

Solution

Quick Deployment of an MHA MySQL Cluster

Issue 1.0.0
Date 2023-04-30



Copyright © Huawei Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

Contents

1 Solution Overview.....	1
2 Resource Planning and Costs.....	3
3 Procedure.....	5
3.1 Preparations.....	5
3.2 Quick Deployment.....	12
3.3 Getting Started.....	19
3.4 Quick Uninstallation.....	29
4 Appendix.....	31
5 Change History.....	32

1 Solution Overview

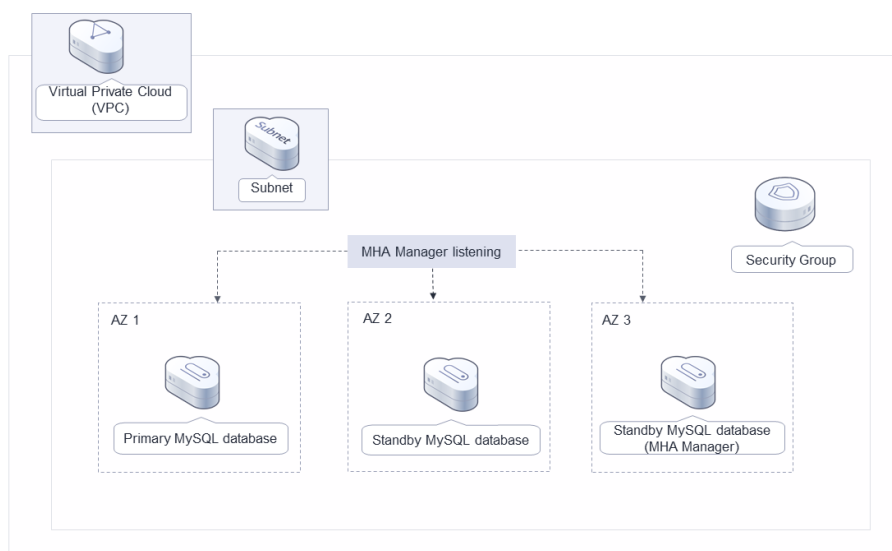
Scenarios

This solution is developed based on Master High Availability (MHA) and helps you deploy highly available MySQL clusters on Huawei Cloud cloud servers. **MHA** is a mature and open source MySQL HA program that provides heartbeat detection, primary/standby replication, failover, and alert email sending. It is suitable for enterprises that require high availability, data integrity, and almost uninterrupted maintenance of primary servers.

Architecture

This solution helps you quickly deploy an MHA MySQL cluster on Huawei Cloud. The following figure shows the deployment architecture.

Figure 1-1 Architecture



This solution will:

- Create three **FlexusX instances**, add them to the same FlexusX group, configure an anti-affinity policy, deploy them across AZs, and install the MHA and MySQL software on each FlexusX instance.
- Create a virtual IP address (VIP) for primary/standby switchover of the MySQL databases.
- Assign three **EIPs** for Internet access and alarm notifications in case of any faults.
- Create a security group to secure the FlexusX instance environment by controlling access to FlexusX instances.

Advantages

- **High reliability**
The cloud servers configured with the anti-affinity policy and deployed across AZs provide cross-AZ disaster recovery. MHA is installed for automated failover and primary/standby replication, helping ensure data consistency in the event of a failure.
- **Lower cost**
MHA is composed of MHA Manager and MHA Node. MHA Manager is installed on one of the cloud servers where the standby databases are deployed. No additional cloud server is required.
- **Easy deployment**
Resources can be quickly provisioned and a highly available MySQL cluster can be easily created.

Constraints

- Before deploying this solution, ensure that you have created a Huawei ID with access to the target region and enabled Huawei Cloud services.
- If you select the yearly/monthly billing mode, ensure that your account has sufficient balance. If you do not have sufficient balance, you can go to the **Billing Center** to manually pay for the order.
- Ensure that you have an email account for sending emails over SMTP.
- Currently, FlexusX instances can be deployed only in the **AP-Bangkok** region.

2 Resource Planning and Costs

This solution will deploy the resources listed in the following table. The costs are only estimates and may differ from the final prices. For details, see [Price Calculator](#).

Table 2-1 Resource planning and costs (Pay-per-use)

Huawei Cloud Service	Example Configuration	Estimated Monthly Cost
Flexus X Instance	<ul style="list-style-type: none">• Pay-per-use: \$0.08 USD/hour• Region: AP-Bangkok• Specifications: FlexusX Performance mode (disabled) x1.2u.4g 2 vCPUs 4 GiB• Image: CentOS 7.6 64bit• System Disk: High I/O 40 GiB• Data disk: General Purpose SSD 100 GiB• Quantity: 3	\$181.44 USD
Elastic IP (EIP)	<ul style="list-style-type: none">• Pay-per-use: \$0.11 USD/hour• Region: AP-Bangkok• Routing Type: Dynamic BGP• Billed By: Traffic• EIP Quantity: 3 (After two of the created EIPs are released, they will not be billed.)	\$0.11 USD/GB
Total	-	\$181.44 USD + Price of public network traffic

Table 2-2 Resource planning and costs (Yearly/Monthly)

Huawei Cloud Service	Example Configuration	Estimated Monthly Cost
Flexus X Instance	<ul style="list-style-type: none"> ● Region: AP-Bangkok ● Specifications: FlexusX Performance mode (disabled) x1.2u.4g 2 vCPUs 4 GiB ● Image: CentOS 7.6 64bit ● System Disk: High I/O 40 GiB ● Data Disk: General Purpose SSD 100 GiB ● Quantity: 3 	\$199.80 USD
Elastic IP (EIP)	<ul style="list-style-type: none"> ● Region: AP-Bangkok ● Routing Type: Dynamic BGP ● Billed By: Traffic ● Bandwidth: 5 Mbit/s ● EIP Quantity: 3 (After two of the created EIPs are released, they will not be billed.) 	\$0.11 USD/GB
Total	-	\$47.73 USD + Price of public network traffic

3 Procedure

- [3.1 Preparations](#)
- [3.2 Quick Deployment](#)
- [3.3 Getting Started](#)
- [3.4 Quick Uninstallation](#)

3.1 Preparations

Creating the rf_admin_trust Agency

- Step 1** Log in to [Huawei Cloud management console](#), move your mouse over the account name, and choose **Identity and Access Management**.

Figure 3-1 Console page

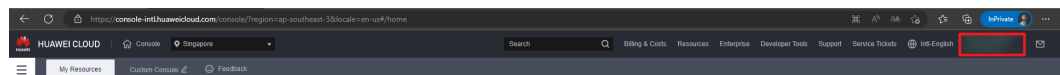
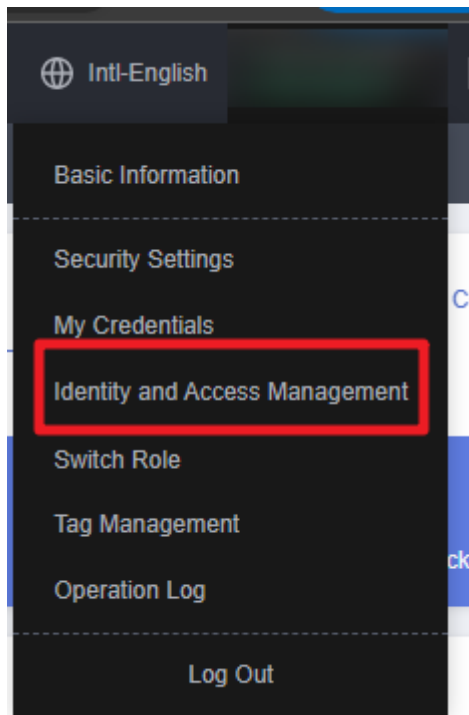
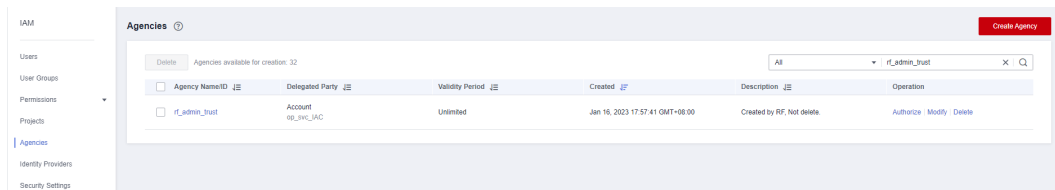


Figure 3-2 Identity and Access Management



Step 2 Choose **Agencies** in the left navigation pane and search for the **rf_admin_trust** agency.

Figure 3-3 Agency list



- If the agency is found, skip the following steps.
- If the agency is not found, perform the following steps to create it.

Step 3 Click **Create Agency** in the upper right corner of the page. On the displayed page, enter **rf_admin_trust** for **Agency Name**, select **Cloud service** for **Agency Type**, enter **RFS** for **Cloud Service**, and click **Next**.

Figure 3-4 Create Agency

Agencies / Create Agency

* Agency Name

* Agency Type Account
Delegate another HUAWEI CLOUD account to perform operations on your resources.
 Cloud service
Delegate a cloud service to access your resources in other cloud services.

* Cloud Service

* Validity Period

Description
0/255

Step 4 Search for **Tenant Administrator** and select it in the search results.

Figure 3-5 Selecting a policy

Authorize Agency

1 Select Policy/Role 2 Select Scope 3 Finish

Assign selected permissions to rf_admin_trust1. Create Policy

View Selected (1) Copy Permissions from Another Project

Policy/Role Name	Type
<input type="checkbox"/> DME AdministratorAccess Data Model Engine tenant administrator with full permissions.	System-defined policy
<input checked="" type="checkbox"/> Tenant Administrator Tenant Administrator (Exclude IAM)	System-defined role
<input type="checkbox"/> CS Tenant Admin Cloud Stream Service Tenant Administrator, can manage multiple CS users	System-defined role

Step 5 Select **All resources** and click **OK**.

