访问控制

用户指南

用户指南

《RAM 用户指南》是对 RAM 产品核心功能及其应用场景的详细介绍。

RAM 的核心功能主要包括用户身份与授权管理，应用场景可以覆盖企业子账号与分权管理、针对移动 app 的临时授权管理，和不同组织之间的资源互操作与授权管理。

身份管理部分

- 用户身份管理
- 对用户进行分组管理
- 角色身份管理

授权管理部分

- 授权策略（Policy）管理
- 对用户或角色身份进行授权
- 授权策略语言

典型使用场景

- 企业子账号与分权管理
- 针对不可信客户端 app 的临时授权管理
- 不同组织之间的资源互操作与授权管理

限制

限制项	限制值
用户总数	100
组总数	20
每个用户可以加入的组	5
每个用户允许创建AccessKey	2
每个用户可绑定MFA数	1
虚拟 MFA 设备数	100
自定义授权策略数	50
自定义策略版本数	5
附加给用户的授权策略数	5

附加给组的授权策略数	5
用户名字符数	64
组名字符数	64
授权策略名称字符数	128
角色名称字符数	64
角色数	100
别名字符数	3-64
自定义授权策略字符数	2048

身份管理

用户，是 RAM 中用到的一种身份；它对应到某一个操作实体，如操作员或应用程序。如果有新的用户或应用程序需要访问您的云资源，您需要创建 RAM 用户并授权其访问相关资源。一般操作步骤如下：

1. 主账户（或拥有 RAM 操作权限的 RAM 用户）登录到 RAM 控制台。
2. 创建 RAM 用户，并将该用户 添加到一个或多个组。
3. 给用户（或其所属的组）添加一个或多个 授权策略。
4. 设置用户密钥。如果用户是通过控制台进行操作，则需为用户设置登录密码；如果用户是通过 API 进行调用，则需为用户创建 API 访问密钥（Access Key）。
5. 如果用户需要使用特权操作（如停止虚拟机），那么可以为用户设置多因素认证（MFA），并要求用户必须使用 MFA 口令才能登录到阿里云控制台。
6. 向用户提供登录 URL，用户名及其登录密码。

基本设置

设置 企业别名、密码强度 及 子用户登录限制。

设置企业别名

操作步骤如下：

登录到 RAM 控制台。

依次选择 **设置** > **企业别名设置** > **编辑企业别名**。

输入 **企业别名**，并单击 **确定**，完成设置。

设置 RAM 用户的密码策略

操作步骤如下：

登录到 RAM 控制台。

依次选择 **设置** > **密码强度设置**。

按照页面提示，配置密码长度、字符格式、有效期、重试约束策略等规则，完成后单击 **保存修改**，使规则生效。

注意：一旦设置成功，该密码策略适用于所有 RAM 用户。

设置子用户安全限制

登录到 RAM 控制台。

依次选择 **设置** > **子用户安全设置**。

对于子用户，勾选是否:

- 允许登录时保存 MFA 登录状态（保存7天）
- 允许自主管理密码（重置）
- 允许自主管理 AccessKey（创建、禁用、删除）
- 允许自主管理多因素设备（启用、禁用）

完成后单击 **保存修改**，使设置生效。

创建 RAM 用户

操作步骤如下：

登录到 RAM 控制台。

依次选择 **用户管理** > **新建用户**。

按照页面提示，输入用户信息，完成后单击 **确认**，完成创建。

创建 RAM 用户后，需要根据使用需求为 RAM 用户 **设置登录密码**、**创建访问密钥** 及 **设置多因素认证设备**（MFA）。

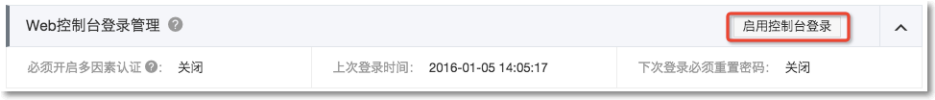
设置登录密码

为需要通过控制台进行操作的用户设置登录密码，操作步骤如下：

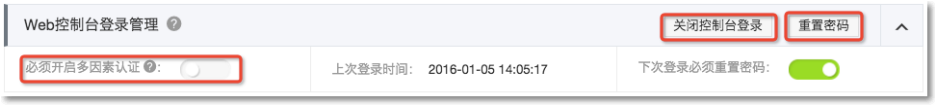
登录到 RAM 控制台。

在用户清单中找到需要设置登录密码的用户（可使用用户名进行模糊查询），单击其用户名或其操作列下的 **管理**，进入 **用户详情** 页面。

单击 **启用控制台登录**，在弹窗中为用户设置初始密码，并可以指定用户登录时必须更换密码。



登录密码设置成功后，可以进一步设置 **多因素认证**、**重置密码**，及 **关闭控制台登录**。



创建访问密钥（AK）

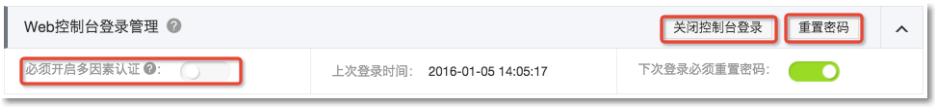
用户的访问密钥 AccessKey 相当于登录密码，只是使用场景不同。AccessKey 用于程序方式调用云服务 API，登录密码用于登录控制台。如果用户不需要调用 API，那么就不需要创建 AccessKey。

创建 AccessKey 的操作步骤如下：

登录到 RAM 控制台。

在用户清单中找到需要创建 AccessKey 的用户（可使用用户名进行模糊查询），单击其用户名或其操作列下的 **管理**，进入 **用户详情** 页面。

单击 **创建 AccessKey**，在弹窗中查看新建的 AccessKey 信息，并可选择 **保存 AK 信息**。



注意：

- 新创建的 AccessKey 只会在创建时显示，安全起见 RAM 并不提供查询接口，请您妥善保管。
- 如果 AccessKey 泄露或丢失，则需要创建新的 AccessKey。

为用户设置 MFA

多因素认证（Multi-Factor Authentication, MFA）是一种简单有效的最佳安全实践方法，它能够在用户名和密码之外再额外增加一层安全保护。启用 MFA 后，用户登录阿里云时，系统将要求输入用户名和密码（第一安全要素），然后要求输入来自其 MFA 设备的可变验证码（第二安全要素）。这些多重要素结合起来将为您的账户提供更高的安全保护。

虚拟 MFA 设备是产生一个 6 位数字验证码的应用程序，它遵循基于时间的一次性密码 (TOTP) 标准 （RFC 6238）。此应用程序可在移动硬件设备上运行（例如智能手机）。

设置 MFA 的操作步骤如下：

登录到 RAM 控制台。

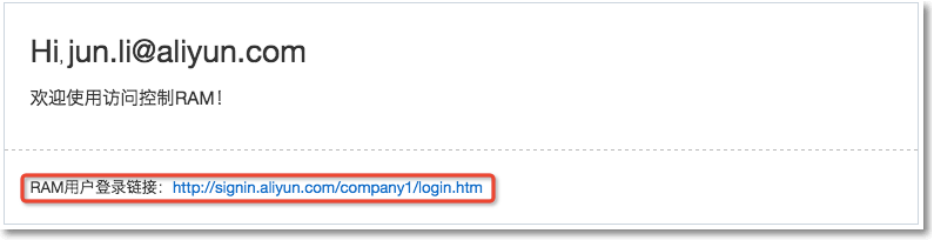
在用户清单中找到需要设置 MFA 的用户（可使用用户名进行模糊查询），单击其用户名或其操作列下的 **管理**，进入 **用户详情** 页面。

单击 **启用虚拟 MFA 设备**，启动 绑定 MFA 设备 流程。

RAM 用户登录

RAM 用户不同于云账户，登录入口也是有所区别，RAM 用户不能通过云账户登录页面进行登录。

在 RAM 控制台的概览页中，您可以找到 RAM 用户登录链接：



RAM 用户可以通过该登录 URL 登录到阿里云控制台：

注意：RAM 用户默认是没有任何访问权限的。如果没有被授权，即使能登入控制台，但仍然无权做任何操作。了解如何给 RAM 用户授权，请参考 **授权**。

对云账号下有多个 RAM 用户的情况，为更好的管理用户及其权限，建议您使用群组（Group）。为职责相同的 RAM 用户创建群组进行归类，并在授权时选择 **给群组授权**；这样，

- 在具体用户职责发生变化时，只用将其移动到相应职责的群组下，不会对其他用户产生影响。
- 当群组的权限发生变化时，只用修改群组的授权策略，可以直接应用到所有用户身上。

本文具体介绍了 **创建**、**重命名**、**删除** 群组，以及 **管理组成员**、**给群组授权** 的操作流程。

创建群组

操作步骤如下：

登录到 RAM 控制台。

依次选择 **群组管理** > **新建群组**。

输入群组名称，并单击 **确认**，完成群组创建。

组成员管理

操作步骤如下：

登录到 RAM 控制台。

单击 **群组管理**。

在群组清单中找到要管理的群组（可使用组名称进行模糊查询），单击其操作列下的 **编辑组成员**。

从左列中选择要添加到组中的用户（可使用关键字查询），单击右向箭头将其添加到右侧已选列下；选择右侧已选列下的用户，单击左向箭头可撤销选择。选择完成后单击 **确认**，完成组成员编辑。

编辑完成后，进入 **组成员管理** 页面（群组清单中单击 **组名称** 或其操作列下的 **管理**）查看组成员清单，单击成员后的 **移除出组** 可将其从当前群组中删除。

重命名群组

操作步骤如下：

登录到 RAM 控制台。

单击 **群组管理**。

在群组清单中找到要重命名的群组（可使用组名称进行模糊查询），单击其组名称或其操作列下的 **管理**，进入 **群组详情** 页面。

单击 **编辑基本信息**。

输入 **组名称** 并点击 **确认**， 完成修改。

删除群组

操作步骤如下：

登录到 RAM 控制台。

单击 **群组管理**。

在群组清单中找到要删除的群组（可使用组名称进行模糊查询），单击其操作列下的 **删除**。

注意：如果群组有包含组成员或者有绑定的授权策略，那么需要选定 **强制解除关联关系** 才能删除群组。

给群组授权

关于群组的授权管理，请参考 **授权**。

角色，与用户一样，都是 RAM 中使用的身份。与 RAM 用户相比，RAM 角色是一种虚拟用户，它没有确定的身份认证密钥，且需要被一个受信的实体用户扮演才能正常使用。

本文详细解释了角色的 **概念** 和 **应用场景**，帮助您正确理解；也介绍了 RAM 角色的 **类型**、**创建方法** 和 **使用方法**，指导您正确实践。

注意：如果没有特别说明，文中出现的 **角色** 都是指 **RAM 角色**。

理解 RAM 角色

RAM 角色（RAM-Role）是一种虚拟用户（或影子账号），它是 RAM 用户类型的一种。

RAM 角色不同于教科书式角色（Textbook-Role）。教科书式角色（或传统意义上的角色）是指一组权限集合，类似于 RAM 里的授权策略（Policy）。如果一个用户被赋予了这种角色，也就意味着该用户被赋予了一组权限，然后该用户就能访问被授权的资源。

