



Module 3: AWS Security, Identity, and Access Management

AWS Shared Responsibility Model



Customers

Customer Applications & Content

Platform, Applications, Identity, and Access Management

Operating system, network, and firewall configuration

Client-side data
encryption

Server-side data
encryption

Network Traffic
Protection

Customers are responsible for security **IN** the cloud

AWS Foundation Services

Compute

Storage

Database

Networking

AWS Global
Infrastructure

Availability Zones

Regions

Edge
locations

AWS is responsible for the security **OF** the cloud



Physical Security

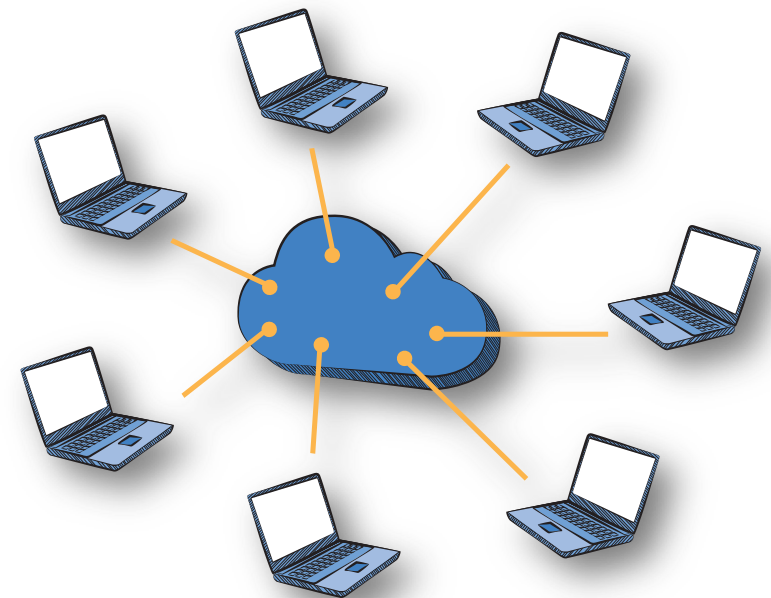


- 📦 24/7 trained security staff
- 📦 AWS data centers in nondescript and undisclosed facilities
- 📦 Two-factor authentication for authorized staff
- 📦 Authorization for data center access

Hardware, Software, and Network



- ❏ Automated change-control process
- ❏ Bastion servers that record all access attempts
- ❏ Firewall and other boundary devices
- ❏ AWS monitoring tools



Assurance Programs



Global



USA



Europe



Asia Pacific



SSL Endpoints



SSL Endpoints

Secure Transmission

Use secure endpoints to establish secure communication sessions (HTTPS).

Security Groups

Instance Firewalls

Use security groups to configure firewall rules for instances.

VPC

Network Control

Use public and private subnets, NAT, and VPN support in your virtual private cloud to create low-level networking constraints for resource access.

Security Groups



SSL Endpoints

Secure Transmission

Use secure endpoints to establish secure communication sessions (HTTPS).