Figure 3-6 Setting the authorization scope

Authorize Agency

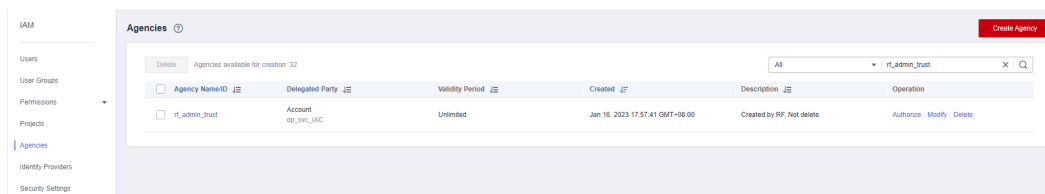
1 Select Policy/Role 2 Select Scope 3 Finish

The following are recommended scopes for the permissions you selected. Select the desired scope requiring minimum authorization.

Scope

All resources
IAM users will be able to use all resources, including those in enterprise projects, region-specific projects, and global services under your account based on assigned permissions.
[Show More](#)

Step 6 Check that the **rf_admin_trust** agency is created in the agency list.

Figure 3-7 Agency list

Agency Name/ID	Delegated Party	Validity Period	Created	Description	Operation
rt_admin_trust	Account: RP_ADMIN	Unlimited	Jan 16, 2023 17:57:41 GMT+08:00	Created by RP, Not delete	Authorize, Modify, Delete

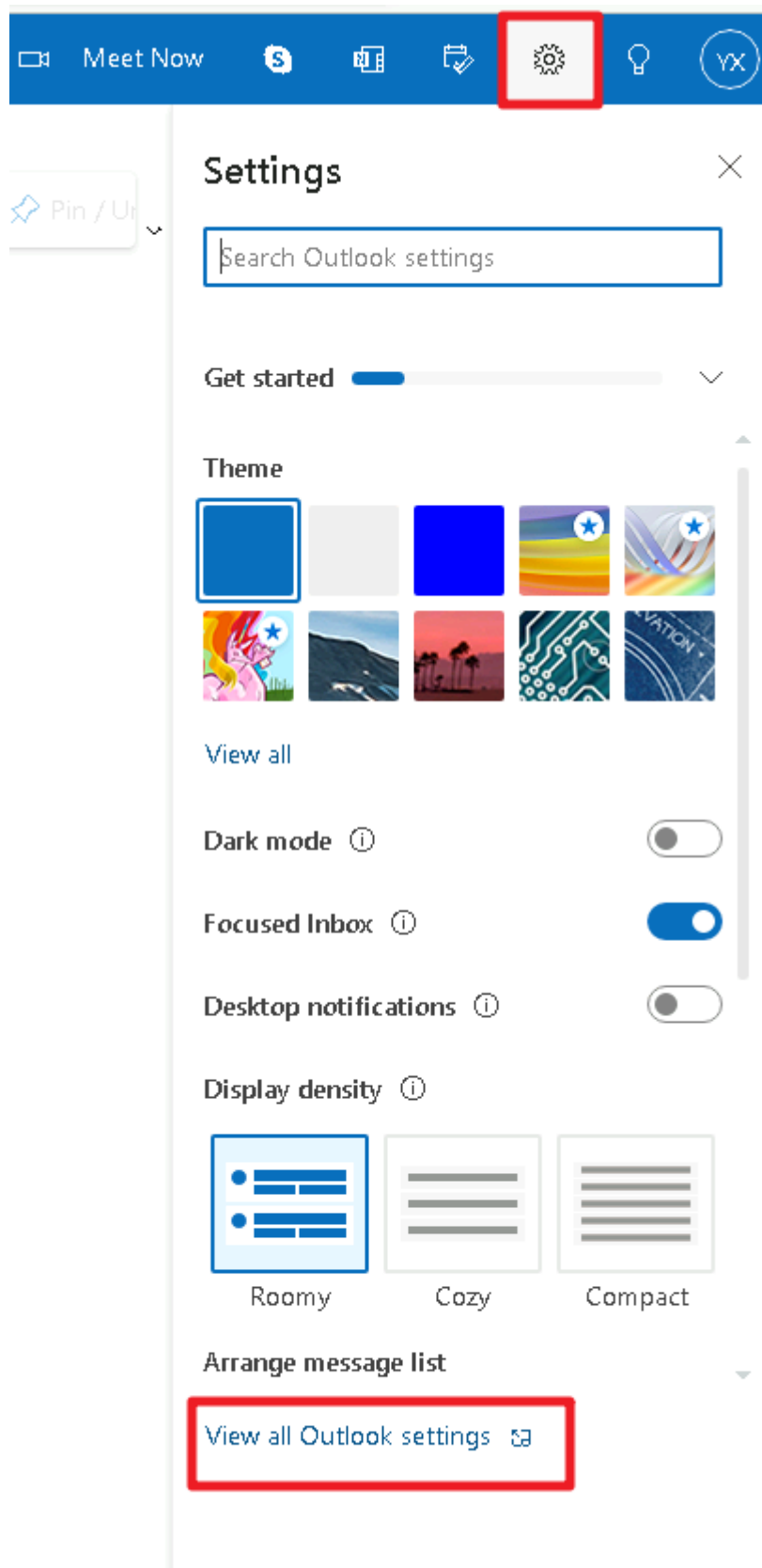
----End

Obtaining an Authorization Code

When MHA performs a failover or stops working due to an error, you can execute **send_report** to obtain the failover report by an email. In this way, users can learn about the current database status in time. In this solution, emails are sent through SMTP. You need to log in to the mailbox to enable SMTP. In addition, an independent password or authorization code is required for sending emails. Perform the following operations in this section to obtain the authorization code. The Outlook mailbox is used as an example.

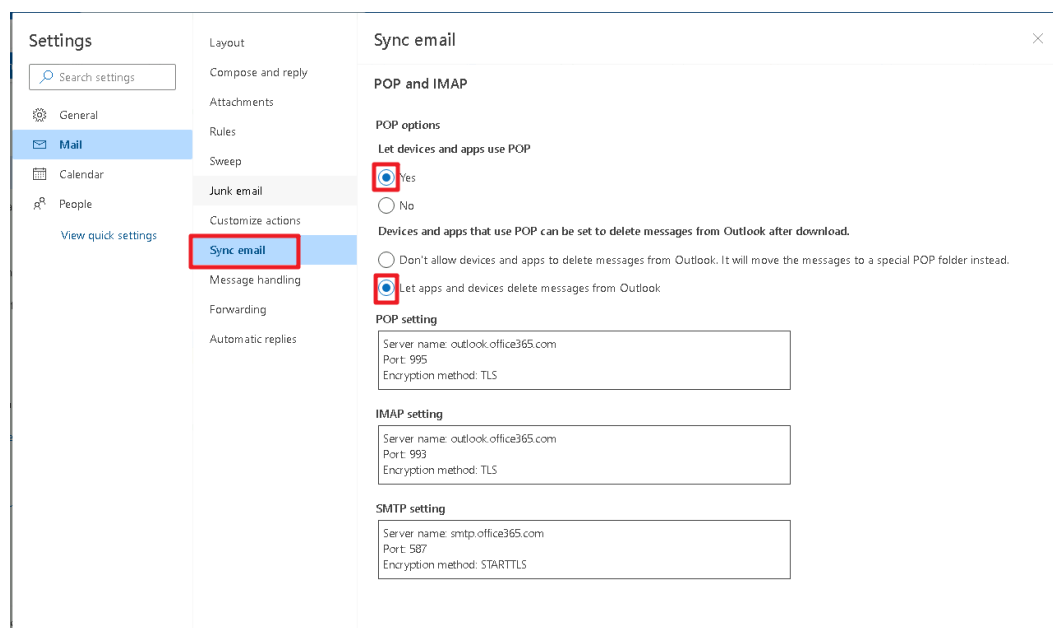
- Step 1** Log in to your mailbox. Click the **Settings** icon in the upper right corner and click **View all Outlook settings**.

Figure 3-8 Settings



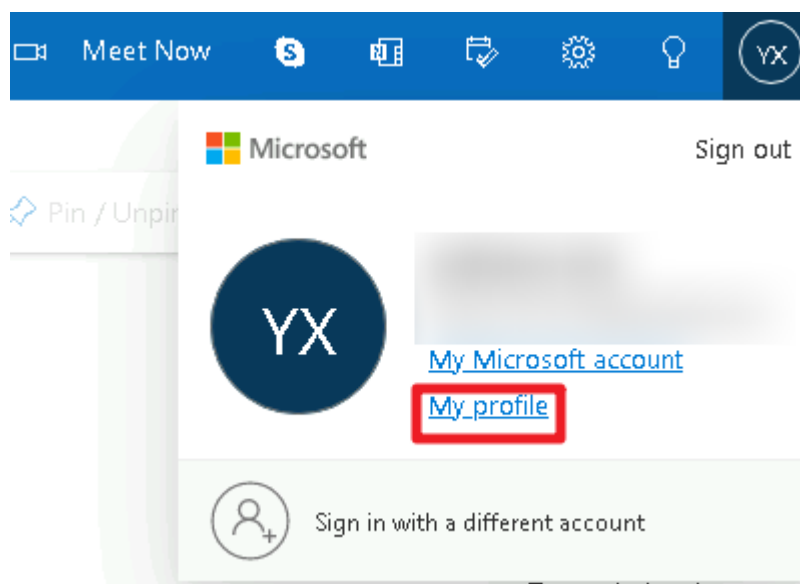
Step 2 Choose **Sync email** on the left, select **Yes** and **Let apps and devices to delete messages from Outlook** (if required). View and record the value of **Server name**.