RAM 角色作为虚拟用户，它有确定的身份，可以被赋予一组授权策略（Policy），但它没有确定的身份认证密钥（登录密码或 AccessKey）。

虚拟用户 vs 实体用户： 虚拟用户与实体用户的区别在于是否能被直接身份认证：

- 实体用户拥有确定的登录密码或 AccessKey，比如云账号、RAM-User 账号、云服务账号。
- 虚拟用户没有确定的认证密钥，比如 RAM-Role。

相比于 RAM 用户，在使用方法上 RAM 角色需要被一个授信的实体用户扮演，扮演成功后实体用户将获得 RAM 角色的临时安全令牌，使用这个临时安全令牌就能以角色身份访问被授权的资源。

使用须知

RAM-Role 必须与一种实体用户身份 **联合** 起来才能使用。

如果一个实体用户要想使用被赋予的某个 RAM 角色，实体用户必须先以自己身份登录，然后执行 **切换到角色** 操作将自己从 **实体身份** 切换到 **角色身份**。

当切换到角色身份后，将只能执行该角色身份被授权的所有操作，而登录时实体身份所对应的访问权限被隐藏。

如果用户希望从 **角色身份** 回到 **实体身份**，那么只需执行 **切回登录身份** 操作。

此时将拥有实体身份所对应的访问权限，而不再拥有角色身份所拥有的权限。

相关概念

与 RAM 角色相关的概念间关系释义如下图所示：

相关概念的具体释义见下表：

名称	释义
RoleARN	RoleARN 是角色的全局资源描述符，用来指定具体角色。 - RoleARN 遵循阿里云 ARN 的命名规范。比如，某个云账号下的 devops 角色的 ARN 为：acs:ram:*:1234567890123456:role/devops。 - 创建角色后，可在其 **角色详情** 页查看其 Arn。
受信演员	角色的受信演员是指可以扮演角色的实体用户身份。 - 创建角色时必须指定受信演员，角色只能被受信的演员扮演。 - 受信演员可以是受信的云账号，或者受信服务。
授权策略	一个角色可以绑定一组授权策略（Policy）。没有绑定授权策略的角色也可以存在，但不能使用。
扮演角色	扮演角色（AssumeRole）是实体用户获取角色身份的安全令牌的方法。 一个实体用户通过调用 AssumeRole 的 API 可以获得角色的安全令牌，使用安全令牌可以访问云服务 API。
切换身份	切换身份（SwitchRole）是在控制台中实体用户从当前登录身份切换到角色身份的方法。

	- 一个实体用户登录到控制台之后，可以切换到被许可扮演的某一种角色身份，然后以角色身份操作云资源。切换到角色身份后，原实体用户身份的访问权限将被屏蔽。 - 用户不需要使用角色身份时，可以从角色身份切换回原来的登录身份。
角色令牌	角色令牌是角色身份的一种临时访问密钥。 角色身份没有确定的访问密钥，当一个实体用户要使用角色时，必须通过扮演角色来获取对应的角色令牌，然后使用角色令牌来调用阿里云服务 API。

RAM 角色应用场景

RAM 角色主要用于解决委托其他云账号及其下 RAM 用户操作您所控制的资源、委托云服务操作您所控制的资源。

跨账号的资源操作与授权管理

场景概述：企业 A 和 企业 B 代表不同的企业。企业 A 购买了多种云资源（如 ECS 实例/ RDS 实例/ SLB 实例/ OSS 存储空间/...）来开展业务。

需求说明	解决方案
企业 A 希望能专注于业务系统，而将云资源运维监控管理等任务委托或授权给企业 B。	云账号 A 在 RAM 中创建一个角色，给角色授予合适的权限，并允许云账号 B 使用该角色。
企业 B 可以进一步将代运维任务分配给 B 的员工。B 可以精细控制其员工对 A 的云资源操作权限。	如果云账号 B 下的某个员工（RAM 用户）需要使用该角色，那么云账号 B 可以自主进行授权控制。代运维操作时，账号 B 下的 RAM 用户将使用被授予的角色身份来操作账号 A 的资源。
如果 A 和 B 的这种代运维合同终止，A 随时可以撤销对 B 的授权。	如果账号 A 与账号 B 的合作终止，A 只需要撤销账号 B 对该角色的使用。一旦账号 B 对角色的使用权限被撤销，那么 B 下的所有 RAM 用户对该角色的使用权限将被自动撤销。

临时授权移动 app 客户端直接操作您所控制的资源

场景概述：企业 A 开发了一款移动 app，并购买了 OSS 服务。移动 app 需要上传数据到 OSS（或从 OSS 下载数据）；由于移动 app 运行在用户自己的终端设备上，这些设备并不受 A 的控制。出于安全考虑，A 不能将访问密钥保存到移动 app 中。

需求说明	解决方案
企业 A 不希望所有 app 都通过 appServer 来进行数据中转，而希望让 app 能直连 OSS 上传/下载	- 云账号 A 在 RAM 中创建一个角色，给角色授予合适的权限，并允许

数据。	appServer（以 RAM 用户身份运行）使用该角色。 - 当 app 需要直连 OSS 上传/下载数据时，appServer 可以使用角色，获取角色的一个临时安全令牌并传送给 app，app 就可以使用临时安全令牌直接访问 OSS API。
企业 A 希望将安全风险控制到最小，比如，每个移动 app 直连 OSS 时都必须使用最小权限的访问令牌，而且访问时效也要很短（比如30分钟）。	如果需要更精细地控制每个 app 的权限，appServer 可以在使用角色时进一步限制临时安全令牌的资源操作权限，比如，不同 app 用户只能操作不同的子目录，那么 appServer 在使用角色时就可以进行这种限制。

委托云服务操作您的云资源

场景概述：企业 A 购买了云服务器ECS，并在其中部署了一款应用程序；应用程序需要访问 A 的 OSS 存储空间。通常情况下，

云账号 A 要将其 AccessKey（AK） 保存在应用程序的配置文件中，并在定期更换 AK 时修改应用程序的配置文件。

在进行多地域一致性部署时，AK 会随镜像以及使用镜像创建的实例扩散出去；这种情况下，当 A 需要更换 AK 时，就需要逐台更新和重新部署实例与镜像。

需求说明	解决方案
- 安全性考虑，A 不希望其应用程序通过 AK 取得其 API 操作的完整权限，希望应用程序以临时凭证访问其他产品的 API。 - 操作性考虑，A 不希望在应用程序端更新 AK，也不希望在多地域维护 AK。	使用 RAM 服务角色： - 云账号 A 在 RAM 中创建一个 ECS 服务角色（只允许 ECS 实例扮演），给角色授予合适的权限（如 OSS 的只读权限），并将该服务角色关联其 ECS 实例。 - 在连接 ECS 实例后，通过访问 ECS 实例元数据获取服务角色的 STS 临时身份凭证；ECS 中的应用程序使用该临身份凭证访问 OSS。

注意：其他场景如授权 EMR 操作客户的 ECS，函数计算 FC 操作客户的 OSS，媒体转码 MTS 操作用户的 OSS 数据等需要跨产品相互调用的场景，都可使用 RAM 服务角色授权操作。参考 创建服务角色 查看 RAM 提供的所有服务角色类型及场景。

RAM 角色类型

RAM 支持以下两种类型的角色：

用户角色： 允许 RAM 用户所扮演的角色。扮演角色的 RAM 用户可以属于自己云账号，也可以是属于其他云账号。用户角色主要用来解决 **跨账号访问** 和 **临时授权** 问题。

服务角色: 允许云服务所扮演的角色。服务角色主要用于 **授权云服务代理** 您进行资源操作。

创建 RAM 角色

通过 RAM 控制台来创建 RAM 角色包含以下步骤：

1. 选择角色类型
2. 选择受信的演员身份
3. 填写角色名称
4. 给角色绑定授权策略

创建用户角色

操作步骤如下：

登录到 RAM 控制台。

在左侧导航栏单击 **角色管理**。

单击右上角 **新建角色**。

在选择角色类型子页，单击 **用户角色**。

在填写类型信息子页，选择 **受信云账号**，如下图所示：

若创建的角色是给您自己名下的 RAM 用户使用（比如授权移动 app 客户端直接操作 OSS 资源），请选择 **当前云账号** 为受信云账号。

若创建的角色是给其他云账号名下的 RAM 用户使用（比如跨账号的资源授权），请选择 **其他云账号**，并在受信云账号 ID 中填写其他云账号的 ID。

在配置角色基本信息子页，输入 **角色名称** 和 **备注** 后，单击 **创建**。

创建成功。成功创建角色后，角色没有任何权限，单击 **授权** 可直接为该角色授权（**编辑授权策略**），授权方法请参考 **授权**。

至此，您已完成用户角色的创建。

返回 RAM 控制台，在 **角色管理** 页面找到新创建的角色（可使用角色名进行模糊查询），单击其 **角色名称** 或其对应操作列下的 **管理**，可以查看相应的角色详情，如下图所示：

创建服务角色

操作步骤如下：

登录到 RAM 控制台。

在左侧导航栏单击 **角色管理**。

单击右上角 **新建角色**。

在选择角色类型子页，单击 **服务角色**。可用的服务角色包括：

MTS 多媒体转码服务，用于将 OSS Bucket 设置为 MTS 任务的数据源时，创建以 MTS 为受信服务的角色，并使用 MTS 服务扮演该角色访问 OSS 中的数据。

OAS 归档存储服务，用于将 OSS Bucket 设置为归档存储服务的数据源时，创建以归档存储为受信服务的角色，并使用归档存储服务扮演该角色访问 OSS 中的数据

LOG 日志服务，用于将日志服务收集的日志导入 OSS 时，创建以日志服务为受信服务的角色，并使用日志服务扮演该角色将数据写入 OSS。

ApiGateway API 网关服务，用于将函数服务设置为 API 网关的后端服务时，创建以 API 网关服务为受信服务的角色，并使用 API 网关扮演该角色调用函数服务。

ECS 云服务器，用于授权 ECS 服务访问您在其他云服务中的云资源。

在填写类型信息子页，选择 **受信服务**。

在配置角色基本信息子页，输入 **角色名称** 和 **备注** 后，单击 **创建**。

创建成功。成功创建角色后，角色没有任何权限，单击 **授权** 可直接为该角色授权（**编辑授权策略**），授权方法请参考 **授权**。

至此，您已完成服务角色的创建。

返回 RAM 控制台，在 **角色管理** 页面找到新创建的角色（可使用角色名进行模糊查询），单击其 **角色名称** 或其对应操作列下的 **管理**，可以查看相应的角色详情。

使用 RAM 角色

RAM 角色只能通过 RAM 用户身份来扮演（AssumeRole）使用，不允许受信云账号以自己身份扮演角色。因此，受信云账号必须通过创建一个 RAM 用户账号，并授予该 RAM 用户账号的 AssumeRole 权限，然后以 RAM 用户身份去扮演角色。

操作步骤：

1. 创建一个 RAM 用户，并为该用户创建 AccessKey 或设置登录密码。
2. 给该 RAM 用户授权，授权时添加系统授权策略：AliyunSTSAssumeRoleAccess。

