AWS Architecting and SysOps

Introducing and Securing AWS

June-July 2019

Contents

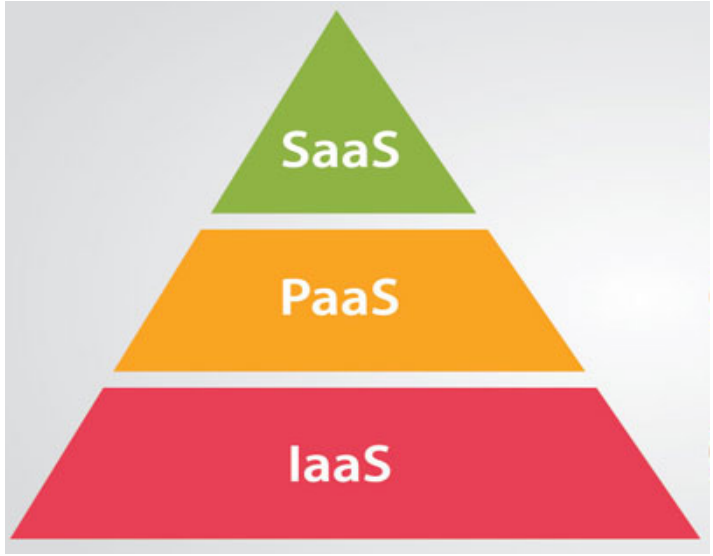
Introduction

Amazon Web Services Overview

What is AWS?

- Global cloud platform
  - Compute power
  - Database storage
  - Content delivery
  - Other functionality to help business scale and grow

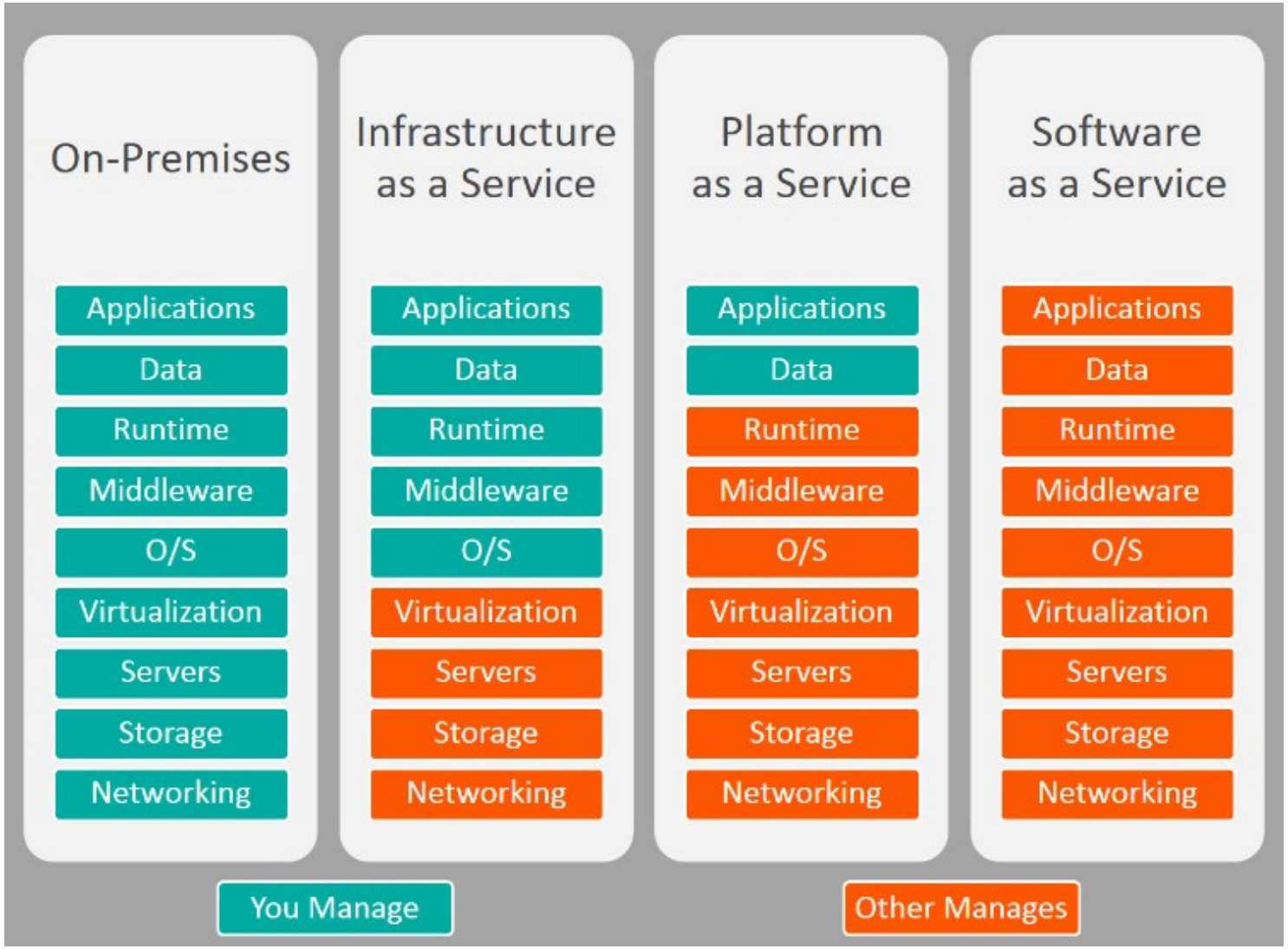
- As a cloud provider, it offers:
  - Infrastructure as a Service (IaaS)
  - Platform as a Service (PaaS)
  - Software as a Service (SaaS)



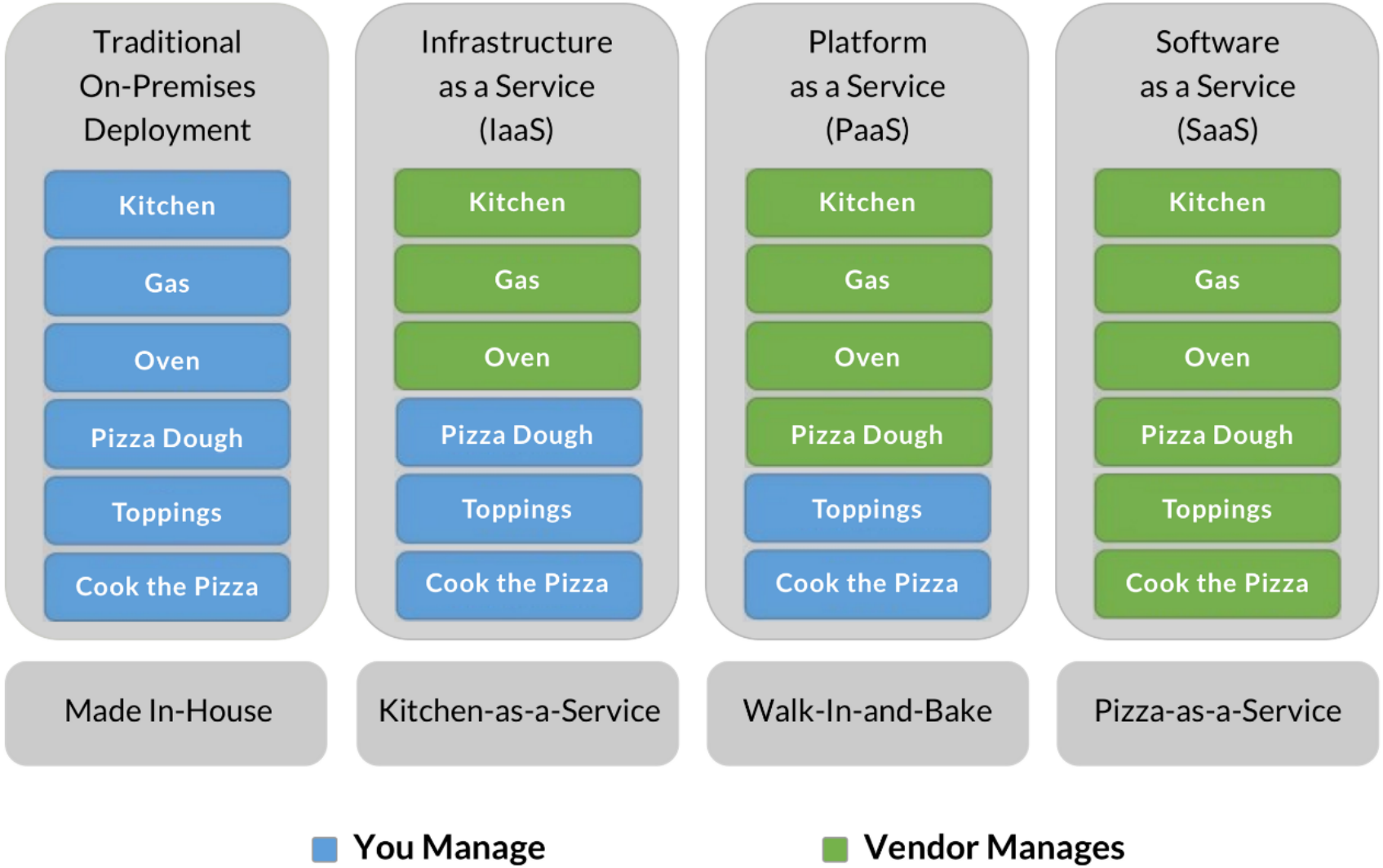
AWS cloud services

- Infrastructure as a service (IaaS)
  - You rent IT infrastructure (servers, network) from AWS on a pay-as-you-go basis
  - You can outsource and build a “virtual data center” in the cloud and have access to the associated resources
  - ➢ Elastic Compute Cloud (EC2)
- Platform as a service (PaaS)
  - The vendor provides a platform on which software can be developed and deployed
  - You deploy your own application using programming languages, tools...
  - ➢ Elastic Beanstalk to deploy web applications and services
- Software as a service (SaaS)
  - The vendor host and manage the software application on a subscriptions basis
  - You maintain the control of a software environment but do not maintain any equipment
  - ➢ Amazon Web Services in general

AWS cloud services



New Pizza as a Service



Why AWS?

- Easy sign up process
  - Just sign up with an id and credit card

- Billing is very clear and simple
  - Pay-Per-Use and it is very transparent
  - Dashboard with several reports

- Cost-Effectiveness
  - No upfront investment, long-term commitment or minimum expense

- Flexibility
  - More time for core business tasks through instant availability of new services
  - You get a choice in running services and applications

Why AWS?

