Стратегия управления доступом в AWS

SOFTPROM
softprom.com • info@softprom.com

amazon web services | Partner Network
ADVANCED CONSULTING PARTNER

2020

Стратегия управления доступом в AWS

Пользователи: создавайте отдельных пользователей.

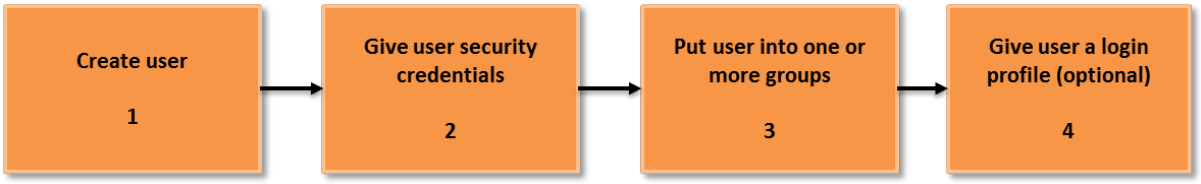
It is a security best practice to not use your root account because the root account grants access to all services and resources. Grant users the minimum amount of privilege necessary, which is known as least privilege.

You have other people in your group who have varied access and authorization permissions. When you use IAM users, it is easier to assign policies to specific users that access specific services and associated resources.

An IAM user can use the AWS CLI.

An IAM user can use a role.

The following diagram describes the canonical use case for creating an IAM user:



Группы: управляйте разрешениями с помощью групп.

A group is a collection of IAM users. Groups let you assign permissions to a collection of users, which can make it easier to manage the permissions for those users. For example, you could have a group called Admins and give that group the types of permissions that administrators typically need. Any user in that group automatically has the permissions that are assigned to the group. If a new user joins your organization and should have administrator privileges, you can assign the appropriate permissions by adding the user to that group. Similarly, if a person changes jobs in your organization, instead of editing that user's permissions, you can remove him or her from the old group and add him or her to the new group.

Разрешения: назначайте минимальные привилегии.

Чтобы назначить разрешения для пользователя, группы, роли или ресурса, создайте политику, которая позволит указать следующее.

- Actions. Разрешенные действия для сервиса AWS. Например, можно разрешить пользователю вызывать действие Amazon S3 ListBucket. Любые действия, которые вы явно не разрешаете выполнять, запрещаются.
- Resources. Ресурсы AWS, для которых разрешено выполнять определенные действия. Например, список корзин Amazon S3, для которых вы разрешаете пользователю выполнять действие ListBucket. Пользователи не могут получить доступ к тем ресурсам, для которых вы явно не предоставляете разрешения.
- Effect. Разрешать или запрещать доступ. Поскольку по умолчанию доступ запрещен, вы обычно создаете правила, которые разрешают определенные действия.

- Conditions. Условия, которые должны выполняться, чтобы применялась политика. Например, можно разрешить доступ только к определенным корзинам S3, если пользователь подключается с определенного диапазона IP-адресов или использует при входе в систему многофакторную аутентификацию.

Политики создаются с помощью визуального редактора или в формате JSON. Политика состоит из одного или нескольких выражений, каждое из которых описывает один набор разрешений. Подробнее о языке правил см. в справке по правилам AWS IAM.

Визуальный редактор проведет вас через процесс предоставления разрешений с помощью правил IAM без необходимости писать правила в формате JSON (при этом возможность создавать и редактировать правила в JSON сохраняется). Правило, показанное на следующем снимке экрана, было создано с помощью визуального редактора. Оно предоставляет разрешение на пять действий Amazon S3 типа List и Read для корзины S3 и объектов в SampleBucket, префикс которых начинается с MyPrefix.

При использовании для управления разрешениями Консоли управления AWS можно просматривать сводную информацию о правиле. В сводной информации о правиле перечислены уровень доступа, ресурсы и условия для каждого сервиса, определенного в правиле (см. пример на приведенных ниже снимках экрана). Чтобы было легче понять разрешения, определенные в правиле, действия каждого сервиса AWS разбиты на четыре категории по уровню доступа: List, Read, Write и Permissions management.

