阿里云
云服务器 ECS

最佳实践

文档版本：20181109

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
	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。	 **禁止：** 重置操作将丢失用户配置数据。
	该类警示信息可能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，恢复业务所需时间约10分钟。
	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。	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。	单击 **确定**。
`courier` 字体	命令。	执行 `cd /d C:/windows` 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。	`swich` *`{stand | slave}`*

目录

1 安全

1.1 ECS安全组实践（一）

本文主要介绍如何配置安全组的入网规则。

在云端安全组提供类似虚拟防火墙功能，用于设置单个或多个 ECS 实例的网络访问控制，是重要的安全隔离手段。创建 ECS 实例时，您必须选择一个安全组。您还可以添加安全组规则，对某个安全组下的所有 ECS 实例的出方向和入方向进行网络控制。

在配置安全组的入网规则之前，您应已经了解以下安全组相关的信息：

- 安全组限制
- 安全组默认规则
- 设置安全组 *In* 方向的访问权限
- 设置安全组 *Out* 方向的访问权限

安全组实践的基本建议

在开始安全组的实践之前，下面有一些基本的建议：

- 最重要的规则：安全组应作为白名单使用。
- 开放应用出入规则时应遵循“最小授权”原则，例如，您可以选择开放具体的端口（如 80 端口）。
- 不应使用一个安全组管理所有应用，因为不同的分层一定有不同的需求。
- 对于分布式应用来说，不同的应用类型应该使用不同的安全组，例如，您应对 Web、Service、Database、Cache 层使用不同的安全组，暴露不同的出入规则和权限。
- 没有必要为每个实例单独设置一个安全组，控制管理成本。
- 优先考虑 VPC 网络。
- 不需要公网访问的资源不应提供公网 IP。
- 尽可能保持单个安全组的规则简洁。因为一个实例最多可以加入 5 个安全组，一个安全组最多可以包括 100 个安全组规则，所以一个实例可能同时应用数百条安全组规则。您可以聚合所有分配的安全规则以判断是否允许流入或留出，但是，如果单个安全组规则很复杂，就会增加管理的复杂度。所以，应尽可能地保持单个安全组的规则简洁。
- 阿里云的控制台提供了克隆安全组和安全组规则的功能。如果您想要修改线上的安全组和规则，您应先克隆一个安全组，再在克隆的安全组上进行调试，从而避免直接影响线上应用。

说明：

调整线上的安全组的出入规则是比较危险的动作。如果您无法确定，不应随意更新安全组出入规则的设置。

设置安全组的入网规则

以下是安全组的入网规则的实践建议。

不要使用 0.0.0.0/0 的入网规则

允许全部入网访问是经常犯的错误。使用 0.0.0.0/0 意味着所有的端口都对外暴露了访问权限。这是非常不安全的。正确的做法是，先拒绝所有的端口对外开放。安全组应该是白名单访问。例如，如果您需要暴露 Web 服务，默认情况下可以只开放 80、8080 和 443 之类的常用TCP端口，其它的端口都应关闭。

```
{ "IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "
SourceCidrIp" : "0.0.0.0/0", "Policy": "accept"} ,
{ "IpProtocol" : "tcp", "FromPort" : "8080", "ToPort" : "8080", "
SourceCidrIp" : "0.0.0.0/0", "Policy": "accept"} ,
{ "IpProtocol" : "tcp", "FromPort" : "443", "ToPort" : "443", "
SourceCidrIp" : "0.0.0.0/0", "Policy": "accept"} ,
```

关闭不需要的入网规则

如果您当前使用的入规则已经包含了 0.0.0.0/0，您需要重新审视自己的应用需要对外暴露的端口和服务。如果确定不想让某些端口直接对外提供服务，您可以加一条拒绝的规则。比如，如果您的服务器上安装了 MySQL 数据库服务，默认情况下您不应该将 3306 端口暴露到公网，此时，您可以添加一条拒绝规则，如下所示，并将其优先级设为100，即优先级最低。

```
{ "IpProtocol" : "tcp", "FromPort" : "3306", "ToPort" : "3306", "
SourceCidrIp" : "0.0.0.0/0", "Policy": "drop", Priority: 100} ,
```

上面的调整会导致所有的端口都不能访问 3306 端口，极有可能会阻止您正常的业务需求。此时，您可以通过授权另外一个安全组的资源进行入规则访问。

授权另外一个安全组入网访问

不同的安全组按照最小原则开放相应的出入规则。对于不同的应用分层应该使用不同的安全组，不同的安全组应有相应的出入规则。

例如，如果是分布式应用，您会区分不同的安全组，但是，不同的安全组可能网络不通，此时您不应该直接授权 IP 或者 CIDR 网段，而是直接授权另外一个安全组 ID 的所有的资源都可以直接访

问。比如，您的应用对 Web、Database 分别创建了不同的安全组：sg-web 和 sg-database。在sg-database 中，您可以添加如下规则，授权所有的 sg-web 安全组的资源访问您的 3306 端口。

```
{ "IpProtocol" : "tcp", "FromPort" : "3306", "ToPort" : "3306", "
SourceGroupId" : "sg-web", "Policy": "accept", Priority: 2} ,
```

授权另外一个 CIDR 可以入网访问

经典网络中，因为网段不太可控，建议您使用安全组 ID 来授信入网规则。

VPC 网络中，您可以自己通过不同的 VSwitch 设置不同的 IP 域，规划 IP 地址。所以，在 VPC 网络中，您可以默认拒绝所有的访问，再授信自己的专有网络的网段访问，直接授信可以相信的 CIDR 网段。

```
{ "IpProtocol" : "icmp", "FromPort" : "-1", "ToPort" : "-1", "
SourceCidrIp" : "10.0.0.0/24", Priority: 2} ,
{ "IpProtocol" : "tcp", "FromPort" : "0", "ToPort" : "65535", "
SourceCidrIp" : "10.0.0.0/24", Priority: 2} ,
{ "IpProtocol" : "udp", "FromPort" : "0", "ToPort" : "65535", "
SourceCidrIp" : "10.0.0.0/24", Priority: 2} ,
```

变更安全组规则步骤和说明

变更安全组规则可能会影响您的实例间的网络通信。为了保证必要的网络通信不受影响，您应先尝试以下方法放行必要的实例，再执行安全组策略收紧变更。

 **说明：**

执行收紧变更后，应观察一段时间，确认业务应用无异常后再执行其它必要的变更。

- 新建一个安全组，将需要互通访问的实例加入这个安全组，再执行变更操作。
- 如果授权类型为 **安全组访问**，则将需要互通访问的对端实例所绑定的安全组 ID 添加为授权对象；
- 如果授权类型为 **地址段访问**，则将需要互通访问的对端实例内网 IP 添加为授权对象。

具体操作指引请参见 经典网络内网实例互通设置方法。

1.2 ECS安全组实践（二）

本文将介绍安全组的以下几个内容：

- 授权 和 撤销 安全组规则。
- 加入安全组 和 离开安全组。

阿里云的网络类型分为 **经典网络** 和 **VPC**，它们对安全组支持不同的设置规则：

- 如果是经典网络，您可以设置以下几个规则：内网入方向、内网出方向、公网入方向和公网出方向。
- 如果是 VPC 网络，您可以设置：入方向 和 出方向。

安全组内网通讯的概念

本文开始之前，您应知道以下几个安全组内网通讯的概念：

- 默认只有同一个安全组的 ECS 实例可以网络互通。即使是同一个账户下的 ECS 实例，如果分属不同安全组，内网网络也是不通的。这个对于经典网络和 VPC 网络都适用。所以，经典网络的 ECS 实例也是内网安全的。
- 如果您有两台 ECS 实例，不在同一个安全组，您希望它们内网不互通，但实际上它们却内网互通，那么，您需要检查您的安全组内网规则设置。如果内网协议存在下面的协议，建议您重新设置。
  - 允许所有端口；
  - 授权对象为 CIDR 网段 (SourceCidrIp)：*0.0.0.0/0* 或者 *10.0.0.0/8* 的规则。 如果是经典网络，上述协议会造成您的内网暴露给其它的访问。
- 如果您想实现在不同安全组的资源之间的网络互通，您应使用安全组方式授权。对于内网访问，您应使用源安全组授权，而不是 CIDR 网段授权。

安全规则的属性

安全规则主要是描述不同的访问权限，包括如下属性:

- Policy：授权策略，参数值可以是 *accept*（接受）或 *drop*（拒绝）。
- Priority：优先级，根据安全组规则的创建时间降序排序匹配。规则优先级可选范围为 1-100，默认值为 1，即最高优先级。数字越大，代表优先级越低。
- NicType：网络类型。如果只指定了 SourceGroupId 而没有指定 SourceCidrIp，表示通过安全组方式授权，此时，NicType 必须指定为 *intranet*。
- 规则描述：
  - IpProtocol：IP 协议，取值：*tcp*、*udp*、*icmp*、*gre* 或 *all*。all 表示所有的协议。
  - PortRange：IP 协议相关的端口号范围：
    - IpProtocol 取值为 *tcp* 或 *udp* 时，端口号取值范围为 1~65535，格式必须是“起始端口号/终止端口号”，如“1/200”表示端口号范围为1~200。如果输入值为“200/1”，接口调用将报错。

- IpProtocol 取值为 *icmp*、*gre* 或 *all* 时，端口号范围值为 -1/-1，表示不限制端口。

— 如果通过安全组授权，应指定 SourceGroupId，即源安全组 ID。此时，根据是否跨账号授权，您可以选择设置源安全组所属的账号 SourceGroupOwnerAccount；

— 如果通过 CIDR 授权，应指定 SourceCidrIp，即源 IP 地址段，必须使用 CIDR 格式。

授权一条入网请求规则

在控制台或者通过 API 创建一个安全组时，入网方向默认 *deny all*，即默认情况下您拒绝所有入网请求。这并不适用于所有的情况，所以您要适度地配置您的入网规则。

比如，如果您需要开启公网的 80 端口对外提供 HTTP 服务，因为是公网访问，您希望入网尽可能多访问，所以在 IP 网段上不应做限制，可以设置为 *0.0.0.0/0*，具体设置可以参考以下描述，其中，括号外为控制台参数，括号内为 OpenAPI 参数，两者相同就不做区分。

- 网卡类型（NicType）：公网（internet）。如果是 VPC 类型的只需要填写 intranet，通过 EIP 实现公网访问。
- 授权策略（Policy）：允许（accept）。
- 规则方向（NicType）：入网。
- 协议类型（IpProtocol）：TCP（tcp）。
- 端口范围（PortRange）：80/80。
- 授权对象（SourceCidrIp）：0.0.0.0/0。
- 优先级（Priority）：1。

说明：

上面的建议仅对公网有效。内网请求不建议使用 CIDR 网段，请参考 经典网络的内网安全组规则不要使用 *CIDR* 或者 *IP* 授权。

禁止一个入网请求规则

禁止一条规则时，您只需要配置一条拒绝策略，并设置较低的优先级即可。这样，当有需要时，您可以配置其它高优先级的规则覆盖这条规则。例如，您可以采用以下设置拒绝 6379 端口被访问。

- 网卡类型（NicType）：内网（intranet）。
- 授权策略（Policy）：拒绝（drop）。
- 规则方向（NicType）：入网。
- 协议类型（IpProtocol）：TCP（tcp）。

- 端口范围（PortRange）：6379/6379。
- 授权对象（SourceCidrIp）：0.0.0.0/0。
- 优先级（Priority)：100。

经典网络的内网安全组规则不要使用 CIDR 或者 IP 授权

对于经典网络的 ECS 实例，阿里云默认不开启任何内网的入规则。内网的授权一定要谨慎。

说明：

为了安全考虑，不建议开启任何基于 CIDR 网段的授权。

对于弹性计算来说，内网的 IP 经常变化，另外，这个 IP 的网段是没有规律的，所以，对于经典网络的内网，建议您通过安全组授权内网的访问。

例如，您在安全组 sg-redis 上构建了一个 redis 的集群，为了只允许特定的机器（如 sg-web）访问这个 redis 的服务器编组，您不需要配置任何 CIDR，只需要添加一条入规则：指定相关的安全组 ID 即可。

- 网卡类型（NicType）：内网（intranet）。
- 授权策略（Policy）：允许（accept）。
- 规则方向（NicType）：入网。
- 协议类型（IpProtocol）：TCP（tcp）。
- 端口范围（PortRange）：6379/6379。
- 授权对象（SourceGroupId）：sg-web。
- 优先级（Priority）：1。

对于 VPC 类型的实例，如果您已经通过多个 VSwitch 规划好自己的 IP 范围，您可以使用 CIDR 设置作为安全组入规则；但是，如果您的 VPC 网段不够清晰，建议您优先考虑使用安全组作为入规则。

将需要互相通信的 ECS 实例加入同一个安全组

一个 ECS 实例最多可以加入 5 个安全组，而同一安全组内的 ECS 实例之间是网络互通的。如果您在规划时已经有多个安全组，而且，直接设置多个安全规则过于复杂的话，您可以新建一个安全组，然后将需要内网通讯的 ECS 实例加入这个新的安全组。

安全组是区分网络类型的，一个经典网络类型的 ECS 实例只能加入经典网络的安全组；一个 VPC 类型的 ECS 实例只能加入本 VPC 的安全组。

这里也不建议您将所有的 ECS 实例都加入一个安全组，这将会使得您的安全组规则设置变成梦魇。对于一个中大型应用来说，每个服务器编组的角色不同，合理地规划每个服务器的入方向请求和出方向请求是非常有必要的。

在控制台上，您可以根据文档 加入安全组 的描述将一个实例加入安全组。

如果您对阿里云的 OpenAPI 非常熟悉，您可以参考 使用 OpenAPI 弹性管理 ECS 实例，通过 OpenAPI 进行批量操作。对应的 Python 片段如下。

```
def join_sg(sg_id, instance_id):
    request = JoinSecurityGroupRequest()
    request.set_InstanceId(instance_id)
    request.set_SecurityGroupId(sg_id)
    response = _send_request(request)
    return response
# 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)
```

将 ECS 实例移除安全组

如果 ECS 实例加入不合适的安全组，将会暴露或者 Block 您的服务，这时您可以选择将 ECS 实例从这个安全组中移除。但是在移除安全组之前必须保证您的 ECS 实例已经加入其它安全组。

说明：

将 ECS 实例从安全组移出，将会导致这个 ECS 实例和当前安全组内的网络不通，建议您在移出之前做好充分的测试。

对应的 Python 片段如下。

```
def leave_sg(sg_id, instance_id):
    request = LeaveSecurityGroupRequest()
    request.set_InstanceId(instance_id)
    request.set_SecurityGroupId(sg_id)
    response = _send_request(request)
    return response
# 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)
```

定义合理的安全组名称和标签

合理的安全组名称和描述有助于您快速识别当前复杂的规则组合。您可以通过修改名称和描述来帮助自己识别安全组。

您也可以通过为安全组设置标签分组管理自己的安全组。您可以在控制台直接 设置标签，也通过 API 设置标签。

删除不需要的安全组

安全组中的安全规则类似于一条条白名单和黑名单。所以，请不要保留不需要的安全组，以免因为错误加入某个 ECS 实例而造成不必要的麻烦。

1.3 ECS安全组实践（三）

在安全组的使用过程中，通常会将所有的云服务器放置在同一个安全组中，从而可以减少初期配置的工作量。但从长远来看，业务系统网络的交互将变得复杂和不可控。在执行安全组变更时，您将无法明确添加和删除规则的影响范围。

合理规划和区分不同的安全组将使得您的系统更加便于调整，梳理应用提供的服务并对不同应用进行分层。这里推荐您对不同的业务规划不同的安全组，并设置不同的安全组规则。

区分不同的安全组

- **公网服务的云服务器和内网服务器尽量属于不同的安全组**

  是否对外提供公网服务，包括主动暴露某些端口对外访问（例如 80、443 等），被动地提供（例如云服务器具有公网 IP、EIP、NAT 端口转发规则等）端口转发规则，都会导致自己的应用可能被公网访问到。

  2 种场景的云服务器所属的安全组规则要采用最严格的规则，建议拒绝优先，默认情况下应当关闭所有的端口和协议，仅仅暴露对外提供需要服务的端口，例如 80、443。由于仅对属于对外公网访问的服务器编组，调整安全组规则时也比较容易控制。

  对于对外提供服务器编组的职责应该比较明晰和简单，避免在同样的服务器上对外提供其它的服务。例如 MySQL、Redis 等，建议将这些服务安装在没有公网访问权限的云服务器上，然后通过安全组的组组授权来访问。

  如果当前有公网云服务器已经和其它的应用在同一个安全组 SG_CURRENT。您可以通过下面的方法来进行变更。

1. 梳理当前提供的公网服务暴露的端口和协议，例如 80、443。
2. 新创建一个安全组，例如 SG_WEB， 然后添加相应的端口和规则。

   说明：

   授权策略：允许，协议类型：ALL， 端口: 80/80，授权对象: 0.0.0.0/0， 授权策略:允许，协议类型:ALL，端口: 443/443 授权对象: 0.0.0.0/0。
3. 选择安全组 SG_CURRENT， 然后添加一条安全组规则，组组授权，允许 SG_WEB 中的资源访问SG_CURRENT。

   说明：

   授权策略：允许，协议类型：ALL，端口：-1/-1，授权对象：SG_WEB，优先级：按照实际情况自定义[1-100]。
4. 将一台需要切换安全组的实例 ECS_WEB_1 添加到新的安全组中。
   a. 在 ECS 控制台中，选择 安全组管理。
   b. 选择 **SG_WEB** > 管理实例 > 添加实例，选择实例 ECS_WEB_1 加入到新的安全组 SG_WEB 中，确认 ECS_WEB_1 实例的流量和网络工作正常。
5. 将 ECS_WEB_1 从原来的安全组中移出。
   a. 在 ECS 控制台中，选择 安全组管理。
   b. 选择 **SG_WEB** > 管理实例 > 添加实例，选择 ECS_WEB_1 ，从 SG_CURRENT 移除，测试网络连通性，确认流量和网络工作正常。
   c. 如果工作不正常，将 ECS_WEB_1 仍然加回到安全组 SG_CURRENT 中，检查设置的 SG_WEB 暴露的端口是否符合预期，然后继续变更。
6. 执行其它的服务器安全组变更。

- **不同的应用使用不同的安全组**

  在生产环境中，不同的操作系统大多情况下不会属于同一个应用分组来提供负载均衡服务。提供不同的服务意味着需要暴露的端口和拒绝的端口是不同的，建议不同的操作系统尽量归属于不同的安全组。

  例如，对于 Linux 操作系统，可能需要暴露 TCP（22）端口来实现 SSH，对 Windows 可能需要开通 TCP(3389) 远程桌面连接。

除了不同的操作系统归属不同的安全组，即便同一个镜像类型，提供不同的服务，如果之间不需要通过内网进行访问的话，最好也划归不同的安全组。这样方便解耦，并对未来的安全组规则进行变更，做到职责单一。

在规划和新增应用时，除了考虑划分不同的虚拟交换机配置子网，也应该同时合理的规划安全组。使用网段+安全组约束自己作为服务提供者和消费者的边界。

具体的变更流程参见上面的操作步骤。

- **生产环境和测试环境使用不同的安全组**

  为了更好的做系统的隔离，在实际开发过程中，您可能会构建多套的测试环境和一套线上环境。为了更合理的做网络隔离，您需要对不同的环境配置使用不通的安全策略，避免因为测试环境的变更刷新到了线上影响线上的稳定性。

  通过创建不同的安全组，限制应用的访问域，避免生产环境和测试环境联通。同时也可以对不同的测试环境分配不同的安全组，避免多套测试环境之间互相干扰，提升开发效率。

仅对需要公网访问子网或者云服务器分配公网 IP

不论是经典网络还是专有网络 (VPC) 中，合理的分配公网 IP 可以让系统更加方便地进行公网管理，同时减少系统受攻击的风险。在专有网络的场景下，创建虚拟交换机时，建议您尽量将需要公网访问的服务区的 IP 区间放在固定的几个交换机(子网 CIDR)中，方便审计和区分，避免不小心暴露公网访问。

在分布式应用中，大多数应用都有不同的分层和分组，对于不提供公网访问的云服务器尽量不提供公网IP，如果是有多台服务器提供公网访问，建议您配置公网流量分发的负载均衡服务来公网服务，提升系统的可用性，避免单点。

对于不需要公网访问的云服务器尽量不要分配公网 IP。专有网络中当您的云服务器需要访问公网的时候，优先建议您使用 *NAT 网关*，用于为 VPC 内无公网 IP 的 ECS 实例提供访问互联网的代理服务，您只需要配置相应的 SNAT 规则即可为具体的 CIDR 网段或者子网提供公网访问能力，具体配置参见 *SNAT*。避免因为只需要访问公网的能力而在分配了公网 IP(EIP) 之后也向公网暴露了服务。

最小原则

安全组应该是白名单性质的，所以需尽量开放和暴露最少的端口，同时尽可能少地分配公网 IP。若想访问线上机器进行任务日志或错误排查的时候直接分配公网 IP 或者挂载 EIP 虽然简便，但是毕竟会将整个机器暴露在公网之上，更安全的策略是建议通过跳板机来管理。

使用跳板机

跳板机由于其自身的权限巨大，除了通过工具做好审计记录。在专有网络中，建议将跳板机分配在专有的虚拟交换机之中，对其提供相应的 EIP 或者 NAT 端口转发表。

首先创建专有的安全组 SG_BRIDGE，例如开放相应的端口，例如 Linux TCP(22) 或者 Windows RDP(3389)。为了限制安全组的入网规则，可以限制可以登录的授权对象为企业的公网出口范围，减少被登录和扫描的概率。

然后将作为跳板机的云服务器加入到该安全组中。为了让该机器能访问相应的云服务器，可以配置相应的组授权。例如在 SG_CURRENT 添加一条规则允许 SG_BRIDGE 访问某些端口和协议。

使用跳板机 SSH 时，建议您优先使用 *SSH* 密钥对 而不是密码登录。

总之，合理的安全组规划使您在扩容应用时更加游刃有余，同时让您的系统更加安全。

1.4 ECS数据安全最佳实践

本文档从使用云服务器ECS的角度出发，结合相关产品和运维架构经验，介绍如何打造云端的数据安全。

适用对象

本文档适用于刚开始接触阿里云的个人或者中小企业用户。

主要内容

- 定期备份数据
- 合理设计安全域
- 安全组规则设置
- 登录口令设置
- 服务器端口安全
- 系统漏洞防护
- 应用漏洞防护
- 安全情报收集

定期备份数据

数据备份是容灾的基础，目的是降低因系统故障、操作失误、以及安全问题而导致数据丢失的风险。云服务器ECS自带有快照备份的功能，合理运用ECS快照功能即可满足大部分用户数据备份的需求。建议用户根据自身的业务情况，制定适合自己的备份策略，您可以选择 手动创建快照，或

者 创建自动快照策略，并 将此策略应用到指定磁盘。推荐每日做一次自动快照，每次快照最少保存7天。养成良好的备份习惯，在故障发生时，有利于迅速恢复重要数据，减少损失。

合理设计安全域

基于SDN（Software Defined Network）技术研发的VPC专有网络，可以供用户构建自定义专属网络，隔离企业内部不同安全级别的服务器，避免互通网络环境下一台服务器感染后影响到其它应用服务器。

建议用户 创建专有网络，选择自有 IP 地址范围、划分网段、配置路由表和网关等。用户可以将比较重要的数据存储在一个跟互联网网络完全隔离的内网环境，日常运维可以用弹性IP（EIP）或者跳板机的方式，对数据进行管理。

安全组规则设置

安全组是重要的网络安全隔离手段，用于设置单台或多台云服务器的网络访问控制。用户通过安全组设置实例级别的防火墙策略，可以在网络层过滤服务器的主动/被动访问行为，限定服务器对外/对内的的端口访问，授权访问地址，从而减少攻击面，保护服务器的安全。

例如Linux系统默认远程管理端口22，不建议向外网直接开放，可以通过 设置安全组配置ECS公网访问控制，只授权本地固定IP对服务器进行访问。您可以查看其它 应用案例，加深对安全组的熟悉程度。对访问控制有更高要求的用户或者也可以使第用三方VPN产品，对登录行为进行数据加密，更多软件尽在 云市场。

登录口令设置

弱口令一直是数据泄露的一个大症结，因为弱口令是最容易出现的也是最容易被利用的漏洞之一。服务器的口令建议至少8位以上，从字符种类上增加口令复杂度，如包含大小写字母、数字和特殊字符等，并且要不定时更新口令，养成良好的安全运维习惯。

服务器端口安全

服务器只要给互联网提供服务，就会将对应的服务端口暴露在互联网，从安全管理的角度来说，开启的服务端口越多，就越不安全。建议只对外开放提供服务的必要端口，并修改常见端口为高端口（30000以后），再对提供服务的端口做访问控制。

例如数据库服务尽量在内网环境使用，避免暴露在公网；如果必须要在公网访问，则需要修改默认连接端口3306为高端口，并根据业务授权可访问客户端地址。

系统漏洞防护

系统漏洞问题这种长期都存在的安全风险，可以通过系统补丁程序，或者 安骑士补丁管理 来解决。Windows系统的补丁更新要一直开启，Linux系统要设置定期任务执行`yum update -y`来更新系统软件包及内核。

云盾旗下的 安骑士产品 时还能识别防御非法破解密码的行为，避免被黑客多次猜解密码而入侵，批量维护服务器安全。安骑士同时还提供针对服务器应用软件不安全的配置检测和修复方案，帮助用户成功修复弱点，提高服务器安全强度。强烈推荐用户使用。

应用漏洞防护

应用漏洞是指针对Web应用、缓存、数据库、存储等服务，通过利用渗透攻击而非法获取数据的一种安全缺陷。常见应用漏洞包括：SQL注入、XSS跨站、Webshell上传、后门隔离保护、命令注入、非法HTTP协议请求、常见Web服务器漏洞攻击、核心文件非授权访问、路径穿越等。这种漏洞不同于系统漏洞，修复存在很大难度，如果程序在设计应用之初，不能对这些应用安全基线面面俱到，服务器安全的堡垒，就往往在这最后一公里被攻破。所以我们推荐通过接入 *Web*应用防火墙（Web Application Firewall, 简称 WAF）这种专业的防护工具，来轻松应对各类Web应用攻击，确保网站的Web安全与可用性。

安全情报收集

在当今暗流涌动的互联网安全领域，安全工程师和黑客比拼的就是时间，云盾态势感知 可以理解为一种基于大数据的安全服务，即在大规模云计算环境中，对能够引发网络安全态势发生变化的要素进行全面、快速和准确地捕获和分析。然后把客户当前遇到的安全威胁与过去的威胁进行关联回溯和大数据分析，最终产出未来可能发生的威胁安全的风险事件，并提供一个体系化的安全解决方案。

所以，技术人员除了在做好日常安全运维的同时，还要尽可能掌握全面的信息，提升预警能力，在发现安全问题的时候可以及时进行修复和处理，才能真正保证云服务器ECS的数据安全闭环。

1.5 如何提高ECS实例的安全性

云服务器 ECS 实例是一个虚拟的计算环境，包含了 CPU、内存、操作系统、磁盘、带宽等最基础的服务器组件，是 ECS 提供给每个用户的操作实体。

我们基本可以理解为一个实例就等同于一台虚拟机，那么我们在本地维护的虚拟机一般会做虚拟机实例级别的安全防护，以防止虚拟机被攻击和入侵等。同样的，云上的ECS实例也需要做安全性防护。

ECS实例放置在云上，除了置身于阿里云自身的安全平台外，用户也需要根据实际的需求进一步定制化安全，所以说ECS的安全是阿里云和用户共同构建的。如果ECS实例没有安全的防护，可能会带来不少不良的影响，比如遭受到DDoS而导致业务中断，比如受到Web入侵而导致网页被篡改、挂马，比如被注入而导致信息和数据泄漏等，影响ECS的使用和无法正常提供服务。

一般可以通过设置安全组、AntiDDoS、态势感知、安装安骑士、接入Web应用防火墙等方式提高ECS实例的安全性。下面就从实例层面分别讲解一下如何提高ECS实例的安全性。

安全组是一个逻辑上的分组，这个分组是由同一个地域（Region）内具有相同安全保护需求并相互信任的实例组成。每个实例至少属于一个安全组，在创建的时候就需要指定。同一安全组内的实例之间网络互通，不同安全组的实例之间默认内网不通。可以授权两个安全组之间互访。

设置安全组

- **设置安全组的好处**

  安全组是一种虚拟防火墙，具备状态检测包过滤功能。安全组用于设置单台或多台云服务器的网络访问控制，它是重要的网络安全隔离手段，用于在云端划分安全域。安全组规则可以允许或者禁止与安全组相关联的云服务器 ECS 实例的公网和内网的入出方向的访问。

  如果没有很好地设置安全组或者安全组规则过于开放，则降低了访问的限制级别，在一定程度上为攻击者敞开了大门。

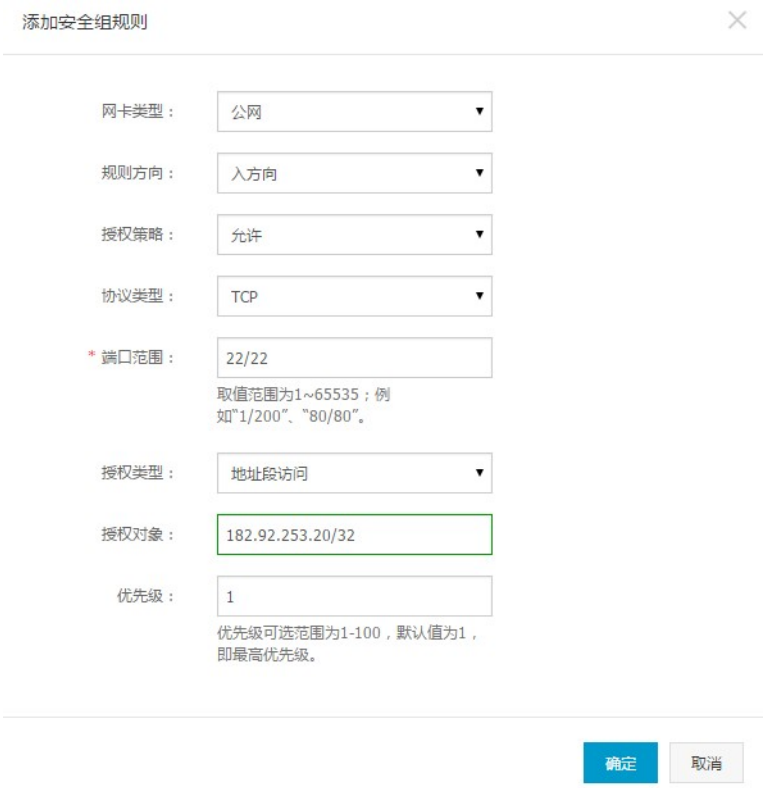
- **操作步骤**

  1. 登录 云服务器管理控制台。
  2. 单击左侧导航中的 安全组。
  3. 选择地域。
  4. 单击添加安全组规则。
  5. 在弹出的对话框中，分别设置网络类型、规则方向、授权策略、协议类型、端口范围、授权类型、授权对象和优先级。
  6. 点击 确定，成功为该安全组授权一条安全组规则 。

  下面结合一个案例来阐述一下，比如只允许特定IP远程登录到实例。

  通过配置安全组规则可以设置只让特定 IP 远程登录到实例。只需要在公网入方向配置规则就可以了，以 Linux 服务器为例，设置只让特定 IP 访问 22 端口。

1. 添加一条公网入方向安全组规则，允许访问，协议类型选择 TCP，端口写 22/22，授权类型为地址段访问，授权对象填写允许远程连接的 IP 地址段，格式为 x.x.x.x/xx，即 IP地址/子网掩码，本例中的地址段为 182.92.253.20/32。优先级为 1。



2. 再添加一条规则，拒绝访问，协议类型选择 TCP，端口写 22/22，授权类型为地址段访问，授权对象写所有 0.0.0.0/0，优先级为 2。

最终的效果如下：

来自 IP 182.92.253.20 访问 22 端口优先执行优先级为 1 的规则允许。

来自其他 IP 访问 22 端口优先执行优先级为 2 的规则拒绝了。

AntiDDoS

阿里云云盾可以防护SYN Flood，UDP Flood，ACK Flood，ICMP Flood，DNS Flood，CC攻击等3到7层DDoS的攻击。DDoS基础防护免费为阿里云用户提供最高5G的默认DDoS防护能力。

阿里云在此基础上，推出了安全信誉防护联盟计划，将基于安全信誉分进一步提升DDoS防护能力，用户最高可获得100G以上的免费DDoS防护资源。

- 为什么需要**AntiDDoS**

DDoS（Distributed Denial of Service）即分布式拒绝服务。攻击指借助于客户/服务器技术，将多个计算机联合起来作为攻击平台，对一个或多个目标发动DDoS攻击，从而成倍地提高拒绝服务攻击的威力，影响业务和应用正常对用户提供服务。

使用AntiDDoS，无需采购昂贵清洗设备，可以在受到DDoS攻击不会影响访问速度，带宽充足不会被其他用户连带影响，保证业务可用和稳定。

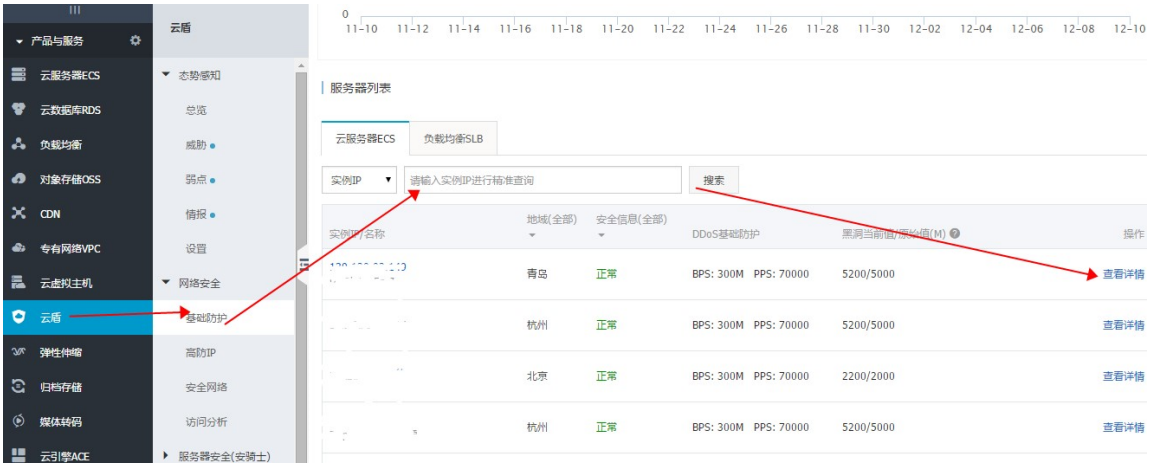
- 操作步骤

1. 进入阿里云官网，登录到 管理控制台。
2. 输入用户名密码。
3. 通过云盾 > **DDOS防护** > 基础防护，查看基础防护配置。
4. 可以加入安全信誉防护联盟。勾选服务条款，点选加入安全信誉防护联盟加入联盟。如下图所示。

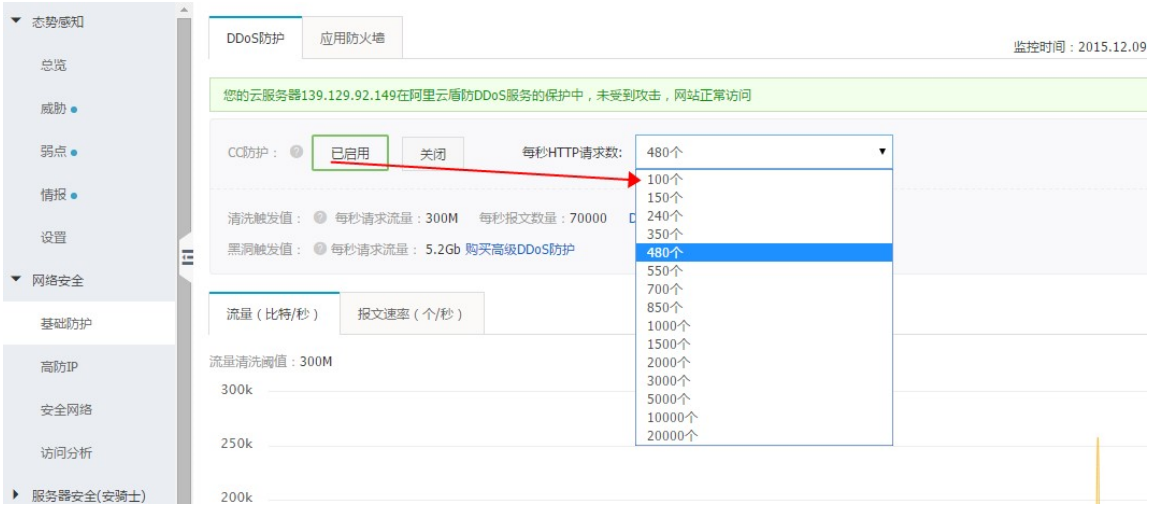
云盾DDoS基础版提供不大于5G的DDoS防护，在此基础上推出了安全信誉防护联盟计划，您可通过加入此联盟，在获得原默认防护能力基础上，会得到免费增量防护带宽机会。

加入联盟后，可查看自己的安全信誉分，并查看安全信誉组成，维护安全信誉，获得更大的防护能力。加盟成功后在基础防护界面显示如下信誉界面。

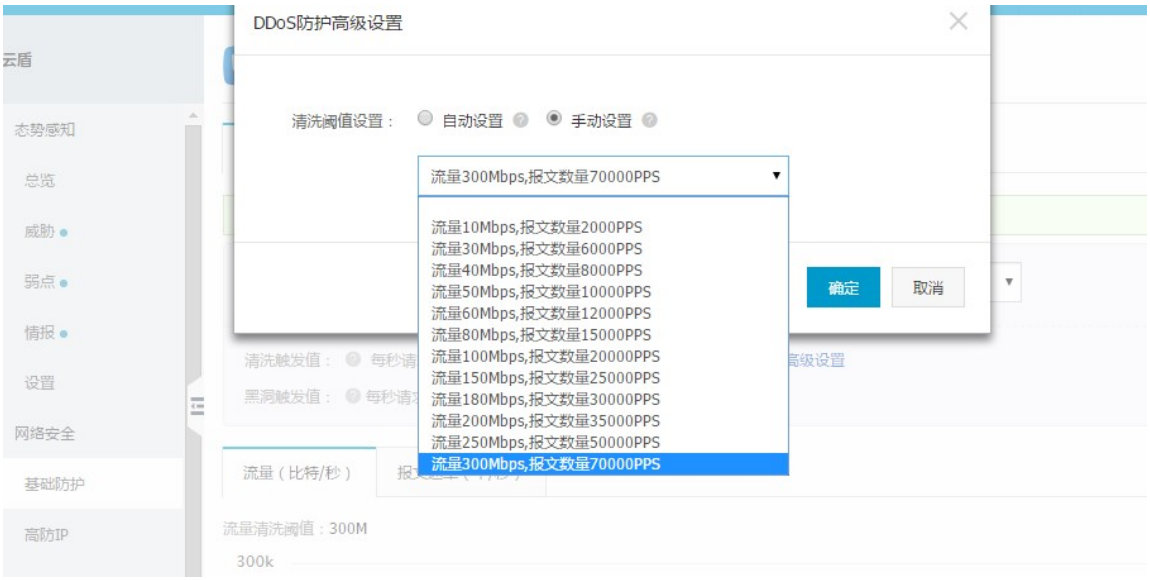
5. 在基础防护页面，点击对应ECS服务器的查看详情，如果服务器数量比较多，可以在云服务器ecs列表中通过实例IP和实例名称搜索服务器，再点击对应服务器的查看详情。



6. 进入页面后，可以在**CC防护**页面点击已启用开启CC防护，点击关闭则关闭CC防护功能，在每秒**HTTP**请求数可以对每秒http请求数设置清洗阈值，达到阈值后便会触发云盾的清洗。



7. 如果购买了高级DDoS防护，可以点击**DDoS防护高级设置**可以设置清洗阈值，选择**自动设置**后系统会根据云服务器的流量负载动态调整清洗阈值，选择**手动设置**可以手动对流量和报文数量的阈值进行设置，当超过此阈值后云盾便会开启流量清洗(建议如果网站在做推广或者活动时适当调大)。



态势感知

态势感知态势感知提供的是一项SAAS服务，即在大规模云计算环境中，对那些能够引发网络安全态势发生变化的要素进行全面、快速和准确地捕获和分析。然后，把客户当前遇到的安全威胁与过去的威胁进行关联回溯和大数据分析，最终产出未来可能产生的安全事件的威胁风险，并提供一个体系化的安全解决方案。

- **态势感知的优势**

对“渗透攻击”有所感知，以云计算数据平台支撑，因此具有强大的安全数据分析能力，对各种常见类型的攻击可以实时分析和展示。

- 操作步骤

1. 在 管理控制台的态势感知中点击免费开启服务，即可使用态势感知。

2. 通过紧急时间、威胁、弱点、情报、日志等方面，辅以直观的可视化的分析，让安全一目了然。

安装安骑士

服务器安全(安骑士)是云盾推出的一款服务器安全运维管理产品。通过安装在服务器上的轻量级Agent插件与云端防护中心的规则联动，实时感知和防御入侵事件，保障服务器的安全。

- 安装安骑士的好处

安骑士是很轻量的，服务器上运行的Agent插件，正常状态下只占用1%的CPU、10MB内存。安骑士可以自动识别服务器的Web目录，对服务器的Web目录进行后门文件扫描，支持通用Web软件漏洞扫描和Windows系统漏洞扫描，对服务器常见系统配置缺陷进行检测，包括可疑系统账户、弱口令、注册表等进行检测。

我们可以将安骑士理解为ECS实例上的防病毒软件，如果没有安骑士，相当于少了一个可靠的卫士，我们ECS实例的健康性水平也会相应降低。

- 操作步骤

1. 服务器安全(安骑士)Agent插件目前集成于安全镜像中，在购买ECS后，一般都已经默认安装，您可以进入安骑士控制台-配置中心，查看每台服务器的在线状态。

2. 若不在线，请按照如下方式下载并安装。

   a. 进入服务器安全(安骑士)控制台-设置-安装Agent页面，根据页面提示获取最新版本下载地址，以管理员权限在服务器上运行并安装。

   b. 对于非阿里云服务器，在安装过程中会提示输入验证Key，这个验证Key用于关联阿里云账号，通过阿里云账号在安骑士控制台使用相关功能，验证key会显示在安装页面中。

   c. 大约安装完成2分钟后在云盾·服务器安全(安骑士)控制台-配置中心里查看到在线数据，阿里云服务器将会从离线变成在线，非阿里云机器会新增在服务器列表中。

接入Web应用防火墙

云盾Web应用防火墙(Web Application Firewall, 简称 WAF)基于云安全大数据能力实现，通过防御SQL注入、XSS跨站脚本、常见Web服务器插件漏洞、木马上传、非授权核心资源访问等OWASP常见攻击，过滤海量恶意CC攻击，避免您的网站资产数据泄露，保障网站的安全与可用性。

- 接入Web应用防火墙的好处

  无需安装任何软、硬件，无需更改网站配置、代码，它可以轻松应对各类Web应用攻击，确保网站的Web安全与可用性，淘宝天猫都在用。除了具有强大Web防御能力，还可以指定网站的专属防护，背后是大数据的安全能力。适用于在金融、电商、o2o、互联网+、游戏、政府、保险、政府等各类网站的Web应用安全防护上。

如果缺少WAF，光靠前面提到的防护措施会存在短板，例如在面对如数据泄密、恶意CC、木马上传篡改网页等攻击的时候，就不能拿很好地防护了，可能会导致Web入侵。

- 操作步骤

1. 控制台配置。

a. 登录阿里云控制台，找到云盾 > Web应用防火墙 > 域名配置，点击添加域名按钮。

b. 弹出的对话框中输入相关信息：

c. 获取CNAME。配置好域名后，WAF会自动分配给当前域名一个CNAME，可点击域名信息来查看：

d. 上传HTTPS证书和私钥（仅针对HTTPS站点）。如果防护HTTPS站点，必须上传服务器的证书和私钥到WAF，否则访问HTTPS站点会有问题。勾选HTTPS后，会看到红色的“异常”字样，提示当前证书有问题，点击上传证书来上传：

e. 接入状态异常排查，刚添加完域名时，接入状态可能会提示异常。这是正常的，待修改DNS使用CNAME解析接入WAF后，或者是有正常流量经过WAF以后会变成正常的。

2. 放行回源IP段。

3. 本地验证。

a. 以前面步骤中添加的域名“www.aliyundemo.cn”为例，hosts文件应该添加如下内容，其中前面的IP地址为对应的WAFIP地址，WAF的IP可以通过ping提供的CNAME来获得。