- Scalability / Elasticity
  - Autoscaling and elastic load balancing automatically scale resources
  - They are scale up or down depending on customer demand

- Security
  - End-to-end security and privacy following major security standards (ISO, DoD…)

- Stability
  - In all these years, only 2 o 3 major outages, only affecting a specific region

- Trusted vendor
  - It is used by everyone in the industry, independently of the size of the company
  - AWS listens to its customer feedback always

AWS customers

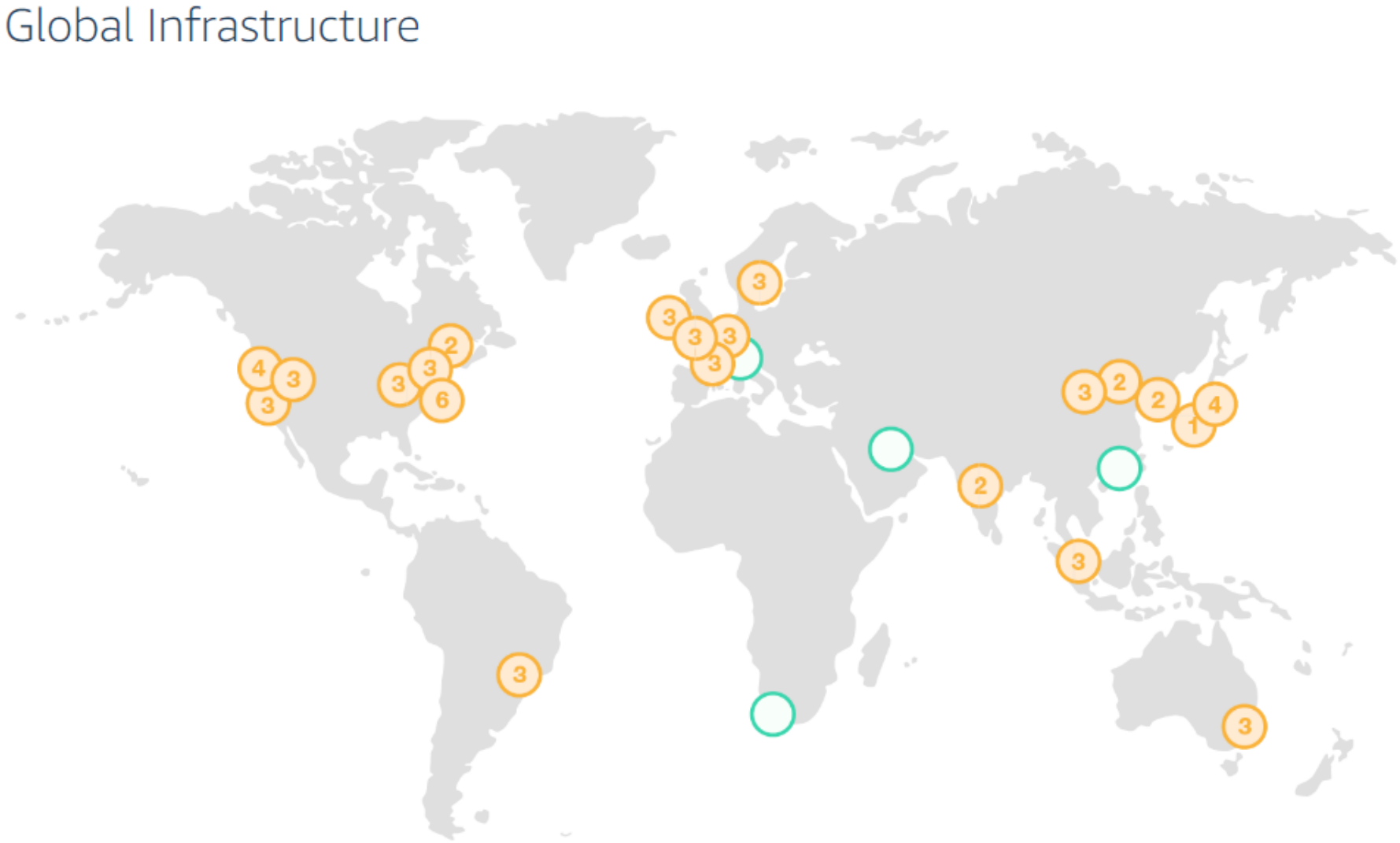
- Pinterest can maintain:
  - Site scalability (S3, EC2)
  - Manages multiple petabytes of data everyday
  - ➢ It is a small team
  - ➢ Did not want to spend effort in IT

- Spotify used AWS to:
  - Scale its capacity (S3)
  - Store its vast repository
  - ➢ Needed quick scalability
  - ➢ (without long lead times)

- Netflix is able to:
  - Deploy servers for storage
  - Allow users to stream shows from anywhere in the world
  - ➢ All its computer and storage needs
  - ➢ Used tools to replace inefficient processes

- Expedia chose AWS due to:
  - Highly scalable infrastructure
  - Better cloud services
  - ➢ All IT management to AWS
  - ➢ Own developers have faster development

AWS Global Infrastructure



- AWS Cloud spans
  - ✓ 66 Availability Zones [AZ] (Massive Data Centers)
  - ✓ Within 21 regions

AWS Global Infrastructure

Region & Number of Availability Zones

US East
N. Virginia (6),
Ohio (3)

US West
N. California (3),
Oregon (4)

Asia Pacific
Mumbai (3),
Seoul (3),
Singapore (3),
Sydney (3),
Tokyo (4),
Osaka-Local (1)[1]

Canada
Central (2)

China
Beijing (2),
Ningxia (3)

Europe
Frankfurt (3),
Ireland (3),
London (3),
Paris (3),
Stockholm (3)

South America
São Paulo (3)

GovCloud (US)
US-East (3),
US-West (3)

New Region (coming soon)

Bahrain

Cape Town

Hong Kong SAR

Milan

- Not all regions cost the same
  - North Virginia is the cheapest, followed by Ohio
  - In Europe: Stockholm, followed by Ireland
    https://www.concurrencylabs.com/blog/choose-your-aws-region-wisely/
- 100% High Availability target
  - There are a minimum of 2 availability zones per region (for non-local regions like Osaka)
  - They are connected to each other with fast, private fiber-optic networking
  - You can increase redundancy and fault tolerance further by replicating data between regions

Future

- 160+ top level services spread across 23 categories (June 2019)
- Continuous launch of new services in all domains
  - Opening new fields like Robotics, Satellite…
  - Yearly re:Invent conference is used as service launch
- Continuous cost reduction
  - They are focused on cost effectiveness
- Continuous evolution of certification
  - 11 certifications, 4 levels

https://cloudacademy.com/blog/choosing-the-right-aws-certification/

Security with IAM

The most relevant aspects securing AWS

Who is responsible for security?

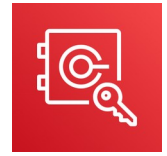
- Responsibilities of AWS
  - Protecting the network through automated monitoring systems and robust internet access, to prevent Distributed Denial of Service (DDoS) attacks
  - Performing background checks on employees who have access to sensitive areas
  - Decommissioning storage devices by physically destroying them after end of life
  - Ensuring the physical and environmental security of data centers, including fire protection and security staff
- Responsibilities of customers
  - Implementing access management that restricts access to AWS resources like S3 and EC2 to a minimum, using AWS IAM
  - Encrypting network traffic to prevent attackers from reading or manipulating data (HTTPS)
  - Configuring a firewall for your virtual network that controls incoming and out- going traffic with security groups and ACLs
  - Encrypting data at rest. For example, enable data encryption for your database / storage systems
  - Managing patches for the OS and additional software on virtual machines

Security, Identity and Compliance

AWS Identity and Access Management (IAM)

AWS WAF

AWS Key Management Service



AWS CloudHSM

IAM (Identity and Access Management)

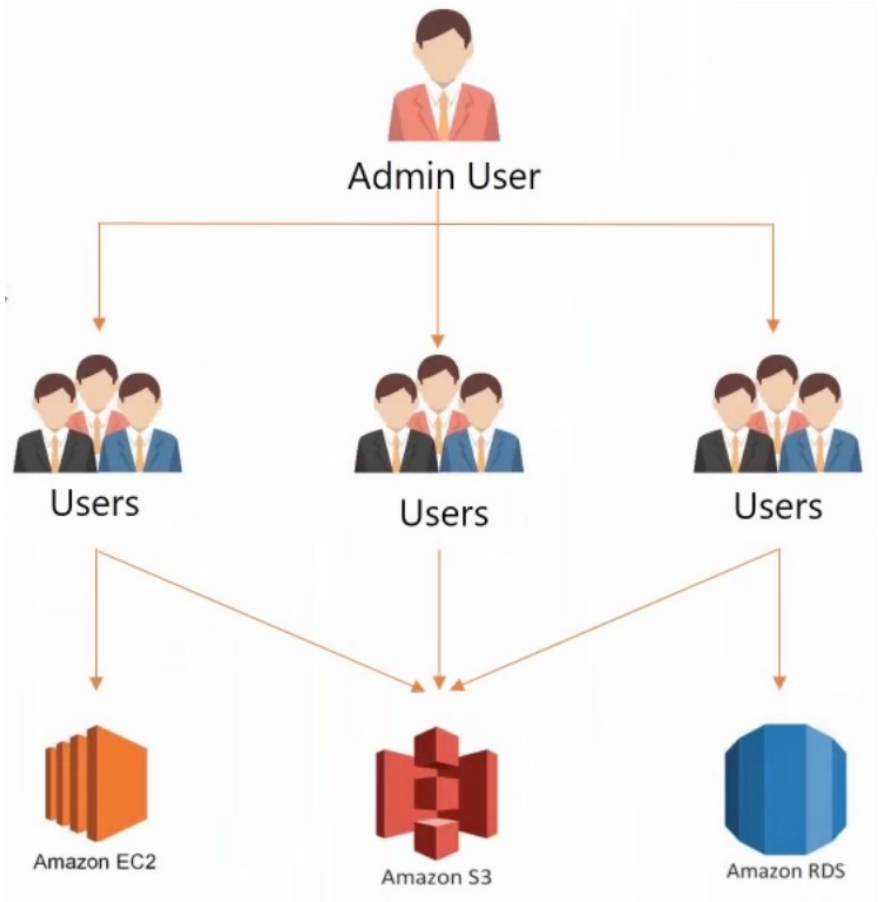
- IAM is a web service that helps you to manage access to AWS services and resources securely
- Using IAM you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources
- IAM support users inside and outside AWS through third party applications
- IAM is offered at no additional charge.
  - You will be charged only for use of other AWS services by your users