Service ▾	Access level	Resource	Request condition
Allow (10 of 94 services)			
CloudFormation	**Full**: List **Limited**: Read, Write	All resources	None
CloudWatch Logs	Full access	Multiple	None
EC2	**Full**: List **Limited**: Read	All resources	None
Elastic Beanstalk	Full access	All resources	elasticbeanstalk:InApplication = arn:aws:elasticbeanstalk:*:111122223333:application/Bank-Devl

Вы можете выбрать предопределенное правило под управлением AWS или создать собственное, используя генератор правил. Подробнее см. в разделе Обзор правил IAM Руководства по использованию IAM.

Аудит: включите сервис AWS CloudTrail.

AWS CloudTrail позволяет отслеживать историю аккаунта и автоматически реагировать на действия, которые угрожают безопасности используемых ресурсов AWS. Используя интеграцию с Amazon CloudWatch Events, можно определить рабочие процессы, которые требуется запускать при обнаружении событий, способных привести к уязвимостям системы безопасности. Например, можно создать рабочий процесс, который будет добавлять определенную политику к корзине Amazon S3, когда CloudTrail обнаружит вызов API, открывающий публичный доступ к этой корзине.

Пароль: настройте политику требований к надежности паролей.

Управление данными доступа с помощью IAM

Сервис AWS Identity and Access Management (IAM) позволяет управлять несколькими типами долгосрочных данных для безопасного доступа пользователей IAM.

- Пароли. Используются для входа на защищенные страницы AWS, такие как Консоль управления AWS и форумы AWS.
- Ключи доступа. Используются для программных запросов к AWS из API AWS, интерфейса командной строки AWS, AWS SDK или инструментов AWS для Windows PowerShell.
- Пары ключей Amazon CloudFront. Используются в CloudFront для создания подписанных URL-адресов.
- Открытые ключи SSH. Используются при прохождении аутентификации в репозиториях AWS CodeCommit.

Назначить своим пользователям IAM данные для доступа к ресурсам AWS можно с помощью API, интерфейса командной строки или Консоли управления AWS. Можно осуществлять ротацию и отзывать данные для доступа, когда это необходимо. Кроме управления этими данными для доступа пользователей, можно дополнительно повысить безопасность доступа пользователей IAM к ресурсам AWS путем принудительного использования multi-factor authentication (MFA).

Временные данные для доступа

IAM также позволяет предоставлять пользователям временные данные для доступа к вашим ресурсам AWS с определенным сроком действия. Например, использование временного доступа полезно в следующих случаях.

- Создается мобильное приложение, в котором используется сторонний вход.
- Создается мобильное приложение с собственной аутентификацией.
- Для предоставления доступа к ресурсам AWS используется система аутентификации некой организации.
- Для предоставления доступа к ресурсам AWS используется система аутентификации некой организации и SAML.
- Используется единый вход в систему (SSO) через веб-интерфейс для Консоли управления AWS.
- Сторонним лицам делегируется доступ к API для обращения к ресурсам в вашем аккаунте или в другом аккаунте, которым вы владеете.

MFA: включите аутентификацию MFA для привилегированных пользователей.

AWS Multi-Factor Authentication (MFA) is a simple best practice that adds an extra layer of protection on top of your user name and password. With MFA enabled, when a user signs in to an AWS Management Console, they will be prompted for their user name and password (the first factor—what they know), as well as for an authentication code from their AWS MFA device (the second factor—what they have). Taken together, these multiple factors provide increased security for your AWS account settings and resources.
You can enable MFA for your AWS account and for individual IAM users you have created under your account. MFA can be also be used to control access to AWS service APIs.
After you've obtained a supported hardware or virtual MFA device, AWS does not charge any additional fees for using MFA.
You can also protect cross-account access using MFA.

Virtual MFA Applications

Applications for your smartphone can be installed from the application store that is specific to your phone type. The following table lists some applications for different smartphone types.