Figure 3-9 Sync email



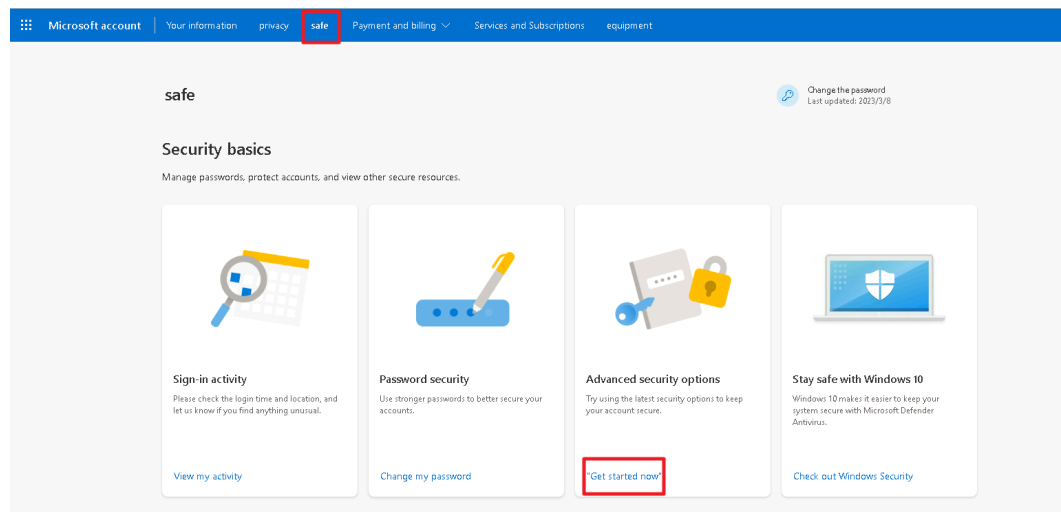
Step 3 Click the avatar in the upper right corner and click **My profile**.

Figure 3-10 My profile



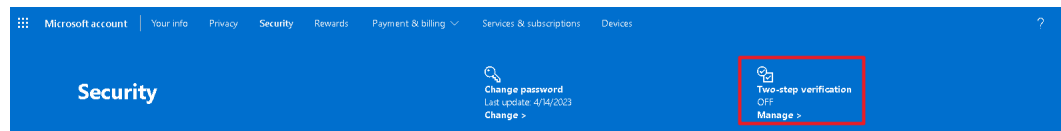
Step 4 Choose **safe** on the top menu bar. On the displayed page, click **Get started now**.

Figure 3-11 safe



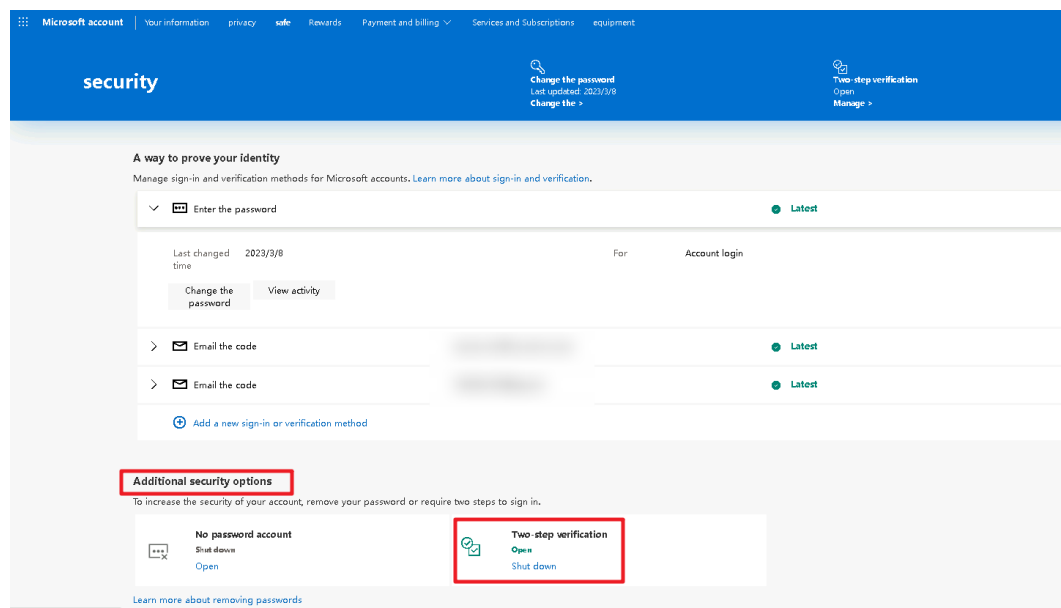
Step 5 After the system verification is successful, click **Manage** under **Two-step verification**, perform required operations to enable **Two-step verification**.

Figure 3-12 Two-step verification



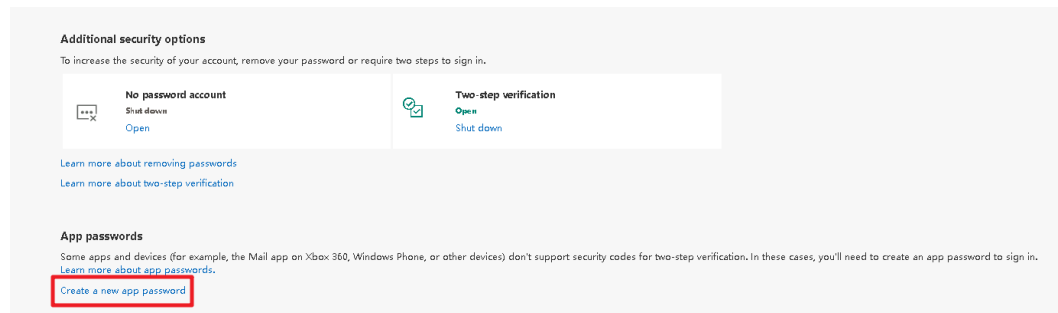
Step 6 In the **Additional security options** area, verify that the status of **Two-step verification** is **Open**.

Figure 3-13 Two-step verification



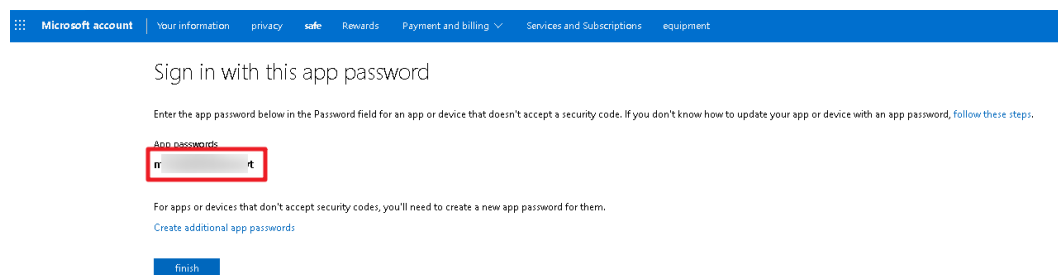
Step 7 In the **App passwords** area, click **Create a new app password** to set a password for connecting to a third-party application.

Figure 3-14 App passwords



Step 8 Obtain the authorization code and enter it in the text box of `email_authorization_code`.

Figure 3-15 Obtaining an authorization code



NOTE

By default, SMTP is disabled for Outlook. To send emails over SMTP, enable it in advance.

----End

3.2 Quick Deployment

This section describes how to quickly deploy this solution.

Table 3-1 Parameter description

Parameter	Type	Mandatory	Description	Default value
vpc_name	string	Yes	VPC name. This template uses a newly created VPC and the VPC name must be unique. The name contains 1 to 54 characters, including digits, letters, underscores (_), hyphens (-), and periods (.).	highly-available-mha-mysql-cluster-demo

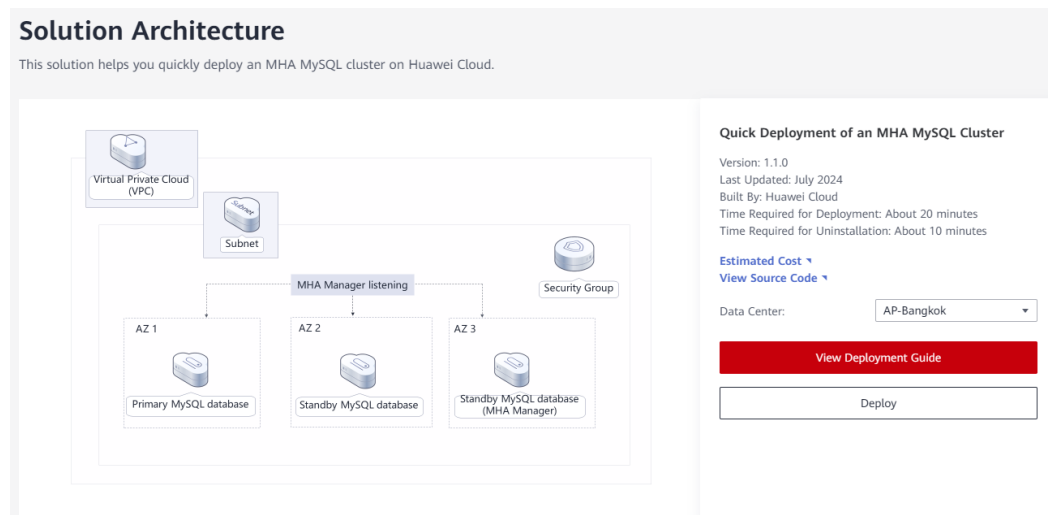
Parameter	Type	Mandatory	Description	Default value
security_group_name	string	Yes	Security group name. This template uses a newly created security group. The name contains 1 to 64 characters, including letters, digits, underscores (_), hyphens (-), and periods (.).	highly-available-mha-mysql-cluster-demo
ecs_name	string	Yes	Cloud server name, which must be unique. The name contains 1 to 57 characters, including lowercase letters, digits, underscores (_), and hyphens (-).	highly-available-mha-mysql-cluster-demo
ecs_flavor	string	Yes	Instance flavor of ECS or FlexusX. The flavor ID format of a FlexusX instance is x1.?u.?g. For example, the flavor ID of a FlexusX instance with 2 vCPUs and 4 GiB memory is x1.2u.4g. For details about FlexusX instance flavors, see the console. For details about ECS flavors, see A Summary List of x86 ECS Specifications .	x1.2u.4g