IAM Key Features

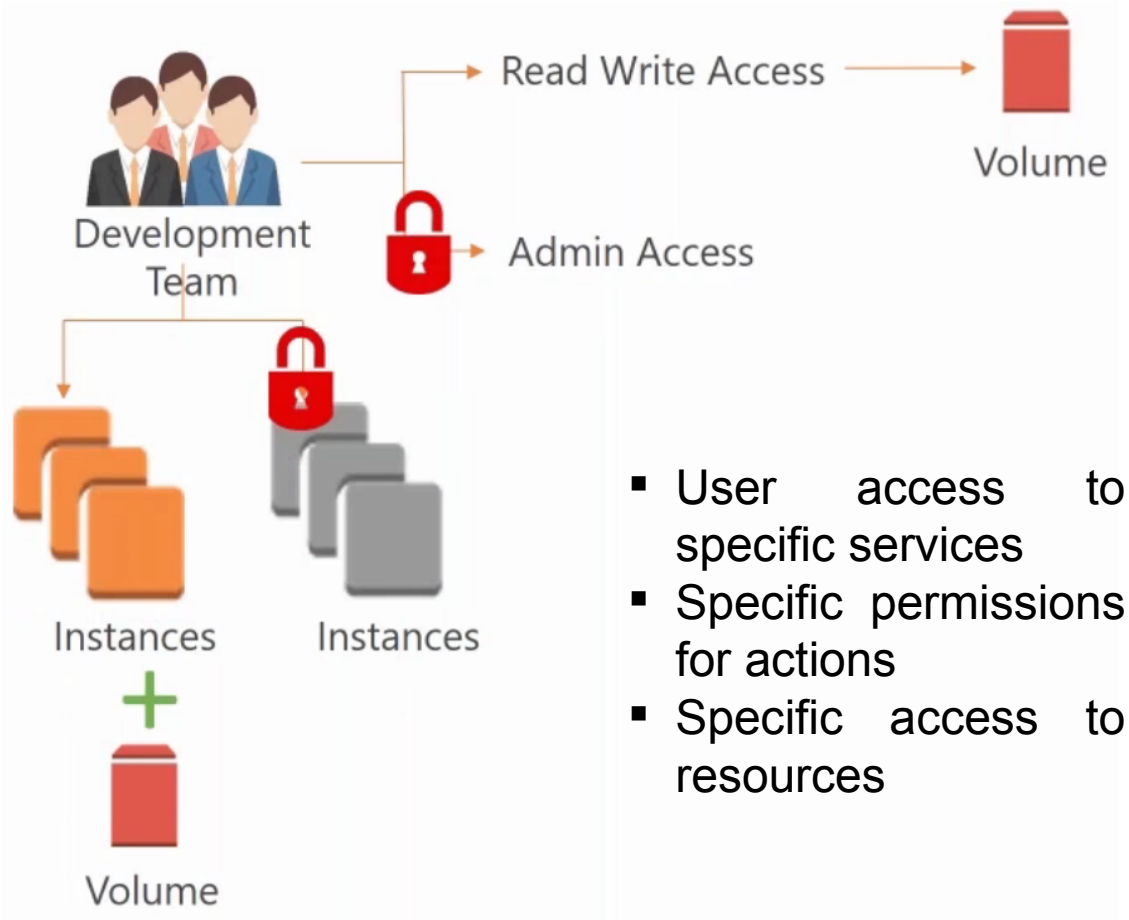
- Shared access to your AWS account
- Granular permissions
- Secure access to AWS resources for applications running on EC2
- Identity federation (trusting authentication from a 3rd party)
- Multi-factor authentication (MFA)
- Identity information
  - Log, monitor and track what users are doing with your AWS resources
- Free to use
  - No additional charges for creating users, groups, policies…
- PCI DSS (Payment Card Industry and Data Security Standard) compliance
- Password policy

IAM Key Features

Shared access to your AWS account and resources



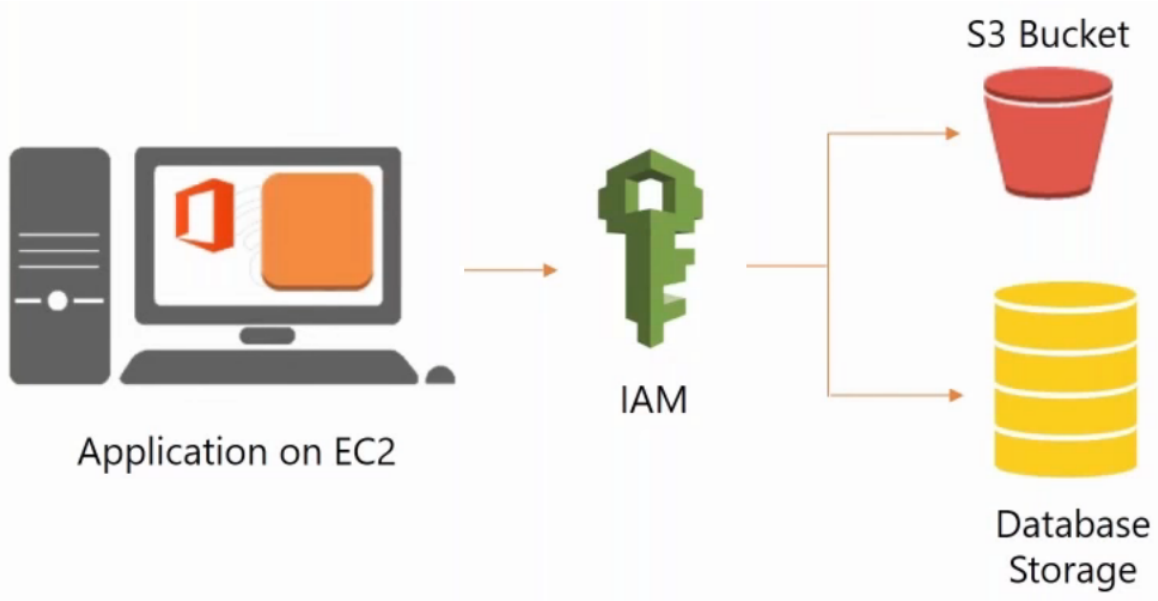
Granular permissions to the different users



- User access to specific services
- Specific permissions for actions
- Specific access to resources

IAM Key Features

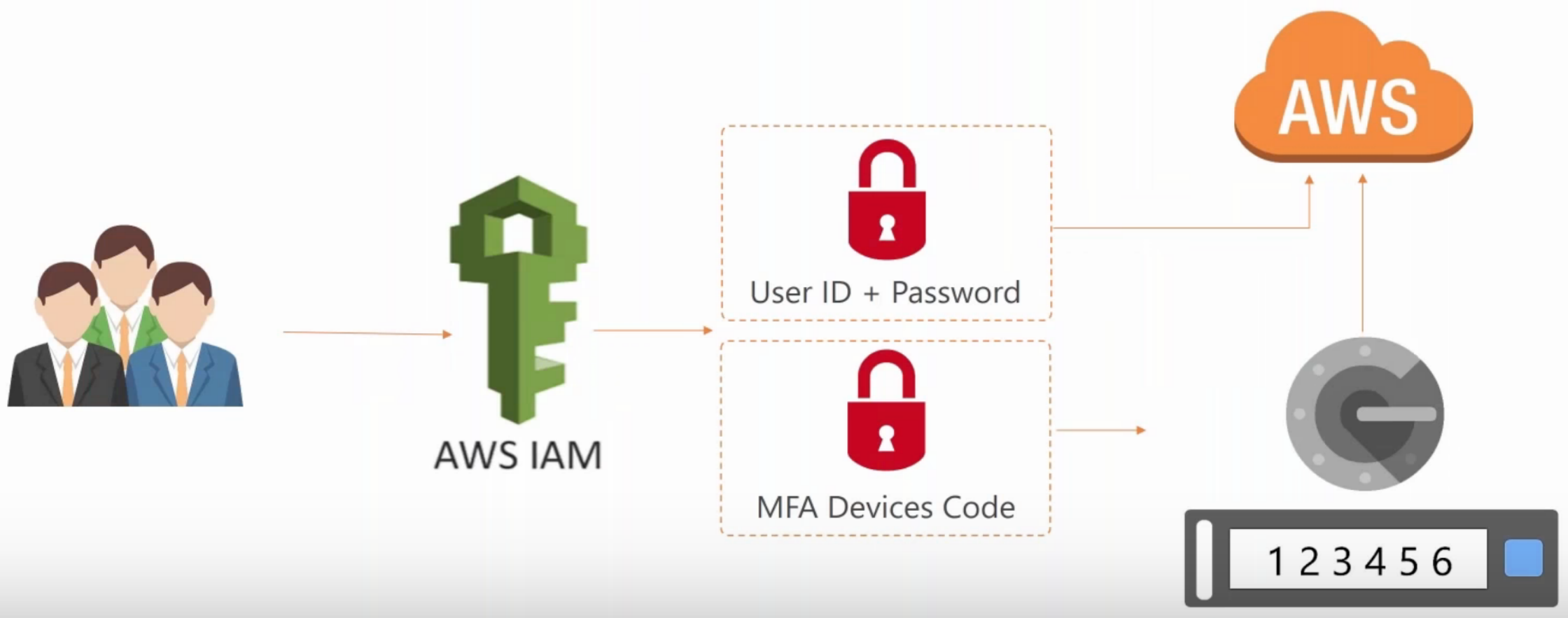
Secure access for applications running on EC2



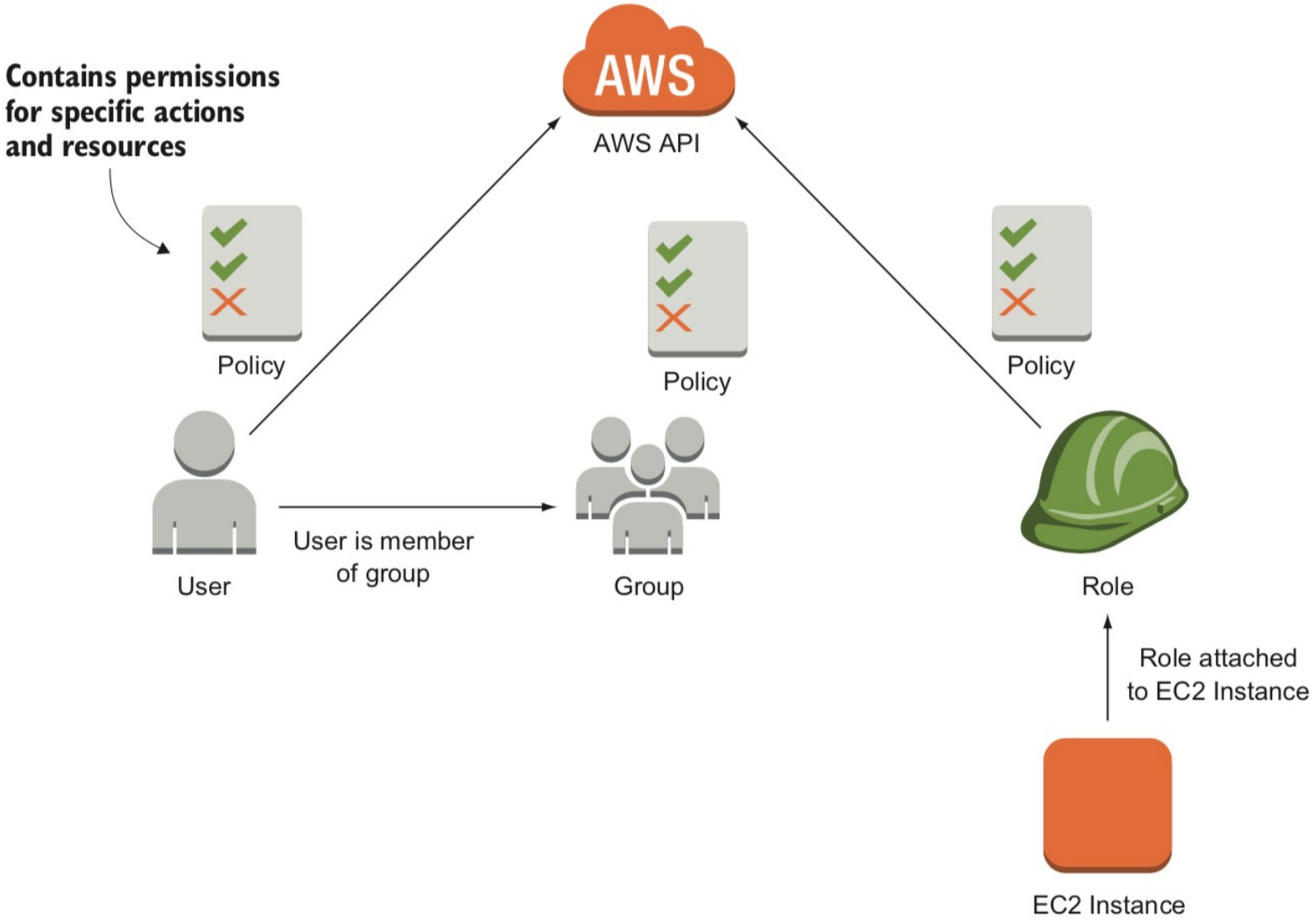
Identity federation: grating temporary access to AWS