Android	Authy, Duo Mobile, LastPass Authenticator, Microsoft Authenticator, Google Authenticator
iPhone	Authy, Duo Mobile, LastPass Authenticator, Microsoft Authenticator, Google Authenticator

U2F Security Key

AWS supports U2F security key as a MFA device for accessing the AWS Management Console using certain web browsers. We encourage you to use virtual or hardware MFA for

the AWS Console Mobile App. For more information, please review the configurations associated with U2F security key supported by AWS.

Роли: используйте роли IAM для инстансов Amazon EC2.

Роли IAM позволяют предоставить права доступа пользователям или сервисам, у которых обычно нет доступа к ресурсам AWS вашей организации. Пользователям IAM или сервисам AWS можно присвоить роли для получения временных данных для доступа, которые они могут использовать для вызовов API AWS. В результате не требуется предоставлять долгосрочные данные для доступа или назначать разрешения для каждого объекта, которому требуется доступ к определенному ресурсу.
В следующих сценариях выделены некоторые из проблем, с которыми вы можете столкнуться при делегировании доступа.

- Предоставление приложениям, работающим на инстансах Amazon EC2, доступа к ресурсам AWS

Чтобы предоставить приложениям на инстансе Amazon EC2 доступ к ресурсам AWS, разработчики могут распространить на каждый инстанс свои данные для доступа. Затем приложения могут использовать эти данные для доступа к ресурсам, таким как корзины Amazon S3 или данные Amazon DynamoDB. Однако предоставление каждому инстансу данных для доступа на длительный срок – спорный момент, создающий потенциальную угрозу безопасности. В приведенном выше видеоматериале подробно описывается, как использовать роли для решения подобной проблемы безопасности.

- Доступ к нескольким аккаунтам

Для контроля или управления доступом к ресурсам, например изолирования рабочей среды от среды разработки, может понадобиться несколько аккаунтов AWS. Однако в некоторых случаях пользователям из одного аккаунта может потребоваться доступ к ресурсам другого аккаунта. К примеру, пользователю из среды разработки может быть необходим доступ к рабочей среде для продвижения обновлений. Поэтому у пользователей должны быть данные, подтверждающие права доступа к каждому аккаунту; однако управление несколькими наборами таких данных для нескольких аккаунтов затрудняет управление удостоверениями. Использование роли IAM может упростить эту задачу.

- Предоставление разрешений сервисам AWS

Прежде чем сервисы AWS смогут выполнять какие-либо действия от вашего имени, необходимо предоставить им соответствующие разрешения. Можно использовать роли AWS IAM, чтобы предоставить разрешения сервисам AWS вызывать другие сервисы AWS от вашего имени или создавать ресурсы AWS и управлять ими в вашем аккаунте. Сервисы AWS, такие как Amazon Lex, также предлагают связанные с сервисом роли, которые являются предварительно настроенными и могут приниматься только этим конкретным сервисом.

Совместный доступ: используйте роли IAM для предоставления совместного доступа.

Identity federation is a system of trust between two parties for the purpose of authenticating users and conveying information needed to authorize their access to resources. In this

system, an identity provider (IdP) is responsible for user authentication, and a service provider (SP), such as a service or an application, controls access to resources. By administrative agreement and configuration, the SP trusts the IdP to authenticate users and relies on the information provided by the IdP about them. After authenticating a user, the IdP sends the SP a message, called an assertion, containing the user's sign-in name and other attributes that the SP needs to establish a session with the user and to determine the scope of resource access that the SP should grant. Federation is a common approach to building access control systems which manage users centrally within a central IdP and govern their access to multiple applications and services acting as SPs.
AWS offers distinct solutions for federating your employees, contractors, and partners (workforce) to AWS accounts and business applications, and for adding federation support to your customer-facing web and mobile applications. AWS supports commonly used open identity standards, including Security Assertion Markup Language 2.0 (SAML 2.0), Open ID Connect (OIDC), and OAuth 2.0

Enabling federated AWS access for your workforce

You can use two AWS services to federate your workforce into AWS accounts and business applications: AWS Single Sign-On (SSO) or AWS Identity and Access Management (IAM). AWS SSO is a great choice to help you define federated access permissions for your users based on their group memberships in a single centralized directory. If you use multiple directories, or want to manage the permissions based on user attributes, consider AWS IAM as your design alternative. To learn more about service quotas and other design considerations in AWS SSO, see the AWS SSO User Guide. For AWS IAM design considerations, see the AWS IAM User Guide.