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost

 58.255 www.aliyundemo.cn
```

b. 修改hosts文件后保存。然后本地ping一下被防护的域名，预期此时解析到的IP地址应该是刚才绑定的WAF IP地址。如果依然是源站地址，可尝试刷新本地的DNS缓存（Windows的cmd下可以使用ipconfig/flushdns命令）。

c. 确认hosts绑定已经生效（域名已经本地解析为WAF的IP）后，打开浏览器，输入该域名进行访问，如果WAF的配置正确，网站预期能够正常打开。

d. 尝试一下手动模拟一些简单的web攻击命令，如www.aliyundemo.cn/?alert(xss) 预期WAF能够弹出阻拦页面：

4. 通过DNS供应商或者其他域名解析系统，修改DNS解析。

阿里云给我们ECS实例的安全性提供了这么多的安全产品保驾护航，我们可以根据实际需要选择相应的产品，加强对系统和数据的防护，减少ECS实例接受到的侵害，使其稳定、持久地运行。

1.6 经典网络内网实例互通设置方法

安全组是实例级别防火墙，为保障实例安全，设置安全组规则时要遵循“最小授权”原则，下面介绍四种安全的内网实例互通设置方法。

方法 1. 使用单 IP 地址授权

- 适用场景：适用于小规模实例间内网互通场景。
- 优点：以IP地址方式授权，安全组规则清晰，容易理解。
- 缺点：内网互通实例数量较多时，会受到安全组规则条数 100 条的限制，另外后期维护工作量比较大。
- 设置方法：

1. 选择需要互通的实例，进入 本实例安全组。
2. 选择需要配置安全组，单击 配置规则。
3. 单击 内网入方向，并单击 添加安全组规则。
4. 按以下描述添加安全组规则：
   - 授权策略：允许。
   - 协议类型：根据实际需要选择协议类型。
   - 端口范围：根据您的实际需要设置端口范围，格式为“起始端口号/终止端口号”。
   - 授权类型：地址段访问。
   - 授权对象：输入想要内网互通的实例的内网 IP 地址，格式必须是 *a.b.c.d/32*。其中，子网掩码必须是 */32*。

方法 2. 加入同一安全组

- 适用场景：如果您的应用架构比较简单，可以为所有的实例选择相同的安全组，绑定同一安全组的实例之间不用设置特殊规则，默认网络互通。
- 优点：安全组规则清晰。
- 缺点：仅适用于简单的应用网络架构，网络架构调整时授权方法要随之进行修改。

方法 3. 绑定互通安全组

- 适用场景：为需要互通的实例增加绑定一个专门用于互通的安全组，适用于多层应用网络架构场景。
- 优点：操作简单，可以迅速建立实例间互通，可应用于复杂网络架构。
- 缺点：实例需绑定多个安全组，安全组规则阅读性较差。
- 设置方法：

  1. 新建一个安全组，命名为“互通安全组”，不需要给新建的安全组添加任何规则。
  2. 将需要互通的实例都添加绑定新建的“互通安全组”，利用同一安全组的实例之间默认互通的特性，达到内网实例互通的效果。

方法 4. 安全组互信授权

- 适用场景：如果您的网络架构比较复杂，各实例上部署的应用都有不同的业务角色，您就可以选择使用安全组互相授权方式。
- 优点：安全组规则结构清晰、阅读性强、可跨账户互通。
- 缺点：安全组规则配置工作量较大。
- 设置方法：

  1. 选择需要建立互信的实例，进入 **本实例安全组**。
  2. 选择需要配置安全组，单击 **配置规则**。
  3. 单击 **内网入方向**，并单击 **添加安全组规则**。
  4. 按以下描述添加安全组规则：

     - **授权策略**：允许。
     - **协议类型**：根据您的实际需要选择协议类型。
     - **端口范围**：根据实际需求设置。
     - **授权类型**：安全组访问。
     - **授权对象**：

- 如果您选择 本账号授权：按照您的组网要求，将有内网互通需求的对端实例的安全组 ID 填入 授权对象 即可。
- 如果您选择 跨账号授权：授权对象 应填入对端实例的安全组 ID，账号 ID 是对端账号 ID（可以在 账号管理 > 安全设置 里查到）。

建议

如果前期安全组授权过大，建议采用以下流程收紧授权范围。

图中的删除 0.0.0.0是指删除原来的允许 *0.0.0.0/0* 地址段的安全组规则。

如果安全组规则变更操作不当，可能会导致您的实例间通信受到影响，请在修改设置前备份您要操作的安全组规则，以便出现互通问题时及时恢复。

安全组映射了实例在整个应用架构中的角色，推荐按照应用架构规划防火墙规则。例如：常见的三层 Web 应用架构就可以规划三个安全组，将部署了相应应用或数据库的实例绑定对应的安全组：

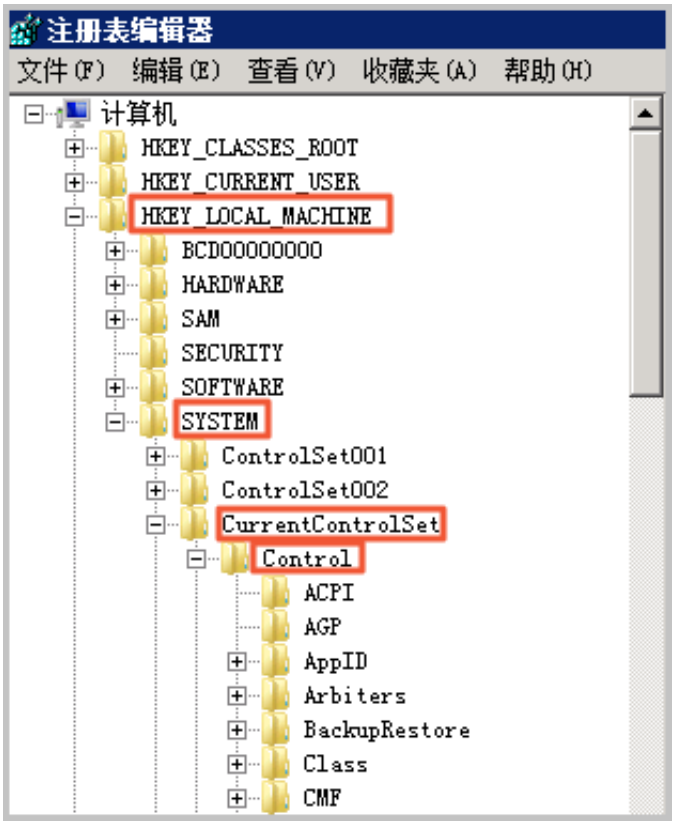
- Web 层安全组：开放 80 端口。
- APP 层安全组：开放 8080 端口。
- DB 层安全组：开放 3306 端口。

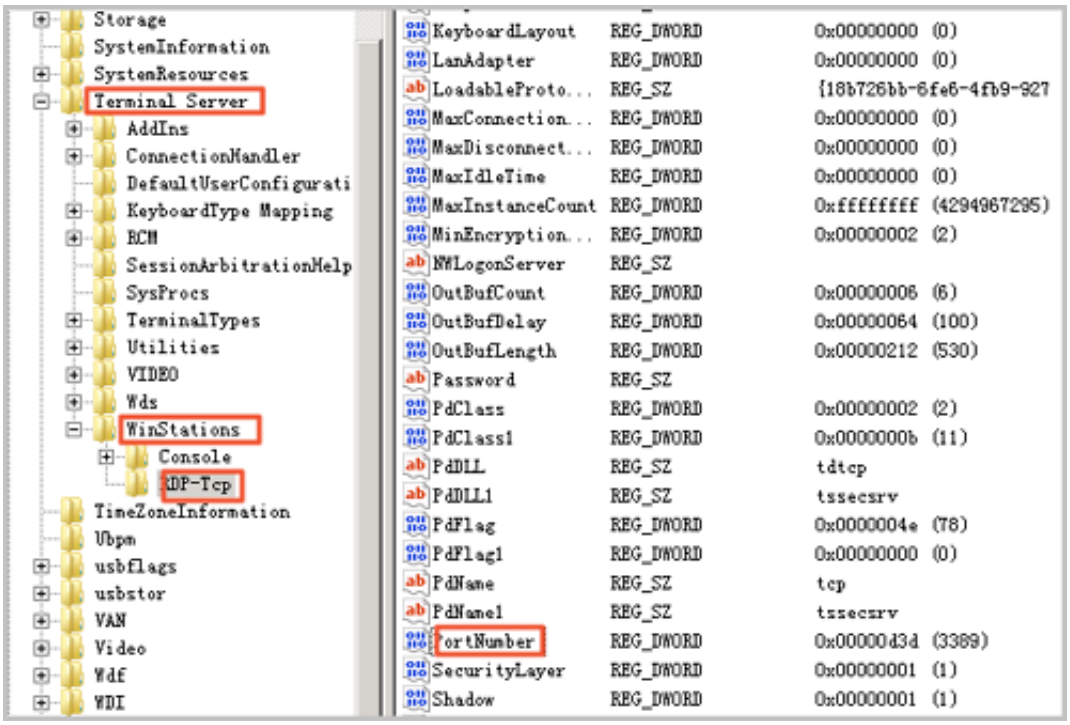
1.7 修改服务器默认远程端口

本节以 CentOS 6.8 为例介绍如何修改 Linux 服务器默认远程端口。

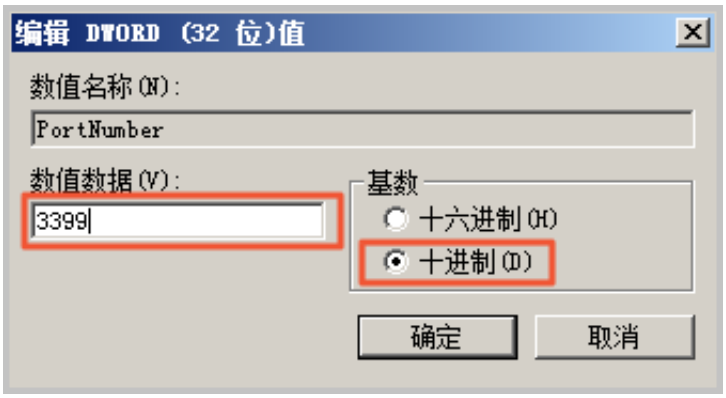
修改 Windows 服务器默认远程端口

1. 远程连接并登录到 Windows 实例。
2. 运行*regedit.exe*打开注册表编辑器。
3. 找到如下注册表子项：*HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp\PortNumber*





4. 在弹出的对话框中，选择十进制，在数值数据中输入新的远程端口号，在本例中即 3399。单击确定。



5. （可选）如果您开启了防火墙，需要将新的端口号添加到防火墙并设置允许连接。

   具体方法参见设置 *ECS* 实例远程连接防火墙。

6. 登录 *ECS*管理控制台，找到该实例，选择更多 > 重启。

7. 实例重新启动后，在实例的右侧单击管理，进入实例详情页面。选择本实例安全组。

8. 在安全组列表页面，找到相应的安全组，单击配置规则。
9. 在安全组规则页面，单击添加安全组规则。根据实际的使用场景来定义安全规则，允许新配置的远程端口进行连接。关于如何设置安全组参见添加安全组规则。

10. 以上步骤完成后，远程访问服务器，在远程地址后面添加新远程端口号即可连接实例。例如：192.168.1.2:3399。

> 说明：
> 调整 3389 端口后，使用 Mac 的远程桌面连接客户仅支持默认的 3389 端口。

修改 Linux 服务器默认远程端口

本节以 CentOS 6.8 为例介绍如何修改 Linux 服务器默认远程端口。

> 说明：
> 不要直接修改 22 端口，先添加需要的默认远程端口。之所以先设置成两个端口，测试成功后再关闭一个端口，是为了防止在修改配置文件及网络调试过程中，万一出现新端口无法连接的情况下，还能通过 22 端口进行登录调试。

1. 远程连接并登录到 Linux 实例。
2. 运行 **`vim /etc/ssh/sshd_config`** 命令。
3. 在键盘上按“I”键，进入编辑状态。添加新的远程服务端口，本节以 1022 端口为例。在*`Port 22`*下输入*`Port 1022`*。
4. 在键盘上按“Esc”，输入 **`:wq`** 退出编辑状态。
5. 执行以下命令重启实例，之后您可以通过 22 端口和 1022 端口 SSH 登录到 Linux 实例。

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

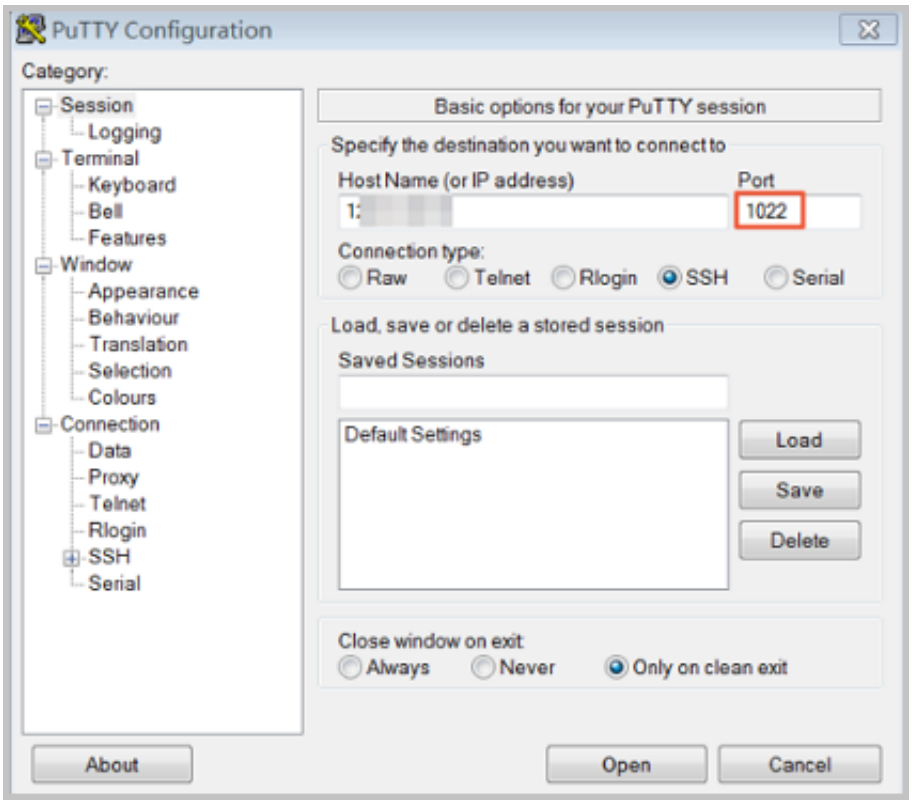
6. （可选）配置防火墙。使用 CentOS 7 以前的版本并开启默认防火墙 iptables 时，应注意 iptables 默认不拦截访问，如果您配置了 iptables 规则，需要执行**`iptables -A INPUT -p`**

tcp --dport 1022 -j ACCEPT配置防火墙。然后执行**service iptables restart** 重启防火墙。

> 说明：
> CentOS 7 以后版本默认安装 Firewalld。如果您已经启用 firewalld.service，需要放行 TCP 1022 端口：运行命令 firewall-cmd --add-port=1022/tcp --permanent。返回结果为 success 即表示已经放行 TCP 1022 端口。

7. 登录 *ECS*管理控制台，找到该实例，选择管理。
8. 进入实例详情页面。选择本实例安全组。

9. 在安全组列表页面，找到相应的安全组，单击配置规则。
10. 在安全组规则页面，单击添加安全组规则。根据实际的使用场景来定义安全规则，允许新配置的远程端口进行连接。关于如何设置安全组参见添加安全组规则。
11. 使用 SSH 工具连接新端口，来测试是否成功。登录时在 **Port** 一栏输入新修改的端口号，在本例中即 1022。



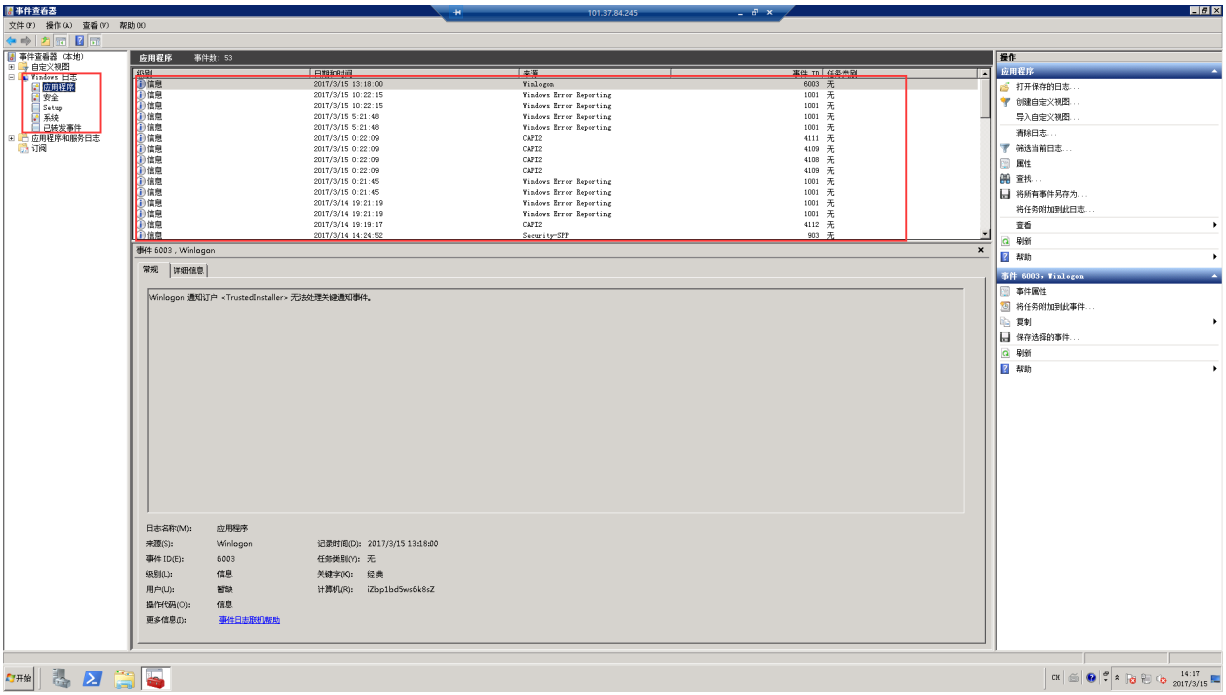
12. 使用 1022 端口连接成功后，再次运行**vim /etc/ssh/sshd_config**命令，将 Port 22 删除。
13. 运行 **/etc/init.d/sshd restart** 命令重启实例，服务器默认远程端口修改完成。再次登录时使用新端口号登录即可。

1.8 使用Windows实例的日志

日志记录了系统中硬件、软件和系统问题的信息，同时还监视着系统中发生的事件。当服务器被入侵或者系统（应用）出现问题时，管理员可以根据日志迅速定位问题的关键，再快速处理问题，从而极大地提高工作效率和服务器的安全性。Windows系统日志主要分为：系统日志、应用程序日志、安全日志以及应用程序和服务日志。本文以Windows Server 2008 R2为例，简单地介绍四种日志的使用和简要分析。

进入事件查看器

进入事件查看器：打开 运行 窗口，输入 **eventvwr**，打开 事件查看器。



之后，您可以在 **事件查看器** 里查看以下四种日志。

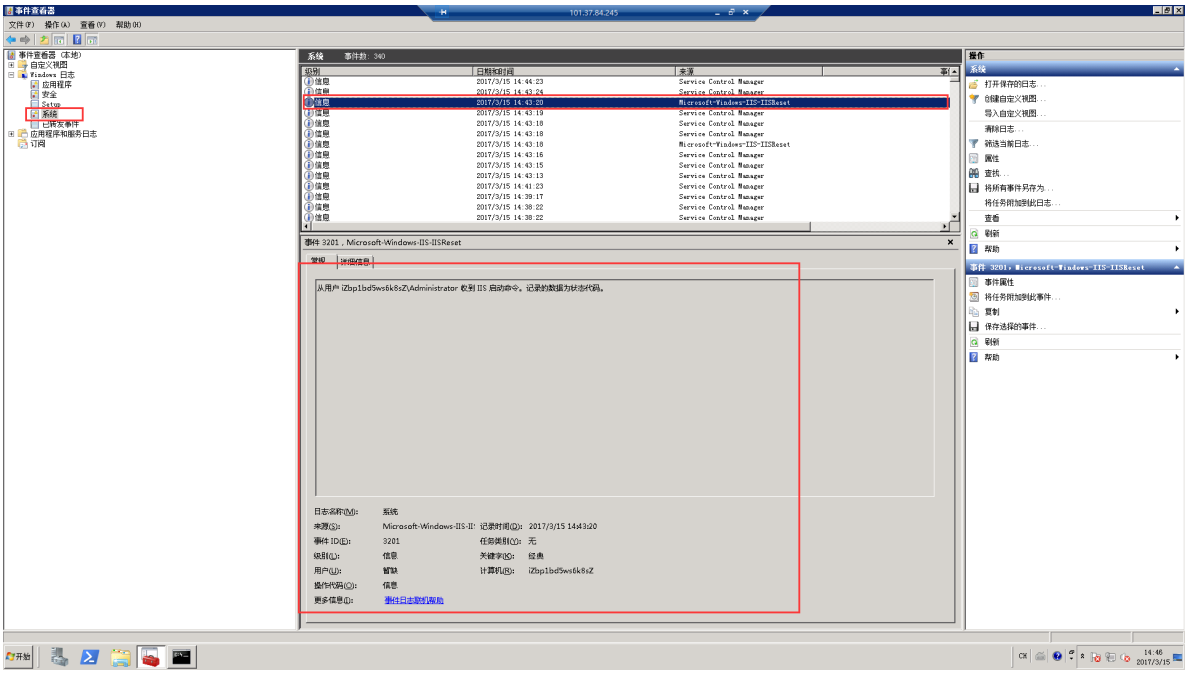
说明：

通过本文所述四种日志的查看方法找到的所有错误日志事件ID，您可以用于在微软知识库找到解决方法。

- 系统日志

  系统日志包含Windows系统组件记录的事件。例如，系统日志中会记录在启动过程中加载驱动程序或其他系统组件失败。

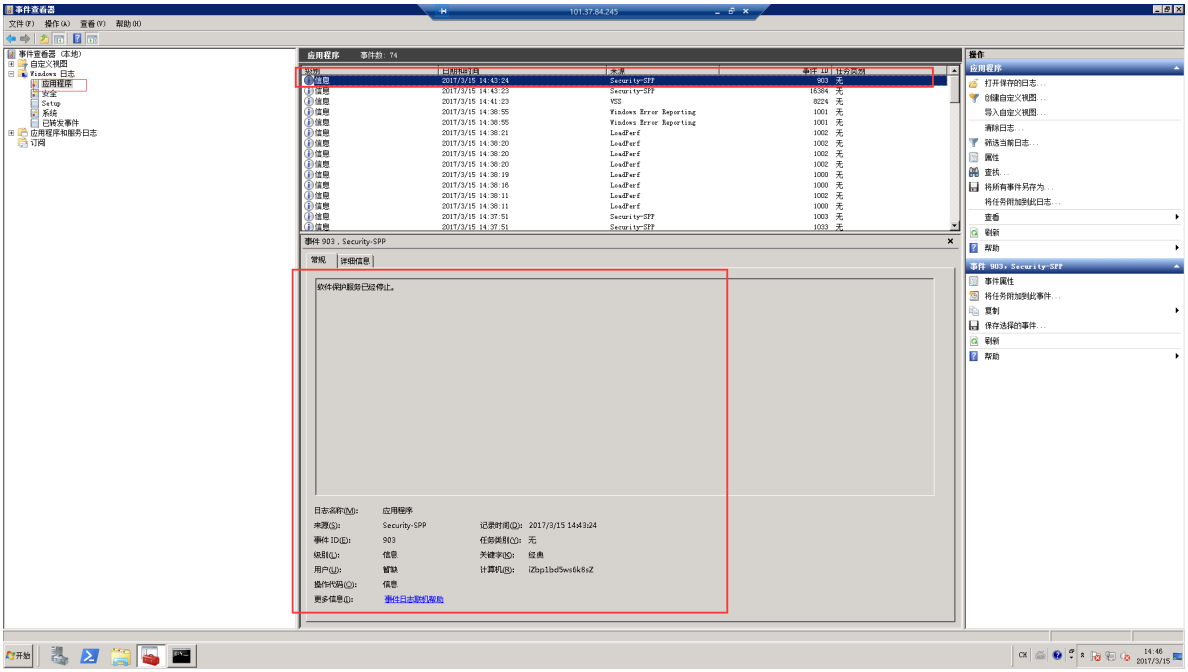
  系统组件所记录的事件类型由Windows预先确定。



- 应用程序日志

  应用程序日志包含由应用程序或程序记录的事件。例如，数据库程序可在应用程序日志中记录文件错误。

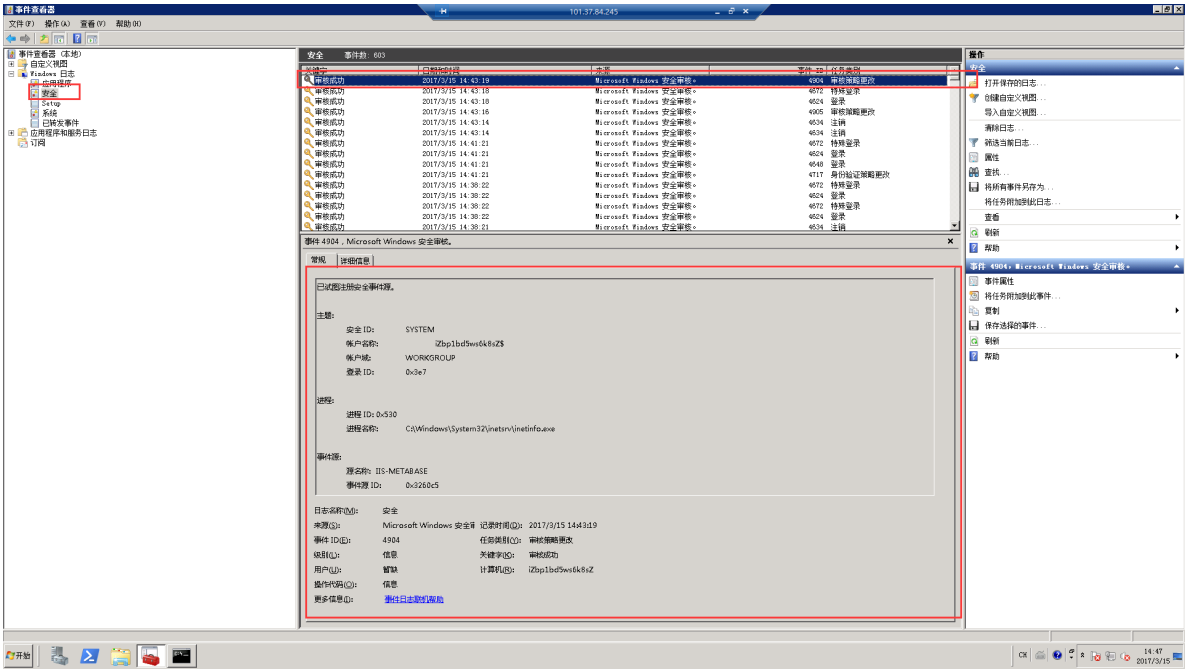
  程序开发人员决定记录哪些事件。

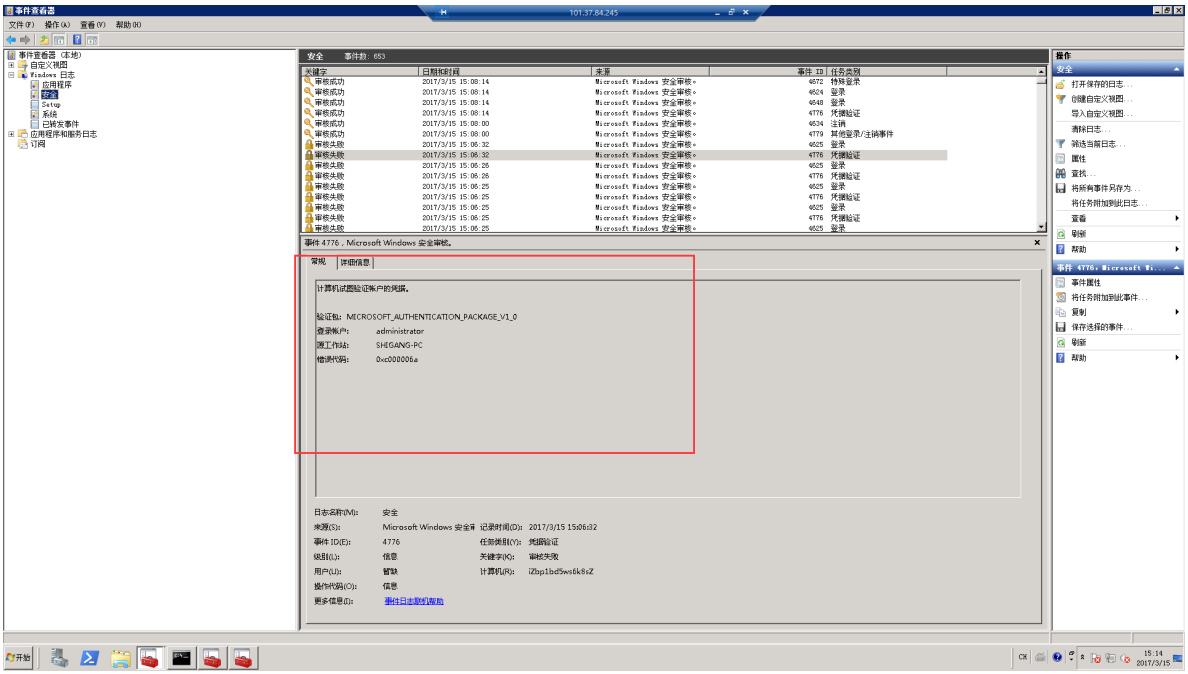


- 安全日志

  安全日志包含诸如有效和无效的登录尝试等事件，以及与资源使用相关的事件，如创建、打开或删除文件或其他对象。

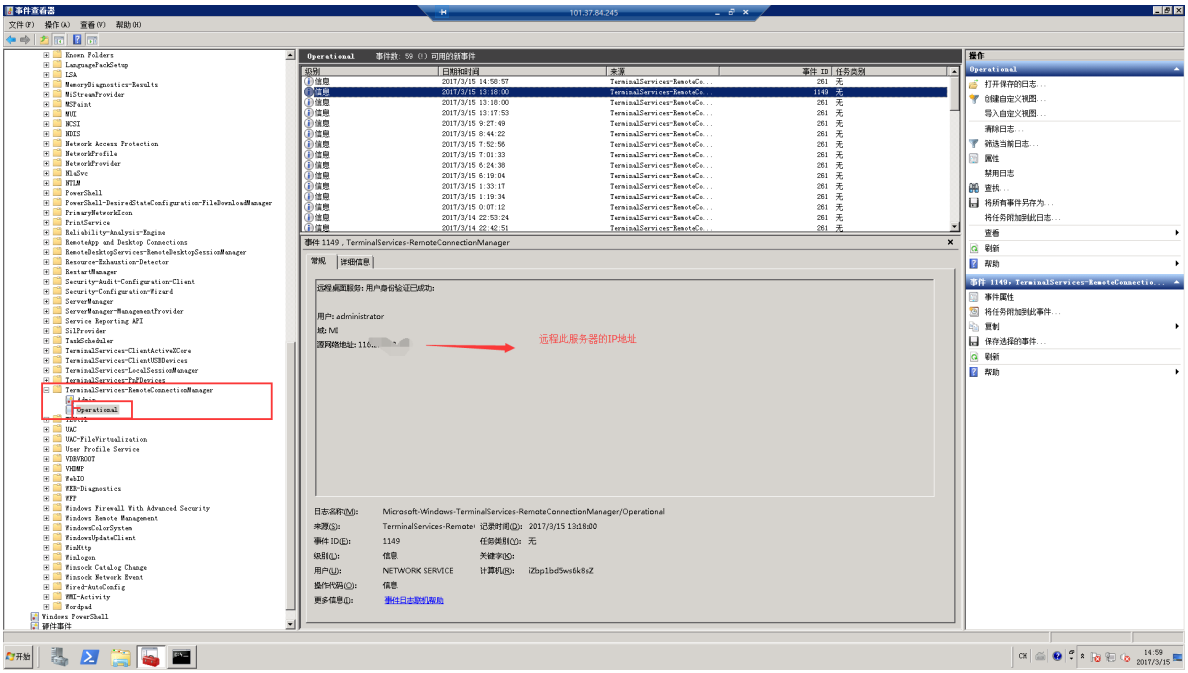
管理员可以指定在安全日志中记录什么事件。例如，如果已启用登录审核，则安全日志将记录对系统的登录尝试。





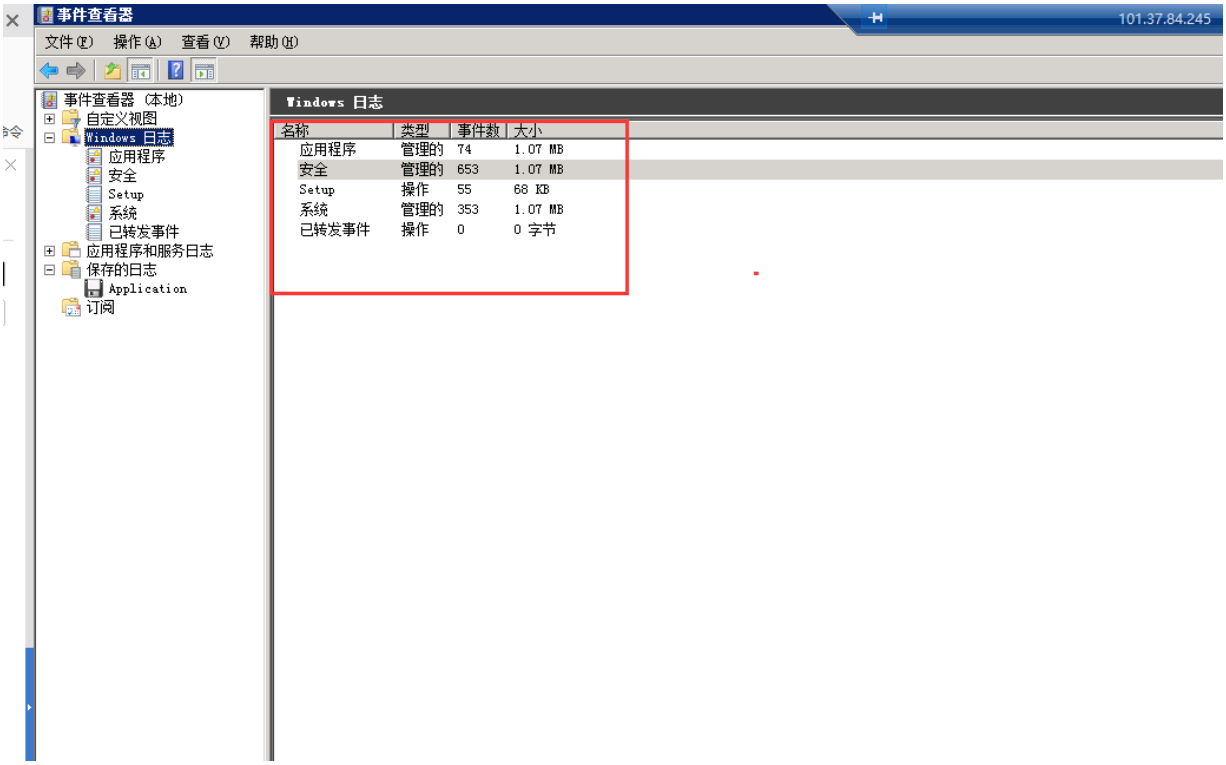
- 应用程序和服务日志

应用程序和服务日志是一种新类别的事件日志。这些日志存储来自单个应用程序或组件的事件，而非可能影响整个系统的事件。



修改日志路径并备份日志

日志默认保存在系统盘里面。日志最大值默认是20 MB，超过20 MB时会覆盖之前的事件。您可以根据自己的需求修改。



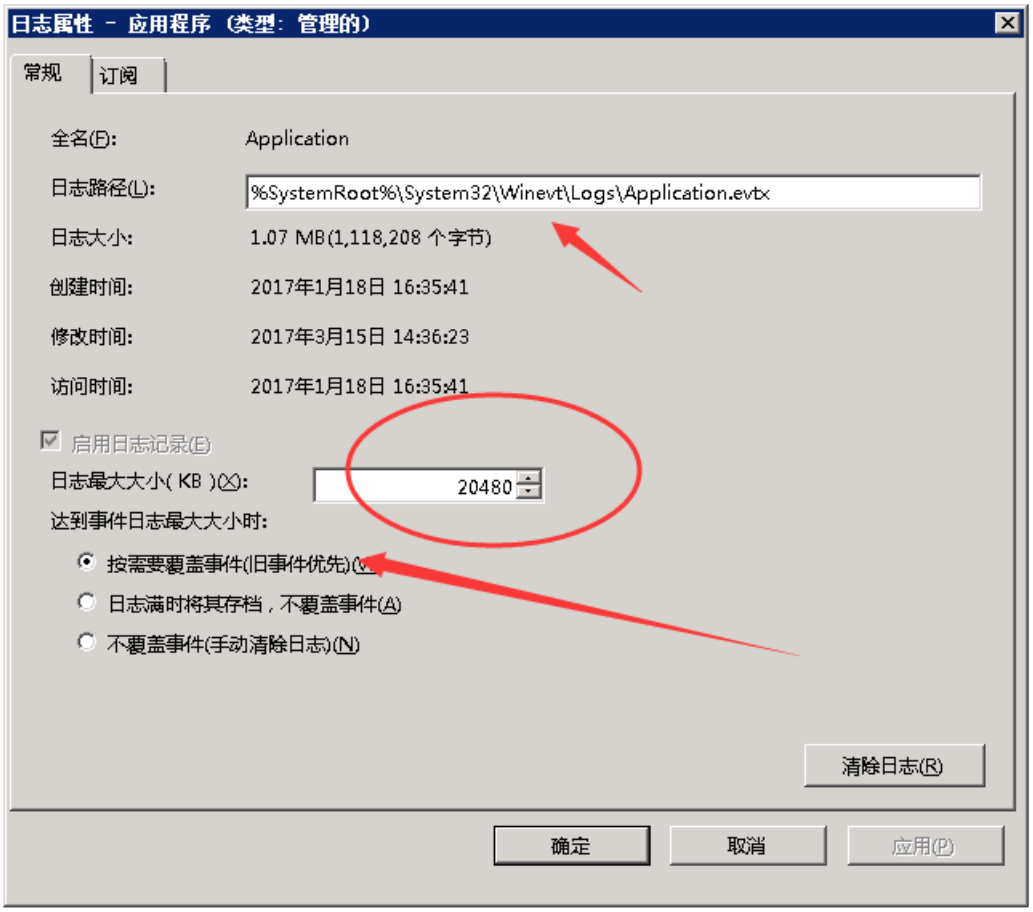
按以下步骤修改日志路径并备份日志。

1. 在 事件查看器 窗口，在左侧导航栏里，单击 **Windows** 日志。

2. 在右边列表中，选中一个日志目录，右键这一类日志，如截图所示的 应用程序。

3. 在 日志属性 窗口，按界面显示修改以下信息：

   - 日志路径。
   - 日志最大大小。
   - 达到事件日志最大大小时系统应采取的操作。



相关链接

云服务器 *ECS Windows* 安全审计日志简要说明

1.9 高级安全Windows防火墙概述以及最佳实践

本文简单介绍Windows防火墙的概念，给出使用场景并列出了常见的防火墙操作。

简介

在Windows NT6.0之后微软推出了高级安全Windows防火墙(简称WFAS)，高级安全Windows防火墙是分层安全模型的重要部分， 通过为计算机提供基于主机的双向网络通讯筛选， 高级安全Windows防火墙 阻止未授权的网络流量流向或流出本地计算机。高级安全 Windows 防火墙 还是用网络感知，以便可以将相应安全设置应用到计算机连接到的网络类型。Windows 防火墙和 Internet 协议保护 (sec) 配置设置集成到名为 高级安全 Windows 防火墙 的单个 Microsoft 管理控制台 (MMC)，高级安全Windows防火墙也成为网络隔离策略的重要部分。

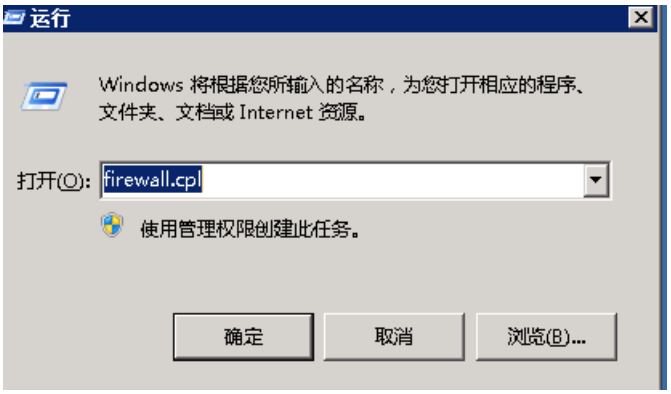
使用场景

作为一个运维人员，越来越多的用户反映服务器被恶意攻击，密码被暴力破解等等，其实大多数原因都是自己给那些“入侵者”留的“后门”导致的。入侵者通过扫描主机开放的端口，一旦发现可以利用的端口，就会进行下一步的入侵，例如Windows的远程端口（3389）和Linux的远程端口（22）。既然知道了问题的关键，那么我们也有相应的对策，我们可以通过修改默认的远程端口以及限制远程的访问来关闭所谓的“后门”。那么如何限制远程访问呢？接下来我们就以阿里云ECS实例Windows Server 2008 R2为例，来实现对远程桌面的限制。

操作步骤

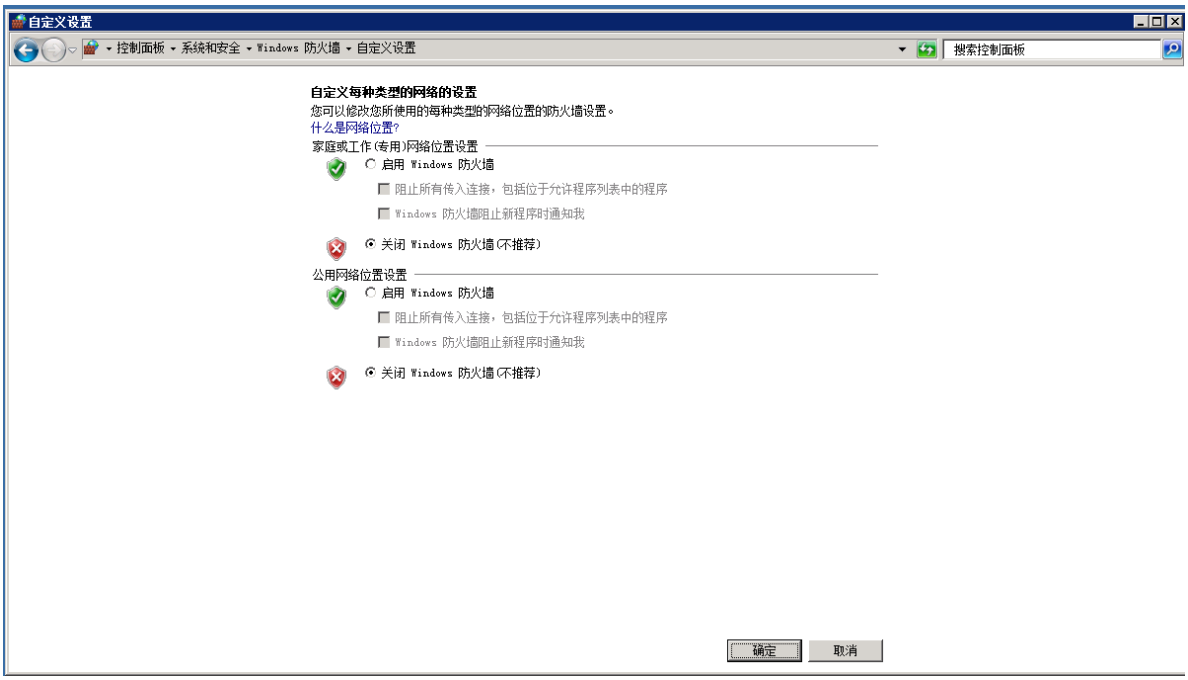
1. 查看防火墙状态

阿里云ECS实例Windows Server 2008 R2防火墙默认是关闭的，键盘输入Win+R打开运行输入*firewall.cpl* 回车来打开Windows防火墙控制台，见下图。



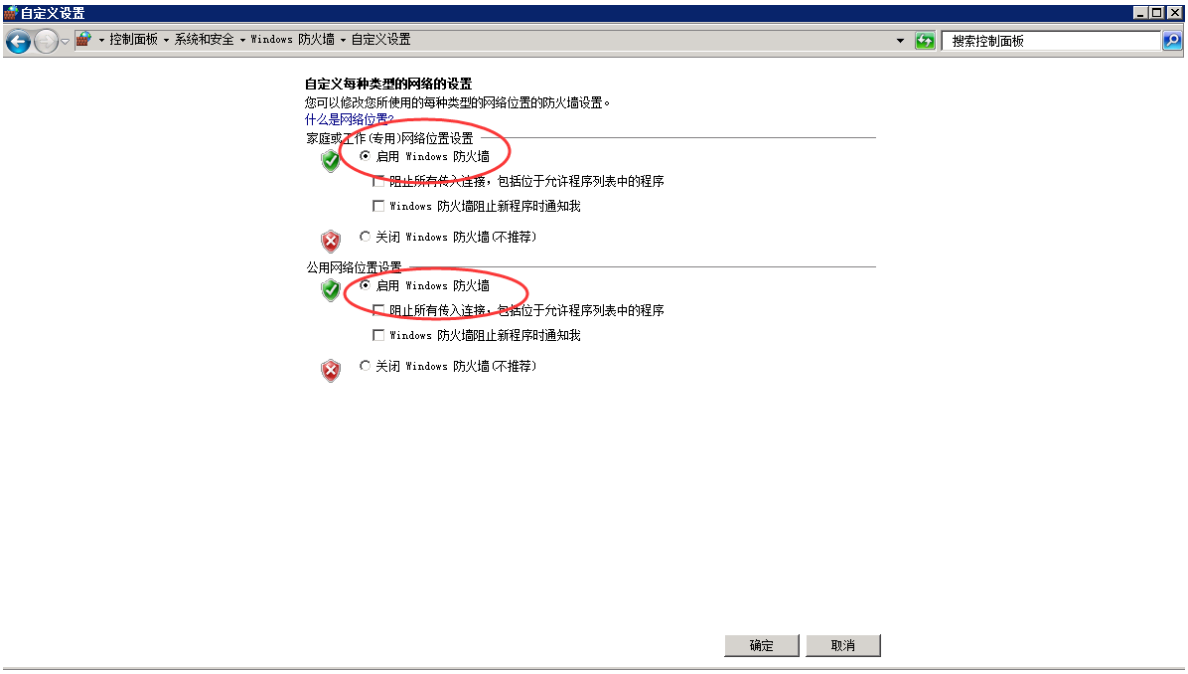
选择打开或关闭Windows防火墙。

如下图，我们看到防火墙是默认关闭的。



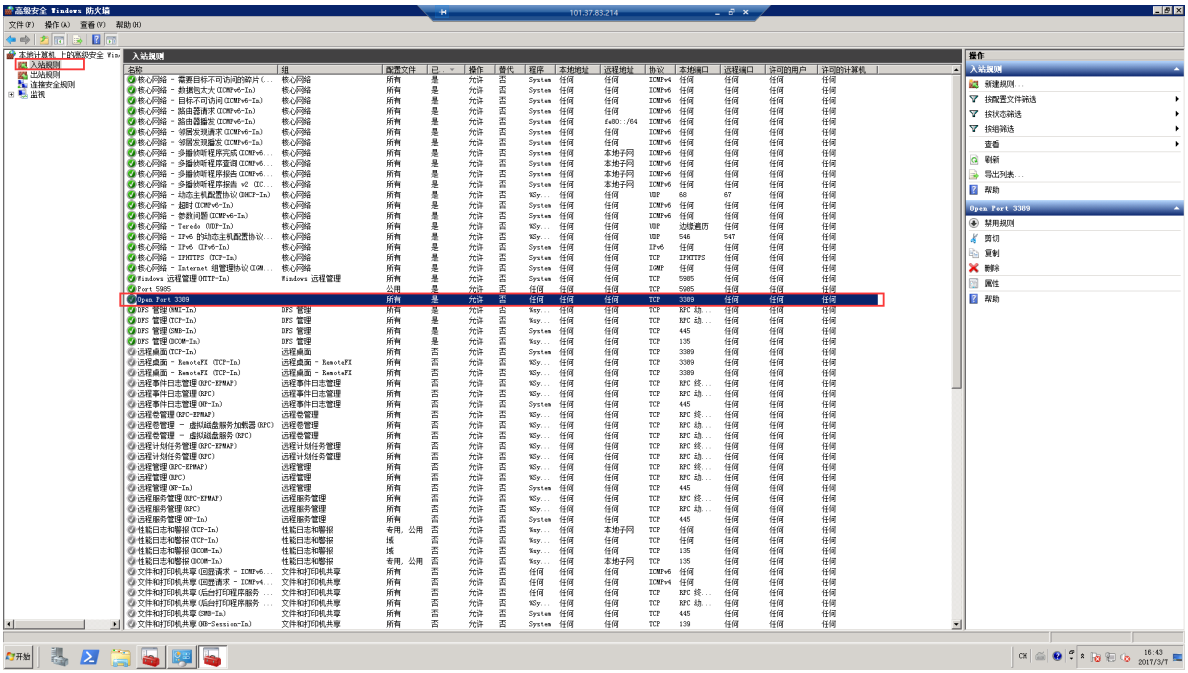
2. 启用防火墙

还是通过上面的步骤开启防火墙，见下图。



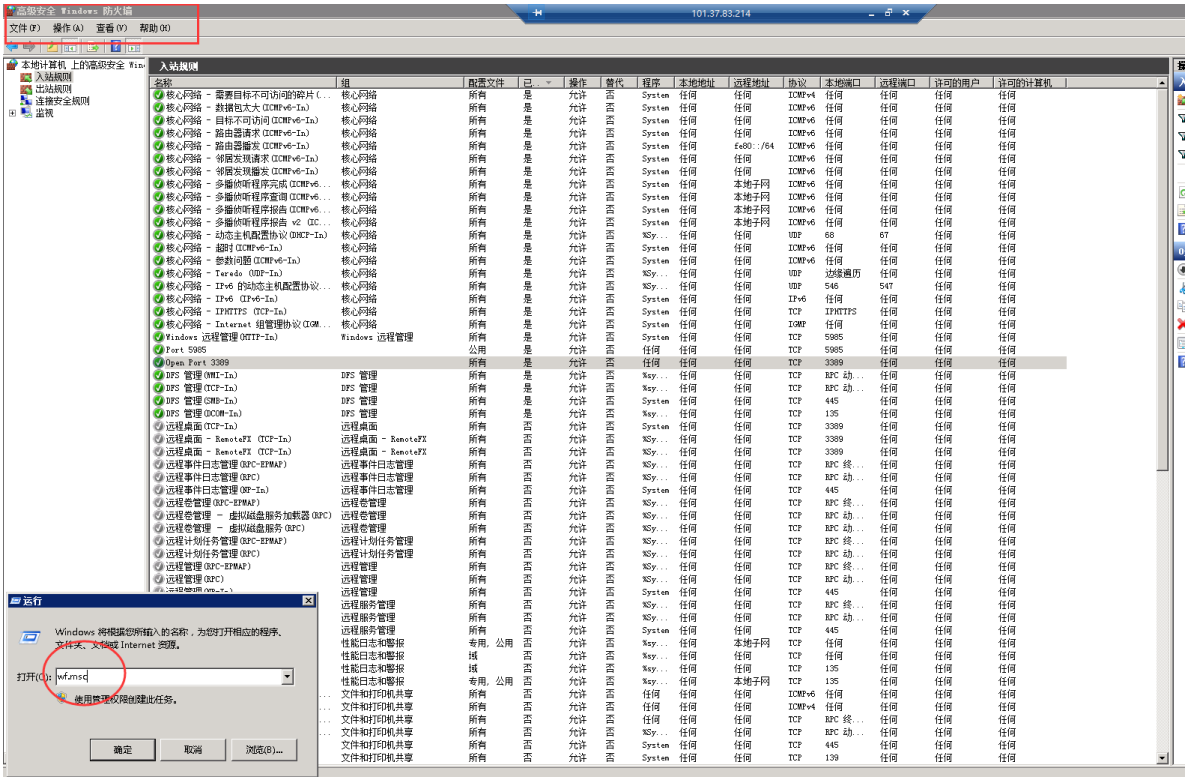
这里需要注意一点的是：启用之前请确认远程端口已经在里面，否则自己也将无法远程，不过高级安全Windows防护墙入站规则默认是放行3389端口的选择高级设置。

选择入站规则，我们看到open port 3389这条入站规则默认是放行3389端口的。

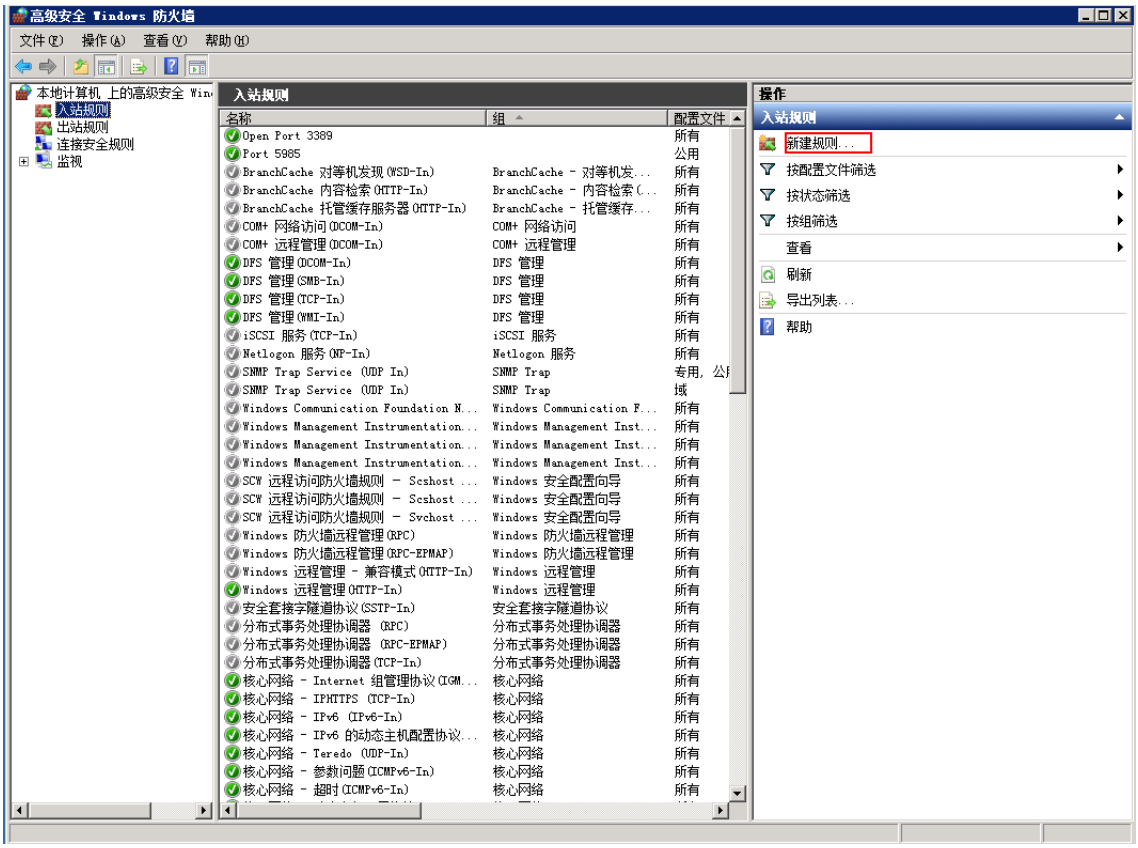


3. 配置高级安全Windows防火墙

键盘输入Win+R打开运行输入*wf.msc* 回车来打开高级安全Windows防火墙，如下图。

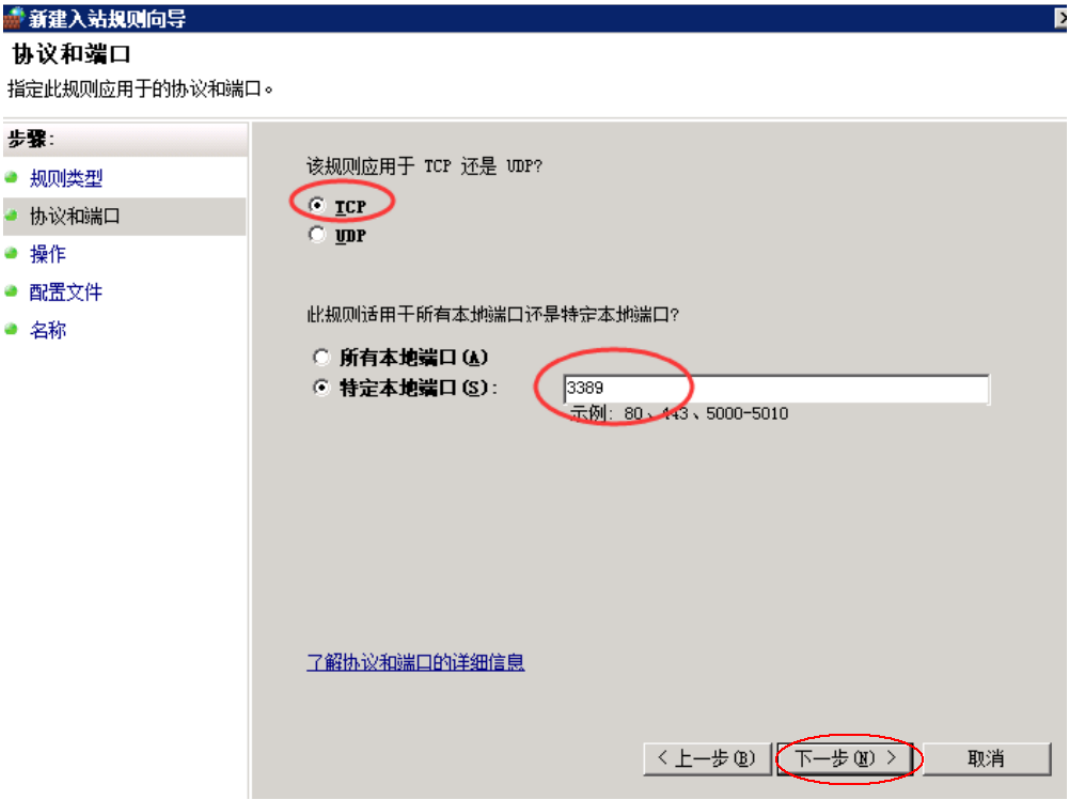


a. 通过手工新建入站规则

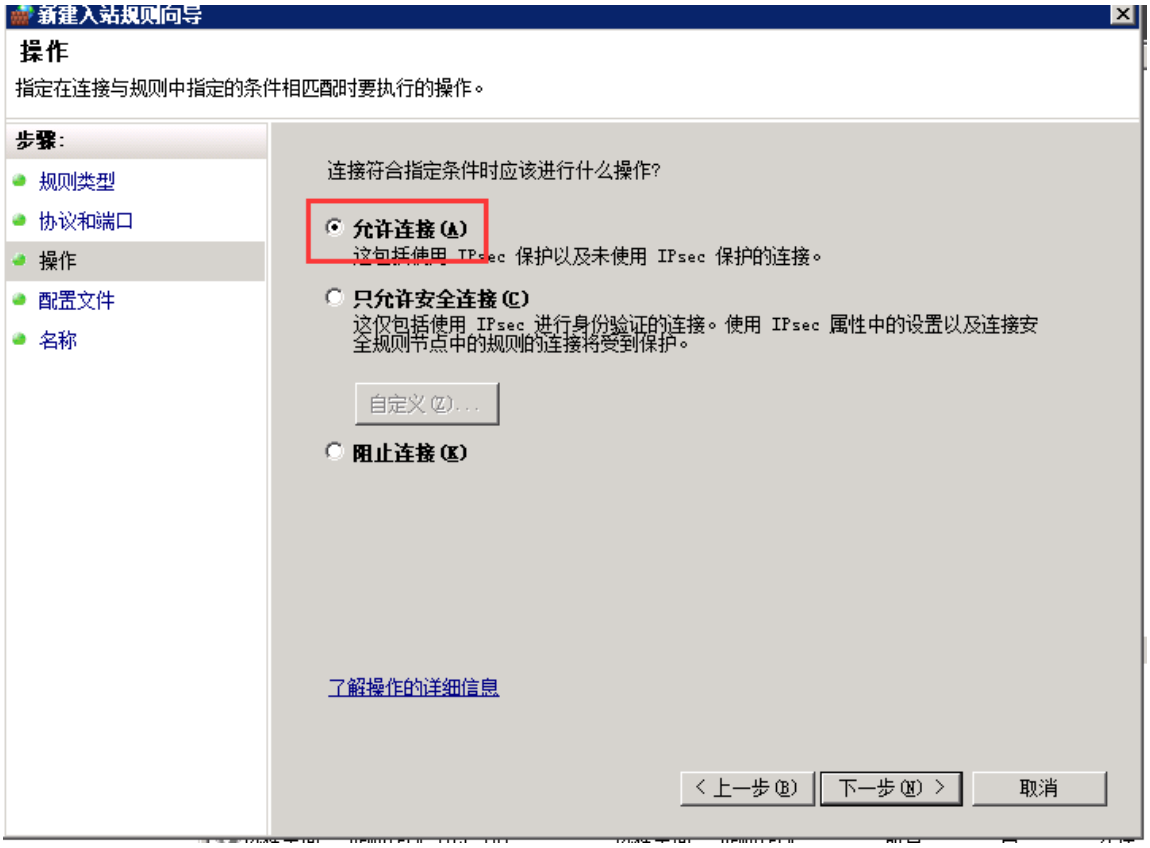


在弹出的新建入站规则向导窗口，选择 端口 然后鼠标左键单击下一步。

而后选择 TCP 并设置特定本地端口3389。

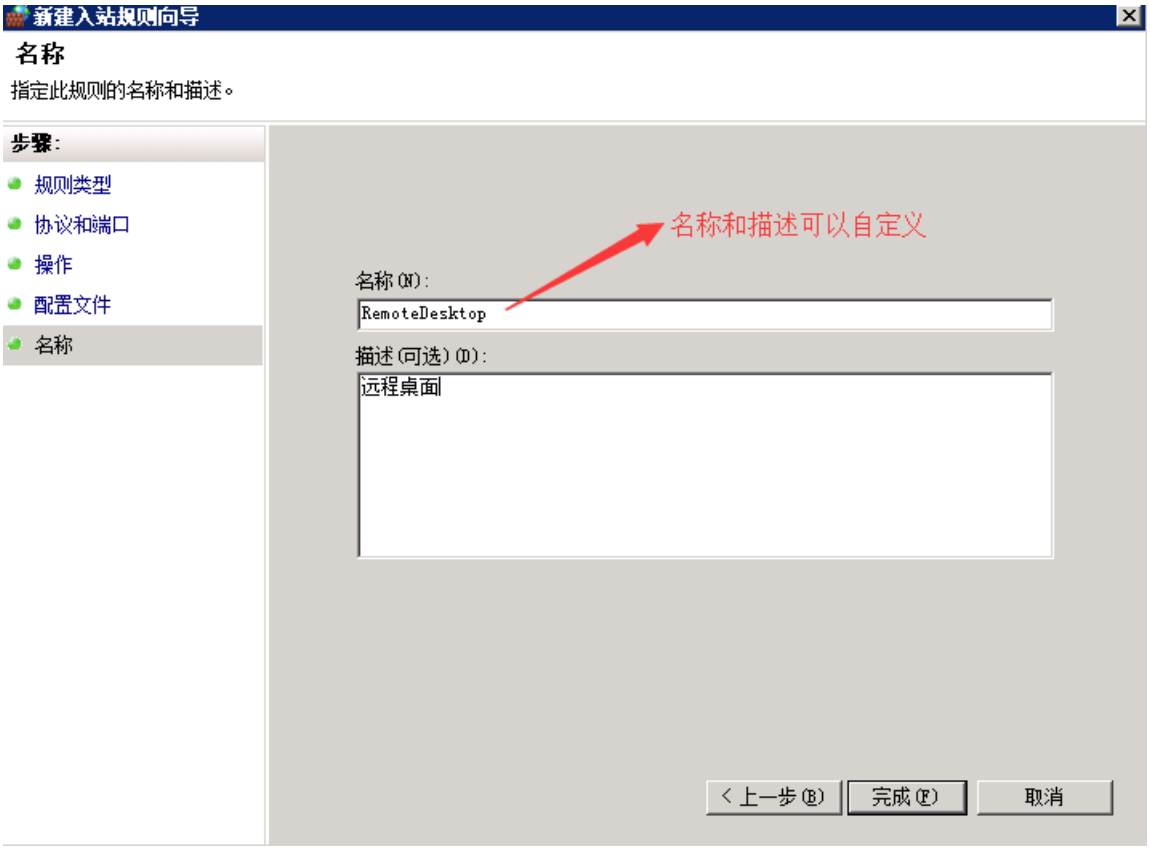


下一步选择允许链接。

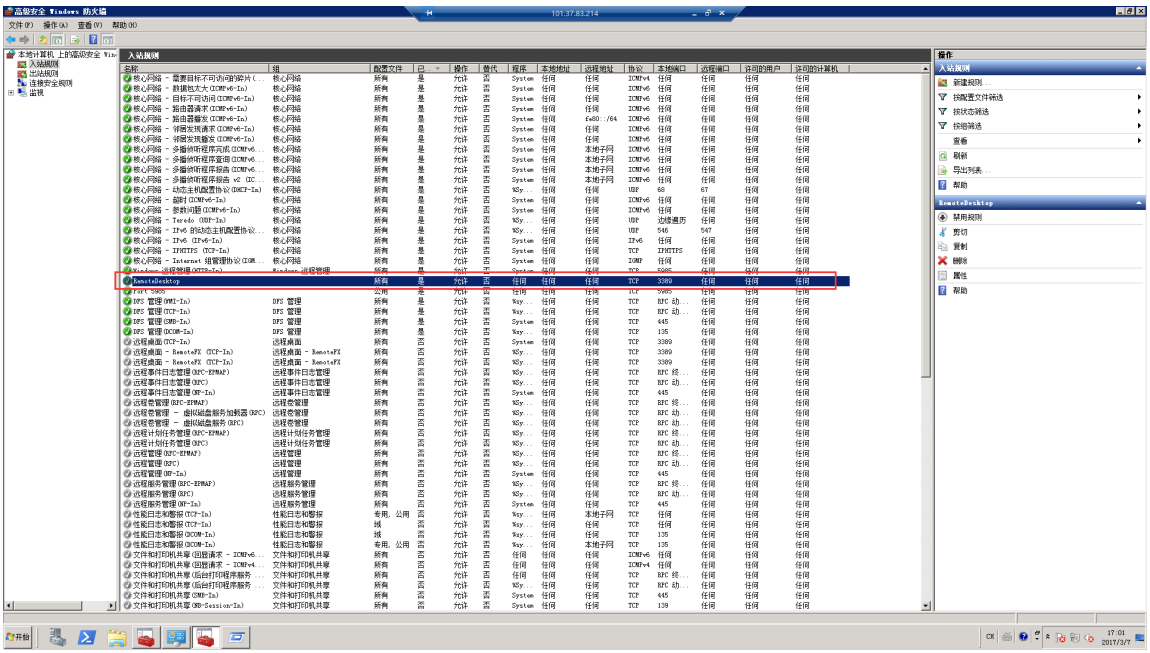


下一步 默认配置即可。

下一步 填写规则名称，例如 RemoteDesktop ,最后鼠标左键单击完成。



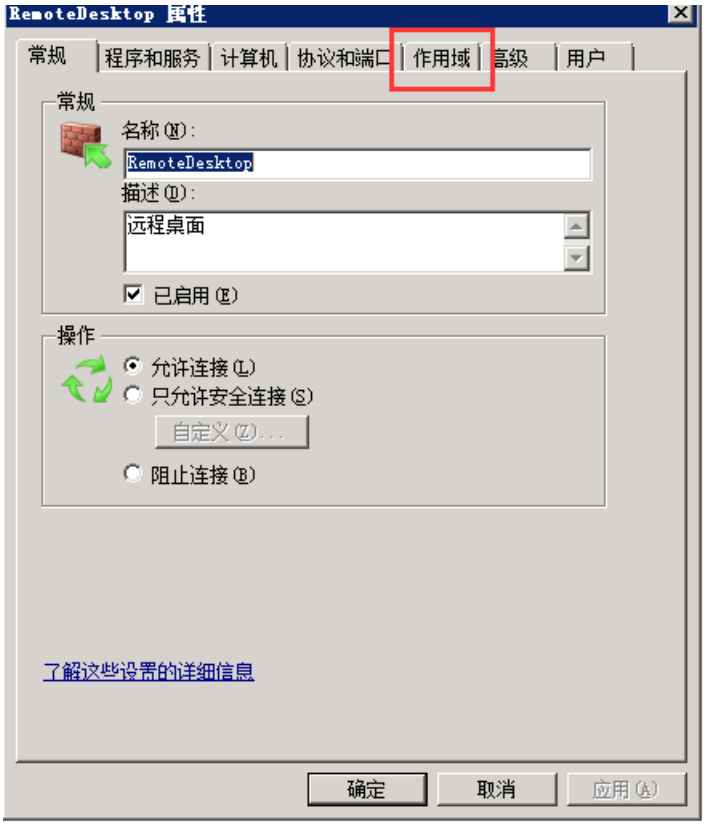
看到我们刚刚添加的规则。



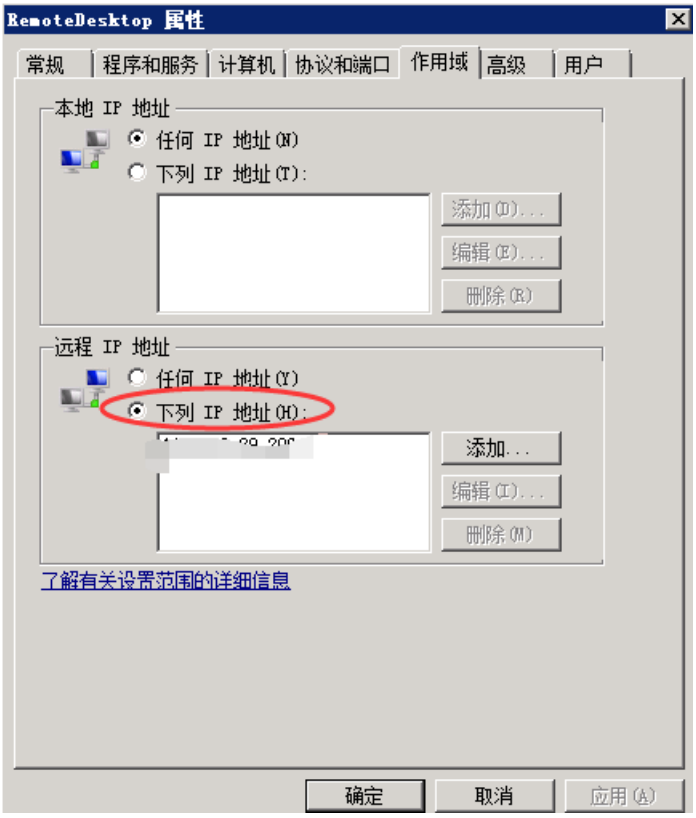
以上步骤就是把Windows远程端口加入到高级安全Windows防火墙了，但是依然没有实现我们的限制访问，接下来我们来实现访问限制。

b. 配置作用域

右键选中我们刚刚创建的入站规则，然后选择属性>作用域>远程IP地址>添加（将需要远程此服务器的IP地址填写进去，注意：一旦启用作用域，除了作用域里面的IP地址，别的地址将无法远程链接此服务器）。

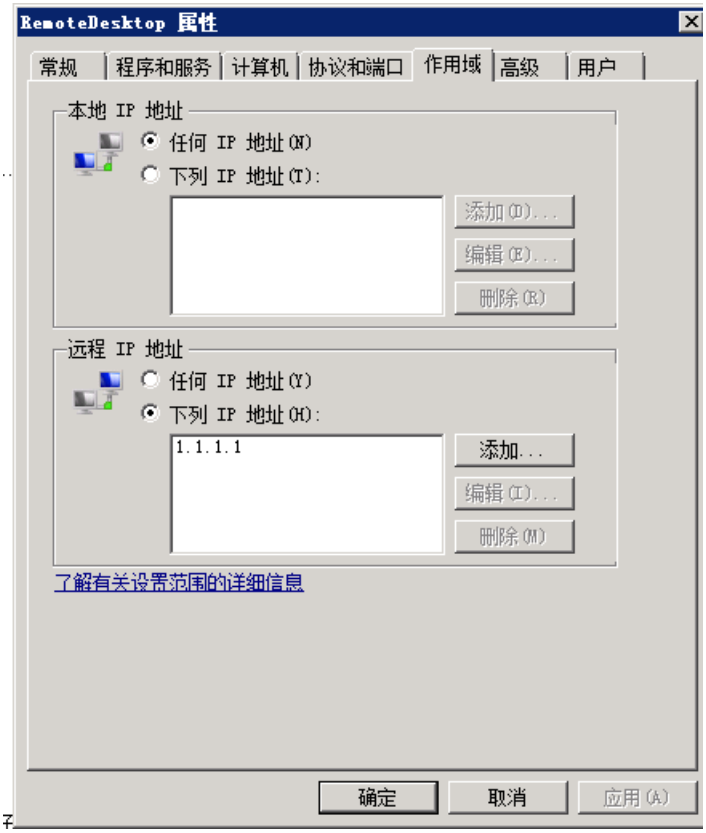


添加远程IP地址。

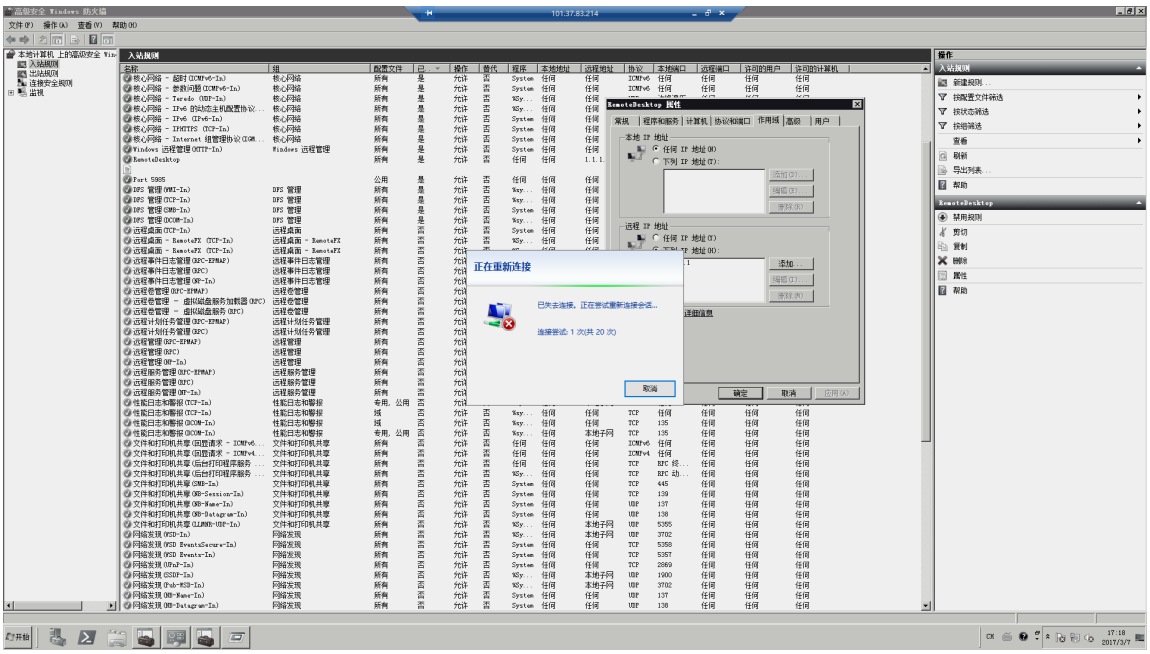


c. 验证作用域

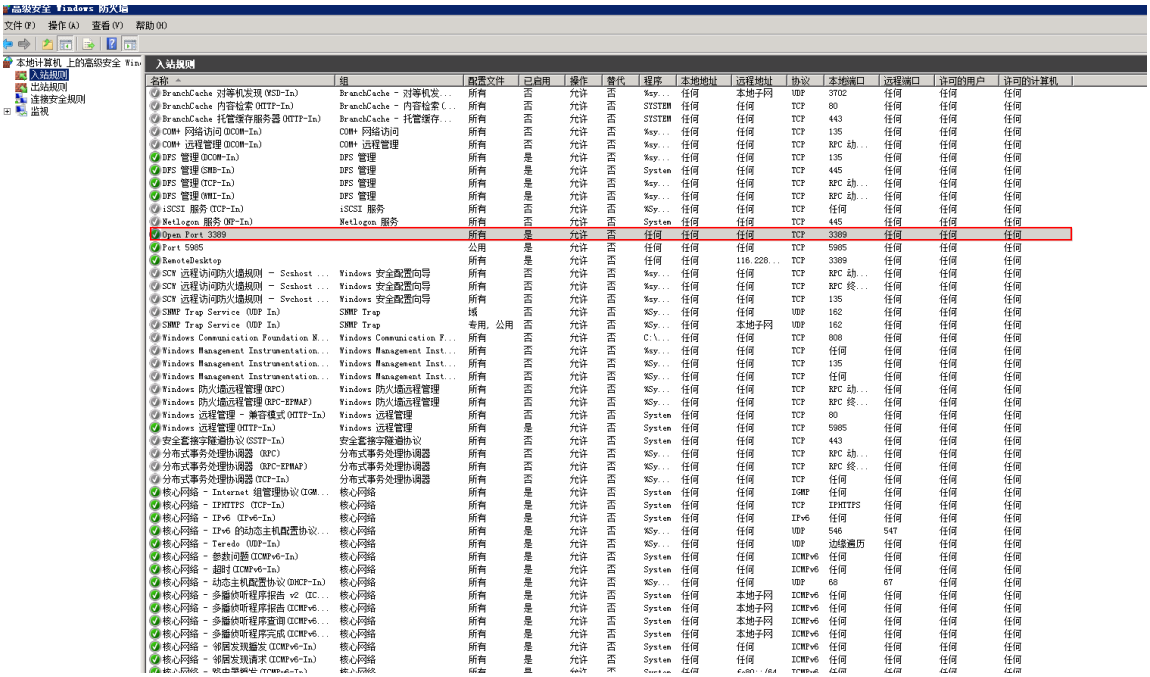
我们在作用域——远程IP地址里面随便写个地址，看看远程连接会发生什么。



远程连接断掉。



如果远程连接没有断开，让我们把下图中open port 3389这条入站规则禁用掉就可以了。

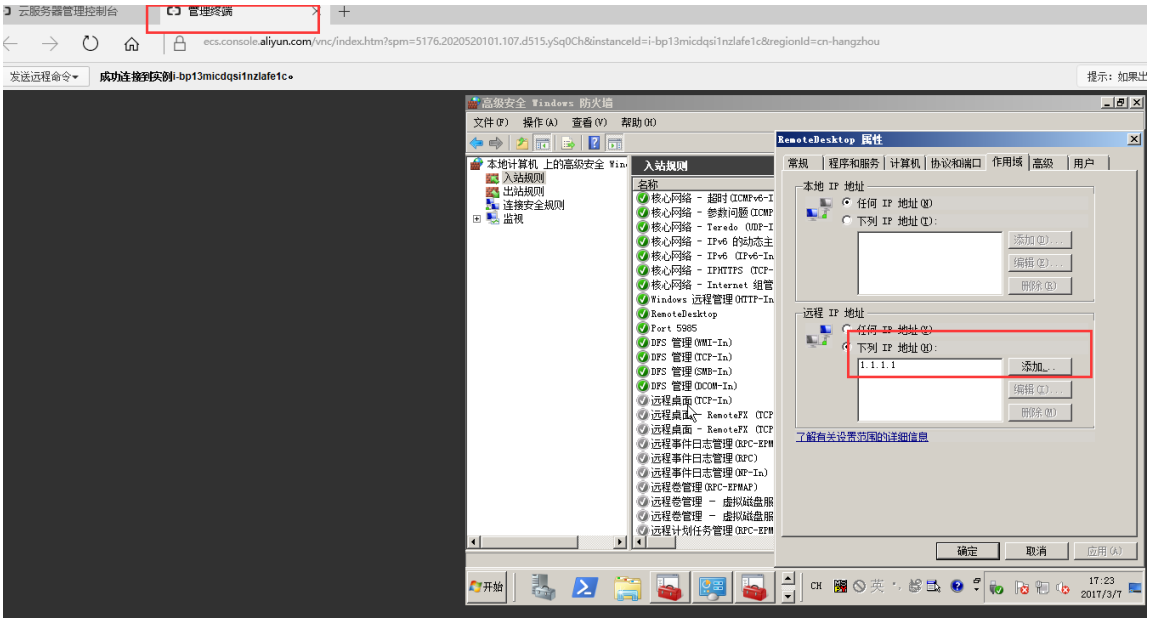


远程连接自己断开了，这就说明我们的作用域生效了，那现在自己都无法远程了，怎么办呢？别急，我们还有阿里云控制台，登录阿里云控制台，然后将上面的作用域地址换成自己的地址（这里要写办公环境的公网地址，除非您的办公环境和阿里云线上的环境打通，）就可以正常远程了。

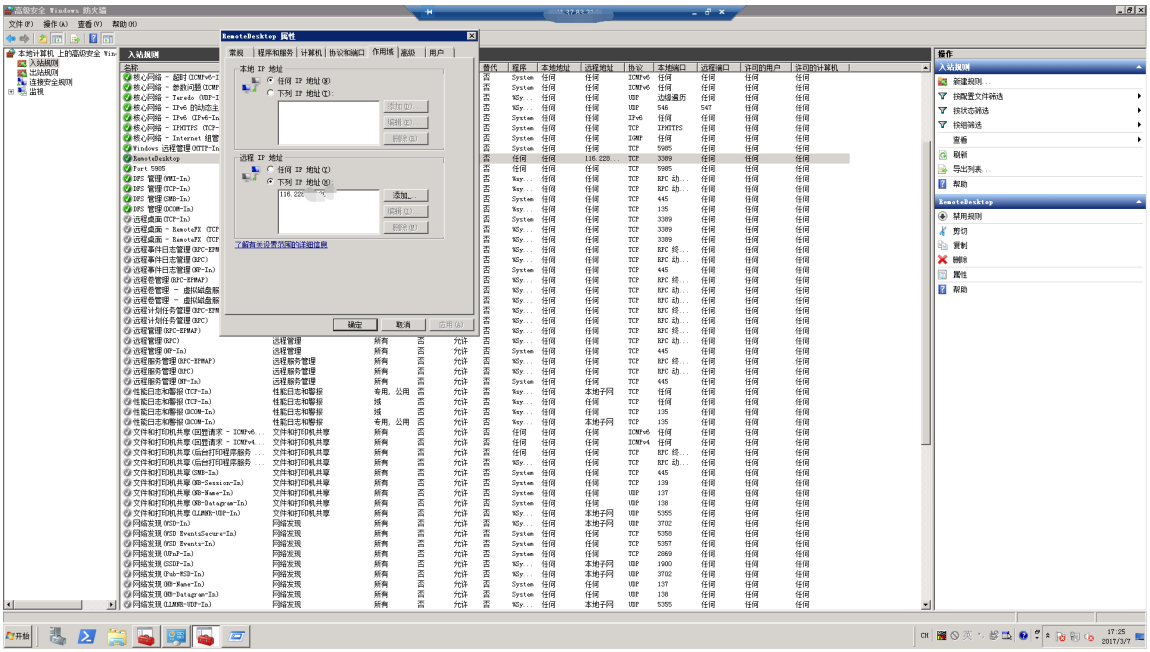
进入阿里云的控制台界面，找到相应实例打开远程连接。

登录系统。

与之前同样的方式，修改RemoteDesktop的作用域的远程IP地址，将之前测试设置的1.1.1.1换回自己的IP地址。



换回自己的IP地址后可以正常远程了，如果不知道自己的公网IP，可以点击此处查看。



以上就是使用高级安全Windows防火墙来实现对服务器远程访问的限制，其他的服务和端口都可以按照上面的方法来实现，例如，关闭不常用的135 137 138 445 端口，限制FTP和相关服务的访问等等，这样才能做到最大限度地保障服务器安全的运行。

命令行的方式

1. 导出防火墙配置到文件。

```
netsh advfirewall export c:\adv.pol
```

2. 导入防火墙配置文件到系统中。

```
netsh advfirewall import c:\adv.pol
```

3. 防火墙恢复默认设置。

```
Netsh advfirewall reset
```

4. 关闭防火墙。

```
netsh advfirewall set allprofiles state off
```

5. 开启防火墙。

```
netsh advfirewall set allprofiles state on
```

6. 在所有配置文件中设置默认阻挡入站并允许出站通信。

```
netsh advfirewall set allprofiles firewallpolicy blockinbound,
allowoutbound
```

7. 删除名为 ftp 的规则。

```
netsh advfirewall firewall delete rule name=ftp
```

8. 删除本地端口 80 的所有入则。

```
netsh advfirewall firewall delete rule name=all protocol=tcp
localport=80
```

9. 添加远程桌面入站规则允许端口3389。

