Alibaba Cloud
Resource Access Management

ベストプラクティス

Document Version20190805

目次

1 クラウドベースのサービスのセキュリティを維持するための RAM の使用

本ページでは、RAM を使用してクラウドベースのリソースにアクセスとセキュリティの設定を適用し、きめ細かいアクセス制御でアクセス権限をより適切に管理できるようにする方法について説明します。

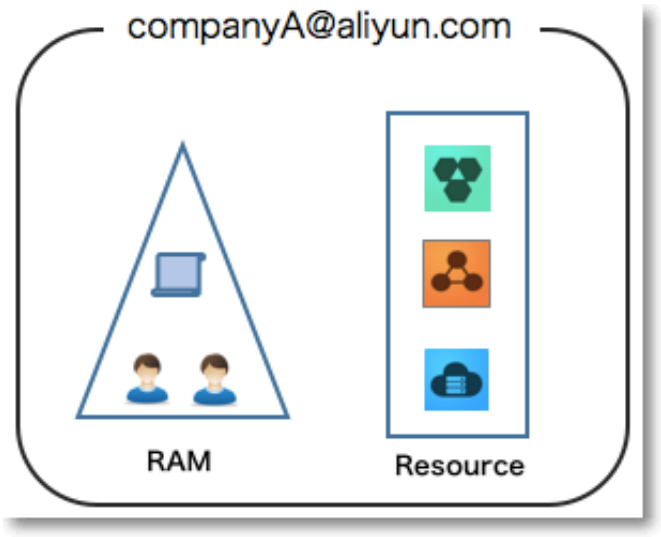
シナリオ

ビジネスリソースをクラウドに移行すると、従来の組織構造および以前のリソース管理方法では、要件を満たさなくなる可能性があります。 その結果、サービスの移行により、次のようにセキュリティ管理上の問題が発生する可能性があります。

- RAM ユーザーの責任は明確ではありません。
- セキュリティ上のリスクがあるため、アカウントの所有者はアカウント AccessKey を RAM ユーザーと共有したくありません。
- RAM ユーザーはさまざまな方法でリソースにアクセスできますが、これは統一されておらず、誤ってセキュリティ上のリスクを引き起こす可能性があります。
- ユーザーがこれらの権限を必要としなくなった場合は、RAM ユーザーのリソースアクセス権限を頻繁に呼び出す必要があります。

解決法

上記の問題を解決するには、RAM を使用して RAM ユーザーを作成し、それらにリソースアクセス許可を適用します。 具体的には、RAM を使用してアカウント AccessKey を RAM ユーザーから分離し、必要に応じて最小限のアクセス権限をユーザーに付与して、リソースのセキュリティを確保することができます。



セキュリティ管理ソリューション

- 独立した RAM ユーザーの作成

  企業に必要なアカウントは 1 つだけです (つまり、Alibaba Cloud アカウント)。 ベストプラクティスとして、このアカウントを日常業務に使用しないでください。 ただし、そのアカウントの異なるユーザーに対して複数の RAM ユーザーを作成し、必要に応じてリソースへの必要なアクセス許可を付与することができます。

  詳しくは、RAM ユーザーの作成 をご参照ください。

- API ユーザーからコンソールユーザーを分離

  RAM ユーザーに対して、コンソール操作用のログインパスワードと API 操作用の AccessKey を同時に作成しないことを推奨します。

  - アプリケーションがオープン API を介してのみクラウドリソースにアクセスできるようにするには、そのアプリケーション用の AccessKey を作成するだけで済みます。
  - 社員は、コンソールでクラウドリソースを操作する必要がある場合、その社員にログインパスワードを設定するだけです。

  詳しくは、RAM ユーザーの作成 をご参照ください。

- RAM ユーザーとグループを作成

  アカウントに複数の RAM ユーザーがある場合は、同じ責任を持つ RAM ユーザーをグループ化し、必要に応じてそのグループに権限を付与することができます。

  詳しくは、（オプション）RAM ユーザグループの作成 をご参照ください。

- 異なる RAM ユーザーグループに最小限の権限を付与

  必要に応じて、適切なシステムポリシーを RAM ユーザーまたはユーザーグループに添付できます。 きめ細かい権限管理用のカスタムポリシーを作成することもできます。 このように、さまざまな RAM ユーザーおよびユーザーグループに最小限のアクセス権限を付与することで、クラウドリソースに対する RAM ユーザーの操作をより適切に管理できます。

  詳しくは、#unique_5 をご参照ください。
- 強力なパスワードポリシーの設定

  最小長、必須文字、および検証期間に関するカスタム規則を使用して、RAM コンソールで RAM ユーザーのパスワードポリシーを設定できます。 RAM ユーザーが自分のログインパスワードを変更することを許可されている場合、ユーザーは強力なログインパスワードを作成し、定期的にパスワードまたは AccessKey を変更する必要があります。

  詳しくは、RAM 構成の初期設定 をご参照ください。
- アカウントの MFA を有効化

  アカウントに対して多要素認証 (MFA) を有効にして、アカウントのセキュリティを強化することができます。 MFA が有効になると、システムは RAM ユーザーに Alibaba Cloud にログオンして次の 2 つのセキュリティ要素を入力するように要求します。

  - 最初のセキュリティ要素: アカウント名とパスワード
  - 第 2 のセキュリティ要素: 仮想 MFA デバイスからの可変検証コード

  詳しくは、（オプション）MFA の設定 をご参照ください。
- RAM ユーザーに対して SSO を有効化

  シングルサインオン (SSO) を有効にすると、企業のすべての内部アカウントが認証されます。 その後、ユーザーは内部アカウントを使用することによってのみ Alibaba Cloud にログインして対応するリソースにアクセスできます。

  詳しくは、#unique_7 をご参照ください。
- アカウントの AccessKey を共有しないこと

  アカウントはその下のリソースに対するフルコントロールのアクセス権限を持ち、その AccessKey はログインパスワードと同じアクセス権限を持ちます。 ただし、AccessKey はプログラムへのアクセスに使用され、ログインパスワードはコンソールへのログインに使用されます。 そのため、AccessKey の誤用による情報漏えいを防ぐため、アカウントの

AccessKey を共有または使用しないことを推奨します。 代わりに、RAM ユーザーを作成し、このユーザーに適切な権限を付与してください。

詳細については、「AccessKey の管理」をご参照ください。

- セキュリティを強化するために動作条件を指定

  クラウドリソースを使用する前に RAM ユーザーが満たす必要がある動作条件を指定できます。 たとえば、RAM ユーザーがセキュアチャネル (SSL など) を使用するか、指定された送信元 IP アドレスを使用するか、または指定された期間内に操作する必要があることを指定できます。

  詳しくは、ポリシー要素 をご参照ください。

- クラウドリソースの権限の管理

  デフォルトでは、すべてのリソースはアカウントの下にあります。 RAM ユーザーはリソースを使用できますが、リソースを所有しません。 これにより、RAM ユーザーが作成したインスタンスやデータを簡単に管理できます。

  - 不要になった既存の RAM ユーザーについては、RAM ユーザーアカウントを削除するだけで、対応するすべてのアクセス権限を削除できます。
  - 権限を必要とする RAM ユーザーの場合は、まず RAM ユーザーを作成し、それにログインパスワードまたは AccessKey を設定してから、必要に応じて RAM ユーザーに適切な権限を付与する必要があります。

  詳しくは、RAM ユーザーへの権限付与 をご参照ください。

- STS を使用した RAM ユーザーへの一時的な権限の付与

  セキュリティトークンサービス (STS) は、RAM の拡張認証サービスです。 STS を使用して RAM ユーザーに一時的な権限を付与し、必要に応じてトークンの権限と自動有効期限を指定できます。

  詳しくは、#unique_11 をご参照ください。

結果

サービスをクラウドに移行した後は、前述のソリューションを使用してクラウドベースのリソースを効果的に管理し、アカウントとすべてのビジネス資産を安全に保つことができます。

次のステップ

RAM を使用して、O&M 要件を分類し、必要に応じてさまざまなエンジニアにタスクを割り当てることができます。 詳しくは、RAM を使用した多様な O&M エンジニアの権限の管理 をご参照ください。

2 RAM を使用した多様な O&M エンジニアの権限の管理

RAM を使用してさまざまな O&M エンジニアの権限を付与および管理して、多様な O&M 要件を満たしながら、管理と制御を容易にすることができます。

シナリオ

あなたの会社は、いくつかの Alibaba Cloud プロダクトを購入し、クラウド上に多数のアプリケーションシステムを展開しています。それは、より多くの O&M 要件をもたらします。

- さまざまな O&M 所有者がさまざまな Alibaba Cloud プロダクトを担当しています。
- クラウドリソースへのアクセス、操作および管理に、複数の O&M エンジニアが多様な権限を必要とします。

解決法

管理が簡単になるように、O&M 要件をプロダクト別に分類できます。 具体的には、次の図に示すように、O&M 所有者を設定し、さまざまな O&M エンジニアをさまざまなカテゴリの要件に割り当て、指定されたポリシーをこれらのエンジニアに添付することができます。

