ALIBABA CLOUD

阿里云

加密服务
用户指南

文档版本：20220712

阿里云

法律声明

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

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

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

目录

1.开通安全审计服务

购买加密服务后，您可以开通安全审计服务，将密码机实例的运行信息自动保存到对象存储OSS（Object Storage Service）中，并以特定的审计日志格式进行持久化储存，以满足合规和审计需求。本文介绍如何开通安全审计服务。

前提条件

- 已购买加密服务。具体操作，请参见购买加密服务。
- 已开通OSS服务并且已创建存储空间（Bucket）。具体操作，请参见开通OSS服务和创建存储空间。

> **注意** Bucket所属地域需要和将要开通安全审计服务的地域相同。

使用限制

- 安全审计服务只支持同地域开通，不支持跨地域开通。例如，您在地域A和地域B的密码机实例都需要开通安全审计服务，那么您需要在地域A和地域B分别开通安全审计服务。
- 目前安全审计服务为Beta版本，仅提供给GVSM和EVSM使用。

操作步骤

1. 登录加密服务管理控制台。
2. 在顶部导航栏，选择目标地域，然后在左侧导航栏，选择**安全审计服务（Beta）**。

   目前支持开通安全审计服务的地域包括：**华东1（杭州）**、**华东2（上海）**、**华北2（北京）**、**华南1（深圳）**。
3. 在**安全审计服务**页面，单击**为地域开通安全审计**，然后单击**授权**。

   完成**授权**后，加密服务会为您自动创建一个服务关联角色**AliyunServiceRoleForHSMLogDelivery**，并且该角色拥有您的OSS服务中Bucket的读写权限。关于加密服务服务管理角色的更多信息，请参见加密服务服务关联角色。
4. 在**OSS Bucket**下拉列表，选择存储密码机实例审计日志的Bucket，单击**确定**。

操作结果

在**安全审计服务**页面，开关图标变为绿色并显示**已开通**。同时，在**审计日志OSS投递规则**区域，您可以看到存储当前地域中所有密码机实例的审计日志的Bucket。

关闭安全审计服务

如果您需要关闭安全审计服务，在**安全审计服务**页面，单击**已开通**右侧开关图标，并在**关闭安全审计服务**对话框中单击**关闭**。

2.使用密码机实例

密码机实例是密码机的硬件加密模块虚拟化形成的资源，具备对数据的加解密运算能力，可以实现加密服务的所有功能。本文介绍如何使用密码机实例。

前提条件

您已经创建专有网络VPC（Virtual Private Cloud）。具体操作，请参见创建专有网络。

步骤一：创建密码机实例

1. 登录加密服务管理控制台，在顶部菜单栏，选择目标地域。
2. 在**实例列表**页面，单击**创建密码机实例**。
3. 在加密服务购买页面，参考下表配置信息，然后单击**立即购买**并完成支付。

配置项	说明
区域	选择开通加密服务的地域，包括**华南1（深圳）**、**华北2（北京）**、**华东2（上海）**、**华东1（杭州）**。 加密服务只能在VPC中使用，且加密服务的地域必须与您的ECS及VPC的地域相同。
可用区	选择开通加密服务的可用区。建议您将密码机部署在不同的可用区，保障您的业务不会因可用区内某一机房发生事故而受影响。 **说明** ○ 只有处于同一地域的可用区之间才能实现网络互通。 ○ 加密服务和ECS实例可以在不同的可用区。
密码机类型	选择设备型号，包括**金融数据密码机EVSM**、**通过服务器密码机GVSM**和**签名验证服务器SVSM**。关于密码机设备类型的更多信息，请参见功能特性。
购买数量	选择需要购买的密码机数量。如果您在生产环境中使用加密服务，为了保证加密服务的高可用性，建议您至少购买2个密码机实例。
购买时长	选择购买的有效服务时间。 为了防止加密服务到期未及时续费而导致的密钥永久性丢失，建议您购买时选择**到期自动续费**。当您选择**到期自动续费**后，阿里云会在服务到期前9个自然日从您购买密码机时使用的支付账户自动扣款，为了防止扣费失败，请确保您的支付账户余额充足。