```
netsh advfirewall firewall add rule name=远程桌面(TCP-In-3389)
protocol=TCP dir=in localport=3389 action=allow
```

相关链接

用户可通过云中沙箱平台体验上述文档中的操作，点击此处

*Windows*防火墙限制端口*/IP/*应用访问的方法以及例外的配置

Windows 系统远程桌面端口查看和修改方法

Linux 修改默认远程端口方法

更多开源软件尽在云市场

2 数据恢复

2.1 误删文件后如何恢复数据

本文档主要以CentOS7操作系统为例，介绍如何使用开源工具Extundelete快速恢复被误删除掉的数据。

简介

在日常使用中有时难免会出现数据被误删除的情况，在这个时候该如何快速、有效地恢复数据呢？在阿里云上恢复数据有多种方式，例如：

- 通过阿里云控制台回滚备份好的快照，自定义镜像恢复等方式。
- 购买多台ECS，实现业务的负载均衡，高可用。
- 利用对象存储 *OSS#Object Storage Service#*，存储静态网页和海量图片、视频等重要数据。

在Linux下，基于开源的数据恢复工具有很多，常见的有debugfs、R-Linux、ext3grep、extundelete等，比较常用的有ext3grep和extundelete，这两个工具的恢复原理基本一样，只是extundelete功能更加强大。

Extundelete是基于linux的开源数据恢复软件。在使用阿里云的云服务器时，如果您不小心误删除数据，并且Linux系统也没有与Windows系统下回收站类似的功能，您可以方便快速安装此工具。

Extundelete能够利用inode信息结合日志去查询该inode所在的block位置，以次来查找和恢复所需的数据，该工具最给力的一点就是支持ext3/ext4双格式分区恢复，基于整个磁盘的恢复功能较为强大。

在数据被误删除后，第一时间要做的是卸载被删除数据所在的磁盘或磁盘分区。因为将文件删除后，仅仅是将文件的inode结点中的扇区指针清零，实际文件还存储在磁盘上，如果磁盘以读写模式挂载，这些已删除的文件的数据块就可能被操作系统重新分配出去，在这些数据块被新的数据覆盖后，这些数据就真的丢失了，恢复工具也回力无天。所以，以只读模式挂载磁盘可以尽量降低数据块中数据被覆盖的风险，以提高恢复数据成功的几率。

说明：

在实际线上恢复过程中，切勿将extundelete安装到您误删的文件所在硬盘，这样会有一定几率将需要恢复的数据彻底覆盖，切记操作前做好快照备份。

适用对象

- 磁盘中文件误删除的用户，且未对磁盘进行过写入等操作
- 网站访问量小、少量 ECS 实例的用户

使用方法

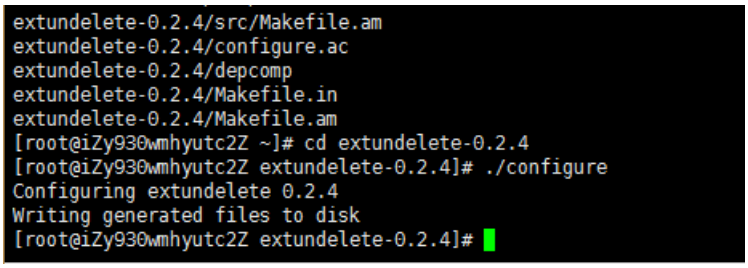
需安装的软件及版本：e2fsprogs-devel e2fsprogs gcc-c++ make（编译器等）Extundelete-0.2.4。

> 说明：
> extundelete需要libext2fs版本1.39或更高版本来运行，但是对于ext4支持，请确保您有e2fsprogs版本1.41或更新版本（可以通过运行命令“dumpe2fs”并记录其输出的版本）。

以上版本是写文档时的软件版本。您下载的版本可能与此不同。

- 部署**extundelete**工具

```
wget  http://zy-res.oss-cn-hangzhou.aliyuncs.com/server/extundelete-0.2.4.tar.bz2
yum -y install  bzip2  e2fsprogs-devel  e2fsprogs  gcc-c++  make
 #安装相关依赖和库
tar -xvjf extundelete-0.2.4.tar.bz2
cd extundelete-0.2.4                                    #进入程序目录
./configure                                             #如下图表示安装成功
```



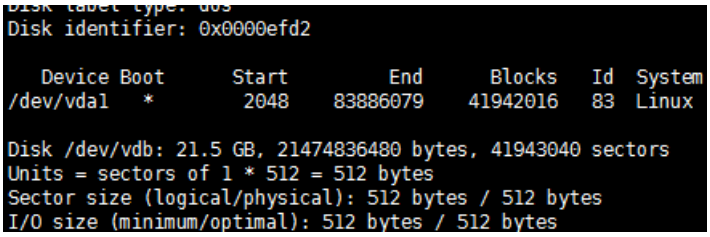

```
make && make install
```

这个时候会出现src目录，下面有个extundelete可执行文件以及相应路径，如下图，其实默认文件安装在*usr/local/bin*下面，下面演示就在*usr/local/bin*目录下。

- 使用**extundelete**，模拟数据误删除然后恢复的过程

1. 检查ECS现有的磁盘和可用分区，并对/dev/vdb进行分区，格式化，此处不在介绍磁盘分区格式化方式，如果不会的话可以点击此文档查看操作方式格式化和挂载数据盘。

```
fdisk -l
```

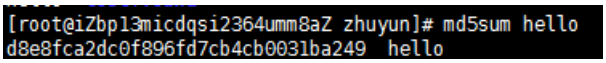


2. 将分区后的磁盘挂载到/zhuyun目录下，然后在/zhuyun下面新建测试文件hello,写入test。

```
mkdir /zhuyun                                    #新建zhuyun目录
mount /dev/vdb1 /zhuyun                          #将磁盘挂载到zhuyun目录
下
echo test > hello                                #写入测试文件
```

3. 记录文件MD5值，md5sum命令用于生成和校验删除前和恢复后俩个文件的md5值。

```
md5sum hello
```



4. 模拟删除hello文件。

```
rm -rf hello
cd ~
```

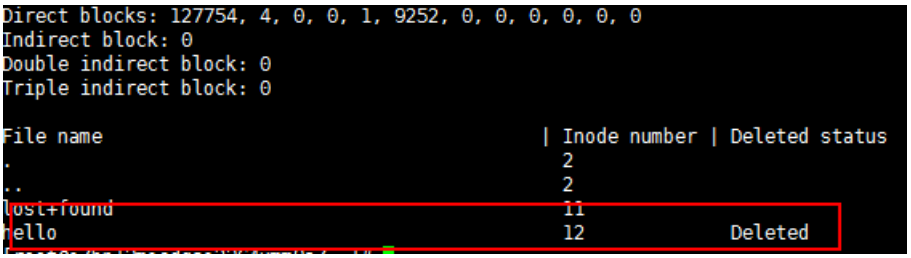
```
fuser -k /zhuyun                                    #结束使用某分区的进程树（确认没有资源占用的话，可以跳过此步）
```

5. 卸载数据盘。

```
umount /dev/vdb1                                    #任何的文件恢复工具，在使用前，均要将要恢复的分区卸载或挂载为只读，防止数据被覆盖使用
```

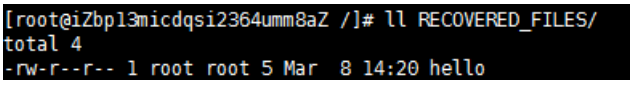
6. 使用Extundelete工具恢复文件。

```
extundelete --inode 2 /dev/vdb1          #为查找某i节点中的内容，使用2则说明为整个分区搜索，如果需要进入目录搜索，只须要指定目录I节点即可。这是可以看到删除的文件名和inode
```




```
/usr/local/bin/extundelete  --restore-inode 12  /dev/vdb1    #恢复删除的文件
```

这个时候会在执行命令的同级目录下出现RECOVERED_FILES目录，查看是否恢复。



通过md5值查看，前后俩个文件，一样说明恢复成功。

```
--restore-inode 12                  # --restore-inode 按指定的I节点恢复
--extundelete --restore-all         # --restore-all   全部恢复
```

相关链接

用户可通过云中沙箱平台体验上述文档中的操作，点击此处。

2.2 Windows 实例磁盘空间满的问题处理及最佳实践

本文主要介绍 Windows 实例磁盘空间不足时对应的解决方法以及磁盘日常维护的最佳实践。

本文中的方法适用于 Windows Server 2003 以上系统，这里以 Windows Server 2008 R2 为例。

说明：

Linux 实例磁盘空间不足时对应的处理方法参考 *ECS Linux* 磁盘空间满排查处理。

解决方法

解决 Windows 磁盘空间满的问题，有以下两种处理方式：

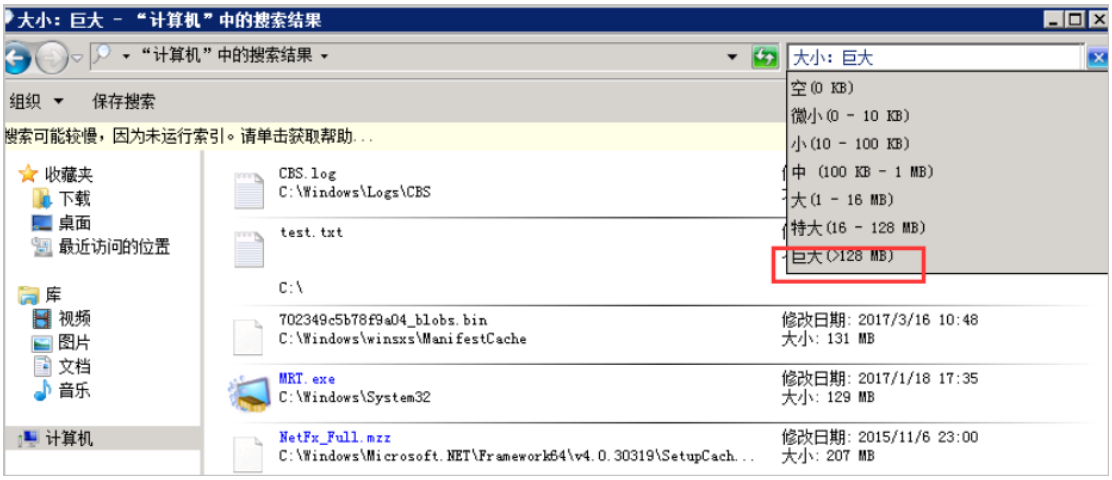
- 释放磁盘空间
- 扩容磁盘

- 释放磁盘空间

  您可以通过清理磁盘中不需要的文件来解决磁盘空间满的问题，首先找出占用磁盘空间过多的文件，然后删除不需要的文件，具体步骤如下：

  — 找出占用磁盘空间过多的文件

  1. 远程连接并登录到 Windows 实例。
  2. 双击计算机，单击要清理的磁盘，按下键盘的 Ctrl+F 键，定位到搜索框。
  3. 在搜索框中，选择大小，然后根据系统定义大小筛选指定磁盘的大文件。



> **说明：**
> 您也可以自定义文件大小范围进行检索，如输入大小*#>500M*，会检索该磁盘大于 500 M 的文件。如输入大小*#> 100M < 500M* ，会检索大于 100 M 但小于 500 M 的文件。

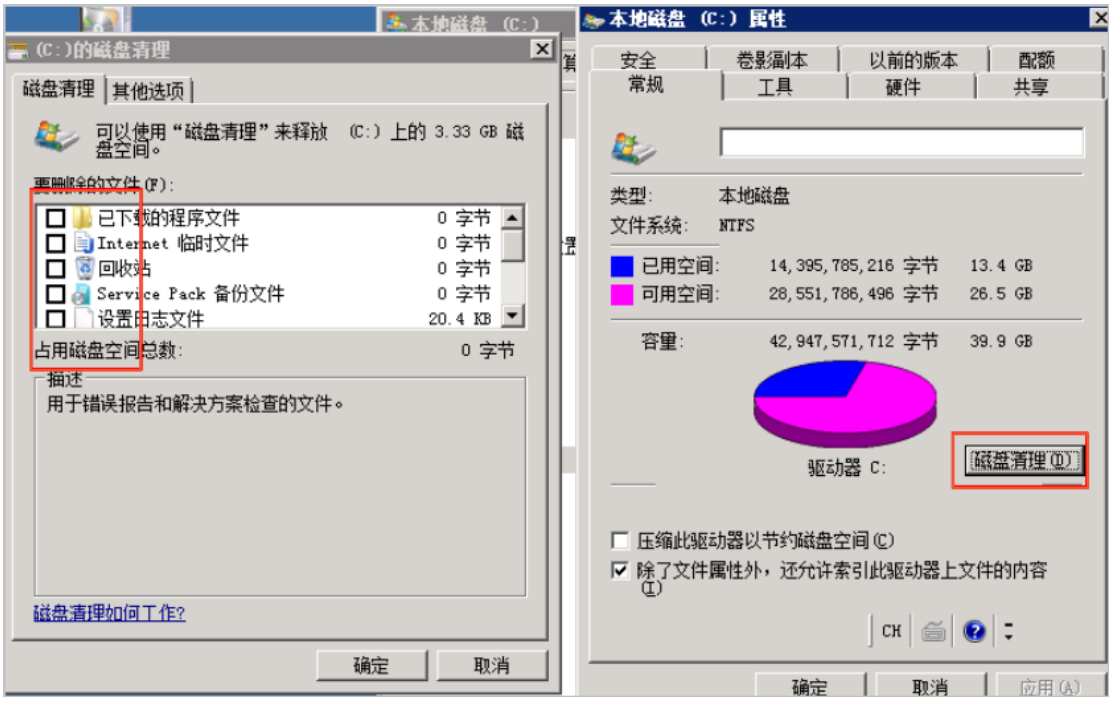
  — 删除不需要的文件

  找出占用了磁盘空间过多的文件后，如果文件不再需要，可以及时清理。

  推荐您使用系统自带的磁盘清理工具，删除日志文件及系统上其他不需要文件，并清空回收站。磁盘清理工具服务器默认没有安装，需要手动安装，具体安装以及删除文件的步骤如下：

  1. 打开服务器管理器，单击功能，然后单击添加功能。

2. 在添加功能向导窗口，勾选墨迹手写服务和桌面体验，然后单击下一步。
3. 在弹出的窗口中，单击安装。
4. 安装页面上，系统将提示您手动重新启动服务器，单击是重新启动服务器。重新启动服务器之后，确认已安装了桌面体验。
5. 安装完成后，选择开始 > 所有程序 > 附件 > 系统工具 > 磁盘清理，选择要清理的选项，单击确定。



- 扩容磁盘

  您可以通过扩容磁盘的方式解决磁盘空间满的问题，具体步骤参考扩容 Windows 系统盘，扩容 Windows 数据盘。

最佳实践

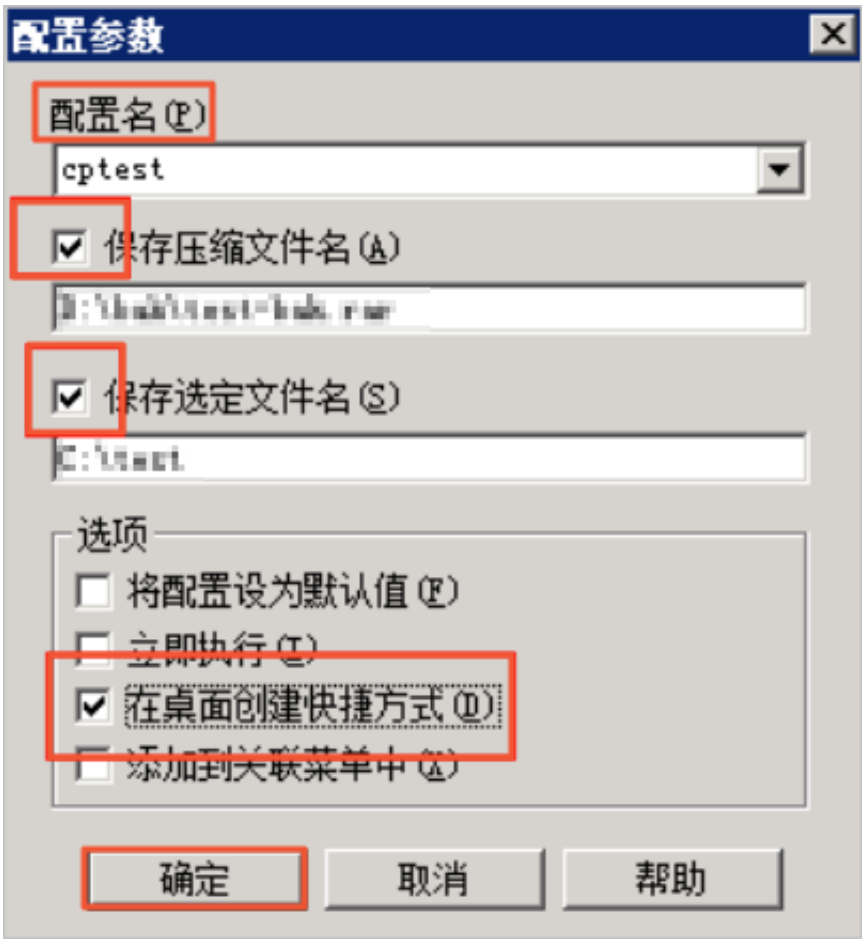
日常需要养成良好的磁盘使用习惯，这里推荐以下几个磁盘使用的最佳实践：

- 文件压缩保存
- 定期清理不必要的应用程序
- 设置磁盘监控

- 文件压缩保存

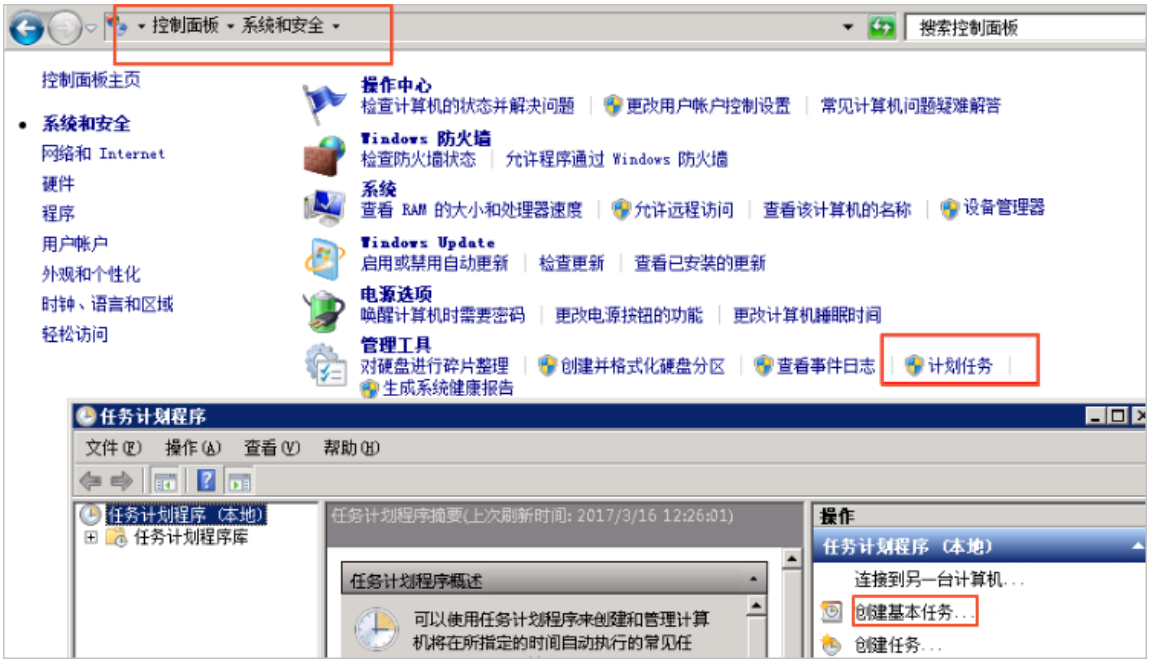
  磁盘中一些定期生成的文件可以进行归档压缩后保存，以提高磁盘使用率。压缩工具推荐使用 WinRAR，配置压缩策略过程如下：

1. 安装好软件后找到需要压缩的文件，右键该文件，选择添加到压缩文件。
2. 在设置界面单击窗口上方备份选项卡，然后勾选按掩码产生文件名，注意此时不要单击确定。
3. 单击窗口上方常规选项卡，单击浏览来定义压缩文件的路径。单击配置，选择保存当前配置为新配置。
4. 在弹出的配置参数窗口中，输入配置名，勾选保存压缩文件名、保存选定文件名、桌面创建快捷方式，单击确定。

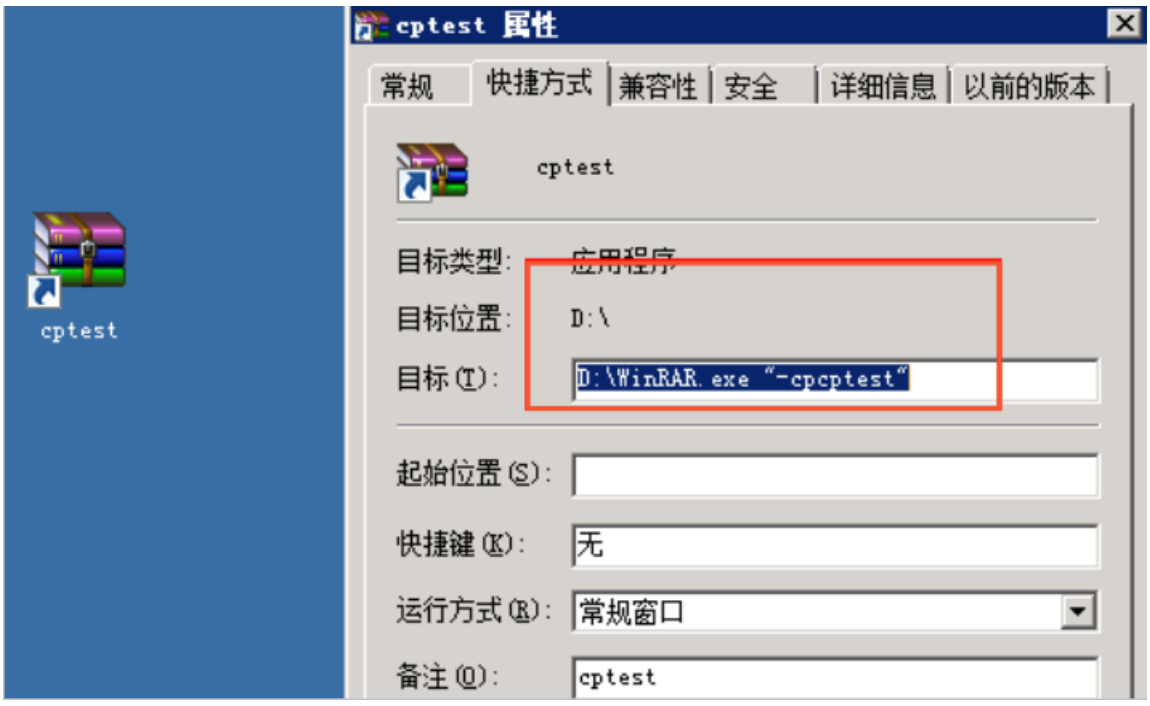


5. 然后在压缩文件名和参数窗口，单击确定。桌面会生成一个此压缩包的快捷键。

6. 选择开始 > 控制面板，单击系统和安全，单击计划任务，然后在任务计划程序窗口中，选择创建基本任务。



7. 在弹出的窗口中为新任务命名，单击下一步。
8. 选择触发周期，单击下一步。然后选择启动程序，单击下一步。
9. 此时会弹出窗口需要您输入程序或脚本。先找到刚才生成的压缩包快捷键，右键该快捷键，选择属性，复制目标内容。

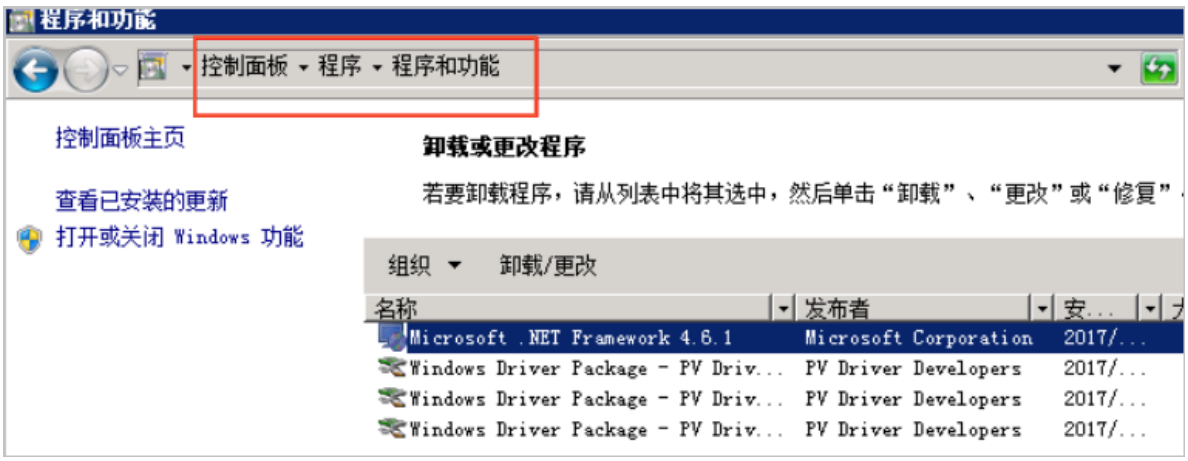


10. 然后将复制内容粘贴到启动程序操作中的程序或脚本文本框中，单击确定完成创建。

设置好备份策略以后，可以定期的去清理过期的备份文件，避免占用过大的空间。

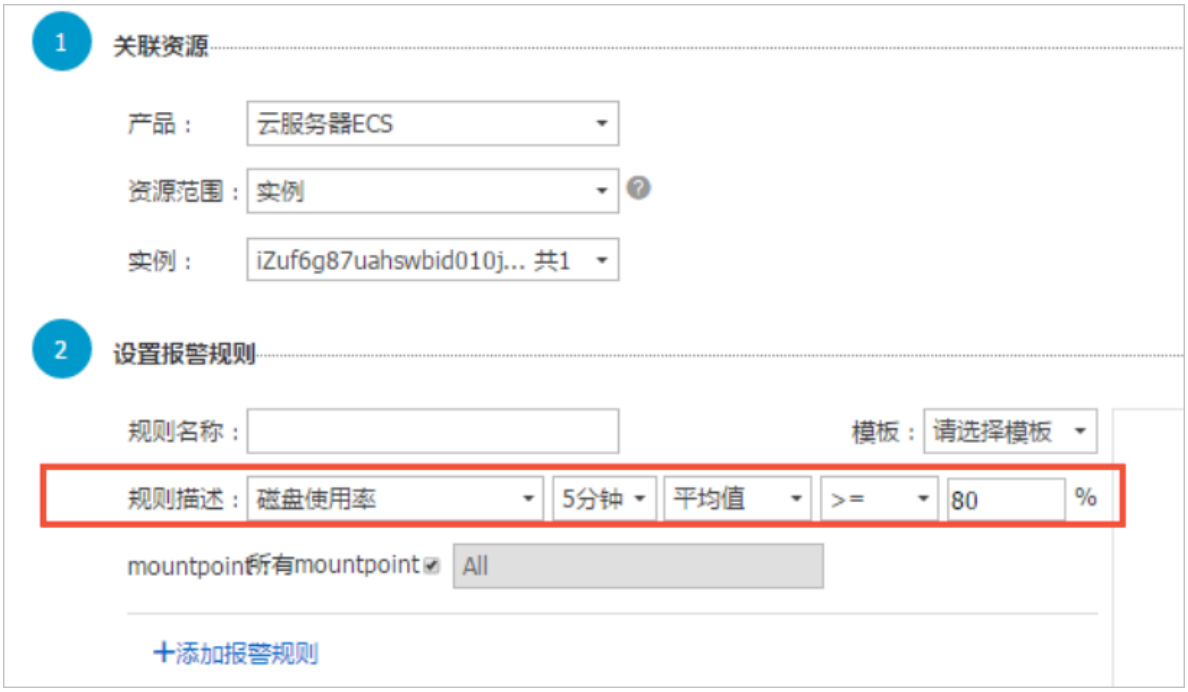
- 定期清理不必要的应用程序

  定期清理不必要的应用程序，您可以通过**控制面板**中的**程序和功能**窗口清理不再使用的程序软件。



- 设置磁盘监控

  阿里云的 ECS 服务器默认安装了监控插件，您可以在云监控控制台中创建磁盘报警规则。这样可以实时了解磁盘空间使用率是否到达一个高位值，以便及时清理。



2.3 Linux实例中数据恢复

在处理磁盘相关问题时，您可能会碰到操作系统中数据盘分区丢失的情况。本文介绍了Linux系统下常见的数据盘分区丢失的问题以及对应的处理方法，同时提供了使用云盘的常见误区以及最佳实践，避免可能的数据丢失风险。

在修复数据前，您必须先对分区丢失的数据盘创建快照，在快照创建完成后再尝试修复。如果在修复过程中出现问题，您可以通过快照回滚将数据盘还原到修复之前的状态。

前提条件

在修复数据前，您必须先对分区丢失的数据盘创建快照，在快照创建完成后再尝试修复。如果在修复过程中出现问题，您可以通过快照回滚将数据盘还原到修复之前的状态。

工具说明

在Linux实例里，您可以选择以下任一种工具修复磁盘分区并恢复数据：

- fdisk：Linux系统默认安装的分区工具。
- testdisk：主要用恢复Linux系统的磁盘分区或者数据。Linux系统默认不安装，您需要自行安装这个软件，比如，在CentOS系统里，您可以运行 **`yum install -y testdisk`** 在线安装。
- partprobe：Linux系统默认安装的工具。主要用于不重启系统时让kernel重新读取分区。

Linux系统下数据盘分区丢失和数据恢复处理办法

在Linux实例里，您重启系统后，可能会出现数据盘分区丢失或者数据丢失的问题。这可能是因为您未在 *etc/fstab* 文件里设置自动挂载。此时，您可以先手动挂载数据盘分区。如果手动挂载时报分区表丢失，您可以通过如下三种办法尝试进行处理：通过*fdisk*恢复分区、通过*testdisk*恢复分区或者 通过*testdisk*直接恢复数据。

- 通过**fdisk**恢复分区

  对数据盘分区时，分区磁盘的起止扇区一般使用默认的值，所以可以先尝试直接使用 fdisk 新建分区进行恢复。具体操作，请参考 *Linux* 格式化和挂载数据盘。

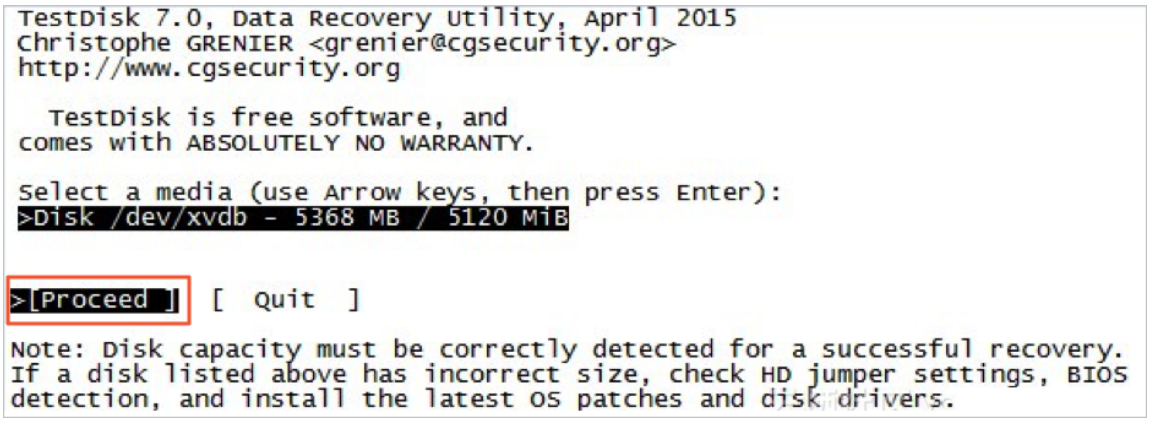
```
[root@Aliyun ~]# fdisk /dev/xvdb
Welcome to fdisk (util-linux 2.23.2).

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
```

  如果上述操作无效，您可以使用 testdisk 工具尝试修复。

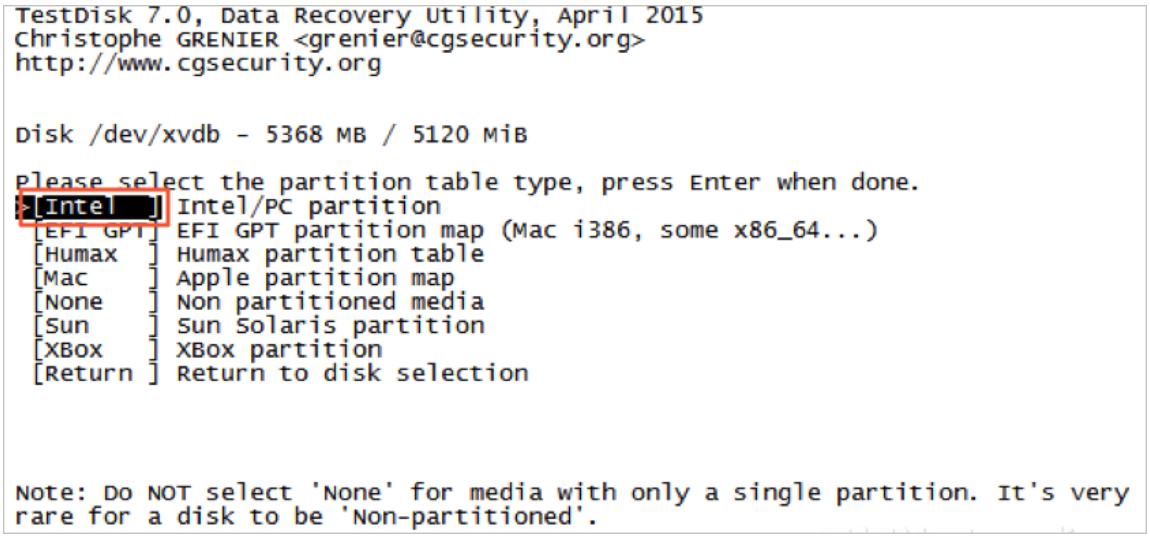
- 通过 **testdisk** 恢复分区

  这里假设云盘的设备名为 */dev/xvdb*。按以下步骤使用 testdisk 恢复分区：

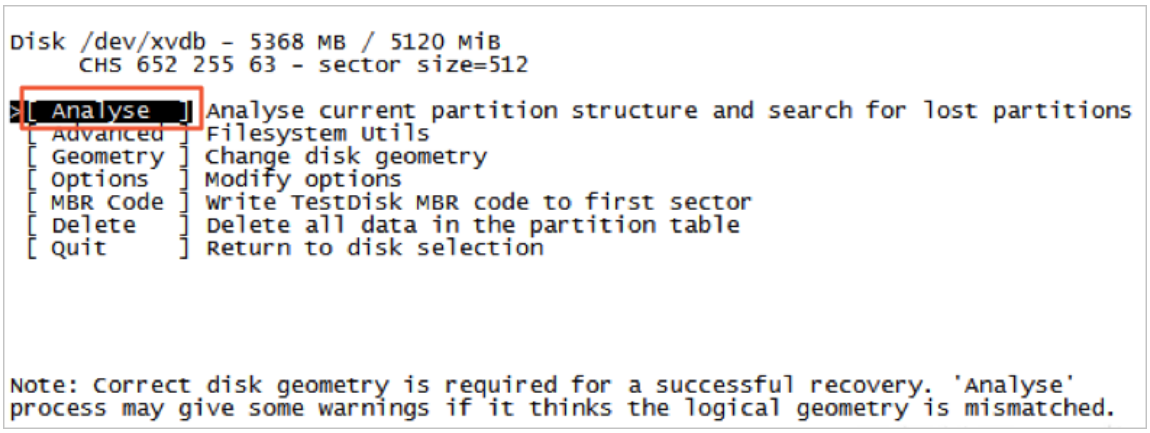
  1. 运行 **`testdisk /dev/xvdb`**（根据实际情况替换设备名），再选择 **`Proceed`**（默认值）后按回车键。



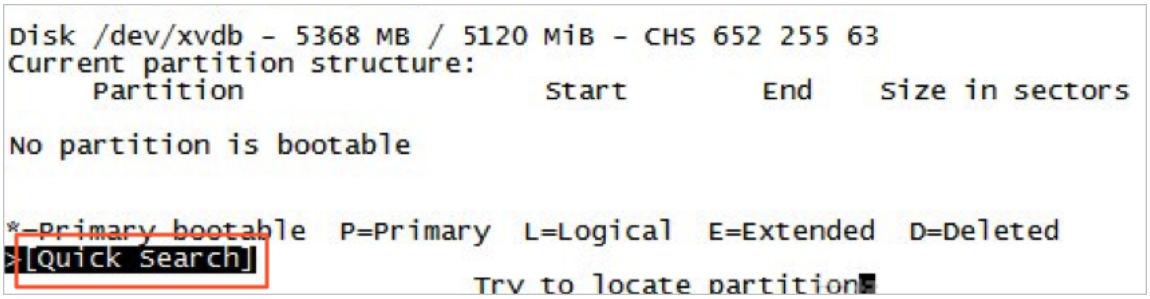
2. 选择分区表类型进行扫描：一般选择 *Intel*（默认）。如果您的数据盘采用GPT分区，选择 *EFI GPT*。



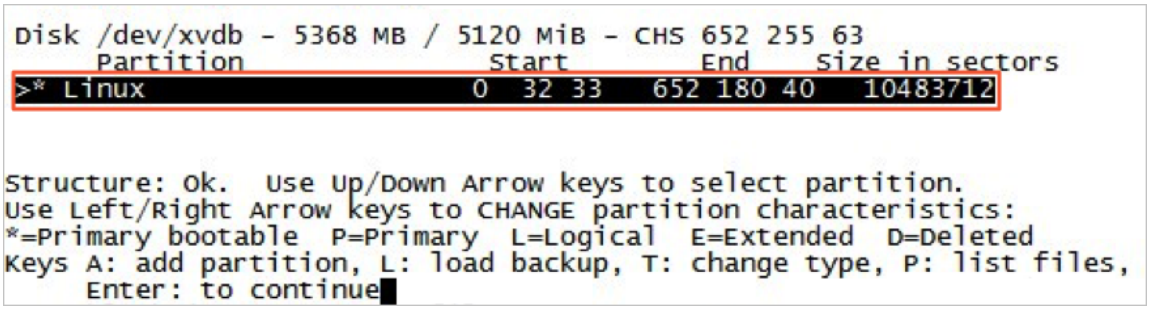
3. 选择 *Analyse* 后按回车键。



4. 如果您没有看到没有任何分区信息，选择 *Quick Search* 后按回车键快速搜索。

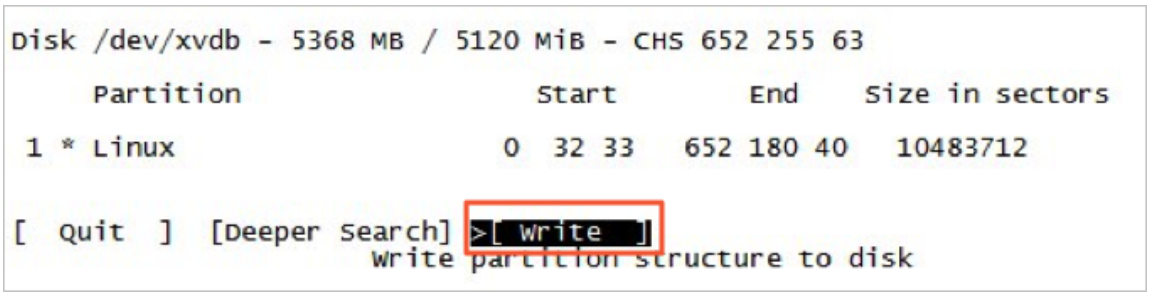


在返回结果中会显示分区信息，如下图所示。

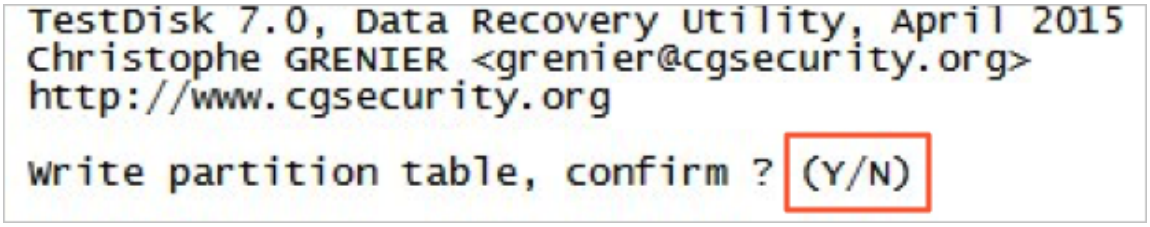


5. 选中分区后，按回车键。
6. 选择 *Write* 保存分区。

> 说明：
> 如果不是您需要的分区，可以选择 *Deeper Search* 继续搜索。



7. 按 Y 键确认保存分区。



8. 运行 **partprobe /dev/xvdb**（根据实际情况替换设备名）手动刷新分区表。
9. 重新挂载分区，查看数据盘里的数据情况。

```
[root@Aliyun home]# mount /dev/xvdb1 /mnt/
[root@Aliyun home]# ls /mnt/
123.sh  configclient  data  diamond  install_edsd.sh  install.sh  ip.gz  logs  lost+found  test
```

- **通过testdisk直接恢复数据**

  在某些情况下，您可以用testdisk扫描出磁盘分区，但是无法保存分区，此时，您可以尝试直接恢复文件。具体操作步骤如下所示：

  1. 按 通过*testdisk*恢复分区 的第1步到第4步描述找到分区。
  2. 按 P 键列出文件。 返回结果如下图。

```
   * Linux                                0  32 33   652 180 40   10483712
Directory /

 drwxr-xr-x    0     0      4096 21-Feb-2017 11:57 .
 drwxr-xr-x    0     0      4096 21-Feb-2017 11:57 ..
 drwx------    0     0     16384 21-Feb-2017 11:56 lost+found
 -rw-r--r--    0     0      1701 21-Feb-2017 11:57 install_edsd.sh
 -rw-r--r--    0     0      5848 21-Feb-2017 11:57 install.sh
>-rw-r--r--    0     0     12136 21-Feb-2017 11:57 ip.gz
 -rw-r--r--    0     0         0 21-Feb-2017 11:57 test
 drwxr-xr-x    0     0      4096 21-Feb-2017 11:57 123.sh
 drwxr-xr-x    0     0      4096 21-Feb-2017 11:57 configclient
 drwxr-xr-x    0     0      4096 21-Feb-2017 11:57 data
 drwxr-xr-x    0     0      4096 21-Feb-2017 11:57 diamond
 drwxr-xr-x    0     0      4096 21-Feb-2017 11:57 logs


                                                  Next
Use Right to change directory, h to hide deleted files
    q to quit, : to select the current file, a to select all files
    C to copy the selected files. c to copy the current file
```

  3. 选中要恢复的文件，再按 C 键。
  4. 选择目标目录。本示例中以恢复到 */home* 为例。

```
Please select a destination where /ip.gz will be copied.
Keys: Arrow keys to select another directory
      C when the destination is correct
      Q to quit
Directory /
 drwxr-xr-x     0     0      4096 11-Jan-2017 09:32 .
 drwxr-xr-x     0     0      4096 11-Jan-2017 09:32 ..
 dr-xr-xr-x     0     0      4096 25-Jul-2016 16:23 boot
 drwxr-xr-x     0     0      2940 21-Feb-2017 12:30 dev
 drwxr-xr-x     0     0      4096 21-Feb-2017 12:12 etc
>drwxr-xr-x     0     0      4096 16-Feb-2017 11:48 home
 drwx------     0     0     16384 12-May-2016 19:58 lost+found
 drwxr-xr-x     0     0      4096 12-Aug-2015 22:22 media
 drwxr-xr-x     0     0      4096 21-Feb-2017 11:57 mnt
 drwxr-xr-x     0     0      4096 12-Aug-2015 22:22 opt
 dr-xr-xr-x     0     0         0 16-Feb-2017 21:35 proc
 dr-xr-x---     0     0      4096 21-Feb-2017 11:57 root
 drwxr-xr-x     0     0       560 21-Feb-2017 12:12 run
 drwxr-xr-x     0     0      4096 12-Aug-2015 22:22 srv
 dr-xr-xr-x     0     0         0 16-Feb-2017 21:35 sys
 drwxrwxrwt     0     0      4096 21-Feb-2017 12:34 tmp
 drwxr-xr-x     0     0      4096 16-Feb-2017 11:48 usr
 drwxr-xr-x     0     0      4096 16-Feb-2017 21:35 var
 lrwxrwxrwx     0     0         7  3-May-2016 13:48 bin
 lrwxrwxrwx     0     0         7  3-May-2016 13:48 lib
 lrwxrwxrwx     0     0         9  3-May-2016 13:48 lib64
 lrwxrwxrwx     0     0         8  3-May-2016 13:48 sbin
```

如果您看到 `Copy done! 1 ok, 0 failed` 说明复制成功。如下图所示。

```
   * Linux                          0  32 33   652 180 40   10483712
Directory /
Copy done! 1 ok, 0 failed
 drwxr-xr-x     0     0      4096 21-Feb-2017 11:57 .
 drwxr-xr-x     0     0      4096 21-Feb-2017 11:57 ..
 drwx------     0     0     16384 21-Feb-2017 11:56 lost+found
 -rw-r--r--     0     0      1701 21-Feb-2017 11:57 install_edsd.sh
 -rw-r--r--     0     0      5848 21-Feb-2017 11:57 install.sh
>-rw-r--r--     0     0     12136 21-Feb-2017 11:57 ip.gz
 -rw-r--r--     0     0         0 21-Feb-2017 11:57 test
 drwxr-xr-x     0     0      4096 21-Feb-2017 11:57 123.sh
 drwxr-xr-x     0     0      4096 21-Feb-2017 11:57 configclient
 drwxr-xr-x     0     0      4096 21-Feb-2017 11:57 data
 drwxr-xr-x     0     0      4096 21-Feb-2017 11:57 diamond
 drwxr-xr-x     0     0      4096 21-Feb-2017 11:57 logs
```

5. 切换到 */home* 目录查看。如果您能看到文件，说明文件恢复成功。

```
[root@Aliyun /]# ls /home/
admin  ip.gz
[root@Aliyun /]#
```

常见误区与最佳实践

数据是用户的核心资产，很多用户在ECS上构建网站、自建数据库(MYSQL/MongoDB/Redis)。数据丢失会给用户的业务带来巨大的风险。如下是在数据安全方面的常见误区和最佳实践。

- 常见误区

  阿里云的底层存储基于三副本，因此有些用户认为操作系统内数据没有任何丢失风险。实际上这是误解。底层存储的三副本提供对数据磁盘的物理层保护，但是，如果系统内部使用云盘逻辑上出现问题，比如中毒、误删数据、文件系统损坏等情况，还是可能出现数据丢失。此时，您需要通过快照、异地备份等相关技术最大保证数据的安全性。

- 最佳实践

  数据盘分区恢复以及数据恢复是处理数据丢失问题最后的一道防线，但未必一定能够恢复数据。强烈建议您参考如下最佳实践，通过对数据创建快照（自动或手动）以及各类备份方案，最大程度地保证数据的安全性。

  — 启用自动快照

  根据实际业务，对系统盘、数据盘创建自动快照。注意，在更换系统盘、实例到期后或手动释放磁盘时，自动快照可能会被释放。

  您可以在ECS控制台上通过 **修改磁盘属性** 选择 **自动快照随磁盘释放**。如果想保留自动快照，您可以手动去掉该选项。

  详情请参考：*ECS云服务器自动快照FAQ*。

  — 创建手动快照

  在做下列重要或有风险的操作前，请手动为磁盘创建快照。例如：

  ■ 系统升级内核

  ■ 应用升级变更

  ■ 磁盘数据恢复

  在恢复磁盘时，一定要先对磁盘创建快照，快照完成后做相应的操作。

  — OSS、线下、异地备份

  您可酌情使用OSS、线下、异地等方式备份重要数据。

2.4 Windows实例中数据恢复

在处理磁盘相关问题时，您可能会碰到操作系统中数据盘分区丢失的情况。本文介绍了Windows系统下常见的数据盘分区丢失的问题以及对应的处理方法，同时提供了使用云盘的常见误区以及最佳实践，避免可能的数据丢失风险。

前提条件

在修复数据前，您必须先对丢失分区的数据盘创建快照，在快照创建完成后再尝试修复。如果在修复过程中出现问题，您可以通过快照回滚将数据盘还原到修复之前的状态。

工具说明

在Windows实例里，您可以选择以下任一种工具恢复数据盘数据：

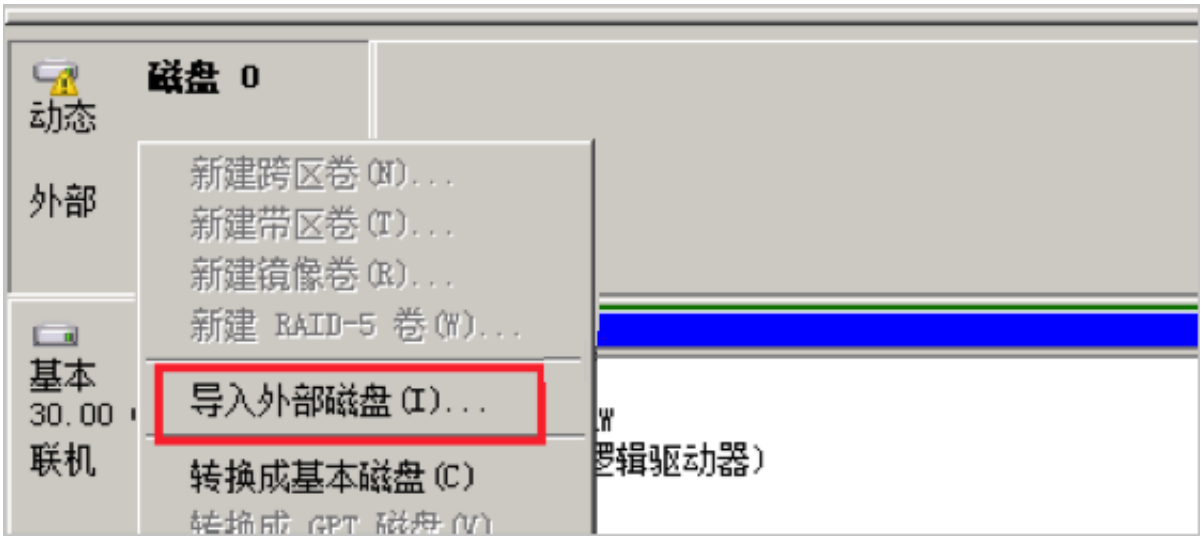
- 磁盘管理：Windows系统自带工具，主要用于分区格式化数据盘等。
- 数据恢复软件：一般是商业软件，您可以去相应的官网下载使用。主要作用是文件系统异常恢复数据。

磁盘显示为“外部”，无法显示分区

在Windows系统中，您在 磁盘管理器 中看到磁盘显示为 外部，而且不显示分区情况，如下图所示。

此时，按以下方式处理：

在 外部 磁盘处，右键单击右边的空白处，选择 导入外部磁盘，再单击 确定。

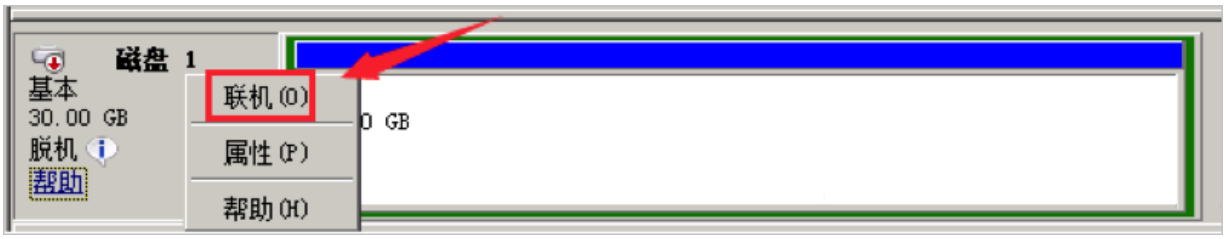


磁盘显示为“脱机”，无法显示分区

在Windows系统中，您在 磁盘管理器 中看到磁盘显示为 脱机，而且不显示分区情况，如下图所示。

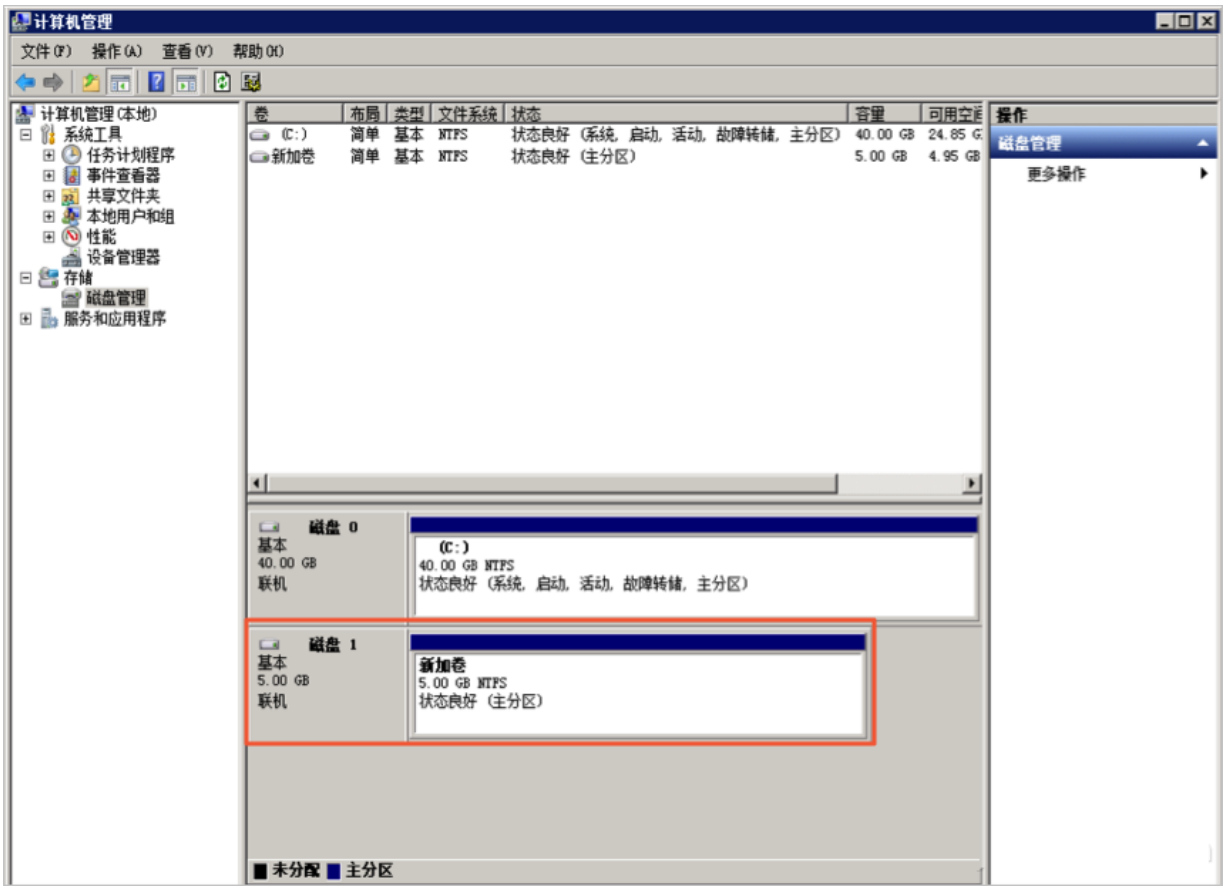
此时，按以下方式处理：

在 脱机 磁盘处，右键单击磁盘名称（如上图中的 磁盘**1**）周边的空白区，在弹出菜单中，选择 联机，再单击 确定。



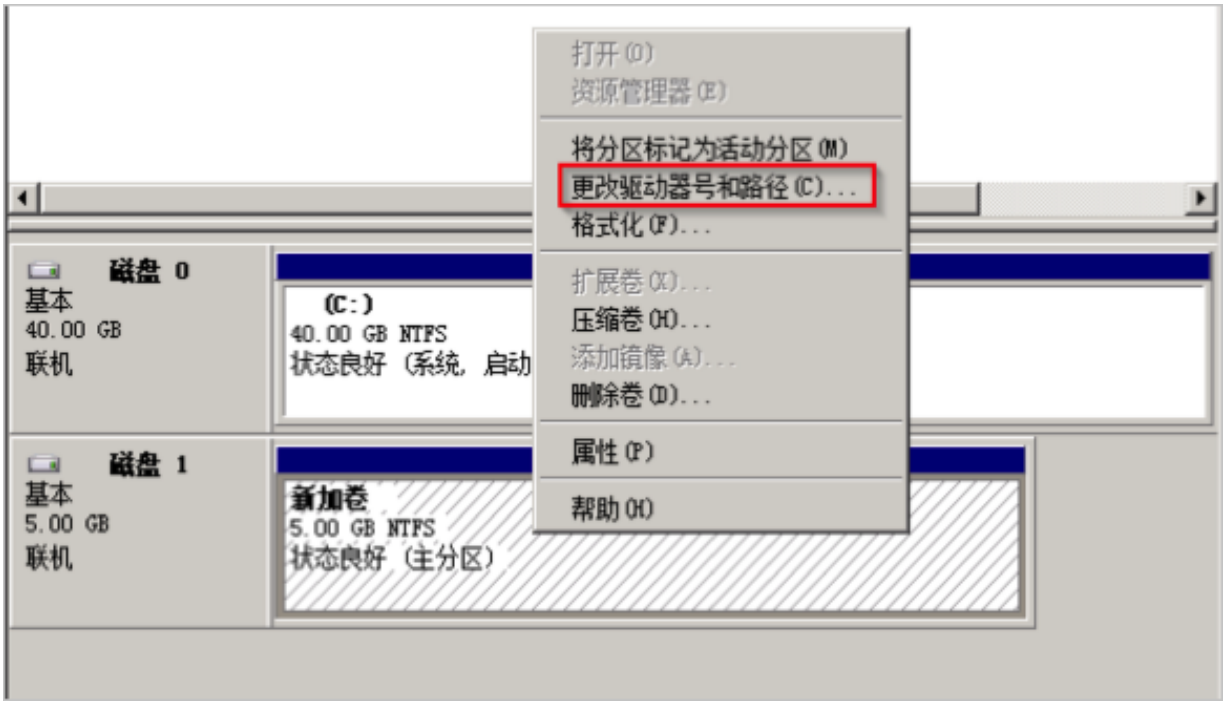
未分配盘符，无法显示分区

在Windows系统中，您在 磁盘管理器 中能看到数据盘的信息，但数据盘未分配盘符，如下图所示。



此时，按以下方式处理：

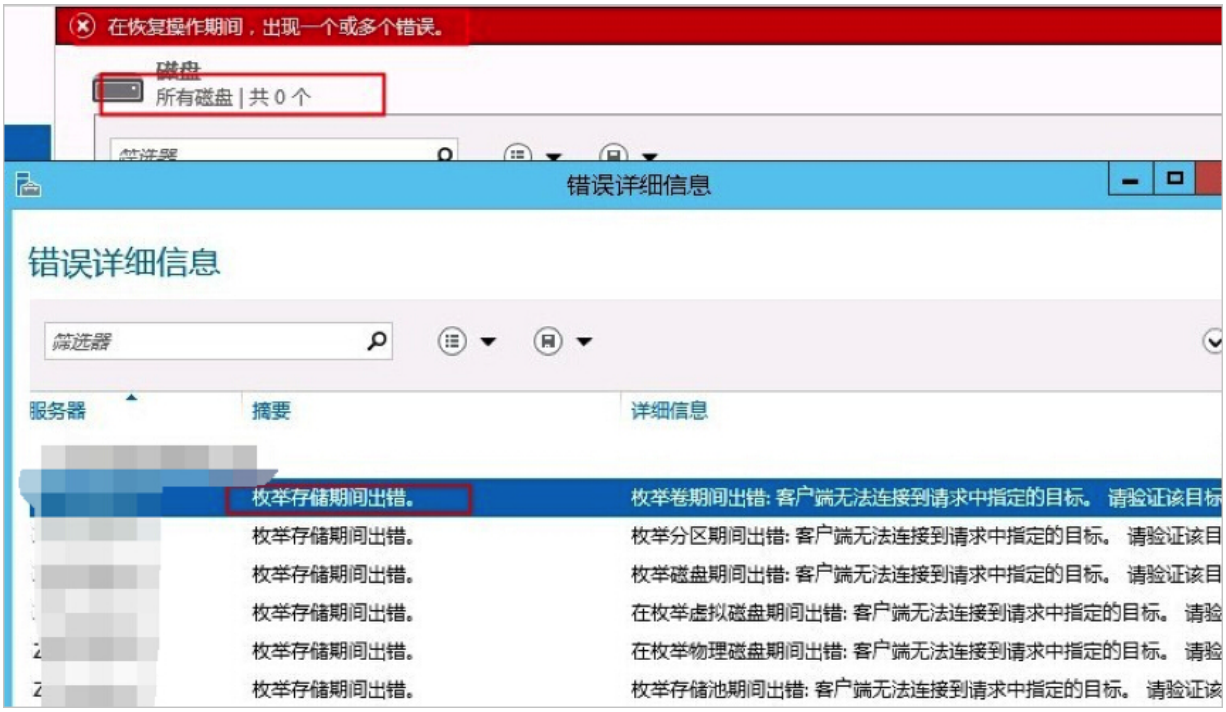
右键单击磁盘（如上图所示的 磁盘**1**）的主分区，在弹出菜单中，选择 更改驱动器号和路径，并按提示完成操作。



在磁盘管理器无法查看数据盘，报错“枚举存储期间出错”

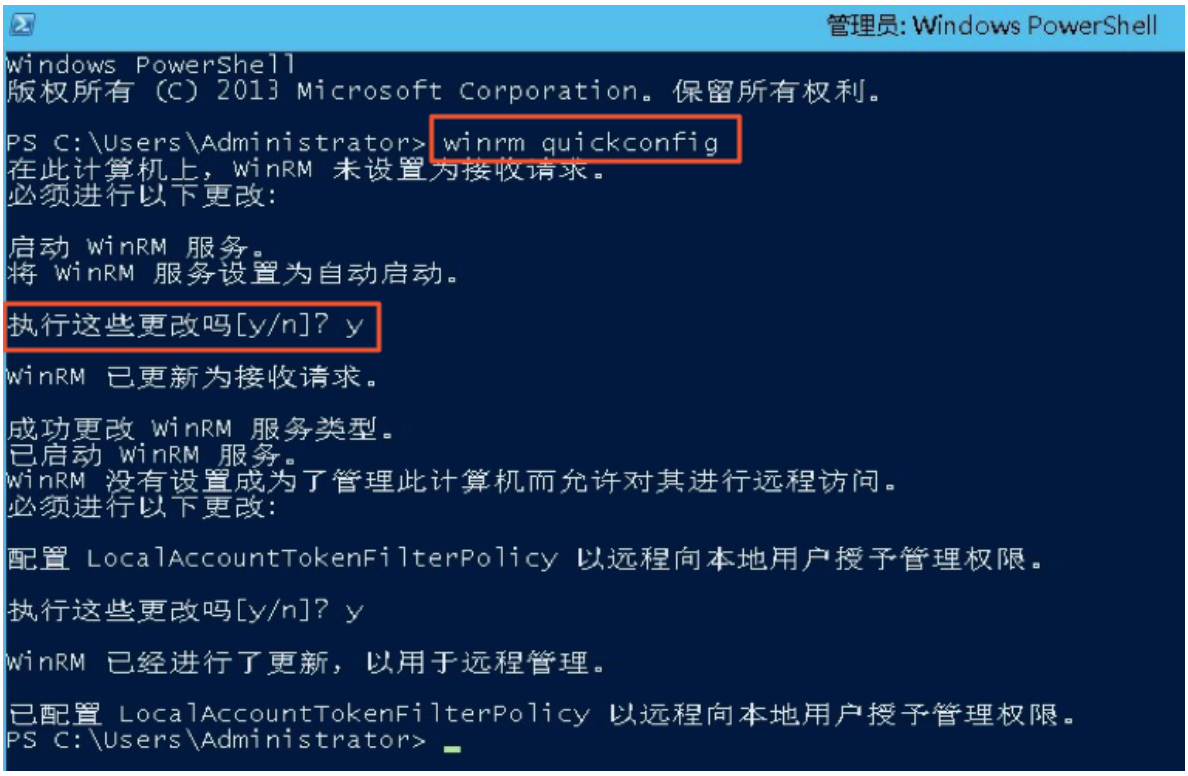
在Windows系统中，您在 磁盘管理器 里无法查看数据盘。系统日志里报错“枚举存储期间出错”，如下图所示。

> **说明：**
> 操作系统的版本不同，报错内容也可能是“枚举卷期间出错”。



此时，按以下步骤处理：

1. 启动Windows PowerShell。
2. 运行命令 **`winrm quickconfig`** 进行修复。当界面上询问“执行这些更改吗[y/n]?”时，输入 y 确认执行。



修复完成后，再打开 磁盘管理器，一般数据盘已经能正常显示。

数据盘变成RAW格式

在某些特殊情况下，您可能会发现Windows下磁盘变为RAW格式。

磁盘显示为RAW格式是因为Windows无法识别磁盘上的文件系统。一般是因为记录文件系统类型或者位置的信息丢失或者损坏，比如partition table或者boot sector。以下列出了一些比较常见的原因：

- 外接硬盘发生这种问题通常是因为没有使用 **Safely remove hardware** 选项断开磁盘。
- 意外断电导致的磁盘问题。

- 硬件层故障也可能导致磁盘分区信息丢失。
- 底层与磁盘相关的驱动或应用，例如您使用的diskprobe工具就可以直接修改磁盘的表结构。
- 计算机病毒。

您可以参考微软官方的 *Dskprobe Overview* 文档修复磁盘。

此外，Windows下有大量免费或商业的数据恢复软件可用于找回丢失的数据。例如，您可以尝试使用Disk Genius工具扫描，来尝试恢复相应的文件。

常见误区和最佳实践

数据是用户的核心资产，很多用户在ECS上构建网站、自建数据库(MYSQL/MongoDB/Redis)。如果出现数据丢失，会给用户的业务带来巨大的风险。如下是在数据安全方面的常见误区和最佳实践。

- **常见误区**

  阿里云的底层存储基于 三副本，因此有些用户认为操作系统内数据没有任何丢失风险。实际上这是误解。底层存储的三副本提供对数据磁盘的物理层保护，但是，如果系统内部使用云盘逻辑上出现问题，比如中毒、误删数据、文件系统损坏等情况，还是可能出现数据丢失。此时，您需要通过快照、异地备份等相关技术最大保证数据的安全性。

- **最佳实践**

  数据盘分区恢复以及数据恢复是处理数据丢失问题最后的一道防线，但未必一定能够恢复数据。强烈建议您参考如下最佳实践，通过对数据创建快照（自动或手动）以及各类备份方案，最大程度地保证数据的安全性。

  — **启用自动快照**

    根据实际业务，对系统盘、数据盘创建自动快照。注意，在更换系统盘、实例到期后或手动释放磁盘时，自动快照可能会被释放。

    您可以在ECS控制台上通过 **修改磁盘属性** 选择 **自动快照随磁盘释放**。如果想保留自动快照，您可以手动去掉该选项。

    详情请参考：*ECS云服务器自动快照FAQ*。

  — **创建手动快照**

    在做下列重要或有风险的操作前，请手动为磁盘创建快照。例如：

    ■ 系统升级内核

- 应用升级变更
- 磁盘数据恢复

在恢复磁盘时，一定要先对磁盘创建快照，快照完成后做相应的操作。

— OSS、线下、异地备份

您可酌情使用OSS、线下、异地等方式备份重要数据。

3 实例配置

3.1 时间设置：设置Windows实例NTP服务

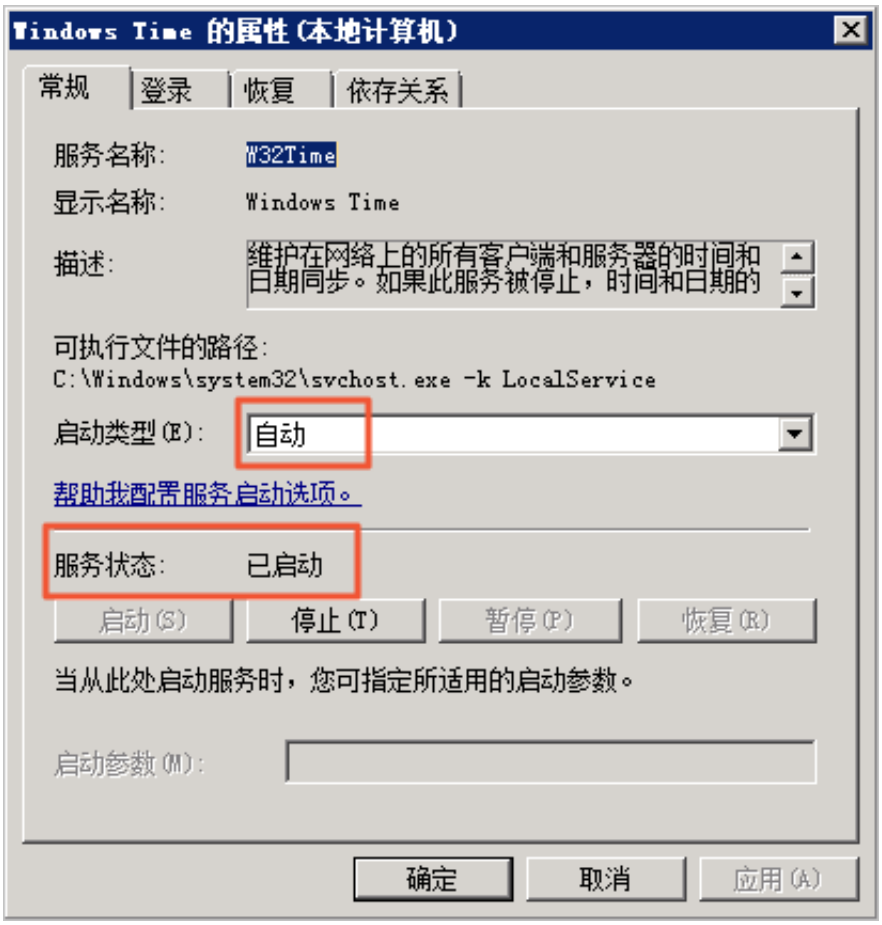
网络时间协议（Network Time Protocol，NTP）是用来同步网络中各个计算机的时间的协议。一些对时间极度敏感的应用（例如，通信行业的应用），如果不同机器时间不一致，就可能导致读取到不同的值。您可以使用NTP服务同步网络中所有服务器的时钟。目前，所有地域的阿里云ECS实例的默认时区为CST（China Standard Time），您可以根据自己的业务需求并参照本文为ECS实例设置或者修改时区。

本文以Windows Server 2008 R2企业版64位为例，说明如何使用NTP服务同步Windows实例的时间。您也可以使用命令完成本文描述的任务，具体操作说明，请参考 开启*Windows*实例*NTP*服务。

Windows Server操作系统默认开启Windows Time服务。为了保证NTP服务配置成功后能正常同步时间，实例中必须开启NTP服务。按以下步骤检查并开启NTP服务：

1. 远程连接*Windows*实例。选择 开始 > 所有程序 > 附件 > 运行，打开 运行 对话框，并运行命令 `services.msc`。
2. 在 服务 窗口，找到并双击 **Windows Time** 服务。
3. 在 **Windows Time的属性(本地计算机)** 对话框中，执行以下操作：
   a. 将 启动类型 设置为 自动。
   b. 确认 服务状态 为 已启动。如果不是，单击 启动。

   完成设置后，单击 应用，并单击 确定。



修改默认**NTP**服务器地址

Windows Server操作系统默认都配置微软默认的NTP服务器（time.windows.com），但是因为网络的原因可能经常同步出错。使用阿里云ECS实例时，您可以将默认的NTP服务器更换成阿里云提供的内网NTP服务器。具体信息，请参考 时间配置*#NTP*服务器与其他基础服务。按以下步骤修改默认的NTP服务器地址：

1. 远程连接*Windows*实例。
2. 在任务栏的通知区域，单击日期和时间，并单击 更改日期和时间设置。

3. 在 日期和时间 对话框里，单击 **Internet** 时间 选项卡，并单击 更改设置。

4. 在 **Internet 时间设置** 对话框里，选择 **与Internet时间服务器同步**，填写一个阿里云内网NTP服务器地址（详细列表请参考 时间配置#NTP服务器与其他基础服务），并单击 **立即更新**。

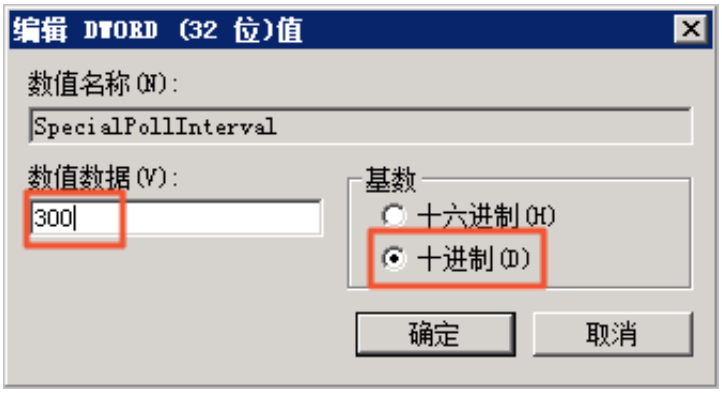
界面会提示是否同步成功。

修改NTP同步的间隔

NTP同步的间隔默认是5分钟。按以下步骤修改NTP同步时间间隔：

1. 远程连接Windows实例。
2. 选择 **开始** > **所有程序** > **附件** > **运行**，打开 **运行** 对话框，并运行命令 `regedit`。
3. 在 **注册表编辑器** 的左侧目录树中，找到 *HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/services/W32Time/TimeProviders/NtpClient*，并双击SpecialPollInterval键值。

4. 在 编辑 **DWORD (32** 位) 值 对话框中，在 基数 栏里选择 十进制，并按需要填写 数值数据。填入的数值即是您需要的同步时间间隔。单位为秒。



3.2 ECS实例数据传输的实现方式

在信息化高速发展的今天，服务器每天都会与其它单机交换大量文件数据，文件传输对大家来说是家常便饭。因此，其重要性就不言而喻了。文件传输方式各有不同，选择一款合适自己的文件传输工具，在工作中能起到事半功倍的效果。节省资源、方便传输、提升工作效率、加密保护等等。因此，很多文件传输工具应运而生，例如：NC、FTP、SCP、NFS、SAMBA、RSYNC/SERVERSYNC等等，每种方式都有自己的特点。本文将首先简单介绍一下文件传输的基本原理，然后，详细介绍类Unix/Linux、Windows平台上常用文件传输方式，并针对它们各自的特点进行比较，让读者对文件传输方式有比较详尽地了解，从而能够根据不同的需要选择合适的文件传输方式。

文件传输原理

文件传输是信息传输的一种形式，它是在数据源和数据宿之间传送文件数据的过程，也称文件数据通信。操作系统把文件数据提取到内存中做暂存，再复制到目的地，加密就是在文件外加了一个壳，文件本身还是一个整体，复制只是把这个整体转移到其它地方，不需要解密，只有打开压缩包时才需解密。一个大文件作为一个数据整体，是不可能瞬间从一台主机转移到其它的主机，传输是一个持续的过程，但不是把文件分割了，因此，如果在传输的过程中意外中断，目标路径中是不会有传输的文件，另外，如果传输的是多个文件，那么，这些文件是按顺序分别传输，如果中间中断，则正在传输的文件会传输失败，但是，之前已经传完的文件传输成功（如果传输的是文件压缩包，那么，不管里面有几个文件，它本身被视为一个文件）。

通常我们看到的 NC、FTP、SCP、NFS 等等，都是可以用来传输文件数据的工具，下面我们将详细介绍主要文件传输工具的特点以及用法。

NETCAT

在网络工具中有“瑞士军刀”的美誉，它功能强大，作为网络工具的同时，它传输文件的能力也不容小觑。

常用参数

参数	说明
-g <网关>	设置路由器跃程通信网关，最多可设置8个
-G <指向器数目>	设置来源路由指向器，其数值为4的倍数
-i <延迟秒数>	设置时间间隔，以便传送信息及扫描通信端口
-l	使用监听模式，管控传入的资料
-o <输出文件>	指定文件名称，把往来传输的数据以16进制字码倾倒成该文件保存
-p <通信端口>	设置本地主机使用的通信端口
-r	指定本地与远端主机的通信端口
-u	使用UDP传输协议
-v	显示指令执行过程
-w <超时秒数>	设置等待连线的时间
-z	使用0输入/输出模式，只在扫描通信端口时使用
-n	直接使用IP地址，而不通过域名服务器

用法举例

1.端口扫描21-24(以IP192.168.2.34为例)。