Parameter	Type	Mandatory	Description	Default value
ecs_password	string	Yes	Initial password of the cloud server, MySQL replication account, and MHA administrator account. The value consists of 8 to 26 characters. It includes at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (\$!@%-_+=[:./^,{}?), and cannot contain the username or the username spelled backwards. The default ECS administrator account is root , the MySQL replication account is repl , and the MHA administrator account is mha .	Left blank
system_disk_size	number	Yes	System disk size of the cloud server. The default disk type is high I/O and the ECS disk space cannot be scaled down. The value ranges from 40 to 1,024, in GiB.	40
data_disk_size	number	Yes	Data disk size of the cloud server. The default disk type is general-purpose SSD. Value range: 10 GiB to 32,768 GiB.	100
sender_email_address	string	Yes	Sender's email address, which sends alarm emails to inform users of a MySQL database failover through MHA. Example: mha@huawei.com	Left blank
recipient_email_address	string	Yes	Recipient's email address, which receives alarm emails. Example: recipient@huawei.com	Left blank

Parameter	Type	Mandatory	Description	Default value
email_authorization_code	string	Yes	Email account authorization code, which is used to authenticate operations for sending emails using the sender's email account through MHA. For details about how to obtain the authorization code, see 3.1 Preparations .	Left blank
smtp_server_address	string	Yes	SMTP server address of the sender's mailbox. For example, the SMTP address of Outlook is smtp.office365.com . SMTP may be disabled for some mailboxes by default and needs to be enabled before you send emails over SMTP.	Left blank
charging_mode	string	Yes	Billing mode. The value can be postPaid (pay-per-use) or prePaid (yearly/monthly). The default value is postPaid . By default, an order is automatically paid from the account balance.	postPaid
charge_period_unit	string	Yes	Subscription period unit. This parameter is valid only when charge_mode is set to prePaid . The value can be month or year .	month
charge_period	number	Yes	Subscription period unit. This parameter is valid only when charge_mode is set to prePaid . If charge_period_unit is set to month , the value ranges from 1 to 9 . If charge_period_unit is set to year , the value ranges from 1 to 3 .	1

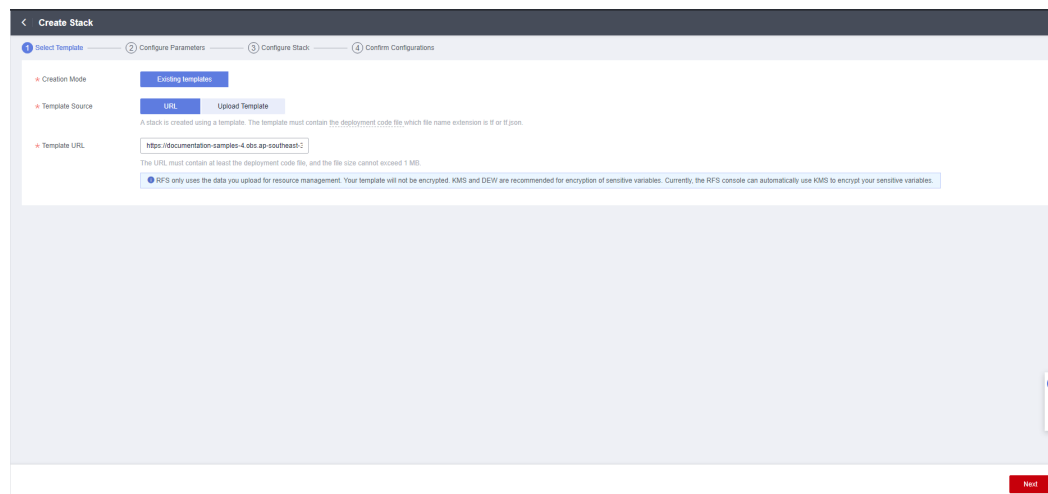
Step 1 Log in to Huawei Cloud Solution Practice and choose **Quick Deployment of an MHA MySQL Cluster**.

Figure 3-16 Selecting a solution



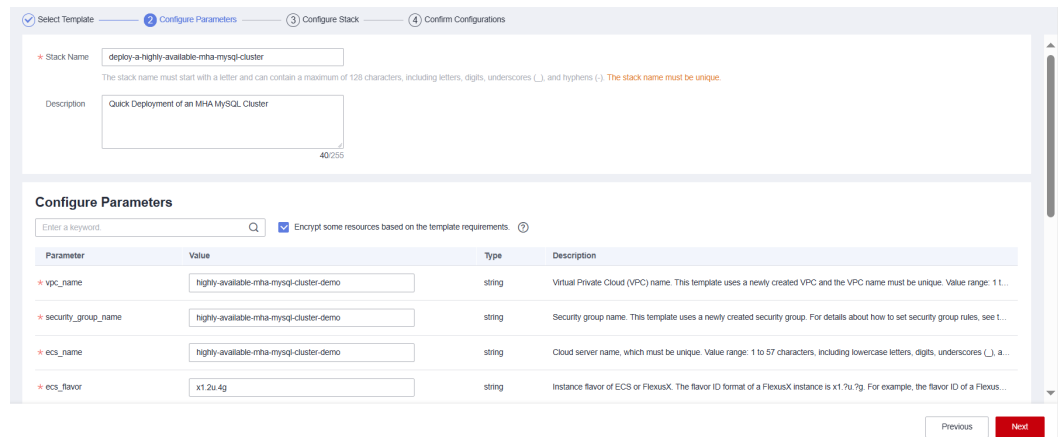
Step 2 Click **Deploy Now** to switch to the **Create Stack** page.

Figure 3-17 Create Stack



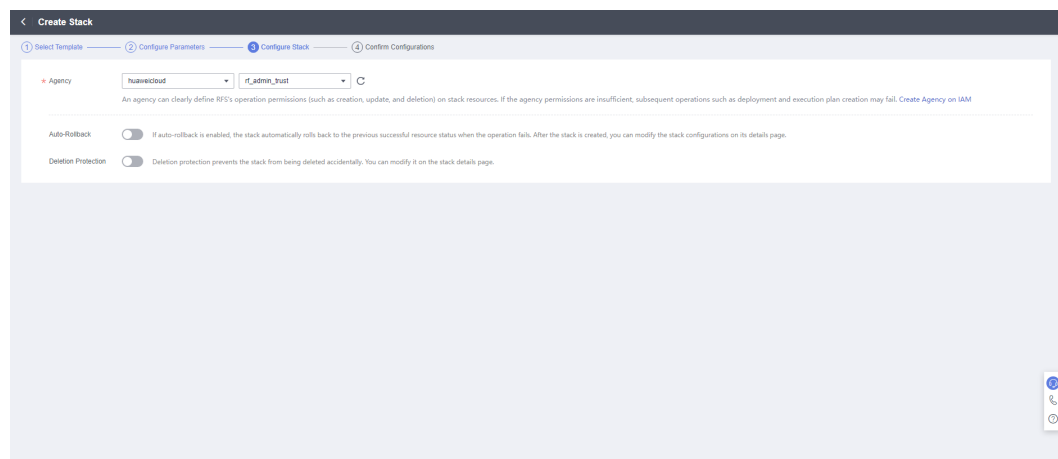
Step 3 Click **Next**, and set parameters by referring to **Table 3-1**.

Figure 3-18 Configuring parameters



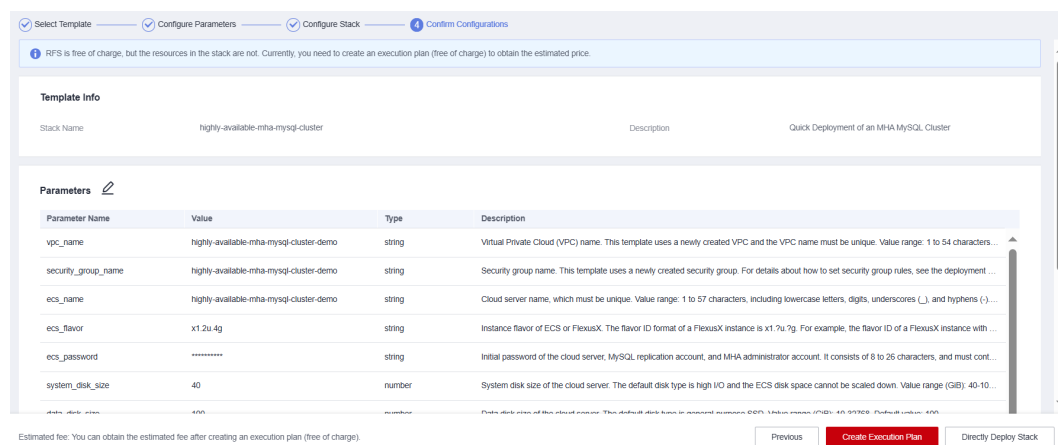
Step 4 On the **Configure Stack** page, select **rf_admin_trust** from the **Agency** drop-down list and click **Next**.