操作控制台

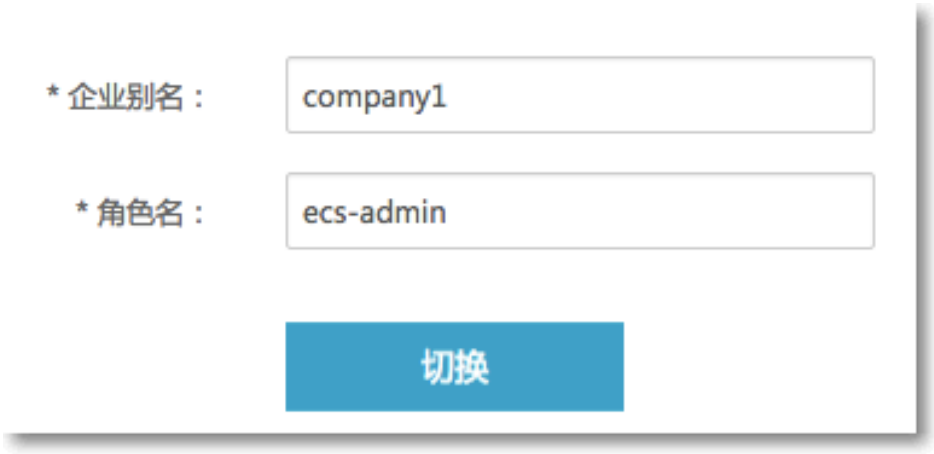
使用角色身份进行控制台操作的步骤如下：

RAM 用户登录控制台。

在右上角账号菜单下，选择 **切换身份**。

例如，company2（企业别名）下的 RAM 用户 zhangsan 登录控制台之后，控制台右上角会显示该用户的身份信息，如下图所示：

单击 **切换身份**，进入 **角色切换** 的页面，选择相应的 **企业别名** 和 **角色名**（假设当前用户已被授权允许扮演 company1（企业别名）下的 ecs-admin 角色），单击 **切换**。



切换成功后，将以角色身份访问控制台。此时控制台右上角将显示角色身份（即当前身份）和登录身份。

在扮演角色身份时，选择 **返回登录身份** 可以切换回登录身份。

访问云服务 API

当 RAM 用户被授予 AssumeRole 权限之后，可以使用其 AccessKey 调用安全令牌服务(STS) 的 AssumeRole 接口，以获取某个角色的临时安全令牌。关于 AssumeRole API 的调用方法，请参考 STS API文档。

授权管理

阿里云使用权限来描述内部身份（如用户、用户组、角色）对具体资源的访问能力。权限指在某种条件下 **允许 (Allow)** 或 **拒绝 (Deny)** 对某些资源执行某些操作。权限的载体是授权策略，授权策略是一组访问权限的集合。

本文梳理了阿里云权限与授权策略的相关属性，帮助您正确理解和使用它们。

权限

主账户（资源 Owner）控制所有权限

- 每个资源有且仅有一个属主（资源 Owner）。该属主必须是云账户，是对资源付费的人，对资源拥有完全控制权限。
- 资源属主不一定是资源创建者。比如，一个 RAM 用户被授予创建资源的权限，该用户创建

的资源归属于主账户，该用户是资源创建者但不是资源属主。

RAM 用户（操作员）默认无任何权限

- RAM 用户代表的是操作员，其所有操作都需被显式授权。
- 新建 RAM 用户默认没有任何操作权限，只有在被授权之后，才能通过控制台和 API 操作资源。

资源创建者（RAM 用户）不会自动拥有对所创建资源的任何权限

- 如果 RAM 用户被授予创建资源的权限，用户将可以创建资源。
- 但是 RAM 用户不会自动拥有对所创建资源的任何权限，除非资源 Owner 对他有显式的授权。

授权策略

授权策略（Policy）是用 **访问策略语言** 所描述的一组权限，它可以精确地描述被授权的资源集、操作集以及授权条件。当授权策略中既有 Allow 又有 Deny 的授权语句时，遵循 **Deny 优先** 的原则。

在 RAM 中，访问策略是一种资源实体，用户可以创建、更新、删除和查看访问策略。RAM 支持以下两种授权策略：

系统访问策略：由阿里云创建和管理的一组常用的权限集，比如对 ECS 的只读权限、对 ECS 的完全权限等；用户只能使用而不能修改。

自定义访问策略：由用户自己创建和管理的权限集，是对系统访问策略的扩展和补充。

系统访问策略所描述的权限粒度较粗，如果用户需要更精细的授权描述，比如精确控制对某个 ECS 实例的权限或添加授权条件限制，则需要用户创建自定义授权策略。

给 RAM 用户授权

给 RAM 用户授权，指给用户、用户组或角色绑定一个或多个授权策略。

绑定的授权策略可以是系统授权策略也可以是自定义授权策略。

如果绑定的授权策略被更新，更新后的授权策略自动生效，无需重新绑定授权策略。

授权策略是一组权限的集合，它以阿里云定义的 **授权策略语言** 来描述。通过给用户或群组附加授权策略，用户或群组中的所有用户就能获得授权策略中指定的访问权限。

RAM 支持两种类型的授权策略：**系统授权策略** 和 **自定义授权策略**。本文系统介绍了授权策略的管理方法，具

体包括：查看 系统授权策略，以及 创建、修改 和 删除 自定义授权策略。

系统授权策略

系统授权策略是阿里云提供的一组通用授权策略，主要针对不同产品的 **只读权限** 或 **所有权限**。对于阿里云提供的这组授权策略，

- 用户只能用于授权，而不能编辑和修改。
- 阿里云会自动进行更新或修改。

查看系统授权策略

如果要查看阿里云支持的所有系统授权策略，请登录到 RAM 控制台，并进入 **授权策略管理** 页面，在 **系统授权策略** 子页下，通过系统授权策略列表查看或搜索。

自定义授权策略

由于系统授权策略的授权粒度比较粗，如果这种粗粒度授权策略不能满足您的需要，那么您可以创建自定义授权策略。比如，您想控制对某个具体的 ECS 实例的操作权限，或者您要求访问者的资源操作请求必须来自于指定的 IP 地址，您必须使用自定义授权策略才能满足这种细粒度要求。

应用场景

如果您有更细粒度的授权需求，比如授权用户 bob 只能对 oss://sample_bucket/bob/ 下的所有对象执行只读操作，而且限制 IP 来源必须为您的公司网络 (通过搜索引擎查询 “我的IP” 可以获知您的公司网络 IP 地址)，那么您可以通过创建自定义授权策略来进行访问控制。

创建自定义授权策略

在创建自定义授权策略时，您需要了解授权策略语言的基本结构和语法，相关内容的详细描述请参考 授权策略语言描述。

操作步骤

在了解授权策略语言之后，您通过 RAM 控制台可以很方便地创建满足上述需求的自定义授权策略。

登录到 RAM 控制台。

单击 **策略管理** > **自定义授权策略**。

单击 **新建授权策略**，打开新建授权策略弹窗，如下图所示：

选择一个模板（这里选择 AliyunOSSReadOnlyAccess），我们可以基于该模板进行 Policy 编辑，如下图所示：

我们修改了自定义的授权策略名称，备注和策略内容。上图策略内容中的选中部分是我们新增的细粒度授权限制内容。

代码样例如下所示：

```
{
"Version": "1",
"Statement": [
{
"Action": [
"oss:Get*",
"oss:List*"
```

```
],
"Effect": "Allow",
"Resource": "acs:oss:*:*:samplebucket/bob/*",
"Condition": {
"IpAddress": {
"acs:SourceIp": "127.0.27.1"
}
}
}
]
}
```

单击 **新建授权策略**，完成新建自定义授权策略。

后续操作

如果将这个自定义的授权策略附加给用户 bob，那么 bob 对oss://samplebucket/bob/ 下的对象有只读操作权限，且限制条件是必须从您的公司网络（假设为 121.0.27.1）进行访问。

具体操作请参考 RAM 授权。

修改自定义授权策略

当用户的权限发生变更时，比如新增或撤销权限，您需要修改授权策略。当您修改授权策略时可能会遇到以下问题：

- 希望一段时间后，老的授权策略还能继续使用。
- 修改完成后，您发现授权策略修改错了，需要回滚。

授权策略具备 **版本管理** 机制，用于解决在使用中存在的问题：

- 您可以为一个授权策略保留多个版本。
- 如果超出限制，您需要自主删除不需要的版本。
- 对于一个存在多版本的授权策略，只有一个版本是活跃的，即默认版本。

操作步骤

登录到 RAM 控制台。

单击 **策略管理** > **自定义授权策略**。

通过 **授权策略名称**（可使用关键字查询）找到需要管理的授权策略，单击其名称或对应操作列下的 **查看**。

在左侧导航栏单击 **版本管理**。

如上图所示，在 **版本管理** 页面，您可以：

- 选择 **查看** 所有历史版本的策略内容。
- 将非默认版本策略 **设为当前** 版本（即默认版本）。
- 选择 **删除** 非默认版本策略。

删除自定义授权策略

您可以创建多个自定义授权策略，每个策略也可以维护多个版本。当您不再需要自定义授权策略时，您应该将授权策略删除。

前提

在删除某个授权策略前，应保证：

当前授权策略不存在多版本，只有一个默认版本。若该授权策略存在多个版本，您必须先删除除默认版本之外的所有版本。

当前授权策略未被引用（即附加给用户、用户组或角色）。若该授权策略已被引用，您可以：

- 在该授权策略的 **引用记录** 中 **解除授权**。
- 还可以选择在删除过程中 **强制解除关联关系**。

操作步骤

登录到 RAM 控制台。

单击 **策略管理** > **自定义授权策略**。

通过 **授权策略名称**（可使用关键字查询）找到需要删除的授权策略，单击其对应操作列下的 **删除**。

确认删除授权策略，可选择勾选 **强制解除关联关系**（该策略有被引用记录时强制删除引用关系）。

至此，您已成功删除一条自定义授权策略。

在 RAM 中，授权指将一个或多个授权策略附加到用户、用户组或角色的过程。其中，

给用户或用户组授权，用于对 **当前云账号** 下的 RAM 用户授权。

给角色授权，既用于对 **当前云账号** 下的 RAM 用户授权，也用于对 **其他云账号** 下的 RAM 用户或 **服务角色** 授权；不同的是，被授权的对象需要扮演角色以获取角色的身份与权限。

给用户或用户组授权

在对当前云账号下的 RAM 用户授权时，您可选择给具体用户授权，也可以向用户所在用户组授权。区别在于给用户组授权会应用到用户组下所有用户，便于对资源访问需求类似的用户（创建并添加到同一组别中）进行统一授权。

给用户授权

操作步骤如下：

登录到 RAM 控制台。

单击 **用户管理**。

通过 **用户名/显示名** 找到需要授权的用户（可使用模糊查询），单击其对应操作列下的 **授权** 按钮。

在 **编辑个人授权策略** 页面，

- 从左侧 **可选授权策略名称** 中找到需要授予当前用户的权限（可使用关键字查询），选中该策略，并单击右向箭头，可将该策略添加到右侧 **已选授权策略名称** 下。
- 在右侧 **已选授权策略名称** 下，选择某条策略，并单击左向箭头，可撤销该策略。

添加完授权策略后，单击 **确认**，完成授权。

至此，您已完成给用户授权。

给用户组授权

操作步骤如下：

登录到 RAM 控制台。

单击 **群组管理**。

通过 **组名称** 找到需要授权的用户组（可使用模糊查询），单击其对应操作列下的 **授权** 按钮。

在 **编辑群组授权策略** 页面，

- 从左侧 **可选授权策略名称** 中找到需要授予当前群组的权限（可使用关键字查询），选中该策略，并单击右向箭头，可将该策略添加到右侧 **已选授权策略名称** 下。
- 在右侧 **已选授权策略名称** 下，选择某条策略，并单击左向箭头，可撤销该策略。