図 2-1 : O&M オーナー

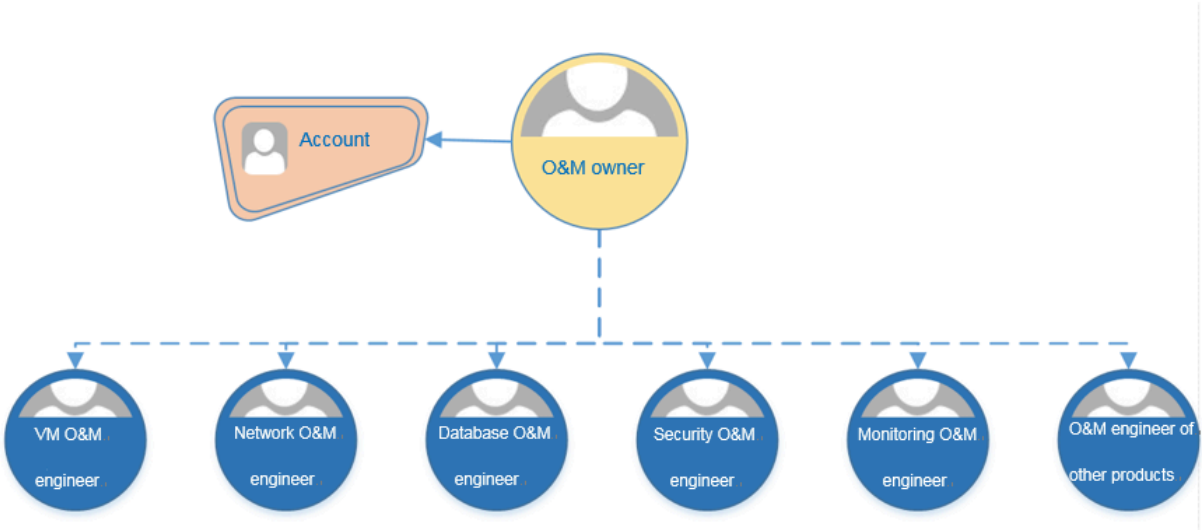


表 2-1 : ポリシー

O&M オーナー	ポリシー	説明
O&M オーナー	AdministratorAccess	このポリシーは、O&M の所有者に Alibaba Cloud のすべてのリソースを管理する権限を付与します。
VM O&M エンジニア	AliyunECSFullAccess	このポリシーは、VM O&M エンジニアに Elastic Compute Service (ECS) を管理する権限を付与します。
	AliyunESSFullAccess	このポリシーは、VM O&M エンジニアにElastic Scaling Service (ESS) を管理する権限を付与します。
	AliyunSLBFullAccess	このポリシーは、VM O&M エンジニアにServer Load Balancer (SLB) を管理する権限を付与します。
	AliyunNASFullAccess	このポリシーは、VM O&M エンジニアに Network Attached Storage (NAS) を管理する権限を付与します。
	AliyunOSSFullAccess	このポリシーは、VM O&M エンジニアに Object Storage Service (OSS) の管理権限を付与します。
	AliyunOTSFullAccess	このポリシーは、VM O&M エンジニアにTable Store (OTS) を管理する権限を付与します。
ネットワーク O&M エンジニア	AliyunCDNFullAccess	このポリシーは、ネットワーク O&M エンジニアに Content Delivery Network (CDN) を管理する権限を付与します。
	AliyunCENFullAccess	このポリシーは、ネットワーク O&M エンジニアに Cloud Enterprise Network (CEN) を管理する権限を付与します。
	AliyunCommonBandwidthPackageFullAccess	このポリシーでは、ネットワーク O&M エンジニアにInternet Shared Bandwidth を管理する権限を付与します。

O&M オーナー	ポリシー	説明
	AliyunEIPFullAccess	このポリシーは、ネットワーク O&M エンジニアに Elastic IP (EIP) を管理する権限を付与します。
	AliyunExpressConnectFullAccess	このポリシーは、ネットワーク O&M エンジニアに ExpressConnect を管理する権限を付与します。
	AliyunNATGatewayFullAccess	このポリシーは、ネットワーク O&M エンジニアに NAT Gateway を管理する権限を付与します。
	AliyunSCDNFullAccess	このポリシーは、ネットワーク O&M エンジニアに Secure Content Delivery Network (SCDN) を管理する権限を付与します。
	AliyunSmartAccessGatewayFullAccess	このポリシーでは、ネットワーク O&M エンジニアに Smart Access Gateway の管理権限を付与します。
	AliyunVPCFullAccess	このポリシーでは、ネットワークの O&M エンジニアに Virtual Private Cloud (VPC) を管理する権限を付与します。
	AliyunVPNGatewayFullAccess	このポリシーは、ネットワーク O&M エンジニアに VPN Gateway の管理権限を付与します。
データベース O&M エンジニア	AliyunRDSFullAccess	このポリシーは、データベース O&M エンジニアにRelational Database Service (RDS) を管理する権限を付与します。
	AliyunDTSFullAccess	このポリシーはデータベース O&M エンジニアにData Transmission Service (DTS) を管理する許可を与えます。
セキュリティ O&M エンジニア	AliyunYundunFullAccess	このポリシーは、セキュリティ O&M エンジニアに Alibaba Cloud Security の管理権限を付与します。
モニタリング O&M エンジニア	AliyunActionTrailFullAccess	このポリシーでは、モニタリング O&M エンジニアに ActionTrail を管理する権限を付与します。

O&M オーナー	ポリシー	説明
	AliyunARMSFullAccess	このポリシーは、モニタリング O&M エンジニアにApplication Real-Time Monitoring Service (ARMS) の管理権限を付与します。
	AliyunCloudMonitorFullAccess	このポリシーは、モニタリング O&M エンジニアに Cloud Monitor を管理する権限を付与します。
	(オプション) ReadOnlyAccess	(オプション) このポリシーは、モニタリング O&M エンジニアに、すべての Alibaba Cloud リソースに対する読み取り専用権限を付与します。
	AliyunSupportFullAccess	このポリシーは、モニタリング O&M エンジニアに Alibaba Cloud サポートシステムを管理する権限を付与します。

例

この例では、RAM ユーザーを `alice @ secloud . onaliyun . com` をデータベースの O&M 所有者として設定して、ユーザーが RDS と DTS を管理できるようにする方法を説明します。

1. RAM コンソールにログインします。
2. RAM ユーザーの作成 ユーザーに `alice @ secloud . onaliyun . com` という名前をつけます。
3. 作成した RAM ユーザーを見つけて、[権限の追加] をクリックします。
4. ポリシー名列で、 `AliyunRDSF ullAccess` および `AliyunDTSF ullAccess` を選択して、 [OK] をクリックします。

> 注：
> RAM ユーザーに他の O&M アクセス権限を付与するには、前の表に記載されているポリシーをご参照ください。

3 ユーザー管理とアクセス制御

本ページでは、Alibaba Cloud RAM を使用してユーザーの権限とリソースを管理する方法を説明するシナリオ例を示します。

シナリオ

Enterprise A が、Project-X 用に ECS インスタンス、RDS インスタンス、SLB インスタンス、OSS バケットなど、数種類の Alibaba Cloud リソースを購入したとします。このプロジェクトでは、複数の従業員がこれらのクラウドリソースに対して操作を実行する必要があります。具体的には、それぞれの従業員が異なる操作を実行するための多様な権限が必要となります。

要件分析

- アカウントパスワードまたは AccessKey の漏洩を防ぐため、従業員は Alibaba Cloud アカウントを共有しません。
- さまざまな従業員用に独立した RAM ユーザーを作成し、RAM ユーザーには独立した権限を付与します。
- すべての RAM ユーザーのすべての操作が監査可能です。
- 料金は各 RAM ユーザーにではなく、RAM ユーザーが属する Alibaba Cloud アカウントに課金されます。

ソリューション

図 3-1 : ソリューション

1. Alibaba Cloud アカウントのパスワードの漏洩に関するリスクを回避するため、多要素認証 (MFA) を設定します。 詳細は、（オプション）MFA の設定をご参照ください。
2. さまざまな従業員 (またはアプリケーション) 用に RAM ユーザーを作成し、ログインパスワードを設定するか、AccessKeys を作成します。 詳細は、RAM ユーザーの作成をご参照ください。
3. 複数の RAM ユーザーに同じ権限が必要な場合は、ユーザーグループを作成し、対象ユーザーをこのユーザーグループに追加することを推奨します。 詳細は、（オプション）RAM ユーザグループの作成をご参照ください。
4. グループまたはユーザーに 1 つまたはそれ以上のシステムポリシーを添付します。 詳細は、#unique_15をご参照ください。 きめ細かい権限管理では、1 つまたはそれ以上のカスタマイズポリシーを作成し、それらを個々のユーザーまたはユーザーグループに添付できま

す。 詳細は、#unique_16/unique_16_Connect_42_section_qpw_vwf_xdbをご参照ください。

4 モバイルアプリへの一時的な権限の付与

本ドキュメントでは、RAM ロールの STS トークンを使用してモバイルアプリに一時的な権限を付与する方法について説明します。

シナリオ

エンタープライズ A は、ユーザー自身のデバイス上で動作するモバイルアプリを開発しました。モバイルアプリを開発したエンタープライズ A は、Alibaba Cloud OSS を使用して、モバイルアプリで OSS にデータをアップロードし、OSS からデータをダウンロードできるようにする必要があります。

エンタープライズ A の要件は以下のとおりです。

- エンタープライズ A は、アプリによる appServer を使用してデータを伝送するのではなく、データを直接 OSS にアップロードし OSS からダウンロードするようにする必要があります。
- アカウントのセキュリティを維持するために、アプリを実行するモバイルデバイスはエンタープライズ A によって直接管理されていないため、エンタープライズ A は AccessKey をモバイルアプリに保存しません。
- エンタープライズ A は、アプリが OSS への接続に使用できる一時アクセス認証情報（STS トークンを使用）をアプリに付与し、アクセス期間を指定期間に制限することで、セキュリティ上のリスクを最小限に抑えたいと考えています。

ソリューション

- エンタープライズ A の Alibaba Cloud アカウント（アカウントA）を使用して、RAM 内にロールを作成し、そのロールに適切な権限を付与し、RAM ユーザーとしてログインしている appServer にこのロールの使用権限を付与します。

  詳細は、RAM ロールとユーザーの作成及び権限の付与をご参照ください。
- アプリが OSS に直接接続してデータをアップロードまたはダウンロードする必要がある場合は、appServer がロールを引き受け (STS AssumeRole API を呼び出す)、一時的な STS トークンを取得してアプリに転送できます。 そして、アプリは一時的な STS セキュリティトークンを使用して OSS API に直接アクセスできます。

  詳細は、ロール STS トークンの取得及び転送、OSS へのアクセスをご参照ください。
- appServer は、ロールを引き受けるときに一時 STS トークンのリソース操作権限にさらに制限をかけ、各アプリの権限をより適切に管理することができます。