```
nc -v -w 2 192.168.2.34 -z 21-24
```

返回示例：

```
nc: connect to 192.168.2.34 port 21 (tcp) failed: Connection refused
Connection to 192.168.2.34 22 port [tcp/ssh] succeeded!
nc: connect to 192.168.2.34 port 23 (tcp) failed: Connection refused
nc: connect to 192.168.2.34 port 24 (tcp) failed: Connection refused
```

2. 从192.168.2.33拷贝文件到192.168.2.34。

- 在192.168.2.34上：`nc-l 1234 > test.txt`
- 在192.168.2.33上：`nc192.168.2.34 < test.txt`

3.用nc命令操作memcached。

- 存储数据：`printf “set key 0 10 6rnresultrn” |nc 192.168.2.34 11211`
- 获取数据：`printf “get keyrn” |nc 192.168.2.34 11211`
- 删除数据：`printf “delete keyrn” |nc 192.168.2.34 11211`
- 查看状态：`printf “statsrn” |nc 192.168.2.34 11211`
- 模拟top命令查看状态：`watch “echo stats” |nc 192.168.2.34 11211`
- 清空缓存：

```
printf “flush_allrn” |nc 192.168.2.34 11211        #谨慎操作，清空了缓存就没了
```

SCP 安全拷贝

SCP（ Secure Copy）命令的用法和 RCP 命令格式非常类似，区别就是 SCP 提供更安全保障，SCP 在需要进行验证时会要求你输入密码或口令，一般推荐使用 SCP 命令，因为它比 RCP 更安全。SCP 命令使用 SSH 来传输数据，并使用与 SSH 相同的认证模式，提供同样的安全保障，SSH 是目前较可靠得，为远程登录会话和其他网络服务提供安全性的协议，利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。SCP 是基于 SSH 的应用，所以进行数据传输的机器上必须支持 SSH 服务。

特点

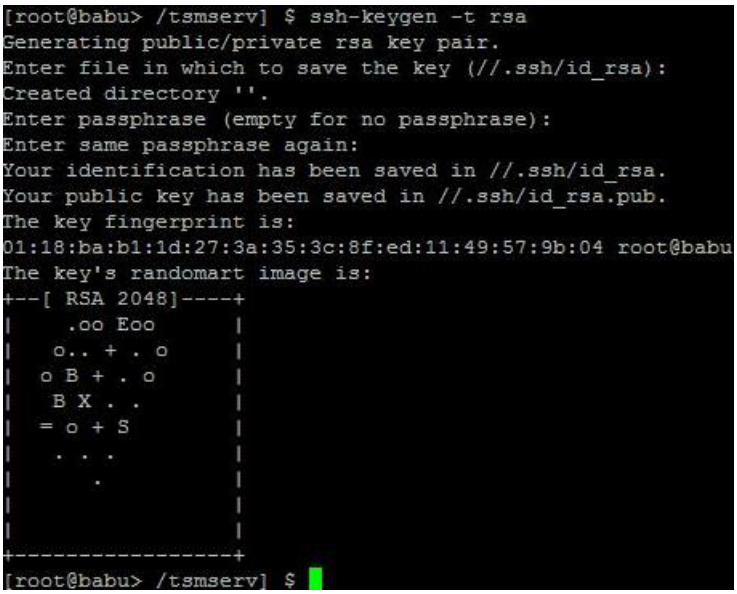
SCP 类似于RCP, 它能够保留一个特定文件系统上的文件属性，能够保留文件属性或者需要递归的拷贝子目录。

SCP它具备更好文件传输保密性。与此同时，付出的代价就是文件传输时需要输入密码而且涉及到 SSH 的一些配置问题，这些都影响其使用的方便性，对于有特定需求的用户，是比较合适的传输工具。

常用示例

使用 SCP 命令，需要输入密码，如果不想每次都输入，可以通过配置 SSH，这样在两台机器间拷贝文件时不需要每次都输入用户名和密码：

生成 RSA 类型的密钥：



上述命令生成 RSA 类型的密钥。在提示密钥的保存路径和密码时，可以直接回车使用默认路径和空密码。这样，生成的公共密钥保存/.ssh/id_rsa.pub，私有密钥保存在 /.ssh/id_rsa 。然后把这个密钥对中的公共密钥的内容复制到要访问的机器上的 /.ssh/authorized_keys 文件中。这样，下次再访问那台机器时，就不用输入密码了。

在两台Linux主机间复制文件

命令基本格式：

```
scp [可选参数] file_source file_target
```

从本地复制到远程（如下四种方式）：

```
scp local_file remote_username@remote_ip:remote_folder
scp local_file remote_username@remote_ip:remote_file
scp local_file remote_ip:remote_folder
scp local_file remote_ip:remote_file
```

说明：

第1,2个指定了用户名，命令执行后需要再输入密码，第1个仅指定了远程的目录，文件名字不变，第2个指定了文件名。

第3,4个没有指定用户名，命令执行后需要输入用户名和密码，第3个仅指定了远程的目录，文件名字不变，第4个指定了文件名。

从远程复制到本地：

```
scp root@www.cumt.edu.cn:/home/root/others/music /home/space/music/i.
mp3
```

```
scp -r www.cumt.edu.cn:/home/root/others/ /home/space/music/
```

说明：

从远程复制到本地，只要将从本地复制到远程的命令的后2个参数调换顺序即可

Rsync

Rsync是linux/Unix文件同步和传送工具。用于替代rcp的一个工具，rsync可以通过rsh或ssh使用，也能以daemon模式去运行，在以daemon方式运行时rsync server会开一个873端口，等待客户端去连接。连接时rsync server会检查口令是否相符，若通过口令查核，则可以通过进行文件传输，第一次连通完成时，会把整份文件传输一次，以后则就只需进行增量备份。

安装方式

说明：

可以使用每个发行版本自带的安装包管理器安装。

```
sudo apt-get  install  rsync      #在debian、ubuntu 等在线安装方法；
slackpkg  install  rsync          #Slackware 软件包在线安装；
yum install rsync                 #Fedora、Redhat 等系统安装方法；
```

源码编译安装：

```
wget http://rsync.samba.org/ftp/rsync/src/rsync-3.0.9.tar.gz
tar xf rsync-3.0.9.tar.gz
cd rsync-3.0.9
./configure && make && make install
```

参数介绍：

参数	说明
-v	详细模式输出
-a	归档模式，表示以递归的方式传输文件，并保持所有文件属性不变，相当于使用了组合参数-rlptgoD
-r	对子目录以递归模式处理
-l	保留软链接
-p	保持文件权限
-t	保持文件时间信息
-g	保持文件属组信息
-o	保持文件属主信息

参数	说明
-D	保持设备文件信息
-H	保留硬链结
-S	对稀疏文件进行特殊处理以节省DST的空间
-z	对备份的文件在传输时进行压缩处理

rsync六种不同的工作模式

- 拷贝本地文件，将/home/coremail目录下的文件拷贝到/cmbak目录下。

```
rsync -avSH /home/coremail/ /cmbak/
```

- 拷贝本地机器的内容到远程机器。

```
rsync -av /home/coremail/ 192.168.11.12:/home/coremail/
```

- 拷贝远程机器的内容到本地机器。

```
rsync -av 192.168.11.11:/home/coremail/ /home/coremail/
```

- 贝远程rsync服务器(daemon形式运行rsync)的文件到本地机。

```
rsync -av root@172.16.78.192::www /databack
```

- 拷贝本地机器文件到远程rsync服务器(daemon形式运行rsync)中。当DST路径信息包含”::”分隔符时启动该模式。

```
rsync -av /databack root@172.16.78.192::www
```

- 显示远程机的文件列表。这类似于rsync传输，不过只要在命令中省略掉本地机信息即可。

```
rsync -v rsync://192.168.11.11/data
```

rsync配置文件说明

```
cat/etc/rsyncd.conf                 #内容如下
port = 873                          #端口号
uid = nobody                        #指定当模块传输文件的守护进程UID
gid = nobody                        #指定当模块传输文件的守护进程GID
use chroot = no                     #使用chroot到文件系统中的目录中
max connections = 10                #最大并发连接数
strict modes = yes                  #指定是否检查口令文件的权限
pid file = /usr/local/rsyncd/rsyncd.pid        #指定PID文件
lock file = /usr/local/rsyncd/rsyncd.lock      #指定支持max connection的
锁文件，默认为/var/run/rsyncd.lock
motd file = /usr/local/rsyncd/rsyncd.motd      #定义服务器信息的，自己写
rsyncd.motd 文件内容
```

```
log file = /usr/local/rsyncd/rsync.log        #rsync 服务器的日志
log format = %t %a %m %f %b
syslog facility = local3
timeout = 300
[conf]                                        #自定义模块
path = /usr/local/nginx/conf                  #用来指定要备份的目录
comment = Nginx conf
ignore errors                                 #可以忽略一些IO错误
read only = no                                #设置no，客户端可以上传文件，yes是
只读
write only = no                               #no为客户端可以下载，yes不能下载
hosts allow = 192.168.2.0/24                  #可以连接的IP
hosts deny = *                                #禁止连接的IP
list = false                                  #客户请求时，使用模块列表
uid = root
gid = root
auth users = backup                           #连接用户名，和linux系统用户名无关
系
secrets file = /etc/rsyncd.pass               #验证密码文件
```

3.3 通过读写分离提升数据吞吐性能

一般情况下，对数据库的读和写都在同一个数据库服务器中操作时，业务系统性能会降低。为了提升业务系统性能，优化用户体验，可以通过读写分离来减轻主数据库的负载。本文分别从应用层和系统层来介绍读写分离的实现方法。

应用层实现方法

应用层中直接使用代码实现，在进入Service之前，使用AOP来做出判断，是使用写库还是读库，判断依据可以根据方法名判断，比如说以query、find、get等开头的就走读库，其他的走写库。

优点

1、多数据源切换方便，由程序自动完成。

2、不需要引入中间件。

3、理论上支持任何数据库。

缺点

1、由程序员完成，运维参与不到。

2、不能做到动态增加数据源。

系统层实现方法

方式一：使用 *DRDS*实现读写分离

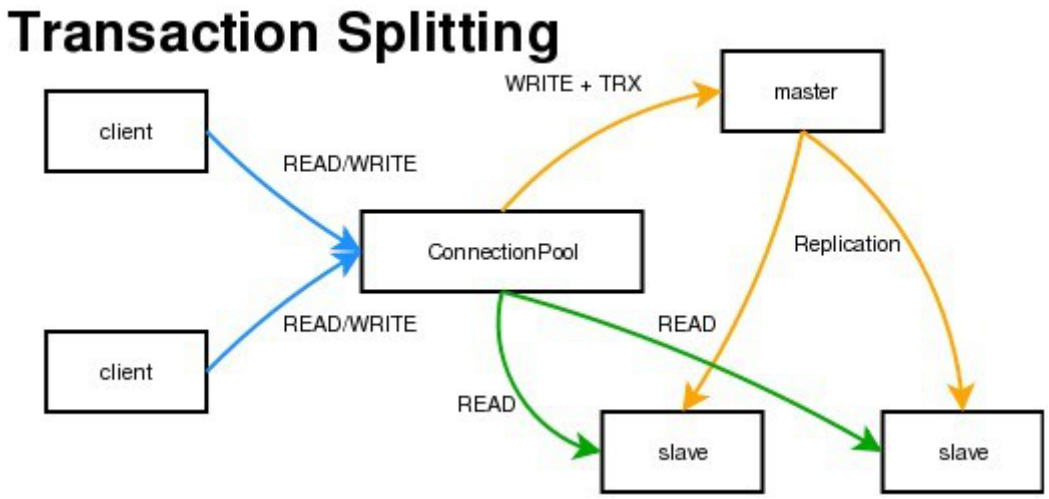
方式二：使用中间件MySQL-proxy实现

本教程使用MySQL-proxy实现读写分离。

MySQL-proxy

MySQL Proxy是一个处于Client端和MySQL server端之间的简单程序，它可以监测、分析或改变它们的通信。它使用灵活，没有限制，常见的用途包括：负载平衡，故障、查询分析，查询过滤和修改等等。

MySQL-proxy原理



MySQL Proxy是一个中间层代理，简单的说，MySQL Proxy就是一个连接池，负责将前台应用的连接请求转发给后台的数据库，并且通过使用lua脚本，可以实现复杂的连接控制和过滤，从而实现读写分离和负载平衡。对于应用来说，MySQL Proxy是完全透明的，应用则只需要连接到MySQL Proxy的监听端口即可。当然，这样proxy机器可能成为单点失效，但完全可以使用多个proxy机器做为冗余，在应用服务器的连接池配置中配置到多个proxy的连接参数即可。

优点：

- 源程序不需要做任何改动就可以实现读写分离。
- 动态添加数据源不需要重启程序。

缺点：

- 序依赖于中间件，会导致切换数据库变得困难。
- 由中间件做了中转代理，性能有所下降。

操作步骤

环境说明：

- 主库IP：121.40.18.26

- 从库IP：101.37.36.20
- MySQL-proxy代理IP：116.62.101.76

前期准备：

- 1、新建3台ECS，并安装mysql。
- 2、搭建主从，必须保证主从数据库数据一致。

主环境

1. 修改mysql配置文件。

```
vim /etc/my.cnf
[mysqld]
server-id=202                    #设置服务器唯一的id，默认是1
log-bin=mysql-bin                # 启用二进制日志
```

从环境

```
[mysqld]
server-id=203
```

2. 重启主从服务器中的MySQL服务。

```
/etc/init.d/mysqld restart
```

3. 在主服务器上建立帐户并授权slave。

```
mysql -uroot -p95c7586783
grant replication slave on *.* to 'syncms'@'填写slave-IP' identified by
 '123456';
flush privileges;
```

4. 查看主数据库状态。

```
mysql> show master status;
```

```
mysql> show master status;
+------------------+----------+--------------+------------------+-------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+------------------+----------+--------------+------------------+-------------------+
| mysql-bin.000005 |      602 |              |                  |                   |
+------------------+----------+--------------+------------------+-------------------+
1 row in set (0.00 sec)
```

5. 配置从数据库。

```
change master to master_host='填写master-IP', master_user='syncms',
     master_password='123456', master_log_file='mysql-bin.000005',
master_log_pos=602;
```

6. 启动slave同步进程并查看状态。

```
start slave;
show slave status\G
```

```
mysql> show slave status\G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 116.62.101.35
                  Master_User: syncms
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000007
          Read_Master_Log_Pos: 154
               Relay_Log_File: iZbp17p8l1ul3oj2nztb4kZ-relay-bin.000003
                Relay_Log_Pos: 367
        Relay_Master_Log_File: mysql-bin.000007
             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:
```

7. 验证主从同步。

主库上操作

```
mysql> create database testproxy;
mysql> create table testproxy.test1(ID int primary key,name char(10)
not null);
mysql> insert into testproxy.test1 values(1,'one');
mysql> insert into testproxy.test1 values(2,'two');
```

```
mysql> select * from testproxy.test1;
```

```
mysql> create database testproxy;
Query OK, 1 row affected (0.01 sec)

mysql> create table testproxy.test1(ID int primary key,name char(10) not null);
Query OK, 0 rows affected (0.07 sec)

mysql> insert into testproxy.test1 values(1,'one');
Query OK, 1 row affected (0.02 sec)

mysql> insert into testproxy.test1 values(2,'two');
Query OK, 1 row affected (0.03 sec)

mysql> select * from testproxy.test1;
+----+------+
| ID | name |
+----+------+
|  1 | one  |
|  2 | two  |
+----+------+
2 rows in set (0.01 sec)
```

从库操作

从库中查找testproxy.test1表的数据，与主库一致，主从同步成功

```
select * from testproxy.test1;
```

```
mysql> select * from testproxy.test1;
+----+------+
| ID | name |
+----+------+
|  1 | one  |
|  2 | two  |
+----+------+
2 rows in set (0.00 sec)
```

读写分离配置

1.安装MySQL-Proxy。

```
wget https://cdn.mysql.com/archives/mysql-proxy/mysql-proxy-0.8.5-
linux-glibc2.3-x86-64bit.tar.gz
mkdir /alidata
tar xvf mysql-proxy-0.8.5-linux-glibc2.3-x86-64bit.tar.gz
mv mysql-proxy-0.8.5-linux-glibc2.3-x86-64bit/  /alidata/mysql-proxy-0
.8.5
```

2.环境变量设置。

```
vim /etc/profile                        #加入以下内容
PATH=$PATH:/alidata/mysql-proxy-0.8.5/bin
```

```
export $PATH
source /etc/profile                    #使变量立即生效
mysql-proxy -V
```

```
[root@iZbp1ajyjlht1reyxsfu4xZ ~]# mysql-proxy -V
mysql-proxy 0.8.5
  chassis: 0.8.5
  glib2: 2.16.6
  libevent: 2.0.21-stable
  LUA: Lua 5.1.4
    package.path: /alidata/mysql-proxy-0.8.5/lib/mysql-proxy/lua/?.lua;
    package.cpath: /alidata/mysql-proxy-0.8.5/lib/mysql-proxy/lua/?.so;
-- modules
  proxy: 0.8.5
```

3.读写分离设置。

```
cd /alidata/mysql-proxy-0.8.5/share/doc/mysql-proxy/
vim  rw-splitting.lua
```

MySQL Proxy会检测客户端连接，当连接没有超过min_idle_connections预设值时，不会进行读写分离默认最小4个(最大8个)以上的客户端连接才会实现读写分离，现改为最小1个最大2个，便于读写分离的测试，生产环境中，可以根据实际情况进行调整。

调整前：

```
-- connection pool
if not proxy.global.config.rwsplit then
        proxy.global.config.rwsplit = {
                min_idle_connections = 4,
                max_idle_connections = 8,

                is_debug = false
        }
end
```

调整后：

```
-- connection pool
if not proxy.global.config.rwsplit then
        proxy.global.config.rwsplit = {
                min_idle_connections = 1,
                max_idle_connections = 2,

                is_debug = true
        }
end
```

4.将lua管理脚本（admin.lua）复制到读写分离脚本(rw-splitting.lua)所在目录。

```
cp /alidata/mysql-proxy-0.8.5/lib/mysql-proxy/lua/admin.lua /alidata/
mysql-proxy-0.8.5/share/doc/mysql-proxy/
```

授权

1.主库中操作授权，因主从同步的原因，从库也会执行。

```
mysql -uroot -p95c7586783
grant all on *.* to 'mysql-proxy'@'填写MySQL Proxy IP' identified by '
123456';
flush privileges;
```

2.开启MySQL-Proxy。

```
mysql-proxy --daemon --log-level=debug --log-file=/var/log/mysql-
proxy.log --plugins=proxy -b 填写master-IP:3306 -r 填写slave-IP:3306
 --proxy-lua-script="/alidata/mysql-proxy-0.8.5/share/doc/mysql-proxy
/rw-splitting.lua" --plugins=admin --admin-username="admin" --admin-
password="admin" --admin-lua-script="/alidata/mysql-proxy-0.8.5/share/
doc/mysql-proxy/admin.lua"
```

3.启动MySQL-Proxy之后，查看端口和相关进程。

```
netstat -tpln
```

```
[root@iZbp1ajyjlht1reyxsfu4xZ ~]# netstat -tpln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      826/sshd
tcp        0      0 0.0.0.0:4040            0.0.0.0:*               LISTEN      22767/mysql-proxy
tcp        0      0 0.0.0.0:4041            0.0.0.0:*               LISTEN      22767/mysql-proxy
```

```
ps -ef | grep mysql
```

```
[root@iZbp1ajyjlht1reyxsfu4xZ ~]# ps -ef | grep mysql
root     22767     1  0 10:59 ?        00:00:00 /alidata/mysql-proxy-0.8.5/libexec/mysql-proxy --
og-level=debug --log-file=/var/log/mysql-proxy.log --plugins=proxy -b 121.40.18.26:3306 -r 101.37.
6 --proxy-lua-script=/alidata/mysql-proxy-0.8.5/share/doc/mysql-proxy/rw-splitting.lua --plugins=a
min-username=admin --admin-password=admin --admin-lua-script=/alidata/mysql-proxy-0.8.5/share/doc/
xy/admin.lua
root     22794 22602  0 11:02 pts/0    00:00:00 grep --color=auto mysql
```

测试读写分离

1.关闭从复制

```
stop slave;
```

2.MySQL-Proxy上操作，登录mysql-proxy后台管理。

```
mysql  -u  admin -padmin -P 4041 -h MySQL-Proxy-IP
select * from backends;                 #查看状态
```

```
MySQL [(none)]> select * from backends;
+-------------+--------------------+---------+------+------+-------------------+
| backend_ndx | address            | state   | type | uuid | connected_clients |
+-------------+--------------------+---------+------+------+-------------------+
|           1 | 121.40.18.26:3306  | unknown | rw   | NULL |                 0 |
|           2 | 101.37.36.20:3306  | unknown | ro   | NULL |                 0 |
+-------------+--------------------+---------+------+------+-------------------+
2 rows in set (0.00 sec)
```

第一次连接，会连接到主库上。

```
mysql -umysql-proxy -p123456 -h 116.62.101.76 -P 4040
insert into testproxy.test1 values(3,'three');                #新增一条数据，由于测试需要，关闭了从复制，因此该数据在主库中存在，在从库中不存在
```

```
[root@iZbp1ajyjlht1reyxsfu4xZ ~]# mysql -umysql-proxy -p123456 -h 116.62.101.76 -P 4040
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.17-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> insert into testproxy.test1 values(3,'three');
Query OK, 1 row affected (0.03 sec)

MySQL [(none)]>
```

多开几个连接进行测试，当查询testproxy.test1表的数据显示是从库的数据时，读写分离成功。

```
mysql -umysql-proxy -p123456 -h 116.62.101.76 -P 4040
```

```
select * from testproxy.test1;
```

```
MySQL [(none)]> select * from testproxy.test1
    -> ;
+----+------+
| ID | name |
+----+------+
|  1 | one  |
|  2 | two  |
+----+------+
2 rows in set (0.00 sec)

MySQL [(none)]> insert into testproxy.test1 values(9,'nine')
    -> ;
Query OK, 1 row affected (0.02 sec)