添加完授权策略后，单击 **确认**，完成授权。

至此，您已完成给用户组授权。

给角色授权

新建角色 时，可以选择新建用户角色（包括以当前云账号、其他云账号为受信云账号）或服务角色，并需选择相应的受信云账号或云服务（即允许其使用所创建的角色来访问您的云资源）。

- 对 **当前云账号用户角色** 授权，则当前云账号下的 RAM 用户可扮演角色并访问被授权的云资源。
- 对 **其他云账号用户角色** 授权，则指定的其他云账号下的 RAM 用户可扮演角色并访问被授权的云资源。
- 对 **服务角色** 授权，则受信的云服务可扮演角色并访问被授权的云资源。

操作步骤

登录到 RAM 控制台。

单击 **角色管理**。

通过 **角色名** 找到需要授权的角色（可使用模糊查询），单击其对应操作列下的 **授权** 按钮。

在 **编辑角色授权策略** 页面，

- 从左侧 **可选授权策略名称** 中找到需要授予当前角色的权限（可使用关键字查询），选中该策略，并单击右向箭头，可将该策略添加到右侧 **已选授权策略名称** 下。
- 在右侧 **已选授权策略名称** 下，选择某条策略，并单击左向箭头，可撤销该策略。

添加完授权策略后，单击 **确认**，完成授权。

至此，您已完成给角色授权。

经授权后，RAM 用户可以通过控制台或 API 访问相关资源，也可以通过登录控制台后切换身份，或调用 AssumeRole 获取角色令牌（STS）以扮演相关角色，以角色身份操作相关资源。

RAM 用户登录控制台操作资源

RAM 用户登录需要使用独立的登录 URL（可以在 RAM 控制台 查看），登录时使用 **主账号别名**、**用户名** 和 **密码** 登录控制台后，登录成功后可以操作被授权的资源。如果用户点击了没有授权的操作，一般会报告“没有操作权限”的错误。

如果 RAM 用户被许可扮演角色，

在 RAM 用户登录控制台之后，可以通过 **切换身份** 操作将当前登录身份切换到角色身份，并使用角色身份的权限来操作资源；

通过 **返回登录身份** 操作从角色身份返回到当前登录身份，以使用原登录身份操作资源。

关于使用角色的更多内容请参考 角色。

应用程序调用云服务 API 操作资源

如果您的应用程序需要调用云服务 API，您需要为应用程序创建一个 RAM 用户账号并授予合适的权限，给 RAM 用户创建 AccessKey，应用程序使用该 AccessKey 来调用云服务 SDK 或 API。

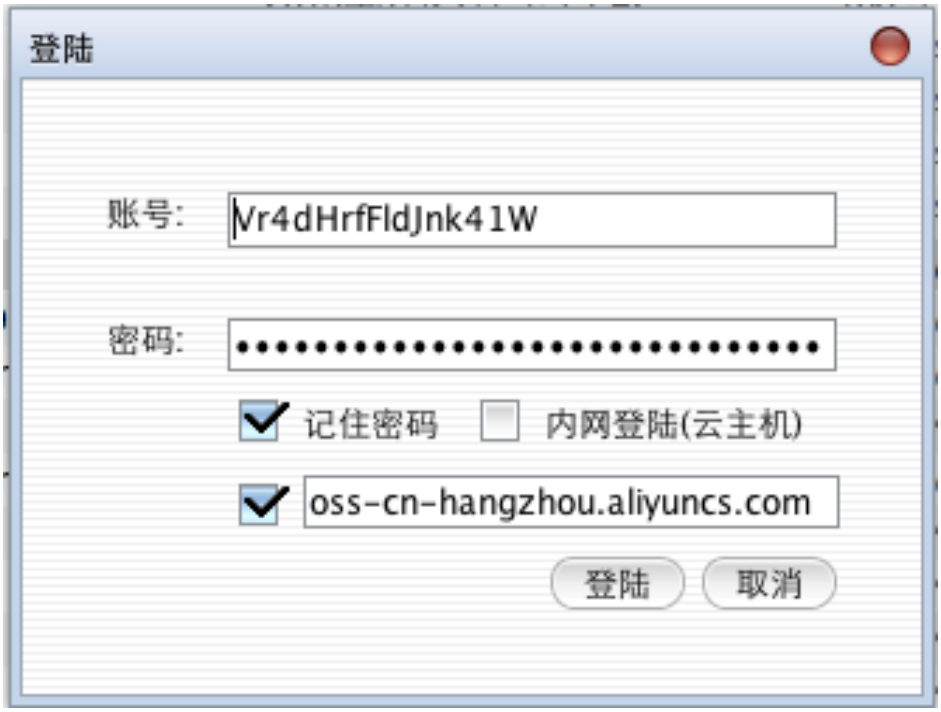
通过客户端工具操作云资源

某些云服务提供了易用的客户端工具，这些工具支持使用 RAM 用户 AccessKey 来操作云资源。

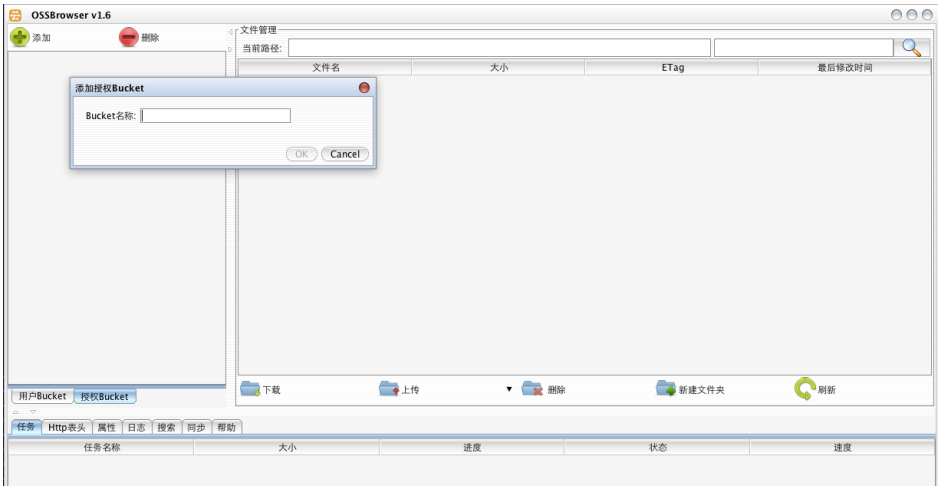
下面以 OSS 服务为例，假设 RAM 用户获得了某个 Bucket 的访问授权，那么可以使用 OSS 客户端工具 ossbrowser 来访问指定的Bucket。

操作步骤如下：

打开 ossbrowser，在账号和密码处分别输入 RAM 用户的 AccessKeyId 和 AccessKeySecret。如下图所示：



登录后进入 ossbrowser 界面，选择 **授权Bucket** 标签，点击 **添加** 即可添加一个授权 Bucket。如下图所示：



然后您就可以操作被授权 Bucket 的内容。

使用场景

本文介绍了具体场景下，使用 RAM 创建企业子账号进行分权管理的解决方案和操作步骤。

场景描述

企业 A 的某个项目（Project-X）上云，购买了多种云资源（如 ECS 实例/ RDS 实例/ SLB 实例/ OSS 存储空间/...）。项目里有多个员工需要操作这些云资源，比如有的负责购买，有的负责运维，还有的负责线上应用。由于每个员工的工作职责不一样，需要的权限也不一样。

- 出于安全或信任的考虑，A 不希望将云账号密钥直接透露给员工，而希望能给员工创建相应的用户账号。
- 用户账号只能在授权的前提下操作资源，不需要对用户账号进行独立的计量计费，所有开销都算在 A 的头上。
- 当然，A 随时可以撤销用户账号身上的权限，也可以随时删除其创建的用户账号。

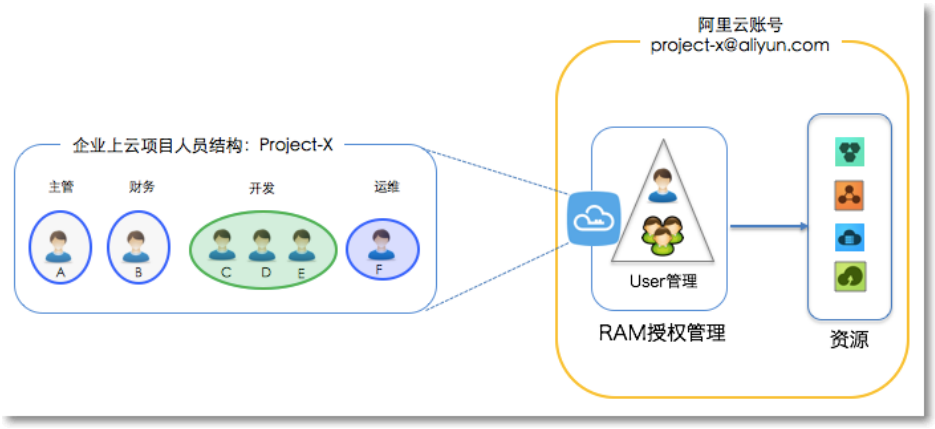
需求分析

分析以上场景，

- 杜绝多员工共享主账号，防止主账号密码或 AK 泄露导致风险不可控。
- 给不同员工分配独立的用户账号（或操作员账号）并独立分配权限，做到责权一致。
- 所有用户账号的所有操作行为可审计。
- 不需要分别核算每个操作人员的成本，所发生费用统一计入主账号账单。

解决方案

使用 RAM 的用户账号与授权管理功能，如下图所示：



操作流程如下：

给主账号绑定 MFA 设备。给主账号设置多因素认证，避免因主账号密码泄露导致风险。

开通 RAM。

创建用户账号。为不同员工（或应用系统）创建 RAM 用户账号，并按需设置登录密码或创建 AccessKey。

创建群组。如果有多个员工的职责相同，建议创建群组，并将用户添加到群组。

授权。给群组或用户添加一条或多条系统授权策略。如果需要更细粒度的授权，可以创建 **自定义授权策略**，然后给群组或用户授权。

本文介绍了具体场景下，使用 RAM 角色令牌对移动 app 客户端进行临时授权的解决方案与操作步骤。

场景描述

企业 A 开发了一款移动 app，并购买了 OSS 服务。移动 app 需要上传数据到 OSS（或从 OSS 下载数据）；移动 app 运行在用户自己的终端设备上，这些设备并不受 A 的控制。

- A 不希望所有 app 都通过 appServer 来进行数据中转，而希望让 app 能直连 OSS 上传/下载数据。
- 出于安全考虑，A 不能将访问密钥保存到移动 app 中。
- A 希望将安全风险控制到最小，比如，每个移动 app 直连 OSS 时都必须使用最小权限的访问令牌，而且访问时效也要很短（比如 30 分钟）。

需求分析

分析以上场景，

- 移动 app 需要直传数据到 OSS，不需要经过数据 proxy。
- 不能将 AK 交给移动 app，因为移动设备归属于您的用户来控制（并不可信），最佳实践是使用带过期时间的访问令牌。
- 每个移动 app 的访问权限都可以限制，支持到 OSS 对象的粒度。

解决方案

针对以上需求，**使用 RAM 角色令牌对 OSS 做临时访问授权。**

云账号 A 在 RAM 中创建一个角色，给角色授予合适的权限，并允许 appServer（以 RAM 用户身份运行）使用该角色。操作流程见 **创建角色、用户及授权。**