创建成功后，密码机实例将显示在**实例列表**页面，并且**状态**为**未启用**。

步骤二：启用并配置密码机实例

1. 登录加密服务管理控制台，在顶部菜单栏，选择目标地域。
2. 在**实例列表**页面，找到创建后的密码机实例，在**操作**列单击**启用**。

3. 在**密码机实例配置**对话框，配置密码机实例，然后单击**确定**。

配置项	说明
所属的VPC网络ID	选择密码机实例需要绑定的VPC。
VPC子网	选择密码机实例所属的VPC子网IP。
私网IP地址	为密码机实例分配一个所属的私网IP。 **注意** • 您分配给密码机实例的私网IP地址必须是所属的VPC子网网段中的IP地址，否则配置将会失败。 • 末位为253、254、255的IP地址为系统保留地址。当您使用系统保留地址的网络时，该密码机实例只能在局域网内部进行通信，不能与其他网络互连。为了不影响网络互通，在分配私网IP地址时，请不要分配系统保留地址。
设置密码机实例白名单	设置访问该密码机实例的访问白名单，不在白名单内的访问请求将被拒绝。

配置成功后，密码机实例的**状态**变为**已启用**。

步骤三：配置密码机实例管理工具

您可以使用密码机实例管理工具来管理和配置密码机，包括密钥管理、配置服务器端口等。

注意 密码机实例管理工具仅支持安装在Windows操作系统。

1. 登录加密服务管理控制台，在顶部菜单栏，选择目标地域。
2. 下载并安装密码机实例管理工具。
   i. 在**实例列表**页面，找到目标密码机实例。
   ii. 单击密码机实例的规格列信息，然后单击**下载密码机实例管理工具**。

   iii. 解压获取到的密码机软件包，选择以下任一方式安装密码机实例管理工具。
   - 在本地终端安装密码机客户端管理工具，然后通过VPN或物理专线使本地终端连接到密码机实例所属的VPC网络。
   - 在ECS实例上安装密码机客户端管理工具，然后通过本地终端远程登录ECS实例，在ECS实例上操作密码机实例管理工具。购买ECS实例的具体操作，请参见使用向导创建实例。
3. 登录密码机实例管理工具，配置服务端口属性。具体操作，请参见密码机实例管理工具用户管理手册的《登录》章节和《设备配置》章节。

您可以在解压后的密码机软件包中获取密码机实例管理工具用户管理手册。

步骤四：使用密码机实例

您可以从步骤三：配置密码机实例管理工具获取到的密码机软件包中找到密码机的开发手册、SDK测试程序等。在完成密码机实例配置后，您可以参考开发手册，调用API接口使用密码机实例。