Security Groups

Instance Firewalls

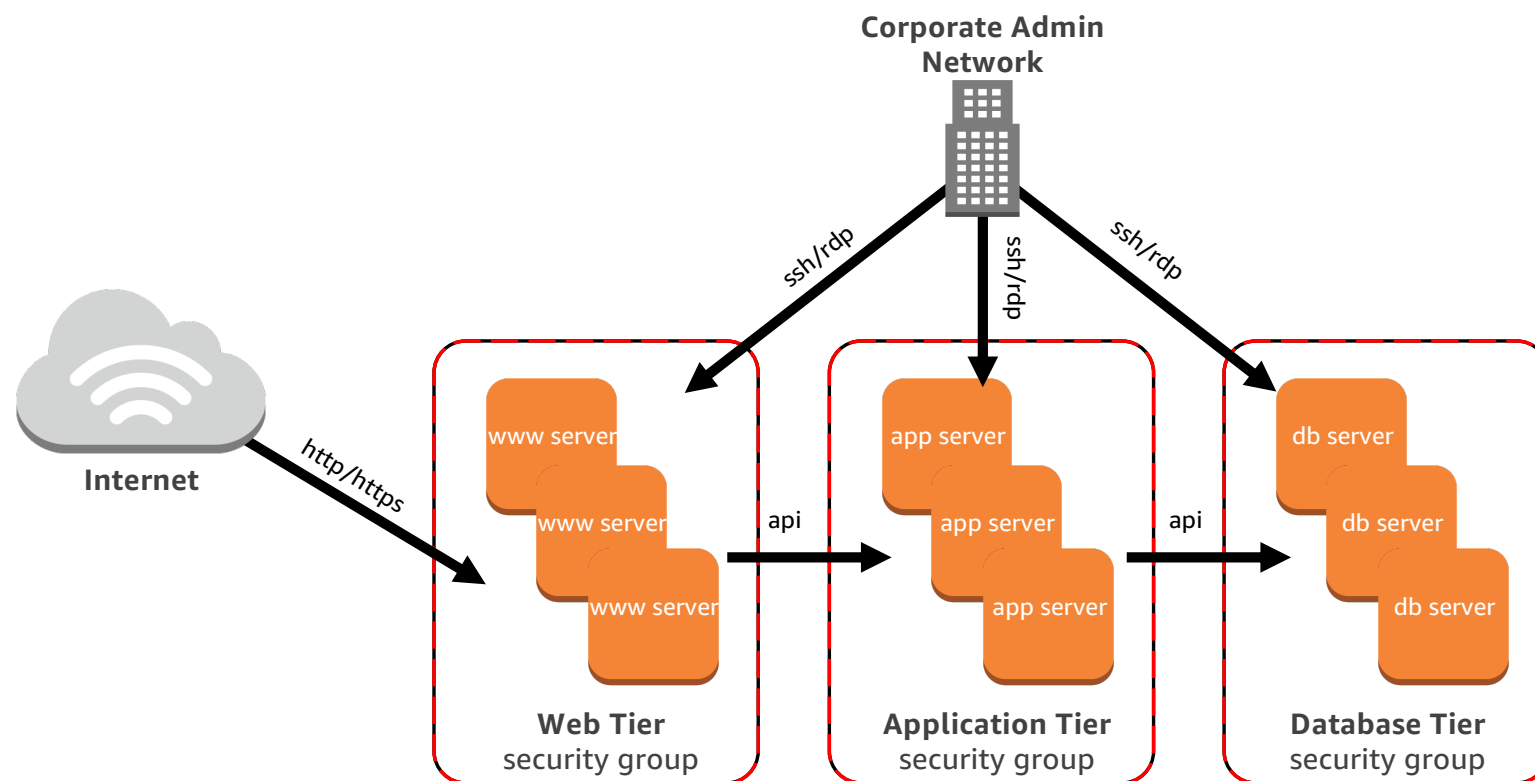
Use security groups to configure firewall rules for instances.

VPC

Network Control

Use public and private subnets, NAT, and VPN support in your virtual private cloud to create low-level networking constraints for resource access.

AWS Multi-Tier Security Groups



(all other ports are blocked)

Amazon Virtual Private Cloud



SSL Endpoints

Secure Transmission

Use secure endpoints to establish secure communication sessions (HTTPS).