当 app 需要直连 OSS 上传/下载数据时，appServer 可以扮演角色（调用 STS AssumeRole），获取角色的一个临时安全令牌（STS-Token）并传送给 app，app 就可以使用临时安全令牌直接访问 OSS API。操作流程见 **获取、传递角色令牌及访问。**

appServer 可以在使用角色时进一步限制临时安全令牌的资源操作权限，以更精细地控制每个 app 的权限。操作方法见 限制 STS-Token 权限。

创建角色、用户及授权

假设云账号 A 的 AccountID 为：11223344。为 appServer 创建角色、用户，并配置权限的操作流程如下：

云账号 A 创建用户角色（假设角色名为 oss-readonly），并选择 **当前云账号** 作为受信云账号，即只允许云账号 A 下的 RAM 用户来扮演该角色。具体操作请参考 角色。

角色创建成功后，在角色详情中可以查看到该角色的基本信息：

角色的全局名称 Arn 如下：

```
acs:ram::11223344:role/oss-readonly
```

角色的信任策略（只允许 云账号 A 下的 RAM 角色来扮演角色）如下：

```
{
"Statement": [
{
"Action": "sts:AssumeRole",
"Effect": "Allow",
"Principal": {
"RAM": [
"acs:ram::11223344:root"
]
}
}
],
"Version": "1"
}
```

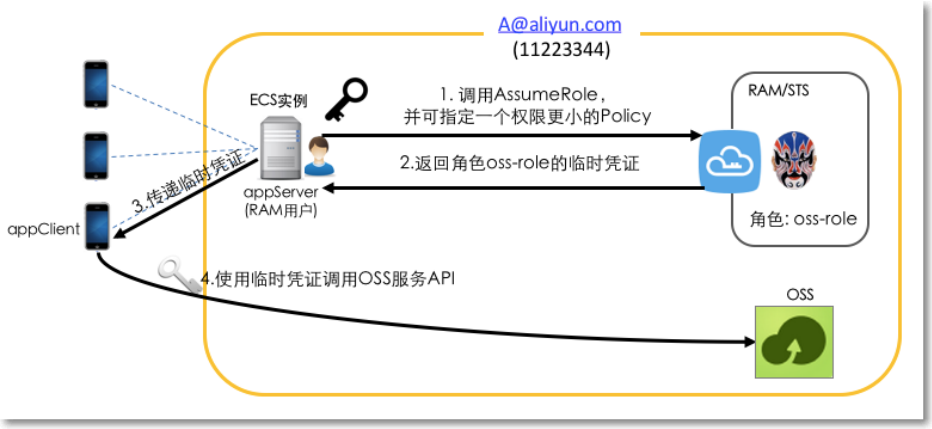
云账号 A 给角色授权，向用户角色（oss-readonly）附加只读访问 OSS 的权限（AliyunOSSReadOnlyAccess）。

云账号 A 为 appServer 创建 RAM 用户身份（假设用户名为 appserver），并为该 RAM 用户

- 创建 AccessKey，即许可 RAM 用户（appserver）调用 API。
- 附加调用 STS AssumeRole 接口的权限（AliyunSTSAssumeRoleAccess），即许可 RAM 用户（appserver）去扮演角色。

获取、传递角色令牌及访问

appClient 获取并使用角色令牌调用 OSS API 的操作示意图如下：



操作流程如下：

appServer 使用 RAM 用户（appserver）的 AccessKey 调用 STS AssumeRole。 使用 aliyuncli 来调用 AssumeRole 的命令示例如下：

注意：必须配置 appServer 的 AccessKey，而不允许是主账号 A 的 AccessKey。

```
 $ aliyuncli sts AssumeRole --RoleArn acs:ram::11223344:role/oss-readonly --RoleSessionName client-001

{
"AssumedRoleUser": {
"AssumedRoleId": "391578752573972854:client-001",
"Arn": "acs:ram::11223344:role/oss-readonly/client-001"
},
"Credentials": {
"AccessKeySecret": "93ci2umK1QKNEja6WGqi1Ba7Q2Fv9PwxZqtVF2VynUvz",
"SecurityToken":
"CAES6AIIARKAAUiwSHpkD3GXRMQk9stDr3YSVbyGqanqkS+fPlEEkjZ+dlgFnGdCI2PV93jksole8ijH8dHJr
HRA5JA1YCGsfX5hrzcNM37Vr4eVdWFVQhoCw0DXBpHv//ZcITp+ELRr4MHsnyGiErnDsXLkI7q/sbuWg6P
ACZ/jzQfEWQb/f7Y1Gh1TVFMuRjEzR2pza1hUamszOGRCWTZZeEp0WEFaayISMzkxNTc4NzUyNTczOTcy
ODU0KgpjbGllbnQtMDAxMKT+lIHBKjoGUnNhTUQ1QkoKATEaRQoFQWxsb3cSGwoMQWN0aW9uRXF1Y
WxzEgZBY3Rpb24aAwoBKhIfCg5SZXNvdXJjZUVxdWFscxIIUmVzb3VyY2UaAwoBKkoFNDMyNzRSBTI2OD
QyWg9Bc3N1bWVkUm9sZVVzZXJgAGoSMzkxNTc4NzUyNTczOTcyODU0cgllY3MtYWRtaW544Mbewo/2
6AE=",
"Expiration": "2016-01-13T15:02:37Z",
"AccessKeyId": "STS.F13GjskXTjk38dBY6YxJtXAZk"
},
"RequestId": "E1779AAB-E7AF-47D6-A9A4-53128708B6CE"
}
```

限制 STS-Token 权限

- 上述 AssumeRole 调用时没有指定 Policy 参数，意味着该 STS-Token 拥有 oss-readonly 的所有权限。

  如果需要对 STS-Token 的权限进一步限制，比如只允许访问 sample-

bucket/2015/01/01/*.jpg，那么可以通过 Policy 参数对 STS-Token 的权限进一步限制。比如，

```
$ aliyuncli sts AssumeRole --RoleArn acs:ram::11223344:role/oss-readonly --RoleSessionName client-002 --Policy "{\"Version\":\"1\", \"Statement\": [{\"Effect\":\"Allow\", \"Action\":\"oss:GetObject\", \"Resource\":\"acs:oss:*:*:sample-bucket/2015/01/01/*.jpg\"}]}"

{
"AssumedRoleUser": {
"AssumedRoleId": "391578752573972854:client-002",
"Arn": "acs:ram::11223344:role/oss-readonly/client-002"
},
"Credentials": {
"AccessKeySecret": "28Co5Vyx2XhtTqj3RJgdud4ntyzrSNdUvNygAj7xEMow",
"SecurityToken":
"CAESnQMIARKAASJgnzMzlXVyJn4KI+FsysaIpTGm8ns8Y74HVEj0pOevO8ZWXrnnkz4a4rBEPBAdFkh3197GUsprujsiU78FkszxhnQPKkQKcyvPihoXqKvuukrQ/Uoudk31KAJEz5o2EjlNUREcxWjRDRSISMzkxNTc4NzUyNTczOTcyODU0KgpjbGllbnQtMDAxMKmZxIHBKjoGUnNhTUQ1Qn8KATEaegoFQWxsb3cSJwoMQWN0aW9uRXF1YWxzEgZBY3Rpb24aDwoNb3NzOkdldE9iamVjdBJICg5SZXNvdXJjZUVxdWFscxIIUmVzb3VyY2UaLAoqYWNzOm9zczoqOio6c2FtcGxlLWJ1Y2tldC8yMDE1LzAxLzAxLyouanBnSgU0MzI3NFIFMjY4NDJaD0Fzc3VtZWRSb2xlVXNlcmAAahIzOTE1Nzg3NTI1NzM5NzI4NTRyCWVjcy1hZG1pbnjgxt7Cj/boAQ==",
"Expiration": "2016-01-13T15:03:39Z",
"AccessKeyId": "STS.FJ6EMcS1JLZgAcBJSTDG1Z4CE"
},
"RequestId": "98835D9B-86E5-4BB5-A6DF-9D3156ABA567"
}
```

此外，

- 上述 STS-Token 的默认过期时间为 3600 秒，用户还可以通过 DurationSeconds 参数来限制 STS-Token 的过期时间（最长不超过 3600 秒）。

appServer 获取并解析 Credentials。

- appServer 从 AssumeRole 返回的 Credentials 中获取 AccessKeyId、AccessKeySecret 和 SecurityToken。
- 考虑到 STS-Token 过期时间较短，如果应用业务需要一个较长的过期时间，需要 appServer 重新颁发新的 STS-Token（比如每隔 1800 秒颁发一次 STS-Token）。

appServer 将 STS-Token 安全传递给 appClient。

appClient 使用 STS-Token 直接访问云服务的 API（比如 OSS）。下面是 aliyuncli 使用 STS-Token 访问 OSS 对象的操作命令（颁发给 client-002 的 STS-Token）：

```
配置STS-Token语法：aliyuncli oss Config --host <OssEndPoint> --accessid <AccessKeyId> --accesskey <AccessKeySecret> --sts_token <SecurityToken>

$ aliyuncli oss Config --host oss.aliyuncs.com --accessid STS.FJ6EMcS1JLZgAcBJSTDG1Z4CE --accesskey 28Co5Vyx2XhtTqj3RJgdud4ntyzrSNdUvNygAj7xEMow --sts_token CAESnQMIARKAASJgnzMzlXVyJn4KI+FsysaIpTGm8ns8Y74HVEj0pOevO8ZWXrnnkz4a4rBEPBAdFkh3197
```

```
GUsprujsiU78FkszxhnQPKkQKcyvPihoXqKvuukrQ/Uoudk31KAJEz5o2EjlNUREcxWjRDRSISMzkxNTc4NzUy
NTczOTcyODU0KgpjbGllbnQtMDAxMKmZxIHBKjoGUnNhTUQ1Qn8KATEaegoFQWxsb3cSJwoMQWN0a
W9uRXF1YWxzEgZBY3Rpb24aDwoNb3NzOkdldE9iamVjdBJICg5SZXNvdXJjZUVxdWFscxIIUmVzb3VyY2U
aLAoqYWNzOm9zczoqOio6c2FtcGxlLWJ1Y2tldC8yMDE1LzAxLzAxLyouanBnSgU0MzI3NFIFMjY4NDJaD0F
zc3VtZWRSb2xlVXNlcmAAahIzOTE1Nzg3NTI1NzM5NzI4NTRyCWVjcy1hZG1pbnjgxt7Cj/boAQ==

访问OSS对象

$ aliyuncli oss Get oss://sample-bucket/2015/01/01/grass.jpg grass.jpg
```

更多参考

关于移动应用直传场景，可参考以下文档：

- 30分钟快速搭建移动应用直传服务
- 搭建应用服务器之STS Policy
- 30分钟快速搭建移动应用上传回调服务
- 进阶使用STS

本文介绍了具体场景下，使用 RAM 角色进行跨账号授权的解决方案和操作步骤。

场景描述

云账号 A 和云账号 B 分别代表不同的企业。A 购买了多种云资源（如 ECS 实例/ RDS 实例/ SLB 实例/ OSS 存储空间/...）来开展业务。

- 企业 A 希望能专注于业务系统，而将云资源运维监控管理等任务委托（或授权）给企业 B。
- 企业 B 还可以进一步将 A 的资源访问权限分配给 B 的某一个或多个员工。B 可以精细控制其员工对 A 所控制的资源的操作权限。
- 如果 A 和 B 的这种代运维合同终止，A 随时可以撤销对 B 的授权。