以GVSM为例，您可以使用其中的测试用例*JceTestMain.java*来测试密码机实例。代码示例如下：

> **注意** 您需要将用例中的IP地址（121.41.XX.XX）修改为密码机实例分配的私网IP地址。

```
import cn.com.tass.jce.castle.hsm.connector.pool.PooledConfigReader;
import cn.com.tass.jce.castle.tc.encodings.Hex;
import javax.crypto.*;
import java.security.*;
import java.util.*;
public class APITest {
    public static void main(String[] args) throws Exception {
        //加载配置
        String config = "{"
                    + "[LOGGER];"
                    + "logsw=debug,error;"
                    + "logPath=./;"
                    + "[HOST 1];"
                    + "hsmModel=GHSM;"
                    + "host=121.41.XX.XX;"
                    + "linkNum=-3;"
                    + "port=9021;"
                    + "timeout=5;"
                    + "}";
        PooledConfigReader.setConfig(config);
        //产生随机数
        testGenRandom();
        //SM2签名 验签
        testOutKeySM2SignAndVeirfy();
        //对称密钥加解密数据
        testOutKeyAES128EncAndDec();
    }
    /**
     * 外部密钥AES128加解密
     */
    public static void testOutKeyAES128EncAndDec() {
        try {
            byte[] myAAD = Hex.decode("31313131313131313131313131313131");
            //产生密钥
            KeyGenerator keyGenerator = KeyGenerator.getInstance("AES", "TASS");
            keyGenerator.init(128);
            SecretKey key = keyGenerator.generateKey();
            byte[] iv = Hex.decode("31313131313131313131313131313131");
            byte[] plain = Hex.decode("31313131313131313131313131313131");
            //调用接口进行内部密钥加解密
            symEncAndDec("AES128 out key", key, "AES", plain, iv, myAAD);
        } catch (Exception e) {
            e.printStackTrace();
```

```
        }
    }
    /**
     * ECB、CBC 对称加密
     *
     * @param flag  标识
     * @param key   密钥
     * @param alg   算法名称
     * @param plain 待加密的数据
     * @param iv    IV值
     * @param add   GCM模式下的add值
     */
    private static void symEncAndDec(String flag, Key key, String alg, byte[] plain, byte[]
iv, byte[] add) {
        try {
            //ECB/NoPadding
            Cipher encipher = Cipher.getInstance(alg + "/ECB/NoPadding", "TASS");
            Cipher decipher = Cipher.getInstance(alg + "/ECB/NoPadding", "TASS");
            encipher.init(Cipher.ENCRYPT_MODE, key);
            decipher.init(Cipher.DECRYPT_MODE, key);
            //加密
            byte[] resultEnc = encipher.doFinal(plain);
            //System.out.println("("+flag+")"+alg + "/ECB/NoPadding" + " enc result = " + n
ew String(Hex.encode(resultEnc)));
            //解密
            byte[] resultDec = decipher.doFinal(resultEnc);
            //System.out.println("("+flag+")"+alg + "/ECB/NoPadding" + " dec result = " + n
ew String(Hex.encode(resultDec)));
            if (!verifyResult(plain, resultDec)) {
                System.err.println("(" + flag + ")" + alg + "/ECB/NoPadding" + "解密失败");
                System.out.print("\n");
            } else {