Figure 3-19 Configuring a stack



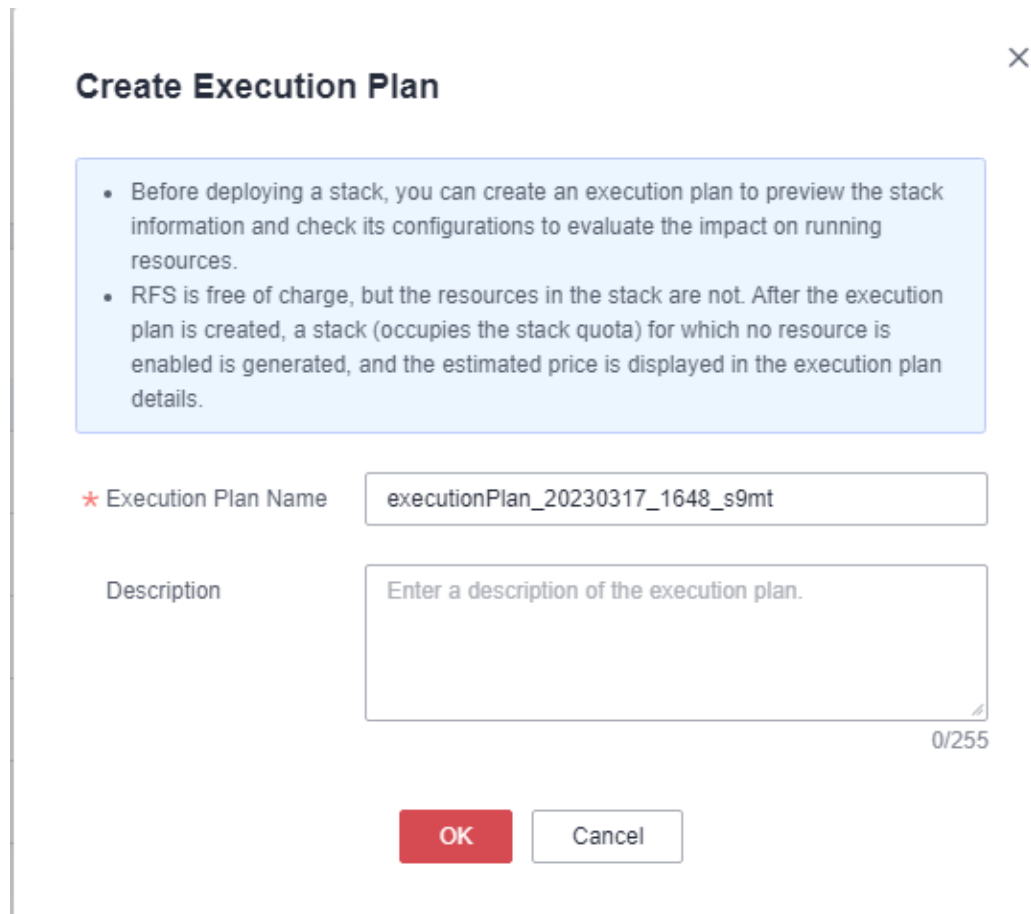
Step 5 On the **Confirm Configurations** page, click **Create Execution Plan**.

Figure 3-20 Confirming the configurations



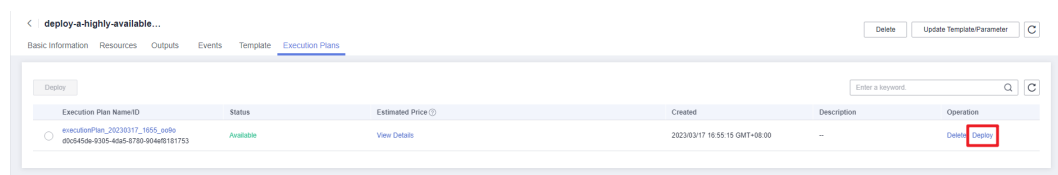
Step 6 In the displayed **Create Execution Plan** dialog box, specify **Execution Plan Name** and click **OK**.

Figure 3-21 Creating an execution plan



Step 7 Locate the target execution plan and click **Deploy** in the **Operation** column. In the displayed dialog box, click **Execute** for resource deployment.

Figure 3-22 Deploying an execution plan



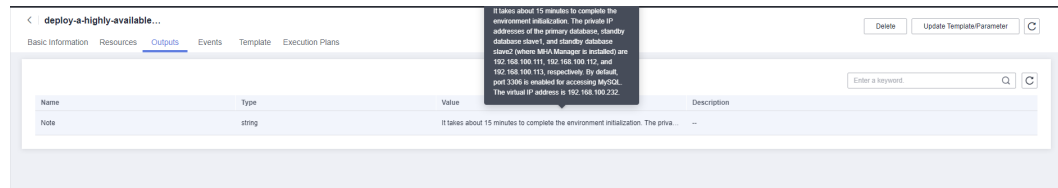
Step 8 Click the **Events** tab and check whether the solution has been deployed. If message "Apply required resource success" is displayed in the **Description** column, the solution has been deployed.

Figure 3-23 Resources created



Step 9 Click the **Outputs** tab to view information about the VIP and ECSs.

Figure 3-24 Output



----End

3.3 Getting Started

(Optional) Modifying Security Group Rules

NOTICE

- By default, IP addresses from the VPC subnet CIDR block created in this solution are allowed to access the MySQL database through port 3306. Configure an IP address whitelist by referring to [Modifying a Security Group Rule](#).
- This solution uses port 22 to remotely log in to the cloud server. By default, the VPC subnet created in this solution allows access from port 22. Configure an IP address whitelist by referring to [Modifying a Security Group Rule](#).

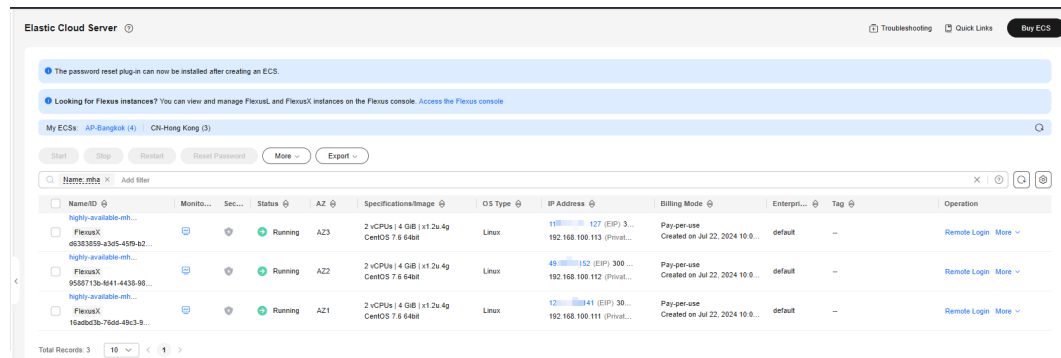
A security group is a collection of access control rules for cloud resources, such as cloud servers, containers, and databases, to control inbound and outbound traffic. Cloud resources associated with the same security group have the same security requirements and are mutually trusted within a VPC.

If the rules of the security group associated with your instance cannot meet your requirements, for example, you need to add, modify, or delete a TCP port, do as follows:

- Adding a security group rule: [Add an inbound rule](#) and enable a TCP port if needed.
- Modifying a security group rule: Inappropriate security group settings can be a serious security risk. You can [modify security group rules](#) to ensure the network security of your instances.
- Deleting a security group rule: If the source or destination IP address of an inbound or outbound security group rule changes, or a port does not need to be enabled, you can [delete the security group rule](#).

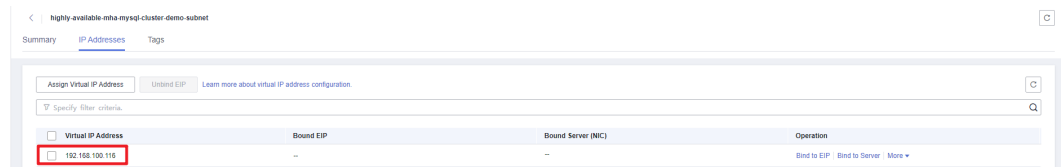
Step 1 Log in to the [ECS console](#) and view the instances created through one-click deployment and the EIPs bound to the instances.

Figure 3-25 Instances



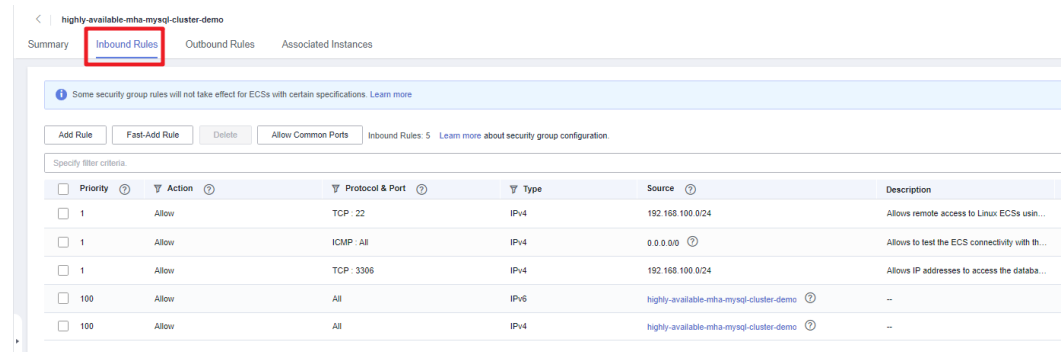
Step 2 Open the **Subnets** where the cloud server is located and click the **IP Addresses** tab to view the virtual IP address.

Figure 3-26 Virtual IP address



Step 3 Check the security group rules. On the **Security Groups** page, locate the security group prefixed with the VPC name, and click the security group to view the security group rules. By default, ports 22 and 3306 are enabled in the inbound rules. Perform operations to modify the security group rules by referring to **(Optional) Modifying Security Group Rules**.

Figure 3-27 Security group rules



Step 4 Log in to the three cloud servers respectively as the administrator **root**.

Figure 3-28 Logging in to cloud servers

```
CentOS Linux 7 (Core)
Kernel 3.10.0-1160.53.1.el7.x86_64 on an x86_64

highly-available-mha-mysql-cluster-demo-master login: [ 57.797087] vdb: vdb1
[ 57.799202] vdb: vdb1
[ 60.076133] EXT4-fs (vdb1): mounted filesystem with ordered data mode. Opts: (null)
root
Password:

Welcome to Huawei Cloud Service

[root@highly-available-mha-mysql-cluster-demo-master ~]# _
```

Step 5 Unbind the EIPs bound to the primary database and standby database suffixed with **master** and **slave1**. Log in to the **ECS console**, locate the target database server, choose **More > Manage Network > Unbind EIP** in the **Operation** column, and click **Yes** in the displayed dialog box. Do not release the EIP bound to the standby database (suffixed with **slave2**) with MHA Manager installed. Otherwise, an alarm email will fail to be sent during a failover.