MySQL [(none)]> select * from testproxy.test1
    -> ;
+----+------+
| ID | name |
+----+------+
|  1 | one  |
|  2 | two  |
+----+------+
2 rows in set (0.00 sec)
```

3.4 Windows Server 2012 搭建 AD 域

Active Directory（简称AD，即" 活动目录"的意思），是微软服务的核心组件，其主要优势是实现高效管理，例如批量管理用户、部署应用和更新补丁等。许多微软组件例如 Exchange 和故障转移群集也需要 AD 域环境。本文通过 Windows Server 2012 实例示范如何搭建 AD 域。

名词解释

- Domain Controllers（DC）：域控制器
- Organizational Unit（OU）：组织单位
- Distinguished name（DN）：识别名
- Canonical Name（CN）：正式名称

安装指南

必要条件

- 安装者必须拥有管理员权限。
- 安装分区为NTFS分区。
- 需要支持DNS。

- 需要支持TCP/IP协议，并且需要有固定IP。任何服务器都应该使用固定IP，防止重启实例后IP地址发生变化。本文采用是阿里云VPC网络，手动修改IP会导致IP失效，如果想修改IP，您可以通过控制台修改。

环境

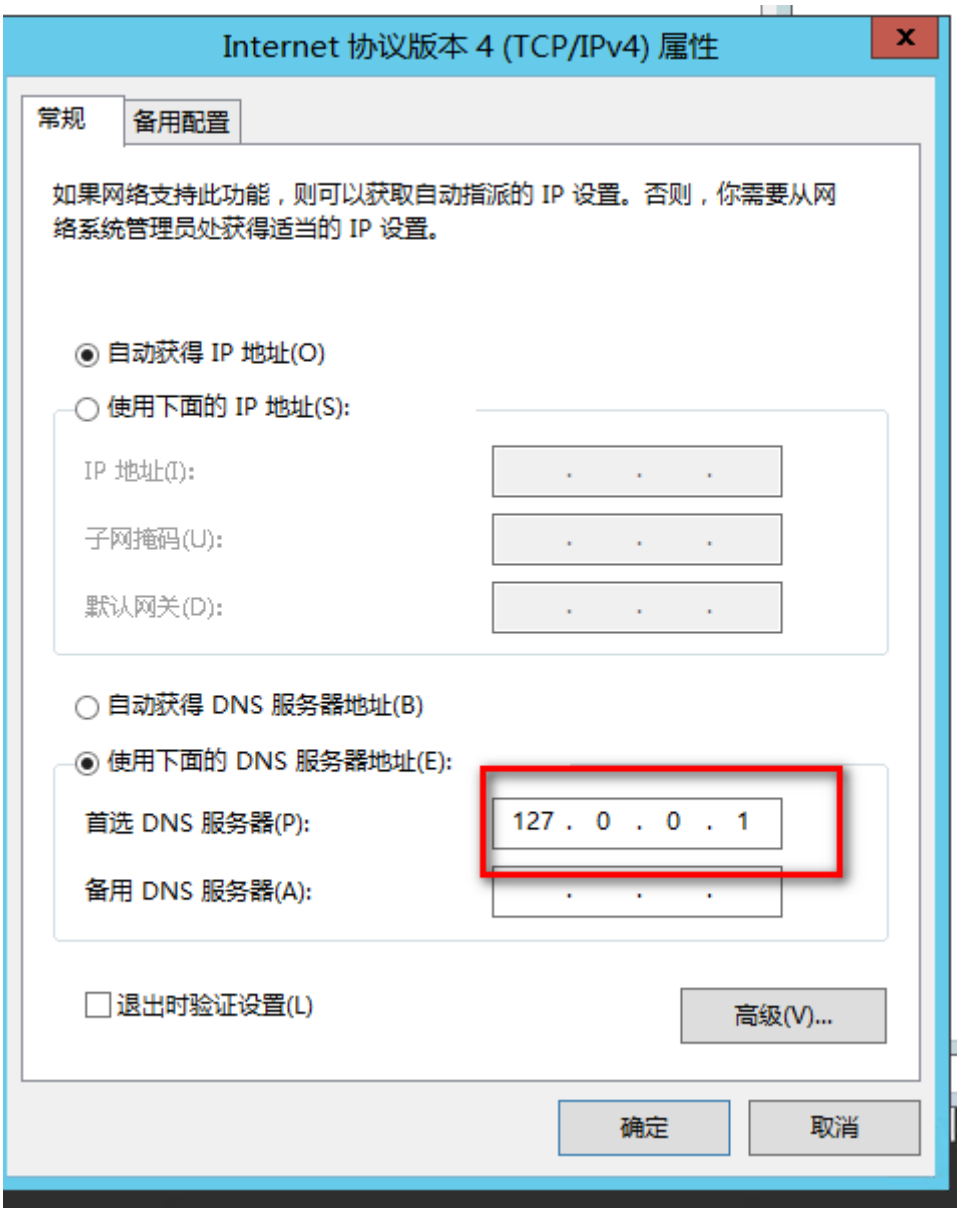
网络采用VPC， 虚拟交换机网段为 192.168.100.0/24，并使用网关。

域名

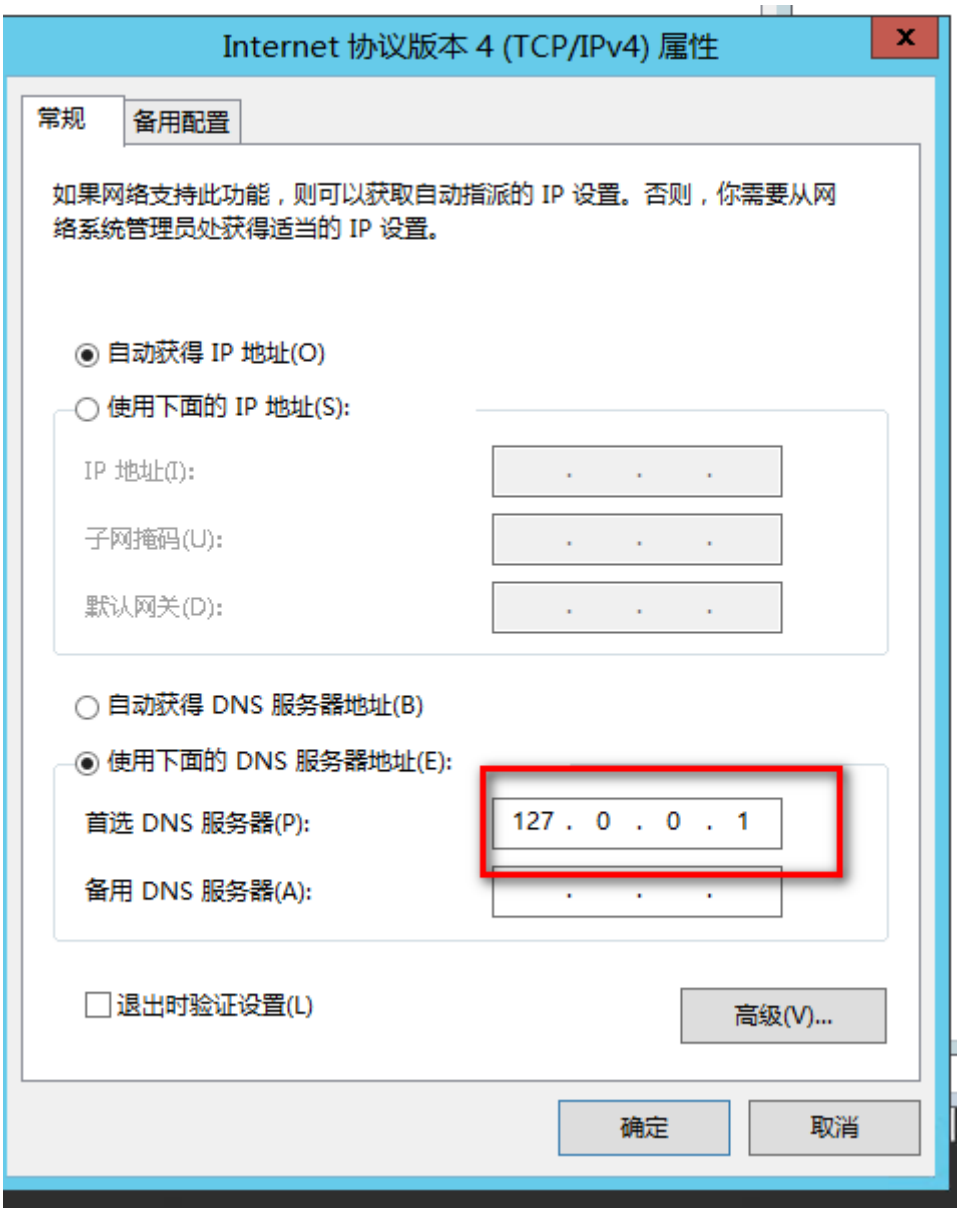
- lyonz.com
- DC：192.168.100.105
- 需要加入域的客户机（Client）IP：192.168.100.106

修改DC 的基本信息

修改DC主机名



修改DC 的DNS（将DNS地址指向自己的IP）



说明：

这里不要手动修改服务器的IP地址（手动修改服务器IP不会生效，也无需担心服务器IP会重启发生改变），如果要修改请在控制台操作。

开始安装

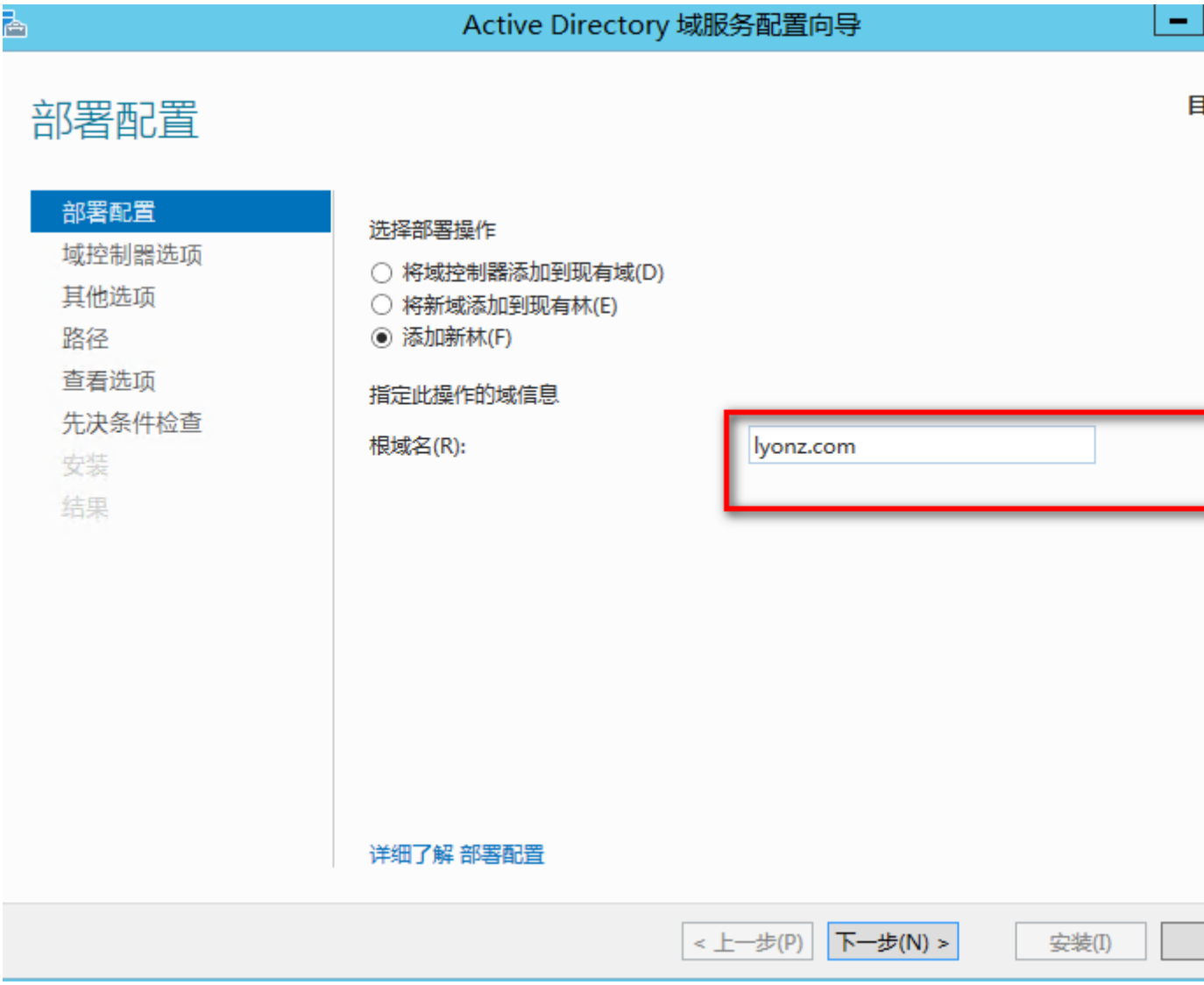

服务器管理器 ▸ 仪表板
仪表板
本地服务器
所有服务器
文件和存储服务
欢迎使用服务器管理器
快速启动(Q)
新增功能(W)
了解详细信息(L)
1 配置此本地服务器
2 添加角色和功能
3 添加要管理的其他服务器
4 创建服务器组
5 将此服务器连接到云服务
角色和服务器组
角色: 1 | 服务器组: 1 | 服务器总数: 1
文件和存储服务 1
本地服务器

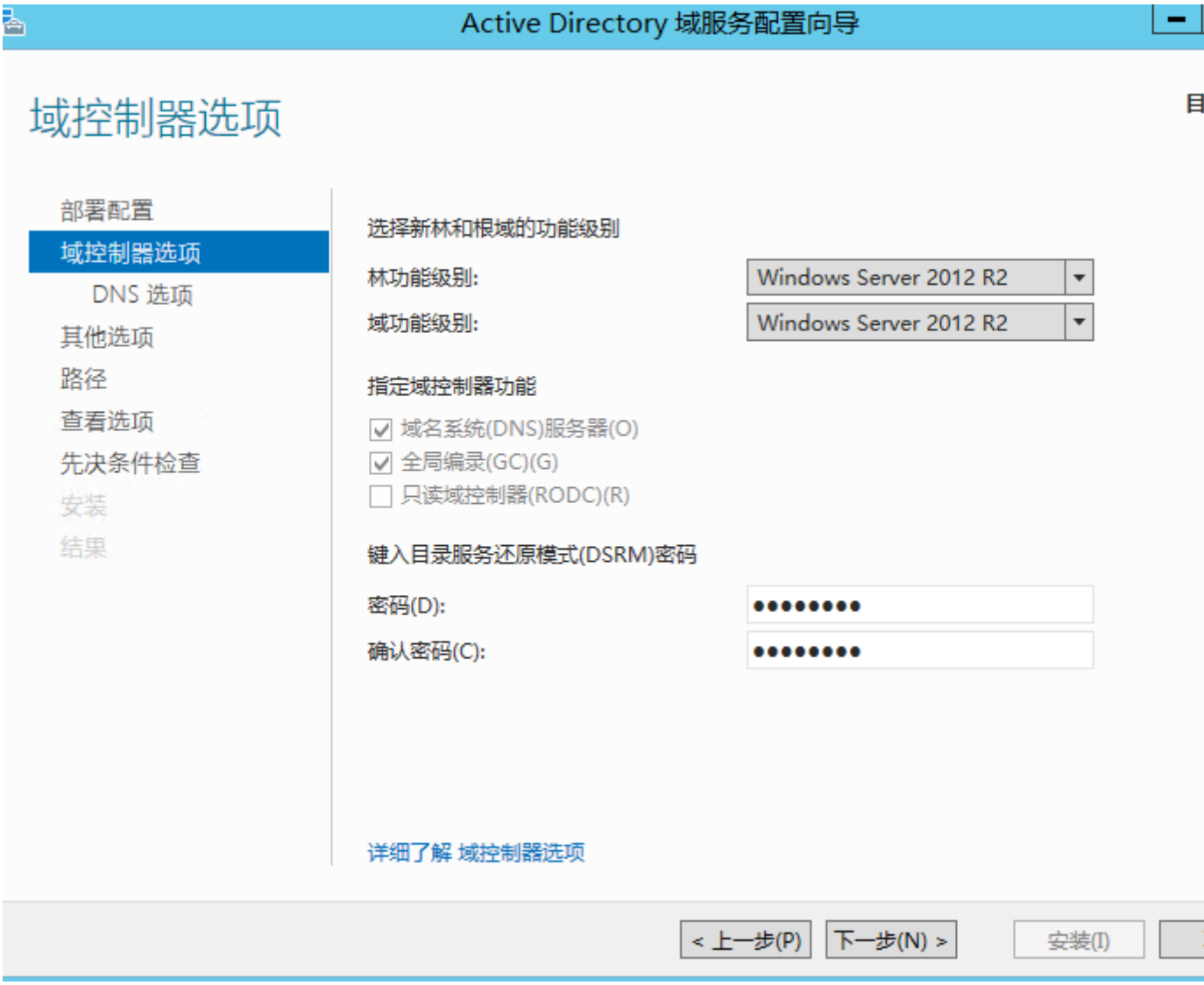

添加角色和功能向导
选择安装类型
开始之前
安装类型
服务器选择
服务器角色
功能
确认
结果
选择安装类型。你可以在正在运行的物理计算机、虚拟机或脱机虚拟硬盘(VHD)上安装角色
基于角色或基于功能的安装
通过添加角色、角色服务和功能来配置单个服务器。
远程桌面服务安装
为虚拟桌面基础结构(VDI)安装所需的角色服务以创建基于虚拟机或基于会话的桌面部署

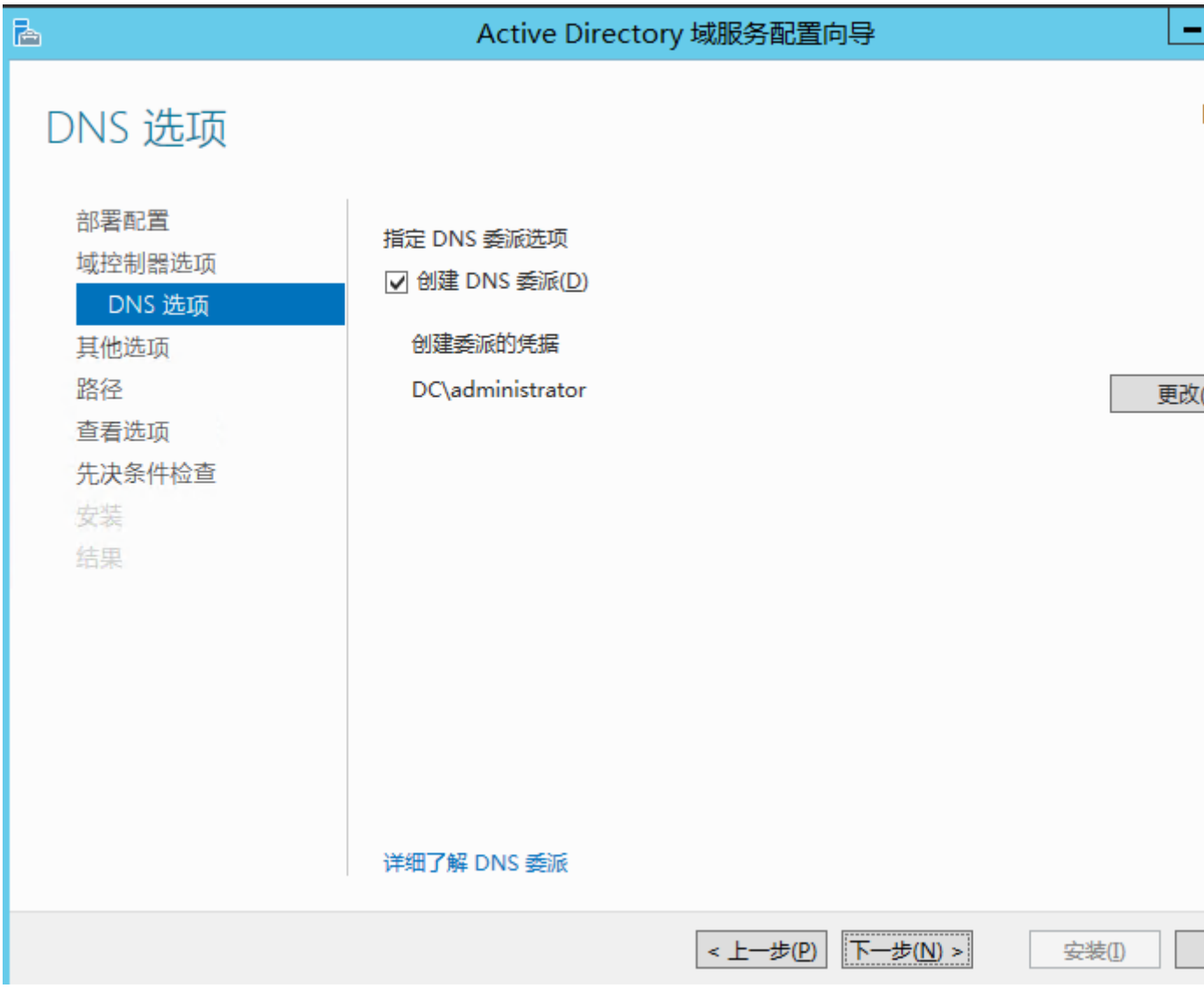

添加角色和功能向导
选择目标服务器
开始之前
安装类型
服务器选择
服务器角色
功能
确认
结果
选择要安装角色和功能的服务器或虚拟硬盘。
从服务器池中选择服务器
选择虚拟硬盘
服务器池
筛选器:
名称
IP 地址
操作系统
iZ3wny1h42ioiuZ
169.254.60.17...
Microsoft Windows Server 2012 R2


添加角色和功能向导
选择服务器角色
开始之前
安装类型
服务器选择
服务器角色
功能
AD DS
DNS 服务器
确认
结果
选择要安装在所选服务器上的一个或多个角色。
角色
描述
Active Directory Federation Services
Active Directory Rights Management Services
Active Directory 轻型目录服务
Active Directory 域服务
Active Directory 证书服务
DHCP 服务器
DNS 服务器
Hyper-V
Web 服务器(IIS)
Windows Server Essentials 体验
Windows Server 更新服务
Windows 部署服务
传真服务器
打印和文件服务
域名系
络提供
Direct
上，DI
果选择
色，你
和 Act
作。
< 上一步(P)
下一步(N) >

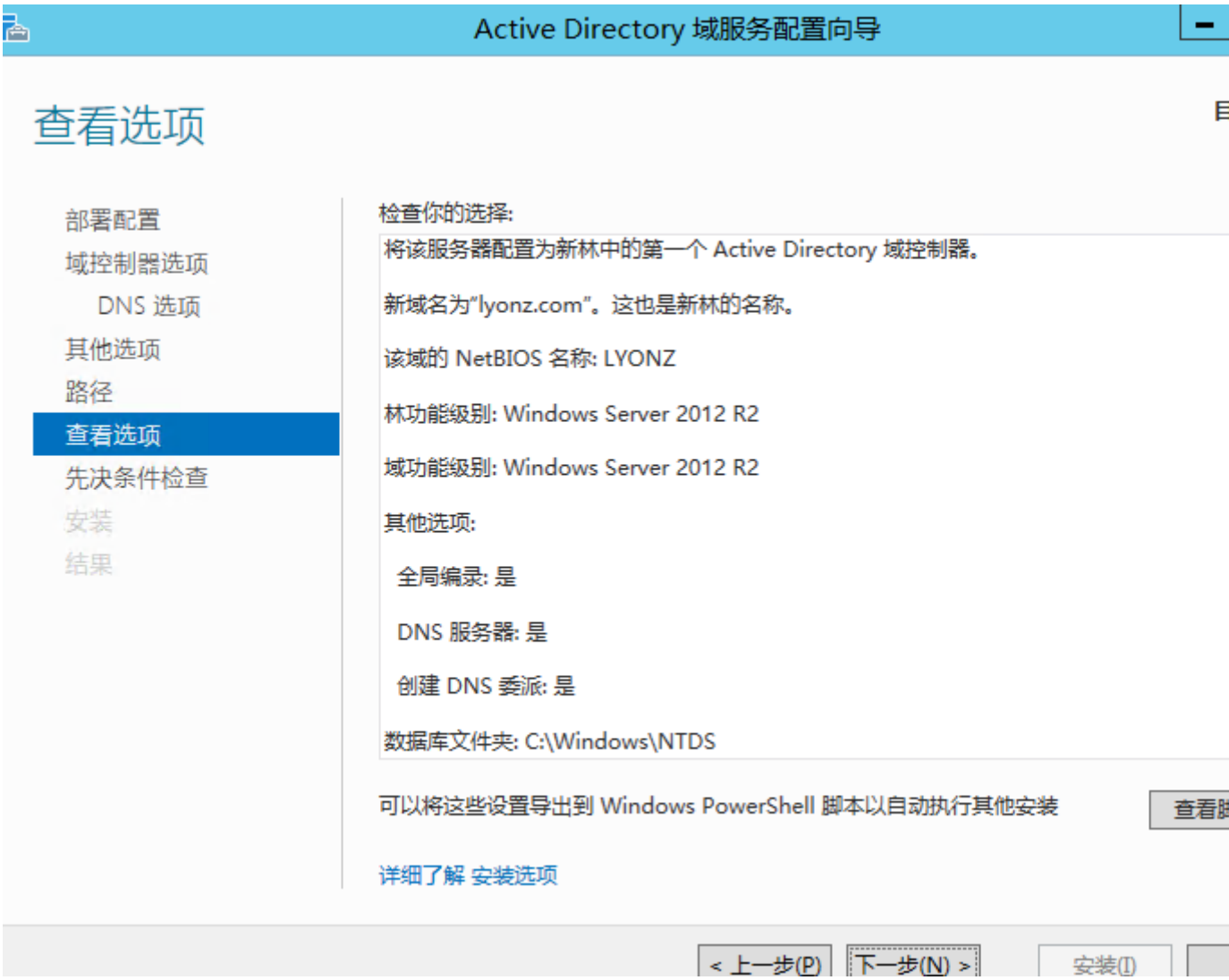

服务器管理器
仪表板
工具(T)
部署后配置
任务
DC 中的 Active Directory 域服务 所需的配置
将此服务器提升为域控制器
任务详细信息
仪表板
本地服务器
所有服务器
AD DS
DNS
文件和存储服务
添加角色和功能
3 添加要管理的其他服务器
4 创建服务器组
5 将此服务器连接到云服务
新增功能(W)
了解详细信息(L)
角色和服务器组

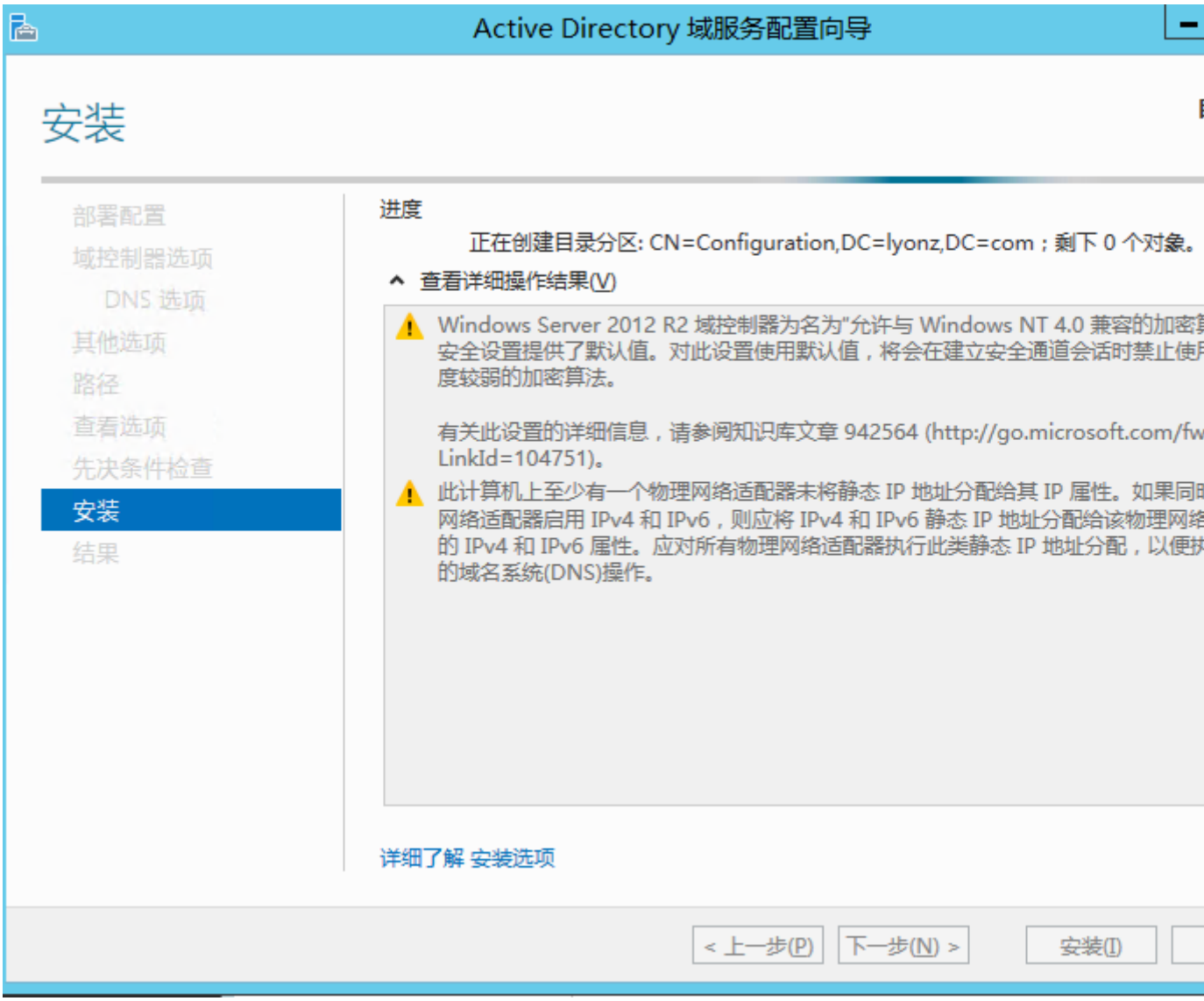

Active Directory 域服务配置向导
部署配置
部署配置
域控制器选项
其他选项
路径
查看选项
先决条件检查
安装
结果
选择部署操作
将域控制器添加到现有域(D)
将新域添加到现有林(E)
添加新林(F)
指定此操作的域信息
根域名(R):
lyonz.com
详细了解 部署配置
< 上一步(P)
下一步(N) >
安装(I)


Active Directory 域服务配置向导
域控制器选项
部署配置
域控制器选项
DNS 选项
其他选项
路径
查看选项
先决条件检查
安装
结果
选择新林和根域的功能级别
林功能级别:
Windows Server 2012 R2
域功能级别:
Windows Server 2012 R2
指定域控制器功能
域名系统(DNS)服务器(O)
全局编录(GC)(G)
只读域控制器(RODC)(R)
键入目录服务还原模式(DSRM)密码
密码(D):
确认密码(C):
详细了解 域控制器选项
< 上一步(P)
下一步(N) >
安装(I)


Active Directory 域服务配置向导
DNS 选项
部署配置
域控制器选项
DNS 选项
其他选项
路径
查看选项
先决条件检查
安装
结果
指定 DNS 委派选项
创建 DNS 委派(D)
创建委派的凭据
DC\administrator
更改(
详细了解 DNS 委派
< 上一步(P)
下一步(N) >
安装(I)


Active Directory 域服务配置向导
其他选项
部署配置
域控制器选项
DNS 选项
其他选项
路径
查看选项
先决条件检查
安装
结果
确保为域分配了 NetBIOS 名称，并在必要时更改该名称
NetBIOS 域名:
LYONZ
详细了解 其他选项


Active Directory 域服务配置向导
查看选项
部署配置
域控制器选项
DNS 选项
其他选项
路径
查看选项
先决条件检查
安装
结果
检查你的选择:
将该服务器配置为新林中的第一个 Active Directory 域控制器。
新域名为"lyonz.com"。这也是新林的名称。
该域的 NetBIOS 名称: LYONZ
林功能级别: Windows Server 2012 R2
域功能级别: Windows Server 2012 R2
其他选项:
全局编录: 是
DNS 服务器: 是
创建 DNS 委派: 是
数据库文件夹: C:\Windows\NTDS
可以将这些设置导出到 Windows PowerShell 脚本以自动执行其他安装
详细了解 安装选项
< 上一步(P)
下一步(N) >
安装(I)


Active Directory 域服务配置向导
安装
部署配置
域控制器选项
DNS 选项
其他选项
路径
查看选项
先决条件检查
安装
结果
进度
正在创建目录分区: CN=Configuration,DC=lyonz,DC=com；剩下 0 个对象。
查看详细操作结果(V)
Windows Server 2012 R2 域控制器为名为"允许与 Windows NT 4.0 兼容的加密
安全设置提供了默认值。对此设置使用默认值，将会在建立安全通道会话时禁止使
度较弱的加密算法。
有关此设置的详细信息，请参阅知识库文章 942564 (http://go.microsoft.com/fw
LinkId=104751)。
此计算机上至少有一个物理网络适配器未将静态 IP 地址分配给其 IP 属性。如果同
网络适配器启用 IPv4 和 IPv6，则应将 IPv4 和 IPv6 静态 IP 地址分配给该物理网络
的 IPv4 和 IPv6 属性。应对所有物理网络适配器执行此类静态 IP 地址分配，以便
的域名系统(DNS)操作。
详细了解 安装选项
< 上一步(P)
下一步(N) >
安装(I)

验证客户端的加入

在云上安装AD和我们线下安装AD步骤其实一样，但客户端加入域的步骤稍有不同，需要先修改客户端的SID，这是因为阿里云ECS Windows Server 2012系统采用的同一个镜像，所以SID是相同的，如果不修改，在加入域的时候会提示SID相同。

修改客户端的SID

Winodws Server 2012：

在 powershell 界面执行如下命令：

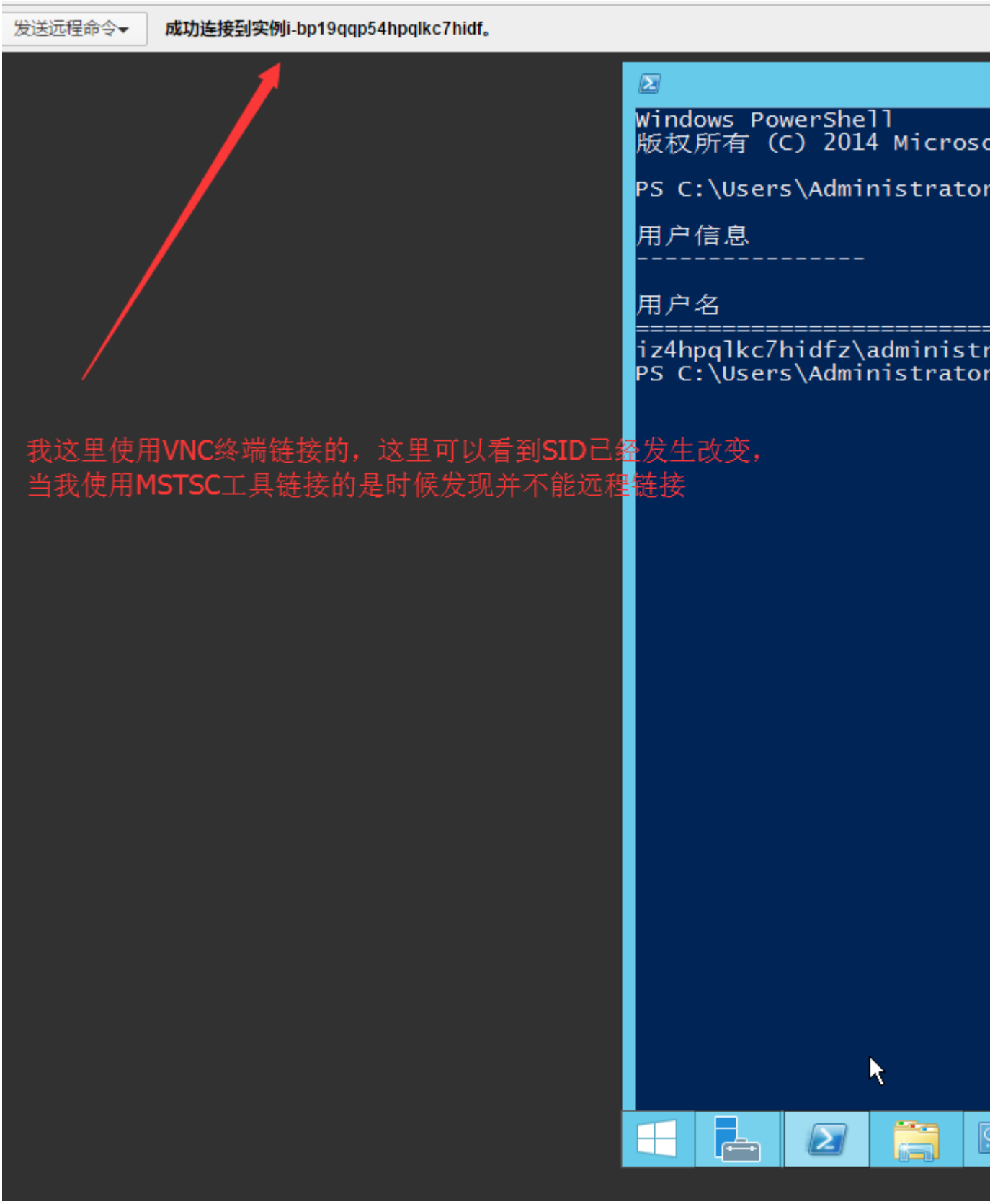
首先切换到脚本存放的路径，

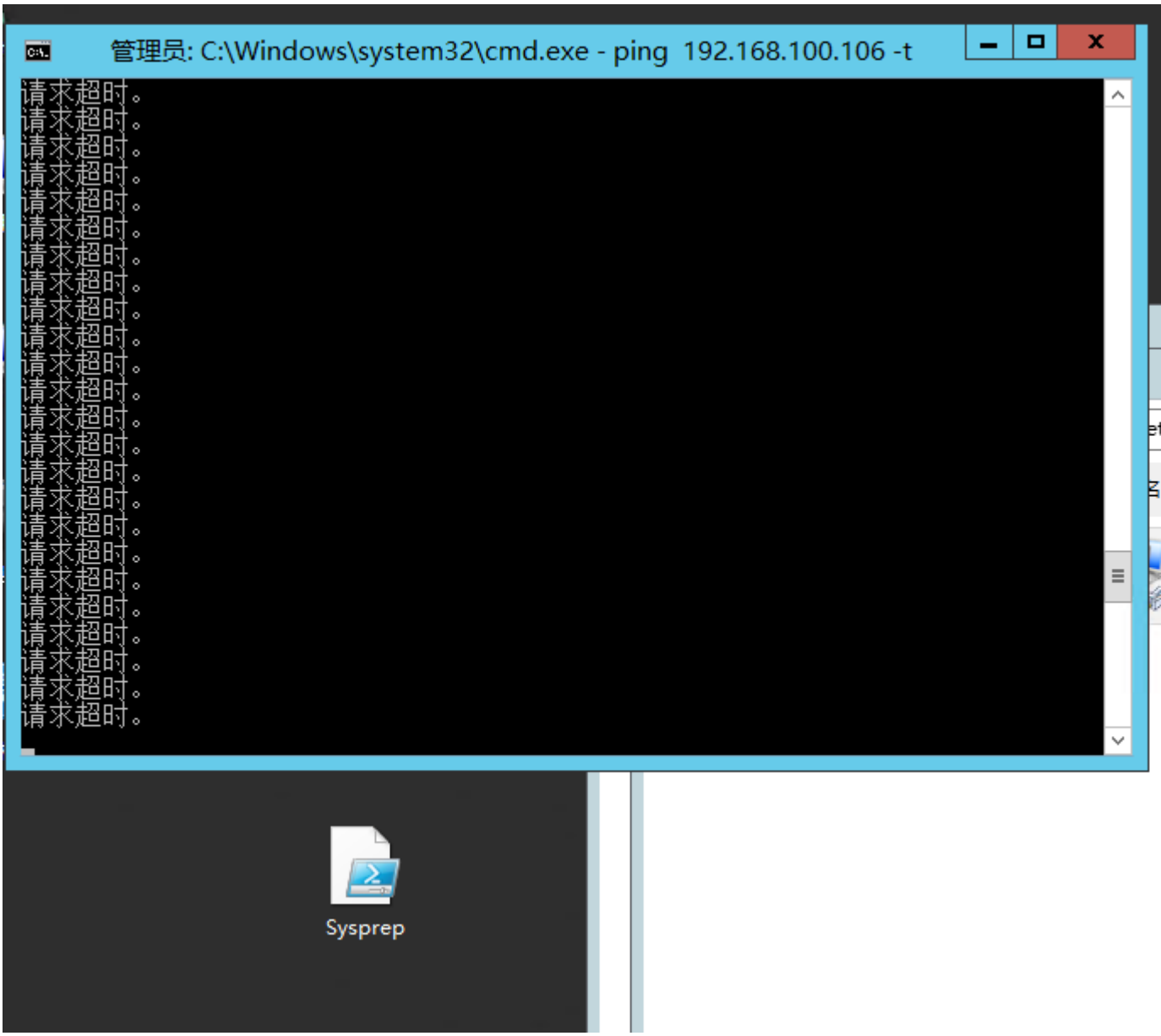
```
.\Sysprep.ps1 -ReserveHostname -ReserveNetwork -skiprearm -post_action
 "reboot"
```

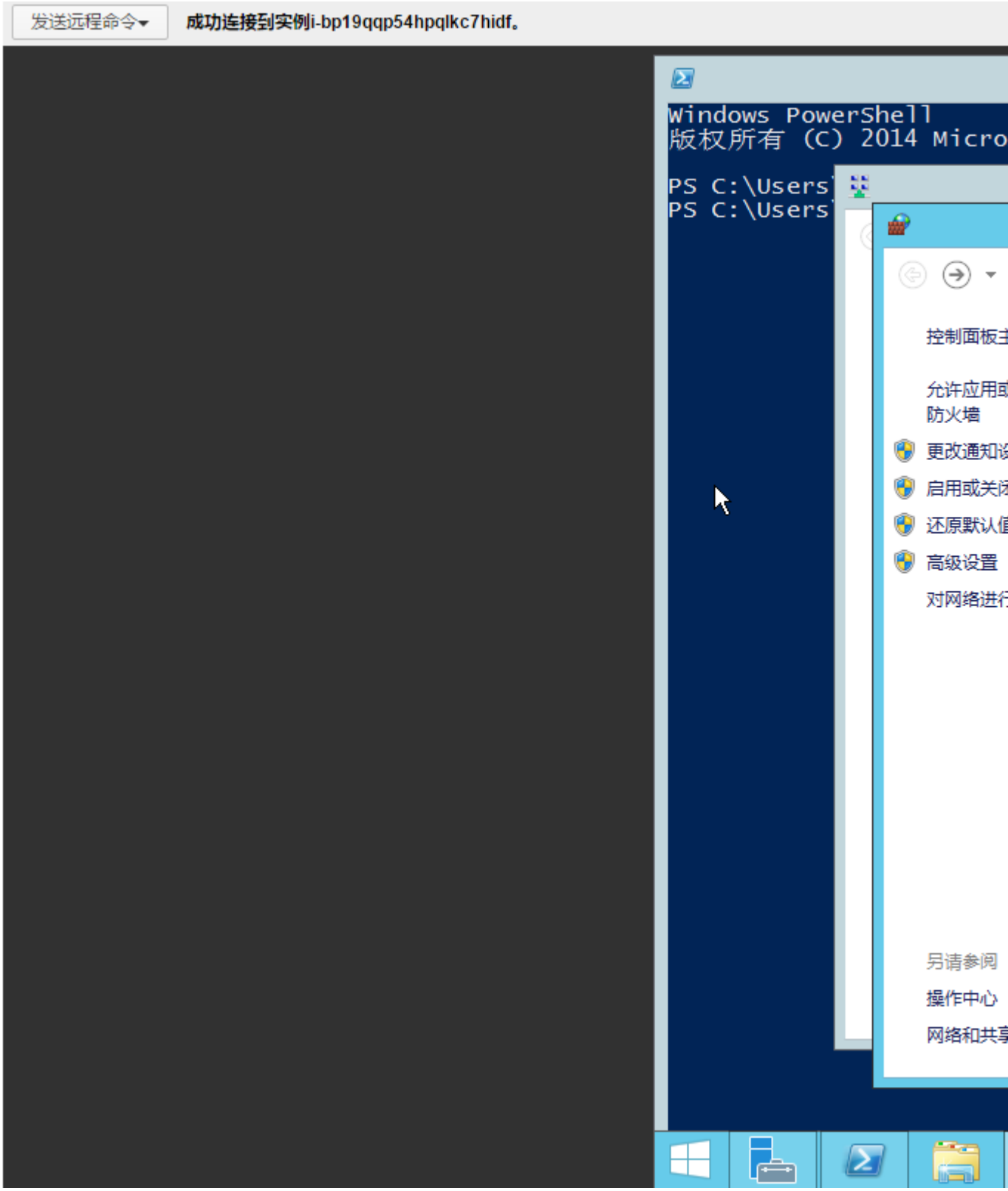
执行上面的命令后，服务器会重新初始化SID，初始化完成后，机器会重启，服务器启动后需要注意两点：

（1）服务器IP地址会从DHCP变成固定IP地址，这里你可以重新改成DHCP，我前面说过，如果想修改ECS 的地址最好从控制台操作。

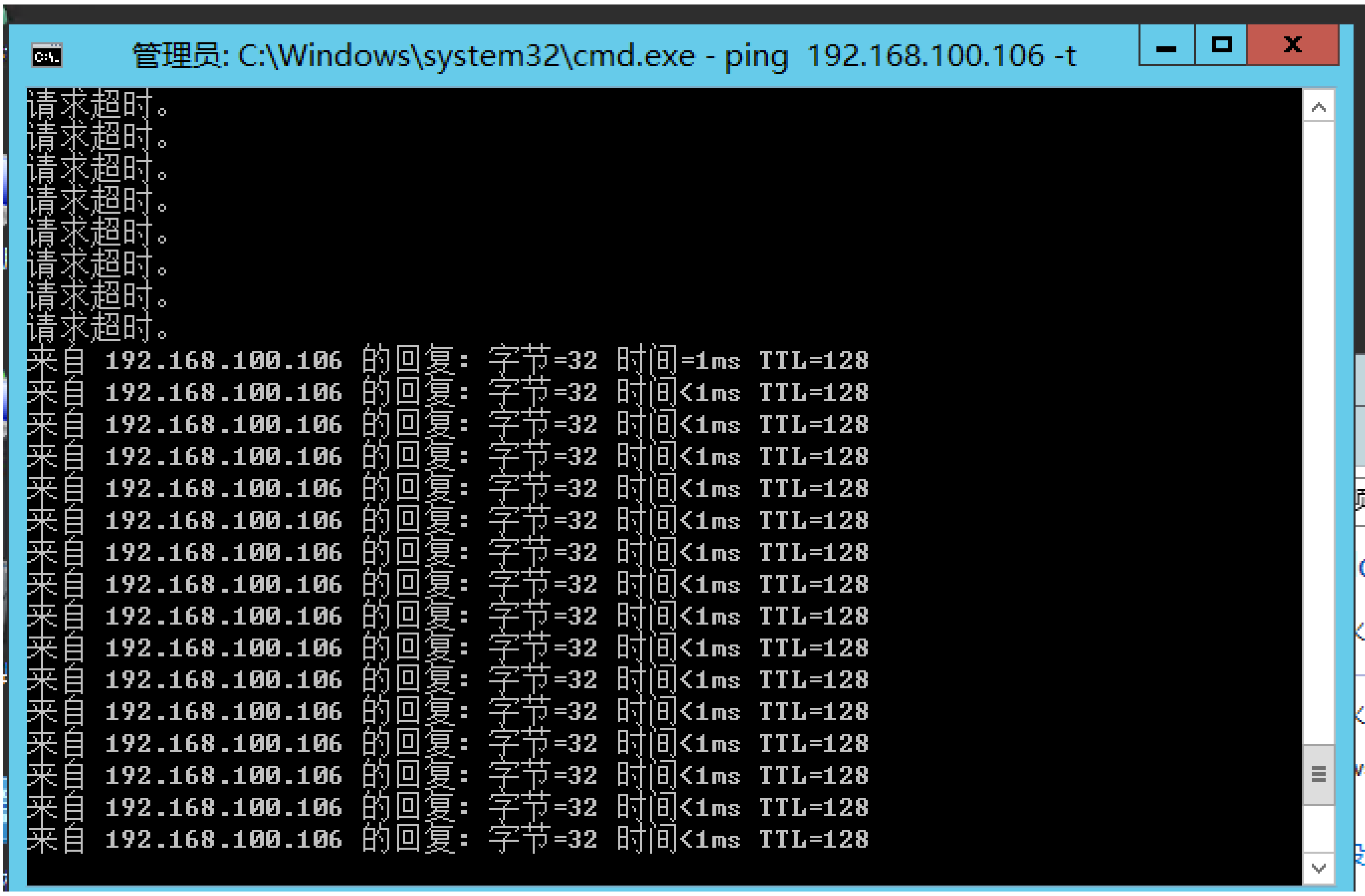
（2）服务器无法PING 通，这是因为服务器SID初始化完成后，也将服务器防火墙的配置修改成微软默认的配置，也就是将"来宾或公用网络"打开，导致无法ping 通服务器和远程。这个时候我们就需要在web console 界面将防火墙"来宾或公用网络"关闭，或者放行需要开放的端口。


发送远程命令
成功连接到实例i-bp19qqp54hpqlkc7hidf。
Windows PowerShell
版权所有 (C) 2014 Microsc
PS C:\Users\Administrator
用户信息
用户名
iz4hpqlkc7hidfz\administr
PS C:\Users\Administrator
我这里使用VNC终端链接的，这里可以看到SID已经发生改变，
当我使用MSTSC工具链接的是时候发现并不能远程链接


管理员: C:\Windows\system32\cmd.exe - ping 192.168.100.106 -t
请求超时。
Sysprep


发送远程命令
成功连接到实例i-bp19qqp54hpqlkc7hidf。
Windows PowerShell
版权所有 (C) 2014 Micro
PS C:\Users
PS C:\Users
控制面板主
允许应用或
防火墙
更改通知设
启用或关闭
还原默认值
高级设置
对网络进行
另请参阅
操作中心
网络和共享


Windows 防火墙
控制面板 ▸ 所有控制面板项 ▸ Windows 防火墙
控制面板主页
允许应用或功能通过 Windows 防火墙
更改通知设置
启用或关闭 Windows 防火墙
还原默认值
高级设置
对网络进行疑难解答
使用 Windows 防火墙来帮助保护你的电脑
Windows 防火墙有助于防止黑客或恶意软件通过 Internet 或网络访问你的电脑。
更新防火墙设置
Windows 防火墙未使用推荐的设置来保护计算机。
推荐的设置有哪些?
域网络(M)
工作区中连接到域的网络
Windows 防火墙状态:
关闭
传入连接:
活动的域网络:
lyonz.com
通知状态:
专用网络(R)
来宾或公用网络(P)
另请参阅

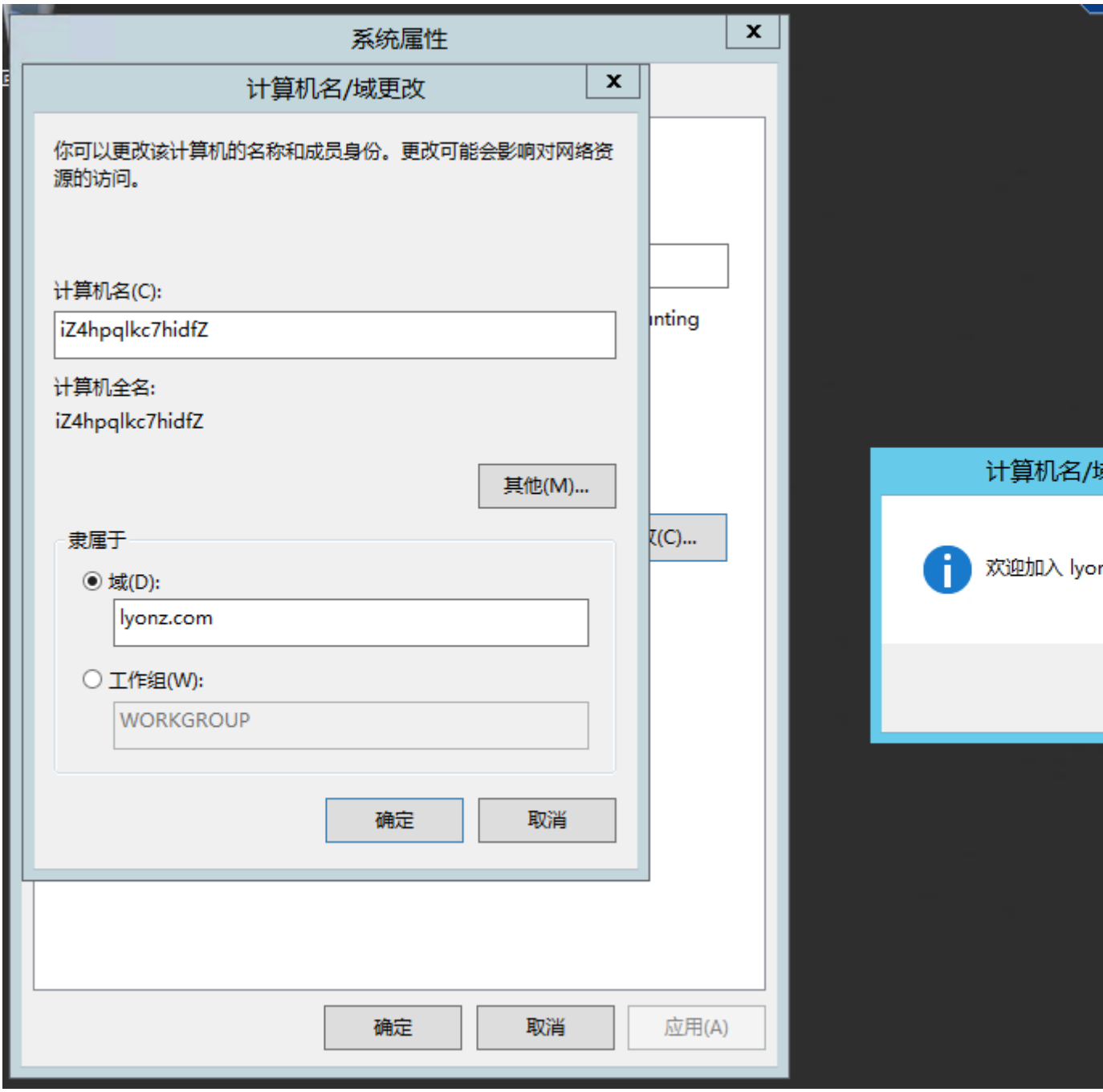


修改客户端的基本信息

DNS 指向 DC 的IP地址，您可以根据业务需求修改主机名。


Internet 协议版本 4 (TCP/IPv4) 属性
常规
备用配置
如果网络支持此功能，则可以获取自动指派的 IP 设置。否则，你需要从网络系统管理员处获得适当的 IP 设置。
自动获得 IP 地址(O)
使用下面的 IP 地址(S):
IP 地址(I):
子网掩码(U):
默认网关(D):
192 . 168 . 100 . 253
自动获得 DNS 服务器地址(B)
使用下面的 DNS 服务器地址(E):
首选 DNS 服务器(P):
192 . 168 . 100 . 105
备用 DNS 服务器(A):
退出时验证设置(L)
高级(V)...


版权所有 (C) 2014 Microsoft Corporation。保留所有权利。
PS C:\Users\Administrator> firewall.cpl
PS C:\Users\Administrator> nslookup
DNS request timed out.
timeout was 2 seconds.
默认服务器: UnKnown
Address: 192.168.100.105
> lyonz.com
服务器: UnKnown
Address: 192.168.100.105
名称: lyonz.com
Address: 192.168.100.105
> exit
PS C:\Users\Administrator> ping lyonz.com
正在 Ping lyonz.com [192.168.100.105] 具有 32 字节的数据:
来自 192.168.100.105 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.100.105 的回复: 字节=32 时间<1ms TTL=128
192.168.100.105 的 Ping 统计信息:
数据包: 已发送 = 2, 已接收 = 2, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
最短 = 0ms, 最长 = 0ms, 平均 = 0ms
Control-C
PS C:\Users\Administrator>
PS C:\Users\Administrator>



以上就是阿里云ECS Windows Server 2012 搭建域以及客户端加入域的过程，如果您已经在线下或者虚拟机搭建了 AD 域，在阿里云上搭建 AD 域时需要注意修改客户端SID。

相关链接

- 域控常见问题配置
- 更多开源软件尽在云市场

3.5 时间配置：NTP服务器与其他基础服务

阿里云ECS提供了内网NTP服务器，对于阿里云以外的设备，阿里云同时提供了公网NTP服务器，供互联网上的设备使用。

内网和公网NTP服务器

ECS为您提供了高精度的时间参考NTP服务器，其中`ntp.cloud.aliyuncs.com`服务器在地域级别上提供原子参考钟服务。适合金融、通讯、科研和天文等以时间精度核心的生产行业。

经典网络内网	专有网络VPC内网	公网
ntp.cloud.aliyuncs.com		ntp1.aliyun.com
ntp1.cloud.aliyuncs.com	ntp7.cloud.aliyuncs.com	ntp2.aliyun.com
ntp2.cloud.aliyuncs.com	ntp8.cloud.aliyuncs.com	ntp3.aliyun.com
ntp3.cloud.aliyuncs.com	ntp9.cloud.aliyuncs.com	ntp4.aliyun.com
ntp4.cloud.aliyuncs.com	ntp10.cloud.aliyuncs.com	ntp5.aliyun.com
ntp5.cloud.aliyuncs.com	ntp11.cloud.aliyuncs.com	ntp6.aliyun.com
ntp6.cloud.aliyuncs.com	ntp12.cloud.aliyuncs.com	ntp7.aliyun.com

其他互联网基础服务

阿里云还提供了其他的互联网基础服务，如下表所示：

公共服务	描述
公共DNS：223.5.5.5 / 223.6.6.6	域名：`http://www.alidns.com`
公共镜像站：`http://mirrors.aliyun.com`	镜像同步频率：每天凌晨2:00–4:00。覆盖了大多数开源软件及Linux发行版。

3.6 为多台Windows实例配置语言偏好

本文使用公共镜像中的Windows Server 2016英语版操作系统为例，从Windows更新下载德语资源包，为多台实例设置德语语言偏好。创建使用德语和德语键盘设置的自定义镜像后，您可以使用该自定义镜像根据需要创建任意数量的实例。

背景信息

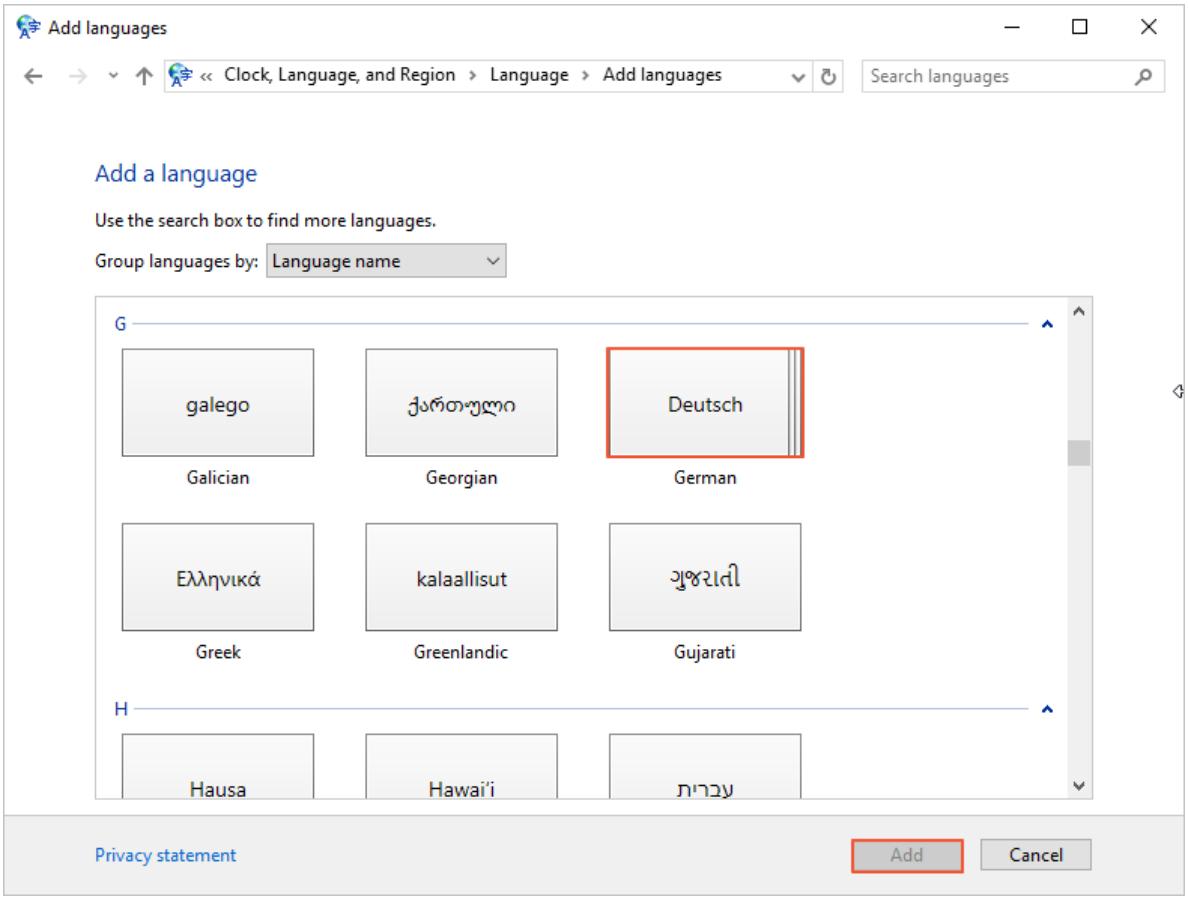
目前，阿里云ECS仅提供中文版和英文版的 Windows Server 镜像。如果要使用其他语言版本，如阿拉伯语、德语或俄语，可以按照本文设置和部署 ECS 实例。

操作步骤

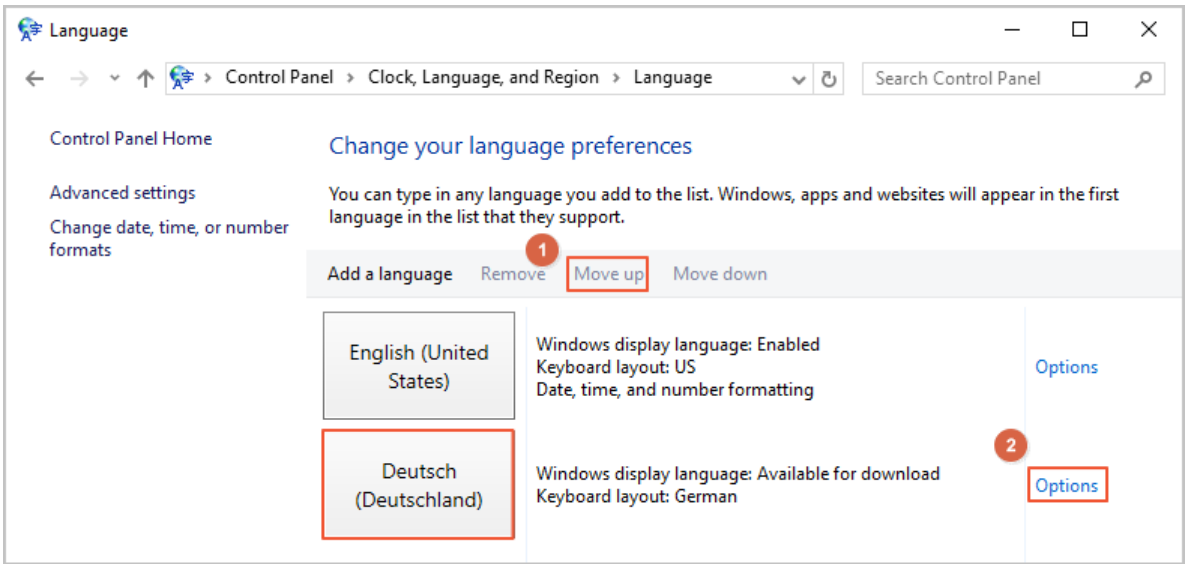
1. 连接到 *Windows* 实例。
2. 打开 PowerShell 模块。
3. 运行以下命令以临时禁用 WSUS。

```
Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows\
WindowsUpdate\AU' -Name UseWUServer -Value 0
Restart-Service -Name wuauserv
```

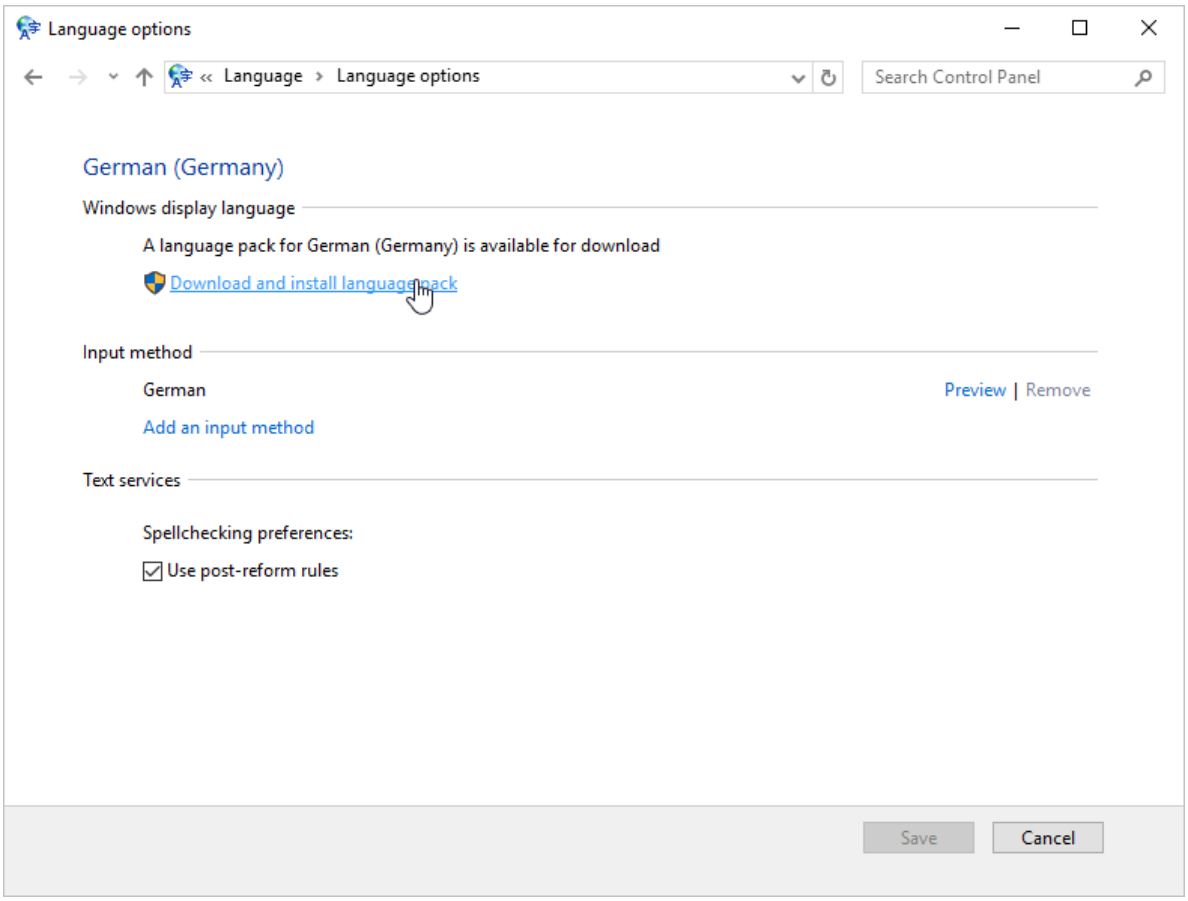
4. 找到控制面板，单击 **Clock, Language, and Region** > **Language** > **Add a language**。
5. 在**Add languages**对话框中，选择一种语言，例如**Deutsch (German)** > **Deutsch (Deutschland)**，然后单击**Add**。

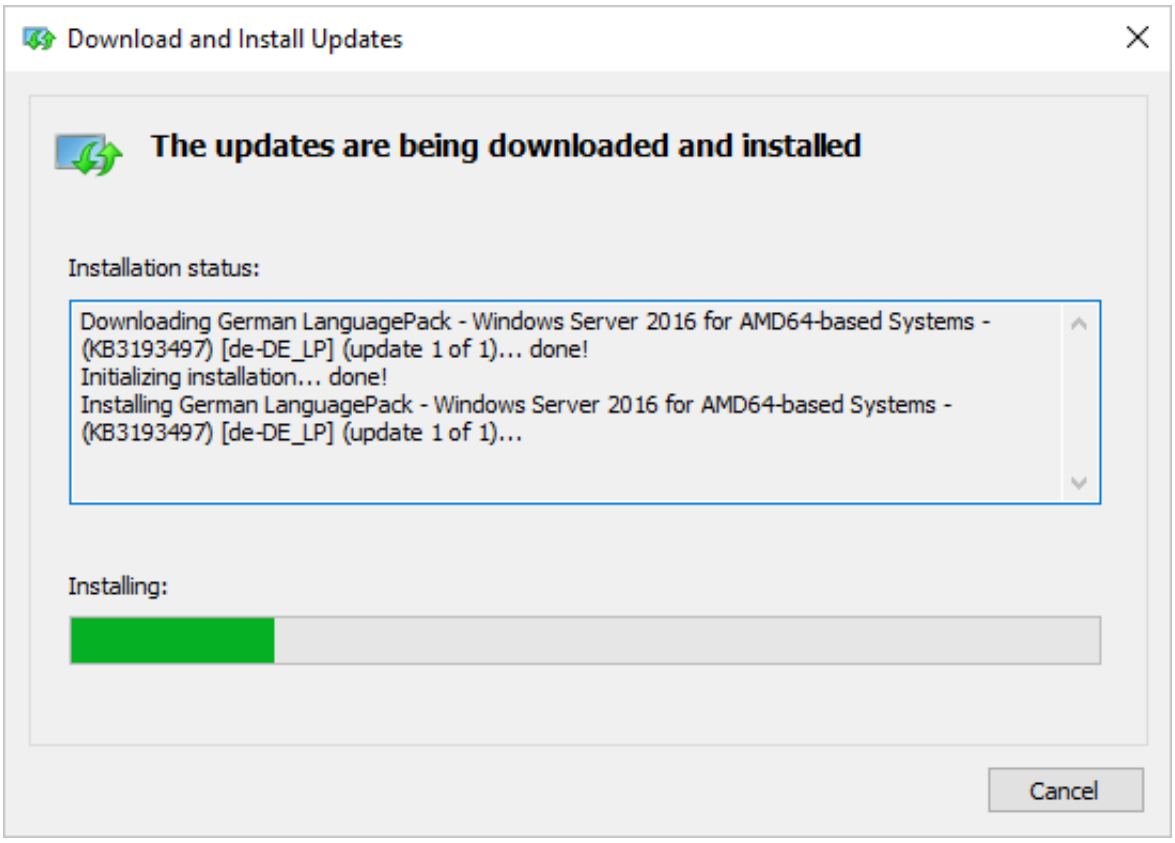


6. 选择语言，例如 **Deutsch (Deutschland)**，然后单击**Move up**以更改语言优先级。
7. 单击所选语言旁边的**Options**以在线检查语言更新。



8. 实例检查更新需等待大约 3 分钟。更新可供下载后，请单击**Download and install language pack**，然后等待安装完成。





9. 重新启动实例，显示语言会在下次登录时更改。
10. 再次 连接到 *Windows* 实例。显示语言现在为德语。
11. 打开 PowerShell ISE 模块，然后运行以下命令重新打开 WSUS。

```
Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows\
WindowsUpdate\AU' -Name UseWUServer -Value 1
Restart-Service -Name wuauserv
```

12. 打开**Windows Update**，检查安全更新，并重新安装配置语言设置之前已完成的所有安全更新。

后续操作

使用相同语言设置创建多台实例：

1. 登录*ECS*管理控制台。
2. 根据该Windows实例创建自定义镜像。
3. 通过自定义镜像创建指定数量的实例。

3.7 时间设置：设置Linux实例时区和NTP服务

目前，所有地域的阿里云ECS实例的默认时区为CST（China Standard Time），您可以根据自己的业务需求并参照本文为ECS实例设置或者修改时区。此外，NTP（Network Time Protocol）服务能保证您的云服务器ECS的时间与标准时间同步，您可以根据本文配置NTP服务。

背景信息

时区和时间的同步性对于云服务器很重要（例如您在更新数据库时，时间的准确性对业务的影响会非常大），为避免实例上运行的业务逻辑混乱和避免网络请求错误，您需要将一台或多台实例设置在同一时区下，比如`Asia/Shanghai`或`America/Los Angeles`。此处以Centos 6.5实例为例，列举通过修改配置文件修改时区的方法：

> 说明：
> 修改时区后，请切记您需要运行`hwclock -w`更新实例硬件时钟。

操作步骤

1. 远程连接 Linux实例。

   > 说明：
   > 您需要以root身份打开并编辑时区配置文件，所以此处使用`sudo`命令。

2. 执行命令`sudo rm /etc/localtime`删除系统里的当地时间链接。
3. 执行命令`sudo vi /etc/sysconfig/clock`用vim打开并编辑配置文件/etc/sysconfig/clock。
4. 输入`i`添加时区城市，例如添加`Zone=Asia/Shanghai`，按下Esc键退出编辑并输入`:wq`保存并退出。

   可执行命令`ls /usr/share/zoneinfo`查询时区列表，`Shanghai`为列表条目之一。
5. 执行命令`sudo ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime`更新时区修改内容。
6. 执行命令`hwclock -w`更新硬件时钟（RTC）。
7. 执行命令`sudo reboot`重启实例。
8. 执行命令`date -R`查看时区信息是否生效，未生效可重走一遍步骤。

后续操作

Linux系统有ntpd和ntpdate两种方式实现NTP时间同步，其中，ntpd同步时间为步进式的逐渐调整时间，ntpdate为断点更新。新购实例可使用ntpdate更新服务，已经运行业务的实例建议使用ntpd同步时间。此处提供标准NTP服务配置和自定义NTP服务配置，您可以根据需要选择性地配置。关于更多NTP服务信息请参考 内网和公共*NTP*服务器。

前提条件

NTP服务的通信端口为UDP 123，设置NTP服务之前请确保您已经打开UDP 123端口。您可以通过netstat -nupl查看实例是否开通UDP 123端口。您可以参考文档 添加安全组规则 放行UDP 123端口。

启用标准NTP服务

1. 远程连接 Linux实例。
2. 执行命令sudo service ntpd start运行NTP服务。
3. 执行命令chkconfig ntpd on启用NTP服务。
4. 执行命令ntpstat查看是否启用了NTP服务。
5. （可选）执行命令ntpq -p可查看NTP服务对等端的列表信息；执行命令sudo chkconfig --list ntpd可查看NTP服务的运行级别。

配置自定义NTP服务

1. 远程连接 Linux实例。
2. 执行命令sudo vi /etc/ntp.conf用vim打开并编辑NTP服务配置文件。
3. 找到server ntp 服务器 iburst的信息后，输入i开始编辑文件，给您暂时不需要的NTP服务器句首加上#隐藏起来。
4. 新添加一行NTP服务器信息，格式为：server 您需要添加的NTP服务器 iburst。完成编辑后按下Esc键并输入:wq保存退出。
5. 执行命令sudo service ntpd start启用自定义的NTP服务。
6. 执行命令chkconfig ntpd on，启用NTP服务。
7. 执行命令ntpstat查看是否启用了NTP服务。

4 监控

4.1 监控ECS实例

一般来说，在本地数据中心我们会对基础设施进行监控，其中包括对主机实例的监控，以便系统地和随时地了解资源使用情况和性能变化，在出现性能瓶颈的时候合理地调配资源，或者在发生故障时追溯原因等等。

在阿里云上，ECS实例也承载着我们的业务应用，ECS实例的资源使用情况和性能负载直接影响着其上应用的运行稳定性和用户体验度。假如没有进行监控，就很有可能在业务高峰期性能不足却无人问津而导致宕机；也可能在出现异常和故障的时候，因为没有历史性能数据而无法进一步追查到原因，可见，没有监控，当问题出现的时候，都非常被动。

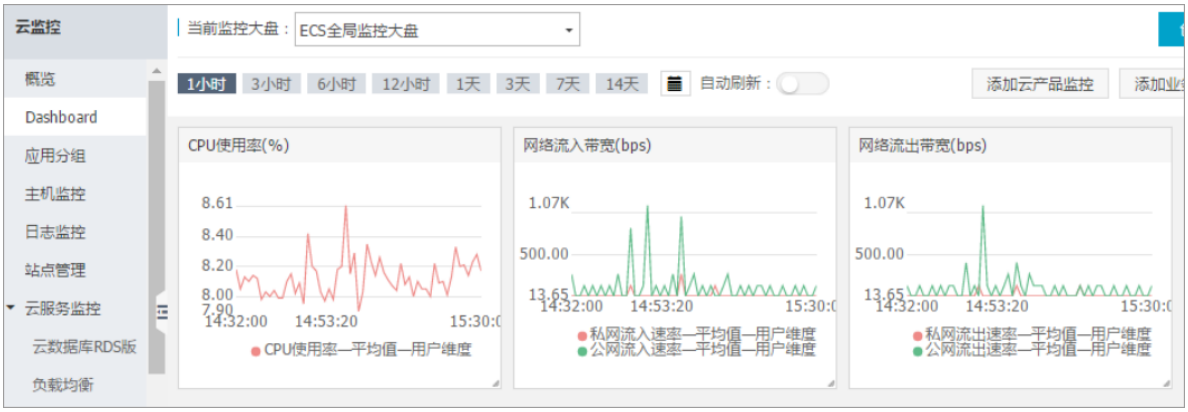
因此，监控是非常有必要的，是构建完整IT系统不可或缺的一个元素，下面就来介绍如何对ECS实例进行监控。

使用Dashboard

云监控的Dashboard功能提供用户自定义查看监控数据的功能。用户可以在一张监控大盘中跨产品、跨实例查看监控数据，将相同业务的不同产品实例集中展现。既能满足排查故障时查看监控细节，又能满足总览大局时查看服务概貌。

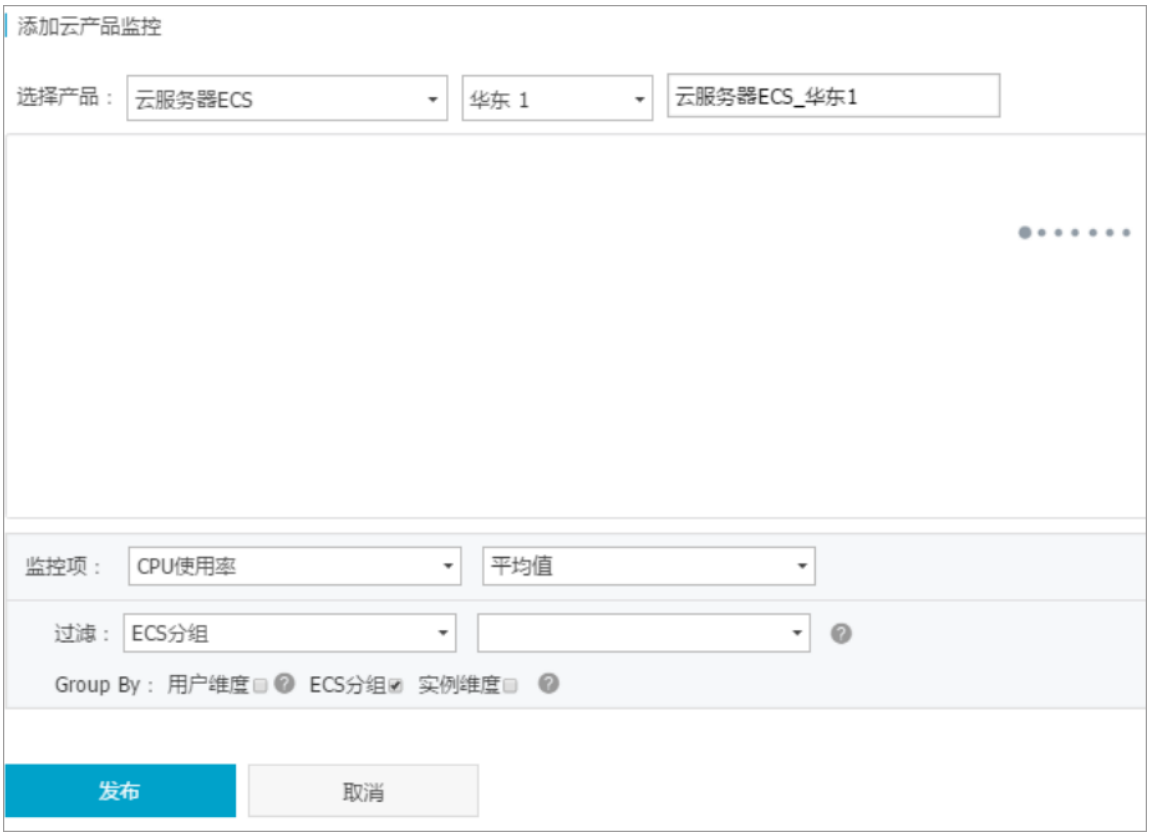
操作步骤

1. 登录云监控控制台。
2. 单击左侧菜单的**Dashboard**选项，进入**Dashboard**页面。可以看到默认展示的ECS全局监控大盘。



3. 可以看到默认的ECS全局监控大盘已经包含了比较丰富的监控项了，包括CPU使用率、网络流入/流出带宽、系统磁盘BPS、系统盘IOPS、网络流入/流出量。基本已经可以满足日常监控需求。
4. 如果业务比较复杂，需要自定义监控可视化需求时，可以创建新的监控大盘，单击页面右上角的**创建监控大盘**，输入监控大盘的名称。

5. 然后可以在该大盘上添加云产品指标和用户的业务监控指标。
6. 添加云产品指标。
   a. 选择需要查看的云产品和实例所在地域。
   b. 定义图标名称，图表名称默认为您生成产品名称+区域，选择图表展现形式。
   c. 选择需要查看的监控项、选择监控数据的聚合方式，常见聚合方式为最大值、最小值、平均值、选择过滤条件、选择Group By的维度。



7. 添加业务指标监控。

a. 定义图表名称、指标名称、图表类型。

b. 选择需要查看的监控数据并定义处理方式。

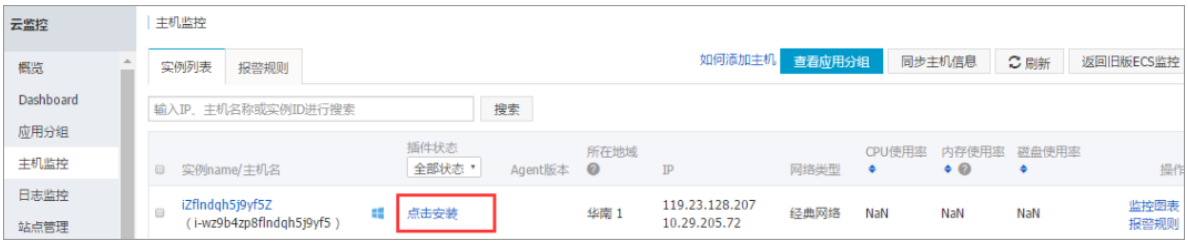
c. 单击发布。

主机监控

云监控主机监控服务通过在服务器上安装插件，为用户提供服务器的系统监控服务。主机监控服务采集丰富的操作系统层面监控指标，可以使用主机监控服务进行服务器资源使用情况的查询和排查故障时的监控数据查询。

操作步骤

1. 登录云监控控制台
2. 通过选择左侧菜单的主机监控，进入主机监控页面。
3. 单击实例列表中的单击安装插件，安装云监控插件。



4. 1-3分钟后即可单击实例列表页的监控图表查看监控数据。

5. 可以看到有操作系统监控、基础监控、进程监控。其中涵盖了CPU、内存、负载、磁盘、网络、进程各面的性能统计，并且可以根据时间范围来展示图标数据。
6. 创建报警规则。
   a. 切换到报警规则页面。

   b. 单击这里创建规则。
   c. 在新建报警规则页面填写设置报警的具体参数。

d. 保存规则设置，完成报警规则的创建。

站点监控

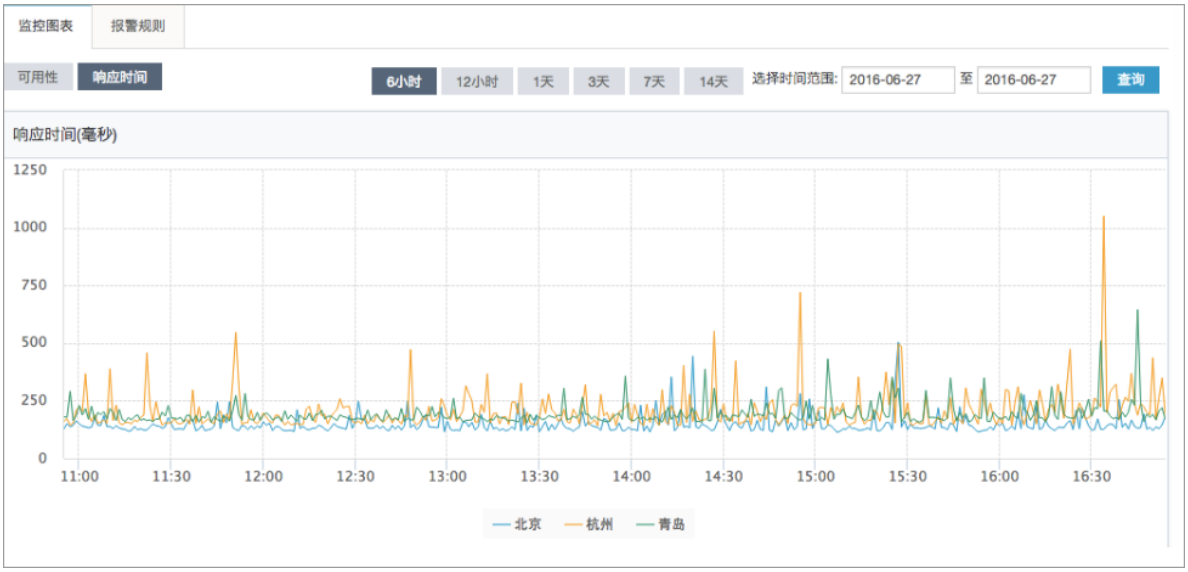
如果ECS实例提供的主要业务应用是网站类型，可以考虑使用站点监控模拟真实用户访问情况，探测API可用性、端口连通性、DNS解析等问题。可以探测域名、IP、端口的连通性、访问响应时间，并对监控结果报警。

操作步骤

1. 登录云监控控制台。
2. 单击站点管理，进入站点监控页面。
3. 单击页面右上角的创建监控点，添加新的监测点。

4. 单击左侧菜单的站点管理选项，进入站点监控页面。

5. 查看站点监控详情。



开源监控产品介绍

目前业内有不少开源的监控软件，包括zabbix、nagios、zenoss等，每个产品都有各自的特色和优势，下面分别简单介绍一下以上几款产品。

- zabbix

Zabbix是一个基于WEB界面的提供分布式系统监控以及网络监控功能的企业级开源运维平台，也是目前国内互联网用户中使用最广的监控软件，85%以上的泛互联网企业都在使用Zabbix做监控解决方案。

zabbix入门容易、上手简单、功能强大并且开源免费，它易于管理和配置，能生成比较漂亮的数据图，其自动发现功能大大减轻日常管理的工作量，丰富的数据采集方式和API接口可以让用户灵活进行数据采集，而分布式系统架构可以支持监控更多的设备。理论上，通过Zabbix提供的插件式架构，可以满足企业的任何需求。

- nagios

Nagios是一款开源的企业级监控系统，能够实现对系统CPU、磁盘、网络等方面参数的基本系统监控，以及SMTP，POP3，HTTP，NNTP等各种基本的服务类型。另外通过安装插件和编写监控脚本，用户可以实现应用监控，并针对大量的监控主机和多个对象部署层次化监控架构。

Nagios最大的特点是其强大的管理中心，尽管其功能是监控服务和主机的，但Nagios自身并不包括这部分功能代码，所有的监控、告警功能都是由相关插件完成的。

- zenoss

Zenoss Core是Zenoss的开源版本，其商用版本为ZenossEnterprise。作为企业级智能监控软件，Zenoss Core允许IT管理员依靠单一的WEB控制台来监控网络架构的状态和健康度。Zenoss Core的强大能力来自于深入的列表与配置管理数据库，以发现和管理公司IT环境的各类资产。Zenoss同时提供与CMDB关联的事件和错误管理系统，以协助提高各类事件和提醒的管理效率。

Zabbix vs 云监控

Zabbix是第三方开源监控软件，是一个基于WEB界面的提供分布式系统监视以及网络监视功能的企业级的开源解决方案。

zabbix能监视各种网络参数，保证服务器系统的安全运营；并提供灵活的通知机制以让系统管理员快速定位/解决存在的各种问题。

云监控既指在云端运行的监控工具，也指监控在云端运行的应用程序的工具。通过和云计算平台的整合，针对网络、系统、应用等内容提供可用性、用户体验和安全性方面的监控服务。

云监控的到来，无疑给那些对技术不太熟悉的人员带来了福音，可以通过页面单击就可以创建自己的监控项。

产品	优点	缺点
Zabbix	• 支持多平台、分布式 • 安装部署简单，多种数据采集插件灵活集成 • 可实现复杂多条件告警 • 自带画图功能，得到的数据可以绘成图形 • 提供多种API接口，支持调用脚本 • 出现问题时可自动远程执行命令	• 项目批量修改不方便 • 中文资料较少，服务支持有限 • 入门容易，但是深层次需要非常熟悉zabbix并进行大量的二次定制开发，难度较大 • 系统级别报警、报警邮件、自定义项目报警需要自己设置，过程繁琐 • 缺少数据汇总功能，数据报表也需要进行二次开发
云监控	• 无前期成本投入 • 无需独立服务器 • 配置及添加监控项简单 • 页面风格比较适合国人操作	• 部分平台免费版功能较少，企业级应用费用较高 • 账户管理功能较弱 • 修改监控点配置不方便 • 自定义监控配置麻烦，部分需写脚本 • 监控项目单一

产品	优点	缺点
		• 部分监控项无法实现图形化显示

可以看出，各有各的优劣势。云监控降低我们监控的门槛，给我们提供了便利，但是在一定程度上限制了自定义和扩展。而zabbix可以灵活集成并可通过二次开发实现复杂功能，但是对人员和技能的要求也比较高。

对于上监控以更好地保障系统上线后稳定运行，我们还需要关注监控的一些方法。

除了需要了解我们的常规的监控项如硬件资源、性能、带宽、端口、进程、服务的检测机制之外，还要具备安全意识，比如需要知道哪些服务器可能出现问题，可能被入侵等。

另外，需要定义监控策略，包括告警的优先级、告警内容等；对监控的业务系统进行分级，比如一级系统7*24小时告警，二级系统7*12小时告警。

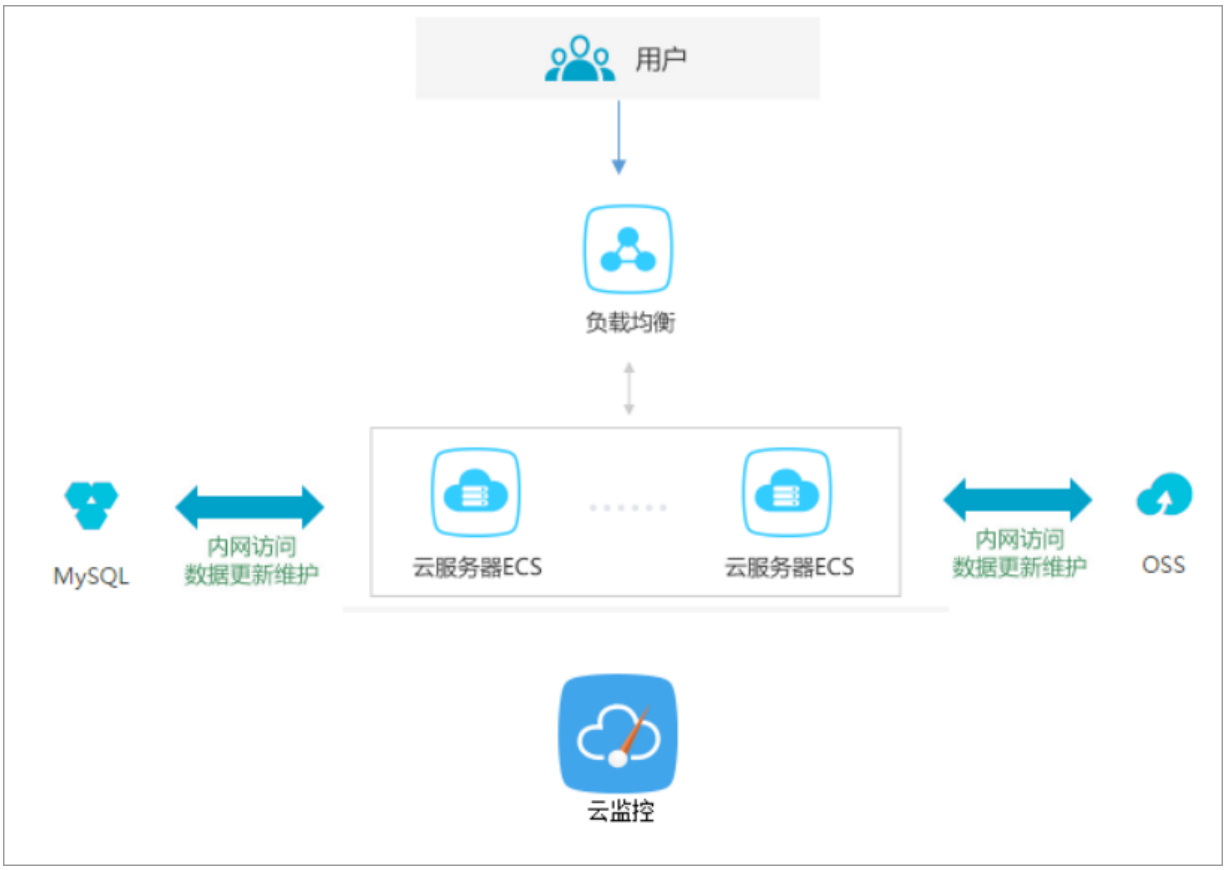
如果架构比较庞大，也可以对监控对象范围进行分类，如服务器监控、应用程序监控、数据库监控、网络监控等，根据监控对象再细分监控项。每个维护人员都可以根据企业环境总结出一套适合于自身的监控体系，并逐渐精细化和智能化。

通过使用阿里云云监控，能较好地对我们的ECS实例进行监控，使我们及时了解业务的运行状态，并及时提供告警，让我们可以快速定位故障，对我们管理和维护ECS提供了可靠的支持。当然，在此基础上我们也可以结合如zabbix之类的开源监控软件，进一步实现对ECS实例更全面和精准的监控。

4.2 使用云监控监控ECS实例

越来越多的业务部署在云上，减轻了运维成本和压力，其中合理的监控设置功不可没。设置合理的监控不仅可以让您实时了解系统业务的运行情况，还能帮助您提前发现问题，避免可能会出现的业务故障。同时，有效的告警机制能让您在故障发生后第一时间发现问题，缩短故障处理时间，以便尽快恢复业务。

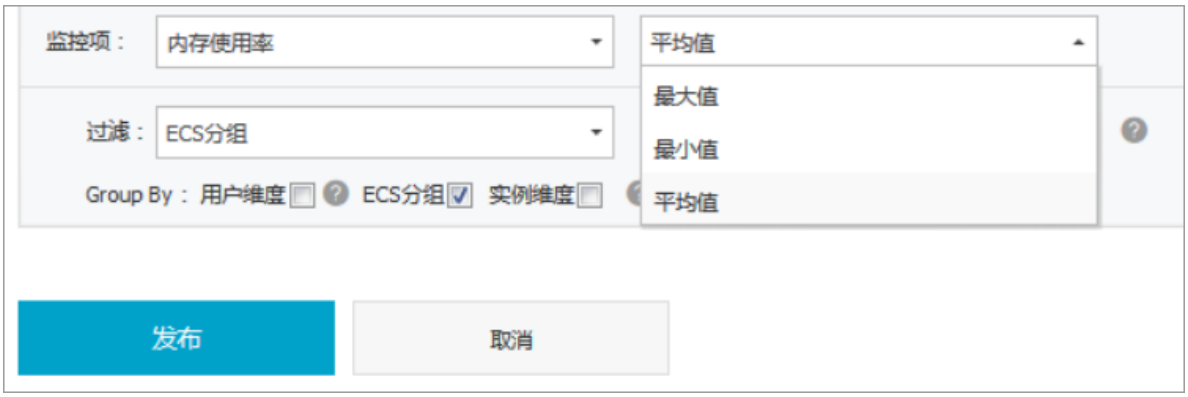
本文以一个示例网站为例（网站架构如下图所示）说明您应如何配置使用云监控。示例网站架构使用了阿里云产品ECS、RDS、OSS和负载均衡。



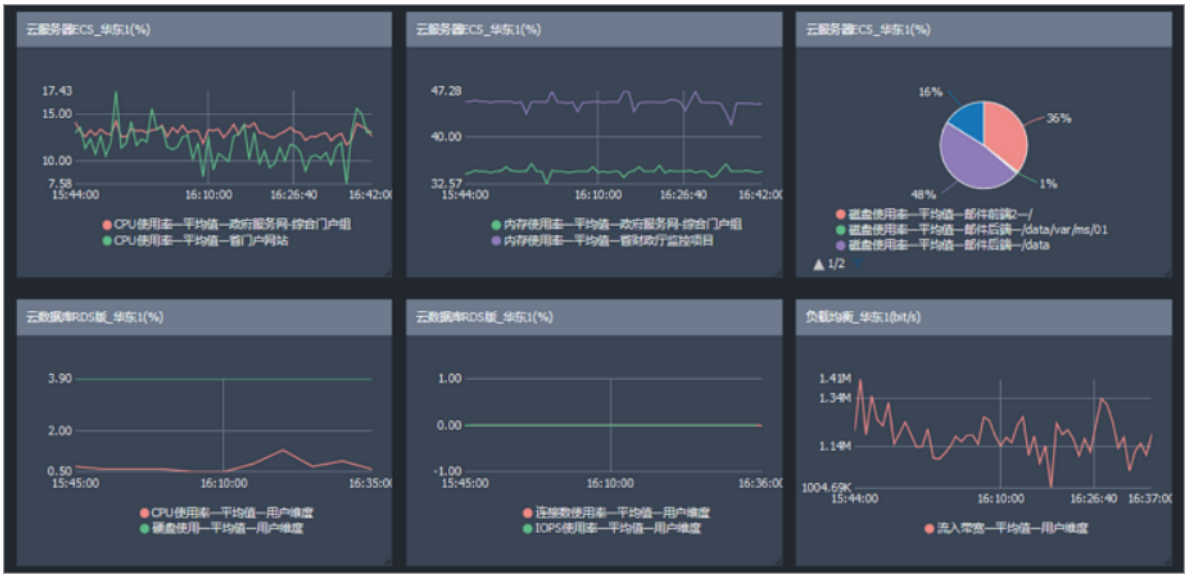
前提条件

在开始设置云监控前，您需要完成以下操作：

- 检查ECS监控插件运行情况，确保监控信息能够正常采集。如果安装失败需要手动安装，请参考云监控插件安装指南。
- 提前 添加报警联系人和联系组，建议设置至少2人以上的联系人，互为主备，以便及时响应监控告警。监控选项的设定，具体可参见 云服务资源使用概览和报警概览。
- 利用云监控的Dashboard功能，给您业务系统的云资源设置一个全局监控总览，可随时检查整个业务系统资源的健康状态。下图根据ECS分组选择添加监控的资源，依次添加内存使用率、CPU使用率等监控项。监控的实例数较少可以选择实例维度作为展示，如果实例较多建议以分组或者用户为维度展示；监控数据取平均值。



为了更好地监控大屏展示效果，这里将ECS的CPU、内存、磁盘的使用率单独分组展示；将RDS的四项指标分两组展示。



设置报警阈值

建议您根据实际业务情况设置各项监控指标的报警阈值。阈值太低会频繁触发报警，影响监控服务体验。阈值太高，在触发阈值后没有足够的预留时间来响应和处理告警。

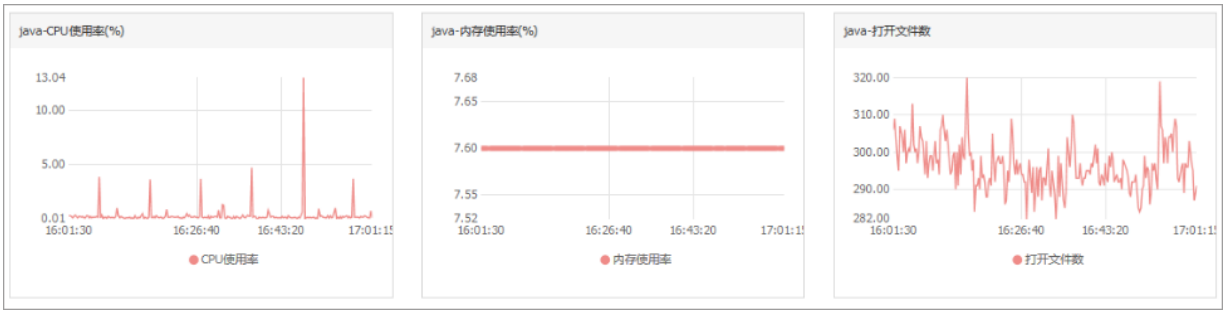
设置报警规则

以CPU使用率为例，因为需要给服务器预留部分处理性能保障服务器正常运行，所以建议您将CPU告警阈值设置为70%，连续三次超过阈值后开始报警，如下图所示。

如果您还需要设置其他资源的报警规则，单击 添加报警规则，继续设置内存或磁盘的报警规则和报警通知人。

设置进程监控

对于常见的web应用，设置 进程监控，不仅可以实时监控应用进程的运行情况，还有助于排查处理故障，下图是Java进程的相关监控示例。具体操作请参见 添加进程监控。



设置站点监控

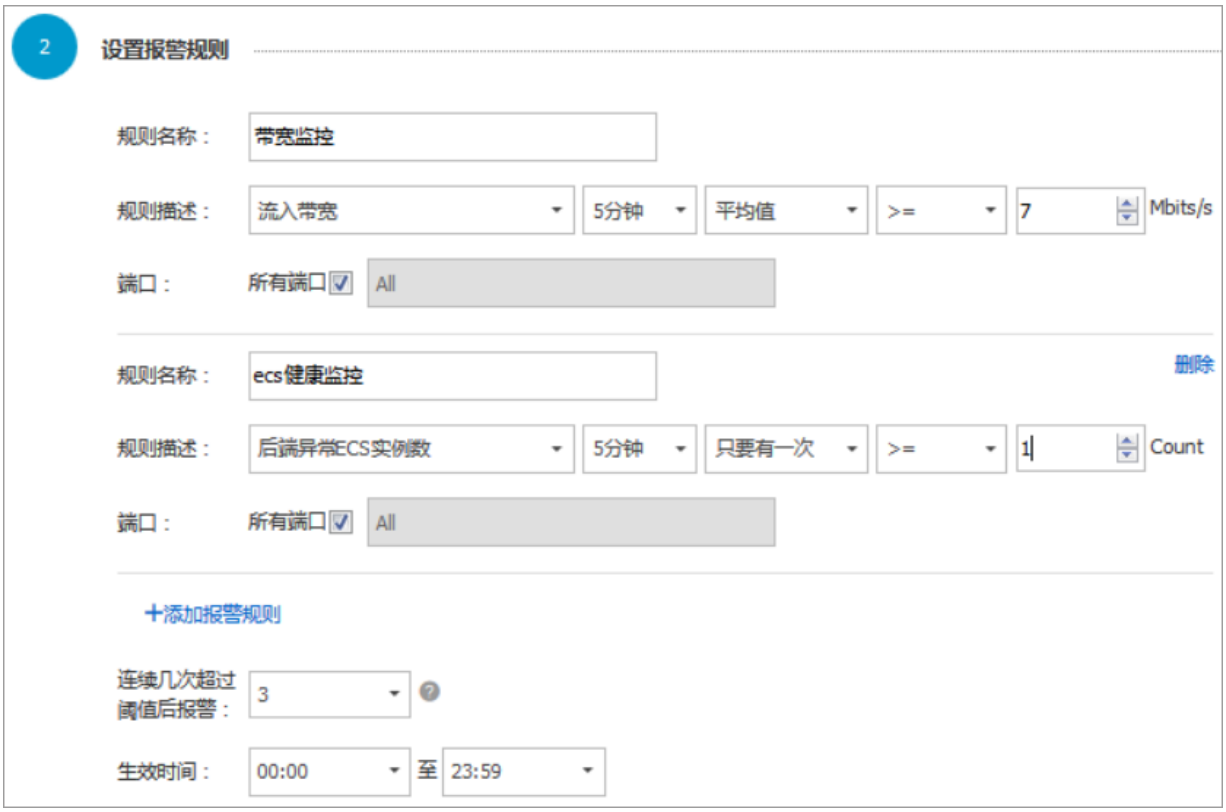
在云服务器外层的监控服务，站点监控主要用于模拟真实用户访问情况，实时测试业务可用性，有助于排查处理故障。

设置RDS监控

建议将RDS的CPU使用率报警阈值设置为70%，连续三次超过阈值后开始报警。您可以根据实际情况设置硬盘使用率、IOPS使用率、连接数等其他 监控项。

设置负载均衡监控

为了更好使用负载均衡的云监控服务，您需要先开启负载均衡的健康检查，将负载均衡带宽值的70%作为告警阈值，如下图所示。



如果以上监控选项不能满足您的实际业务监控需求，您可以使用自定义监控项。

4.3 使用云助手自动化管理实例

运维 ECS 实例的目的是保持 ECS 实例的最佳状态以及确保排错的效率，但是手动维护会花费您大量的时间和精力，因此阿里云研制了云助手，用以解决如何自动化、批量处理日常维护任务。本文举例如何使用云助手API，为 ECS 实例执行相应命令，达到自动化运维 ECS 实例的目的。

前提条件

- 您需要确保目标 ECS 实例的网络类型为 专有网络*#VPC#*。
- 目标 ECS 实例的状态必须为 运行中（`Running`）。
- 目标 ECS 实例必须预先安装云助手客户端。您可以参阅 云助手客户端 安装并使用云助手客户端。
- 执行类型为 PowerShell 的命令时，您需要确保目标 Windows 实例已经配置了 PowerShell 模块。
- 您需要从 *GitHub* 上获取阿里云CLI。
- 以下示例在命令行工具中完成，您需要确保您已经安装了阿里云命令行工具 *CLI#Command-Line Interface#*。
- 您需要 升级 *SDK*。

背景信息

本文举例说明怎么在阿里云 CLI 中通过 API 使用云助手，为 ECS 实例执行相应命令。以执行一条 `echo 123` 命令为例。

目前，云助手支持如下三种命令类型。

命令类型	参数	描述
Shell 脚本	RunShellScript	为运行中的 Linux 实例执行 Shell 脚本，命令内容为需要执行的 Shell 脚本内容。
PowerShell 脚本	RunPowerShellScript	为运行中的 Windows 实例执行 PowerShell 脚本，命令内容为需要执行的 PowerShell 脚本内容。
Bat 脚本	RunBatScript	为运行中的 Windows 实例执行 Bat 脚本，命令内容为需要执行的 Bat 脚本内容。

操作步骤

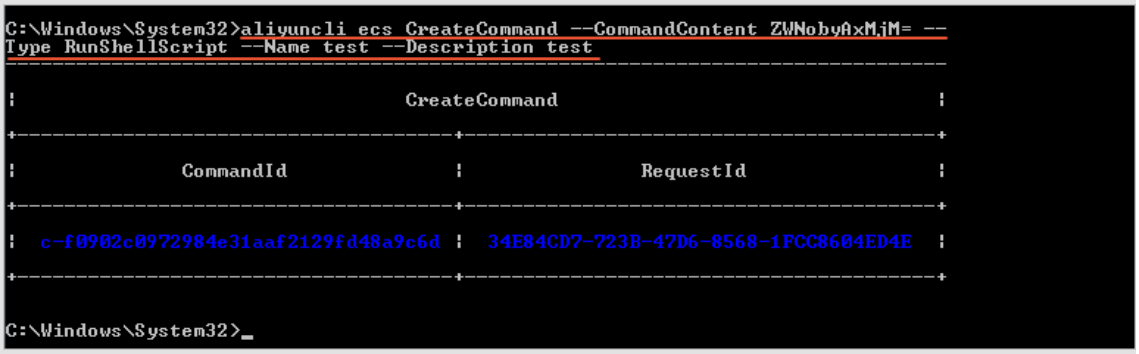
1. 在本地计算机的 CMD、PowerShell 或者 Shell 中运行 `aliyuncli ecs CreateCommand --CommandContent ZWNobyAxMjM= --Type RunShellScript --Name test --Description test` 创建命令（`CreateCommand`）。

   说明：

   - `CommandContent` 中的 `ZWNobyAxMjM=` 是命令 `echo 123` 转化后的 Base64 码。关于 Base64 编码或者译码，您可以参阅 *Wikipedia* 相关介绍。

   - 如果目标 ECS 实例为 Windows 实例，将 `type` 修改为 `RunBatScript` 或者 `RunPowershellScript`。
   - 创建成功后，将返回 `CommandId` 信息。

2. 运行 `aliyuncli ecs InvokeCommand --InstanceId.1 your-vm-instance-id1 --InstanceId.2 your-vm-instance-id2 --CommandId your-command-id --Timed false` 执行命令（`InvokeCommand`）。

   说明：

   - `InstanceIds`为您的 ECS 实例 ID，支持多台 ECS 实例，最多 100 台。
   - `Timed`表示是否为周期性任务，`Timed True` 表示是周期性任务，`Timed False`表示不是周期性任务。
   - 当您的任务为周期性任务时，即参数 `Timed` 取值为 `True` 时，您需要通过参数 `Frequency` 指定周期，例如 `0 */20 * * * *` 表示周期为每 20 分钟。更多关于 Cron 表达式详情，请参阅 *Cron*表达式取值说明。

- 返回结果为所有的目标 ECS 实例返回一个共同的 `InvokeId`。您可以使用该 `InvokeId` 查询命令的执行情况。

3. （可选）运行 `aliyuncli ecs DescribeInvocations --InstanceId your-vm-instance-id --InvokeId your-invoke-id` 查看命令执行状态（`DescribeInvocations`）。其中，`InvokeId` 是 第二步 为 ECS 实例执行命令时返回的执行 ID。

   返回参数 `InvokeStatus` 为 `Finished` 时仅表示命令进程执行完成，不代表一定有预期的命令效果，您需要通过 *DescribeInvocationResults* 中的参数 `Output` 查看实际的具体执行结果。

4. （可选）运行 `aliyuncli ecs DescribeInvocationResults --InstanceId your-vm-instance-id --InvokeId your-invoke-id` 查看指定 *ECS* 实例的命令的实际执行结果（`DescribeInvocationResults`）。其中，`InvokeId` 是 第二步 为 ECS 实例执行命令时返回的执行 ID。

预期结果

在 创建命令（`CreateCommand`）时，您还可以为命令设置如下请求参数。

命令属性	参数	描述
执行目录	WorkingDir	命令将在 ECS 实例中的什么路径下执行。默认值： • 对于 Linux 实例，默认在管理员 root 用户的 home 目录下，具体为 `/root` 目录。 • 对于 Windows 实例，默认在云助手客户端进程所在目录，例如 C:\ProgramData\aliyun\assist\$(version)。
超时时间	TimeOut	修改命令在 ECS 实例中执行时最大的超时时间，单位为秒。 当因为某种原因无法运行您创建的命令时，会出现超时现象；超时后，云助手客户端会强制终止命令进程，即取消命令的 PID。 参数取值必须大于等于 `60`，如果取值小于 `60`，默认为 60 秒。 默认值：3600 • **单次执行**： — 超时后，该命令针对指定的 ECS 实例的执行状态（*DescribeInvocationResults*）变为执行失败（`Failed`）。 • **周期执行**： — 周期执行的超时时间对每一次执行记录均有效。

命令属性	参数	描述
		— 某次执行超时后，该次执行记录的状态（*DescribeInvocationResults*）变为执行失败（`Failed`）。 — 上次执行超时与否不影响下一次执行。

通过 Python SDK 使用云助手的完整代码示例

您也可以通过 阿里云 *SDK* 使用云助手。关于如何配置阿里云 SDK，参阅文档 配置命令行工具和 *SDK*。以下为通过 Python SDK 使用云助手的完整代码示例。