Security Groups

Instance Firewalls

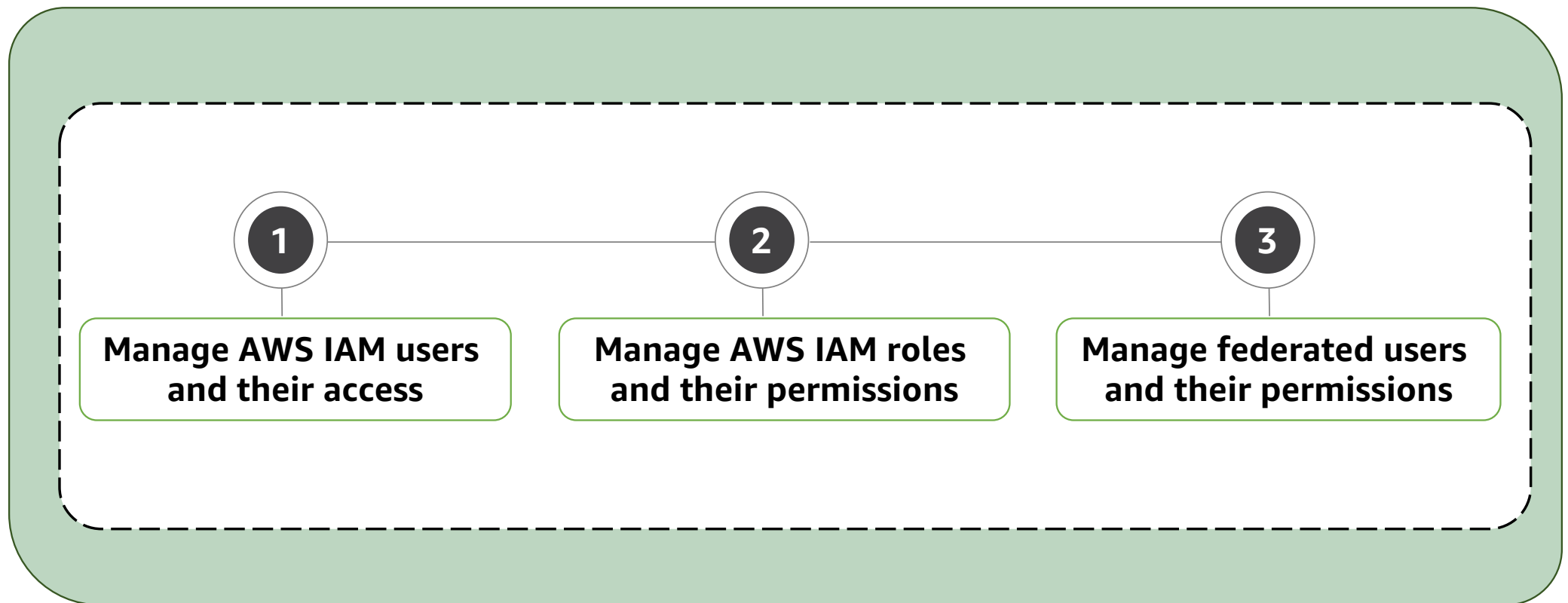
Use security groups to configure firewall rules for instances.

VPC

Network Control

Use public and private subnets, NAT, and VPN support in your virtual private cloud to create low-level networking constraints for resource access.

AWS Identity and Access Management



AWS IAM Authentication



Authentication

AWS Management Console

User name and Password



IAM User

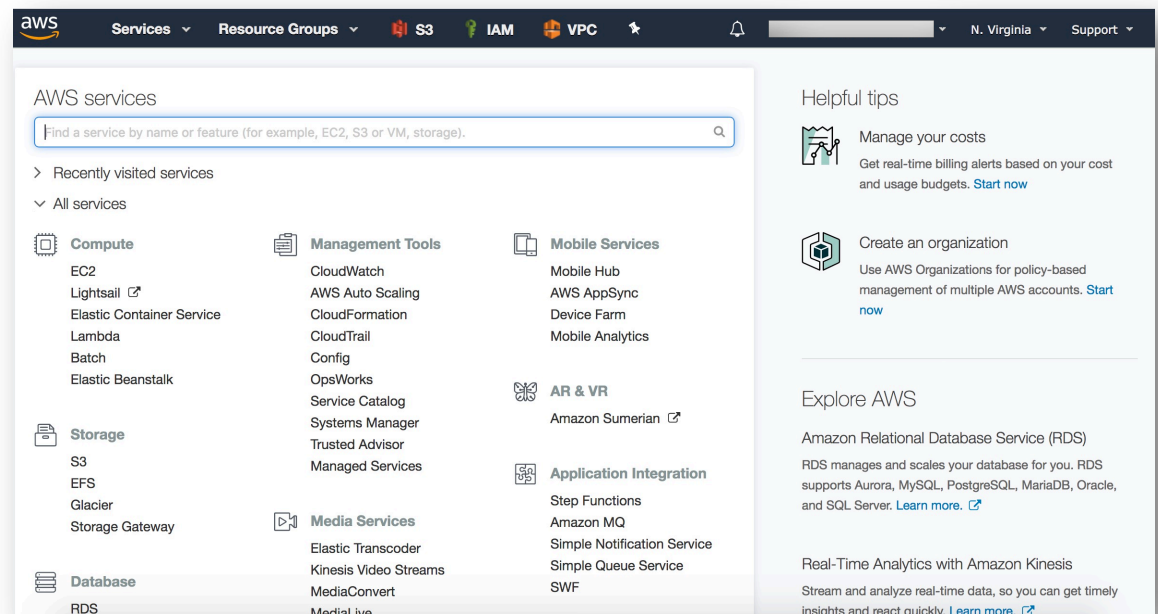
Account ID or alias

IAM user name

Password

Sign In

[Sign-in using root account credentials](#)