  詳細は、STS トークンの権限への制限 をご参照ください。

RAM ロールとユーザーの作成及び権限の付与

アカウント A のアカウント ID が 11223344 であるとします。

1. アカウント A は RAM ロール `oss - readonly` を作成し、アカウント A の RAM ユーザーだけがこのロールを引き受けることができるように信頼済みアカウントとして 現行 Alibaba クラウドアカウントを選択します。

   詳細は#unique_21をご参照ください。

   ロールを作成後、エンタープライズ A は基本情報ページ上のロール情報を表示できます。

   - この例では、ロールの Alibaba Cloud リソース名 (ARN) は以下のとおりです。

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

   - ロール内の信頼ポリシー (アカウント A の RAM ユーザーのみ引き受け可能) は、以下のとおりです。

     ```
     {
     " Statement ": [
     {
      " Action ": " sts : AssumeRole ",
      " Effect ":" Allow ",
      " Principal ": {
        " RAM ": [
          " acs : ram :: 11223344 : root "// ロールの信頼できるエンティティタイプが Alibaba Cloud アカウントの場合、デフォルトで ' root ' が使用されます。
        ]
      }
     }
     ],
     " Version ": " 1 "
     }
     ```

2. アカウント A が、ロール oss-readonly にポリシー `AliyunOSSR eadOnlyAcc ess`（OSS 読み取り専用権限）をアタッチします。

   詳細は、#unique_15 をご参照ください。

3. アカウント A は、appServer 用に RAM ユーザー（ここでは、RAM ユーザーを Appserver と呼びます）を作成し、RAM ユーザー用の AccessKey を作成して、ユーザーが STS AssumeRole API を呼び出すことができるように `AliyunSTSA ssumeRoleA ccess` システムポリシーを添付します。

ロール STS トークンの取得及び転送、OSS へのアクセス

次の図には、ロール STS トークンを取得し、それを使用して OSS API を呼び出すアプリの操作手順を示します。

図 4-1 : 手順

appServer は、RAM ユーザー (appserver) の AK を使用して STS API ロールの引き受け（AssumeRole） を呼び出します。

> 注 :
> appServer の AK が構成されている必要があります。

次は、aliyuncli を使用して AssumeRole API を呼び出す方法の例です。

```
$ aliyuncli sts AssumeRole -- RoleArn acs : ram :: 11223344 :
 role / oss - readonly -- RoleSessio nName client - 001
 {
     " AssumedRol eUser ": {
         " AssumedRol eId ": " 3915787525 73972854 : client - 001
 ",
         " Arn ": " acs : ram :: 11223344 : role / oss - readonly /
 client - 001 "
     },
     " Credential s ": {
         " AccessKeyS ecret ": " 93ci2umK1Q KNEja6WGqi 1Ba7Q2Fv9P
 wxZqtVF2Vy nUvz ",
         " SecurityTo ken ": " CAES6AIIAR KAAUiwSHpk D3GXRMQk9s
 tDr3YSVbyG qanqkS + fPlEEkjZ + dlgFnGdCI2 PV93jksole 8ijH8dHJrH
 RA5JA1YCGs fX5hrzcNM3 7Vr4eVdWFV QhoCw0DXBp Hv // ZcITp +
 ELRr4MHsny GiErnDsXLk I7q / sbuWg6PACZ / jzQfEWQb / f7Y1Gh1TVF
 MuRjEzR2pz a1hUamszOG RCWTZZeEp0 WEFaayISMz kxNTc4NzUy
 NTczOTcyOD U0KgpjbGll bnQtMDAxMK T + lIHBKjoGUn NhTUQ1QkoK
 ATEaRQoFQW xsb3cSGwoM QWN0aW9uRX F1YWxzEgZB Y3Rpb24aAw
 oBKhIfCg5S ZXNvdXJjZU VxdWFscxII UmVzb3VyY2 UaAwoBKkoF
 NDMyNzRSBT I2ODQyWg9B c3N1bWVkUm 9sZVVzZXJg AGoSMzkxNT
 c4NzUyNTcz OTcyODU0cg llY3MtYWRt aW544Mbewo / 26AE =",
         " Expiration ": " 2016 - 01 - 13T15 : 02 : 37Z ",
         " AccessKeyI d ": " STS . F13GjskXTj k38dBY6YxJ tXAZk "
     },
     " RequestId ": " E1779AAB - E7AF - 47D6 - A9A4 - 53128708B6 CE "
 }
```

STS トークンの権限への制限

1. AssumeRole API を呼び出した後、STS トークンにきめ細かいアクセス権限を付与できます。

   AssumeRole API を呼び出すときにポリシーを指定しない場合、STS トークンには `oss - readonly` のすべての権限が付与されます。 この問題を解決するには、STS トーク

ンのアクセス権限にさらに制限をかけるポリシーを指定します。例えば、STS トークンから `sample - bucket / 2015 / 01 / 01 /*. jpg` へのアクセス権限のみを付与します。 例は次のとおりです。

```
$  aliyuncli   sts   AssumeRole  -- RoleArn   acs : ram :: 11223344
 : role / oss - readonly  -- RoleSessio  nName   client - 002  --
 Policy  "{\" Version \":\" 1 \", \" Statement \": [{\" Effect \":\"
 Allow \", \" Action \":\" oss : GetObject \", \" Resource \":\" acs
 : oss :*:*: sample - bucket / 2015 / 01 / 01 /*. jpg \"}]}"
{
   " AssumedRol  eUser ": {
       " AssumedRol  eId ": " 3915787525  73972854 : client - 002
 ",
       " Arn ": " acs : ram :: 11223344 : role / oss - readonly /
 client - 002 "
   },
   " Credential  s ": {
       " AccessKeyS  ecret ": " 28Co5Vyx2X  htTqj3RJgd  ud4ntyzrSN
 dUvNygAj7x  EMow ",
       " SecurityTo  ken ": " CAESnQMIAR  KAASJgnzMz  lXVyJn4KI
 + FsysaIpTGm  8ns8Y74HVE  j0pOevO8ZW  Xrnnkz4a4r  BEPBAdFkh3
 197GUspruj  siU78Fkszx  hnQPKkQKcy  vPihoXqKvu  ukrQ / Uoudk31KAJ
 Ez5o2EjlNU  REcxWjRDRS  ISMzkxNTc4  NzUyNTczOT  cyODU0Kgpj
 bGllbnQtMD  AxMKmZxIHB  KjoGUnNhTU  Q1Qn8KATEa  egoFQWxsb3
 cSJwoMQWN0  aW9uRXF1YW  xzEgZBY3Rp  b24aDwoNb3  NzOkdldE9i
 amVjdBJICg  5SZXNvdXJj  ZUVxdWFscx  IIUmVzb3Vy  Y2UaLAoqYW
 NzOm9zczoq  Oio6c2FtcG  xlLWJ1Y2tl  dC8yMDE1Lz  AxLzAxLyou
 anBnSgU0Mz  I3NFIFMjY4  NDJaD0Fzc3  VtZWRSb2xl  VXNlcmAAah
 IzOTE1Nzg3  NTI1NzM5Nz  I4NTRyCWVj  cy1hZG1pbn  jgxt7Cj / boAQ
 ==",
       " Expiration ": " 2016 - 01 - 13T15 : 03 : 39Z ",
       " AccessKeyI  d ": " STS . FJ6EMcS1JL  ZgAcBJSTDG  1Z4CE "
   },
   " RequestId ": " 98835D9B - 86E5 - 4BB5 - A6DF - 9D3156ABA5  67 "
}
```

注：

STS トークンのデフォルトの有効期間は 3600 秒 (最大制限) です。 DurationSeconds パラメーターを使用して、STS トークンの有効期限を制限できます。

2. appServer は資格情報を取得して解析します。
   - appServer は、AssumeRole API により返される資格情報の中から AccessKeyId、AccessKeySecret、および SecurityToken を取得します。
   - STS トークンの有効期間が決定されています。 アプリケーションがより長い有効期間を必要とする場合、appServer は 1800 秒ごとに 1 つの STS トークンを発行するなど新しい STS トークンを再発行する必要があります。
3. appServer は、STS トークンを安全にアプリに転送します。

4. アプリは STS トークンを使用して Alibaba Cloud サービス (OSS など) API に直接アクセスします。

   以下は、aliyuncli と STS トークンを使用した OSS オブジェクトへのアクセス方法の例です（ここでは、STS トークンは client-002 に発行されます）：

```
 Configure   the   STS   token   syntax :  aliyuncli   oss   Config
 -- host   -- accessid   -- accesskey   -- sts_token
$  aliyuncli   oss   Config  -- host   oss . aliyuncs . com  --
 accessid   STS . FJ6EMcS1JL  ZgAcBJSTDG  1Z4CE  -- accesskey
 28Co5Vyx2X  htTqj3RJgd  ud4ntyzrSN  dUvNygAj7x  EMow  -- sts_token
   CAESnQMIAR  KAASJgnzMz  lXVyJn4KI + FsysaIpTGm  8ns8Y74HVE
 j0pOevO8ZW  Xrnnkz4a4r  BEPBAdFkh3  197GUspruj  siU78Fkszx
 hnQPKkQKcy  vPihoXqKvu  ukrQ / Uoudk31KAJ  Ez5o2EjlNU  REcxWjRDRS
 ISMzkxNTc4  NzUyNTczOT  cyODU0Kgpj  bGllbnQtMD  AxMKmZxIHB
 KjoGUnNhTU  Q1Qn8KATEa  egoFQWxsb3  cSJwoMQWN0  aW9uRXF1YW
 xzEgZBY3Rp  b24aDwoNb3  NzOkdldE9i  amVjdBJICg  5SZXNvdXJj
 ZUVxdWFscx  IIUmVzb3Vy  Y2UaLAoqYW  NzOm9zczoq  Oio6c2FtcG
 xlLWJ1Y2tl  dC8yMDE1Lz  AxLzAxLyou  anBnSgU0Mz  I3NFIFMjY4
 NDJaD0Fzc3  VtZWRSb2xl  VXNlcmAAah  IzOTE1Nzg3  NTI1NzM5Nz
 I4NTRyCWVj  cy1hZG1pbn  jgxt7Cj / boAQ ==
 OSS  オブジェクトへのアクセス
$  aliyuncli   oss   Get   oss :// sample - bucket / 2015 / 01 / 01
 / grass . jpg   grass . jpg
```