```
# 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 os
import time
import datetime
import base64
from aliyunsdkcore import client

from aliyunsdkecs.request.v20140526.CreateCommandRequest import
CreateCommandRequest
from aliyunsdkecs.request.v20140526.InvokeCommandRequest import
InvokeCommandRequest
from aliyunsdkecs.request.v20140526.DescribeInvocationResultsRequest
import DescribeInvocationResultsRequest

# 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',filename='
aliyun_assist_openapi_test.log', filemode='w')
#access_key = 'Your Access Key Id'
#acess_key_secrect = 'Your Access Key Secrect'
#region_name = 'cn-shanghai'
#zone_id = 'cn-shanghai-b'

access_key = 'LTAIXXXXXXXXXXX'
acess_key_secrect = '4dZXXXXXXXXXXXXXXXXXXXXXXX'
region_name = 'cn-hangzhou'
zone_id = 'cn-hangzhou-f'

clt = client.AcsClient(access_key, acess_key_secrect, region_name)

def create_command(command_content, type, name, description):
    request = CreateCommandRequest()
```

```
    request.set_CommandContent(command_content)
    request.set_Type(type)
    request.set_Name(name)
    request.set_Description(description)
    response = _send_request(request)
    if response is None:
        return None
    command_id = response.get('CommandId')
    return command_id;

def invoke_command(instance_id, command_id, timed, cronat):
    request = InvokeCommandRequest()
    request.set_Timed(timed)
    InstanceIds = [instance_id]
    request.set_InstanceIds(InstanceIds)
    request.set_CommandId(command_id)
    request.set_Frequency(cronat)
    response = _send_request(request)
    invoke_id = response.get('InvokeId')
    return invoke_id;

def get_task_output_by_id(instance_id, invoke_id):
    logging.info("Check instance %s invoke_id is %s", instance_id,
invoke_id)
    request = DescribeInvocationResultsRequest()
    request.set_InstanceId(instance_id)
    request.set_InvokeId(invoke_id)
    response = _send_request(request)
    invoke_detail = None
    output = None
    if response is not None:
        result_list = response.get('Invocation').get('Invocation
Results').get('InvocationResult')
        for item in result_list:
            invoke_detail = item
            output = base64.b64decode(item.get('Output'))
            break;
        return output;

def execute_command(instance_id):
    command_str = 'yum check-update'
    command_id = create_command(base64.b64encode(command_str), '
RunShellScript', 'test', 'test')
    if(command_id is None):
        logging.info('create command failed')
        return

    invoke_id = invoke_command(instance_id, command_id, 'false', '')
    if(invoke_id is None):
        logging.info('invoke command failed')
        return

    time.sleep(15)

    output = get_task_output_by_id(instance_id, invoke_id)
    if(output is None):
        logging.info('get result failed')
        return

    logging.info("output: %s is \n", output)

# 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__':
    execute_command('i-bp17zhpbXXXXXXXXXXXX')
```

后续操作

以上示例示范了如何通过阿里云 CLI 以及云助手 API *CreateCommand*、*InvokeCommand*、*DescribeInvocations* 和 *DescribeInvocationResults* 自动化运维 ECS 实例，您还可以使用云助手其他 API 便捷地管理您的 ECS 实例。

- *StopInvocation*：停止正在进行的命令进程
- *ModifyCommand*：修改已创建的命令的内容
- *DescribeCommands*：查询您已经创建的命令
- *DeleteCommand*：删除已创建的命令

5 实例自定义数据

5.1 自定义 yum 源、NTP 服务和 DNS 服务

实例自定义脚本是阿里云 ECS 为用户提供的一种自定义实例启动行为的脚本，详细信息请参考阿里云线上帮助文档：实例自定义数据。

本文档主要介绍在创建实例时，您怎么使用这个自定义脚本来配置自己的 yum 源、NTP 服务和 DNS 服务。您也可以使用这个脚本自定义 Windows 实例的 NTP 服务和 DNS 服务。

场景

目前，实例启动时，阿里云会为实例自动配置预定义的 yum 源、NTP 服务和 DNS 服务。但是，您可能想拥有自己的 yum 源、NTP 服务和 DNS 服务，此时，您就可以使用实例自定义脚本来实现这个需求，此时您要注意：

- 如果您自定义了 yum 源，阿里云官方将不再提供 yum 源相关支持。
- 如果您自定义了 NTP 服务，阿里云官方不再提供相关时间服务。

配置方法

您可以按以下步骤实现上述场景需求。

1. 登录 云服务器ECS管理控制台，创建实例，配置如下：
   - 网络类型：VPC 网络
   - 实例规格：I/O 优化实例
   - 镜像：公共镜像的 CentOS 7.2
2. 在创建页面的 自定义数据 输入框中输入如下内容：

```
#!/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
```

> 说明：
>
> - 第一行必须是 `#!/bin/sh`，前面不能带空格。
> - 全文不能有多余的空格和回车。
> - 您可以根据实例情况定制具体的 DNS、NTP Server 和 yum 源 URL。
> - 上述内容适用于 CentOS 7.2 镜像，如果是其他镜像，请根据需要修改实例自定义脚本。
> - 您也可以使用 `cloud config` 类脚本更改 yum 源设置，但是不够灵活，不能适配阿里云对部分 yum 源进行预配置的情况。建议大家使用 `script` 类的脚本修改 yum 源设置。

3. 根据需要完成 安全设置。
4. 完成上述配置后，再单击 立即购买，并按页面指示开通实例。

实例购买完成后，您就可以登录实例查看具体的效果，如下图所示。

```
[root@iZwz99v9qbmmk2dswgnzg8Z yum.repos.d]# cat /etc/resolv.conf
nameserver 8.8.8.8
[root@iZwz99v9qbmmk2dswgnzg8Z yum.repos.d]# ping www.baidu.com
PING www.a.shifen.com (103.235.46.39) 56(84) bytes of data.
64 bytes from 103.235.46.39: icmp_seq=1 ttl=48 time=73.3 ms
^C64 bytes from 103.235.46.39: icmp_seq=2 ttl=48 time=74.8 ms

--- www.a.shifen.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 73.393/74.113/74.833/0.720 ms
[root@iZwz99v9qbmmk2dswgnzg8Z yum.repos.d]# cat /etc/ntp.conf
server ntp1.aliyun.com
[root@iZwz99v9qbmmk2dswgnzg8Z yum.repos.d]# systemctl status ntpd.service
● ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2017-03-13 11:08:11 CST; 1min 58s ago
  Process: 6235 ExecStart=/usr/sbin/ntpd -u ntp:ntp $OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 6237 (ntpd)
   CGroup: /system.slice/ntpd.service
           └─6237 /usr/sbin/ntpd -u ntp:ntp -g

Mar 13 11:08:11 iZwz99v9qbmmk2dswgnzg8Z ntpd[6237]: 0.0.0.0 c01d 0d kern kernel time sync enabled
Mar 13 11:08:11 iZwz99v9qbmmk2dswgnzg8Z ntpd[6237]: ntp_io: estimated max descriptors: 1024, initial socket boundary: 16
Mar 13 11:08:11 iZwz99v9qbmmk2dswgnzg8Z ntpd[6237]: Listen and drop on 0 v4wildcard 0.0.0.0 UDP 123
Mar 13 11:08:11 iZwz99v9qbmmk2dswgnzg8Z ntpd[6237]: Listen and drop on 1 v6wildcard :: UDP 123
Mar 13 11:08:11 iZwz99v9qbmmk2dswgnzg8Z ntpd[6237]: Listen normally on 2 lo 127.0.0.1 UDP 123
Mar 13 11:08:11 iZwz99v9qbmmk2dswgnzg8Z ntpd[6237]: Listen normally on 3 eth0 172.18.48.114 UDP 123
Mar 13 11:08:11 iZwz99v9qbmmk2dswgnzg8Z ntpd[6237]: Listening on routing socket on fd #20 for interface updates
Mar 13 11:08:11 iZwz99v9qbmmk2dswgnzg8Z ntpd[6237]: 0.0.0.0 c016 06 restart
Mar 13 11:08:11 iZwz99v9qbmmk2dswgnzg8Z ntpd[6237]: 0.0.0.0 c012 02 freq_set kernel 0.000 PPM
Mar 13 11:08:11 iZwz99v9qbmmk2dswgnzg8Z ntpd[6237]: 0.0.0.0 c011 01 freq_not_set
[root@iZwz99v9qbmmk2dswgnzg8Z yum.repos.d]# cat /etc/yum.repos.d/myrepo.repo
[base]
name=myrepo
baseurl=http://mirror.centos.org/centos
gpgcheck=0
enabled=1
[root@iZwz99v9qbmmk2dswgnzg8Z yum.repos.d]#
```

由上图可知，您已经成功自定义了 DNS 服务、NTP 服务和 yum 源。

5.2 自定义实例的管理员账号

实例自定义脚本是阿里云 ECS 为用户提供的一种自定义实例启动行为的脚本，详细信息请参考阿里云线上帮助文档：实例自定义数据。

本文档以 Linux 实例为例，说明在创建实例时，您应该怎样使用实例自定义脚本自定义实例的管理员账号。您也可以使用脚本自定义 Windows 实例的管理员账号。

场景

购买 ECS 实例时，如果您想达到如下效果，您就需要使用实例自定义脚本。

- 不使用 ECS 实例默认自带的 root 用户作为管理员。您可以在实例自定义脚本中自定义具体的禁用方式和禁用程度。
- 创建一个新的管理员账号，并自定义用户名。
- 新创建的管理员账号在管理该实例的时候只使用 SSH 密钥对进行远程登录，不使用用户密码。
- 该用户如果需要进行与管理员权限相关的操作，可在免密码的情况下使用 `sudo` 提权。

配置方法

您可以按以下步骤实现上述场景需求。

1. 登录 云服务器*ECS*管理控制台，创建一个实例，配置如下：
   - 网络类型：VPC 网络
   - 实例规格：I/O 优化的实例
   - 镜像：公共镜像的 CentOS 7.2
2. 在创建页面的 **自定义数据** 输入框中输入如下内容：

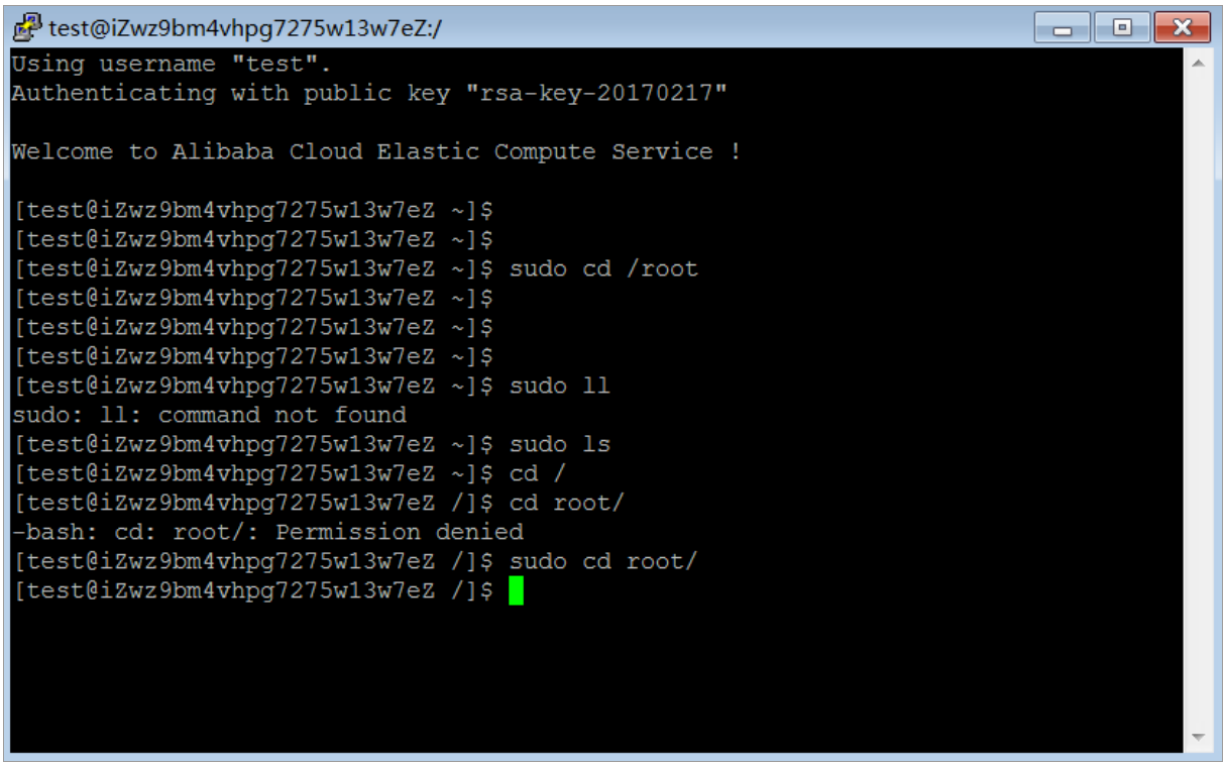
```
#!/bin/sh
useradd test
echo "test   ALL=(ALL)        NOPASSWD:ALL" | tee -a /etc/sudoers
mkdir /home/test/.ssh
touch /home/test/.ssh/authorized_keys
echo "ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEAhGqhEh/rGbIMCGItF
VtYpsXPQrCaunGJKZVIWtINrGZwusLc290qDZ93KCeb8o6X1Iby1Wm+psZY8THE+/
BsXq0M0HzfkQZD2vXuhRb4xi1z98JHskX+0jnbjqYGY+Brgai9BvKDXTTSyJtCYU
nEKxvcK+d1ZwxbNuk2QZ0ryHESDbSaczlNFgFQEDxhCrvko+zWLjTVnomVUDhdMP2g6f
Z0tgFVwkJFV0bE7oob3NOVcrx2TyhfcAjA4M2/Ry7U2MFADDC+EVkpoVDm0SOT/
hYJgaVM1xMDlSeE7kzX7yZbJLR1XAWV1xzZkNclY5w1kPnW8qMYuSwhpXzt4gsF0w==
rsa-key-20170217" | tee -a /home/test/.ssh/authorized_keys
```

> **说明：**
>
> - 第一行必须是 `#!/bin/sh`，前面不能带空格。
> - 全文不要有多余的空格和回车。
> - 最后一行的密钥为您的公钥，您可以自定义。
> - 如果需要做其他的配置，可以直接在脚本中添加。

- 示例脚本仅限于 CentOS 7.2 镜像，其他镜像请根据操作系统类型进行自定义修改。

3. 在 **安全设置** 中选择 **创建后设置**。
4. 完成上述配置后，再单击 **立即购买**，并按页面指示开通实例。

实例购买完成后，您可以使用自定义的 **test** 用户通过 SSH 私钥登录到实例中，同时也可以使用 `sudo` 提权，并执行各种需要管理员权限的操作，如图中示例所示。



6 FaaS 实例最佳实践

6.1 使用f1 RTL

本文描述如何使用f1 RTL（Register Transfer Level）。

> 说明：
>
> - 本文所述所有操作都必须由同一个账号在同一地域里执行。
> - 强烈建议您使用RAM用户操作FaaS实例。为了防止意外操作，您需要让RAM用户仅执行必要的操作。在操作FPGA镜像及下载时，因为您需要从指定的OSS Bucket下载原始DCP工程，所以您必须为FaaS管理账号创建一个角色，并授予临时权限，让FaaS管理账号访问指定的OSS Bucket。如果需要对IP加密，必须授予RAM用户KMS相关权限。如果需要做权限检查，必须授予查看用户资源的权限。

前提条件

- 创建f1实例，确保实例能访问公网，并且实例所在安全组中已经添加规则放行SSH（22）端口的访问。

  > 说明：
  > f1实例只能使用镜像市场的FaaS F1基础镜像。详细信息，请参见 创建*f1*实例。

- 您已经在 云服务器*ECS*管理控制台 f1实例的详情页上获取实例ID。
- 您必须先开通OSS服务，并 创建一个*OSS Bucket* 用于上传您的文件。Bucket与f1实例必须属于同一个账号、同一个地域。
- 如果需要加密服务，您还需要 开通密钥管理服务*#KMS#*。
- 使用RAM用户操作FPGA，必须完成以下操作：
  - 创建*RAM*用户 并 授权。
  - 创建*RAM*角色 并 授权。
  - 获取AccessKey ID和AccessKey Secret。

操作步骤

按以下步骤使用f1 RTL。

第 1 步. 远程连接f1实例

远程连接Linux实例。

第 2 步. 配置基础环境

运行以下脚本配置基础环境。

```
source /opt/dcp1_1/script/f1_env_set.sh
```

第 3 步. 编译工程

运行以下命令：

```
cd /opt/dcp1_1/hw/samples/dma_afu
afu_synth_setup --source hw/rtl/filelist.txt build_synth
cd build_synth/
run.sh
```

说明：

编译时间较长，请耐心等待。

第 4 步. 制作镜像

按以下步骤制作镜像：

1. 运行命令初始化 `faascmd`。

```
#如果需要，添加环境变量及运行权限
export PATH=$PATH:/opt/dcp1_1/script/
chmod +x /opt/dcp1_1/script/faascmd
# 将hereIsYourSecretId替换为您的AccessKey ID，hereIsYourSecretKey替换为您的AccessKey Secret
faascmd config --id=hereIsYourSecretId --key=hereIsYourSecretKey
# 将hereIsYourBucket换为华东1地域里OSS Bucket名称
faascmd auth --bucket=hereIsYourBucket
```

2. 确认在/opt/dcp1_1/hw/samples/dma_afu目录下，运行以下命令上传gbs文件。

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

3. 运行以下命令制作镜像。

```
# 将hereIsYourImageName替换为您的镜像名称
```

```
faascmd create_image --object=dma_afu.gbs --fpgatype=intel --name=
hereIsYourImageName --tags=hereIsYourImageTag --encrypted=false --
shell=V1.1
```

第 5 步. 下载镜像

按以下步骤下载镜像到f1实例：

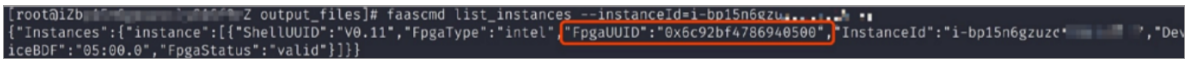
1. 查看镜像是否制作成功：运行命令 `faascmd list_images`。

   返回结果里，如果出现`"State":"success"`，表示镜像制作成功。请记录返回结果里显示的**FpgaImageUUID**，稍后会用到。

2. 运行命令获取FPGA ID。

```
# 将hereIsYourInstanceId替换为您的f1实例ID
faascmd list_instances --instanceId=hereIsYourInstanceId
```

   以下为返回结果。请记录FpgaUUID。



3. 运行命令下载FPGA镜像到f1实例。

```
# 将hereIsYourInstanceID替换为刚刚保存的实例ID；将hereIsFpgaUUID替换为上一
条命令中记下的FpgaUUID；将hereIsImageUUID替换为上一步记下FpgaImageUUID
faascmd download_image --instanceId=hereIsYourInstanceID --fpgauuid
=hereIsFpgaUUID --fpgatype=intel --imageuuid=hereIsImageUUID --
imagetype=afu --shell=V0.11
```

4. 运行命令检查是否下载成功。

```
# 将hereIsYourInstanceID替换为刚刚保存的实例ID；将hereIsFpgaUUID替换为上一
条命令中记下的FpgaUUID；
faascmd fpga_status --instanceId=hereIsYourInstanceID --fpgauuid=
hereIsFpgaUUID
```

   如果返回结果里出现`"TaskStatus":"operating"`时，且FpgaImageUUID和下载镜像时的FpgaImageUUID一致，说明下载成功。

第 6 步. 测试

依次运行以下命令。

```
cd /opt/dcp1_1/hw/samples/dma_afu/sw
make
sudo LD_LIBRARY_PATH=/opt/dcp1_1/hw/samples/dma_afu/sw:$LD_LIBRARY
_PATH ./fpga_dma_test 0
```

如果您看到如图所示的输出结果，说明测试完成。

```
[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!
```

说明：

如果没有开启Huge pages，运行以下命令启用Huge pages。

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

6.2 f1实例OpenCL开发最佳实践

本文介绍如何在f1实例上使用OpenCL（Open Computing Language）制作镜像文件，并烧写到FPGA芯片中。

说明：

- 本文所述所有操作都必须由同一个账号在同一地域里执行。
- 强烈建议您使用RAM用户操作FaaS实例。为了防止意外操作，您需要让RAM用户仅执行必要的操作。在操作FPGA镜像及下载时，因为您需要从指定的OSS Bucket下载原始DCP工程，所以您必须为FaaS管理账号创建一个角色，并授予临时权限，让FaaS管理账号访问指定的OSS

Bucket。如果需要对IP加密，必须授予RAM用户KMS相关权限。如果需要做权限检查，必须授予查看用户资源的权限。

前提条件

- 创建f1实例，确认实例能访问公网，并且实例所在安全组中已经添加规则放行SSH（22）端口的访问。

  说明：

  f1实例只能使用镜像市场的FaaS F1基础镜像。详细信息，请参见 创建*f1*实例。
- 您已经在 云服务器*ECS*管理控制台 f1实例的详情页上获取实例ID。
- 您必须先开通OSS服务，并 创建一个*OSS Bucket* 用于上传您的文件。Bucket与f1实例必须属于同一个账号、同一个地域。
- 如果需要加密文件，开通密钥管理服务（KMS）。
- 使用RAM用户操作FPGA，必须完成以下操作：
  - 创建*RAM*用户 并 授权。
  - 创建*RAM*角色 并 授权。
  - 获取AccessKey ID和AccessKey Secret。

操作步骤

按以下步骤在f1实例上使用OpenCL Example制作镜像文件，并烧写到FPGA芯片中。

第 1 步. 远程连接实例

远程连接*Linux*实例。

第 2 步. 安装基础环境

运行以下脚本安装基础环境。

```
source /opt/dcp1_1/script/f1_env_set.sh
```

第 3 步. 下载官方的OpenCL Example

按以下步骤下载官方的OpenCL Example。

1. 创建并切换到*/opt/tmp*目录。

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

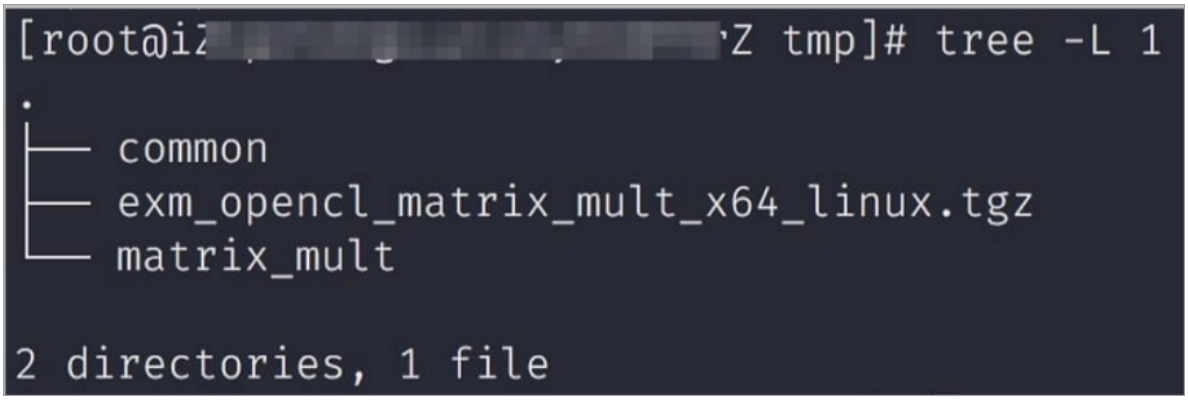
```
cd /opt/tmp
```

此时，您在*/opt/tmp*目录下。

2. 依次执行以下命令下载并解压Example文件。

```
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
```

解压后的目录如下图所示。



3. 进入*matrix_mult*目录下，执行编译命令。

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

编译过程可能会持续数个小时，您可以再开一个会话，使用`top`命令监控系统占用，确定编译状态。

第 4 步. 上传配置文件

按以下步骤上传配置文件。

1. 运行以下命令初始化faascmd。

```
# 如果需要，要添加环境变量及运行权限
export PATH=$PATH:/opt/dcp1_1/script/
chmod +x /opt/dcp1_1/script/faascmd
# 将hereIsYourSecretId换为您的AccessKey ID，hereIsYourSecretKey替换为您
的AccessKey Secret
faascmd config --id=hereIsYourSecretId --key=hereIsYourSecretKey
# 将hereIsYourBucket换为华东1OSS的Bucket名称
```

```
faascmd auth --bucket=hereIsYourBucket
```

2. 进入matrix_mult/output_files，上传配置文件。

```
cd matrix_mult/output_files # 此时您应该在/opt/tmp/matrix_mult/
matrix_mult/output_files
faascmd upload_object --object=afu_fit.gbs --file=afu_fit.gbs
```

3. 使用gbs制作FPGA镜像。

```
# 将hereIsYourImageName换为您的镜象名，将hereIsYourImageTag替换为您的镜像
标签
faascmd create_image --object=dma_afu.gbs --fpgatype=intel --name=
hereIsYourImageName --tags=hereIsYourImageTag --encrypted=false --
shell=V1.1
```

4. 查看镜像是否制作成功：运行命令`faascmd list_images`。 返回结果里，如果显示`"State":"success"`，表示镜像制作成功。请记录返回结果里显示的**FpgaImageUUID**，稍后会用到。

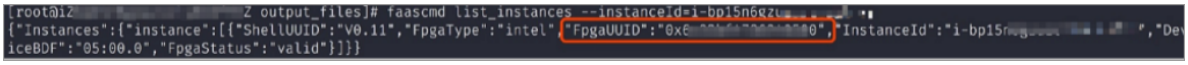
第 5 步. 下载镜像到f1实例

按以下步骤将镜像下载到f1实例。

1. 运行命令获取FPGA ID。

```
# 将hereIsYourInstanceId替换为您的FPGA实例ID
faascmd list_instances --instanceId=hereIsYourInstanceId
```

以下为返回结果。请记录FpgaUUID。



2. 运行命令下载镜像到f1实例。

```
# 将hereIsYourInstanceID替换为刚刚保存的实例ID；将hereIsFpgaUUID替换为上一
条命令中记下的FpgaUUID；将hereIsImageUUID替换为上一步记下的FpgaImageUUID
faascmd download_image  --instanceId=hereIsYourInstanceID --fpgauuid
=hereIsFpgaUUID --fpgatype=intel --imageuuid=hereIsImageUUID --
imagetype=afu --shell=V0.11
```

3. 运行命令检查是否下载成功。

```
# 将hereIsYourInstanceID替换为刚刚保存的实例ID；将hereIsFpgaUUID替换为上一
条命令中记下的FpgaUUID；
```

```
faascmd fpga_status --fpgauuid=hereIsFpgaUUID --instanceId=
hereIsYourInstanceID
```

如果返回结果里显示`"TaskStatus":"operating"`，说明下载成功。

第 6 步. 将FPGA镜像烧录到FPGA芯片

按以下步骤将FPGA镜像烧录到FPGA芯片。

1. 打开第2步环境的窗口。如果已关闭，重新执行第2步操作。
2. 运行命令配置OpenCL的运行环境。

```
sh /opt/dcp1_1/opencl/opencl_bsp/linux64/libexec/setup_permissions.
sh
```

3. 返回上级目录。

```
cd ../.. # 此时您在/opt/tmp/matrix_mult
```

4. 执行编译命令。

```
make
# 输出环境配置
export CL_CONTEXT_COMPILER_MODE_ALTERA=3
cp matrix_mult.aocx ./bin/matrix_mult.aocx
cd bin
host matrix_mult.aocx
```

当您看到如下输出时，说明配置完成。请注意，最后一行必须为`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
```

6.3 f2实例OpenCL开发最佳实践

本文介绍如何在f2实例上使用OpenCL（Open Computing Language）制作镜像文件，并烧写到FPGA芯片中。

说明：

- 本文所述所有操作都必须由同一个账号在同一地域里执行。
- 强烈建议您使用RAM用户操作FaaS实例。为了防止意外操作，您需要让RAM用户仅执行必要的操作。您需要为FaaS管理账号创建一个角色，并授予临时权限，让FaaS管理账号能访问指定的OSS Bucket。

前提条件

- 创建f2实例，确保实例能访问公网，并且实例所在安全组中已经添加规则放行SSH（22）端口的访问。

  说明：

  f2实例只能使用镜像市场的FaaS F2基础镜像。详细信息，请参见 创建*f2*实例。
- 登录 云服务器*ECS*管理控制台，在f2实例的详情页上，获取实例ID。
- 开通OSS服务，并 创建一个*OSS Bucket*。Bucket与f2实例必须属于同一个账号、同一个地域。
- 使用RAM用户操作FPGA，必须完成以下操作：
  - 创建*RAM*用户 并 授权。
  - 创建*RAM*角色 并 授权。
  - 获取AccessKey ID和AccessKey Secret。

操作步骤

按以下步骤在f2实例上使用OpenCL制作镜像文件，并烧写到FPGA芯片中。

步骤 1. 配置环境

按以下步骤配置环境：

1. 远程连接*f2*实例。
2. 使用 `vim` 修改/root/xbinst_oem/setup.sh：在第5行前加一个 `#`，注释掉 `unset XILINX_SDX`，再保存退出。

```
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. 运行以下命令安装Screen，用于后续的持续链接。

```
yum install screen -y
```

4. 运行以下命令进入Screen。

```
screen -S f2opencl
```

5. 运行以下命令配置安全烧写环境。

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

步骤 2. 编译二进制文件

按以下步骤编译二进制文件：

1. 进入命令目录 。

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

2. 运行命令 `cat sdaccel.mk | grep "XDEVICE"`，查看 `XDEVICE` 配置是否为 `xilinx:aliyun-ku115-f2:4ddr-xpr:4.2`。如果不是，必须改为这个配置。

3. 使用 `vim` 修改 `common.mk` 文件。

```
vim ../common/common.mk
```

将第 63 行代码（参数可能在 60-62 行，由您的文件确定）

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

修改为

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

说明：

您必须向编译服务器提交DCP文件，而不是bit文件，所以必须添加编译参数 `--xp param:compiler.acceleratorBinaryContent=dcp`，使Xilinx® OpenCL™ Compiler（xocc）编译生成一个布局布线后的DCP文件，而不是bit文件。

4. 运行以下命令编译程序。

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

如果您看到如下界面，说明二进制文件编译已经开始。编译过程可能会持续数个小时，请您耐心等待。

```
[root@iZ                    '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
```

步骤 3. 检查打包脚本

运行以下命令检查打包脚本是否存在。

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

说明：

如果返回结果中包含 `cannot open (No such file or directory)`，说明不存在该文件，您需要手动下载打包脚本。

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

步骤 4. 制作镜像

按以下步骤制作镜像文件。

1. 运行命令配置OSS环境。

```
# 将此处的hereIsMySecretId、hereIsMySecretKey、hereIsMyBucket分别替换为您的AccessKeyID、AccessKeySecret和Bucket名称
faascmd config --id=hereIsMySecretId --key=hereIsMySecretKey
```

```
faascmd auth --bucket=hereIsMyBucket
```

2. 运行 `ls`，获取文件名。

```
[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
```

3. 打包二进制文件。

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

打包完成后，在同一目录下，您会看到一个打包好的文件，如本示例中的 `17_10_28-021904_SDAccel_Kernel.tar.gz`。

```
[root@ 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
```

4. 运行命令将打包好的文件上传到您指定的OSS Bucket中。

```
# 将文件名改为打包好的文件名，您需要根据您的 ls 命令修改
faascmd upload_object --object=bit.tar.gz --file=bit.tar.gz
```

5. 运行如下命令制作镜像。

```
# bit.tar.gz、hereIsFPGAImageName、hereIsFPGAImageTag分别替换为刚创建的
压缩包文件名、镜像名和镜像的tag
faascmd create_image --object=bit.tar.gz --fpgatype=xilinx --name=
hereIsFPGAImageName --tags=hereIsFPGAImageTag --encrypted=false --
shell=20171121
```

返回结果示例如下图所示。如果出现 `"State":"queued"`，说明这个任务已经加入队列，开始制作镜像。

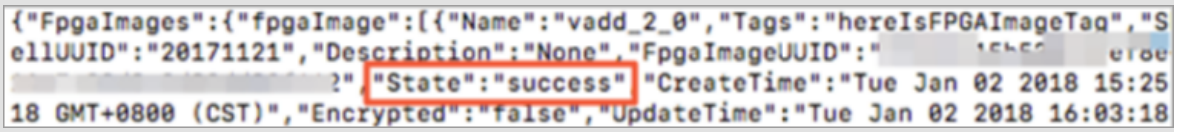
```
{"FpgaImages":{"fpgaImage":[{"Name":"vadd_2_0","Tags":"hereIsFPGAImageTag","Sh
ellUUID":"20171121","Description":"None","FpgaImageUUID":"xilinx15b530c1-ef8e-
                                    "State":"queued" "CreateTime":"Tue Jan 02 2018 15:25:1
8 GMT+0800 (CST)","Encrypted":"false","UpdateTime":"Tue Jan 02 2018 16:03:18 G
```

说明：

制作镜像比较耗时。等待一段时间后，运行以下命令，查看镜像状态。

```
faascmd list_images
```

返回结果里，如果出现 `"State":"success"`，说明镜制作成功。



记录返回结果中的FPGAImageUUID。

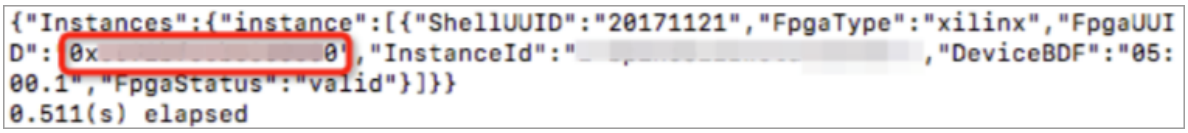
步骤 5. 烧写镜像

按以下步骤将镜像烧写入FPGA芯片。

1. 运行以下命令获取FpgaUUID。

```
# 将hereIsYourInstanceId替换为你的FPGA云服务器的实例ID
faascmd list_instances --instanceId=hereIsYourInstanceId
```

返回结果如下图所示。



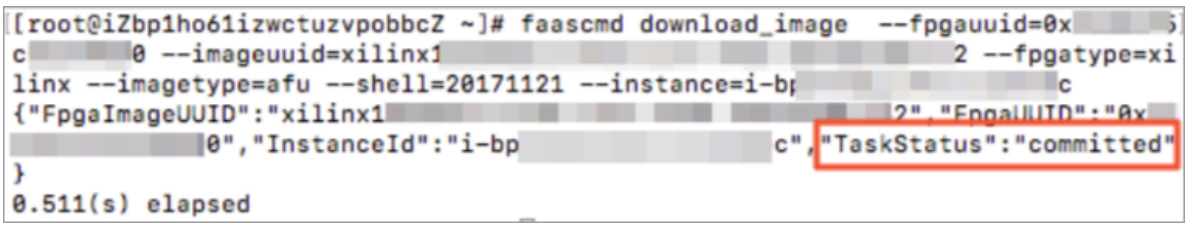
说明：

记录返回结果中的FpgaUUID。

2. 运行以下命令下载镜像。

```
# 将hereIsYourInstanceID替换为这个f2实例的ID，将hereIsFpgaUUID替换为您记录
的FpgaUUID，将hereIsImageUUID替换为您记录的FpgaImageUUID
faascmd download_image  --instanceId=hereIsYourInstanceId --fpgauuid
=hereIsFpgaUUID --fpgatype=xilinx --imageuuid=hereIsImageUUID --
imagetype=afu --shell=20171121
```

在返回结果里，如果看到 `"State":"committed"`，说明镜像下载成功。

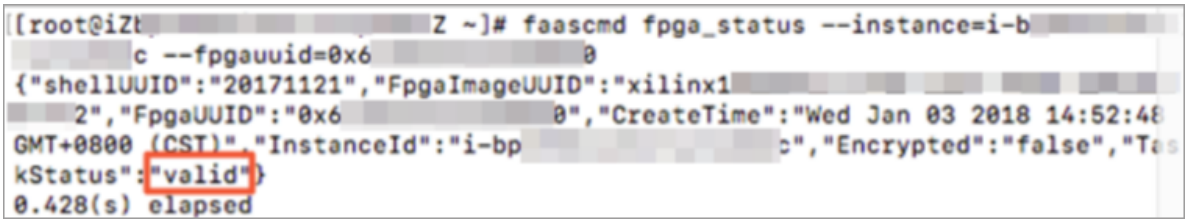


说明：

您也可以运行以下命令查看镜像是否下载成功。

```
# 将 hereIsYourInstanceID 替换为这个 f2 实例的 ID，将 hereIsFpgaUUID 替换为您记录的 FpgaUUID
faascmd fpga_status  --instanceId=hereIsYourInstanceID --fpgauuid=hereIsFpgaUUID
```

返回结果里，如果看到 `"TaskStatus":"valid"`，而且FpgaImageUUID和下载镜像时的FpgaImageUUID一致，说明镜像下载正常。



步骤 6. 运行**Host**程序

执行以下命令运行Host程序。

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

如果返回结果中出现 `Test Passed`，说明测试通过。

其他操作

这里介绍 FPGA 实例部分常用的操作。

任务	命令
查看帮助文档	`make -f ./sdaccel.mk help`
软件仿真	`make -f ./sdaccel.mk run_cpu_em`
硬件仿真	`make -f ./sdaccel.mk run_hw_em`
只编译 host 代码	`make -f ./sdaccel.mk host`
编译生成可以下载的文件	`make -f sdaccel.mk xbin_hw`
清理工作目录	`make -f sdaccel.mk clean`
强力清除工作目录	`make -f sdaccel.mk cleanall`

说明：

- sdx2017.2在仿真时，device需要用xilinx_aliyun-ku115-f2_4ddr-xpr_4_2。
- 仿真时只需要按照Xilinx标准流程操作，不需要配置F2_env_setup环境。

6.4 f3实例OpenCL开发最佳实践

本文介绍如何在f3实例上使用OpenCL（Open Computing Language）制作镜像文件，并烧写到FPGA芯片中。

说明：

- 本文所述所有操作都必须由同一个账号在同一地域里执行。
- 建议您使用RAM用户操作FaaS实例。您需要为FaaS管理账号创建一个角色，并授予临时权限，让FaaS管理账号能访问指定的OSS Bucket。

前提条件

- 已创建*f3*实例。

  说明：

  - f3实例只能使用我们共享给您的镜像。
  - 创建实例时选择分配公网IP，确保实例能访问公网。
  - 实例所在安全组中已经添加规则放行SSH（22）端口的访问。
- 已在ECS控制台f3实例的详情页上，获取实例ID。
- 使用同一个账号创建了与f3实例在同一地域的OSS Bucket。详细信息参见开通OSS服务和 创建一个*OSS Bucket*。
- 如果您使用RAM用户操作FPGA，确保已经完成以下操作：
  - 创建*RAM*用户并授权。
  - 创建*RAM*角色并授权。
  - 获取AccessKey ID和AccessKey Secret。

操作步骤

按以下步骤在f3实例上使用OpenCL制作镜像文件，并烧写到FPGA芯片中。

步骤 1. 配置环境

按以下步骤配置环境：

1. 远程连接*f3*实例。
2. 执行以下命令打开安装脚本，并在第5行前加上#，注释掉 `unset XILINX_SDX`，再保存退出。

```
vim /root/xbinst_oem/setup.sh
```

```
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. 运行以下命令安装Screen。

```
yum install screen -y
```

4. 运行以下命令进入Screen。

```
screen -S f3opencl
```

5. 运行以下命令配置安全烧写环境。

```
source /root/xbinst_oem/f3_env_setup.sh xocl
```

步骤 2. 编译二进制文件

按以下步骤编译二进制文件：

1. 进入命令目录。

```
cd /opt/Xilinx/SDx/2017.4.op/examples/vadd
```

2. 运行命令`cat sdaccel.mk | grep "XDEVICE="`，确保XDEVICE配置为 `xilinx_aliyun-f3_dynamic_5_0`。

3. 使用 `vim` 修改 `common.mk` 文件。

```
vim ../common/common.mk
```

将如下所示第 63 行代码（参数可能在 60-62 行，视您的文件而定）

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

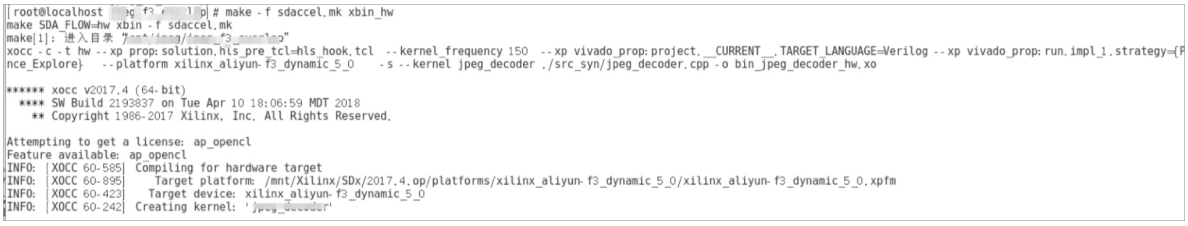
修改为：

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

4. 运行以下命令编译程序。

```
export XILINX_SDX=/opt/Xilinx/SDx/2017.4.op
make -f sdaccel.mk xbin_hw
```

如果您看到如下界面，说明二进制文件编译已经开始。编译过程可能会持续数个小时，请您耐心等待。



步骤 3. 检查打包脚本

运行以下命令检查打包脚本是否存在。

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

如果返回结果中包含 `cannot open (No such file or directory)`，说明不存在该文件，您需要运行以下命令手动下载打包脚本。

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

步骤 4. 制作镜像

按以下步骤制作镜像文件。

1. 运行命令配置OSS环境。

```
# 将此处的hereIsMySecretId、hereIsMySecretKey、hereIsMyBucket分别替换为
您的AccessKeyID、AccessKeySecret和Bucket名称
faascmd config --id=hereIsMySecretId --key=hereIsMySecretKey
faascmd auth --bucket=hereIsMyBucket
```

2. 运行 `ls`，获取后缀为`.xclbin`的文件名。

```
[root@ 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
```

3. 打包二进制文件。

```
/root/xbinst_oem/sdaccel_package.sh -xclbin=/opt/Xilinx/SDx/2017.4.
op/examples/vadd/bin_vadd_hw.xclbin
```

打包完成后，在同一目录下，您会看到一个打包好的文件，如下图所示。

```
[root@ vadd]# ls
17_10_28-021904-primary.bit          krnl_vadd.cl
_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
```

步骤 5. 烧写镜像

您可以采用脚本化流程或者单步操作流程来上传网表文件，并下载FPGA镜像。

- 脚本化流程：仅适用于配备单块FPGA卡的f3实例。

1. 运行以下命令上传并生成镜像文件。

```
sh /root/xbinst_oem/tool/faas_upload_and_create_image.sh
```

```
[root@iZ                Z tool]# ./faas_upload_and_create_image.sh 18_05_05-190152_SDAccel_Kernel.tar.gz
18_05_05-190152_SDAccel_Kernel.tar.gz
i-uf64dyi6ps4n662o9dsl
18_05_05-190152_SDAccel_Kernel.tar.gz
18_05_05-190152_SDAccel_Kernel.tar.gz
4.980(s) elapsed
0.158(s) elapsed
FPGAImageUUID has been copied to createimageoutput.txt
image uuid: xilinx6                    01
0.037(s) elapsed
```

2. 下载镜像文件。

```
sh /root/xbinst_oem/tool/faas_download_image.sh  0 # 最后的数字为实例中FPGA的序号
```

> 说明：
> 0为FaaS实例中的第一个FPGA，单芯片实例序号一律为0，对多芯片实例，例如4芯片的序号为0,1,2,3。

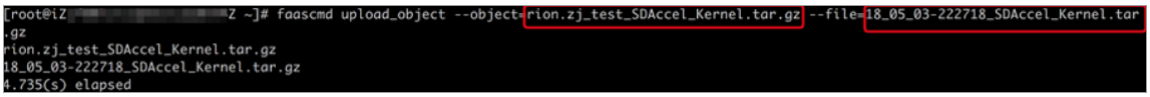
如果需要对多个FPGA下载同一个镜像，可以在末尾添加序号，例如：

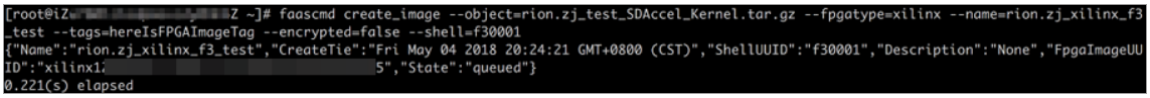
```
sh faas_download_image.sh bit.tar.gz 0 1 2
```

- 单步操作流程：

1. 运行以下命令，将压缩包上传到您个人的OSS Bucket，再将存放在您个人OSS Bucket中的gbs上传到FaaS管理单元的OSS Bucket中。

```
faascmd upload_object --object=bit.tar.gz --file=bit.tar.gz
faascmd create_image --object=bit.tar.gz --fpgatype=xilinx --name=hereIsFPGAImageName --tags=hereIsFPGAImageTag --encrypted=false --shell=f30001
upload_object示例
```





2. 运行命令查看FPGA镜像是否处于可下载状态。

```
faascmd list_images
```

在返回结果中，如果看到 `"State":"success"`，表示FPGA镜像已经可以下载。找到并记录FpgaImageUUID。

```
[root@iZ                    Z ~]# faascmd list_images
{
  "FpgaImages": {
    "fpgaImage": [
      {
        "CreateTime": "Fri May 04 2018 20:24:21 GMT+0800 (CST)",
        "Description": "None",
        "Encrypted": "false",
        "FpgaImageUUID": "xilir                                    5",
        "Name": "                    ",
        "ShellUUID": "f30001",
        "State": "success",
        "Tags": "hereIsFPGAImageTag",
        "UpdateTime": "Fri May 04 2018 21:01:48 GMT+0800 (CST)"
      },
```

3. 运行以下命令，在返回结果中，找到并记录FpgaUUID。

```
faascmd list_instances --instanceId=hereIsYourInstanceId # 将
hereIsYourInstanceId替换为f3实例ID
```

4. 运行以下命令下载FPGA镜像。

```
faascmd download_image --instanceId=hereIsYourInstanceId --
fpgauuid=hereIsFpgaUUID --fpgatype=xilinx --imageuuid=hereIsImag
eUUID --imagetype=afu --shell=f30001
# hereIsYourInstanceId替换为f3的实例ID，hereIsFpgaUUID替换为您获取的
FpgaUUID，hereIsImageUUID替换为您获取的FpgaImageUUID
```

```
[root@iZ                    4Z ~]# faascmd download_image  --instanceId=i-u                    4 --fpgauuid=0x                    0 --fpgatype=xilinx
 --imageuuid=xilinx12                    5 --imagetype=afu --shell=f30001
{"FpgaImageUUID":"xilinx12                    5","FpgaUUID":"0x                    0","InstanceId":"i-u                    4","TaskStat
us":"committed"}
0.223(s) elapsed
```

5. 运行以下命令查看镜像是否下载成功。

```
faascmd fpga_status --fpgauuid=hereIsFpgaUUID --instanceId=
hereIsYourInstanceId # hereIsFpgaUUID替换为您获取的FpgaUUID，
hereIsYourInstanceId替换为f3实例ID。
```

以下为返回结果示例。如果显示的FpgaImageUUID与您获取的FpgaImageUUID一致，并且显示 `"TaskStatus":"valid"`，说明镜像下载成功。

```
[root@iZu                    Z ~]# faascmd fpga_status --fpgauuid=0xe                    0 --instanceId=i-u                    4
{"shellUUID":"f30001","FpgaImageUUID":"xilinx1                    5","FpgaUUID":"0xe                    0","InstanceId":"i-u                    p
                    4","CreateTime":"Fri May 04 2018 21:25:53 GMT+0800 (CST)","TaskStatus":"valid","Encrypted":"false"}
0.263(s) elapsed
```

步骤 6. 运行Host程序

执行以下命令运行Host程序。

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

如果返回结果中出现 `Test Passed`，说明测试通过。

其他操作

这里介绍 FPGA 实例部分常用的操作。

任务	命令
查看帮助文档	`make -f ./sdaccel.mk help`
软件仿真	`make -f ./sdaccel.mk run_cpu_em`
硬件仿真	`make -f ./sdaccel.mk run_hw_em`
只编译 host 代码	`make -f ./sdaccel.mk host`
编译生成可以下载的文件	`make -f sdaccel.mk xbin_hw`
清理工作目录	`make -f sdaccel.mk clean`
强力清除工作目录	`make -f sdaccel.mk cleanall`