AWS IAM Authentication



Authentication

AWS CLI or SDK API

Access Key and Secret Key



IAM User

Access Key ID: AKIAIOSFODNN7EXAMPLE
Secret Access Key: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY

AWS CLI

```
~$ aws configure
AWS Access Key ID [*****O22A]:
AWS Secret Access Key [*****4m8i]:
Default region name [ap-southeast-1]:
Default output format [json]:
```

AWS SDK & API



Java

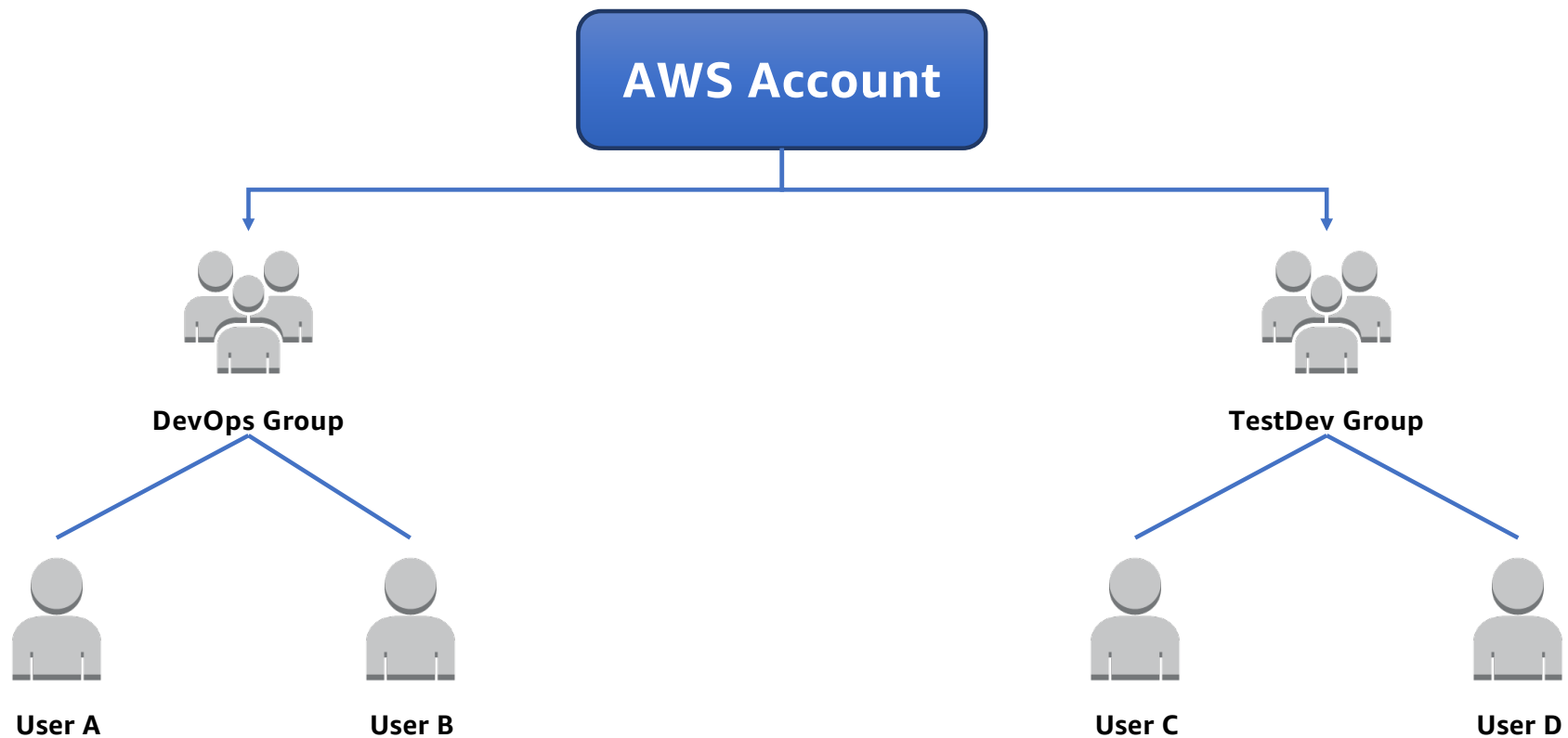


Python



.NET

AWS IAM User Management - Groups



AWS IAM Authorization



Authorization

Policies:

- Are JSON documents to describe permissions.
- Are assigned to users, groups, or roles.