Figure 3-29 Unbinding an EIP

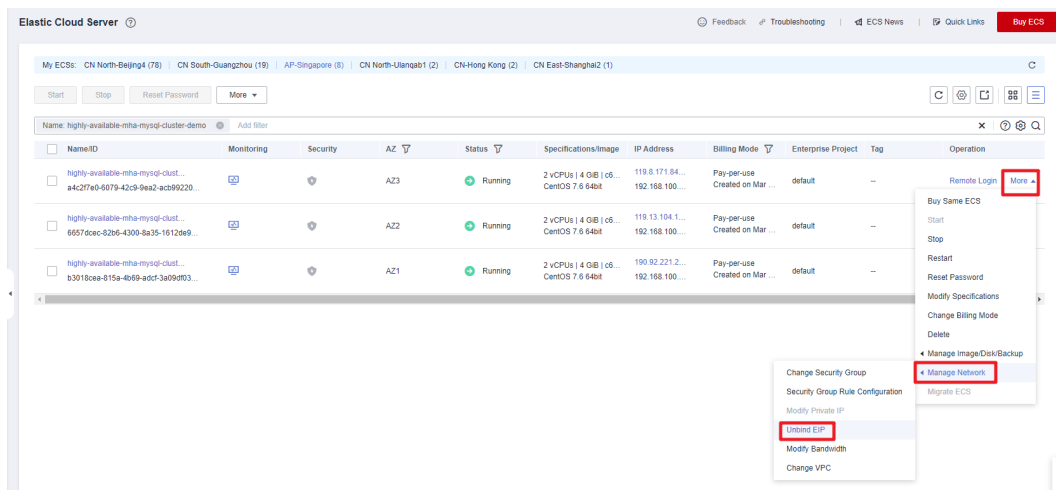


Figure 3-30 Confirming unbinding



Step 6 Release the EIPs. Locate the two EIPs whose **Status** is **Unbound**, choose **More > Release**, and click **Yes** in the displayed dialog box.

Figure 3-31 Releasing the EIPs

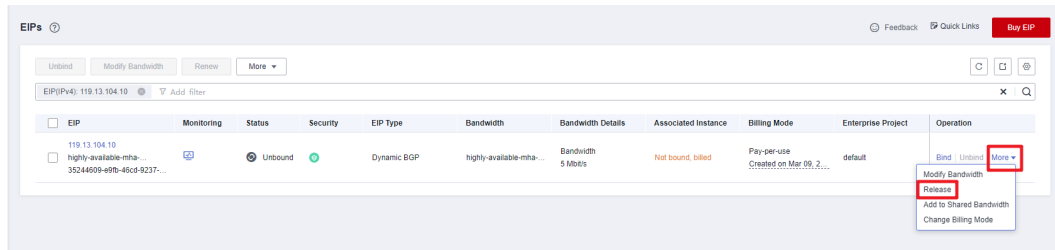


Figure 3-32 Confirming the release of the EIPs



----End

Initializing the Environment

Restarting ECSs may cause the MHA MySQL cluster to stop. You need to manually start MHA monitoring. Refer to the following operations.

- Step 1** Reconfigure the VIP. Log in to the ECS where the primary database is deployed and run the `ifconfig eth0:1 VIP/24` command. *VIP* is the virtual IP address obtained in [Step 2](#) or [Step 9](#).

Figure 3-33 Reconfiguring the VIP

```
[root@mha-mysql-master ~]# ifconfig eth0:1 192.168.100.95/24
[root@mha-mysql-master ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.100.111 netmask 255.255.255.0 broadcast 192.168.100.255
    inet6 fe80::f816:3eff:fec9:8590 prefixlen 64 scopeid 0x20<link>
    ether fa:16:3e:c9:85:90 txqueuelen 1000 (Ethernet)
    RX packets 431 bytes 72480 (70.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 347 bytes 57795 (56.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.100.95 netmask 255.255.255.0 broadcast 192.168.100.255
    ether fa:16:3e:c9:85:90 txqueuelen 1000 (Ethernet)

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@mha-mysql-master ~]#
```

- Step 2** Enable the MHA service. Log in to the ECS (suffixed with **slave2**) with MHA Manager installed and run the `mha_app1_start` command to start MHA monitoring. If the MHA status is **running**, the cluster service is started.

Figure 3-34 Enabling MHA

```
[root@mha-mysql-slave2 ~]# mha_app1_start
[1] 1553
[root@mha-mysql-slave2 ~]# mha_app1_status
app1 (pid:1553) is running(0:PING_OK), master:192.168.100.111
[root@mha-mysql-slave2 ~]#
```

----End

Working with MySQL Databases

In this solution, three cloud servers are created by default, each of which has a MySQL 5.7.34 database installed. The databases include one primary database (suffixed with **master**) and two standby databases (suffixed with **slave1** and **slave2**). By default, a user group **mysql** is created, the **mysql** user is added to the user group, and the service port 3306 is enabled. You need to create account **repl** on the primary database for primary/standby replication. Set the account password to be the same as the initial password. Allow only the IP addresses from 192.168.100.0/24 to access the primary database. Install MHA Manager on the standby database (suffixed with **slave2**). The default administrator account is **mha** and the password is the same as the initial password.

Check the primary/standby replication status.

```
SHOW SLAVE STATUS\G;
```

Figure 3-35 Replication statuses of the standby databases

```
> show slave status\G;
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: 192.168.100.111
      Master_User: repl
      Master_Port: 3306
      Connect_Retry: 10
      Master_Log_File: mysql_bin.000002
      Read_Master_Log_Pos: 740
      Relay_Log_File: mha-mysql-slave2-relay-bin.000002
      Relay_Log_Pos: 953
      Relay_Master_Log_File: mysql_bin.000002
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      Replicate_Do_DB:
      Replicate_Ignore_DB:
      Replicate_Do_Table:
      Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
      Replicate_Wild_Ignore_Table:
      Last_Errno: 0
      Last_Error:
      Skip_Counter: 0
      Exec_Master_Log_Pos: 740
      Relay_Log_Space: 1171
      Until_Condition: None
      Until_Log_File:
      Until_Log_Pos: 0
      Master_SSL_Allowed: No
      Master_SSL_CA_File:
      Master_SSL_CA_Path:
      Master_SSL_Cert:
      Master_SSL_Cipher:
      Master_SSL_Key:
      Seconds_Behind_Master: 0
      Master_SSL_Verify_Server_Cert: No
      Last_IO_Errno: 0
      Last_IO_Error:
      Last_SQL_Errno: 0
      Last_SQL_Error:
      Replicate_Ignore_Server_Ids:
      Master_Server_Id: 111
```

Create a replication account on the primary database. (By default, the **repl** user has been created, and the password is set to the initial password.)

```
mysql -uroot -S /tmp/mysql.sock -e "grant replication slave on *.* to Account@'%' identified by 'Password'";
Example:
mysql -uroot -S /tmp/mysql.sock -e "grant replication slave on *.* to repl@'192.168.100.%' identified by '123'";
```

Create an MHA administrator account on the primary database. (By default, the **mha** user has been created, and the password is set to the initial password.)

```
mysql -uroot -e "GRANT ALL PRIVILEGES ON *.* TO Account@'Allowed login address' IDENTIFIED BY 'Password'";
Example:
mysql -uroot -e "GRANT ALL PRIVILEGES ON *.* TO mha@'192.168.100.%' IDENTIFIED BY '123'";
```

Change the password of a MySQL database account.

```
update mysql.user set authentication_string=password('Password') where user='Account' and Host = 'localhost';
Example:
update mysql.user set authentication_string=password('123456') where user='repl' and Host = '192.168.100.%';
```

 NOTE

- If the password of the primary/standby replication account is changed, reconfigure the primary database information for the standby databases.

> CHANGE MASTER TO

```
MASTER_HOST='192.168.100.111',
MASTER_USER='repl',
MASTER_PASSWORD='Password',
MASTER_PORT=3306,
MASTER_CONNECT_RETRY=10,
MASTER_AUTO_POSITION=1;
```

> START SLAVE;

- After changing the passwords of the primary/standby replication account and MHA administrator account, change the passwords in the **app1.cnf** configuration file on the ECS (suffixed with **slave2**) with MHA Manager installed.

Run the **vim /datadisk/mha/conf/app1.cnf** command to change the value of **password**.

Working with MHA

Install MHA Manager on the standby database (suffixed with **slave2**). An MHA program can manage multiple sets of primary and standby databases. You need to create a different configuration file for each set. In the initial solution, a configuration file for only one set of primary and standby databases is available, the default MHA administrator account is **mha**, the password is the initial password, and the configuration file is stored in **/datadisk/mha/conf/app1.cnf**.

MHA commands:

Checking the SSH connection statuses of the primary and standby databases

```
masterha_check_ssh --conf=/datadisk/mha/conf/app1.cnf
```

If "All SSH connection tests passed successfully" is displayed, the three cloud servers can access each other without entering a password.

Figure 3-36 Checking SSH connection status

```
[root@mha-mysql-slave2 ~]# masterha_check_ssh --conf=/mha/conf/app1.cnf
Thu Sep 1 10:50:48 2022 - [warning] Global configuration file /etc/masterha_default.cnf not found. Skipping.
Thu Sep 1 10:50:48 2022 - [info] Reading application default configuration from /mha/conf/app1.cnf..
Thu Sep 1 10:50:48 2022 - [info] Reading server configuration from /mha/conf/app1.cnf..
Thu Sep 1 10:50:49 2022 - [info] Starting SSH connection tests..
Thu Sep 1 10:50:48 2022 - [debug] Connecting via SSH from root@192.168.100.111(192.168.100.111:22) to root@192.168.100.112(192.168.100.112:22)..
Thu Sep 1 10:50:49 2022 - [debug] ok.
Thu Sep 1 10:50:49 2022 - [debug] Connecting via SSH from root@192.168.100.111(192.168.100.111:22) to root@192.168.100.113(192.168.100.113:22)..
Thu Sep 1 10:50:49 2022 - [debug] ok.
Thu Sep 1 10:50:49 2022 - [debug] Connecting via SSH from root@192.168.100.112(192.168.100.112:22) to root@192.168.100.111(192.168.100.111:22)..
Thu Sep 1 10:50:49 2022 - [debug] ok.
Thu Sep 1 10:50:49 2022 - [debug] Connecting via SSH from root@192.168.100.112(192.168.100.112:22) to root@192.168.100.113(192.168.100.113:22)..
Thu Sep 1 10:50:50 2022 - [debug] ok.
Thu Sep 1 10:50:50 2022 - [debug] Connecting via SSH from root@192.168.100.113(192.168.100.113:22) to root@192.168.100.111(192.168.100.111:22)..
Thu Sep 1 10:50:50 2022 - [debug] ok.
Thu Sep 1 10:50:50 2022 - [debug] Connecting via SSH from root@192.168.100.113(192.168.100.113:22) to root@192.168.100.112(192.168.100.112:22)..
Thu Sep 1 10:50:51 2022 - [info] All SSH connection tests passed successfully.
[root@mha-mysql-slave2 ~]#
```

Checking the replication statuses of the primary and standby databases

```
masterha_check_repl --conf=/datadisk/mha/conf/app1.cnf
```

If "MySQL Replication Health is OK" is displayed, the primary/standby replication status is normal.