说明：

仿真时只需要按照Xilinx标准流程操作，不需要配置f3_env_setup环境。

6.5 f3 RTL开发最佳实践

本文描述基于f3的RTL（Register Transfer Level）开发流程。

说明：

- 本文所述所有操作必须由同一个账号在同一个地域执行。
- 强烈建议您使用RAM用户操作FPGA实例。为了防止意外操作，您需要让RAM用户仅执行必要的操作。在操作及下载FPGA镜像时，因为您需要从指定的OSS存储空间下载原始DCP工程，所以您需要为FaaS账号创建一个角色，并授予临时权限，让FaaS管理账号可以访问指定的OSS存储空间。如果需要对IP加密，必须授予RAM用户KMS相关的权限。如果需要做权限检查，必须授予查看用户资源的权限。

前提条件

- 您已经 创建f3实例，实例能访问公网，并且实例所在安全组中已经添加规则放行SSH（22）端口的访问。
- 登录 云服务器ECS管理控制台，在f3实例的详情页上，获取实例ID。
- 在华东2 创建一个OSS Bucket，专门用于FaaS服务。

  说明：
  这个Bucket会对FaaS管理账号开通读写权限，因此不建议您存储与FaaS无关的内容。
- 如果使用RAM用户操作FPGA，必须完成以下操作：
  - 创建RAM用户 并 授权。
  - 创建RAM角色 并 授权。
  - 获取AccessKey ID和AccessKey Secret。

操作步骤

1. 远程连接Linux实例。

   说明：
   编译工程时需要 2 ~ 3 小时。建议您使用nohup或者VNC连接实例，以免编译时意外退出。
2. 下载 RTL参考设计。
3. 解压文件。
4. 运行以下脚本配置f3环境。

   ```
   source /root/xbinst_oem/F3_env_setup.sh xdma
   ```

   说明：
   每打开一个terminal窗口都需要运行此命令。
5. 指定OSS存储空间。

   ```
   faascmd config --id=hereIsYourSecretId --key=hereIsYourSecretKey #将hereIsYourSecretId和hereIsYourSecretKey替换为您的RAM用户AK信息
   faascmd auth --bucket=hereIsYourBucket # 将hereIsYourBucket替换为您创建的OSS Bucket名称
   ```
6. 运行以下命令编译RTL工程。

   ```
   cd <您之前解压的路径>/hw/ # 进入解压后的hw路径
   ```

```
sh compiling.sh
```

说明：

编译工程需要2 ~ 3小时。

7. 上传网表文件，并下载FPGA镜像。您可以采用脚本化流程或者单步操作流程完成该步骤。

- 脚本化流程：仅适用于配备单块FPGA卡的f3实例。

1. 运行以下命令上传并生成镜像文件。

```
sh /root/xbinst_oem/tool/faas_upload_and_create_image.sh <bit.
tar.gz需要上传的压缩包文件名>
```

2. 下载镜像文件。

```
sh /root/xbinst_oem/tool/faas_download_image.sh  0 # 最后的数字为
实例中fpga的序号
```

0为FaaS实例中的第一个FPGA，单芯片实例序号一律为0，对多芯片实例，例如4芯片的序号为0,1,2,3。

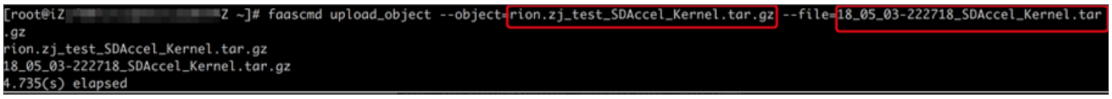
如果需要对多个FPGA下载同一个镜像，可以在末尾添加序号，例如：

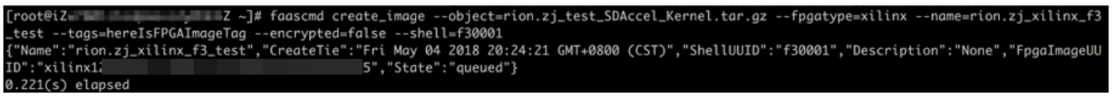
```
sh faas_download_image.sh bit.tar.gz 0 1 2
```

- 单步操作流程：

1. 运行以下命令，将压缩包上传到您个人的OSS Bucket，再将存放在您个人OSS Bucket中的gbs上传到FaaS管理单元的OSS Bucket中。

```
faascmd upload_object --object=bit.tar.gz --file=bit.tar.gz
faascmd create_image --object=bit.tar.gz --fpgatype=xilinx --
name=hereIsFPGAImageName --tags=hereIsFPGAImageTag --encrypted=
false --shell=f30001
```

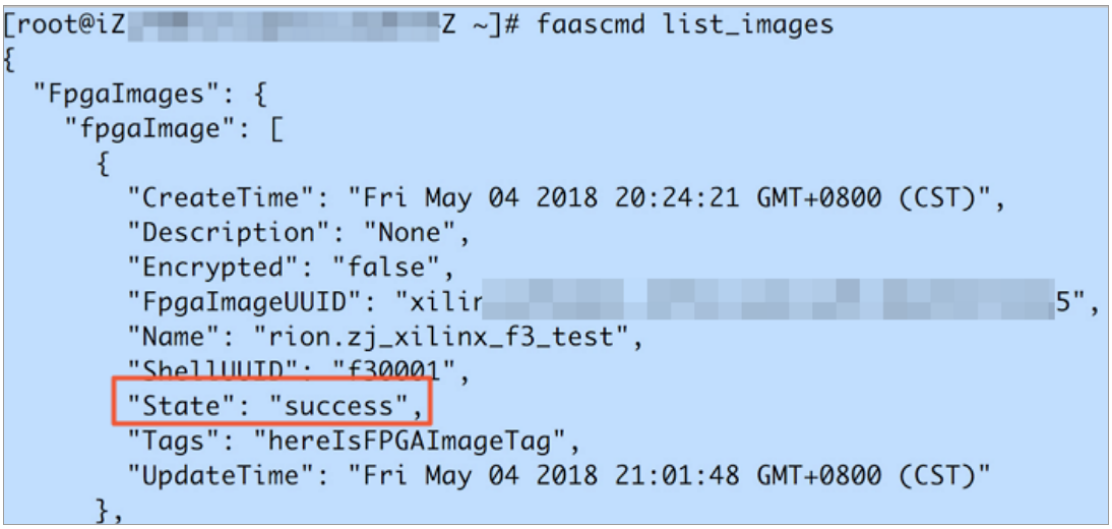




2. 运行命令查看FPGA镜像是否处于可下载状态。

```
faascmd list_images
```

在返回结果中，如果看到 `"State":"success"`，表示FPGA镜像已经可以下载。找到并记录FpgaImageUUID。

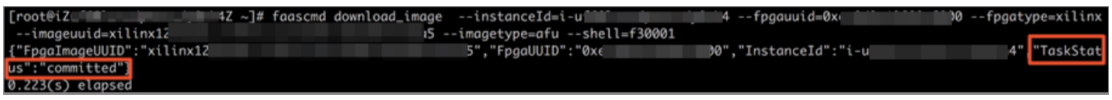


3. 运行以下命令，在返回结果中，找到并记录FpgaUUID。

```
faascmd list_instances --instanceId=hereIsYourInstanceId # 将hereIsYourInstanceId替换为f3实例ID
```

4. 运行以下命令下载FPGA镜像。

```
faascmd download_image --instanceId=hereIsYourInstanceId
 --fpgauuid=hereIsFpgaUUID --fpgatype=xilinx --imageuuid=
hereIsImageUUID --imagetype=afu --shell=f30001
# hereIsYourInstanceId替换为f3的实例ID，hereIsFpgaUUID替换为您获取的FpgaUUID，hereIsImageUUID替换为您获取的FpgaImageUUID
```



5. 运行以下命令查看镜像是否下载成功。

```
faascmd fpga_status --fpgauuid=hereIsFpgaUUID --instanceId=
hereIsYourInstanceId # hereIsFpgaUUID替换为您获取的FpgaUUID，
hereIsYourInstanceId替换为f3实例ID。
```

以下为返回结果示例。如果显示的FpgaImageUUID与您获取的FpgaImageUUID一致，并且显示 `"TaskStatus":"valid"`，说明镜像下载成功。

FAQ

上传镜像时出现异常，如何查看异常详情？

如果您的工程在上传生成镜像的过程中出现异常，例如云上编译服务器编译报错，你可以通过以下两种方式来查看异常详情：

- 查看faas_compiling.log。使用上传脚本faas_upload_and_create_image.sh时，如果编译失败会自动下载并打印faas_compiling.log到terminal中。
- 手动执行命令查看编译log文件：`sh /root/xbinst_oem/tool/faas_checklog.sh <bit.tar.gz之前上传的压缩包文件名>`

如何重新加载镜像？

您可以参考以下步骤重新加载镜像：

1. 在实例中运行以下命令卸载驱动。

```
sudo rmmod xdma
sudo rmmod xocl
```

2. 下载镜像。可以使用以下两种方式之一：
   - 使用脚本，最后的数字为实例中FPGA的序号：`sh faas_download_image.sh bit.tar.gz 0`
   - 使用faascmd：`faascmd download_image --instanceId=hereIsYourInstanceId --fpgauuid=hereIsFpgaUUID --fpgatype=xilinx --imageuuid=hereIsImageUUID --imagetype=afu --shell=f30001`
3. 安装驱动。

```
sudo depmod
sudo modprobe xdma
```

7 P2V 迁云实践

7.1 什么是迁云工具和 P2V

阿里云自主研发的迁云工具平衡了 ECS 用户的线上线下服务器负载或者各种不同云平台之间的负载。以其轻巧便捷的特点，迁云工具支持在线迁移物理机服务器、虚拟机以及其他云平台云主机至 ECS 经典网络平台或专有网络平台，实现统一部署资源的目的。

迁云工具属于 P2V 或者 V2V 工具范畴。P2V（Physical to virtual）代表从物理 IDC 环境迁移到 ECS，V2V（Virtual to virtual）代表从虚拟机环境或者云平台主机迁移到 ECS。迁云工具能将计算机磁盘中的操作系统、应用程序以及应用数据等迁移到 ECS 或是虚拟磁盘分区中生成 ECS 镜像，您可以使用该镜像快速创建 ECS 实例，以实现 P2V 和 V2V。

适用的操作系统

迁云工具适用于以下操作系统（32 位或 64 位均可）的物理机服务器、虚拟机和其他云平台云主机。

Windows	Linux
• Windows Server 2003 • Windows Server 2008 • Windows Server 2012 • Windows Server 2016	• CentOS 5/6/7 • Ubuntu 10/12/14/16/17 • Debian 7/8/9 • Red Hat 5/6/7 • SUSE 11.4/12.1/12.2 • OpenSUSE 13.1 • Gentoo 13.0

如果您使用的操作系统没有包含在上述列表中，请认真阅读 使用迁云工具迁移服务器至阿里云 并谨慎操作。

计费详情

迁云工具是免费工具，不收取额外的费用。但是，在迁云过程中会涉及少量资源计费：

- 迁云过程中，会创建快照以生成自定义镜像，该快照会按照实际占用容量收取少部分费用。详情请参阅 快照服务费用细则。

- 迁云时，系统默认在您的阿里云账号下创建一个默认名为 INSTANCE_FOR_GOTOALIYUN 的 ECS 实例做中转站。该中转实例付费类型为按量付费，您需要确保账号余额大于等于 100 元。按量付费实例产生的资源耗费及计费说明请参阅 按量付费。

  说明：

  迁云失败后，该实例保留在 ECS 控制台，便于重新迁云。如果您不再需要该实例，请自行 释放实例 以免造成不必要的扣费。

参考链接

- 迁云工具不仅能实现在线迁移物理机服务器、虚拟机以及其他云平台云主机，还可以为 ECS 用户提供缩容磁盘的功能。更多详情，请参阅 磁盘缩容。
- 目前，ECS 支持的 P2V 或 V2V 迁云的方式除迁云工具外，还可以 导入镜像。
- 如果您有数据库迁云需求，请访问 数据迁移。
- 迁云工具操作视频示例。

更新历史

下表为迁云工具的版本更新信息。

更新时间	版本	描述
2018/08/29	1.3.0	• 提速迁云进程并优化一些已知问题 • 增加 Windows 服务器修复环节，您无需手动运行文件权限重置工具
2018/07/04	1.2.9.5	• 支持迁移 Ubuntu 17 服务器 • 优化迁云服务端功能，修复和完善个别细微问题
2018/06/11	1.2.9	• 增加 Windows GUI 简易界面版本 • 修复 Windows 数据盘过滤文件默认不存在问题
2018/04/28	1.2.8	• 增加命令行参数选项，您可以在工具所在路径运行 `--help` 查看详情。 • 支持从专线 VPC 私有网络迁移上云，保障数据安全
2018/04/03	1.2.6	• 修复 Linux 服务器数据盘上级目录重复拷贝子目录数据的问题 • 增加文件传输参数选项
2018/03/07	1.2.3	• 修复 Linux 服务器界面服务启动异常问题 • 修复提示服务实例磁盘空间可能不足问题

更新时间	版本	描述
		• 支持 Ubuntu 10 系统
2018/02/08	1.2.1	• 优化文件传输信息的显示 • 支持临时关闭 Linux 服务器的 SELinux，无需重开机源服务器
2018/01/18	1.2.0	• 拓展资源支撑，支持迁移更多类型资源 • 提升创建镜像的效率和稳定性
2018/01/11	1.1.8	• 支持 SUSE 12 SP2 系统 • 优化连接速度 • 优化日志信息提示 • 修复 NetworkManager 网络问题
2017/12/21	1.1.7	• 支持 SUSE 12 SP1 系统 • 新增限制数据传输带宽的功能
2017/12/14	1.1.6	• 新增版本更新提示功能 • 修复数据传输 6144 错误 • 自动检查用户配置文件 user_config.json 中请求参数的正确性
2017/12/08	1.1.5	• 修复 Linux 服务器数据盘路径问题 • 优化日志信息提示
2017/12/01	1.1.3	支持 Debian 系统

7.2 使用迁云工具迁移服务器至阿里云

本文描述如何使用迁云工具迁移 IDC 服务器、虚拟机或者云主机到阿里云。如果您有数据库迁云需求，请访问 数据迁移。

注意事项

使用迁云工具前，您需要注意：

- 确保系统本地时间与实际时间一致，否则会报错 IllegalTimestamp 异常。
- 待迁云的源服务器必须能够访问公网，且防火墙入方向必须放行下列通信端口以访问相关公网服务：
  - 通过 HTTP 80 端口访问 ECS 主接入地址 `http://ecs.aliyuncs.com`。更多详情，请参阅 接入地址。

  - 通过 HTTP 80 端口访问 VPC `http://vpc.aliyuncs.com`。
  - 通过 HTTPS 443 端口访问 STS `https://sts.aliyuncs.com`。
  - 通过 8080 和 8703 代理端口访问中转实例的公网 IP 地址。
- 迁云工具暂不支持迁移增量数据。对于源服务器上需要保持数据完整的业务，您可以选择一个业务空闲时段，暂时停止这些业务，再迁移数据。
- 迁云工具会在您的云账号下创建一台临时中转实例，将源服务器系统数据传输到中转实例。为避免迁云失败，请勿停止、重启或者释放中转实例。迁云完成后，该中转实例会自动释放。
- 如果您使用的是 RAM 子账号，请确保您已被授权云服务器 ECS `AliyunECSFullAccess` 权限和专有网络 VPC `AliyunVPCFullAccess` 权限。更多详情，参阅 *RAM* 文档 授权策略管理。
- 如果您的源服务器中挂载了共享存储设备，迁云时可以做如下处理：
  - 默认行为：
    - Windows 服务器：迁云工具默认将挂载在 C 盘驱动中的共享存储部分的数据合并为系统盘数据并上传。
    - Linux 服务器：迁云工具默认将共享存储部分的数据合并为系统盘数据并上传。
  - 自定义行为：
    - 您可以设置共享存储的挂载路径（`src_path`）为一个数据盘，将共享存储当作单独的数据盘迁移上云。
    - 或者，您可以过滤共享存储的数据，过滤后共享存储不会被迁移上云。

Linux 服务器注意事项

当您的源服务器为 Linux 系统时，会有以下额外要求：

- 源服务器必须已经安装了 Rsync 库：
  - CentOS：运行 `yum install rsync -y`。
  - Ubuntu：运行 `apt-get install rsync -y`。
  - Debian：运行 `apt-get install rsync -y`。
  - 其他发行版：参考发行版官网安装相关的文档。
- 确保源服务器已关闭 SELinux。您可以运行 `setenforce 0` 临时关闭 SELinux。同时，建议您在 /etc/selinux/config 中，设置 `SELINUX=disabled`，禁用 SELinux。
- 确保源服务器已 安装 *Virtio#KVM#*驱动。

- 对于 CentOS 5、Red Hat 5 和 Debian 7 等系统，需要 安装 *1.9* 以上版本的系统引导程序 *GRUB* 。

前提条件

您的云账号必须已经开通快照服务，您可以在*ECS*管理控制台开通快照服务。

步骤 1：下载并安装迁云工具

1. 下载迁云工具压缩包，解压后包含的文件列表如下：

   表 7-1: Windows 服务器

文件（夹）名	描述
Excludes 文件夹	过滤筛选文件夹，设置不迁云的路径，默认包含 rsync_excludes_win.txt。
client_data	迁云过程中的数据文件，包含ECS 中转实例信息、迁移进度等。
user_config.json	源服务器信息配置文件。
go2aliyun_gui.exe	迁云工具 Windows GUI 版本主程序，详情请参阅 迁云工具 *Windows GUI* 版本介绍。
go2aliyun_client.exe	迁云工具命令行版本主程序。

   表 7-2: Linux 服务器

文件（夹）名	描述
Check	检测工具文件夹，默认包含辅助程序 client_check。
client_data	迁云过程中的数据文件。
user_config.json	源服务器信息配置文件。
Excludes文件夹	过滤筛选文件夹，设置不迁云的路径，默认包含 rsync_excludes_linux.txt 。
go2aliyun_client	迁云工具主程序。

2. 登录待迁云的服务器、虚拟机或者云主机。
3. 将下载的迁云工具压缩包解压到指定的目录。

步骤 2：编辑 **user_config.json** 文件

user_config.json 是一份以 JSON 语言编写的配置文件，位于迁云工具所在路径中。user_config.json 主要包含源服务器的必要配置信息，例如，AccessKey 和目标自定义镜像的配置信息等。

> 说明：
>
> 如果您使用的 Windows GUI 版本主程序，您可以在 GUI 界面完成 user_config 配置。更多详情，请参阅 迁云工具 *Windows GUI* 版本介绍。

1. 在迁云工具路径中使用编辑器打开 user_config.json 文件。以下为文件初始状态：

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

2. 根据下表中的参数说明编辑文件。

表 7-3: 服务器配置参数说明

参数名	类型	是否必填	描述
access_id	String	是	您的阿里云账号的 API 访问密钥 AccessKeyID。 说明： 迁云工具需要使用 AccessKeyID 以及 AccessKeySecret，AccessKey 是您的重要凭证，请妥善保管，防止泄露。
secret_key	String	是	您的阿里云账号的 API 访问密钥 AccessKeySecret。
region_id	String	是	您的服务器迁移入阿里云的地域 ID，如 cn-hangzhou（华东1），取值参阅 地域与可用区。
image_name	String	是	为您的服务器镜像设定一个镜像名称，该名称不能与同一地域下现有镜像名重复。长度为[2, 128]个英文或中文字符。必须以大小字母或中文开头，不

参数名	类型	是否必填	描述
			能以http://和https://开头。可以包含数字、半角冒号 (:) 、下划线 (_) 或者连字符 (-) 。
system_disk_size	int	是	为系统盘指定大小，单位为 GiB。取值范围：[40, 500] 说明： 该参数取值需要大于源服务器系统盘实际占用大小，例如，源系统盘大小为 500 GiB，实际占用 100 GiB，那该参数取值只要大于 100 GiB 即可。
platform	String	否	源服务器的操作系统。取值范围：Windows Server 2003 | Windows Server 2008 | Windows Server 2012 | Windows Server 2016 | CentOS | Ubuntu | SUSE | OpenSUSE | Debian | RedHat | Others Linux 说明： 参数 `platform` 的取值需要与以上列表保持一致，必须区分大小写，并保持空格一致。
architecture	String	否	系统架构。取值范围：i386 | x86_64
bandwidth_limit	int	否	数据传输的带宽上限限制，单位为 KB/s。 默认值：0，0 表示不限制带宽速度。
data_disks	Array	否	数据盘列表，最多支持 16 块数据盘。具体参数参阅下表数据盘配置参数说明。该参数可以置为缩容数据盘的预期数值，单位为 GiB，该值不能小于数据盘实际使用空间大小。

表 7-4: 数据盘配置参数说明

参数名	类型	是否必填	描述
data_disk_index	int	是	数据盘序号。取值范围：[1, 16] 初始值：1
data_disk_size	int	是	数据盘大小。单位为 GiB。取值范围：[20, 32768] 说明：

参数名	类型	是否必填	描述
			该参数取值需要大于源服务器数据盘实际占用大小。例如，源数据盘大小为 500 GiB，实际占用 100 GiB，那该参数取值需要大于 100 GiB。
src_path	String	是	数据盘源目录。取值举例： • Windows 指定盘符，例如，D、E 或者 F。 • Linux 指定目录，例如，/mnt/disk1、/mnt/disk2 或者 /mnt/disk3。 说明： 不能配置为根目录或者系统目录，例如，/bin、/boot、/dev、/etc、/lib、/lib64、/sbin、/usr 和 /var。

3. 检查 JSON 语言格式的规范性，关于 JSON 的语法标准请参阅 *RFC 7159*。

此处以四种场景为例，为您示范如何根据场景编辑 user_config.json 文件：

场景一：迁移一台无数据盘的 **Windows** 服务器

- 假设您的服务器配置信息为：
  - 操作系统：Windows Server 2008
  - 系统盘：30 GiB
  - 系统架构：64 位
- 您的迁云目标为：
  - 目标地域：阿里云华东 1 地域（cn-hangzhou）
  - 镜像名称：CLIENT_IMAGE_WIN08_01
  - 系统盘设置：50 GiB

```
{
    "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
```

```
}
```

场景二：迁移一台带数据盘的 Windows 服务器

如果您的 Windows 服务器在场景一的基础上加入了 3 块数据盘，源目录和数据盘大小分别为：

- D：100 GiB
- E：150 GiB
- F：200 GiB

```
{
    "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
}
```

场景三：迁移一台无数据盘的 Linux 服务器

- 假设您的服务器配置信息为：
  - 发行版本：CentOS 7.2
  - 系统盘：30 GiB
  - 系统架构：64 位
- 您的迁云目标为：
  - 目标地域：阿里云华东 1 地域（`cn-hangzhou`）
  - 镜像名称：CLIENT_IMAGE_CENTOS72_01
  - 系统盘设置：50 GiB

```
{
    "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
}
```

场景四：迁移一台有数据盘的 Linux 服务器

如果您的 Linux 服务器在场景三的基础上加入了 3 块数据盘，源目录和数据盘大小分别为：

- /mnt/disk1：100 GiB
- /mnt/disk2：150 GiB
- /mnt/disk3：200 GiB

```
{
    "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
}
```

步骤 3：过滤无需迁云的目录

迁云工具能过滤文件或者目录，过滤的文件不会被迁移到云端。具体通过配置 *rsync* 实现过滤，过滤配置放在Excludes目录下。

说明：

建议您排除无需迁云的数据盘或者目录，以减少迁云传输时间以及云端磁盘使用空间。

过滤Windows系统的文件

默认过滤的文件（夹）包括pagefile.sys、$RECYCLE.BIN和System Volume Information。

- 系统盘：配置Excludes目录下的rsync_excludes_win.txt。
- 数据盘：在Excludes目录下新建并配置
  - rsync_excludes_win_disk1.txt
  - rsync_excludes_win_disk2.txt
  - rsync_excludes_win_disk3.txt
  - ……

Windows系统示例

- 假设您需要过滤C盘文件夹 `C:\MyDirs\Docs\Words` 和文件 `C:\MyDirs\Docs\Excels\Report1.xlsx`，可在rsync_excludes_win.txt中添加过滤配置：

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

- 假设您需要过滤D盘文件夹 `D:\MyDirs\Docs\Words` 和文件 `D:\MyDirs\Docs\Excels\Report1.xlsx`，可在rsync_excludes_win_disk1.txt中添加过滤配置：

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

过滤Linux系统的文件

默认过滤的文件或目录包括/dev/*、/sys/*、/proc/*、/media/*、lost+found/*、/mnt/*和/var/lib/lxcfs/*。

说明：

/var/lib/lxcfs/*目录仅针对部分系统版本，例如，无权访问Ubuntu的Linux容器服务缓存目录时，需要排除Ubuntu的/var/lib/lxcfs/*才能顺利迁云。

- 系统盘：配置Excludes目录下的rsync_excludes_linux.txt。
- 数据盘：在Excludes目录下新建并配置
  - rsync_excludes_linux_disk1.txt
  - rsync_excludes_linux_disk2.txt
  - rsync_excludes_linux_disk3.txt
  - ……

Linux系统示例

- 假设您需要过滤系统盘（根目录/）文件夹`/var/mydirs/docs/words`和文件`/var/mydirs/docs/excels/report1.sh`，可在rsync_excludes_linux.txt中添加过滤配置：

```
/var/mydirs/docs/words/
/var/mydirs/docs/excels/report1.sh
```

- 假设您需要过滤数据盘目录/mnt/disk1中的文件夹`/mnt/disk1/mydirs/docs/words` 和文件`/mnt/disk1/mydirs/docs/excels/report1.sh`，可在rsync_excludes_linux_disk1.txt中添加过滤配置：

```
/mydirs/docs/words/
/mydirs/docs/excels/report1.sh
```

说明：

Linux数据盘需要去掉数据盘src_path前缀路径，例如去掉上述示例中的/mnt/disk1。

步骤 4：（可选）编辑 **client_data** 文件

警告：

如果您能直接从自建机房（Integrated Data Center，IDC）、虚拟机环境或者云主机访问某一阿里云地域下的专有网络VPC，您可以编辑 client_data 文件。反之，请勿自行修改配置文件client_data，否则会影响迁云工作，出现进程卡顿等现象。

client_data 文件记录了迁云过程中的数据文件，关于如何编辑和配置 client_data 文件，请参阅*VPC*内网迁云。

每成功迁云一次，配置文件 client_data 会自动记录迁云成功后在 ECS 控制台创建的 ECS 实例的相关数据。再次迁云时，您需要使用初始下载的客户端配置文件。

步骤 5：运行迁云工具

Windows 服务器：右击 go2aliyun_client.exe，选择 **以管理员身份运行**。GUI 版本程序操作指南请参阅 迁云工具 *Windows GUI* 版本介绍。

Linux 服务器：以 root 用户身份运行迁云工具。

1. 运行 `chmod +x ./go2aliyun_client`。
2. 运行 `./go2aliyun_client`。

迁云结果

当提示 `Goto Aliyun Finished!` 时，前往 *ECS*管理控制台 镜像详情页查看结果。您的源服务器中的操作系统、应用程序以及应用数据等以自定义镜像的形式出现在相应地域的 ECS 控制台上。

当提示 `Goto Aliyun Not Finished!` 时，检查同一目录下 Logs 文件夹下的日志文件 排查故障。修复问题后，重新运行迁云工具，迁云工具会从上一次执行的进度中继续迁云。

说明：

- 迁云中断后再次执行工具时或者工具会提示迁云已完成时都是从 client_data 文件获取信息。迁云工作完成后再次运行想重新迁云工具时，您需要使用初始的 client_data 文件或者清空现有的 client_data 文件数据。
- 初始化 client_data 文件后，任务进度信息丢失并且迁云工作会从头开始。在诸如中转实例被意外释放释放或者 VPC、VSwitch 和安全组信息错误等原因导致的迁云中断事件中，您可以在排查故障后使用初始化的 client_data 文件。

下一步

您可以 使用该自定义镜像创建按量付费 *ECS* 实例 或者 使用自定义镜像更换系统盘，测试自定义镜像能否正常运行。

迁移带数据盘的 Linux 服务器后，启动实例时默认不挂载数据盘。您可以在启动 ECS 实例后运行 `ls /dev/vd*` 命令查看数据盘设备，根据实际需要手动挂载，并编辑 `/etc/fstab` 配置开机自动挂载。更多详情，参阅 *Linux* 格式化和挂载数据盘。

7.3 VPC内网迁云

如果您能直接从自建机房（Integrated Data Center，IDC）、虚拟机环境或者云主机访问某一阿里云地域下的专有网络VPC，建议您使用源服务器与VPC内网互连的迁云方案。VPC内网迁云能获得比通过公网更快速更稳定的数据传输效果，提高迁云工作效率。

前提条件

VPC内网迁云要求您能从IDC、虚拟机环境或者云主机访问目标VPC。具体实现方案可以选择高速通道服务或者VPN网关服务，利用高速通道的 专线接入 功能或者在目标VPC中 搭建*VPN*网关 。

说明：

高速通道或者VPN网关为付费云服务，更多详情，请根据您的实际需要使用。请参阅 物理专线连接计费说明 和 预付费。

client_data说明

VPC内网迁云需要您自行编辑client_data文件。client_data记录了迁云过程中的数据文件，包含了以下信息：

- 迁云中转实例的ID、名称、公网带宽和IP地址等属性。
- 迁移数据盘的进程信息。
- 生成的自定义镜像名称。
- 中转实例部署的地域和网络类型。
- 中转实例使用的VPC、虚拟交换机和安全组。

更多详情，请参阅下载后迁云工具的client_data文件。

警告：

为避免迁云失败，若您没有VPC内网迁云需求，请勿自行修改配置文件client_data。否则会影响迁云工作，出现进程卡顿等现象。

下载迁云工具 并打开client_data文件后，您需要修改如下参数：

名称	类型	是否必填	描述
net_mode	Integer	否	选择数据传输方式。取值范围： • 0（默认）：数据从公网传输，此时要求源服务器能访问公网，数据从公网传输。 • 1：数据从VPC内网传输，此时要求源服务器能访问指定VPC。 • 2：数据从VPC内网传输，此时要求源服务器同时能访问公网和指定VPC。 VPC内网迁云需要将net_mode设置为1或者2。
vpc	Array	否	已经配置了高速通道服务或者VPN网关的VPC ID。当net_mode=1或net_mode=2时为必填参数。由必填的vpc_id和选填的vpc_name和description三个字符串（String）参数构成一个JSON数组，分别表示VPC ID、VPC名称和VPC描述。

名称	类型	是否必填	描述
vswitch	Array	否	指定VPC下的一台虚拟交换机ID。当net_mode=1或net_mode=2时为必填参数。由必填的vswitch_id和选填的vpc_name和description三个String参数构成一个JSON数组，分别表示虚拟交换机ID、虚拟交换机名称和虚拟交换机描述。
securegroupid	String	否	指定VPC下的安全组ID。

源服务器能访问指定VPC

以下步骤适用于net_mode=1的情形。迁云工程会分成3个阶段，其中阶段1（Stage 1）和阶段3（Stage 3）在备用服务器中完成，需要备用服务器能访问公网；阶段2（Stage 2）数据传输在待迁移的源服务器中进行。

1. 登录一台您能够访问公网的服务器A。
2. 编辑迁云工具的client_data文件：设置 net_mode=1，填入已经配置了高速通道服务或者VPN网关的 vpc_id、vswitch_id 和 zone_id 参数。
3. （可选）在client_data文件中配置 security_group_id 参数，但安全组入方向必须放行代理端口8080和8703。更多详情，请参阅 添加安全组规则。
4. 按照 公网迁云 步骤在服务器A内运行迁云工具，直到提示Stage 1 Is Done!。

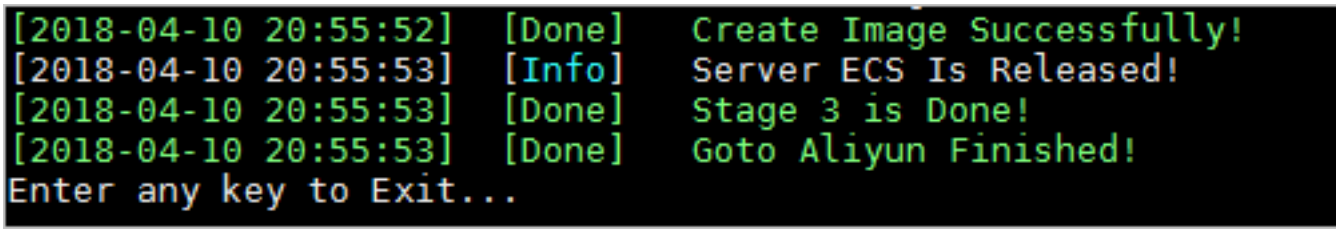
```
[2018-04-10 20:43:16]  [Info]   Server ECS Is Running!
[2018-04-10 20:43:16]  [Done]   Stage 1 is Done!
[2018-04-10 20:43:16]  [Info]   Goto Aliyun Not Finished, Read
Enter any key to Exit...
```

5. 登录您需要迁移的源服务器，复制服务器A的迁云工具配置，包括user_config.json、rsync和client_data文件，保持配置文件内容一致。
6. 按照 公网迁云 步骤在待迁移的源服务器内运行迁云工具，直到提示Stage 2 Is Done!。

```
[2018-04-10 20:47:43]  [Info]   Do Grub...
[2018-04-10 20:48:20]  [Done]   Stage 2 is Done!
[2018-04-10 20:48:20]  [Info]   Goto Aliyun Not Finished, Read
Enter any key to Exit...
```

7. 登录服务器A，复制待迁移的源服务器的迁云工具配置，包括user_config.json、rsync和client_data文件，必须保持配置文件内容一致。

8. 按照 公网迁云 步骤在服务器A内再次运行迁云工具，直到提示 `Stage 3 Is Done!`，表示VPC内网迁云顺利完成。



源服务器能访问公网和指定VPC

以下步骤适用于 `net_mode=2` 的情形，操作过程与 `net_mode=0` 时，即公网迁云相同。`net_mode=2` 时，数据自动从VPC迁移上云，其他过程走公网，传输速度稍微慢于VPC内网迁云方式一（`net_mode=1`）。

1. 登录您能够访问公网的源服务器，按照 公网迁云 步骤运行迁云工具。
2. 编辑迁云工具的client_data文件。设置 `net_mode=2`，填入已经配置了高速通道服务或者VPN网关的 `vpc_id`、`vswitch_id` 和 `zone_id` 参数。
3. （可选）在client_data文件中配置 `security_group_id` 参数，但安全组入方向必须放行代理端口8080和8703。更多详情，请参阅 添加安全组规则。
4. 按照 公网迁云 步骤运行迁云工具。

FAQ

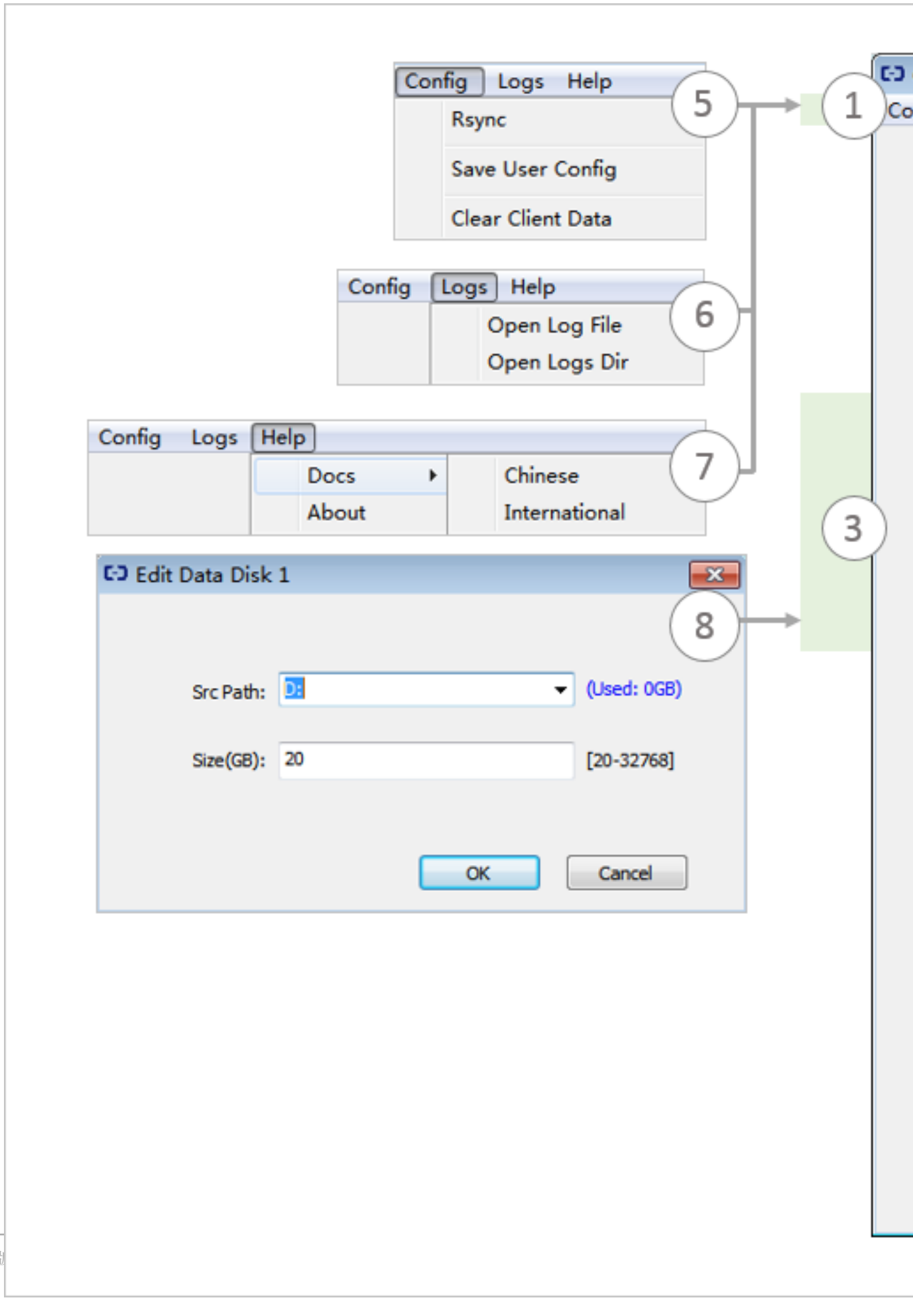
当迁云工作中断后，您可以查看 迁云工具FAQ 或者 添加迁云工具客户反馈钉钉群 联系ECS迁云技术支持。

7.4 迁云工具 Windows GUI 版本介绍

迁云工具从 1.2.9 版本开始支持 Windows GUI 版本，程序文件名为go2aliyun_gui.exe。如果您使用的是旧版本迁云工具，请重新 下载 压缩包以获取更多功能。迁云工具 Windows GUI 界面的设置与命令行界面配置原理一致，并且 Windows GUI 版本与命令行界面运行过程兼容，您可以在使用迁云工具的过程中切换使用方式。

界面介绍

迁云工具 Windows GUI 界面有四块区域组成，包括菜单栏、用户自定义配置（user_config.json）编辑区、磁盘列表和任务进度与日志区。如下图所示：


Config
Logs
Help
Rsync
Save User Config
Clear Client Data
5
1
Config
Logs
Help
Open Log File
Open Logs Dir
6
Config
Logs
Help
Docs
About
Chinese
International
7
3
Edit Data Disk 1
8
Src Path:
D:
(Used: 0GB)
Size(GB):
20
[20-32768]
OK
Cancel

图标说明

1. 菜单栏，由 **Config**、**Logs** 和 **Help** 三个功能页组成。
2. 用户自定义配置（user_config.json）编辑区，主要用于配置源服务器的一些必要配置信息，其中包括您的 AccessKey 信息、源服务器的操作系统信息、系统盘大小、源服务器迁移入阿里云的地域 ID、生成 ECS 镜像后的名称以及生成的目标自定义镜像的配置信息等。更多详情，请参阅 使用迁云工具 编辑 *user_config.json*。
3. 磁盘列表，包括系统盘和数据盘。您可以在该区域通过右键单击添加需要迁云的磁盘，双击进入磁盘信息编辑页面。
4. 任务进度与日志区，运行迁云工具后，您可以通过该区域查看任务进度与或者根据界面提示排查故障。
5. 菜单栏之一，您可以在这里单击 **Rsync** 设置数据传输的带宽上限值，单位为 KB/s，单击 **Save User Config** 保存当前的页面设置便于批量操作，单击 **Clear Client Data** 一键初始化客户端配置文件，更多详情，请参阅 使用迁云工具。
6. 菜单栏之一，您可以在这里单击 **Open Log File** 快速打开日志文件，或者单击 **Open Log Dir** 查找日志文件所在路径。
7. 菜单栏之一，您可以在这里获取在线文档或者迁云工具版本信息。
8. 您可以在这里添加数据盘。迁云工具会自动查询您的服务器里的数据盘盘符列表，显示已使用数据盘空间。数据盘大小设置需要大于源服务器数据盘实际占用大小，例如，源数据盘大小为 500 GiB，实际占用 100 GiB，那么您只要设置成大于 100 GiB 即可。

在 GUI 界面上完成服务器信息配置后，您可以单击 **Start** 开始迁云工作。当任务进度与日志区出现 `Goto Aliyun Finished!` 提示时，前往 *ECS*管理控制台 镜像详情页查看结果。当出现 `Goto Aliyun Not Finished!` 提示时，通过菜单功能页 **Logs** 检查同日志文件 排查故障。修复问题后，重新运行迁云工具即可恢复迁云工作，迁云工具会从上一次执行的进度中继续迁云，无需重新开始。

7.5 CLI参数

迁云工具从1.2.8版本开始支持命令行（Command line interface，CLI）参数，在迁云工具所在路径中运行`--help`可以查看参数列表。CLI参数具有无需打开各种JSON文件就能配置迁云工具、调整自定义使用习惯和一键清除client_data等优点。如果您使用的是旧版本迁云工具，请重新 下载压缩包 以获取更多功能。CLI参数需要您对迁云工具的使用方式有所了解，更多详情，请参阅 使用迁云工具。

Windows版本参数列表

以下为Windows版迁云工具完整的CLI参数列表。

```
usage: go2aliyun_client.exe [options]
options:
  --help                        show usage.
  --version                     show version.
  --nocheckversion              no check for new version.
  --noenterkey                  no enter key to exit.
  --progressfile                set progress file path.
  --cleardata                   clear client data and server ecs.
  --accesssid=<accesss_id>      set access id.
  --secretkey=<secret_key>      set secret key.
  --regionid=<region_id>        set region id.
  --imagename=<image_name>      set image name.
  --systemdisksize=<sdsize>     set system disk size.
  --platform=<platform>         set platform.
  --architecture=<arch>         set architecture.
  --datadisks=<data_disks>      set data disks.
    data_disks=data_disk_index|data_disk_size|src_path;
    e.g. --data_disks=1|100|D:;2|150|E:
  --bandwidthlimit=<limit>      set bandwidth limit.
  --netmode=<net_mode>          set net mode.
  --vpcid=<vpc_id>              set vpc id.
  --vswitchid=<vswitch_id>      set vswitch id.
  --zoneid=<zone_id>            set zone id.
  --securegroupid=<sgid>        set secure group id.
```

Linux版本参数列表

以下为Linux版迁云工具完整的CLI参数列表。

```
usage: ./go2aliyun_client [options]
options:
  --help                        show usage.
  --version                     show version.
  --nocheckversion              no check for new version.
  --noenterkey                  no enter key to exit.
  --progressfile                set progress file path.
  --cleardata                   clear client data and server ecs.
  --accesssid=<accesss_id>      set access id.
  --secretkey=<secret_key>      set secret key.
  --regionid=<region_id>        set region id.
  --imagename=<image_name>      set image name.
  --systemdisksize=<sdsize>     set system disk size.
  --platform=<platform>         set platform.
  --architecture=<arch>         set architecture.
  --datadisks=<data_disks>      set data disks.
    data_disks=data_disk_index|data_disk_size|src_path;
    e.g. --data_disks=1|100|/mnt/disk1;2|150|/mnt/disk2
  --bandwidthlimit=<limit>      set bandwidth limit.
  --netmode=<net_mode>          set net mode.
  --vpcid=<vpc_id>              set vpc id.
  --vswitchid=<vswitch_id>      set vswitch id.
  --zoneid=<zone_id>            set zone id.
```

```
  --securegroupid=<sgid>       set secure  group id.
```

常规参数

以下为Windows和Linux迁云工具的通用参数。常规参数不会影响迁云工具配置，对迁云工作无影响，常用于调整迁云工具的使用习惯和交互界面。

参数	说明
nocheckversion	停止提示版本更新。
noenterkey	迁云结束前不提示输入按键，而是直接退出，减少交互。
progressfile	设置迁移进度输出文件，文件内容有两行，格式如下： • 第一行是进度标识，4个标识主要分为准备数据传输阶段PrepareForRsync、数据传输阶段DoRsync、创建镜像阶段CreateImage、完成迁云Finished。 • 第二行是进度值表示每一阶段的进度。取值为Integer，范围为[0, 100]。
cleardata	清理client_data数据文件，并释放 运行中（Running）的中转实例。

警告：

当您的迁云工作还未完成前，请慎重使用cleardata参数。否则会导致迁云中断，已迁移的进度会被作废。

user_config参数

以下为配置user_config的相关CLI参数，更多有关user_config的信息，请参阅 使用迁云工具。

说明：

使用CLI参数指定了user_config的配置后，迁云工具会以CLI参数为准，而忽略配置文件。

```
--accesssid=<accesss_id>     # 设置user_config中的AccessKey ID
--secretkey=<secret_key>     # 设置user_config中的AccessKey Secret
--regionid=<region_id>       # 设置user_config中的地域配置
--imagename=<image_name>     # 设置user_config中的自定义镜像名称
--systemdisksize=<sdsize>    # 设置user_config中的系统盘容量
--platform=<platform>        # 设置user_config中的镜像发行平台
--architecture=<arch>        # 设置user_config中的镜像系统架构
--datadisks=<data_disks>     # 设置user_config中的数据盘列表，由|和;分隔不
同数据盘取值，例如data_disk_index|data_disk_size|src_path;
```

```
--bandwidthlimit=<limit>      # 设置user_config中的公网出带宽上限
```

client_data参数

以下为指定VPC内网迁移相关参数。更多详情，请参阅 *VPC内网迁云*。

```
--netmode=<net_mode>          # 设置client_data中的迁云方式，取值可以是0、1和
2
--vpcid=<vpc_id>              # 设置client_data中配置了高速通道或者VPN网关的
VPC
--vswitchid=<vswitch_id>      # 设置client_data中VPC下的虚拟交换机
--securegroupid=<sgid>        # 设置client_data中VPC下的安全组
```

7.6 迁云工具 FAQ

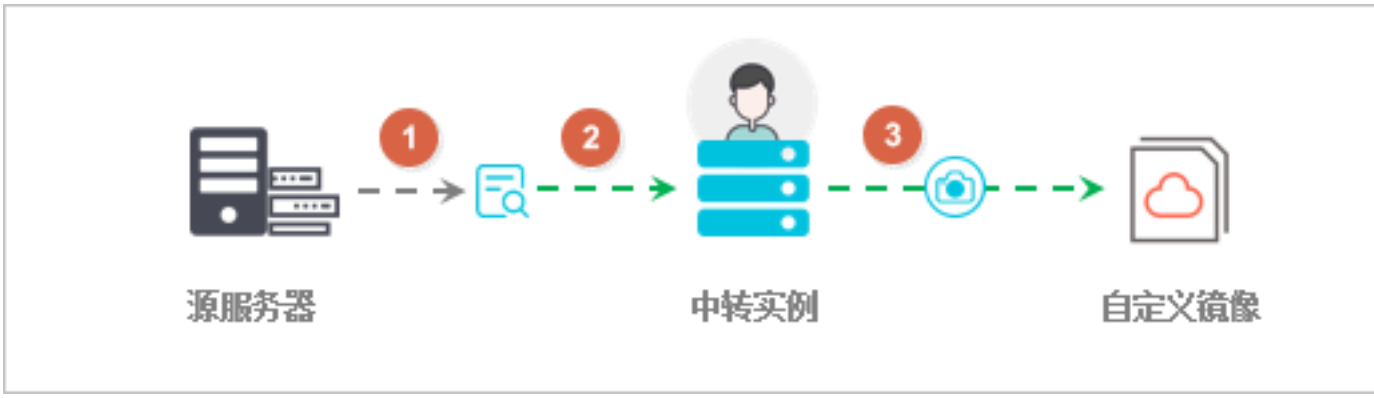
- 我在什么场景下可以使用迁云工具#
- 迁云工具的迁移过程是什么#
- 迁云工具是否支持断点续传#
- 迁云工具是否支持迁移增量数据#
- 迁云完成后的结果是什么#
- 迁移完成得到自定义镜像后该如何操作#
- 如何处理迁云中断或提示失败#
- 关于中转实例#我需要注意什么#
- 关于*user_config.json*#我需要注意什么#
- 什么时候需要过滤目录或文件#
- 关于*client_data*文件#我需要注意什么#
- 什么时候需要清理*client_data*文件#
- 迁云完成后再次迁云该如何操作#
- 误释放了中转实例怎么办#
- 为什么提示账号余额不足*NotEnoughBalance*#
- 为什么提示*RAM*权限不足*Forbidden.RAM*#
- 为什么提示子账号权限不足*Forbidden.SubUser*#
- 我的服务器在出方向需要访问哪些公网地址和端口#
- 迁移*Windows*服务器后怎么检查系统#
- 阿里云支持激活哪些*Windows*服务器许可证#
- 迁移*Linux*服务器前怎么检查是否满足迁云条件#

- 迁移Linux服务器后怎么检查系统#

1. 我在什么场景下可以使用迁云工具？

迁云工具可以将物理服务器、虚拟机以及其他云平台云主机一站式地迁移到阿里云ECS，支持迁移主流Windows和Linux操作系统。更多详情，请参阅 什么是迁云工具与P2V。

2. 迁云工具的迁移过程是什么？



- 检查源服务器是否满足迁移条件。
- 在您的云账号下创建一台临时中转实例，将源服务器系统数据传输到中转实例。
- 从中转实例打快照制作自定义镜像。

3. 迁云工具是否支持断点续传？

支持。数据传输中断后，重新运行迁云工具即可继续迁云。

4. 迁云工具是否支持迁移增量数据？

不支持。建议在迁云前先暂停如数据库或容器服务之类的应用，或者先 过滤 相关数据目录，迁云完成后再同步数据。

5. 迁云完成后的结果是什么？

生成一份源服务器操作系统的自定义镜像，您可以登录 ECS管理控制台，在相应地域的镜像列表中查看。

6. 迁移完成得到自定义镜像后该如何操作？

建议先使用该镜像创建一台按量付费的实例，检查系统是否正常。确认镜像可用后，选择合适您业务的 实例规格 并 创建一台或多台ECS实例。

7. 如何处理迁云中断或提示失败？

- 当迁云工具程序异常退出或者迁云进度卡顿时，可以尝试重新运行迁云工具恢复迁云。

- 如果迁云失败并提示`Not Finished`，您可以查看Logs目录下的日志文件，并参阅 排查故障 或者 *API*错误中心 查看报错原因。

  如果问题仍未解决，建议您添加 迁云工具支持钉钉群。也可以 提交工单 并附上日志信息，联系售后客服支持。

8. 关于中转实例，我需要注意什么？

- 迁云工具自动创建、启动、停止和释放中转实例INSTANCE_FOR_GOTOALIYUN。为保证顺利完成迁云，请勿人为干预中转实例的运行状态。
- 中转实例的默认安全组在入方向开放了8080和8703端口，这是中转实例的迁云服务端口，请勿修改或删除该安全组配置。
- 迁云完成后，中转实例会被自动释放，如果迁云失败，需要手动 释放实例。

9. 关于**user_config.json**，我需要注意什么？

如果已经开始迁云，并且中转实例已经创建，请勿修改user_config.json里的系统盘大小或数据盘大小数量配置。如果仍然需要修改，必须清理client_data文件后重新迁云。

10. 什么时候需要过滤目录或文件？

源服务器中有不需要上传的数据目录或文件，可以通过配置Excludes文件过滤，提高迁云效率。

特别地，您可以过滤无法暂停的数据库、Docker容器或者处于活动状态的数据目录或文件，以提高数据传输的稳定性。

11. 关于**client_data**文件，我需要注意什么？

client_data文件记录了迁云过程数据，包含中转实例信息、迁云进度等。一般情况下请不要手动修改或删除client_data文件，否则可能会导致迁云失败。

12. 什么时候需要清理**client_data**文件？

清理client_data文件可以使用 *CLI*命令 `--cleardata`，或者通过 *Windows GUI* 的Client Client Data菜单项。

- 迁云已经开始后如果想重新迁云，可以清理现有的client_data文件或者使用原始的client_data文件覆盖后再运行。
- 同时在某些迁云失败的情况下，如误释放中转实例、VPC、虚拟交换机或者安全组不存在等，可以尝试清理client_data操作来解决。

13. 迁云完成后再次迁云该如何操作？

清理client_data数据文件，然后运行迁云工具重新迁云。

14. 误释放了中转实例怎么办？

清理client_data数据文件，然后运行迁云工具重新迁云。

15. 为什么提示账号余额不足**NotEnoughBalance**？

迁云工具本身是免费的，但迁云时默认创建 按量付费 中转实例。根据阿里云收费服务标准，创建按量付费实例需要您的云账号余额不低于100元人民币。

16. 为什么提示**RAM**权限不足**Forbidden.RAM**？

您的RAM账号创建的AccessKey没有管理ECS和VPC资源的权限。建议您联系主账号授权 *AliyunECSFullAccess* 和 *AliyunVPCFullAccess* 角色策略。

17. 为什么提示子账号权限不足**Forbidden.SubUser**？

迁云工具需要使用账号AccessKeyID和AccesKeySecret创建中转实例，该操作属于下单操作。RAM账户没有下单权限时会出现报错Forbidden.SubUser。建议您迁云时使用主账号AccessKey。

18. 我的服务器在出方向需要访问哪些公网地址和端口？

迁云工具需要访问下列阿里云服务：

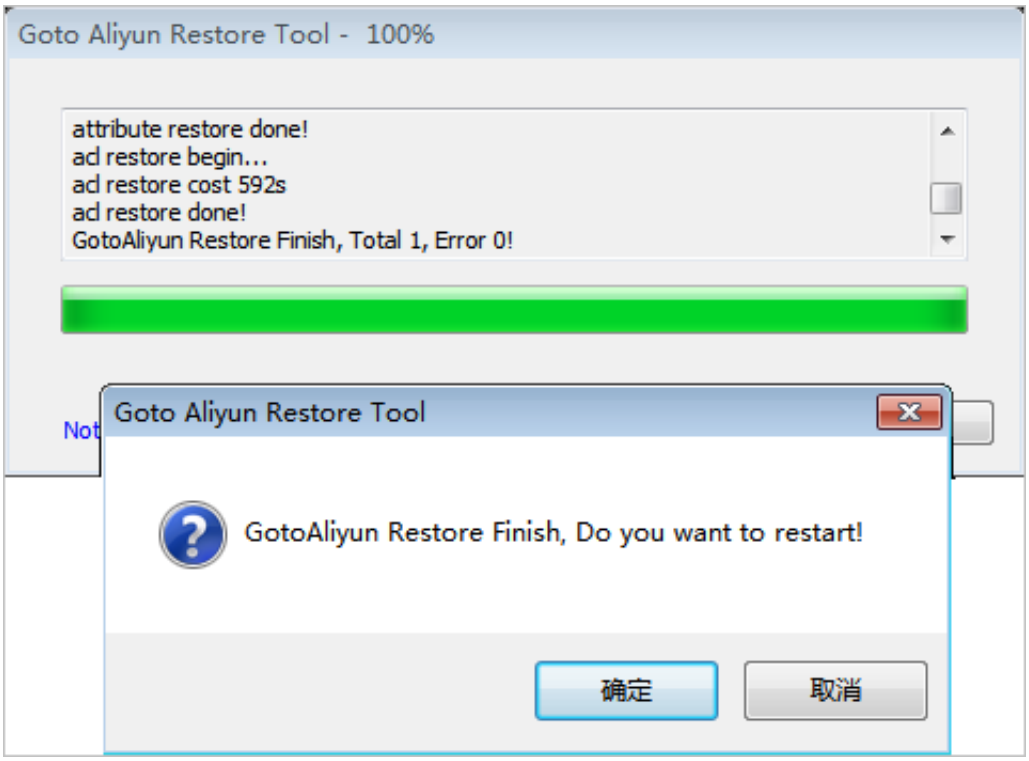
- 通过 HTTP 80 端口访问 ECS 主接入地址 `http://ecs.aliyuncs.com`。更多详情，请参阅 接入地址。
- 通过 HTTP 80 端口访问 VPC `http://vpc.aliyuncs.com`。
- 通过 HTTPS 443 端口访问 STS `https://sts.aliyuncs.com`。
- 通过 8080 和 8703 代理端口访问中转实例的公网 IP 地址。

> **说明：**
> 源服务器不需要开放任何入方向的端口，但是需要在出方向访问上述公网地址和端口。

19. 迁移**Windows**服务器后怎么检查系统？

迁移Windows系统后初次启动实例时：

1. 检查系统盘数据是否完整。
2. 如果有数据盘缺失，进入磁盘管理检查盘符是否丢失。
3. 等待文件系统权限修复过程完成后，选择是否重启实例：



说明：

初次启动ECS实例后，如果文件系统权限修复程序未自启动，您可以运行C:\go2aliyun_prepare\go2aliyun_restore.exe手动修复。执行前要确保实例上的磁盘数量和盘符路径跟源系统保持一致。

4. 检查网络服务是否正常。
5. 检查其他系统应用服务是否正常。

20. 阿里云支持激活哪些Windows Server？

支持自动激活Windows Server 2003、2008、2012和2016。其他不在此列版本的Windows如果迁移至ECS，需要 申请许可移动性证。

21. 迁移Linux服务器前怎么检查是否满足迁云条件？

可以使用迁云工具里面自带的client_check工具检测，运行`./client_check --check`命令即可，如果所有检测项提示OK则表示满足迁云条件。

22. 迁移Linux服务器后怎么检查系统？

迁移Linux系统后初次启动实例时：

1. 检查系统盘数据是否完整。
2. 如果有数据盘，您需要自行 挂载数据盘。

3. 检查网络服务是否正常。
4. 然后检查其他系统服务是否正常。

7.7 排查报错

迁云工具支持断点恢复，文件传输过程支持断点续传。一般情况下如果主程序异常中断或提示迁移不成功，故障排查处理完问题后，您可以再次运行主程序恢复迁云工作。

成功迁移Windows Server 2008及以上版本的Windows服务器，启动实例后您需要先使用 *Reset File Permission* 工具修复默认文件系统权限，以保证实例服务及组件正常。 成功迁移Windows服务器并在初次启动实例后，请等待文件系统权限自动修复进程完成，更多详情，请参阅 *FAQ 19* 迁移*Windows*服务器后怎么检查系统。

- 日志错误提示 *IllegalTimestamp*
- 日志错误提示 *UnKnownError*
- 日志错误提示 *OperationDenied*
- 日志错误提示 *InvalidAccountStatus.NotEnoughBalance*
- 日志错误提示 *Forbidden.RAM*
- 日志错误提示 *InvalidImageName.Duplicated*
- 日志错误提示 *InvalidAccountStatus.SnapshotServiceUnavailable*
- 日志错误提示 *Connect to Server Failed*
- 日志错误提示 *Do Rsync Disk x Failed*
- *Windows* 服务器卡在 *Prepare For Rsync Disk 0* 阶段
- 迁移 *Windows* 服务器后#启动实例被提示需要激活 *Windows*#
- 迁移 *Windows* 服务器后#启动实例发现数据盘缺失或者盘符错乱#
- 迁移 *Windows* 服务器后#启动实例发现文件权限异常或部分系统菜单目录显示语言不统一#
- *Linux* 服务器日志错误提示 *check rsync failed*
- *Linux* 服务器日志错误提示 *check virtio failed*
- *Linux* 服务器日志错误提示 *check selinux failed*
- *Linux* 服务器日志错误提示 *Do Grub Failed*
- 迁移 *Linux* 服务器后#启动实例发现原数据盘目录下没有数据#
- 迁移 *Linux* 服务器后#根据该自定义镜像创建的实例为何不能启动#
- 启动 *Others Linux* 实例后#网络服务不正常#

日志错误提示 IllegalTimestamp

请检查系统时间是否为正确时间。

日志错误提示 UnKnownError

请检查配置文件 user_config.json 中参数 `platform` 取值是否正确。

日志错误提示 OperationDenied

日志文件提示如 `rsync: send_files failed to open "…": Permission denied (13)` 的错误信息时，表明迁云工具无权访问该目录或文件夹，导致 rsync 失败。此时您可以通过配置 rsync_excludes_linux.txt 或者 Rsync/etc/rsync_excludes_win.txt 过滤该目录或文件夹，然后重试。

日志错误提示 InvalidAccountStatus.NotEnoughBalance

中转实例的默认付费模式为 按量付费，您的付费方式余额不足时，无法顺利迁云。您需要更新账户状态后重试。

日志错误提示 Forbidden.RAM

您使用的 RAM 账号权限不足，无法使用相关 API。

您需要被授权 ECS 和 VPC 访问权限 `AliyunECSFullAccess` 和 `AliyunVPCFullAccess`。更多详情，请参阅 *RAM* 文档 授权策略管理。

日志错误提示 InvalidImageName.Duplicated

指定的参数 `image_name` 不能与您已有的镜像名称重复。

日志错误提示 InvalidAccountStatus.SnapshotServiceUnavailable

该错误表示您的账号没有开通快照服务，您可以在*ECS*管理控制台开通快照服务。

日志错误提示 Connect to Server Failed

该错误表示无法连接中转实例。您可以按以下步骤检查：

1. 查看日志文件详细信息。
2. 依次检查：
   - 中转实例状态是否正常。
   - 本地网络服务是否正常。迁云工具需要访问 80、443、8703 和 8080 通信端口，请确保您的服务器已经放行这些端口。
3. 问题解决后，再次运行主程序重试。

日志错误提示 Do Rsync Disk x Failed

该错误表示文件传输中断。您可以按以下步骤检查：

1. 查看错误日志文件详细信息。如果错误日志文件中多次出现 **return: 3072** 或 **return: 7680** 信息提示，请确认源服务器数据库服务或者容器服务是否未开启状态，例如，Oracle、MySQL、MS SQL Server、MongoDB 和 Docker 等服务。您需要先暂停服务或者排除相关数据文件目录后再迁云。
2. 依次检查：
   - 中转实例状态是否正常。
   - 本地网络服务是否正常。迁云工具需要访问 80、443、8703 和 8080 通信端口，请确保您的服务器已经放行这些端口。
3. 问题解决后，再次运行主程序重试。

Windows 服务器卡在 Prepare For Rsync Disk 0 阶段

Windows 服务器迁云停在 Prepare For Rsync Disk 0 阶段，查看日志文件后发现显示 VssSnapshotul::VssSnapshotul GetSnapshotul Failed: 0x80042308。此时您可以：

1. 开启 Volume Shadow Copy 服务：
   a. 在服务器中单击 开始，在搜索框中输入 服务，回车确认。
   b. 找到 Volume Shadow Copy 服务，单击 启动此服务。
2. 卸载 QEMU Guest Agent 软件：
   a. 在服务器中单击 开始，在搜索框中输入 服务，回车确认。
   b. 查看是否有 QEMU Guest Agent VSS Provider 服务，若无该项服务，您可以直接重新运行迁云工具。
   c. 找到卸载脚本，大概位置位于 C:\Program Files (x86)\virtio\monitor\uninstall.bat 目录，执行脚本卸载 QEMU Guest Agent 软件。
3. 重新运行迁云工具。

迁移 Windows 服务器后，启动实例被提示需要激活 Windows？

您可以重装 Windows KMS Client Key 后通过 KMS 激活 Windows 服务。

1. 远程登录 Windows 实例。

2. 在 微软KMS Client Keys页面 查询到 Windows 服务器对应的 KMS Client Key，此处假设为 xxxx-xxxx-xxxx-xxxx-xxxx。
3. 使用管理员权限打开命令行工具，运行以下命令：