需求分析

分析以上场景，

- 涉及 A 和 B 两个云账号之间的授权；账号 A 是资源 Owner，希望授权账号 B 来操作。
- 账号 B 需要进一步给其子用户（代表员工或应用）授权；当 B 的员工加入或离职时，A 无需做任何权限变更。
- 如果双方业务终止，A 随时可以撤销对 B 的授权。

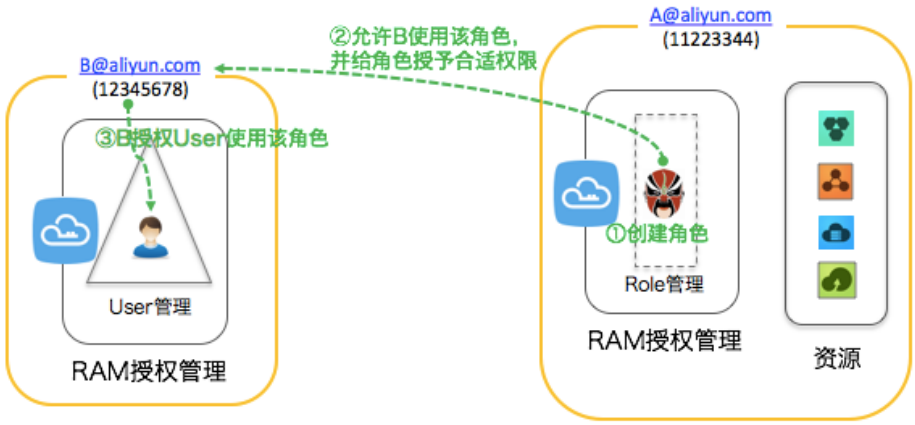
解决方案

针对以上需求，**使用 RAM 角色做跨账号授权及资源访问。**

- 云账号 A 在 RAM 中创建一个角色，给角色授予合适的权限，并允许云账号 B 使用该角色。操作流程见 跨账号授权。
- 如果云账号 B 下的某个员工（RAM 用户）需要使用该角色，那么云账号 B 可以自主进行授权控制。代运维操作时，账号 B 下的 RAM 用户将使用被授予的角色身份来操作账号 A 的资源。操作流程见 跨账号资源访问。
- 如果账号 A 与账号 B 的合作终止，A 只需要撤销账号 B 对该角色的使用。一旦账号 B 对角色的使用权限被撤销，那么 B 下的所有 RAM 用户对该角色的使用权限将被自动撤销。操作方法见 撤销跨账号授权。

跨账号授权

使用 RAM 角色进行跨账号授权的操作示意图如下。其中，假设企业 A（AccountID=11223344，别名 company-a）需要授权企业 B（AccountID=12345678，别名 company-b）的员工对其 ECS 进行操作。



操作步骤如下：

云账号 A 创建用户角色（假设角色名为 ecs-admin），并选择 **其他云账号**（云账号 B：12345678）作为受信云账号，即允许云账号 B 下的 RAM 用户来扮演该角色。具体操作请参考 角色。

角色创建成功后，在角色详情中可以查看到该角色的基本信息：

角色的全局名称 Arn 如下：

```
acs:ram::11223344:role/ecs-admin
```

角色的信任策略（只允许企业 B 来扮演角色）如下：

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

```
"Action": "sts:AssumeRole",
"Effect": "Allow",
"Principal": {
"RAM": [
"acs:ram::12345678:root"
]
}
}
],
"Version": "1"
}
```

云账号 A 给角色授权，向用户角色 ecs-admin 附加管理云服务器 ECS 的权限（AliyunECSFullAccess）。

云账号 B 为其员工 创建 RAM 用户（假设用户名为 zhangsan），并为该 RAM 用户

设置登录密码（假设登录密码为 123456），即许可 RAM 用户 zhangsan 登录控制台。

附加调用 STS AssumeRole 接口的权限（AliyunSTSAssumeRoleAccess），即许可 RAM 用户 zhangsan 去扮演/切换角色。

跨账号资源访问

云账户 B 的 RAM 用户 zhangsan 通过控制台访问 云账号 A 的 ECS 资源。操作步骤如下：

云账号 B 的 RAM 用户（zhangsan）登录控制台。

子用户登录时需正确输入 **企业别名**（company-b）、**子用户名称**（zhangsan） 和 **子用户密码**（123456）。

云账号 B 的 RAM 用户（zhangsan）切换身份。

将鼠标悬置在控制台右上角登录名上，在用户登录信息浮窗中单击 **切换身份**，进入身份切换页面；输入正确的 **企业别名**（company-a）和 **角色名**（ecs-admin），进行角色 **切换**。

云账号 B 的 RAM 用户（zhangsan）操作云账号 A 下的 ECS 资源。

撤销跨账号授权

云账号 A 撤销 云账号 B 对角色 ecs-admin 的使用，操作方法如下：

云账号 A 登录 RAM 控制台，在 角色管理 页面找到 角色 ecs-admin，单击其角色名称或其操作列

下的 **管理**，进入 **角色详情** 页面。

单击右上角 **编辑基本信息**，在角色的 **策略内容** 中，删除 acs:ram::12345678:root 行（即将云账号 B 从角色的受信云账号中撤销）。

注意：也可以选择让云账号 A 在 **角色管理** 页面 **删除** 角色 ecs-admin；但在删除角色前，角色不能有任何授权策略。

RAM操作记录

RAM 已经与 操作审计（ActionTrail）服务进行了集成，您可以在 ActionTrail 中查看所有用户（主账号/RAM 用户）对您的资源实例所进行的运维管控类操作记录。

ActionTrail 记录的 RAM 信息包括：

主账号/RAM 用户登录。详情请参考 ConsoleSignin 操作事件格式样例。

RAM 控制台操作。详情请参考 RAM 操作事件格式样例的“RAM 用户通过控制台操作 RAM”。

RAM/STS 的所有创建、变更、删除类 API 调用。详情请参考 RAM 操作事件格式样例的“RAM 用户通过 SDK 操作 RAM”。

关于操作记录的详细信息，请参考 ActionTrail 操作日志的结构。

附录1: 授权策略语言

RAM 中使用授权策略（Policy）来描述授权的具体内容，授权内容包含以下基本因素：效力（Effect）、资源（Resource）、对资源所授予的操作权限（Action）以及限制条件（Condition）。

效力（Effect）

授权效力包括两种：允许（Allow）和拒绝（Deny）。

资源（Resource）

资源是指被授权的具体对象。

比如，访问策略“允许张三对资源 SampleBucket 执行 GetBucket 操作”中的资源是“SampleBucket”。

操作权限（Action）

操作方法是指对具体资源的操作。

比如，访问策略“允许张三对资源 SampleBucket 执行 GetBucket 操作”中的操作是“GetBucket”。

限制条件（Condition）

限制条件是指授权生效的限制条件。

比如，访问策略“允许张三在2011年12月31日之前对资源 SampleBucket 执行 GetBucket 操作”中的限制条件是“在2011年12月31日之前”。

授权策略样例

下面是一个权限策略实例，它描述的含义：允许对 OSS 的 samplebucket 进行只读操作，条件是请求者的 IP 来源为 42.160.1.0。

```
{
"Version": "1",
"Statement":
[{
"Effect": "Allow",
"Action": ["oss:List*", "oss:Get*"],
"Resource": ["acs:oss:*:*:samplebucket", "acs:oss:*:*:samplebucket/*"],
"Condition":
{
"IpAddress":
{
"acs:SourceIp": "42.160.1.0"
}
}
}]
}
```

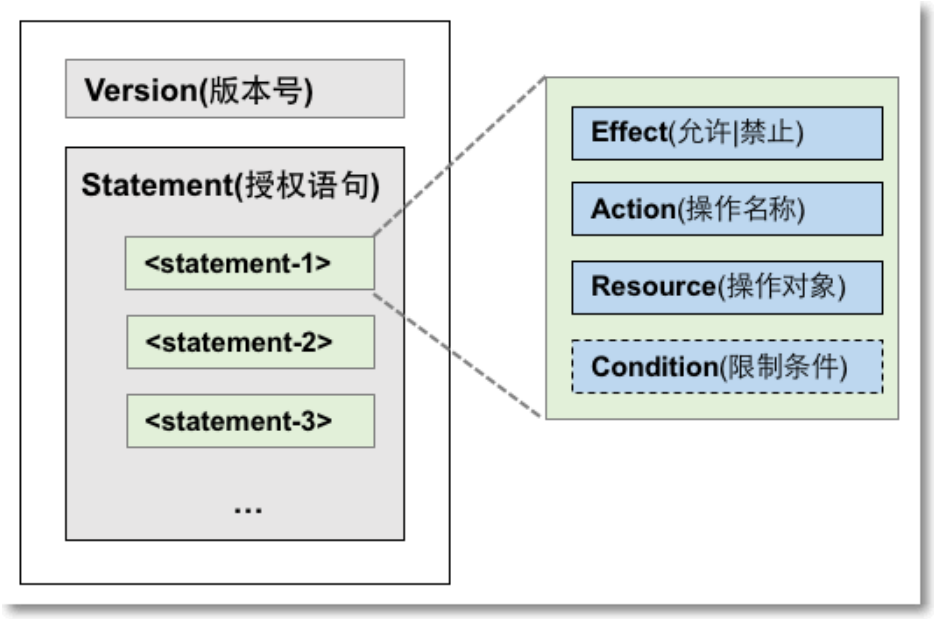
本文介绍 RAM 中授权策略（Policy）的语法结构和规则，帮助您正确理解和使用。日常应用中，您可在本文中快速查阅以下内容：Policy 结构，Policy 语法，Policy 元素使用，以及 Policy 样例。

Policy 结构

授权策略（Policy）结构包括 Policy 版本号及授权语句（Statement）列表。每个授权语句又包括以下元素：Effect（授权类型）、Action（操作名称列表）、Resource（操作对象列表）以及 Condition（条件限制）

，其中 Condition 是可选项。

Policy 结构简述如下：



格式检查（JSON）

RAM 仅支持 JSON 格式的描述。当创建或更新 Policy 时，RAM 会首先检查 JSON 格式的正确性。

- 关于 JSON 的语法标准请参考 RFC 7159。
- 用户也可以使用一些在线的 JSON 格式验证器和编辑器来校验 JSON 文本的有效性。

Policy 语法

了解 Policy 中用到的字符及规则，以及 Policy 语法描述。

字符及规则

Policy 中所包含的 JSON 字符有：{ } [ ] " , : ；描述语法使用的特殊字符有：= < > ( ) |。

字符使用说明如下：

- 当一个元素允许多值时，使用逗号和省略号来表达，比如：[<action_string>, <action_string>, ...]。在所有支持多值的语法中，使用单值也是有效的。而且两种表达方式是等效的："Action": [<action_string>] 和 "Action": <action_string>
- 带有问号的元素表示这是一个可选元素，比如：<condition_block?>
- 多值之间用竖线（|）隔开，表示取值只能选取这些值中的某一个。比如：("Allow" | "Deny")
- 使用双引号引起了的元素，表示它是文本串。比如： <version_block> = "Version" : ("1")