Figure 3-37 Checking the primary/standby replication status

```
Thu Sep 1 10:52:10 2022 - [info] Alive Servers:
Thu Sep 1 10:52:10 2022 - [info] 192.168.100.111(192.168.100.111:3306)
Thu Sep 1 10:52:10 2022 - [info] 192.168.100.112(192.168.100.112:3306)
Thu Sep 1 10:52:10 2022 - [info] 192.168.100.113(192.168.100.113:3306)
Thu Sep 1 10:52:10 2022 - [info] Alive Slaves:
Thu Sep 1 10:52:10 2022 - [info] 192.168.100.112(192.168.100.112:3306) Version=5.7.34-log (oldest major version be
Thu Sep 1 10:52:10 2022 - [info] GTID ON
Thu Sep 1 10:52:10 2022 - [info] Replicating from 192.168.100.111(192.168.100.111:3306)
Thu Sep 1 10:52:10 2022 - [info] Primary candidate for the new Master (candidate_master is set)
Thu Sep 1 10:52:10 2022 - [info] 192.168.100.113(192.168.100.113:3306) Version=5.7.34-log (oldest major version be
Thu Sep 1 10:52:10 2022 - [info] GTID ON
Thu Sep 1 10:52:10 2022 - [info] Replicating from 192.168.100.111(192.168.100.111:3306)
Thu Sep 1 10:52:10 2022 - [info] Current Alive Master: 192.168.100.111(192.168.100.111:3306)
Thu Sep 1 10:52:10 2022 - [info] Checking slave configurations..
Thu Sep 1 10:52:10 2022 - [info] read_only=1 is not set on slave 192.168.100.112(192.168.100.112:3306).
Thu Sep 1 10:52:10 2022 - [info] read_only=1 is not set on slave 192.168.100.113(192.168.100.113:3306).
Thu Sep 1 10:52:10 2022 - [info] Checking replication filtering settings..
Thu Sep 1 10:52:10 2022 - [info] binlog_do_db= , binlog_ignore_db=
Thu Sep 1 10:52:10 2022 - [info] Replication filtering check ok.
Thu Sep 1 10:52:10 2022 - [info] GTID (with auto-pos) is supported. Skipping all SSH and Node package checking.
Thu Sep 1 10:52:10 2022 - [info] Checking SSH publickey authentication settings on the current master..
Thu Sep 1 10:52:10 2022 - [info] HealthCheck: SSH to 192.168.100.111 is reachable.
Thu Sep 1 10:52:10 2022 - [info]
192.168.100.111(192.168.100.111:3306) (current master)
+-192.168.100.112(192.168.100.112:3306)
+--192.168.100.113(192.168.100.113:3306)

Thu Sep 1 10:52:10 2022 - [info] Checking replication health on 192.168.100.112..
Thu Sep 1 10:52:10 2022 - [info] ok.
Thu Sep 1 10:52:10 2022 - [info] Checking replication health on 192.168.100.113..
Thu Sep 1 10:52:10 2022 - [info] ok.
Thu Sep 1 10:52:10 2022 - [info] Checking master_ip_fallover script status:
Thu Sep 1 10:52:10 2022 - [info] /mha/scripts/master_ip_fallover --command=status --ssh_user=root --orig_master_hos
ig_master_port=3306

IN SCRIPT TEST====/sbin/ifconfig eth0:1 down==/sbin/ifconfig eth0:1 192.168.100.99/24===

Checking the Status of the script.. OK
Thu Sep 1 10:52:10 2022 - [info] OK.
Thu Sep 1 10:52:10 2022 - [warning] shutdown_script is not defined.
Thu Sep 1 10:52:10 2022 - [info] Got exit code 0 (Not master dead).

MySQL Replication Health is OK.
```

Starting MHA monitoring

```
mha_app1_start
```

Checking the MHA status

```
mha_app1_status
```

Stopping MHA monitoring

```
mha_app1_stop
```

Viewing MHA logs

```
tail -f /datadisk/mha/logs/manager
```

Simulating a Fault

Step 1 View log changes on the MHA Manager.

```
tail -f /datadisk/mha/logs/manager
```

Figure 3-38 Log changes

```
[root@mha-mysql-slave2 ~]# mha_app1_status
appl (pid:2426) is running(0:PING_OK), master:192.168.100.111
[root@mha-mysql-slave2 ~]# tail -f /mha/logs/manager

IN SCRIPT TEST====/sbin/ifconfig eth0:1 down==/sbin/ifconfig eth0:1 192.168.100.99/24===

Checking the Status of the script.. OK
Thu Sep 1 11:30:37 2022 - [info] OK.
Thu Sep 1 11:30:37 2022 - [warning] shutdown_script is not defined.
Thu Sep 1 11:30:37 2022 - [info] Set master ping interval 3 seconds.
Thu Sep 1 11:30:37 2022 - [warning] secondary_check_script is not defined. It is highly recommended setting it to check master reachability from two or more routes.
Thu Sep 1 11:30:37 2022 - [info] Starting ping health check on 192.168.100.111(192.168.100.111:3306)..
Thu Sep 1 11:30:37 2022 - [info] Ping(SELECT) succeeded, waiting until MySQL doesn't respond..
```

Step 2 Stop the primary database service.

```
systemctl stop mysqld
```

Step 3 Return to the ECS with MHA Manager installed, view the logs, and check whether the information "Master failover to xxxx completed successfully." is displayed.

Figure 3-39 Failover

```

Enabling the VIP - 192.168.100.99 on the new master - 192.168.100.112
Thu Sep 1 11:34:07 2022 - [info] OK.
Thu Sep 1 11:34:07 2022 - [info] ** Finished master recovery successfully.
Thu Sep 1 11:34:07 2022 - [info] * Phase 3: Master Recovery Phase completed.
Thu Sep 1 11:34:07 2022 - [info] * Phase 4: Slaves Recovery Phase..
Thu Sep 1 11:34:07 2022 - [info]
Thu Sep 1 11:34:07 2022 - [info] * Phase 4.1: Starting Slaves in parallel..
Thu Sep 1 11:34:07 2022 - [info] -- Slave recovery on host 192.168.100.113(192.168.100.113:3306) started, pid: 14227. Check tmp log /mha/logs/a
Thu Sep 1 11:34:09 2022 - [info] Log messages from 192.168.100.113 ...
Thu Sep 1 11:34:09 2022 - [info]
Thu Sep 1 11:34:07 2022 - [info] Resetting slave 192.168.100.113(192.168.100.113:3306) and starting replication from the new master 192.168.10
Thu Sep 1 11:34:07 2022 - [info] Executed CHANGE MASTER.
Thu Sep 1 11:34:08 2022 - [info] Slave started.
Thu Sep 1 11:34:08 2022 - [info] gtid_wait(2ed2aeb1-299c-11ed-8501-fa163e4d0914:1-2) completed on 192.168.100.113(192.168.100.113:3306). Execu
Thu Sep 1 11:34:09 2022 - [info] End of log messages from 192.168.100.113.
Thu Sep 1 11:34:09 2022 - [info] -- Slave on host 192.168.100.113(192.168.100.113:3306) started.
Thu Sep 1 11:34:09 2022 - [info] All new slave servers recovered successfully.
Thu Sep 1 11:34:09 2022 - [info]
Thu Sep 1 11:34:09 2022 - [info] * Phase 5: New master cleanup phase..
Thu Sep 1 11:34:09 2022 - [info]
Thu Sep 1 11:34:09 2022 - [info] Resetting slave info on the new master..
Thu Sep 1 11:34:09 2022 - [info] 192.168.100.112: Resetting slave info succeeded.
Thu Sep 1 11:34:09 2022 - [info] Master failover to 192.168.100.112(192.168.100.112:3306) completed successfully.
Thu Sep 1 11:34:09 2022 - [info] Deleted server1 entry from /mha/conf/app1.cnf .
Thu Sep 1 11:34:09 2022 - [info]

----- Failover Report -----
app1: MySQL Master failover 192.168.100.111(192.168.100.111:3306) to 192.168.100.112(192.168.100.112:3306) succeeded
Master 192.168.100.111(192.168.100.111:3306) is down!

Check MHA Manager logs at mha-mysql-slave2:/mha/logs/manager for details.

Started automated(non-interactive) failover.
Invalidated master IP address on 192.168.100.111(192.168.100.111:3306)
Selected 192.168.100.112(192.168.100.112:3306) as a new master.
192.168.100.112(192.168.100.112:3306): OK: Applying all logs succeeded.
192.168.100.112(192.168.100.112:3306): OK: Activated master IP address.
192.168.100.113(192.168.100.113:3306): OK: Slave started, replicating from 192.168.100.112(192.168.100.112:3306)
Master failover to 192.168.100.112(192.168.100.112:3306) completed successfully.

```

Step 4 Log in to the standby database suffixed with **slave1** and check whether the VIP is successfully bound.