次のステップ

詳細は、次をご参照ください。

- モバイルアプリの直接データ転送の設定
- 権限コントロール
- モバイルアプリ向けのデータコールバック
- #unique_26

5 クロスアカウントリソースの権限付与とアクセス

本ページでは、RAM ロールを使用してクロスアカウントリソースの権限付与とアクセスを実行する方法について説明します。

シナリオ

アカウント A とアカウント B は、 2 つの異なる企業 (企業 A と企業 B) を表します。 企業 A は、ビジネスをサポートするためさまざまなクラウドリソース (ECS インスタンス、RDS インスタンス、SLB インスタンス、OSS バケットなど) を購入しました。

要件分析

- アカウント A はリソースの所有者であり、アカウント A のリソースに対して操作を実行させるために、適切な権限をアカウント B に付与する必要があります。
- さらにアカウント B は、RAM ユーザー (従業員またはアプリケーション) に権限を付与する必要があります。 アカウント B の従業員が企業 B に入社または退職する時に、アカウント A は権限を変更する必要がありません。
- 企業 A または企業 B が契約を終了した場合、アカウント A はいつでもアカウント B の権限を削除することができます。

ソリューション

クロスアカウントの権限付与とリソースアクセスを実行するには、RAM ロールを使用します。

- アカウント A は RAM にロールを作成し、RAM ロールに適切な権限を付与し、アカウント B にこのロールの使用を許可します。

  詳細は、「クロスアカウントの権限付与 (Cross-account authorization) 」をご参照ください。
- アカウント B の従業員 (RAM ユーザー) がこのロールを使用する必要がある場合、アカウント B はこの RAM ユーザーにアカウント A のリソースに対する操作を実行するための権限を付与できます。

  詳細は、「クロスアカウントリソースアクセス (Cross-account resource access)」をご参照ください。

- 企業 A または企業 B が契約を終了した場合、アカウント A はアカウント B の権限を取り消すことができます。この場合、アカウント B のすべての RAM ユーザーは、このロールに関連する権限を失います。

  詳細は、「クロスアカウント権限の削除 (Removing cross-account authorization)」をご参照ください。

クロスアカウント権限付与

次の図に、RAM ロールを使用してクロスアカウント権限付与を実現する方法を示します。 この例では、企業 A (アカウント ID は11223344、アカウントエイリアスは company-a) は、企業 B の従業員 (アカウント ID は 12345678、アカウントエイリアスは company-b) に ECS 操作権限を付与する必要があります。

図 5-1 : RAM ロールを使用してクロスロール権限付与を実現

1. 企業 A は RAM ロール (ロールの名前は ecs-admin) を作成し、信頼できるエンティティとして [その他の Alibaba Cloud アカウント] (アカウント ID は 12345678) を選択します。

   詳細は、#unique_21をご参照ください。

   ロールを作成後、アカウント A は [基本情報] ページ上のロール情報を閲覧できます。

   - この例では、ロールの Alibaba Cloud リソース名 (ARN) は以下のとおりです。

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

   - ロール内の信頼ポリシー (アカウント B の RAM ユーザーのみ引き受け可能) は、以下のとおりです。

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

```
}
```

2. アカウント A は `AliyunECSF ullAccess` ポリシーをロール ecs-admin に添付します。

   詳細は、#unique_15をご参照ください。

3. アカウント B は、その従業員の RAM ユーザー (RAM ユーザーの名前は Alice) を作成し、その RAM ユーザーのログインパスワードを設定し、その RAM ユーザーに STS AssumeRole API を呼び出す `AliyunSTSA ssumeRoleA ccess` システムポリシーを添付します。

クロスアカウントリソースアクセス

アカウント B の RAM ユーザー Alice がアカウント A の ECS リソースに (Alibaba Cloud コンソールを介して) アクセスできるようにするには、以下のステップを実行します。

1. RAM コンソールにログインします。

   ログイン時に、アカウントエイリアス company-b、RAM ユーザー名 Alice、およびパスワード 123456 を入力します。

2. ポインタをアカウントアイコンの上に移動し、[ロールの切り替え] をクリックします。

   表示されたページで、[エンタープライズエイリアス/デフォルトドメイン名] に company-a、[ロール名] に ecs-admin と入力します。

> 注 :
> 上記の操作を完了すると、RAM ユーザー Alice はアカウント A の ECS リソースに対して操作を実行できるようになります。

クロスアカウント権限の削除

アカウント A がアカウント B からロール ecs-admin を使用する権限を削除する手順は以下のとおりです。

1. RAM コンソールにログインして [RAM ロール]、ecs-admin のロール名をクリックします。
2. [信頼ポリシー管理] タブをクリックして `acs : ram :: 12345678 : root` を削除します。

> 注 :
> またアカウント A は、[RAM ロール] ページ上で ecs-admin ロールを削除することでアカウント B の ecs-admin ロールを使用する権限も削除できます。 ただし、ロールにポリシーが添付されているとロールを削除することはできません。

6 クラウドアプリケーションの動的 ID および権限管理

本ページでは、Alibaba Cloud RAM を使用して、アプリケーションが RAM ロールの動的 STS トークンを取得することによって Alibaba Cloud API にアクセスできるようにする方法について説明します。

シナリオ

ある企業が ECS インスタンスを購入し、そのアプリケーションを ECS にデプロイする必要があります。 AccessKeys を使用してアプリケーションが他の Alibaba Cloud API にアクセスできるようにするには、企業は以下のいずれかの方法を使用できます。

- AccessKeys をコードに埋め込みます。
- AccessKeys をアプリケーションの設定ファイルに保存します。

ただし、上記の方法を使用すると以下の問題が発生します。

- AccessKey の開示 : AccessKey がプレーンテキストで ECS インスタンスに埋め込まれていると、スナップショットの共有、または共有イメージインスタンスを作成するイメージが原因で誤って別のユーザーに公開される可能性があります。
- O&M の複雑化 : AccessKey が変更 (AccessKey のローテーションまたはユーザー ID の変更により) されると、AccessKey が ECS インスタンスに存在するため、すべてのインスタンスとイメージを更新して再デプロイする必要があります。 その結果、インスタンスとイメージの管理が非常に複雑になります。

ソリューション

上記の問題を解決するため、ECSを RAM のアクセス制御機能と組み合わせることができます。具体的には、管理者は各 ECS インスタンス (アプリケーションの動作環境) の RAM ロールを作成し、各 RAM ロールに適切な権限を付与します。 アプリケーションは、対応する RAM ロールの動的 STS トークンを使用して他の Alibaba Cloud API を呼び出すことができます。

図 6-1 : プロセス

ECS インスタンスの RAM ロールを設定

1. 管理者は Alibaba Cloud アカウントを使用して ECS インスタンスの RAM ロールを作成し、適切なポリシーを RAM ロールに添付します。

    注：

   ECS インスタンスの RAM ロールは、ユーザーによって作成され、許可後ユーザーの ECS インスタンスによって使用される一種の RAM サービスロールです。

2. 管理者は ECS インスタンスを起動し、RAM ロールを設定します。

   - (i) ECS は、設定された RAM ロールに従って AssumeRole API を呼び出して STS にアクセスし、RAMロールの STS トークンを取得するリクエストを送信します。
   - (ii) STS は ECS の ID と RAM ロールに添付されるポリシーを検証します。 検証が成功すると、STS トークンが発行されます。 検証に失敗すると、リクエストは拒否されます。

   詳細は、「#unique_32」または「#unique_33」をご参照ください。

3. STS トークンを取得すると、ECS インスタンスはメタデータサービスを介して STS トークンをアプリケーションに提供します。

   たとえば Linux では、STS トークンやその有効期間などのメタデータ情報の取得時に以下のコマンドを使用できます。

   ```
   $  curl   http :// 100 . 100 . 100 . 200 / latest / meta - data /
    ram / security - credential  s /< roleName >
   ```

    注：

   - STS トークンが有効期間内であればアプリケーションは Alibaba Cloud API を呼び出すことができます。 STS トークンは通常 1 時間後に期限切れになります。 ECS は、期限が切れる前に STS トークンを自動的に更新します。
   - STS トークンに対応する権限がない場合、管理者は関連するポリシーを RAM ロールに添付する必要があります。
   - RAM ロールに関連付けられているポリシーが更新された後、STS トークンに関連付けられている権限が即時有効になり、ユーザーは ECS インスタンスを再起動する必要はありません。

4. アプリケーションは STS トークンを使用して Alibaba Cloud API を呼び出します。

    注：

アプリケーションが Alibaba Cloud SDK を使用している場合、Alibaba Cloud SDK は ECS メタデータサービスから RAM ロールの STS トークンを取得でき、開発者は SDK に AccessKey 関連の機密情報を設定する必要はありません。

詳細は、「RamRole を設定して ECS インスタンスへの AccesKey 以外のアクセスを実現 (Configure RamRole to achieve non-AccesKey access to ECS instances)」をご参照ください。

管理者と一般ユーザーの権限を分ける

ほとんどのシナリオでは、管理者と一般的な ECS インスタンスユーザーの権限は、異なる RAM ユーザーとして設定されています。 次の図に、管理者と一般ユーザーの権限を分ける方法を示します。

図 6-2 : プロセス

重要 :

- ECS は、RAM ユーザー (たとえば ECS へのアクセス権しかなく RAM 権限管理者ではない RAM ユーザー) が ECS インスタンスを作成して RAM ロールを設定する前に、その RAM ユーザーに RAM ロールの `ram : PassRole` 権限があるかどうかチェックします。 権限が見つからない場合、RAM ユーザーは ECS インスタンスを作成できません。
- ECS インスタンスの RAM ロールを設定できるのは許可ユーザーだけです。 このように RAM ロールの使用は厳密に管理されているため、権限の悪用を防止できます。