语法描述及说明

Policy 语法描述如下：

```
policy  = {
<version_block>,
<statement_block>
}

<version_block> = "Version" : ("1")

<statement_block> = "Statement" : [ <statement>, <statement>, ... ]

<statement> = {
<effect_block>,
<action_block>,
<resource_block>,
<condition_block?>
}

<effect_block> = "Effect" : ("Allow" | "Deny")

<action_block> = ("Action" | "NotAction") :
("*" | [<action_string>, <action_string>, ...])

<resource_block> = ("Resource" | "NotResource") :
("*" | [<resource_string>, <resource_string>, ...])

<condition_block> = "Condition" : <condition_map>
<condition_map> = {
<condition_type_string> : {
<condition_key_string> : <condition_value_list>,
<condition_key_string> : <condition_value_list>,
...
},
<condition_type_string> : {
<condition_key_string> : <condition_value_list>,
<condition_key_string> : <condition_value_list>,
...
}, ...
}
<condition_value_list> = [<condition_value>, <condition_value>, ...]
<condition_value> = ("String" | "Number" | "Boolean")
```

语法说明如下：

- **版本**：当前支持的 Policy 版本为 1。
- **授权语句**：一个 Policy 可以有多条授权语句。
  - 每条授权语句要么是 Deny，要么是 Allow。一条授权语句中，Action 是一个支持多个操作的列表，Resource 也是一个支持多个对象的列表。
  - 每条授权语句都支持独立的限制条件（Condition）。一个条件块可以支持多种条件操作类型，以及对这多种条件的逻辑组合。

- **Deny 优先**： 一个用户可以被授予多个 Policy，当这些 Policy 存在多条授权语句既包含有 Allow 又包含有 Deny 时，遵循 Deny 优先（只认 Deny 不认 Allow）原则。

元素取值：

- 当取值为数字（Number）或布尔值（Boolean）时，与字符串类似，需要用双引号引起。

  当元素取值为字符串值（String）时，支持（*）和（ ？）模糊匹配。

  - (*) 代表 0 个或多个任意的英文字母。
  - (?) 代表 1 个任意的英文字母。

  比如，ecs:Describe* 可以表示 ecs 的所有以 Describe 开头的 API 操作名称。

Policy 元素使用

了解 Policy 语法中各元素的使用规则。

Effect（授权类型）

Effect 取值 为 Allow 或 Deny。比如，"Effect": "Allow"

Action（操作名称列表）

Action支持多值，取值为云服务所定义的API操作名称，其格式定义如下：

```
<service-name>:<action-name>
```

格式说明：

- service-name: 阿里云产品名称，如 ecs, rds, slb, oss, ots 等。
- action-name: service 相关的 api 操作接口名称。

描述样例：

```
"Action": ["oss:ListBuckets", "ecs:Describe*", "rds:Describe*"]
```

Resource（操作对象列表）

Resource 通常指操作对象，比如 ECS 虚拟机实例，OSS 存储对象。我们使用如下格式来命名阿里云服务的资源命名。

```
acs:<service-name>:<region>:<account-id>:<relative-id>
```

格式说明：

- acs: Aliyun Cloud Service 的首字母缩写，表示阿里云的公有云平台。
- service-name: 阿里云提供的 Open Service 的名字，如 ecs, oss, ots 等。
- region: 地区信息。如果不支持该项，可以使用通配符“*”号来代替。
- account-id: 账号 ID，比如 1234567890123456，也可以用“*”代替。
- relative-id: 与 service 相关的资源描述部分，其语义由具体 service 指定。这部分的格式描述支持类似于一个文件路径的树状结构。以 oss 为例，relative-id = “mybucket/dir1/object1.jpg” 表示一个 OSS 对象。

描述样例：

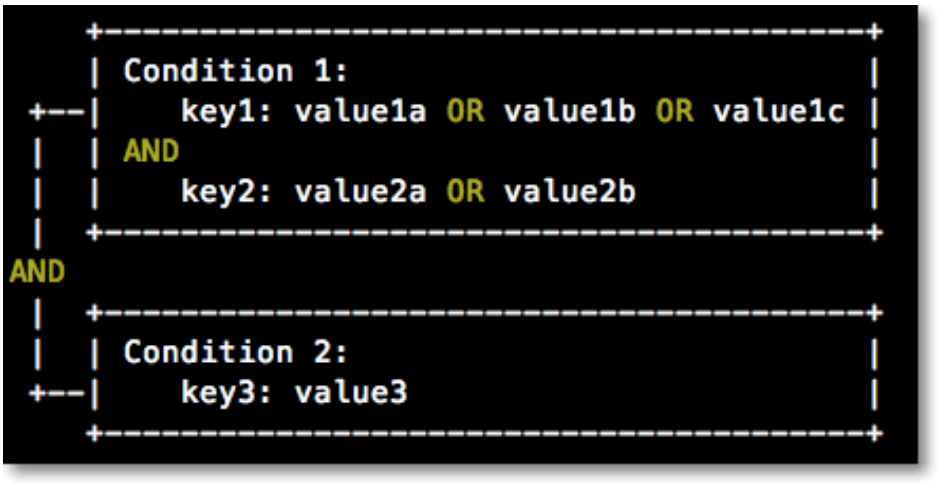
```
"Resource": ["acs:ecs:*:*:instance/inst-001", "acs:ecs:*:*:instance/inst-002", "acs:oss:*:*:mybucket",
"acs:oss:*:*:mybucket/*"]
```

Condition（条件限制）

条件块（Condition Block）由一个或多个条件子句构成。一个条件子句由条件操作类型、条件关键字和条件值组成。条件操作类型和条件关键字在下文中会有详细描述。

条件块判断逻辑

是否满足条件的判断原则如下图所示：



具体规则如下：

- 一个条件关键字可以指定一个或多个值，在条件检查时，如果条件关键字的值与指定值中的某一个相等，即可判定条件满足。
- 同一种条件操作类型的条件子句下的多个条件关键字同时满足的情况下，才能判定该条件子句满足。
- 条件块下的所有条件子句同时满足的情况下，才能判定该条件块满足。

条件操作类型

支持如下条件操作类型：字符串类型（String）、数字类型（Numeric）、日期类型（Data and time）、布尔类型（Boolean）和 IP 地址类型（IP address）。

每种条件操作类型分别支持如下的方法：

String	Numeric	Date and time	Boolean	IP address
StringEquals	NumericEquals	DateEquals	Bool	IpAddress
StringNotEquals	NumericEquals	DateNotEquals	-	NotIpAddress
StringEqualsIgnoreCase	NumericLessThan	DateLessThan	-	-
StringNotEqualsIgnoreCase	NumericLessThanEquals	DateLessThanEquals	-	-
StringLike	NumericGreaterThan	DateGreaterThan	-	-
StringNotLike	NumericGreaterThanEquals	DateGreaterThanEquals	-	-

条件关键字（Condition-key）

阿里云保留的条件关键字命名格式为：

```
acs:<condition-key>
```

阿里云保留了如下通用条件关键字：

ACS 保留条件关键字	类型	说明
acs:CurrentTime	Date and time	Web Server 接收到请求的时间，以 ISO 8601 格式表示，如 2012-11-11T23:59:59Z
acs:SecureTransport	Boolean	发送请求是否使用了安全信道，如 HTTPS
acs:SourceIp	IP address	发送请求时的客户端 IP 地址
acs:MFAPresent	Boolean	用户登录时是否使用了多因素认证（二步认证）

部分产品定义了产品级别的条件关键字，格式如下：

```
<service-name>:<condition-key>
```

不同产品定义的条件关键字，请参见各产品的用户文档。

Policy 样例

如下所示的 Policy 样例中，包含两条授权语句（Statement）：

- 第 1 条授权语句是允许对 region 华东 1（杭州）所有 ecs 资源有查看权限(ecs:Describe*)；
- 第 2 条授权语句是允许对 oss 的 mybucket 存储桶中的对象具有读访问权限(oss:ListObjects, oss:GetObject)，并限制请求者的 IP 来源必须是42.120.88.10或42.120.66.0/24。

```
{
"Version": "1",
"Statement": [
{
"Effect": "Allow",
"Action": "ecs:Describe*",
"Resource": "acs:ecs:cn-hangzhou:*:*"
},
{
"Effect": "Allow",
"Action": [
"oss:ListObjects",
"oss:GetObject"
],
"Resource": [
"acs:oss:*:*:mybucket",
"acs:oss:*:*:mybucket/*"
],
"Condition":{
"IpAddress": {
"acs:SourceIp": ["42.120.88.10", "42.120.66.0/24"]
}
}
}
]
}
```

本文介绍了在 RAM 中使用不同身份访问资源时的权限检查模型及规则，帮助您理解授权策略。

基本模型

在 RAM 中访问资源分为以主账号身份、以 RAM 用户身份和扮演 RAM 角色身份三种情形；针对每种情形，系统的授权判断条件如下表所示。

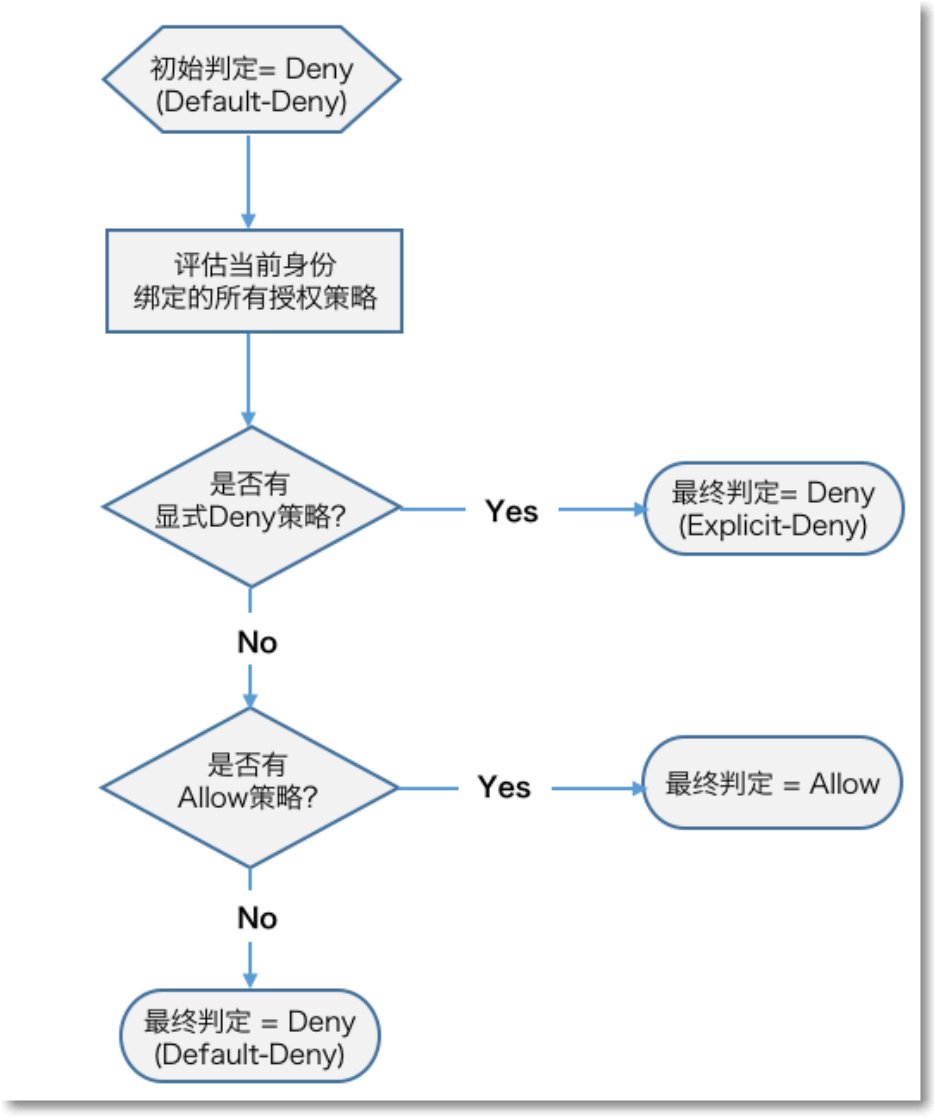
访问类型	允许访问条件（同时满足）
主账号身份访问资源	主账号是资源 Owner。 **例外**：少数云产品（如 SLS）直接支持对跨云账号 ACL 授权，如果通过 ACL 授权检查，则允许访问。
RAM 用户身份访问资源	- RAM 用户所属的主账号对资源有访问权