IAM Key Features

MFA: two-factor authorization for users and resources to ensure absolute security using MFA devices



IAM Big Picture



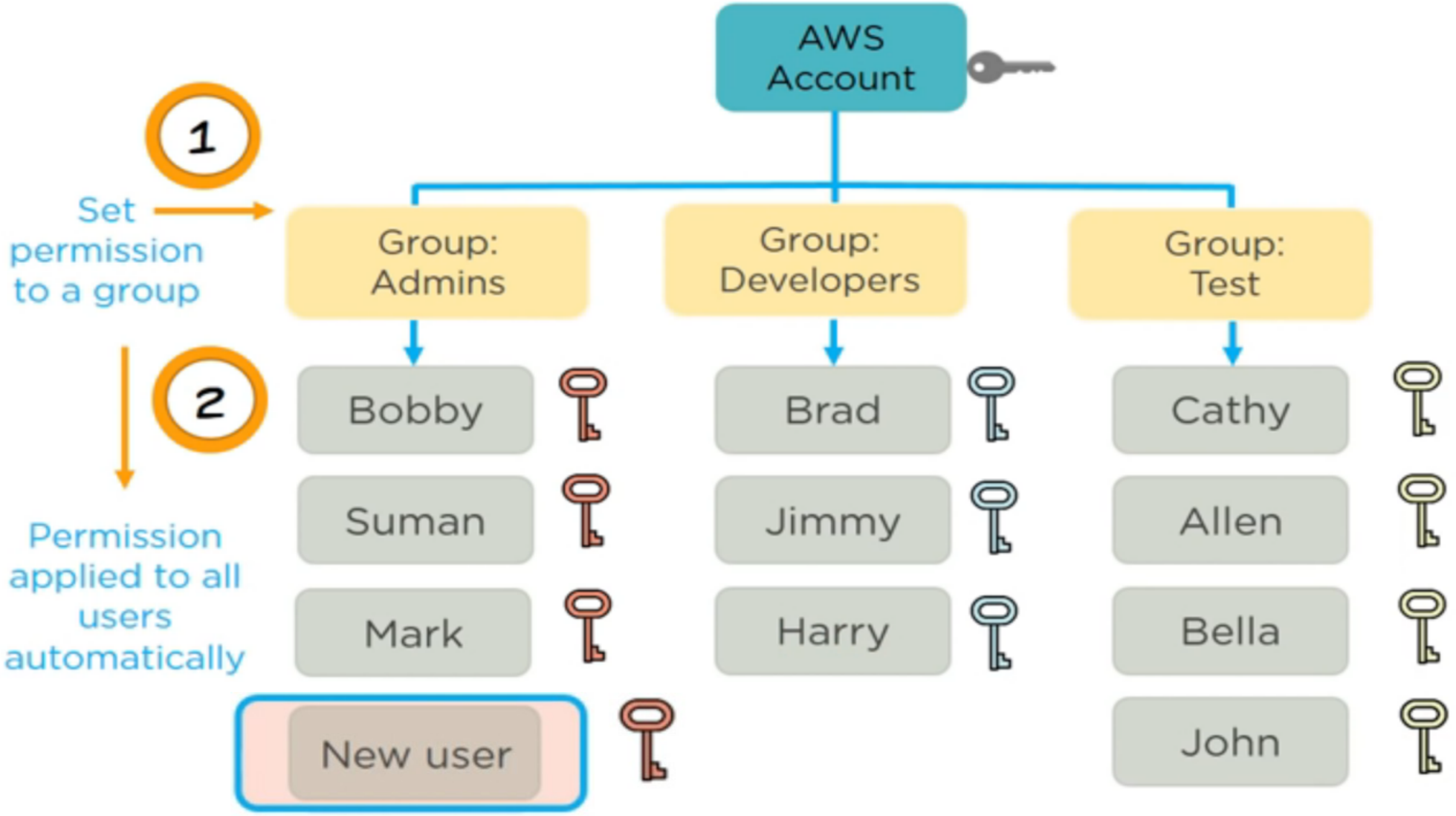
IAM Users

- An IAM user is a trusted entity associated with only one AWS account
- All users must be authenticated before they can use AWS resources
  - Each user can have its own credentials
- You can create individual IAM users that correspond to your own organization
- By default, IAM users cannot access anything in your account
  - You need to grant permissions to users by creating an IAM policy
- The bill is common to the account, it does not depend on individual users

IAM Groups

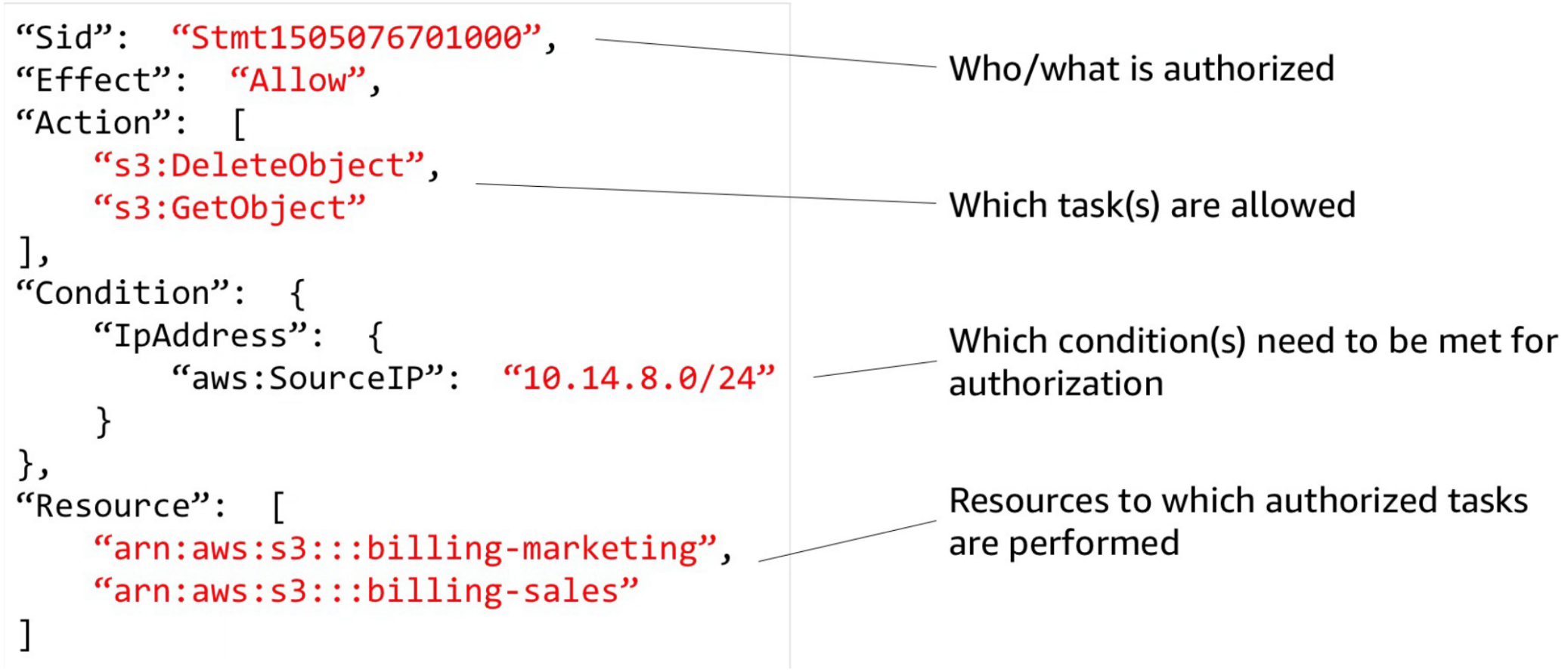
- A collection of IAM users is an IAM group
- You can use IAM groups to grant permissions for multiple users, so that any permission applied to the group, are applied to its users as well
- A recommended practice is to grant permissions to groups, so they can have the same policy
- By default, an IAM user is not included in any specific group

IAM Groups: a small organization example



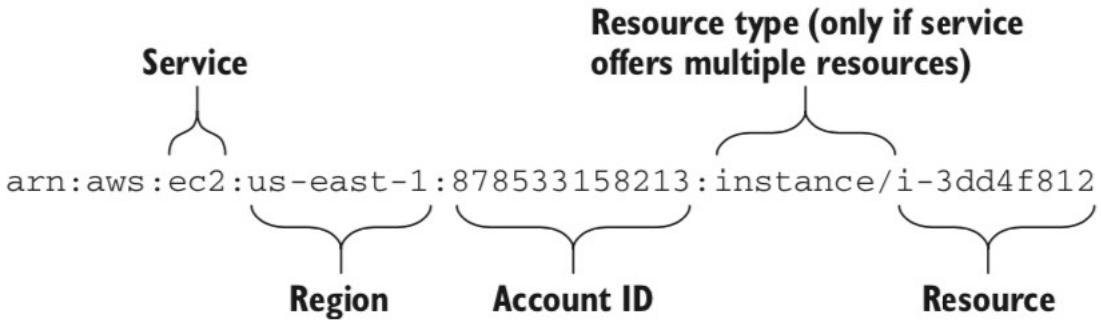
IAM Policies

- IAM policies are JSON documents used to describe permissions within AWS



IAM Policies: JSON document

- Sid
  - Who or what it has been authorized
  - It could be a user, a group or another resource within the AWS
- Effect
  - Allow / Deny
- Action
  - Which tasks are allowed to be performed: read, write, erase…
- Condition
  - Which condition or conditions need to be met for authorization
  - In this case, only calls from the IP address specified have the designed permissions
- Resource
  - Resources to which authorized tasks are performed