                System.out.println("(" + flag + ")" + alg + "/ECB/NoPadding" + "解密成功");
                System.out.print("\n");
            }
            byte[] newPlain = Hex.decode("313131313131313131313131313131313233");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    public static boolean verifyResult(byte[] enc, byte[] dec) {
        return Arrays.equals(enc, dec);
    }
    /**
     * SM2外部密钥签名验签
     */
    public static void testOutKeySM2SignAndVeirfy() {
        try {
            //待签名的原文数据
            byte[] src = Hex.decode("313233");
            //使用KeyPairGenerator产生密钥对充当外部密钥
            KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance("SM2", "TASS")
;
            keyPairGenerator.initialize(256);
```

```
            KeyPair keyPair = keyPairGenerator.generateKeyPair();
            //开始签名
            Signature sign = Signature.getInstance("SM3withSM2", "TASS");
            sign.initSign(keyPair.getPrivate());
            sign.update(src);
            byte[] signResult = sign.sign();
            System.out.println("SM3withSM2 out key sign result = " + new String(Hex.encode(
signResult)));
            //开始验签
            Signature verify = Signature.getInstance("SM3withSM2", "TASS");
            verify.initVerify(keyPair.getPublic());
            verify.update(src);
            boolean verifyResult = verify.verify(signResult);
            System.out.println("SM3withSM2 out key verify result = " + verifyResult);
            //计算摘要值
            MessageDigest digest = MessageDigest.getInstance("SM3", "TASS");
            byte[] digestData = digest.digest(src);
            //SM3withSM2ForHash开始签名
            sign = Signature.getInstance("SM3withSM2ForHash", "TASS");
            sign.initSign(keyPair.getPrivate());
            sign.update(digestData);
            signResult = sign.sign();
            System.out.println("SM3withSM2ForHash int key sign result = " + new String(Hex.
encode(signResult)));
            //SM3withSM2ForHash开始验签
            verify = Signature.getInstance("SM3withSM2ForHash", "TASS");
            verify.initVerify(keyPair.getPublic());
            verify.update(digestData);
            verifyResult = verify.verify(signResult);
            System.out.println("SM3withSM2ForHash int key verify result = " + verifyResult)
;
            System.out.print("\n");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    /**
     * 产生随机数
     */
    public static void testGenRandom() {
        try {
            SecureRandom instance = SecureRandom.getInstance("HSM", "TASS");
            instance.nextInt();
            byte[] bytes = new byte[1024];
            instance.nextBytes(bytes);
            //输出随机数
            System.out.println("1024 random : " + new String(Hex.encode(bytes)));
            System.out.print("\n");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

更多操作

为密码机实例创建集群

您可以将处于不同的密码机实例加入到同一集群进行统一管理，为业务应用提供密码计算的高可用性、负载均衡以及横向扩展的能力。创建集群的具体操作，请参见使用密码机实例集群。

修改密码机实例配置