管理者と一般ユーザーの権限を分けるには、ECS インスタンスの RAM ロールを設定し、一般ユーザーに `PassRole` 権限を付与 (上記の図のステップ 1.5 を参照) する必要があります。

管理者は、カスタマイズポリシーを作成して一般ユーザーに添付することもできます。 例は以下のとおりです。

重要 :

`ロール名`は RAM 名に置き換える必要があります。

```
{
   " Statement ": [
    {
      " Effect ": " Allow ",
      " Action ": " ram : PassRole ",
      " Resource ": " acs : ram :*:*: role /< rolename >"
    }
```

```
  ],
  " Version ": " 1 "
}
```

次のステップ

Alibaba Cloud RAM が権限アプリケーションのすべての要件を満たしていない場合は、アクセス制御機能を提供する Function Compute および MaxCompute など他の Alibaba Cloud サービスを使用して、クラウド上のユーザーやアプリケーションの ID と AccessKey を管理できます。

7 タグを使用したグループごとの ECS インスタンス権限付与

本ページでは、タグを使用してリソース (ECS インスタンスなど) をグループ別に権限付与し、RAM ユーザーがタグ付きリソースのみを表示および操作できるようにする方法について説明します。

シナリオ

10 個の ECS インスタンスがあります。 dev チームがそれらのうちの 5 つを管理し、ops チームが他の 5 つを管理させたいとします。 ただし、各チームには権限付与されたリソースのみを表示 (他のチームに権限付与されたリソースは表示させない) させたいとします。

準備

RAM コンソールに RAM アカウントを使ってログインできることを確認します。

解決法

2 つの RAM ユーザーグループを作成し、これら 2 つのグループにタグを付けて、グループに権限を付与します。

- 5 つに、"team" をキーとして、"dev" を値としてタグ付けます。
- 残りの 5 つに、"team" をキーとして、"ops" を値としてタグ付けます。

手順

1. ECS コンソールにログインし、[インスタンス] をクリックして、ターゲットインスタンスを選択します。 操作 列で その他 > インスタンス設定 > タグの編集 を選択します。
2. [作成] をクリックして、キーと値を入力して、[確認] をクリックします。
3. RAM コンソールにログインし、2 つの RAM ユーザーグループを作成し、"dev" と "ops" という名前を付けます。

   詳しくは、（オプション）RAM ユーザグループの作成をご参照ください。
4. RAM ユーザーを作成し、それらのユーザーを異なるユーザーグループに追加します。

   詳しくは、RAM ユーザーの作成をご参照ください。
5. 2 つのカスタムポリシーを作成し、それらを異なるユーザーグループに適用します。

   詳しくは、RAM での権限付与 をご参照ください。

    注：

ユーザグループにポリシーを適用すると、このグループの RAM ユーザは関連する権限を継承します。

この例では、dev ユーザーグループのポリシー名は policyForDevTeam です。 ポリシーコンテンツは以下のとおりです。

```
{
    " Statement ": [
    {
        " Action ": " ecs :*",
        " Effect ": " Allow ",
        " Resource ": "*",
        " Condition ": {
            " StringEqua  ls ": {
                " ecs : tag / team ": " dev "
            }
        }
    },
    {
        " Action ": " ecs : DescribeTa  g *",
        " Effect ": " Allow ",
        " Resource ": "*"
    }
    ],
    " Version ": " 1 "
}
```

上記のコードでは、

- "Condition" を持つ `" Action ": " ecs :*"` 要素は、`" team ": " dev "` としてタグ付けされたインスタンスをフィルタリングするために使用します。
- `" Action ": " ecs : DescribeTa  g *"` 要素はすべてのタグを表示するために使用します。 ユーザーがECS コンソールで操作を実行すると、システムはユーザーが選択するすべてのタグを表示し、ユーザーが選択したタグキーと値に従ってインスタンスをフィルタリングします。

 注 :
例に従ってポリシー "policyForOpsTeam" を作成し、このポリシーを ops ユーザーグループに付与できます。

権限付与インスタンスの表示

1. ECS コンソールに RAM ユーザーとしてログインします。

 注 :
ユーザーが ECS コンソールにログインした後、システムはデフォルトで ECS 概要ページに移動します。 この場合、ページに表示される ECS インスタンスの数は 0 です。 関連するインスタンスを表示するには、 [インスタンス] をクリックします。

2. [インスタンス] をクリックします。検索ボックスの横にある [タグ] をクリックします。

   > 注：
   > コンソールに表示されるリージョンが、インスタンスが属するリージョンであることを確認する必要があります。

3. タグキーの上にポインタを移動します。 タグ値リストが表示されます。 値を選択すると、対応するインスタンスがフィルタリングされます。

次のステップ

本ページでは、タグを使用してリソース (ECS インスタンスなど) をグループ別に権限付与し、RAM ユーザーがタグ付きリソースのみを表示および操作できるようにする方法について説明します。

> 注：
> タグ付けできるのはカスタム画像のみです。

8 タグを使用したグループごとの RDS インスタンスの権限付与

本ページでは、タグを使用してリソース (RDS インスタンスなど) をグループ別に権限付与し、RAM ユーザーがタグ付きリソースのみを表示および操作できるようにする方法について説明します。

シナリオ

10 個の RDS インスタンスがあります。 dev チームがそれらのうちの 5 つを管理し、ops チームが他の 5 つを管理するようにします。 ただし、各チームには権限付与されたインスタンスのみを表示するようにします (他のチームの権限付与されたリソースは表示しません)。

準備

詳しくは、タグを使用したグループごとの ECS インスタンス権限付与 をご参照ください。

以下は、RDS に関連するカスタマイズポリシーの例です。

```
{
  " Statement ": [
    {
      " Action ": " rds :*",
      " Effect ": " Allow ",
      " Resource ": "*",
      " Condition ": {
        " StringEqua  ls ": {
          " rds : ResourceTa  g / team ": " dev "
        }
      }
    },
    {
       " Action ": " rds : DescribeTa  g *",
       " Effect ": " Allow ",
       " Resource ": "*"
     }
  ],
  " Version ": " 1 "
}
```

上記のコードでは、

- "Condition" を持つ `" Action ": " rds :*"` 要素は、`" team ": " dev "` としてタグ付けされたインスタンスをフィルタリングするために使用します。 "Condition" のキーワードは `rds : ResourceTa  g` です。

- `" Action ": " rds : DescribeTa g *"` 要素は、すべてのタグを表示するために使用します。 ユーザーが RDS コンソールで操作を実行すると、システムはユーザーが選択するすべてのタグを表示し、ユーザーが選択したタグキーと値に従ってインスタンスをフィルタリングします。

次のステップ

RDS インスタンスをグループにタグ付けして権限を付与した後で RAM ユーザーの関連する権限が見つからない場合は、 #unique_40をご参照ください。

9 RAM を利用した ECS 権限管理

本ドキュメントでは、RAM にポリシーを作成して RAM ユーザーの ECS 権限を管理する方法について説明します。

共通ポリシー

次の表は、ECS 権限を管理するために RAM に作成できる共通ポリシーの一覧です。

ポリシー	説明
AliyunECSFullAccess	RAM ユーザーに ECS インスタンスの完全管理権限を付与します。
AliyunECSReadOnlyAccess	RAM ユーザーに ECS インスタンスに対する読み取り専用権限を付与します。

注：

ECS 権限の詳細は、Authorization rules をご参照ください。

RAM ユーザーへのカスタムポリシーのアタッチ

1. 本文の「ECS 権限付与の例」に従ってカスタムポリシーを作成します。

   詳細は、#unique_5 をご参照ください.
2. 対象のポリシーを見つけてクリックします。
3. リファレンス タブで、権限付与 をクリックします。
4. プリンシパルフィールドに、対象 RAM ユーザーの ID または名前を入力します。
5. OK をクリックします。

注：

必要に応じて、RAM ユーザーまたは RAM ユーザーグループにポリシーをアタッチすることもできます。 詳細は、RAM での権限付与 をご参照ください.

ECS 権限付与の例

- 例 1：複数のインスタンスを持つ RAM 管理者として、ユーザーに 2 つのインスタンスのみを操作する権限を与えます。

  これら 2 つの ECS インスタンスの ID は、i-001 と i-002 とします。

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

```
      " Action ": " ecs :*",
      " Effect ":" Allow ",
      " Resource ": [
                  " acs : ecs :*:*: instance / i - 001 ",
                  " acs : ecs :*:*: instance / i - 002 "
                  ]
    },
    {
      " Action ": " ecs : Describe *",
      " Effect ":" Allow ",
      " Resource ": "*"
    }
  ],
  " Version ": " 1 "
}
```

注：

- **権限付与された RAM ユーザーはすべての ECS インスタンスを表示できますが、操作できるのはそのうちの 2 つのみです。**
- **ポリシーには `Describe *`要素が必要です。 ポリシーに `Describe *`要素が含まれていない場合、権限付与された RAM ユーザーはコンソールにインスタンスを表示できません。 ただし、RAM ユーザーは、API の呼び出し、CLI の使用、または ECS SDK の使用によって、指定された 2 つの ECS インスタンスを操作できます。**

- **例 2：RAM 管理者として、RAM ユーザーに青島リージョンの ECS インスタンスを表示する権限を付与しますが、ディスクとスナップショットに関する情報の表示権限を付与しません。**

  リージョンおよびリソースタイプ別に ECS 権限をユーザーに付与できます。

```
{
  " Statement ": [
    {
      " Effect ": " Allow ",
      " Action ": " ecs : Describe *",
      " Resource ": " acs : ecs : cn - qingdao :*: instance /*"
    }
  ],
  " Version ": " 1 "
}
```