Using federation to enable single sign-on (SSO) to your AWS accounts

AWS SSO makes it easy to centrally manage federated access to multiple AWS accounts and business applications and provide users with single sign-on access to all their assigned accounts and applications from one place. You can use AWS SSO for identities in the AWS SSO's user directory, your existing corporate directory, or external IdP.

AWS SSO works with an IdP of your choice, such as Okta Universal Directory or Azure Active Directory (AD) via the Security Assertion Markup Language 2.0 (SAML 2.0) protocol. AWS SSO seamlessly leverages IAM permissions and policies for federated users and roles to help you manage federated access centrally across all AWS accounts in your AWS Organization. With AWS SSO, you can assign permissions based on the group membership in your IdP's directory, and then control the access for your users by simply modifying users and groups in the IdP. AWS SSO also supports the System for Cross-domain Identity Management (SCIM) standard for enabling automatic provisioning of users and groups from Okta Universal Directory, Azure AD, and other supported IdPs to AWS.

AWS SSO can serve as an IdP to authenticate users to AWS SSO integrated applications and SAML 2.0 compatible cloud-based applications, such as Salesforce, Box, and Office 365, with a directory of your choice. You can also use AWS SSO to authenticate users to the AWS Management Console, AWS Console Mobile Application, and AWS Command Line Interface (CLI). For your identity source, you can choose Microsoft Active Directory or AWS SSO's user directory.

Using AWS IAM to manage federated fine-grained access to AWS accounts

You can enable federated access to AWS accounts using AWS Identity and Access Management (IAM). The flexibility of the AWS IAM allows you to enable a separate SAML 2.0 or an Open ID Connect (OIDC) IdP for each AWS account and use federated user attributes for access control. With AWS IAM, you can pass user attributes, such as cost center or job role, from your IdPs to AWS, and implement fine-grained access permissions based on these attributes. AWS IAM helps you define permissions once, and then grant, revoke or modify AWS access by simply changing the attributes in the IdP. You can apply the same federated access policy to multiple AWS accounts by implementing reusable custom managed IAM policies.