IAM User



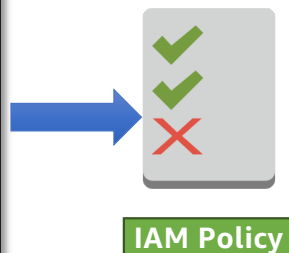
IAM Group



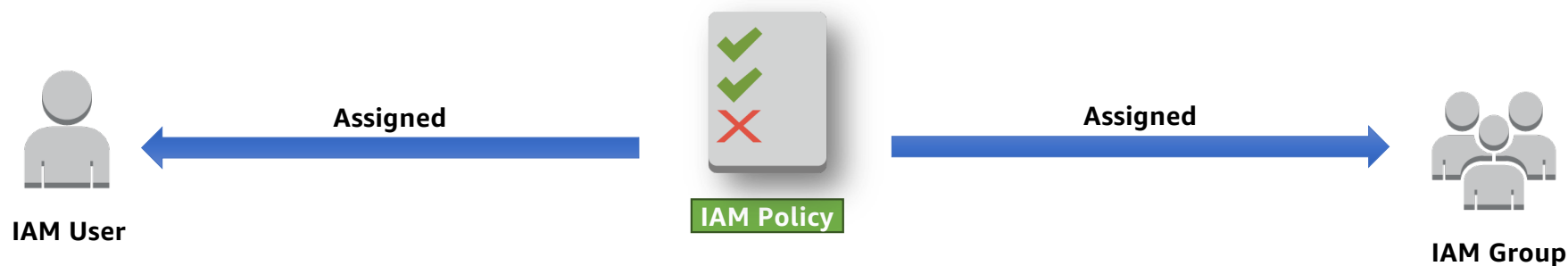
IAM Roles

AWS IAM Policy Elements

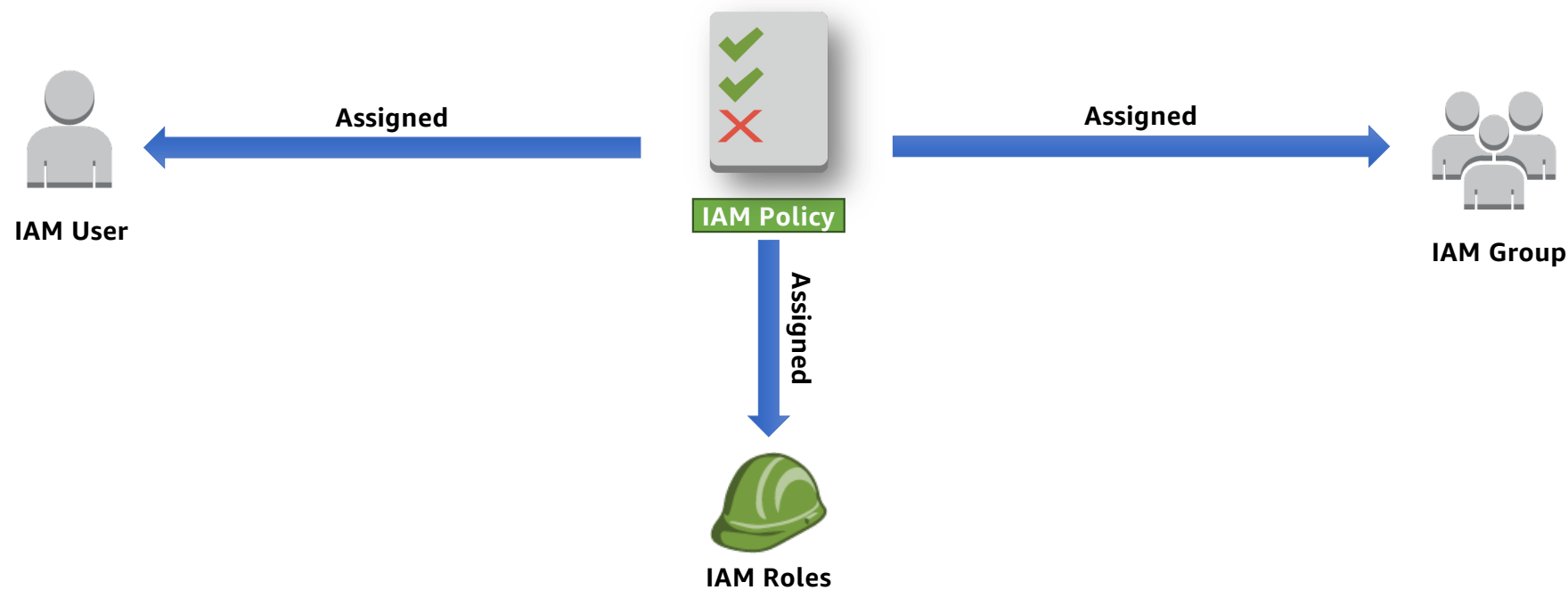
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1453690971587",
      "Action": [
        "ec2:Describe*",
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "54.64.34.65/32"
        }
      }
    },
    {
      "Sid": "Stmt1453690998327",
      "Action": [
        "s3:GetObject*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::example_bucket/*"
    }
  ]
}
```



AWS IAM Policy Assignment



AWS IAM Policy Assignment