- **例 3：RAM 管理者として、RAM ユーザーにスナップショットの作成権限を付与します。**

  RAM ユーザーが ECS インスタンス管理者権限を付与された後にディスクスナップショットを作成できない場合は、ユーザーにディスク権限を再度付与する必要があります。 この例では、ECS インスタンス ID は inst-01、ディスク ID は dist-01 とします。

```
{
  " Statement ": [
    {
      " Action ": " ecs :*",
      " Effect ":" Allow ",
      " Resource ": [
```

```
        " acs : ecs :*:*: instance / inst - 01 "
      ]
    },
    {
      " Action ": " ecs : CreateSnap  shot ",
      " Effect ":" Allow ",
      " Resource ": [
        " acs : ecs :*:*: disk / dist - 01 ",
        " acs : ecs :*:*: snapshot /*"
      ]
    },
    {
      " Action ": [
        " ecs : Describe *"
      ],
      " Effect ": " Allow ",
      " Resource ": "*"
    }
  ],
  " Version ": " 1 "
}
```

10 OSS インスタンスの権限付与

目次

OSS 権限定義を表示する

OSS プロダクトドキュメントの「アクセス制御」をご参照ください。

RAM ユーザーに OSS 読み取り専用権限を割り当て

RAM コンソールで RAM ユーザーを作成し、システム権限付与ポリシー "AliyunSLBReadOnlyAccess" をユーザーに追加します。 権限付与ポリシーの追加方法の詳細は、「権限付与」をご参照ください。

完全な OSS 管理権限を RAM ユーザーに割り当て

システム権限付与ポリシー "AliyunSLBFullAccess" を RAM コンソールの RAM ユーザーに追加します。

RAM ユーザーにバケット内のリソースの一覧表示と読み取りを許可

RAM ユーザー (自分を代表するアプリケーションなど) を許可して、OSS SDK または OSS CMD を使用してバケット内のリソースをリストして読み取る必要がある場合、 権限付与ポリシーを作成する必要があります。次に完了するためのカスタム権限付与ポリシーを作成する必要があります。

バケットが "myphotos" という名前であるとします。 以下のように権限付与ポリシーを作成します。

```
{
    " Version ": " 1 ",
    " Statement  ":[
        {
```

```
            " Effect ": " Allow ",
            " Action ": " oss : ListObject  s ",
            " Resource ": " acs : oss :*:*: myphotos "
        },
        {
            " Effect ": " Allow ",
            " Action ": " oss : GetObject ",
            " Resource ": " acs : oss :*:*: myphotos /*"
        }
    ]
}
```

権限付与された RAM ユーザーに OSS コンソールで操作を実行させたい場合は、権限付与ポリシーに "GetBucketAcl" および "GetObjectAcl" 権限を追加してください。(コンソールは操作エクスペリエンスを最適化するために追加の OSS API を呼び出す必要があります。) 権限付与ポリシー定義の例を次に示します。 これにより、RAM ユーザーは OSS コンソールで操作を実行できます。

```
{
    " Version ": " 1 ",
    " Statement  ":[
        {
            " Effect ": " Allow ",
            " Action ": " oss : ListBucket  s ",
            " Resource ": " acs : oss :*:*:*"
        },
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : ListObject  s ",
                " oss : GetBucketA  cl "
            ],
            " Resource ": " acs : oss :*:*: myphotos "
        },
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : GetObject ",
                " oss : GetObjectA  cl "
            ],
            " Resource ": " acs : oss :*:*: myphotos /*"
        }
    ]
}
```

OSS で IP アドレス固有のアクセス制御を適用

例1: `Allow` **コマンドを使用して IP アドレス固有のアクセス制御を適用します。**

IP アドレスセグメント `42 . 120 . 88 . 0 / 24` **および** `42 . 120 . 66 . 0 / 24` **が "myphotos" ディレクトリの情報を読むことを許可されています。**

```
{
    " Version ": " 1 ",
    " Statement  ":[
        {
            " Sid ": " To   allow   listing   all   buckets ",
```

```
            " Effect ": " Allow ",
            " Action ": [
                " oss : ListBucket  s "
            ],
            " Resource ": [
                " acs : oss :*:*:*"
            ]
        },
        {
            " Sid ": " To   allow   only   the   users   in   the
 specified   IP   address   segment   to   obtain   the   informatio
 n   in   the   myphotos   directory ",
            " Effect ": " Allow ",
            " Action ": [
                " oss : ListObject  s ",
                " oss : GetObject "
            ],
            " Resource ": [
                " acs : oss :*:*: myphotos ",
                " acs : oss :*:*: myphotos /*"
            ],
            " Condition  ":{
                " IpAddress ": {
                    " acs : SourceIp ": " 42 . 120 . 88 . 0 / 24 ", "
 42 . 120 . 66 . 0 / 24 "
                }
            }
        }
    ]
}
```

例2: `Deny` **コマンドを使用して IP アドレス固有のアクセス制御を適用します。**

ユーザーの IP アドレスが `42 . 120 . 88 . 0 / 24` **セグメント内にない場合、ユーザーは OSS 操作を実行できません。 以下のように権限付与ポリシーを作成します。**

```
{
    " Version ": " 1 ",
    " Statement  ":[
        {
            " Sid ": " To   allow   listing   all   buckets ",
            " Effect ": " Allow ",
            " Action ": [
                " oss : ListBucket  s "
            ],
            " Resource ": [
                " acs : oss :*:*:*"
            ]
        },
        {
            " Sid ": " To   allow   obtaining   the   informatio  n
 in   the   myphotos   directory ",
            " Effect ": " Allow ",
            " Action ": [
                " oss : ListObject  s ",
                " oss : GetObject "
            ],
            " Resource ": [
                " acs : oss :*:*: myphotos ",
                " acs : oss :*:*: myphotos /*"
            ]
```

```
        },
        {
            " Sid ": " To   disallow   IP   addresses   not   in   the
   42 . 120 . 88 . 0 / 24   segment   to   access   OSS ",
            " Effect ": " Deny ",
            " Action ": " oss :*",
            " Resource ": [
                " acs : oss :*:*:*"
            ],
            " Condition  ":{
                " NotIpAddre  ss ": {
                    " acs : SourceIp ": [" 42 . 120 . 88 . 0 / 24 "]
                }
            }
        }
    ]
}
```

注："Deny" コマンドを使用したポリシーは、"Allow" コマンドを使用したポリシーよりも優先度が高くなります。 (ユーザーのアクセス操作が "Deny" コマンドのいずれかのポリシーを満たしている場合、そのユーザーはコンテンツにアクセスできません。) したがって、IP アドレスが 42.120.88.0/24 セグメントにないユーザが "myphotos" ディレクトリの情報にアクセスしようとすると、OSS サービスはユーザに操作権限がないことを通知します。

OSS ディレクトリーによる権限付与

ディレクトリによる許可は高度な権限付与機能です。

背景

"myphotos" という名前の写真バケットがあるとします。 バケットには、写真が撮影された場所を示すディレクトリが含まれています。 各ディレクトリには、写真が撮影された年を示すサブディレクトリが含まれています。

ディレクトリツリーは次のとおりです。

```
 myphotos [ Bucket ]
  ├──  beijing
  │    ├──  2014
  │    └──  2015
  ├──  hangzhou
  │  ├──  2013
  │  ├──  2014
  │  └──  2015  // The   read - only   permission   on   this
 directory   must   be   assigned .
  └──  qingdao
       ├──  2014
       └──  2015
```

`myphotos / hangzhou / 2015 /` ディレクトリに RAM ユーザーへの読み取り専用権限を割り当てる必要があるとします。 必要な権限付与ポリシーはアプリケーションのシナリオによって異なります。 以下では、最も単純なものからより複雑なものまで、ポリシーの複雑さによって 3 つのシナリオの権限付与ポリシーについて説明します。

シナリオ1: RAM ユーザーはすべてのファイルパスを知っており、ファイルの内容を読み取るためのアクセス許可のみを必要とし、ファイルを一覧表示するためのアクセス許可は必要としません。

このシナリオでは、RAM ユーザーはすべてのファイルの完全パスを知っており、完全パスを使用してファイルを直接読み取ることができます。 ソフトウェアがそのような権限を必要とするのは、ソフトウェアシステム内のファイルパスは特定の規則に準拠している (たとえば、ファイルは従業員 ID に基づいて名前が付けられている) か、ファイルパスはソフトウェアシステムのデータベースに保持されているためです。

```
{
    " Version ": " 1 ",
    " Statement  ":[
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : GetObject "
            ],
            " Resource ": [
                " acs : oss :*:*: myphotos / hangzhou / 2015 /*"
            ]
        }
    ]
}
```

シナリオ2: RAM ユーザーが OSS CMD を使用して `myphotos / hangzhou / 2015 /` ディレクトリにアクセスするが、そのディレクトリで使用可能なファイルはわかりません。 したがって、ファイルをリストしなければなりません。

一般に、ソフトウェア開発者はそのような許可の割り当てを必要とします。 開発者は、ディレクトリで利用可能なファイルがわかりません。 OSS CMD または API を使用して、ディレクトリ情報を直接取得します。

このシナリオでは、 `scenario 1` では必要ではなかった "ListObjects" 権限を追加する必要があります。 `myphotos / hangzhou / 2015 /` ディレクトリのファイルのみを一覧表示する必要があるので、"oss:Prefix" 条件を "ListObjects" 権限に追加する必要があります。

```
{
    " Version ": " 1 ",
    " Statement  ":[
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : GetObject "
            ],
            " Resource ": [
                " acs : oss :*:*: myphotos / hangzhou / 2015 /*"
            ]
        },
        {
            " Effect ": " Allow ",
```

```
            " Action ": [
                " oss : ListObject  s "
            ],
            " Resource ": [
                " acs : oss :*:*: myphotos "
            ],
            " Condition ":{
                " StringLike ":{
                    " oss : Prefix ":" hangzhou / 2015 /"
                }
            }
        }
    ]
}
```