IAM Policies

- If you have multiple statements that apply to the same action, Deny overrides Allow. The following policy allows all EC2 actions except terminating EC2 instances

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "1",
    "Effect": "Allow",
    "Action": "ec2:*",
    "Resource": "*"
  }, {
    "Sid": "2",
    "Effect": "Deny",
    "Action": "ec2:TerminateInstances",
    "Resource": "*"
  }]
}
```

Deny

Terminating EC2 instances

IAM Policies

- When you deny an action, you can't allow that action with another statement:

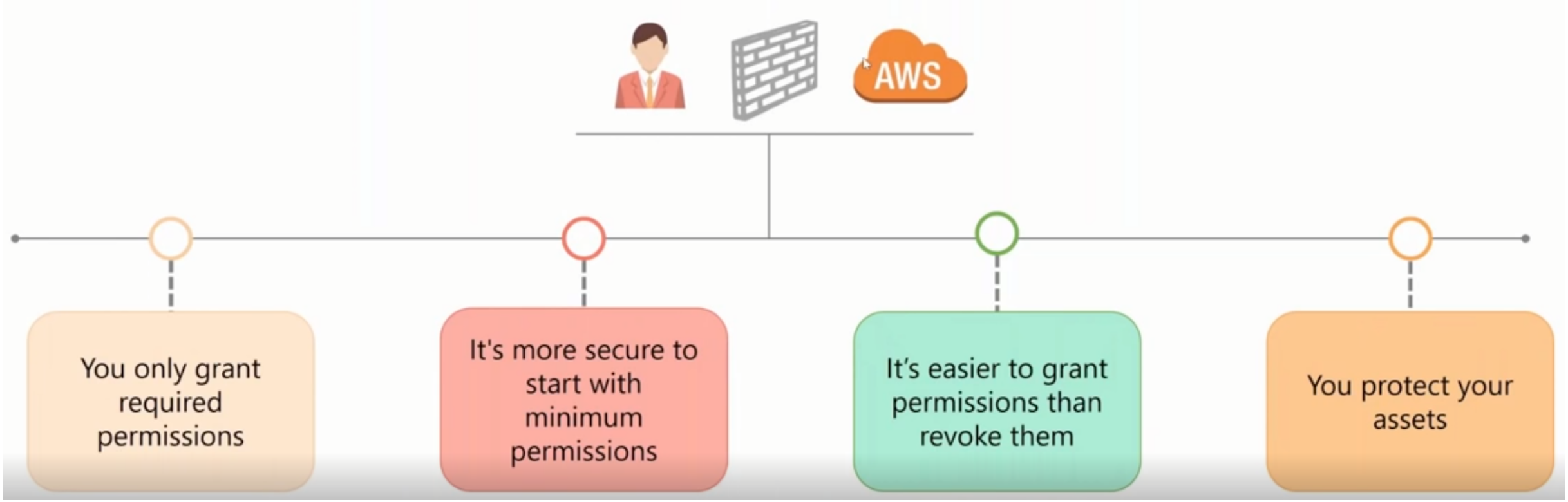
```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "1",
    "Effect": "Deny",
    "Action": "ec2:*",
    "Resource": "*"
  }, {
    "Sid": "2",
    "Effect": "Allow",
    "Action": "ec2:TerminateInstances",
    "Resource": "*"
  }]
}
```

Denies every EC2 action

Allow isn't crucial; Deny overrides Allow.

IAM least privilege for policies

- It is recommended to grant least privilege for policies



IAM Policies and Permissions

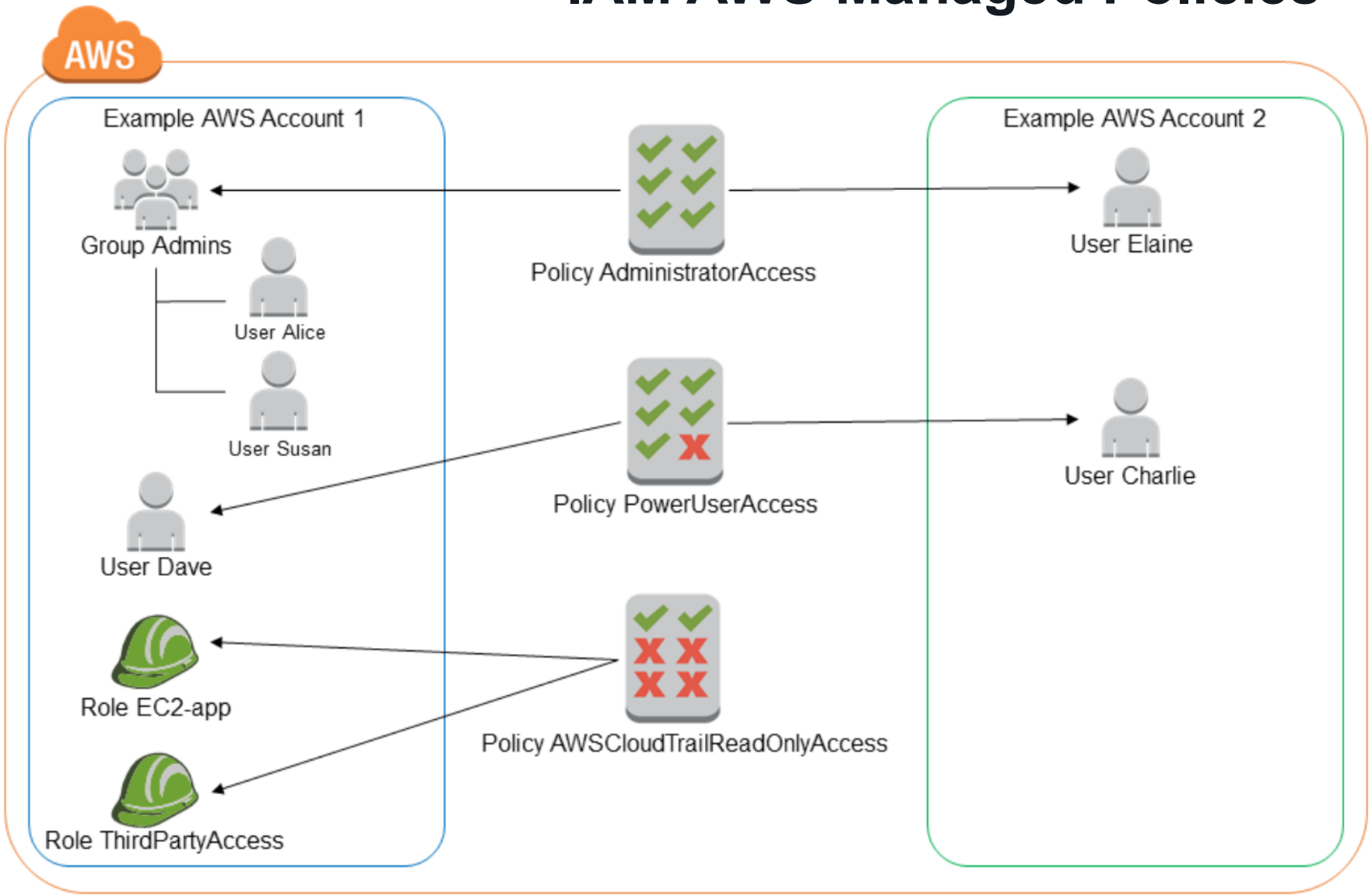
- You manage access to AWS by creating policies and attaching them to IAM identities or AWS resources
- A policy is an object in AWS that, when associated with an identity or resource, defines their permissions
  - AWS evaluates each policy when a principal entity makes a request
  - Permissions in the policies determine whether the request is allowed or denied
- AWS policies can be identity-based or resource-based
- Identity-based policies: Inline policies and Managed policies
  - Attach policies to IAM identities
  - Inline policies can`t exist without the IAM role, user, or group that it belongs to
- Resource-based policies
  - Attach policies to specific resources

IAM Identity-Based Policies

- A managed policy is divided in two types:
  - AWS managed policies
    - Standalone policies created and managed by AWS
  - Customer managed policies
    - Policies that you create and manage in your AWS account
    - They provide more precise control over your policies than the AWS managed one

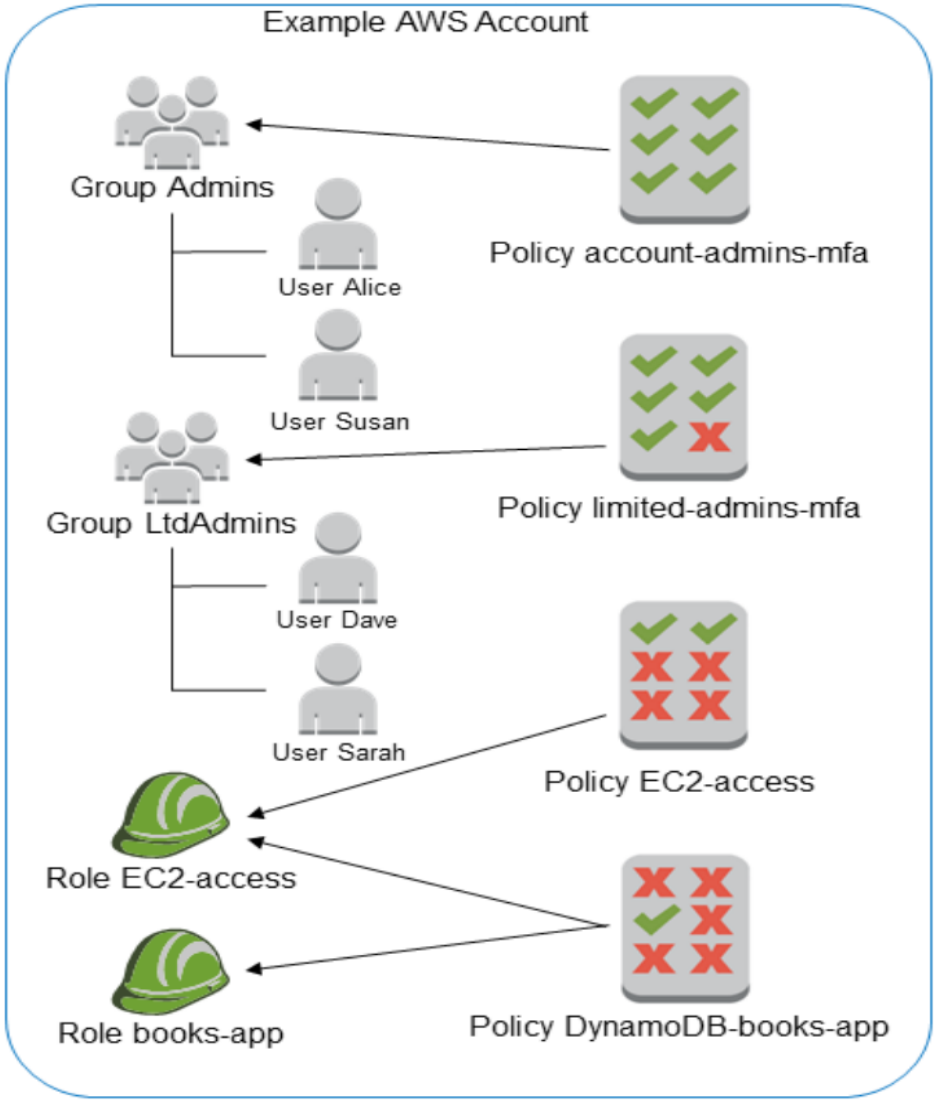
https://docs.aws.amazon.com/es_es/IAM/latest/UserGuide/reference_iam-limits.html

IAM AWS Managed Policies



- Standalone policies created and managed by AWS

IAM Customer Managed Policies



- Policies that you create and manage in your AWS account