AWS IAM Roles

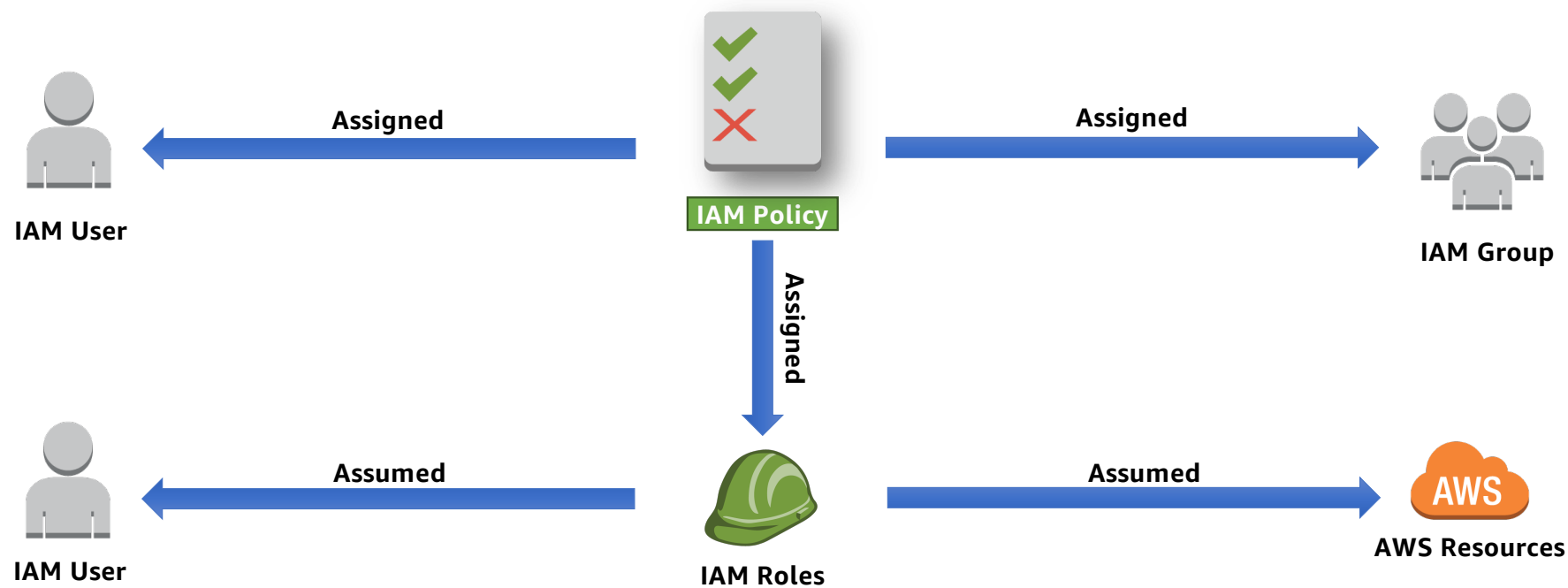


- ❏ An IAM role uses a policy.
- ❏ An IAM role has no associated credentials.
- ❏ IAM users, applications, and services may assume IAM roles.



IAM Roles

AWS IAM Policy Assignment



Example: Application Access to AWS Resources



- Python application hosted on an Amazon EC2 instance needs to interact with Amazon S3.
- AWS credentials are required:
 - ~~Option 1: Store AWS Credentials on the Amazon EC2 instance.~~
 - Option 2: Securely distribute AWS credentials to AWS services and applications.



IAM Roles

AWS IAM Roles – Instance Profiles



Amazon EC2



1