Enabling federated access to your customer-facing web and mobile apps

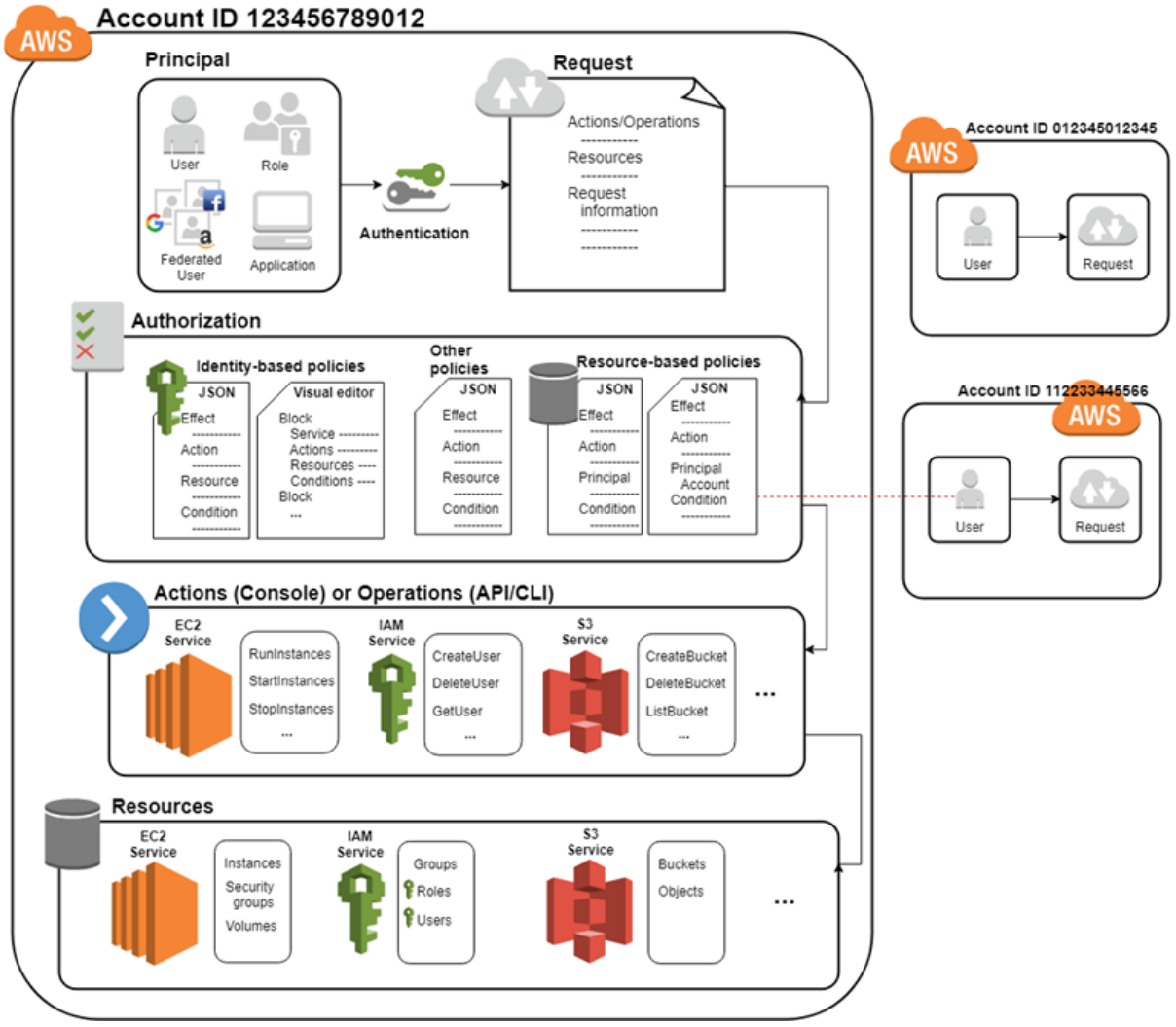
You can add federation support to your customer-facing web and mobile applications using Amazon Cognito. It helps you add user sign-up, sign-in, and access control to your mobile and web apps quickly and easily. Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Apple, Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0.

Ротация: регулярно изменяйте данные для доступа.

Условия: ограничьте привилегированный доступ с помощью дополнительных условий

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. When a principal makes a request in AWS, the AWS enforcement code checks whether the principal is authenticated (signed in) and authorized (has permissions). You manage access in AWS by creating policies and attaching them to IAM identities or AWS resources. Policies are JSON documents in AWS that, when attached to an identity or resource, define their permissions.
During authorization, the AWS enforcement code uses values from the request context to check for matching policies and determine whether to allow or deny the request.
AWS checks each policy that applies to the context of the request. If a single policy denies the request, AWS denies the entire request and stops evaluating policies. This is called an explicit deny. Because requests are denied by default, IAM authorizes your request only if every part of your request is allowed by the applicable policies. The evaluation logic for a request within a single account follows these rules:

- By default, all requests are implicitly denied. (Alternatively, by default, the AWS account root user has full access.)
- An explicit allow in an identity-based or resource-based policy overrides this default.
- If a permissions boundary, Organizations SCP, or session policy is present, it might override the allow with an implicit deny.
- An explicit deny in any policy overrides any allows.



After your request has been authenticated and authorized, AWS approves the request. If you need to make a request in a different account, a policy in the other account must allow you to access the resource. In addition, the IAM entity that you use to make the request must have an identity-based policy that allows the request.

Root: максимально сократите или запретите использование аккаунта с правами root.