シナリオ 3: RAM ユーザーが OSS コンソールを使って `myphotos / hangzhou / 2015 /` ディレクトリにアクセスします。

これは最も使いやすいシナリオです。 RAM ユーザーがビジュアル OSS クライアントを使用して `myphotos / hangzhou / 2015 /` ディレクトリにアクセスすると、Windows のファイルエクスプローラと同様に、ビジュアル OSS クライアントは RAM ユーザがルートディレクトリからサブディレクトリのレベルを通してターゲットディレクトリにアクセスすることを可能にします。

したがって、このタイプのディレクトリナビゲーションを実装するには、次の権限を追加する必要があります。

1. **すべてのバケットを一覧表示する権限**
2. **"myphotos" ディレクトリのサブディレクトリを一覧表示する権限 (この例では、サブディレクトリに "beijing"、"hangzhou" および "qingdao" が含まれます)**
3. **"myphotos/hangzhou" の下にサブディレクトリーをリストする許可 (サブディレクトリーには "2013"、"2014" および "2015" が含まれます)**

```
{
    " Version ": " 1 ",
    " Statement  ":[
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : ListBucket  s ",
                " oss : GetBucketA  cl "
            ],
            " Resource ": [
                " acs : oss :*:*:*"
            ]
        },
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : GetObject ",
                " oss : GetObjectA  cl "
            ],
            " Resource ": [
```

```
                " acs : oss :*:*: myphotos / hangzhou / 2015 /*"
            ]
        },
        {
            " Effect ": " Allow ",
            " Action ": [
                " oss : ListObject  s "
            ],
            " Resource ": [
                " acs : oss :*:*: myphotos "
            ],
            " Condition ": {
                " StringLike ": {
                    " oss : Delimiter ": "/",
                    " oss : Prefix ": [
                        "",
                        " hangzhou /",
                        " hangzhou / 2015 /*"
                    ]
                }
            }
        }
    ]
}
```

RAM ユーザーにバケットの完全管理を許可

最初に権限付与ポリシーを作成する必要があります。 バケットが "myphotos" という名前であるとします。 以下のように権限付与ポリシーを作成します。

```
{
    " Version ": " 1 ",
    " Statement  ":[
        {
            " Effect ": " Allow ",
            " Action ": " oss :*",
            " Resource ": [
                " acs : oss :*:*: myphotos ",
                " acs : oss :*:*: myphotos /*"
            ]
        }
    ]
}
```

そしてそのユーザーに権限付与ポリシーを追加します。

OSS コンソールへのログイン時に操作権限がないことを通知され、バケットの管理を許可された RAM ユーザー

以下の通り権限付与ポリシーを作成して、RAM ユーザーに ("myphotos" のような) バケットからデータオブジェクトを読み取ることを許可するとします。

```
{
  " Version ": " 1 ",
  " Statement  ":[
    {
      " Effect ": " Allow ",
      " Action ": [
```

```
        " oss : ListObject  s "
      ],
      " Resource ": " acs : oss :*:*: myphotos "
    },
    {
      " Effect ": " Allow ",
      " Action ": [
        " oss : GetObject "
      ],
      " Resource ": " acs : oss :*:*: myphotos /*"
    }
  ]
}
```

ただし、OSS コンソールにログインすると、RAM ユーザーに操作権限がないことが通知されました。

その理由は、RAM ユーザーが OSS コンソールにログインすると、OSS コンソールによって、承認されたとおりに RAM ユーザーが OSS サービスにアクセスできるようになるためです。
より良いユーザーインタラクションを体験するために、OSS コンソールは "ListBuckets"、"GetBucketAcl" および "GetObjectAcl" オペレーションも呼び出します。 ("GetBucketAcl" はバケットがプライベートかパブリックかを指定します。 "GetObjectAcl" はオブジェクトがプライベートかパブリックかを指定します。)

したがって、RAM ユーザーが OSS コンソールでバケットを管理できるようにするには、次のように承認ポリシーを作成する必要があります。

```
{
  " Version ": " 1 ",
  " Statement  ":[
    {
      " Effect ": " Allow ",
      " Action ": " oss : ListBucket  s ",
      " Resource ": " acs : oss :*:*:*"
    },
    {
      " Effect ": " Allow ",
      " Action ": [
        " oss : ListObject  s ",
        " oss : GetBucketA  cl "
      ],
      " Resource ": " acs : oss :*:*: myphotos "
    },
    {
      " Effect ": " Allow ",
      " Action ": [
        " oss : GetObject ",
        " oss : GetObjectA  cl "
      ],
      " Resource ": " acs : oss :*:*: myphotos /*"
    }
  ]
}
```

11 RAM を利用した RDS 権限管理

本ドキュメントでは、RAM にポリシーを作成して RAM ユーザーの RDS 権限を管理する方法について説明します。

共通ポリシー

次の表は、RDS 権限を管理するために RAM に作成できる共通ポリシーの一覧です。

ポリシー	説明
AliyunRDSFullAccess	RAM ユーザーに RDS インスタンスの完全管理権限を付与します。
AliyunRDSReadOnlyAccess	RAM ユーザーに RDS インスタンスに対する読み取り専用権限を付与します。


注：
RDS 権限の詳細は、RAM の権限付与 をご参照ください。

RAM ユーザーへのカスタムポリシーのアタッチ

1. 本文の「RDS 権限付与の例」に従ってカスタムポリシーを作成します。

   詳細は、#unique_5 をご参照ください.
2. 対象のポリシーを見つけてクリックします。
3. リファレンスタブで、権限の付与 をクリックします
4. プリンシパルフィールドに、対象 RAM ユーザーの ID または名前を入力します。
5. OK をクリックします。


注：
必要に応じて、RAM ユーザーまたは RAM ユーザーグループにポリシーをアタッチすることもできます。 詳細は、RAM での権限付与 をご参照ください.

RDS 権限付与の例

- 例 1：複数のインスタンスを持つ RAM 管理者として、ユーザーに 2 つのインスタンスのみを操作する権限を付与します。

  これら 2 つの RDS インスタンスの ID は、i-001 と i-002 とします。

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

```
      " Action ": " rds :*",
      " Effect ":" Allow ",
      " Resource ": [
                  " acs : rds :*:*: dbinstance / i - 001 ",
                  " acs : rds :*:*: dbinstance / i - 002 "
                  ]
    },
    {
      " Action ": " rds : Describe *",
      " Effect ":" Allow ",
      " Resource ": "*"
    }
  ],
  " Version ": " 1 "
}
```

注：

- **権限付与された RAM ユーザーはすべての RDS インスタンスを表示できますが、操作できるのはそのうちの 2 つのみです。**
- **ポリシーには `Describe *` 要素が必要です。 ポリシーに `Describe *` 要素が含まれていない場合、権限付与された RAM ユーザーはコンソールにインスタンスを表示できません。 ただし、RAM ユーザーは、API の呼び出し、CLI の使用、または RDS SDK の使用によって、指定された 2 つの RDS インスタンスを操作できます。**

- **例 2：RAM 管理者として、ユーザーに Alibaba Cloud Data Management System（DMS）のデータにアクセスする権限を付与します。**
  - **RAM ユーザーに 2 つの特定 RDS インスタンスへのアクセス権限を付与します。**

```
{
  " Statement ": [
    {
      " Action ": " dms : LoginDatab  ase ",
      " Effect ": " Allow ",
      " Resource ": " acs : rds :*:*: dbinstance / rds783a063
 9ks5k7 ****"
    }
  ],
  " Version ": " 1 "
}
```

注：

`rds783a063 9ks5k7 ****` をアクセスする RDS インスタンスの ID に置き換える必要があります。

  - **RAM ユーザーにすべての RDS インスタンスへのアクセス権限を付与します。**

```
{
  " Statement ": [
    {
      " Action ": " dms : LoginDatab  ase ",
```

```
      " Effect ":" Allow ",
      " Resource ": " acs : rds :*:*:*"
    }
  ],
  " Version ": " 1 "
}
```

12 RAM を使用した SLB 権限管理

本ドキュメントでは、RAM にポリシーを作成して RAM ユーザーの SLB 権限を管理する方法について説明します。

共通ポリシー

次の表は、SLB のアクセス許可を管理するために RAM に作成できる共通ポリシーの一覧です。

ポリシー	説明
AliyunSLBFullAccess	SLB インスタンスに対する完全な管理権限を RAM ユーザーに付与します。
AliyunSLBReadOnlyAccess	SLB インスタンスに対する RAM ユーザーに読み取り専用権限を付与します。

注：

SLB 権限の詳細は、RAM の権限付与 をご参照ください。

RAM ユーザーへのカスタムポリシーのアタッチ

1. 本文の SLB 権限付与の例に従ってカスタムポリシーを作成します。

   詳細は、#unique_5 をご参照ください。
2. 対象のポリシーを見つけてクリックします。
3. リファレンスタブで、権限の付与 をクリックします
4. プリンシパルフィールドに、対象 RAM ユーザーの ID または名前を入力します。
5. OK をクリックします。

注：

必要に応じて、RAM ユーザーまたは RAM ユーザーグループにポリシーをアタッチすることもできます。 詳細は、RAM での権限付与 をご参照ください.