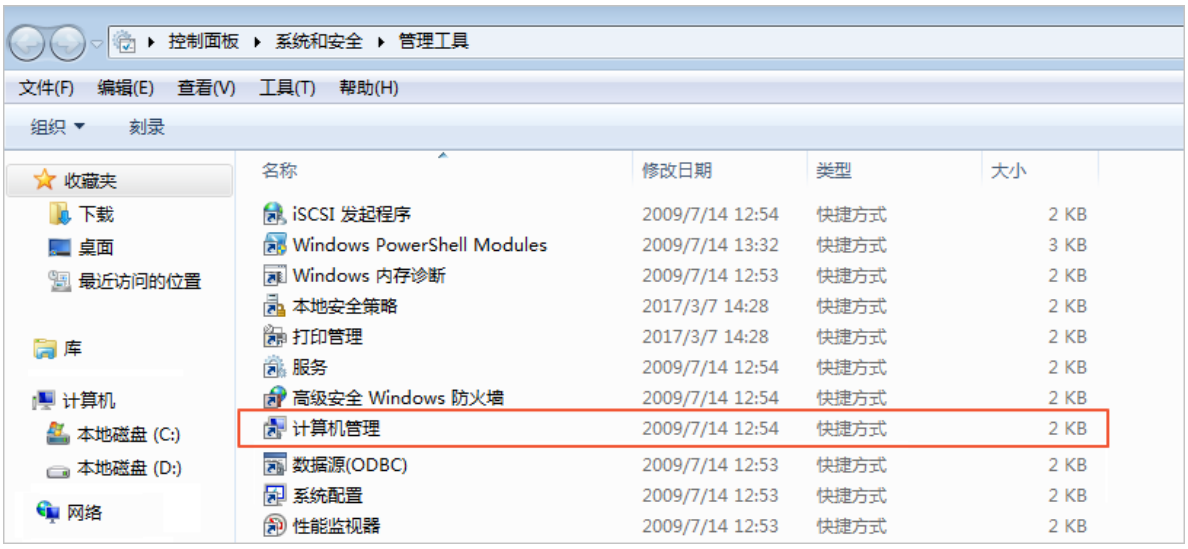
```
slmgr /upk
slmgr /ipk xxxx-xxxx-xxxx-xxxx-xxxx
```

4. 使用 KMS 激活 Windows。更多详情，请参阅 VPC环境下ECS Windows 系统激活方法。

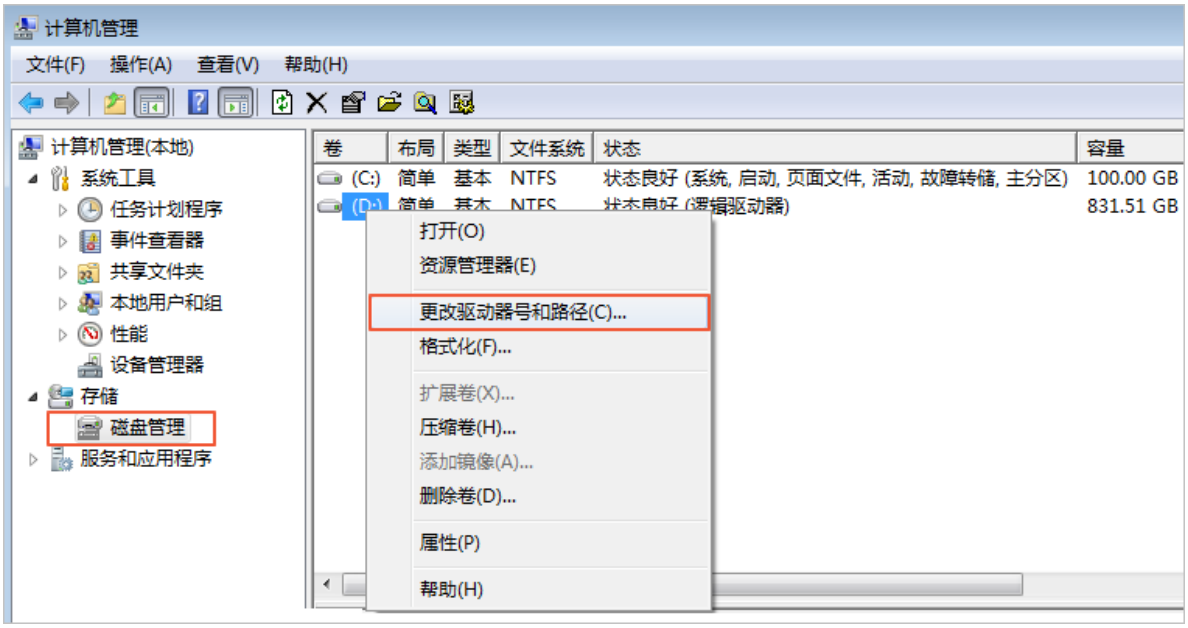
迁移 Windows 服务器后，启动实例发现数据盘缺失或者盘符错乱？

如果数据盘盘符缺失，您可以打开磁盘管理器，重新添加即可。

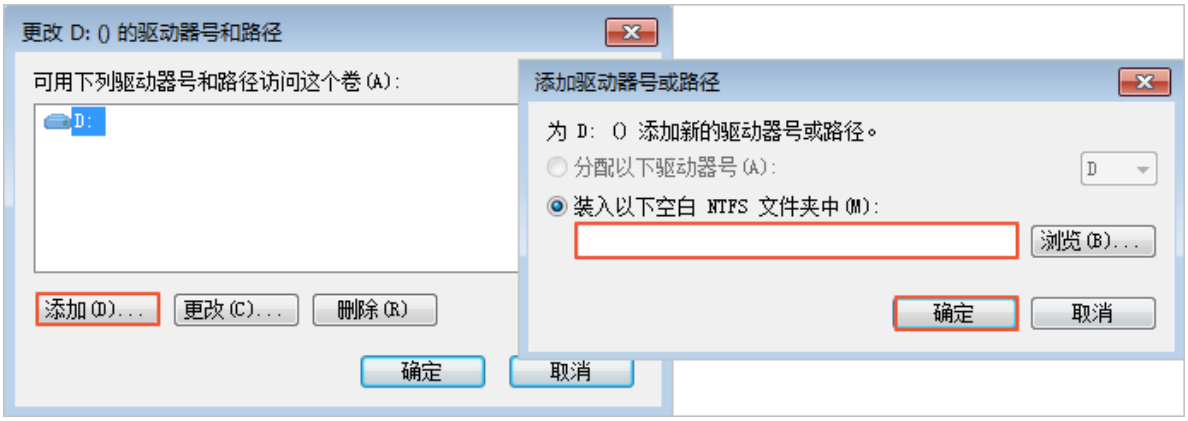
1. 打开控制面板 > 系统与安全 > 管理工具 > 计算机管理。



2. 找到并右击盘符缺失的数据盘，单击 更改驱动器和路径。

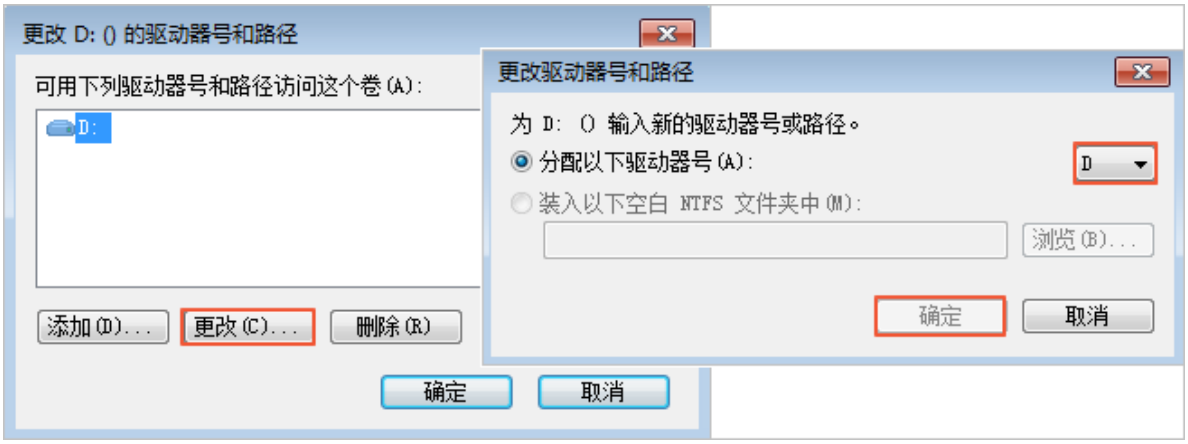


3. 单击 添加 并添加数据盘盘符。



如果数据盘盘符错乱，您可以打开磁盘管理器，重新更改即可。

1. 打开控制面板 > 系统与安全 > 管理工具 > 计算机管理。
2. 找到并右击盘符缺失的数据盘，单击 更改驱动器和路径。
3. 单击 更改 并更改数据盘盘符。



迁移 Windows 服务器后，启动实例发现文件权限异常或部分系统菜单目录显示语言不统一？

您需要等待文件系统权限修复操作成功完成。更多详情，请参阅 *FAQ* 迁移*Windows*服务器后怎么检查系统。

Linux 服务器日志错误提示 check rsync failed

请检查系统是否已安装 rsync 组件。

Linux 服务器日志错误提示 check virtio failed

请检查系统是否安装 *virtio* 驱动。

Linux 服务器日志错误提示 check selinux failed

请检查是否已禁用 SElinux。

您可以运行 `setenforce 0` 临时关闭 SELinux。

Linux 服务器日志错误提示 Do Grub Failed

日志文件提示如 `Do Grub Failed` 的错误信息时，确保源服务器已经安装了系统引导程序 GRUB（GRand Unified Bootloader）。您可以 安装 *1.9* 以上版本的系统引导程序 *GRUB* 后重试。

迁移 Linux 服务器后，启动实例发现原数据盘目录下没有数据？

迁移带数据盘的 Linux 服务器后，启动实例时默认不挂载数据盘。您可以在启动 ECS 实例后运行 `ls /dev/vd*` 命令查看数据盘设备，根据实际需要手动挂载，并编辑 `/etc/fstab` 配置开机自动挂载。

迁移 Linux 服务器后，根据该自定义镜像创建的实例为何不能启动？

- 检查驱动。创建 I/O 优化的实例时，请确保源服务器已经安装 *virtio* 驱动。
- 检查源系统引导配置是否正确。
- 如果您的源服务器系统是内核版本较低的 CentOS 5 或者 Debian 7，而且自带的 GRUB 程序版本低于 1.99，同时在 ECS 控制台 远程连接 登录实例发现开机界面如下图所示。

  您可以 安装 *1.9* 以上版本的系统引导程序 *GRUB* 后重试。

启动 Others Linux 实例后，网络服务不正常？

导入 Others Linux 类型镜像时，阿里云不会对该自定义镜像所创建的实例做任何配置工作，包括相关的网络配置、SSH 配置等。此时，您需要自行修改系统相关网络配置。

自 2018 年 03 月 31 号开始，迁云工具生成的镜像网络配置有变化，默认以 DHCP（Dynamic Host Configuration Protocol）的方式获取 IP 地址。如果网络配置失败，您可以 提交工单 联系阿里云。

7.8 反馈与支持

本文描述了您能在阿里云获取的数据上云反馈与支持渠道。

以下为迁云工具的反馈渠道。更多详情，请参阅 《通用参考》 支持渠道。

- 提交工单。
- 发送邮件至 *server-migration@alibabacloud.com*。
- 添加迁云工具客户反馈钉钉群，交流迁云经验与获取专家支持。钉钉是中国领先的智能移动办公平台，您可以前往 钉钉官网 下载合适的客户端。

8 借助于实例 RAM 角色访问其他云产品

以往部署在 ECS 实例中的应用程序如果需要访问阿里云其他云产品，您通常需要借助AccessKeyID 和 AccessKeySecret（下文简称 AK）来实现。AK 是您访问阿里云 API 的密钥，具有相应账号的完整权限。为了方便应用程序对 AK 的管理，您通常需要将 AK 保存在应用程序的配置文件中或以其他方式保存在 ECS 实例中，这在一定程度上增加了 AK 管理的复杂性，并且降低了 AK 的保密性。甚至，如果您需要实现多地域一致性部署，AK 会随着镜像以及使用镜像创建的实例扩散出去。这种情况下，当您需要更换 AK 时，您就需要逐台更新和重新部署实例和镜像。

现在借助于 ECS 实例 RAM 角色，您可以将 *RAM 角色* 和 ECS 实例关联起来，实例内部的应用程序可以通过 STS 临时凭证访问其他云产品。其中 STS 临时凭证由系统自动生成和更新，应用程序可以使用指定的 实例元数据 URL 获取 STS 临时凭证，无需特别管理。同时借助于 RAM，通过对角色和授权策略的管理，您可以达到不同实例对不同云产品或相同云产品具有各自访问权限的目的。

本文以部署在 ECS 实例上的 Python 访问 OSS 为例，详细介绍了如何借助 ECS 实例 RAM 角色，使实例内部的应用程序可以使用 STS 临时凭证访问其他云产品。

> **说明：**
> 为了方便您随本文样例快速入门，文档里所有操作均在 *OpenAPI Explorer* 完成。OpenAPI Explorer 通过已登录用户信息获取当前账号临时 AK，对当前账号发起线上资源操作，请谨慎操作。创建实例操作会产生费用。操作完成后请及时释放实例。

操作步骤

为了使 ECS 借助实例 RAM 角色，实现内部 Python 可以使用 STS 临时凭证访问 OSS，您需要完成以下步骤：

步骤 1. 创建 RAM 角色并配置授权策略

步骤 2. 指定 RAM 角色创建并设置 ECS 实例

步骤 3. 在实例内部访问实例元数据 URL 获取 STS 临时凭证

步骤 4. 基于临时凭证，使用 Python SDK 访问 OSS

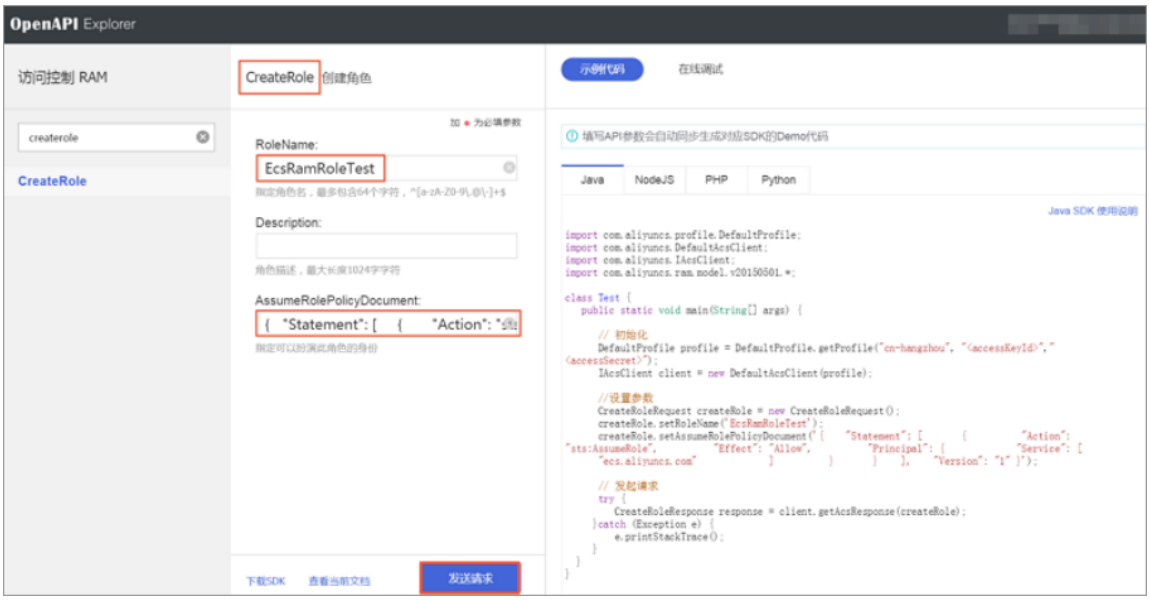
步骤 1. 创建 RAM 角色并配置授权策略

按以下步骤创建 RAM 角色并配置授权策略。

1. 创建 RAM 角色。找到 OpenAPI Explorer RAM 产品下 CreateRole API。其中：

   - **RoleName**：设置角色的名称。根据自己的需要填写，本示例中为 *EcsRamRoleTest*。
   - **AssumeRolePolicyDocument**： 填写如下内容，表示该角色为一个服务角色，受信云服务（本示例中为 ECS）可以扮演该角色。

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

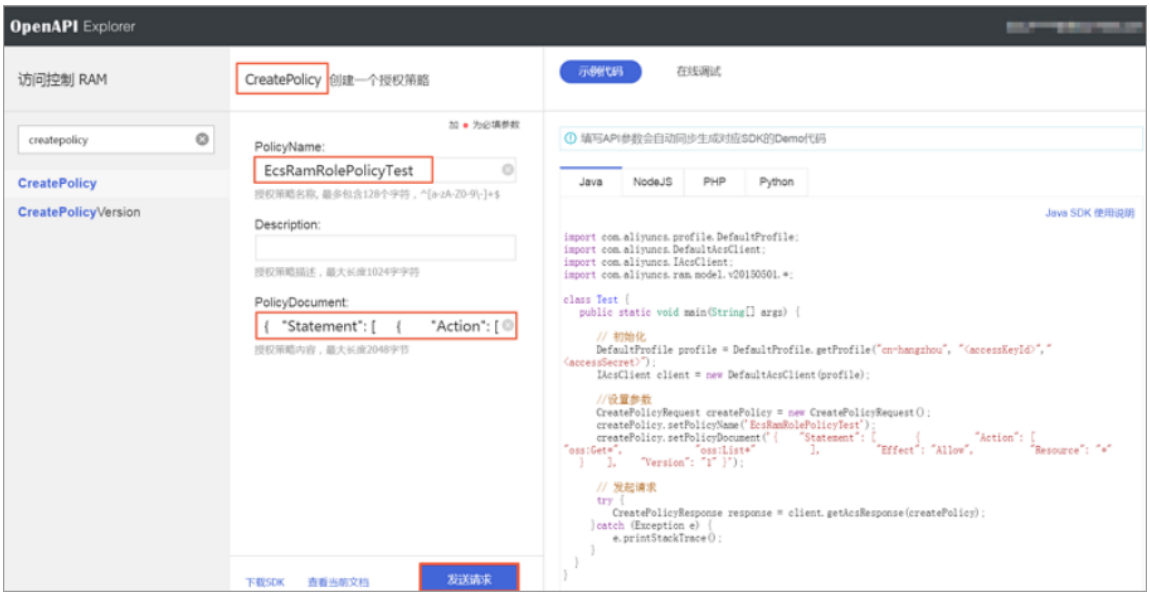


2. 创建授权策略。找到 OpenAPI Explorer RAM 产品下的 CreatePolicy API。其中：

   - **PolicyName**：设置授权策略的名称。本示例中为 *EcsRamRolePolicyTest*。
   - **PolicyDocument**：输入授权策略内容。本示例中填写如下内容，表示该角色具有 OSS 只读权限。

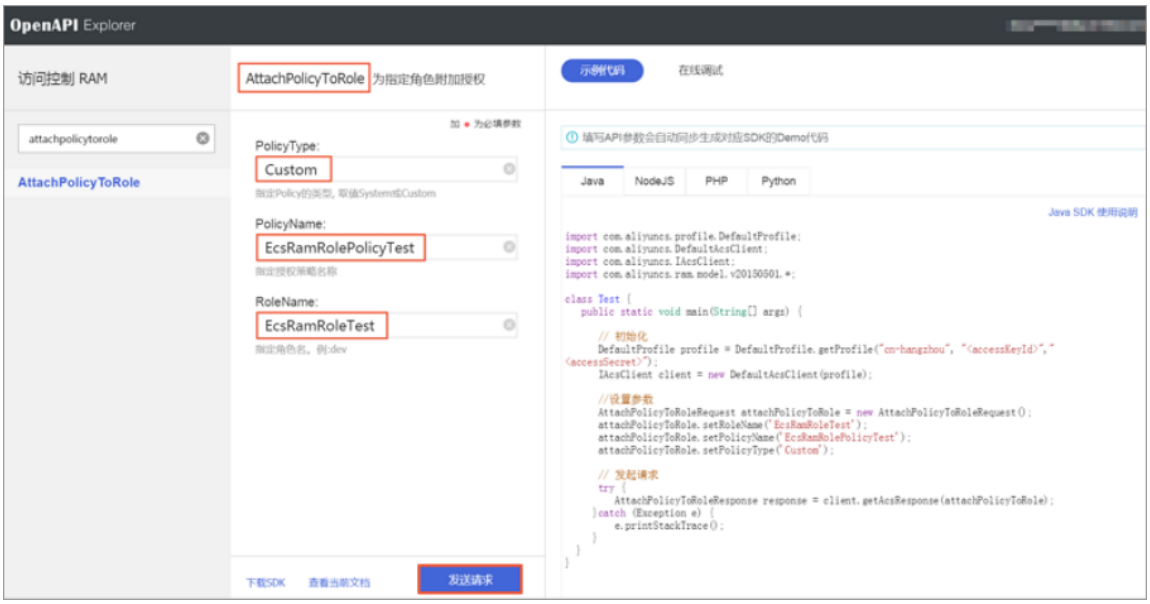
```
{
"Statement": [
{
"Action": [
"oss:Get*",
"oss:List*"
],
```

```
"Effect": "Allow",
"Resource": "*"
}
],
"Version": "1"
}
```



3. 为角色附加授权。找到 OpenAPI Explorer RAM 产品下 AttachPolicyToRole API。其中：

   - **PolicyType**：填写 *Custom*。
   - **PolicyName**：填写第 2 步创建的策略名称，如本示例中的 *EcsRamRolePolicyTest*。
   - **RoleName**：填写第 1 步创建的角色名称，如本示例中的 *EcsRamRoleTest*。



步骤 2. 为 ECS 实例指定 RAM 角色

您可以通过以下任一种方式为 ECS 实例指定 RAM 角色：

- 将实例 RAM 角色附加到一个已有的 VPC 类型ECS实例上
- 指定 RAM 角色创建并设置 ECS 实例

将实例 RAM 角色附加到一个已有的 VPC 类型ECS实例上

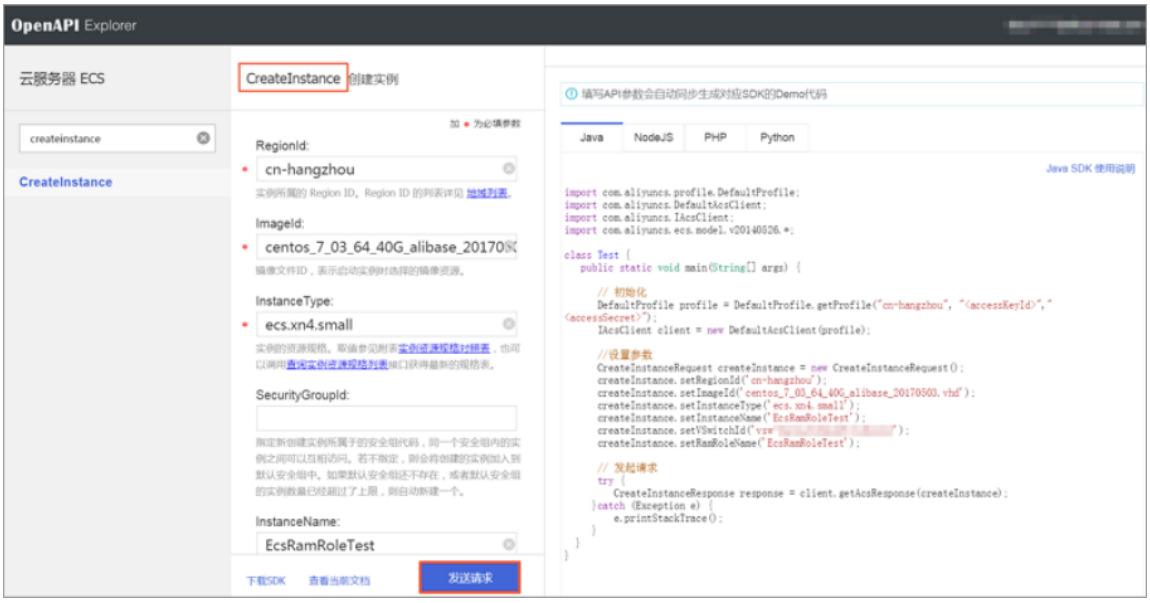
您可以使用 ECS 的 AttachInstanceRamRole API 附加实例 RAM 角色到已有的 VPC 类型 ECS 实例授权访问，设置信息如下：

- **RegionId**：为实例所在的地域 ID。
- **RamRoleName**：RAM 角色的名称。本示例中为 *EcsRamRoleTest*。
- **InstanceIds**：需要附加实例 RAM 角色的 VPC 类型 ECS 实例 ID。本示例中为 ["i-bXXXXXXXX"]。

指定 RAM 角色创建并设置 ECS 实例

按以下步骤指定 RAM 角色创建并设置 ECS 实例。

1. 创建实例。找到 OpenAPI Explorer ECS 产品下的 CreateInstance API，根据实际情况填写请求参数。必须填写的参数包括：
   - **RegionId**：实例所在地域。本示例中为 *cn-hangzhou*。
   - **ImageId**：实例的镜像。本示例中为 *centos_7_03_64_40G_alibase_20170503.vhd*。
   - **InstanceType**：实例的规格。本示例中为 *ecs.xn4.small*。
   - **VSwitchId**：实例所在的 VPC 虚拟交换机。因为 ECS 实例 RAM 角色目前只支持 VPC 类型 ECS 实例，所以 VSwitchId 是必需的。
   - **RamRoleName**：RAM 角色的名称。本示例中为 *EcsRamRoleTest*。



如果您希望授权子账号创建指定 RAM 角色的 ECS 实例，那么子账号除了拥有创建 ECS 实例的权限之外，还需要增加 PassRole 权限。所以，您需要创建一个如下所示的自定义授权策略并绑定到子账号上。如果是创建 ECS 实例，[ECS RAM Action] 可以是 `ecs:CreateInstance`，您也可以根据实际情况添加更多的权限。如果您需要为子账号授予所有 ECS 操作权限，[ECS RAM Action] 应该替换为 `ecs:*`。

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

2. 设置密码并启动实例。
3. 使用 API 或在控制台设置 ECS 实例能访问公网。

步骤 3. 在实例内部访问实例元数据 URL 获取 STS 临时凭证

按以下步骤获取实例的 STS 临时凭证。

> 说明：
> STS 临时凭证失效前半小时会生成新的 STS 临时凭证，在这半小时内，新旧 STS 临时凭证均可使用。

1. 远程连接实例。
2. 访问 `http://100.100.100.200/latest/meta-data/ram/security-credentials/EcsRamRoleTest` 获取 STS 临时凭证。路径最后一部分是 RAM 角色名称，您应替换为自己的创建的 RAM 角色名称。

> 说明：
> 本示例中使用 `curl` 命令访问上述 URL。如果您使用的是 Windows ECS 实例，请参见 实例元数据。

示例输出结果如下。

```
[root@local ~]# curl http://100.100.100.200/latest/meta-data/ram/
security-credentials/EcsRamRoleTest
{
"AccessKeyId" : "STS.J8XXXXXXXXX4",
"AccessKeySecret" : "9PjfXXXXXXXXBf2XAW",
"Expiration" : "2017-06-09T09:17:19Z",
"SecurityToken" : "CAIXXXXXXXXXXwmBkleCTkyI+",
"LastUpdated" : "2017-06-09T03:17:18Z",
"Code" : "Success"
}
```

步骤 4. 基于临时凭证，使用 **Python SDK** 访问 **OSS**

本示例中，我们基于 STS 临时凭证使用 Python SDK 列举实例所在地域的某个 OSS 存储空间（Bucket）里的 10 个文件。

前提条件

您已经远程连接到 ECS 实例。

您的 ECS 实例已经安装了 Python。如果您用的是 Linux ECS 实例，必须安装 pip。

您在实例所在的地域已经创建了存储空间（Bucket），并已经获取 Bucket 的名称和 Endpoint。本示例中，Bucket 名称为 `ramroletest`，Endpoint 为 `oss-cn-hangzhou.aliyuncs.com`。

操作步骤

按以下步骤使用 Python SDK 访问 OSS。

1. 运行命令 `pip install oss2`，安装 OSS Python SDK。

   说明：

   如果您用的是 Windows ECS 实例，参考 对象存储 *OSS SDK* 参考 的 安装 *Python SDK*。

2. 执行下述命令进行测试，其中：

   - `oss2.StsAuth` 中的 3 个参数分别对应于上述 URL 返回的 AccessKeyId、AccessKeySecret 和 SecurityToken。
   - `oss2.Bucket` 中后 2 个参数是 Bucket 的名称和 Endpoint。

```
 import oss2
 from itertools import islice
 auth = oss2.StsAuth(<AccessKeyId>, <AccessKeySecret>, <SecurityTo
ken>)
 bucket = oss2.Bucket(auth, <您的 Endpoint>, <您的 Bucket 名称>)
 for b in islice(oss2.ObjectIterator(bucket), 10):
```

```
    print(b.key)
```

示例输出结果如下。

```
[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 informatio
n.
>>> import oss2
>>> from itertools import islice
>>> auth = oss2.StsAuth("STS.J8XXXXXXXXX4", "9PjfXXXXXXXXBf2XAW",
 "CAIXXXXXXXXXXwmBkleCTkyI+")
>>> 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
```

9 磁盘缩容

由于目前云服务器 ECS 不支持系统盘或者数据盘缩容，如果您有磁盘缩容的需求，可用通过 阿里云迁云工具 达成目的。

迁云工具的研发初衷是为了平衡阿里云用户的云上及线下业务负载，但是您可以利用其工作原理，绕道实现云服务器 ECS 磁盘缩容。

迁云工具可以根据您的 ECS 实例重新制作一份自定义镜像，在制作过程中通过重新指定磁盘大小，以达到缩容的目的。除了将目标对象换成了 ECS 实例之外，磁盘缩容和迁云这两种场景的工具 使用方法和使用限制 完全一致。甚至因为使用对象为已经虚拟化的 ECS 实例，会更加方便，报错机率更低。

然而，这种缩容方式，会引起原有 ECS 实例的部分属性发生变化，例如，实例 **ID**（`InstanceId`）和 公网 **IP**。如果您的实例为 专有网络*#VPC#* 实例，可以将 公网*IP*转换为弹性公网*IP* 以保留该公网 IP。因此，建议使用 弹性公网 *IP#EIP#* 或者对公网 IP 依赖程度较轻的用户使用该方式缩容。

前提条件

- 当磁盘挂载的是 Linux 实例时，您需要预先在实例内安装远程数据同步工具 rsync。
  - CentOS 实例：运行 `yum install rsync -y`
  - Ubuntu 实例：运行 `apt-get install rsync -y`
  - Debian 实例：运行 `apt-get install rsync -y`
  - 其他发行版：参考发行版官网安装相关的文档
- 您需要预先在控制台 创建 *AccessKey*，用于输出到配置文件 *user_config.json* 里。

  说明：

  由于 AccessKey 权限过大，为防止数据泄露，建议您 创建 *RAM* 用户子账号，并使用 RAM 用户子账号 创建 *AccessKey*。
- 其他更多前提条件和限制条件，请参阅 使用迁云工具迁移服务器至阿里云。

操作步骤

1. 使用管理员/root 账号 远程连接 到目标 ECS 实例。
2. 下载 阿里云迁云工具 ZIP 压缩包。
3. 解压迁云工具，并进入对应操作系统及版本的客户端文件目录找到配置文件 user_config.json。
4. 参阅段落 自定义 *user_config.json* 完成配置。

该配置文件 Linux Shell 显示效果如下图所示。

在磁盘缩容的场景中，您需要重点关注的参数有：

- *system_disk_size*：该参数可以置为缩容系统盘的预期数值，单位为 GB，该值不能小于系统盘实际使用空间大小。
- *data_disks*：该参数可以置为缩容数据盘的预期数值，单位为 GB，该值不能小于数据盘实际使用空间大小。

> 说明：
>
> - 当 Linux 实例自带数据盘时，即使您不考虑缩容数据盘，也需要配置参数 data_disks，否则迁云工具默认将数据盘的数据拷贝到系统盘中。
> - 当 Windows 实例自带数据盘时，如果没有缩容数据盘的需求，可以不配置参数 data_disks。

5. 执行客户端主程序 go2aliyun_client.exe：
   - Windows 实例：右击 go2aliyun_client.exe，选择 以管理员身份运行。
   - Linux 实例：
     1. 运行 chmod +x go2aliyun_client 赋予客户端可执行权限。
     2. 运行 ./ go2aliyun_client 运行客户端。
6. 等待运行结果：
   - 当出现 Goto Aliyun Finished! 提示时，前往 *ECS* 控制台镜像详情页 查看经过缩容后的自定义镜像。如果自定义镜像已生成，您可以释放原实例，然后使用生成的自定义镜像 创建 *ECS* 实例，创建完成后，磁盘缩容工作已完成。
   - 当出现 Goto Aliyun Not Finished! 提示时，检查同一目录下 Logs 文件夹下的日志文件 排查故障。修复问题后，重新运行迁云工具即可恢复缩容工作，迁云工具会从上一次执行的进度中继续迁云，无需重头开始。

参考链接

- 关于迁云工具的具体介绍，请参阅什么是阿里云迁云工具。
- 关于迁云工具的操作说明，请参阅使用迁云工具迁移服务器至阿里云。

10 GPU实例最佳实践

10.1 在gn5实例上部署NGC环境

NGC（NVIDIA GPU CLOUD）是NVIDIA开发的一套深度学习生态系统，可以使开发者免费访问深度学习软件堆栈，建立适合深度学习的开发环境。

目前NGC在阿里云gn5实例作了全面部署，并且在镜像市场提供了针对NVIDIA Pascal GPU优化的NGC容器镜像。通过部署镜像市场的NGC容器镜像，开发者能简单快速地搭建NGC容器环境，即时访问优化后的深度学习框架，大大缩减产品开发以及业务部署的时间，实现开发环境的预安装；同时支持调优后的算法框架，并且保持持续更新。

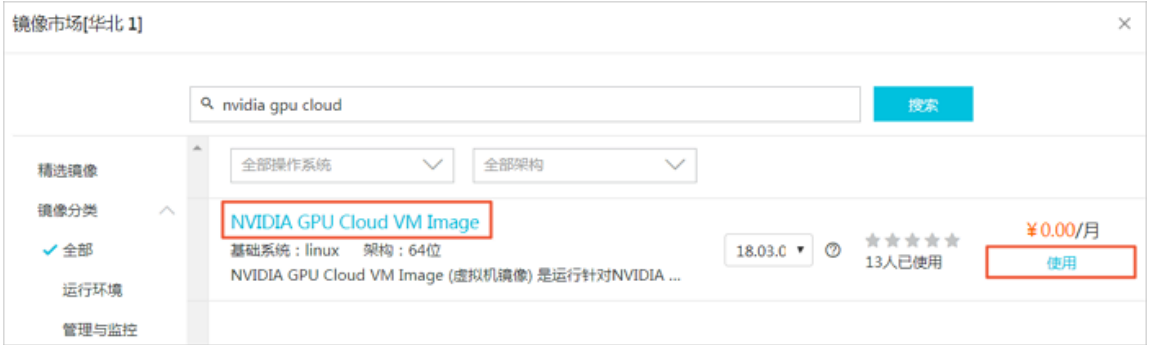
NGC网站 提供了目前主流深度学习框架不同版本的镜像（比如Caffe、Caffe2、CNTK、MxNet、TensorFlow、Theano、Torch），您可以选择需要的镜像搭建环境。本文以搭建TensorFlow深度学习框架为例详细介绍如何在gn5实例上搭建NGC环境。

在开始搭建TensorFlow环境之前，必须先完成以下工作：

- 注册阿里云账号，并完成 实名认证 。
- 登录 *NGC网站*，注册NGC账号。
- 登录 *NGC网站*，获取NGC API key并保存到本地。登录NGC容器环境时需要验证您的NGC API Key。

操作步骤

1. 创建gn5实例。参考 创建*ECS*实例 创建一台gn5实例，注意以下配置信息：
   - **地域**：只能选择 华北1、华北2、华北3、华北5、华东1、华东2、华南1。
   - **实例**：选择gn5实例规格。
   - **镜像**：单击 **镜像市场**，在弹出对话框里，找到 **NVIDIA GPU Cloud VM Image** 后，单击 **使用**。



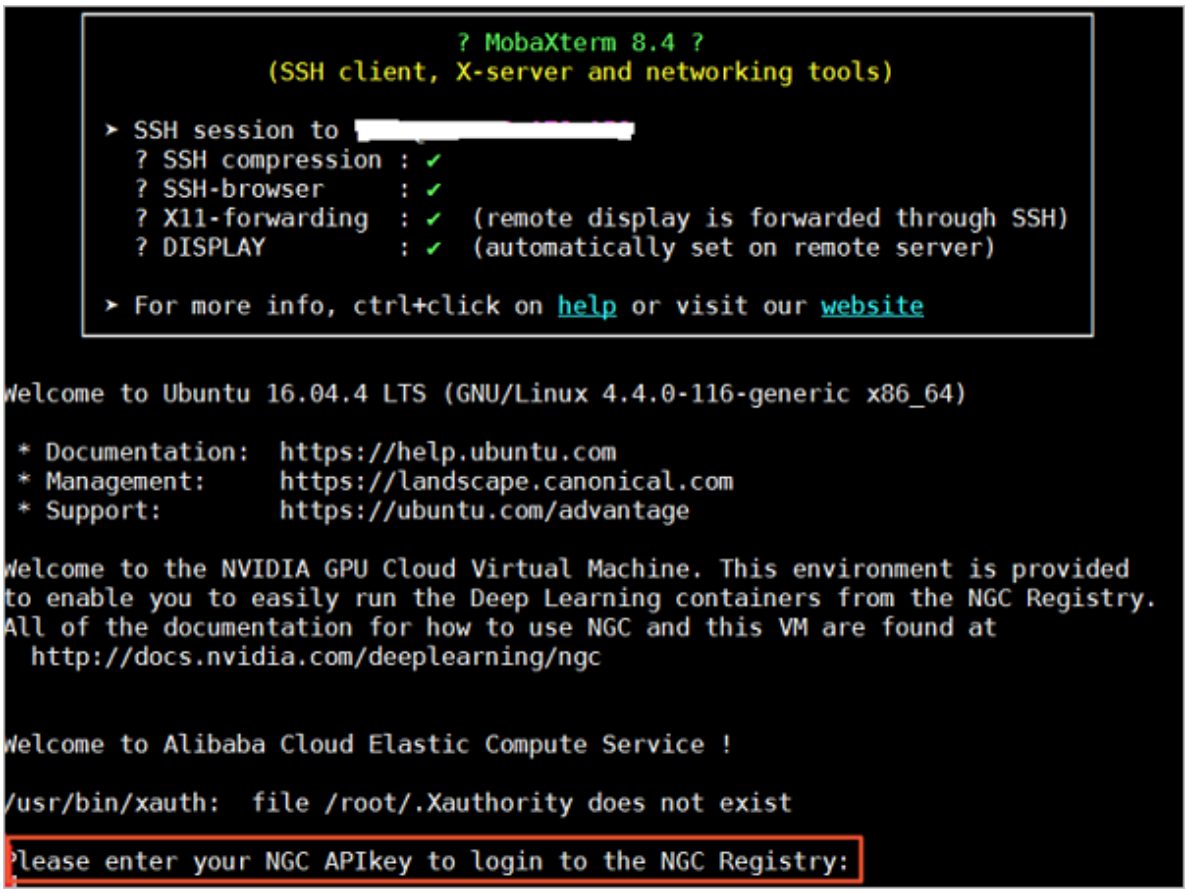
- 公网带宽：选择 分配公网IP地址。

  > 说明：
  > 如果这里不分配公网IP地址，则在实例创建成功后，绑定EIP地址。

- 安全组：选择一个安全组。安全组里必须开放 TCP 22 端口。如果您的实例需要支持HTTPS或 *DIGITS 6* 服务，必须开放TCP 443（用于HTTPS）或TCP 5000（用于DIGITS 6）端口。

ECS实例创建成功后，登录*ECS*管理控制台，记录实例的公网IP地址。

2. 连接ECS实例：根据创建实例时选择的登录凭证，使用密码验证连接*ECS*实例 或者 使用*SSH*密钥对验证连接*ECS*实例 。
3. 按界面提示输入NGC官网获取的NGC API Key后按回车键，即可登录NGC容器环境。

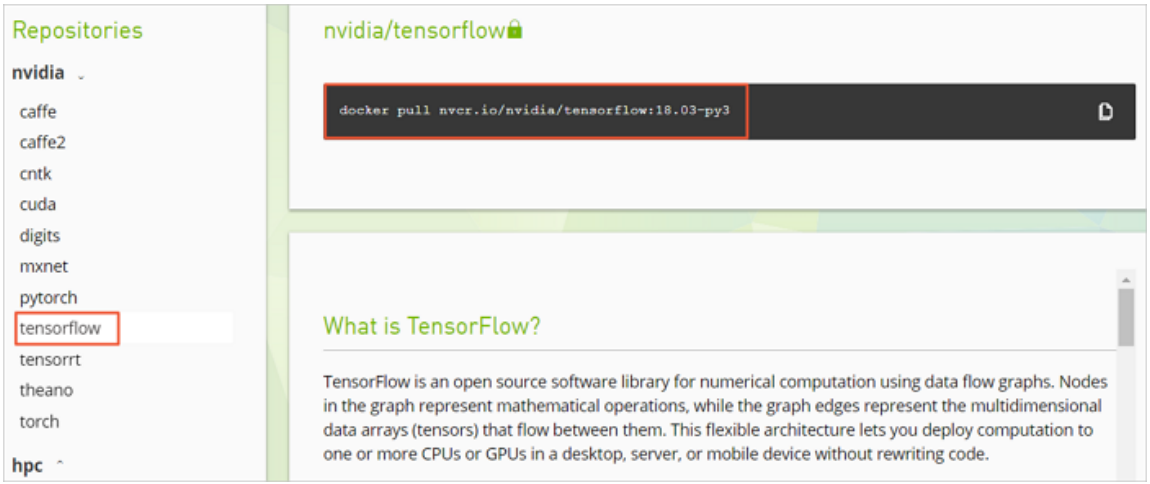


4. 运行 `nvidia-smi`。您能查看当前GPU的信息，包括GPU型号、驱动版本等，如下图所示。

```
root@ --                          ~# nvidia-smi
Thu Mar 29 20:50:01 2018
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 384.111                Driver Version: 384.111                   |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  Off  | 00000000:00:08.0 Off |                    0 |
| N/A   29C    P0    27W / 250W |      0MiB / 16276MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

5. 按以下步骤搭建TensorFlow环境：

   a. 登录 *NGC网站*，找到TensorFlow镜像页面，获取 `docker pull` 命令。



b. 下载TensorFlow镜像。

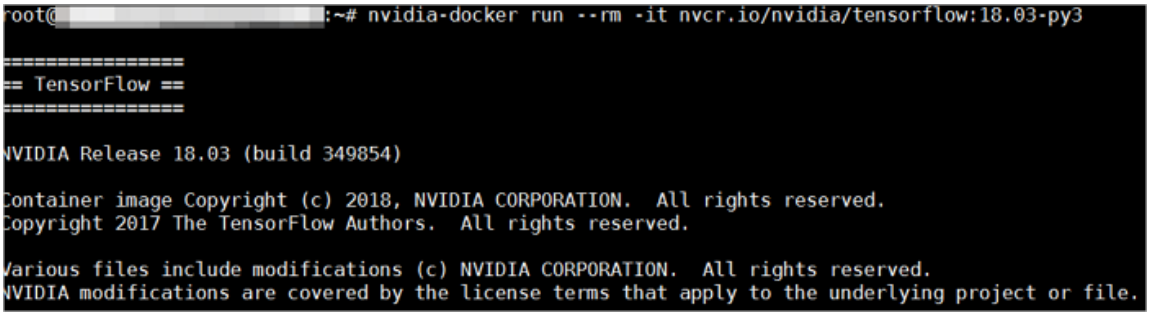
```
docker pull nvcr.io/nvidia/tensorflow:18.03-py3
```

c. 查看下载的镜像。

```
docker image ls
```

d. 运行容器，完成TensorFlow开发环境的部署。

```
nvidia-docker run --rm -it nvcr.io/nvidia/tensorflow:18.03-py3
```



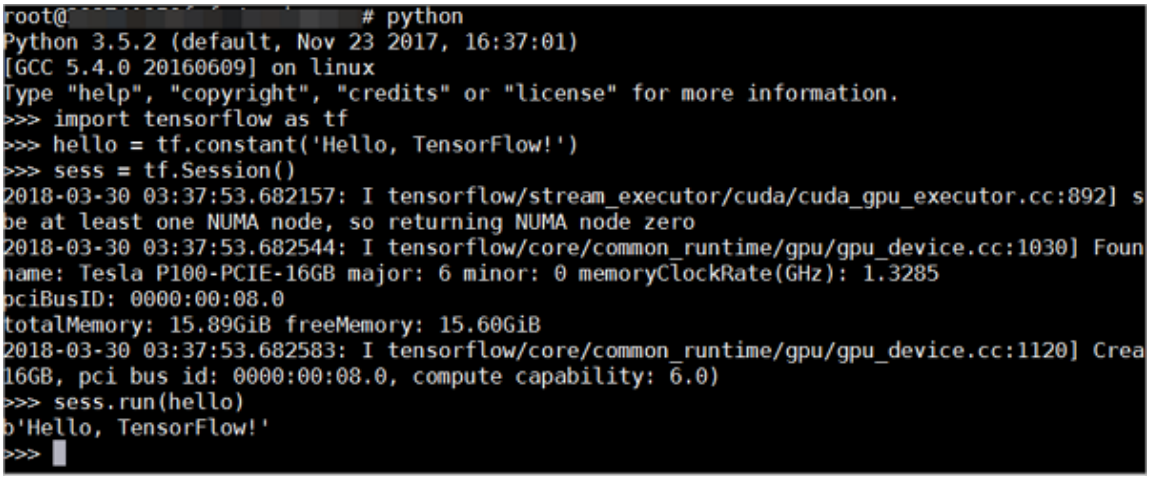
6. 选择以下任一种方式测试TensorFlow：

- 简单测试TensorFlow。

```
$python
```

```
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> sess.run(hello)
```

如果TensorFlow正确加载了GPU设备，返回结果如下图所示。

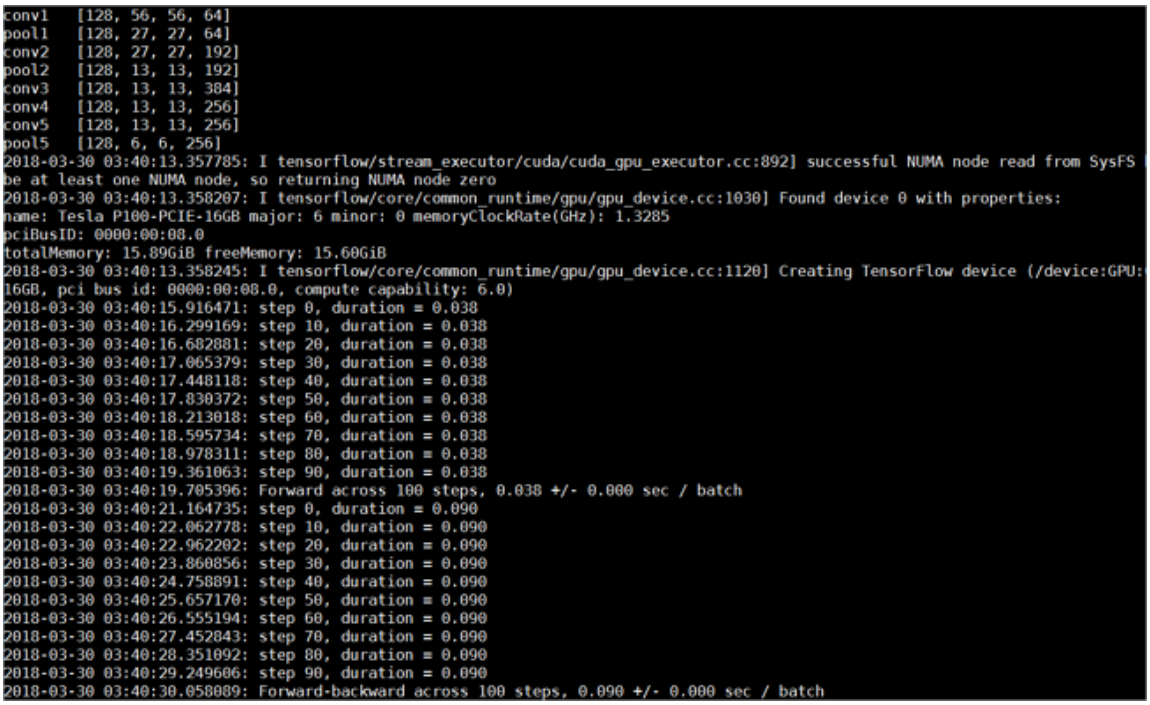


- 下载TensorFlow模型并测试TensorFlow。

```
git clone https://github.com/tensorflow/models.git
```

```
cd models/tutorials/image/alexnet
python alexnet_benchmark.py --batch_size 128 --num_batches 100
```

运行状态如下图所示。



7. 保存TensorFlow镜像的修改。否则，下次登录时配置会丢失。

11 Terraform

11.1 什么是Terraform

Terraform是一种开源工具，用于安全高效地预配和管理云基础结构。

HashiCorp Terraform 是一个IT基础架构自动化编排工具，可以用代码来管理维护 IT 资源。Terraform的命令行接口 (CLI) 提供一种简单机制，用于将配置文件部署到阿里云或其他任意支持的云上，并对其进行版本控制。

它编写了描述云资源拓扑的配置文件中的基础结构，例如虚拟机、存储帐户和网络接口。Terraform 的命令行接口（CLI）提供一种简单机制，用于将配置文件部署到阿里云或任何其他支持的云并对其进行版本控制。

Terraform是一个高度可扩展的工具，通过 Provider 来支持新的基础架构。您可以使用Terraform来创建、修改、删除ECS、VPC、RDS、SLB等多种资源。

优势

- 将基础结构部署到多个云

  Terraform适用于多云方案，将相类似的基础结构部署到阿里云、其他云提供商或者本地数据中心。开发人员能够使用相同的工具和相似的配置文件同时管理不同云提供商的资源。

- 自动化管理基础结构

  Terraform能够创建配置文件的模板，以可重复、可预测的方式定义、预配和配置ECS资源，减少因人为因素导致的部署和管理错误。能够多次部署同一模板，创建相同的开发、测试和生产环境。

- 基础架构即代码（**Infrastructure as Code**）

  可以用代码来管理维护资源。允许保存基础设施状态，从而使您能够跟踪对系统（基础设施即代码）中不同组件所做的更改，并与其他人共享这些配置 。

- 降低开发成本

  您通过按需创建开发和部署环境来降低成本。并且，您可以在系统更改之前进行评估。

应用场景

Terraform的应用场景请参见 *Terraform*详情页。

使用Terraform

Terraform能够让您在阿里云上轻松使用 简单模板语言 来定义、预览和部署云基础结构。以下为Terraform在ECS中预配资源的必要步骤：

1. 安装Terraform。
2. 配置Terraform。
3. 使用Terraform创建一台或多台ECS实例。

更多资料

- *Terraform Alibaba provider*文档
- *Terrafrom Alibaba github*
- *Terraform Registry Alibaba Modules*

11.2 安装和配置Terraform

在使用Terraform的简单模板语言定义、预览和部署云基础结构前，您需要安装预配置Terraform。

操作步骤

1. 前往 *Terraform*官网 下载适用于您的操作系统的程序包。
2. 将程序包解压到`/usr/local/bin`。

   如果将可执行文件解压到其他目录，按照以下方法为其定义全局路径：

   - Linux：参见 在*Linux*系统定义全局路径。
   - Windows：参见 在*Windows*系统定义全局路径。
   - Mac：参见 在*Mac*系统定义全局路径。
3. 运行`terraform`验证路径配置。

   将显示可用的Terraform选项的列表，类似如下所示，表示安装完成。

   ```
   username:~$ terraform
   Usage: terraform [-version] [-help] <command> [args]
   ```
4. 为提高权限管理的灵活性和安全性，建议您创建RAM用户，并为其授权。

   1. 登录 *RAM*控制台。
   2. 创建名为`Terraform`的RAM用户，并为该用户创建AccessKey。具体步骤参见 创建*RAM*用户。

3. 为RAM用户授权。在本示例中，给用户*Terraform*授予AliyunECSFullAccess和AliyunVPCFullAccess权限，具体步骤参见 为*RAM*用户授权。

5. 创建环境变量，用于存放身份认证信息。

```
export ALICLOUD_ACCESS_KEY="LTAIUrZCw3********"
export ALICLOUD_SECRET_KEY="zfwwWAMWIAiooj14GQ2*************"
export ALICLOUD_REGION="cn-beijing"
```

11.3 创建一台ECS实例

本文介绍如何使用Terraform创建一台ECS实例。

操作步骤

1. 创建VPC网络和交换机。

   1. 创建*terraform.tf*文件，输入以下内容，并保存在当前的执行目录中。

```
resource "alicloud_vpc" "vpc" {
  name       = "tf_test_foo"
  cidr_block = "172.16.0.0/12"
}

resource "alicloud_vswitch" "vsw" {
  vpc_id            = "${alicloud_vpc.vpc.id}"
  cidr_block        = "172.16.0.0/21"
  availability_zone = "cn-beijing-b"
}
```

   2. 运行terraform apply开始创建。

   3. 运行terraform show查看已创建的VPC和VSwitch。

      您也可以登录VPC控制台查看VPC和VSwitch的属性。

2. 创建安全组，并将安全组作用于上一步创建的VPC中。

   1. 在*terraform.tf*文件中增加以下内容。

```
resource "alicloud_security_group" "default" {
  name = "default"
  vpc_id = "${alicloud_vpc.vpc.id}"
}

resource "alicloud_security_group_rule" "allow_all_tcp" {
  type              = "ingress"
  ip_protocol       = "tcp"
  nic_type          = "internet"
  policy            = "accept"
  port_range        = "1/65535"
  priority          = 1
  security_group_id = "${alicloud_security_group.default.id}"
  cidr_ip           = "0.0.0.0/0"
```

```
}
```

2. 运行`terraform apply`开始创建。
3. 运行`terraform show`查看已创建的安全组和安全组规则。

   你也可以登录ECS控制台查看安全组和安全组规则。

3. 创建ECS实例。

1. 在*terraform.tf*文件中增加以下内容。

```
resource "alicloud_instance" "instance" {
  # cn-beijing
  availability_zone = "cn-beijing-b"
  security_groups = ["${alicloud_security_group.default.*.id}"]

  # series III
  instance_type        = "ecs.n2.small"
  system_disk_category = "cloud_efficiency"
  image_id             = "ubuntu_140405_64_40G_cloudinit_20161115.
vhd"
  instance_name        = "test_foo"
  vswitch_id = "${alicloud_vswitch.vsw.id}"
  internet_max_bandwidth_out = 10
  password = "<replace_with_your_password>"
}
```

说明：

- 在上述示例中，指定了`internet_max_bandwidth_out = 10`，因此会自动为实例分配一个公网IP。
- 详细的参数解释请参见 阿里云参数说明。

2. 运行`terraform apply`开始创建。
3. 运行`terraform show`查看已创建的ECS实例。
4. 运行**`ssh root@<publicip>`**，并输入密码来访问ECS实例。

```
provider "alicloud" {}

resource "alicloud_vpc" "vpc" {
  name       = "tf_test_foo"
  cidr_block = "172.16.0.0/12"
}

resource "alicloud_vswitch" "vsw" {
  vpc_id            = "${alicloud_vpc.vpc.id}"
  cidr_block        = "172.16.0.0/21"
  availability_zone = "cn-beijing-b"
}


resource "alicloud_security_group" "default" {
```

```
  name = "default"
  vpc_id = "${alicloud_vpc.vpc.id}"
}


resource "alicloud_instance" "instance" {
  # cn-beijing
  availability_zone = "cn-beijing-b"
  security_groups = ["${alicloud_security_group.default.*.id
}"]

  # series III
  instance_type        = "ecs.n2.small"
  system_disk_category = "cloud_efficiency"
  image_id             = "ubuntu_140405_64_40G_cloudinit
_20161115.vhd"
  instance_name        = "test_foo"
  vswitch_id = "${alicloud_vswitch.vsw.id}"
  internet_max_bandwidth_out = 10
}


resource "alicloud_security_group_rule" "allow_all_tcp" {
  type              = "ingress"
  ip_protocol       = "tcp"
  nic_type          = "intranet"
  policy            = "accept"
  port_range        = "1/65535"
  priority          = 1
  security_group_id = "${alicloud_security_group.default.id
}"
  cidr_ip           = "0.0.0.0/0"
}
```

11.4 创建多台ECS实例

本文介绍如何使用Terraform模块批量创建多台ECS实例。

操作步骤

1. 创建VPC网络和交换机。
   1. 创建*terraform.tf*文件，输入以下内容，保存在当前的执行目录中。

```
resource "alicloud_vpc" "vpc" {
  name       = "tf_test_foo"
  cidr_block = "172.16.0.0/12"
}

resource "alicloud_vswitch" "vsw" {
  vpc_id            = "${alicloud_vpc.vpc.id}"
  cidr_block        = "172.16.0.0/21"
  availability_zone = "cn-beijing-b"
}
```

   2. 运行`terraform apply`开始创建。
   3. 运行`terraform show`查看已创建的VPC和VSwitch。

您也可以登录VPC控制台查看VPC和VSwitch的属性。

2. 创建安全组，并将安全组作用于上一步创建的VPC中。

   1. 在*terraform.tf*文件中增加以下内容。

```
resource "alicloud_security_group" "default" {
  name = "default"
  vpc_id = "${alicloud_vpc.vpc.id}"
}

resource "alicloud_security_group_rule" "allow_all_tcp" {
  type              = "ingress"
  ip_protocol       = "tcp"
  nic_type          = "internet"
  policy            = "accept"
  port_range        = "1/65535"
  priority          = 1
  security_group_id = "${alicloud_security_group.default.id}"
  cidr_ip           = "0.0.0.0/0"
}
```

   2. 运行`terraform apply`开始创建。

   3. 运行`terraform show`查看已创建的安全组和安全组规则。

      你也可以登录ECS控制台查看安全组和安全组规则。

3. 使用Module创建多台ECS实例。在本示例中，创建3台ECS实例。

   1. 在*terraform.tf*文件中增加以下内容。

```
module "tf-instances" {
  source = "alibaba/ecs-instance/alicloud"
  vswitch_id = "${alicloud_vswitch.vsw.id}"
  group_ids = ["${alicloud_security_group.default.*.id}"]
  availability_zone = "cn-beijing-b"
  disk_category = "cloud_ssd"
  disk_name = "my_module_disk"
  disk_size = "50"
  number_of_disks = 7

  instance_name = "my_module_instances_"
  host_name = "sample"
  internet_charge_type = "PayByTraffic"
  number_of_instances = "3"
  password="User@123"
}
```

      说明：

      - 在上述示例中，指定了`internet_max_bandwith_out = 10`，因此会自动为实例分配一个公网IP。
      - 详细的参数解释请参见 参数说明。

2. 运行`terraform apply`开始创建。
3. 运行`terraform show`查看已创建的ECS实例。
4. 运行**`ssh root@<publicip>`**，并输入密码来访问ECS实例。

```
provider "alicloud" {}

resource "alicloud_vpc" "vpc" {
  name       = "tf_test_foo"
  cidr_block = "172.16.0.0/12"
}

resource "alicloud_vswitch" "vsw" {
  vpc_id            = "${alicloud_vpc.vpc.id}"
  cidr_block        = "172.16.0.0/21"
  availability_zone = "cn-beijing-b"
}

resource "alicloud_security_group" "default" {
  name = "default"
  vpc_id = "${alicloud_vpc.vpc.id}"
}


resource "alicloud_security_group_rule" "allow_all_tcp" {
  type              = "ingress"
  ip_protocol       = "tcp"
  nic_type          = "intranet"
  policy            = "accept"
  port_range        = "1/65535"
  priority          = 1
  security_group_id = "${alicloud_security_group.default.id
}"
  cidr_ip           = "0.0.0.0/0"
}

module "tf-instances" {
  source = "alibaba/ecs-instance/alicloud"
  vswitch_id = "${alicloud_vswitch.vsw.id}"
  group_ids = ["${alicloud_security_group.default.*.id}"]
  availability_zone = "cn-beijing-b"
  disk_category = "cloud_ssd"
  disk_name = "my_module_disk"
  disk_size = "50"
  number_of_disks = 7

  instance_name = "my_module_instances_"
  host_name = "sample"
  internet_charge_type = "PayByTraffic"
  number_of_instances = "3"
  password="User@123"
```

```
}
```

11.5 部署Web集群

部署一个网站或者API应用时，需要部署一系列的节点，并根据访问数量或者资源使用的情况来自动伸缩，SLB对各个节点分配请求。本文介绍如何使用Terraform部署Web集群。

背景信息

在本示例中，整个应用部署在一个可用区，并且只提供8080端口访问hello world网页。

操作步骤

1. 创建VPC网络和交换机。
   1. 创建*terraform.tf*文件，输入以下内容，并保存在当前的执行目录中。

```
resource "alicloud_vpc" "vpc" {
  name       = "tf_test_foo"
  cidr_block = "172.16.0.0/12"
}

resource "alicloud_vswitch" "vsw" {
  vpc_id            = "${alicloud_vpc.vpc.id}"
  cidr_block        = "172.16.0.0/21"
  availability_zone = "cn-beijing-b"
}
```

   2. 运行`terraform apply`开始创建。
   3. 运行`terraform show`查看已创建的VPC和VSwitch。

      您也可以登录VPC控制台查看VPC和VSwitch的属性。
2. 创建安全组，并将安全组作用于上一步创建的VPC中。
   1. 在*terraform.tf*文件中增加以下内容。

```
resource "alicloud_security_group" "default" {
  name = "default"
  vpc_id = "${alicloud_vpc.vpc.id}"
}

resource "alicloud_security_group_rule" "allow_all_tcp" {
  type              = "ingress"
  ip_protocol       = "tcp"
  nic_type          = "internet"
  policy            = "accept"
  port_range        = "1/65535"
  priority          = 1
  security_group_id = "${alicloud_security_group.default.id}"
  cidr_ip           = "0.0.0.0/0"
```

```
}
```

2. 运行`terraform apply`开始创建。
3. 运行`terraform show`查看已创建的安全组和安全组规则。

   你也可以登录ECS控制台查看安全组和安全组规则。

3. 创建负载均衡实例，为其分配公网IP。在本示例中，为负载均衡实例配置了从前端80端口到后端8080端口的映射，并输出公网IP用于后续测试。

   1. 创建*slb.tf*文件，并增加以下内容。

```
resource "alicloud_slb" "slb" {
  name       = "test-slb-tf"
  vswitch_id = "${alicloud_vswitch.vsw.id}"
  internet = true
}
resource "alicloud_slb_listener" "http" {
  load_balancer_id = "${alicloud_slb.slb.id}"
  backend_port = 8080
  frontend_port = 80
  bandwidth = 10
  protocol = "http"
  sticky_session = "on"
  sticky_session_type = "insert"
  cookie = "testslblistenercookie"
  cookie_timeout = 86400
  health_check="on"
  health_check_type = "http"
  health_check_connect_port = 8080
}

output "slb_public_ip"{
  value = "${alicloud_slb.slb.address}"
}
```

   2. 运行`terraform apply`开始创建。
   3. 运行`terraform show`查看已创建的负载均衡实例。

      你也可以登录SLB控制台查看新建的负载均衡实例。

4. 创建弹性伸缩。

   在本示例中，将创建以下资源：

   - 伸缩组：在模版中指定伸缩最小为2，最大为10，并将伸缩组与新建的负载均衡实例绑定。由于伸缩组的配置要求SLB必须有相应配置的监听器，因此模版中用depends_on属性指定了部署顺序。
   - 伸缩组配置：在模版中指定ECS实例的具体配置。在初始化配置（user-data）中生成一个Hello World的网页，并在8080端口提供服务。为简化操作，本示例中会为虚拟机分配公网IP，并且设置`force_delete=true`用于后续删除环境。

- 伸缩规则：定义具体的伸缩规则。

1. 创建*ess.tf*文件，并增加以下内容。

```
resource "alicloud_ess_scaling_group" "scaling" {
  min_size = 2
  max_size = 10
  scaling_group_name = "tf-scaling"
  vswitch_ids=["${alicloud_vswitch.vsw.*.id}"]
  loadbalancer_ids = ["${alicloud_slb.slb.*.id}"]
  removal_policies   = ["OldestInstance", "NewestInstance"]
  depends_on = ["alicloud_slb_listener.http"]
}

resource "alicloud_ess_scaling_configuration" "config" {
  scaling_group_id = "${alicloud_ess_scaling_group.scaling.id}"
  image_id = "ubuntu_140405_64_40G_cloudinit_20161115.vhd"
  instance_type = "ecs.n2.small"
  security_group_id = "${alicloud_security_group.default.id}"
  active=true
  enable=true
  user_data = "#!/bin/bash\necho \"Hello, World\" > index.html\
nnohup busybox httpd -f -p 8080&"
  internet_max_bandwidth_in=10
  internet_max_bandwidth_out= 10
  internet_charge_type = "PayByTraffic"
  force_delete= true

}

resource "alicloud_ess_scaling_rule" "rule" {
  scaling_group_id = "${alicloud_ess_scaling_group.scaling.id}"
  adjustment_type  = "TotalCapacity"
  adjustment_value = 2
  cooldown = 60
}
```

2. 运行`terraform apply`开始创建。

   创建成功后，会输出SLB的公网IP。

3. 等待大约两分钟，弹性伸缩将自动创建ECS实例。

4. 输入命令`curl http://<slb public ip>`进行验证。

   如果看到`Hello World`，表示成功通过负载均衡实例访问ECS实例提供的网页。

5. 运行`terraform destroy`删除测试环境。经确认后，整个部署的环境将被删除。

   使用Terraform可以便捷地删除和重新部署一个环境。如果您想重新部署，运行`terraform apply`即可。