- They provide more precise control over your policies than the AWS managed one

IAM: Bringing it all together

- Only use root admin account to create an IAM user account with admin rights

IAM: Bringing it all together

- Assuming your company has several departments that will be managing AWS services

- STEP 1
  - Create an IAM group for each department
  - Using groups is highly recommended as it is the most efficient way to handle users

- STEP 2
  - Create a policy and assign it to the group
  - Any users added to this group will inherit those permissions

- STEP 3
  - Create IAM users for each person in each department and added it to the respective groups

IAM Roles

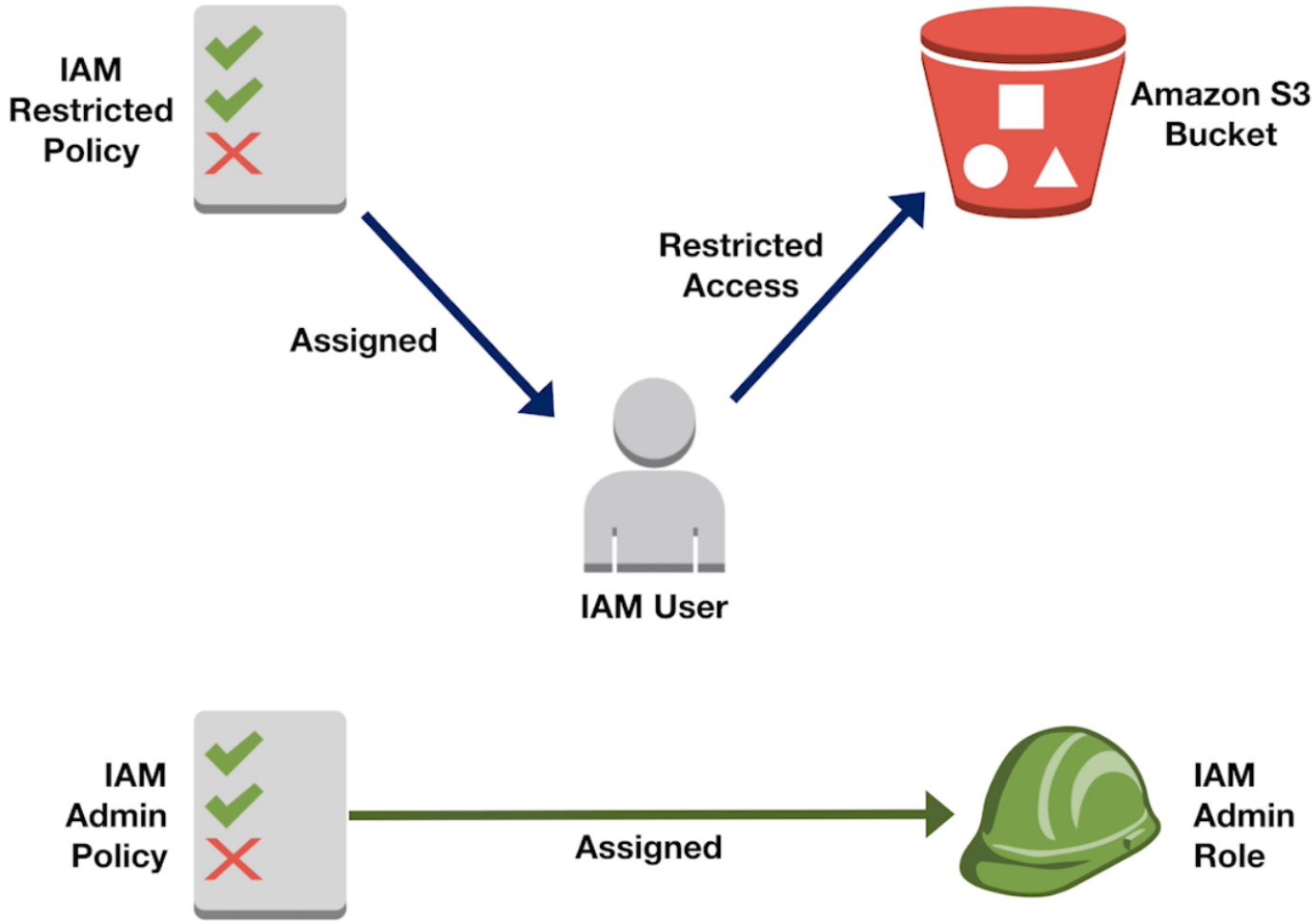
- An IAM role is an IAM entity that defines a set of permissions for making AWS service requests

- Used to delegate (= provide) access to:
  - IAM users managed within your account or under a different AWS account
  - AWS services, such as EC2

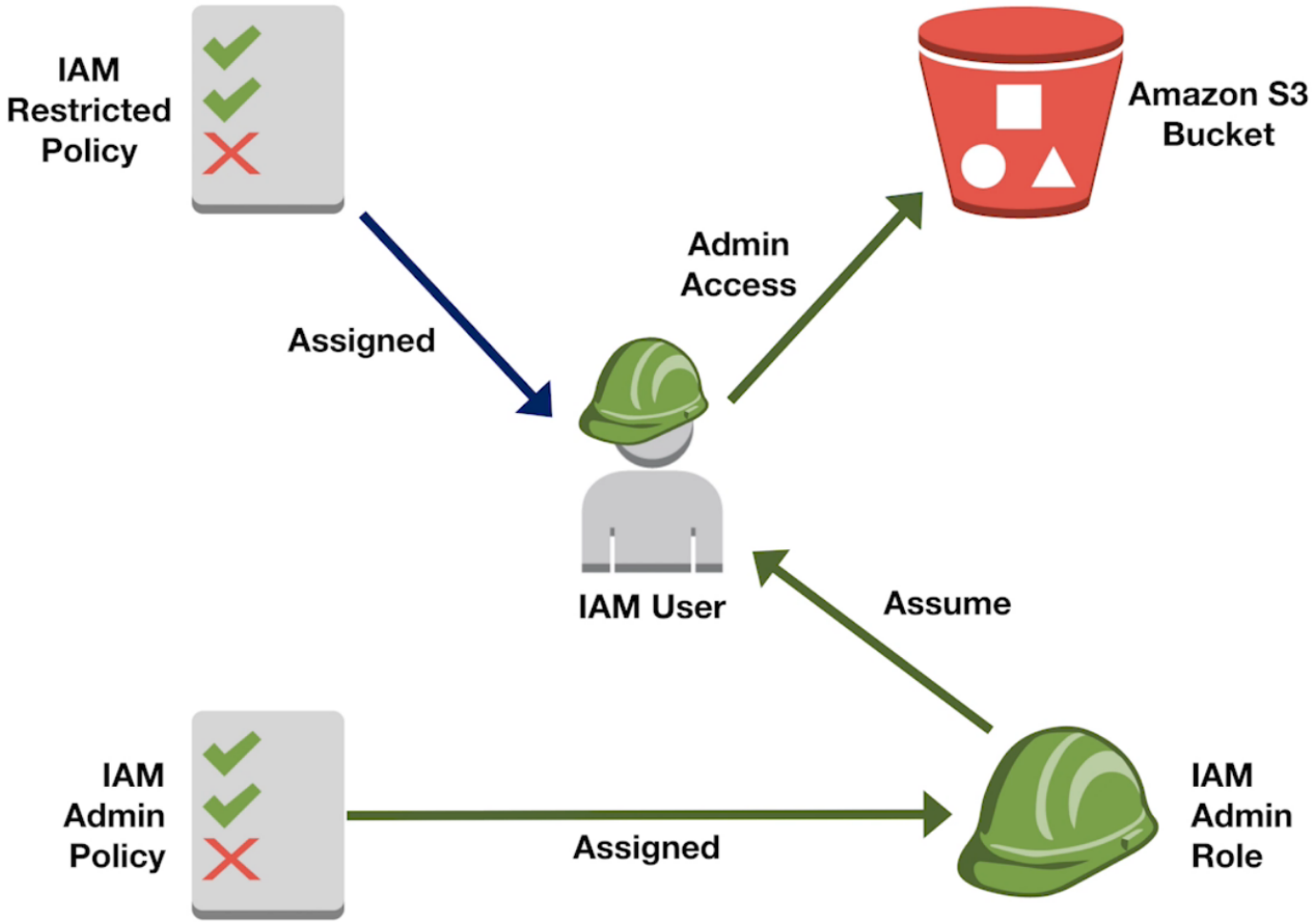
- IAM roles allows you to delegate access with defined permissions to trusted entities without having to share long-term access keys
  - Security credentials are created dynamically and assigned temporary
  - The trusted entity gives up its permissions and gets the permissions of the role

- The creation is similar to a user
  - Name the role and attach a policy to it

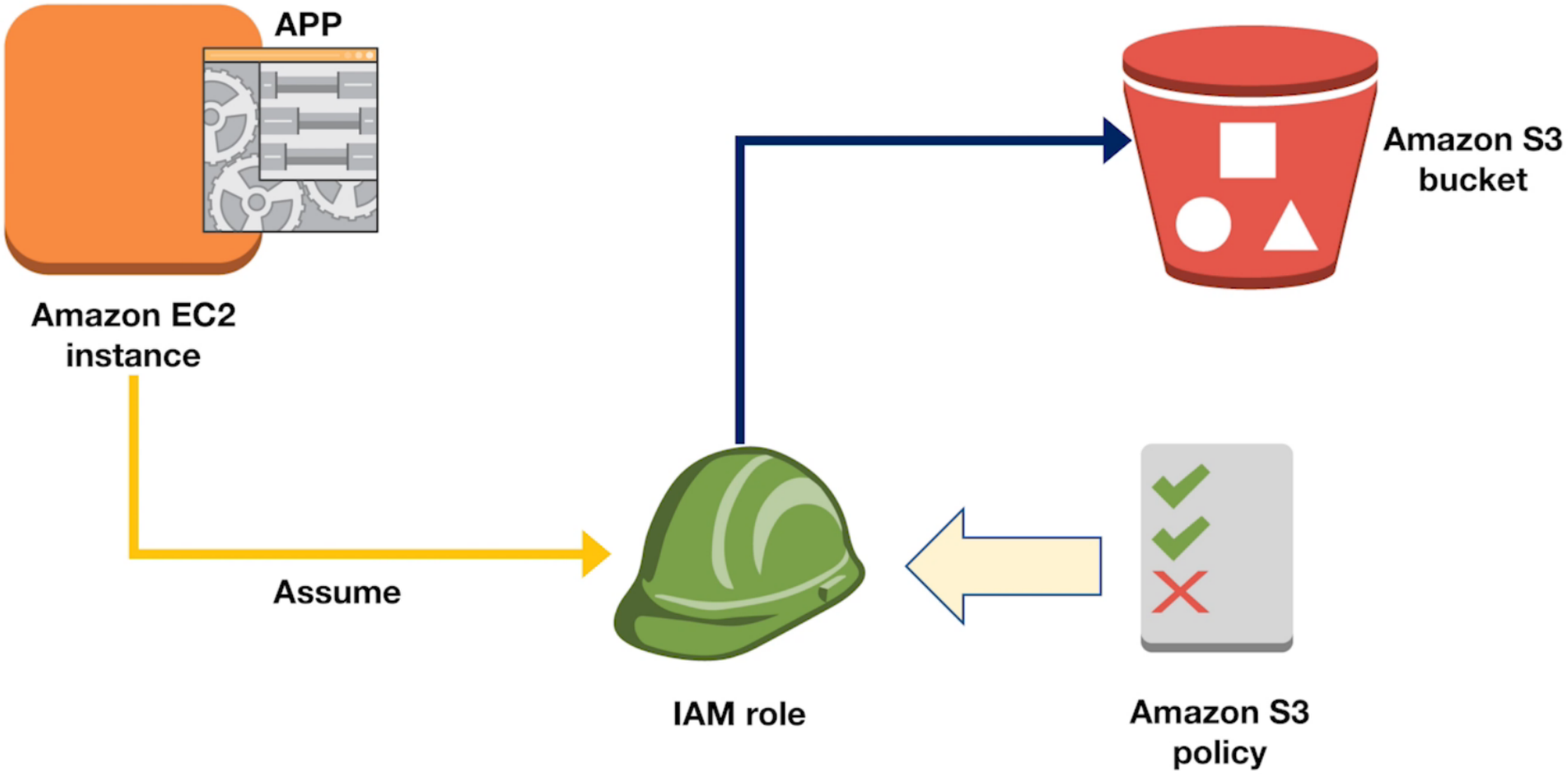
IAM Roles



IAM Roles



IAM Roles



AWS Security Token Service (STS)

➢Who is able to manage roles in AWS? STS

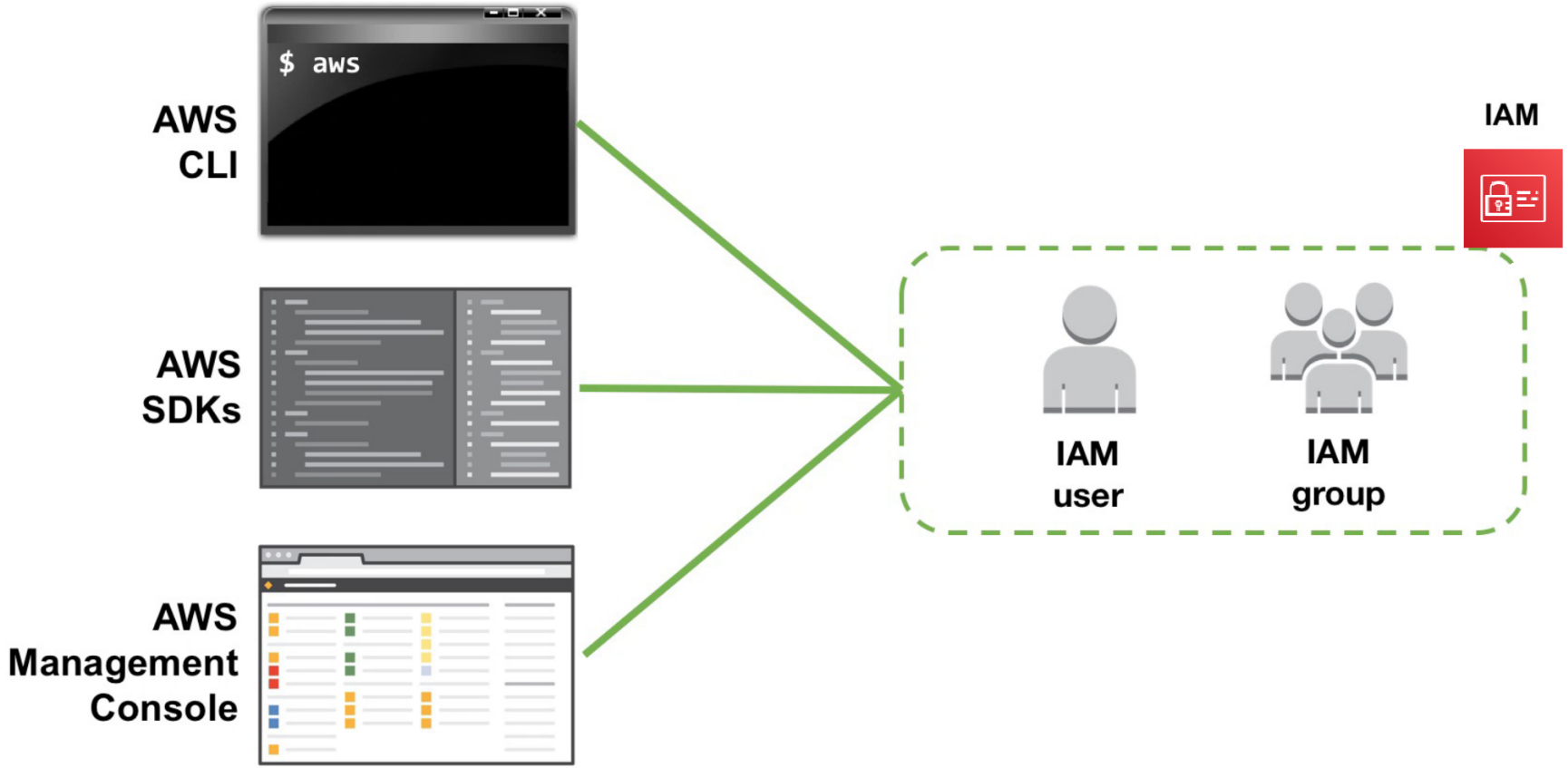
- STS is a web service that enables you to request temporary, limited-privilege credentials for IAM users or for users you authenticate (federate users)

- By default, STS is a global service
  - All STS requests go to a single endpoint at https://sts.amazon.aws.com (mapped US-East)
  - You can optionally send your STS to endpoints in other regions instead to reduce latency

- Advantages
  - No need to distribute long-term security credentials
  - No need to rotate or revoke when no longer needed

IAM Authentication

- You can work with IAM in any of the following ways



- With AWS CLI you can issue commands to build scripts that perform AWS tasks
- AWS SDKs are a set of tools to develop software apps within AWS
- AWS management console is a web-based user interface
  - ➢ It is the only one installed by default after sign-in

AWS CLI setup