Figure 3-40 VIP bound successfully

```

[root@mha-mysql-slave1 ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.100.112 netmask 255.255.255.0 broadcast 192.168.100.255
    inet6 fe80::f816:3eff:fe4d:915 prefixlen 64 scopeid 0x20<link>
    ether fa:16:3e:4d:09:15 txqueuelen 1000 (Ethernet)
    RX packets 500720 bytes 739812916 (705.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 32344 bytes 2830128 (2.6 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.100.99 netmask 255.255.255.0 broadcast 192.168.100.255
    ether fa:16:3e:4d:09:15 txqueuelen 1000 (Ethernet)

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 96 bytes 18912 (18.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 96 bytes 18912 (18.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Step 5 Check the status of the MHA Manager. Its status is **stopped**.

```
mha_app1_status
```

Figure 3-41 Completing a failover

```

[root@mha-mysql-slave2 ~]# mha_app1_status
app1 is stopped(2:NOT_RUNNING).
[root@mha-mysql-slave2 ~]#

```

Step 6 Log in to the recipient's mailbox to view the alarm email.

Figure 3-42 Alarm email

```
app1: MySQL Master failover 192.168.100.111(192.168.100.111:3306) to
192.168.100.112(192.168.100.112:3306) succeeded

Recipient

Master 192.168.100.111(192.168.100.111:3306) is down!

Check MHA Manager logs at mha-mysql-xyh-slave2:/mha/logs/manager for details.

Started automated(non-interactive) failover.
Invalidated master IP address on 192.168.100.111(192.168.100.111:3306)
Selected 192.168.100.112(192.168.100.112:3306) as a new master.
192.168.100.112(192.168.100.112:3306): OK: Applying all logs succeeded.
192.168.100.112(192.168.100.112:3306): OK: Activated master IP address.
192.168.100.113(192.168.100.113:3306): OK: Slave started, replicating from 192.168.100.112(192.168.100.112:3306)
192.168.100.112(192.168.100.112:3306): Resetting slave info succeeded.
Master failover to 192.168.100.112(192.168.100.112:3306) completed successfully.

----End
```

Rectifying a Fault

- Step 1** Restart the MySQL service for the original primary database and add it to the cluster as a standby database. (The IP address of the primary database has been updated.)

```
systemctl start mysqld.service
mysql -uroot
> CHANGE MASTER TO
  MASTER_HOST='192.168.100.112',
  MASTER_USER='repl',
MASTER_PASSWORD='Password',
  MASTER_PORT=3306,
  MASTER_CONNECT_RETRY=10,
  MASTER_AUTO_POSITION=1;
> START SLAVE;
> SHOW SLAVE STATUS\G;
```

- Step 2** Modify the MHA Manager configuration file to add the original primary database to the cluster.

```
vim /datadisk/mha/conf/app1.cnf
[server1]
candidate_master=1
check_repl_delay=0
hostname=192.168.100.111
port=3306
```

- Step 3** Restart the MHA service.

```
mha_app1_start
mha_app1_status
```

----End

Performing a Manual Switchover

Before performing a switchover, stop the MHA service.

```
mha_app1_stop
```

Run the following command on the MHA Manager to switch the primary and standby databases online:

```
masterha_master_switch --conf=/datadisk/mha/conf/app1.cnf --master_state=alive --
new_master_host=192.168.100.111 --orig_master_is_new_slave --running_updates_limit=10000 --
```


interactive=0

If the information "Switching master to 192.168.0.111(192.168.0.111:3306) completed successfully." is displayed, the switchover is successful.

Figure 3-43 Performing a switchover online

```
*****
Enabling the VIP - 192.168.100.99/24 on new master: 192.168.100.111
Enabled the VIP successfully
*****

Thu Sep 1 11:52:11 2022 - [info] ok.
Thu Sep 1 11:52:11 2022 - [info]
Thu Sep 1 11:52:11 2022 - [info] * Switching slaves in parallel..
Thu Sep 1 11:52:11 2022 - [info] -- Slave switch on host 192.168.100.113(192.168.100.113:3306) started, pid: 15196
Thu Sep 1 11:52:11 2022 - [info]
Thu Sep 1 11:52:13 2022 - [info] Log messages from 192.168.100.113 ...
Thu Sep 1 11:52:13 2022 - [info]
Thu Sep 1 11:52:11 2022 - [info] Waiting to execute all relay logs on 192.168.100.113(192.168.100.113:3306)..
Thu Sep 1 11:52:11 2022 - [info] master_pos_wait(mysql_bin.000003:194) completed on 192.168.100.113(192.168.100.113:3306). Executed 0 events.
Thu Sep 1 11:52:11 2022 - [info] done.
Thu Sep 1 11:52:11 2022 - [info] Resetting slave 192.168.100.113(192.168.100.113:3306) and starting replication from the new master 192.168.100.111(192.168.100.111:3306)..
Thu Sep 1 11:52:11 2022 - [info] Executed CHANGE MASTER.
Thu Sep 1 11:52:12 2022 - [info] Slave started.
Thu Sep 1 11:52:12 2022 - [info] End of log messages from 192.168.100.113 ...
Thu Sep 1 11:52:13 2022 - [info] -- Slave switch on host 192.168.100.113(192.168.100.113:3306) succeeded.
Thu Sep 1 11:52:13 2022 - [info] Unlocking all tables on the orig master:
Thu Sep 1 11:52:13 2022 - [info] Executing UNLOCK TABLES..
Thu Sep 1 11:52:13 2022 - [info] ok.
Thu Sep 1 11:52:13 2022 - [info] Starting orig master as a new slave..
Thu Sep 1 11:52:13 2022 - [info] Resetting slave 192.168.100.112(192.168.100.112:3306) and starting replication from the new master 192.168.100.111(192.168.100.111:3306)..
Thu Sep 1 11:52:13 2022 - [info] Executed CHANGE MASTER.
Thu Sep 1 11:52:14 2022 - [info] Slave started.
Thu Sep 1 11:52:14 2022 - [info] All new slave servers switched successfully.
Thu Sep 1 11:52:14 2022 - [info]
Thu Sep 1 11:52:14 2022 - [info] * Phase 5: New master cleanup phase..
Thu Sep 1 11:52:14 2022 - [info]
Thu Sep 1 11:52:14 2022 - [info] 192.168.100.111: Resetting slave info succeeded.
Thu Sep 1 11:52:14 2022 - [info] Switching master to 192.168.100.111(192.168.100.111:3306) completed successfully.
[root@mha-mysql-slave2 ~]#
```

Check whether the VIP is bound to the new primary database (192.168.100.111).

ifconfig

Restart MHA on the ECS where MHA Manager is installed.

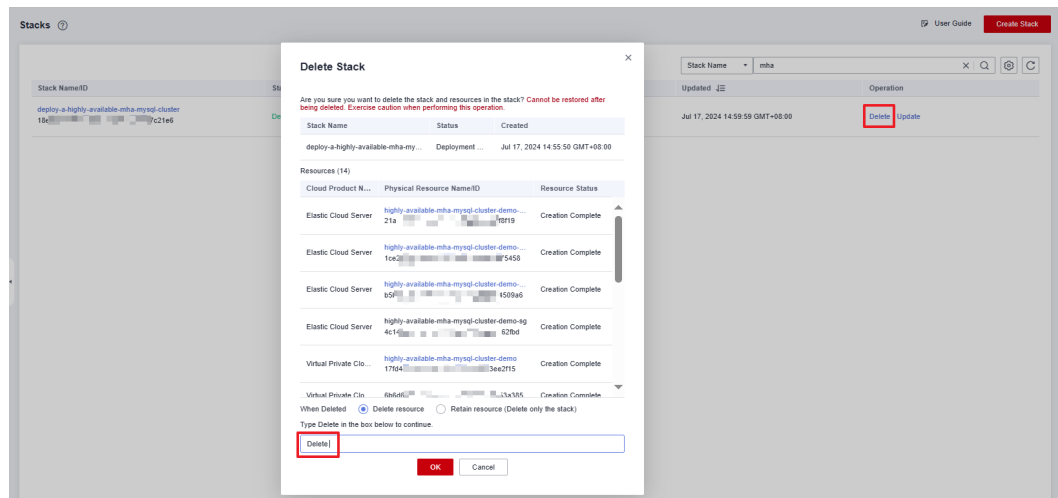
mha_app1_start
mha_app1_status

3.4 Quick Uninstallation

Deleting the Solution

- Step 1** Log in to **Resource Formation Service (RFS) console**, locate the resource stack created for the solution, and click **Delete** in the **Operation** column. In the displayed **Delete Stack** dialog box, enter **Delete** and click **OK** to uninstall the solution.

Figure 3-44 Delete Stack dialog box



----End

4 Appendix

Terms

- **Flexus X Instance (FlexusX):** FlexusX is a next-generation flexible cloud server service designed for small- and medium-sized enterprises (SMEs) and developers. FlexusX provides functions similar to what ECS provides. It supports flexible vCPU/memory ratios, performance mode, and live specification change.
- **Elastic Cloud Server (ECS):** ECS provides secure, scalable, on-demand compute resources, enabling you to flexibly deploy applications and workloads.
- **Cloud server group:** A cloud server group allows you to create cloud servers on different hosts to improve service reliability. With the anti-affinity policy supported, cloud servers in the same cloud server group are deployed on different hosts for higher service reliability.
- **Elastic IP (EIP):** EIP provides static public IP addresses and scalable bandwidths that enable your cloud resources to communicate with the Internet. You can easily bind an EIP to an ECS, BMS, virtual IP address, NAT gateway, or load balancer, enabling immediate Internet access.
- **MHA:** MHA performs automated failover and standby-to-primary promotion with minimal downtime, usually within 10-30 seconds. MHA prevents replication consistency problems and saves on expenses of having to acquire additional servers. All this with zero performance degradation, no complexity (easy-to-install) and requiring no change to existing deployments.

5 Change History

Released On	Change History
2023-04-30	This issue is the first official release.
2024-07-30	Supported Flexus X Instance.