	限。 - 主账号对 RAM 用户有显式的 Allow 授权策略。
RAM 角色身份访问资源	- RAM 角色所属的主账号对资源有访问权限。 - 主账号对 RAM 角色有显式的 Allow 授权策略。 - 主账号对 RAM 角色有安全访问令牌（STS-Token）显式的授权。

RAM 用户身份的授权策略检查逻辑

RAM 用户访问资源时，默认没有任何权限，除非有进行显式的授权（给 RAM 用户绑定授权策略）。授权策略语句支持 Allow（允许）和 Deny（禁止）两种授权类型，当多个授权语句对一个资源操作分别出现 Allow 和 Deny 授权时，遵循 **Deny 优先** 原则。

授权策略检查逻辑如下图所示：



RAM 用户访问资源时，权限检查逻辑如下：

按照 RAM 用户身份所绑定的授权策略是否有授权：

- 如果是 Deny，则拒绝访问。
- 否则进入下一步检查。

检查 RAM 角色所属的主账号是否有访问权限：

- 如果是资源 Owner，则允许访问。
- 否则查看该资源是否有支持跨账号 ACL 许可：
  - 有则允许访问。
  - 否则拒绝访问。

RAM 角色身份的授权策略检查逻辑

RAM角色（使用角色访问令牌）访问资源时，权限检查逻辑如下：

如果当前访问令牌有指定授权策略（调用 AssumeRole 时所传入的授权策略参数），则按照上述授权策略检查逻辑进行判断：

- 如果是 Deny，则拒绝访问。
- 否则进入下一步检查。

如果当前访问令牌没有指定授权策，则直接进入下一步检查。

检查 RAM 角色身份所绑定的授权策略是否有授权：

- 如果是 Deny，则拒绝访问。
- 否则进入下一步检查。

检查 RAM 角色所属的主账号是否有访问权限：

- 如果是资源 Owner，则允许访问。
- 否则查看该资源是否有支持跨账号 ACL 许可：
  - 有则允许访问。
  - 否则拒绝访问。

附录2: Google Authenticator安装及使用指导

谷歌身份认证器是一种支持TOTP(RFC 6238)协议的动态口令生成，今天已经被广泛应用于多因素身份认证场景。

请选择您的操作系统类型

- 苹果iOS系统
- 安卓系统

注意事项

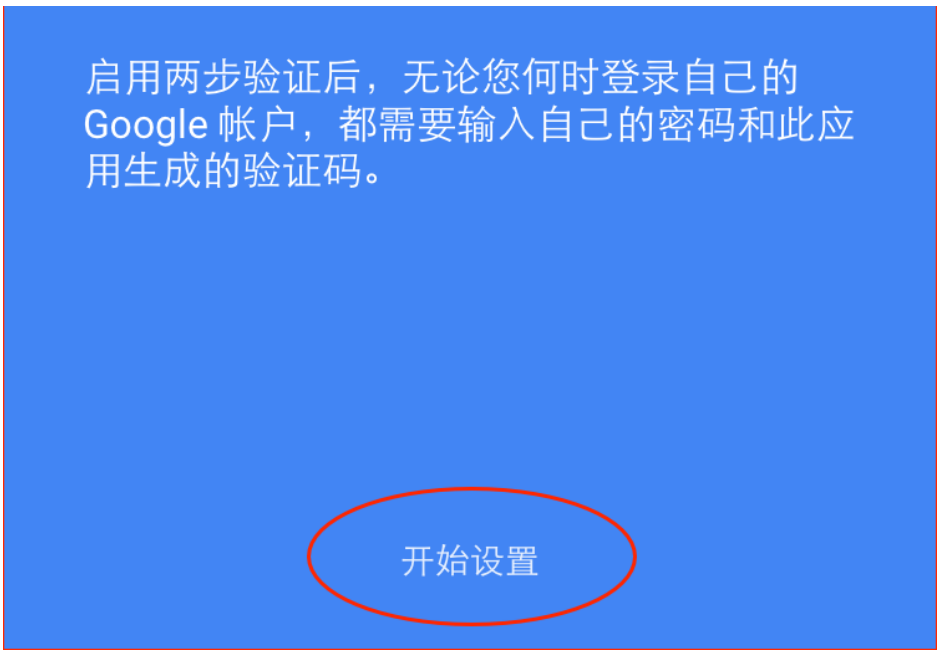
因为MFA客户端需要根据时间信息来生成密钥，请确保您手机上时间信息准确。

安装

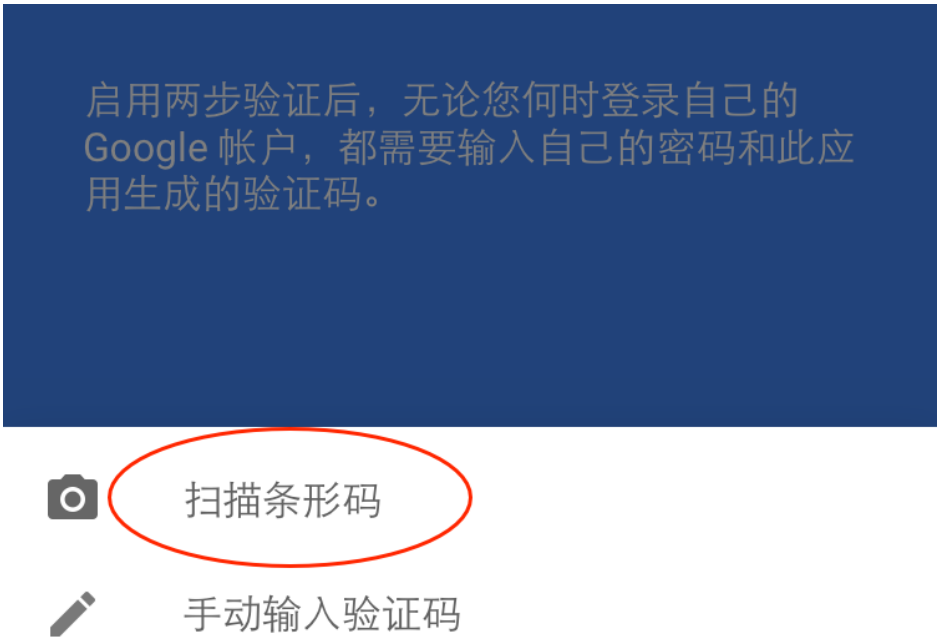
您可以进入苹果应用市场(App Store)搜索”Google Authenticator”进行安装，也可以扫描下面的二维码进行安装。

配置

打开Google Authenticator后，点击最下面的”开始设置”按钮



选择“扫描条形码”，然后会弹出条形码扫描窗，扫描MFA绑定页上生成的二维码



扫描成功后，您可以看到如下界面，包括您的帐号名和MFA密钥

在MFA页输入连续的两组MFA Code，然后点击“确认启用”来完成绑定

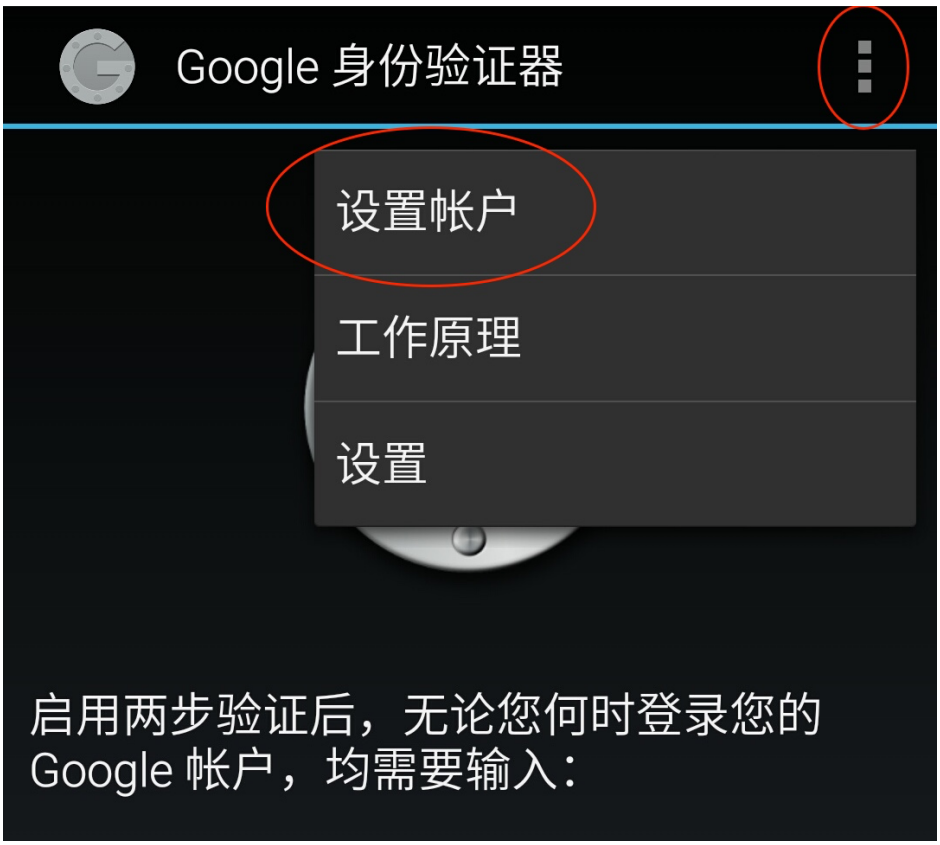
安装

您可以在您常用的应用市场搜索“身份验证器”进行安装。因为安卓版Google Authenticator还依赖外部二维码扫描组件，所以您还需要在应用市场中搜索安装“条码扫描器”。

您也可以扫描下面的二维码来安装以上软件。

配置

打开Google Authenticator后，点击右上角的菜单，选择“设置账户”选项



选择“扫描条形码”，然后会弹出条形码扫描窗口，请扫描MFA绑定页上生成的二维码

扫描成功后，您可以看到如下界面，包括您的帐号名和MFA密钥

在MFA页输入连续的两组MFA Code，然后点击“确认启用”来完成绑定


请输入您从MFA应用程序中获取的连续两组安全码:
* 第一组安全码:
855194
* 第二组安全码:
198992
确定启用