SLB 権限付与の例

- 例 1：複数のインスタンスを持つ RAM 管理者として、ユーザーに 2 つのインスタンスのみを操作する権限を付与します。

  これら 2 つの SLB インスタンスの ID は、i-001 と i-002 とします。

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

```
      " Effect ": " Allow ",
      " Action ": " slb :*",
      " Resource ": [
                  " acs : slb :*:*: loadbalanc  er / i - 001 ",
                  " acs : slb :*:*: loadbalanc  er / i - 002 "
                  ]
    },
    {
      " Effect ": " Allow ",
      " Action ": " slb : Describe *",
      " Resource ": "*"
    }
  ],
  " Version ": " 1 "
}
```

注：

- **権限付与された RAM ユーザーは、すべての SLB インスタンスを表示できますが、操作できるのはそのうちの 2 つのみです。**
- **ポリシーには `Describe *` 要素が必要です。 ポリシーに `Describe *` 要素が含まれていない場合、権限付与された RAM ユーザーはコンソールにインスタンスを表示できません。 ただし、RAM ユーザーは、API の呼び出し、CLI の使用、または SLB SDK の使用によって、指定された 2 つの SLB インスタンスを操作できます。**

- **例 2：RAM 管理者として、ユーザーに SLB インスタンスに ECS インスタンスを追加する権限を付与します。 SLB インスタンスの ID は i-001 とします。**

```
{
  " Statement ": [
    {
      " Effect ": " Allow ",
      " Action ": " slb : AddBackend  Servers ",
      " Resource ": [" acs : slb :*:*: loadbalanc  er / slb - 001 "]
    },
    {
      " Effect ": " Allow ",
      " Action ": " slb : AddBackend  Servers ",
      " Resource ": [" acs : ecs :*:*: instance / i - 001 "]
    },
    {
        " Effect ": " Allow ",
        " Action ": " slb : DescribeLo  adBalancer  s ",
        " Resource ": " acs : slb :*:*: loadbalanc  er /*"
    }
  ],
  " Version ": " 1 "
}
```

注：

例 1 のポリシーに従って SLB 管理権限を RAM ユーザーに付与した後、ユーザーが ECS インスタンスを追加または削除したり、必要に応じて ECS インスタンスの重みを設定できるように、次の権限をユーザーに付与する必要もあります。

- SLB リソースに対する権限付与
- ECS リソースに対する権限付与

- 例 3：RAM 管理者として、ユーザーに指定された SLB インスタンスに対して ECS 関連の操作を実行する権限を付与します。

```
{
  " Statement ": [
    {
      " Effect ": " Allow ",
      " Action ": " slb :*",
      " Resource ": [
                  " acs : slb :*:*: loadbalanc  er / i - 001 ",
                  " acs : slb :*:*: loadbalanc  er / i - 002 "
                  ]
    },
    {
      " Effect ": " Allow ",
      " Action ": " slb : Describe *",
      " Resource ": "*"
    },
    {
      " Effect ": " Allow ",
      " Action ": " slb :*",
      " Resource ": " acs : ecs :*:*:*"
    }
  ],
  " Version ": " 1 "
}
```

 注：

前述のポリシーにより、RAM ユーザーは 2 つの指定された SLB インスタンス（ID：i-001 および i-002）を管理し、これら 2 つの SLB インスタンスに対して ECS 関連のすべての操作を実行できます。たとえば、これら 2 つの SLB インスタンスに対する ECS インスタンスの追加や、ECS の重みの設定を行うなど。

13 RAM を使用した CDN アクセス権限の管理

本ドキュメントでは、RAM にポリシーを作成して RAM ユーザーの CDN 権限を管理する方法について説明します。

共通ポリシー

次の表は、DNS 権限を管理するために RAM に作成できる共通ポリシーの一覧です。

ポリシー	説明
AliyunCDNFullAccess	RAM ユーザーに CDN インスタンスの完全管理権限を付与します。
AliyunCDNReadOnlyAccess	RAM ユーザーに CDN インスタンスに対する読み取り専用権限を付与します。

注：

CDN 権限の詳細は、API 権限付与ルール をご参照ください。

RAM ユーザーに CDN インスタンスに対する読み取り専用権限、キャッシュのリフレッシュ権限、およびプッシュ操作権限を付与します。

1. カスタマイズポリシーを作成します。

```
{
  " Version ": " 1 ",
  " Statement ": [
    {
      " Action ": [
        " cdn : Describe *",
        " cdn : PushObject  Cache ",
        " cdn : RefreshObj  ectCaches "
      ],
      " Resource ": " acs : cdn :*:*:*",
      " Effect ": " Allow "
    }
  ]
}
```

詳細、#unique_5 をご参照ください.

2. 対象のポリシーを見つけてクリックします。
3. リファレンス タブで、権限の付与 をクリックします。
4. プリンシパルフィールドに、対象 RAM ユーザーの ID または名前を入力します。

5. OK をクリックします。

 **注：**

必要に応じて、RAM ユーザーまたは RAM ユーザーグループにポリシーをアタッチすることもできます。 詳細は、RAM での権限付与 をご参照ください.

14 ActionTrail を使用した RAM 操作の記録

本ドキュメントでは、ActionTrail を使用して Alibaba Cloud アカウントまたは RAM ユーザーの操作をリソースに記録する方法について説明します。

ActionTrail を使用して RAM 操作を表示

1. ActionTrail コンソール にログインします。
2. 履歴検索 ページで、フィルタ ドロップダウンリストを使用して対象のイベントを検索します。
3. イベントをクリックして、 イベントの表示 をクリックします。

ActionTrail で記録される操作

ActionTrail は以下の RAM 操作を記録できます。

- Alibaba Cloud アカウントまたは RAM ユーザーのログイン情報。 詳細については、ConsoleSignin イベントログの例 をご参照ください.
- RAM コンソールでの操作。 次は、記録された操作イベントの例です。

```
{
    " apiVersion ":" 2015 - 05 - 01 ",
    " eventId ":" 2cc52dee - d8d2 - 40c2 - 8de0 - 3a2cf1df ****",
    " eventName ":" DeleteGrou  p ",
    " eventSourc  e ":" ram . aliyuncs . com ",
    " eventTime ":" 2015 - 11 - 03T13 : 41 : 49Z ",
    " eventType ":" ApiCall ",
    " eventVersi  on ":" 1 ",
    " requestId ":" 9AE24F49 - C52C - 4F0F - BCF9 - 9A4B8C22B1  47
 ",
    " requestPar  ameters ":{
        " GroupName ":" grp1 ",
    },
    " serviceNam  e ":" Ram ",
    " sourceIpAd  dress ":" 42 . 120 . XX . XX ",
    " userAgent ":" AliyunCons  ole ",
    " userIdenti  ty ":{
        " type ":" ram - user ",
        " principalI  d ":" 2741806465  4829 ****",
        " accountId ":" 1234567890  12 ****",
        " userName ":" Alice ",
        " sessionCon  text ":{
            " sessionAtt  ributes ":{
                " creationDa  te ":" 2015 - 11 - 03T13 : 41 : 48Z ",
                " mfaAuthent  icated ":" true "
            }
        }
    }
}
```

- RAM および STS API は、リソースの作成、変更および削除を要求します。 次は記録されたイベントの一例です。

```
{
    " apiVersion ": " 2015 - 05 - 01 ",
    " eventId ": " 234ef3c7 - 8938 - 4bd7 - bb80 - 11754b7b ****",
    " eventName ": " CreateGrou  p ",
    " eventSourc  e ": " ram . aliyuncs . com ",
    " eventTime ": " 2016 - 01 - 04T08 : 58 : 50Z ",
    " eventType ": " ApiCall ",
    " eventVersi  on ": " 1 ",
    " recipientA  ccountId ": " 43274 ",
    " requestId ": " 1485748C - DB62 - 4693 - AB7E - 4BA3F3A970  E1
 ",
    " requestPar  ameters ": {
        " Comments ": " this   is   a   test   group ",
        " GroupName ": " grp1 "
    },
    " serviceNam  e ": " Ram ",
    " sourceIpAd  dress ": " 42 . 120 . XX . XX ",
    " userAgent ": " aliyuncli / 2 . 0 . 6 ",
    " userIdenti  ty ": {
        " type ": " ram - user ",
        " principalI  d ": " 2741806465  4829 ****",
        " accountId ": " 43274 ",
        " accessKeyI  d ": " f6Iz ******** EI4d ",
        " userName ": " Alice "
    }
}
```

次のステップ

操作の記録について詳しくは、ActionTrail イベントログ構文 をご参照ください。

15 RAM ユーザーに対する ActionTrail リソースの使用の許可

このトピックでは、システムポリシーまたはカスタムポリシーを使用して、RAM ユーザーに ActionTrail リソースの使用を許可する方法について説明します。

始める前に

1. ActionTrail API の操作と説明を確認します。 詳細は、「RAM アカウントへの権限付与」をご参照ください。
2. RAM ポリシーの構造と構文を確認します。 詳細は、「ポリシー構造と構文」 をご参照ください。

手順

1. RAM ユーザーを作成します。

   詳細は、「RAM ユーザーの作成」をご参照ください。
2. RAM ユーザに権限を付与します。

   - RAM ユーザーに必要な権限を付与するには、後述の「ActionTrail 関連のシステムポリシー」に従って 1 つ以上のシステムポリシーを割り当てます。

     詳細は、「RAM での権限付与」をご参照ください。
   - RAM ユーザーに細かく権限を付与するには、後述の「許可例」に従ってカスタムポリシーを作成します。

     詳細は、「カスタマイズポリシーの作成」をご参照ください。

ActionTrail 関連のシステムポリシー

次の表は、ActionTrail で一般的に使用されているシステムポリシーの一覧です。

表 15-1 : システムポリシー

システムポリシー	説明
AliyunActionTrailFullAccess	RAM ユーザーに ActionTrail リソースに対するフル管理権限を付与します。
AliyunActionTrailReadOnlyAccess	RAM ユーザーに ActionTrail リソースに対する読み取り専用権限を付与します。

許可例

- **例 1：RAM 管理者として、ユーザーに読み取り専用権限を付与します。**

```
{
    " Version ": " 1 ",
    " Statement ": [{
        " Effect ": " Allow ",
        " Action ": [
            " actiontrai  l : LookupEven  ts ",
            " actiontrai  l : Describe *",
            " actiontrai  l : Get *"
        ],
        " Resource ": "*"
    }]
}
```

- **例 2：RAM 管理者として、特定の IP アドレスからログインしたユーザーに読み取り専用権限を付与します。**

```
{
    " Version ": " 1 ",
    " Statement ": [{
        " Effect ": " Allow ",
        " Action ": [
            " actiontrai  l : LookupEven  ts ",
            " actiontrai  l : Describe *",
            " actiontrai  l : Get *"
        ],
        " Resource ": "*",
        " Condition ":{
            " IpAddress ": {
                " acs : SourceIp ": " 42 . 120 . XX . X / 24 "
            }
        }
    }]
}
```