密码机实例未加入集群时，您可以修改密码机实例所属的VPC、VPC子网、私网IP地址和密码机实例白名单；密码机实例加入集群后，您只能修改密码机实例的私网IP地址。

1. 在**实例列表**页面，找到目标密码机实例，在**操作**列选择 ⋮ > **配置**。
2. 在**密码机实例配置**对话框，修改密码机实例的配置，然后单击**确定**。

停用密码机实例

重置密码机实例或者从集群中删除密码机实例前，您需要先停用密码机实例的业务功能。

> **注意**
> - 停用会断开密码机实例的网络连接，请谨慎操作。
> - 不允许停用集群中的主密码机实例。

1. 在**实例列表**页面，找到目标密码机实例，在**操作**列单击**停用**。
2. 在弹出的对话框，再次单击**停用**。

   停用后，密码机实例的**状态**列显示为**已停用**。

重置密码机实例

停用密码机实例后，您可以通过重置密码机实例，恢复密码机实例为出厂状态，即未初始化状态。

> **注意** 重置将清空密码机实例中的数据，并恢复到出厂状态，请谨慎操作。

1. 在**实例列表**页面，找到目标密码机实例，在**操作**列单击**重置**。
2. 在弹出的对话框，再次单击**重置**。

恢复密码机实例

通过恢复密码机实例，重新启用已被停用的密码机实例业务。

1. 在**实例列表**页面，找到目标密码机实例，在**操作**列单击**恢复**。
2. 在弹出的对话框，再次单击**恢复**。

   恢复后，密码机实例的**状态**列显示为**已启用**。

3.使用密码机实例集群

加密服务为您提供密码机实例集群功能，实现将处于同一地域不同可用区、用于相同业务的一组密码机实例关联起来，进行统一管理，为业务应用提供密码计算的高可用性、负载均衡以及横向扩展的能力。本文介绍如何使用密码机实例集群。

适用场景

- 应用程序可以通过集群内的任何一个密码机实例使用同一个密钥。
- 应用程序用于生产环境，从而要求加密服务提供业务连续性。

前提条件

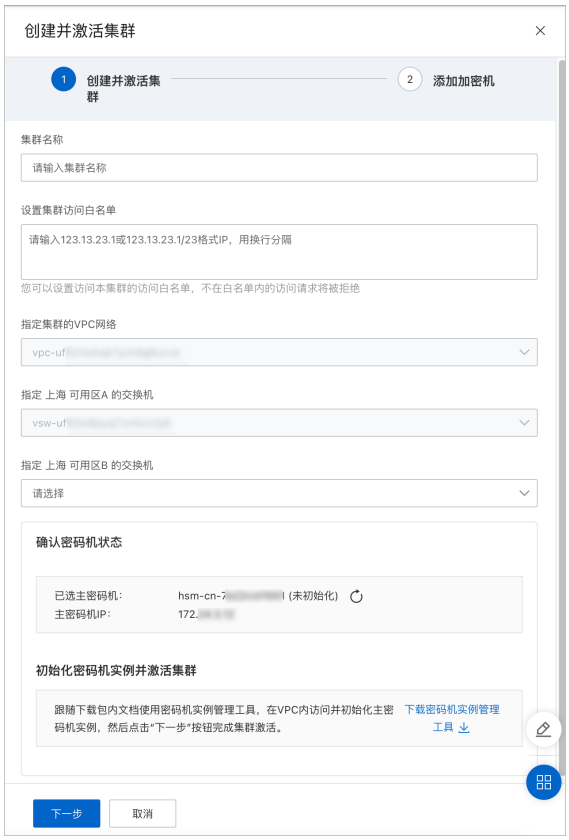
您已经创建密码机实例。具体操作，请参见步骤一：创建密码机实例。

创建并激活集群

一个集群中包括一个主密码机实例与若干个非主密码机实例。集群内一个可用区的密码机实例使用同一子网。您可以通过同步集群的操作，将主密码机实例的数据、状态同步到其他非主密码机实例，包括但不限于密钥，应用许可。

1. 登录加密服务管理控制台，在顶部菜单栏，选择目标地域。
2. 在**实例列表**，找到目标密码机实例，在**操作**列单击**创建集群**。
3. 在**创建并激活集群**面板，创建集群信息并激活集群，然后单击**下一步**。

i. 设置**集群名称**和**设置集群访问白名单**。



- **集群名称**：集群的名称不能重复且长度不允许超过24个字符。
- **设置集群访问白名单**：允许访问集群的IP地址，如果没有设置白名单，则所有IP地址都能访问集群。

> **注意** 集群的白名单优先级高于集群中密码机实例的白名单。例如，您设置的集群中密码机实例的白名单为10.10.10.10，集群的白名单为172.16.0.1，则您只能通过172.16.0.1访问集群中的密码机实例。

ii. 配置试用集群所在地域的另一可用区的交换机。

在密码机实例集群中，您必须要配置2个交换机才能成功创建并激活集群。

iii. 初始化主密码机，然后单击**下一步**。

只有**已选主密码机**状态显示初始化时，才能激活集群。初始化主密码机的具体操作如下：

单击**下载密码机实例管理工具**，解压管理工具包后，在**管理工具**文件夹中找到管理客户端软件的用户管理手册，参考用户管理手册中的**快速初始化**或**原始初始化**章节完成初始化操作。

> **注意** 密码机客户端管理工具只支持运行在Windows系统。

iv. （可选）根据页面提示，添加密码机实例到试用集群。

如果需要更多的密码机实例，您需要购买密码机实例，并添加密码机实例到集群。

v. 单击**完成**。

更多操作

扩展集群

您可以将处于不同的密码机实例加入到同一集群进行统一管理，提高加密服务的高可用性。添加到集群的密码机实例需符合以下要求：

- 密码机实例未初始化。如果已初始化，您需要先恢复该密码机实例为未初始化状态，具体操作，请参见重置密码机实例。
- 密码机实例为**已启用**或**未启用**状态。
- 密码机实例与主密码机为同一类型密码机。
- 密码机实例未配置交换机或与主密码机属于同一交换机。

> **说明** 如果密码机实例已配置了密码机实例白名单，加入集群后该密码机实例的访问白名单将继承集群的访问白名单，原密码机实例白名单将被清空。

1. 在**实例列表**页面，找到目标主密码机实例，在**操作**列单击**扩展集群**。
2. 通过以下方式添加密码机实例到集群。
   - 还未购买密码机实例时，在**添加密码机实例到集群**对话框，单击**订购密码机**，新购密码机实例。具体操作，请参见购买加密服务。

     购买的密码机实例会自动添加到集群，同时加密服务会自动为该密码机实例分配IP地址，并完成集群内数据的同步。
   - 已购买密码机实例时，在**添加密码机实例到集群**对话框，选择需要添加的密码机实例，单击，然后单击**确定**。

同步集群

主密码机数据变更后，您需要同步集群的数据到子密码机，实现实例数据的高可用性。当您第一次创建并激活集群后，您需要将集群的主密码机数据手动同步到集群的子密码机实例。当您对集群进行扩容时，集群数据会自动同步到新添加的密码机实例。

> **注意** 同步集群预计需要5分钟，请在业务空闲期进行同步，以免影响业务运行。

1. 在**实例列表**页面，找到目标主密码机实例，在**操作**列单击**同步集群**。
2. 在弹出的对话框，再次单击**同步集群**。

将子密码机切换为主机

您可以手动将子密码机切换为集群中的主机。

1. 在**实例列表**页面，找到目标子密码机实例，在**操作**列单击**切换主机**。
2. 在弹出的对话框，单击**切换**。

移出集群

移出集群中的密码机实例时，您需要先移出子密码机实例，最后移出主密码机实例。当集群中只有主密码机实例时，主密码机实例可以直接被移出。在主密码机实例被移出之后，集群会自动删除。

1. 停用密码机，具体操作，请参见停用密码机实例。

   > **说明** 您需要先停用密码机后，才可以将密码机移出集群。

2. 在**实例列表**页面，找到目标子密码机实例，在**操作**列单击**移出集群**。

3. 在弹出的对话框，再次单击**移出集群**。

修改集群名称和访问白名单

1. 在**实例列表**页面，找到目标主密码机实例，在实例ID右侧单击⋯。

2. 在弹出的对话框，单击✎图标，编辑集群名称和访问白名单。