- AWS Command Line Interface is a unified tool to manage your AWS services
- Benefits of AWS CLI
  - Easy installation
  - Support most AWS services
  - Saves time
  - Automation by scripting
- Steps to get started with AWS CLI
  1. Download, install AWS CLI & confirm installation

     ```
     aws --version
     ```
  2. Configure AWS CLI to connect to your AWS account

     ```
     aws configure
     AWS Access Key ID [None]: AKIAIFOU367FR4HSARNQ
     AWS Secret Access Key [None]: LV2e8vhFxW2p8WkGzmBtNw4jBNq4pMn27ibDATq+
     Default region name [None]: us-east-1
     Default output format [None]: json
     ```

IAM Security Credentials

- AWS provides different ways to provide secure user access to your AWS resources

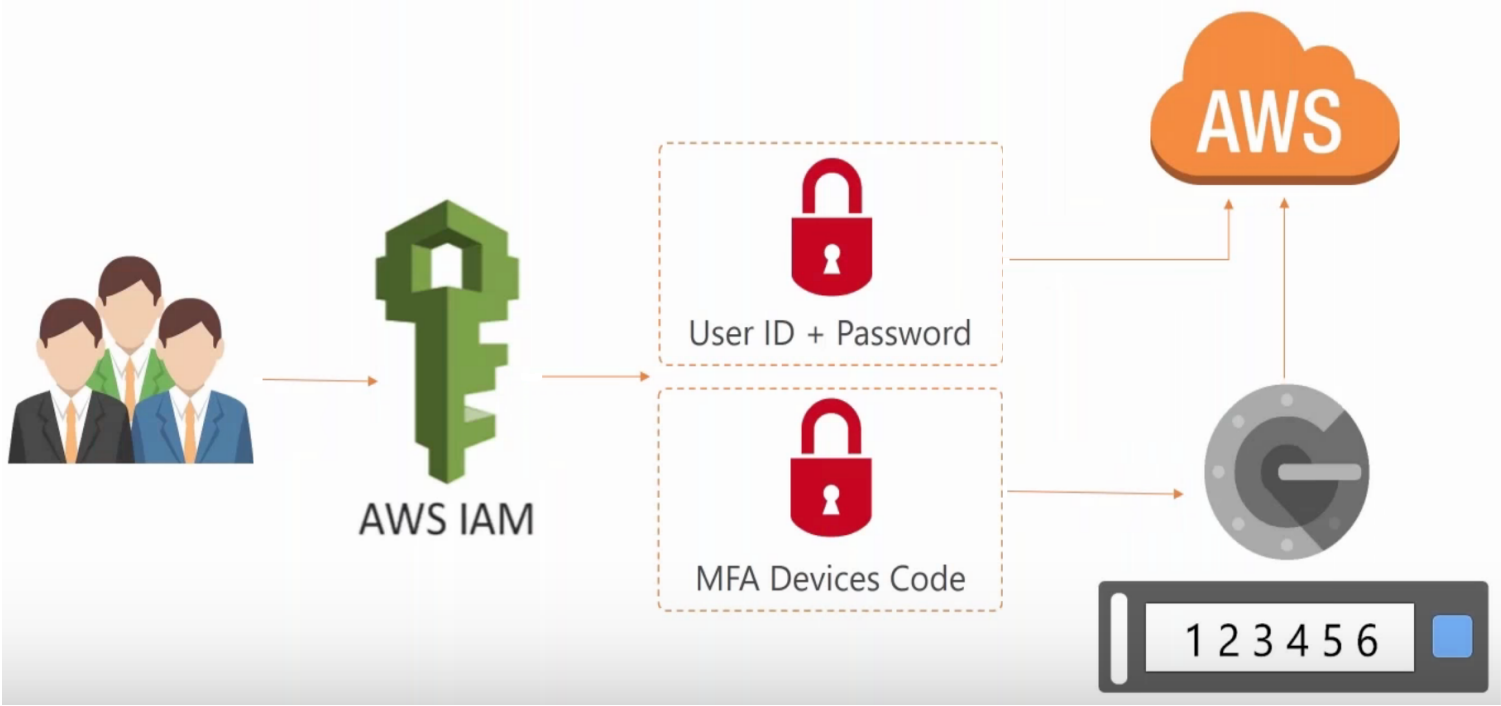
1. Email address and password (root user)
   - When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account
   - This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account
   - ➢ Best practice: use this account to create your first IAM user

2. IAM user name and password
   - IAM users provide their user names and passwords when they sign in to the AWS Management Console, AWS discussion forums, or AWS Support center
   - In some cases, an IAM user name and password are required to use a service, such as sending email with SMTP by using Amazon Simple Email Service (Amazon SES)

IAM Security Credentials

3. Multi-Factor Authentication (MFA)
   - MFA provides an extra level of security that you can apply to your AWS account
   - It is recommend on the AWS account root user and highly privileged IAM users.

   ➢ MFA is not enabled by default



With MFA enabled, when you sign in to the AWS website:

1. Prompted for your user name + password
2. Prompted for an authentication code from an MFA device.

➢ Together, they provide increased security for your AWS account settings and resources

IAM Security Credentials

4. Access keys
   - They consist of two parts (created as a set):
   - ✓ Access key Id. Ex: *AKIAIFOU367FR4HSARNQ*
   - ✓ Secret access key. Ex: *LV2e8vhFxW2p8WkGzmBtNw4jBNq4pMn27ibDATq+*
   - You use access keys to sign programmatic requests that you make to AWS
     - Using AWS CLI commands or AWS API operations
   - During access key creation, AWS give you one opportunity to view and download the key set
     - If you download it, AWS provides you with a CSV file that contains the following:

```
Access key ID, Secret access key
AKIAIFOU367FR4HSARNQ,LV2e8vhFxW2p8WkGzmBtNw4jBNq4pMn27ibDATq+
```

     - If you lose it, you can delete the access key set and create a new one
   - You are limited to 2 access key for each IAM user
   - ➢ It is recommended to rotate your access keys periodically

IAM Security Credentials

5. Key pairs
   - They consist of two parts:

✓ Private key: used to create a digital signature

✓ Public key: used by AWS to validate the signature

   - Key pairs are used:
     - To access Amazon EC2 instances. Ex/ when you use SSH to log in to a Linux instance
     - To create signed URLs for private content within Amazon CloudFront.
       - Example: when you want to distribute restricted content that someone paid for

   - AWS does not provide key pairs for your account, you must create them.
     - You can create EC2 key pairs from the Amazon EC2 console, CLI or API
     - You create CloudFront key pairs from the Security Credentials page. Only the AWS root user (not IAM users) can create CloudFront key pairs.

➢ Key pairs are unrelated to access keys

AWS Security Audit Guidelines

- AWS recommends to periodically audit your security configuration
  - To make sure it meets your current business needs
- An audit gives you an opportunity to remove unneeded:
  - ✓ Users
  - ✓ Roles
  - ✓ Groups
  - ✓ Policies
- It make sure that your users and software have only the permissions that are required

➢ Use IAM policy simulator to test policies: https://policysim.aws.amazon.com/

https://docs.aws.amazon.com/general/latest/gr/aws-security-audit-guide.html

IAM best practices

- Users
  - Create individual users
  - Users for authentication
- Groups
  - Manage permissions with groups
- Permissions
  - Grant least privilege
- Password
  - Configure strong password policy
- MFA
  - Enable MFA for privileged users
- Updates
  - Ensure security patches are up
- Roles
  - Use IAM roles for Amazon EC2 instances
  - Use IAM roles to share access
- Credentials
  - Rotate security credentials regularly
- Conditions
  - Restrict privileged access further with conditions
- Root
  - Reduce or remove use of root
- Audit
  - Periodically audit account security

Security best practices

- IAM
  - See previous best practices
- Storage
  - Keep sensible data properly encrypted
- Compute
  - Ensure security patches are up-to-date
- Networking
  - Configure nets/subnets the right way
  - Apply strict policies for security groups (firewall rules)
  - Encrypt network traffic

IAM references and documentation

- AWS IAM documentation
  - https://docs.aws.amazon.com/iam/
- IAM user guide
  - https://docs.aws.amazon.com/IAM/latest/UserGuide/
    (You can download in pdf the whole guide)
- IAM CLI reference
  - https://docs.aws.amazon.com/cli/latest/reference/iam/
- IAM API reference
  - https://docs.aws.amazon.com/IAM/latest/APIReference/

AWS WAF

Protecting web applications from common web exploits

Web Application Firewall

- AWS WAF is a web application firewall that helps protect your web applications from common attack patterns that can:
  - affect application availability
  - compromise security
  - consume excessive resources

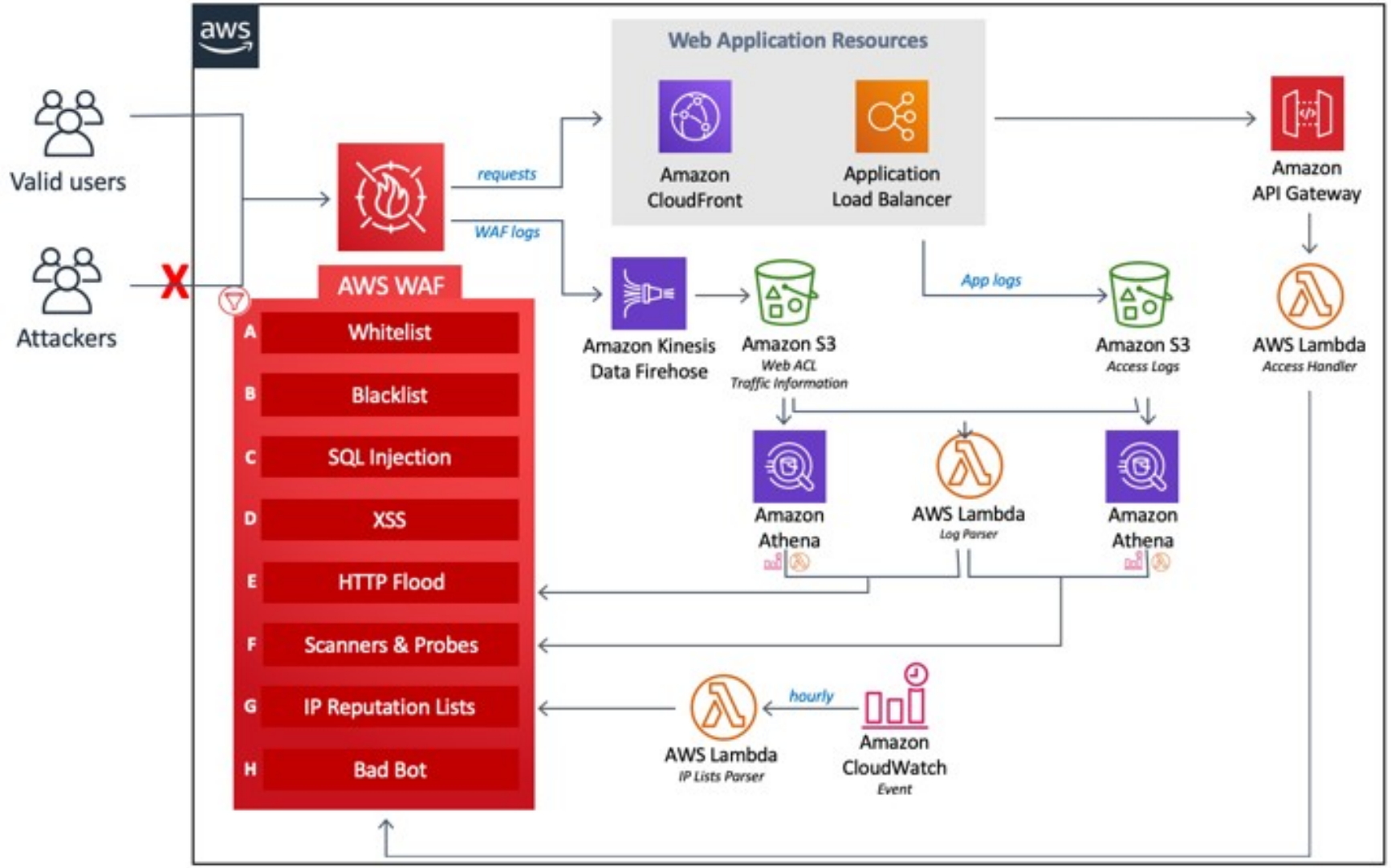
- AWS WAF gives you control over which traffic to allow or block to your web applications by defining customizable web security rules

- It can be completely administered via APIs which makes security automation easy, enabling rapid rule propagation and fast incident response

WAF Key Features

- Preconfigured template allows to quickly get started with AWS WAF
- The template includes a set of AWS WAF rules, which can be customized to best fit your needs, designed to block common web-based attacks
- Blocking IP Address that exceed request limits helps to prevent Distributed Denial of Service (DDoS) attacks
- Blocking IP Address that submit Bad Requests (4xx) rejects scan bots trying to infer what systems are running into the front servers or CloudFront
- Once the solution is deployed, AWS WAF will begin inspecting web requests to the user's existing Amazon CloudFront distributions or Application Load Balancers, and block them when applicable
- AWS CloudFormation can provide an Amazon Athena query and a scheduled AWS Lambda function responsible for orchestrating Athena executing, processing result output, and updating AWS WAF

WAF Security Automations solution architecture



WAF Security Automations solution architecture

- Manual IP lists (A and B): Creates two specific AWS WAF rules that allow you to manually insert IP addresses that you want to block (blacklist) or allow (whitelist)
- SQL Injection (C) and XSS (D): Configures two native rules that protect against common SQL injection or cross-site scripting patterns in the request URI, query string, or body
- HTTP flood (E): Protects against attacks consisting of a large number of requests from a single IP address, such as a web-layer DDoS or a brute-force login attempt
- Scanners and Probes (F): Parses application access logs searching for suspicious behavior, such as an abnormal amount of errors generated by an origin, and blocks those suspicious IP addresses for a customer-defined period of time
- IP Reputation Lists (G): The IP Lists Parser AWS Lambda function which checks third-party IP reputation lists hourly for new ranges to block
- Bad Bots (H): Automatically sets up a honeypot, which is a security mechanism intended to lure and deflect an attempted attack