you filter out the unnecessary directories on the data disks to be migrated, at the same time, the used storage space of cloud disks after the migration is reduced.

Troubleshooting

Log records of Alibaba Cloud Migration Tool are stored in the Logs directory under the main program directory. Abnormal interruptions during migration are recorded in log files. When Goto Aliyun Not Finished is displayed, you can check the log for troubleshooting.

Image test

After migration, the resource of your on-premises server, such as the operating system, applications, and application data, are displayed as a custom image in the image list of the ECS console. You can create a Pay-As-You-Go instance by using the custom image to test whether the custom image works or not.

Further operations

- You can create an instance by using the custom image.
- You can change the system disk by using the custom image.