Create instance

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 Launch into Auto Scaling Group

Purchasing option: ☐ Request Spot instances

Network: vpc-5a1b1c1d (172.31.0.0/16) (default) Create new VPC

Subnet: No preference (default subnet in any Availability Zone) Create new subnet

Auto-assign Public IP: Use subnet setting (Enable)

Domain join directory: None Create new directory

IAM role: **PythonInEC2AccessS3** Create new IAM role

Shutdown behavior: ☐ Enable CloudWatch detailed monitoring Additional charges apply.

Enable termination protection: ☐ Enable termination protection Additional charges apply.

Monitoring: ☐ Enable CloudWatch detailed monitoring Additional charges apply.

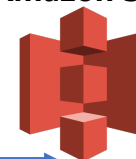
Tenancy: Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.

Advanced Details

2

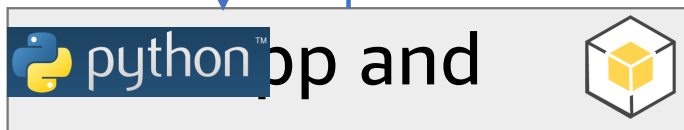
Select IAM Role

Amazon S3



4

Application interacts with Amazon S3

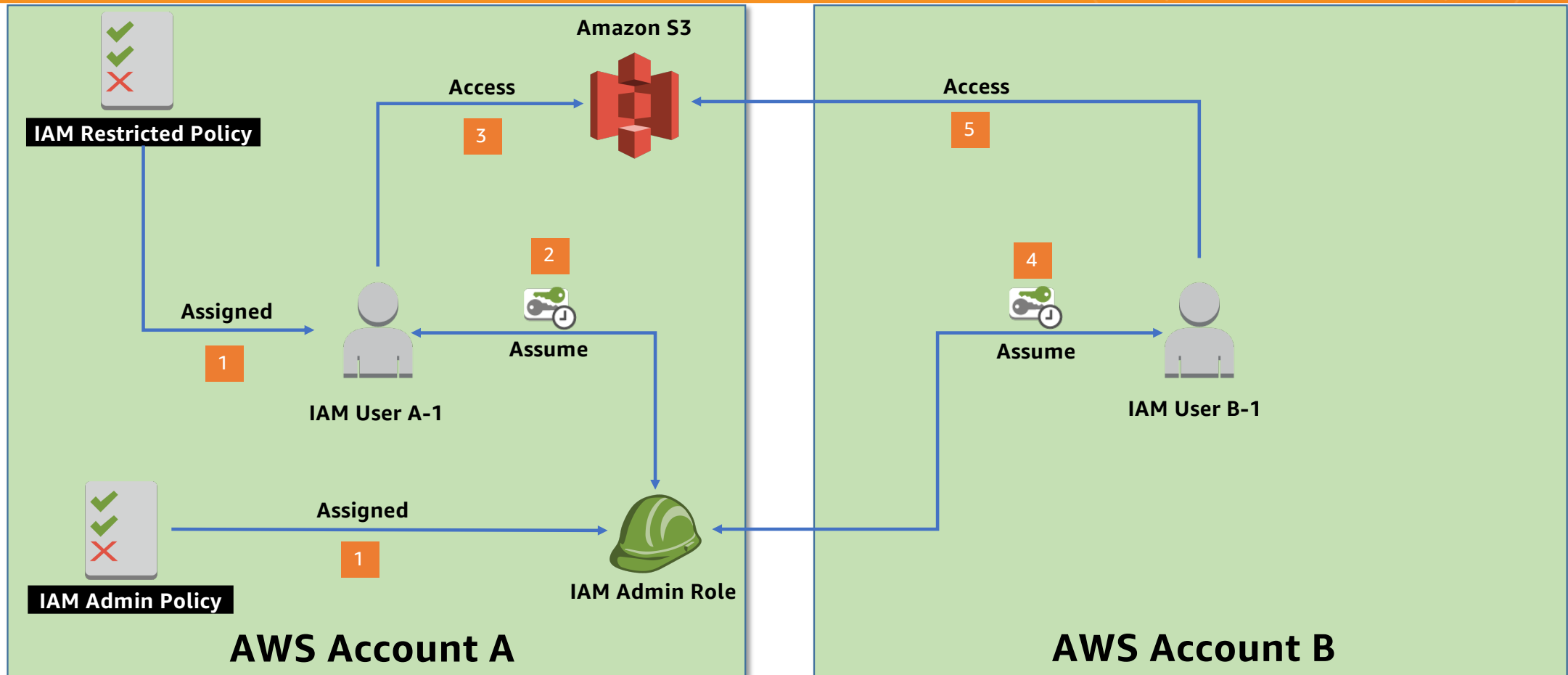


3

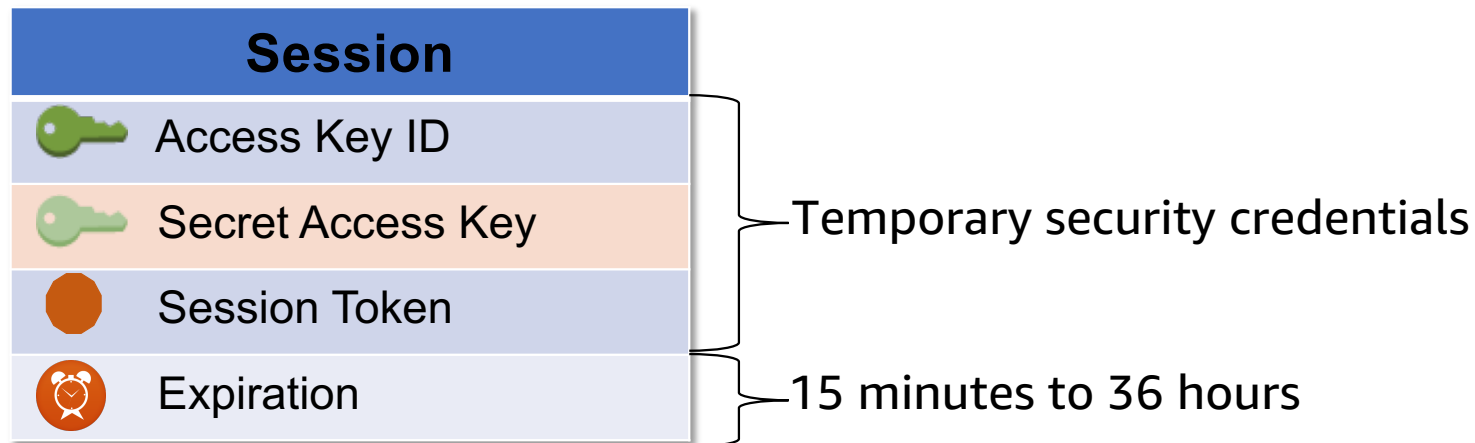
EC2 MetaData Service

<http://169.254.169.254/latest/meta-data/iam/security-credentials/rolename>

AWS IAM Roles – Assume Role



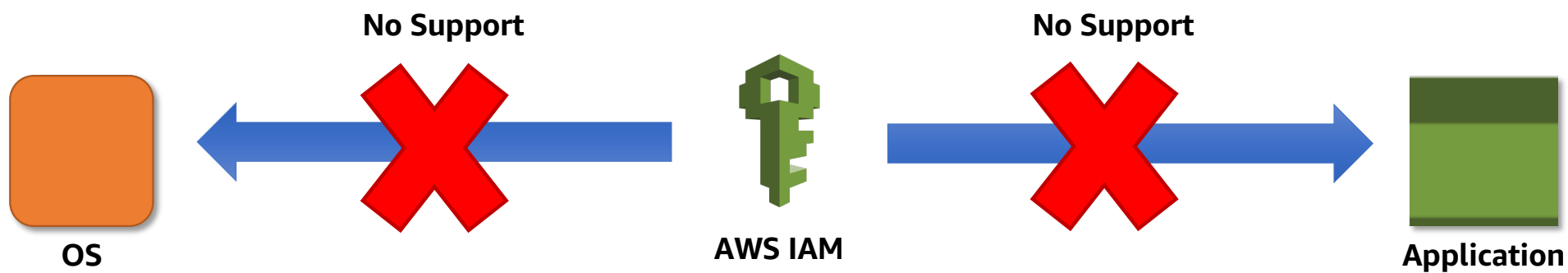
Temporary Security Credentials (AWS STS)



Use Cases

- 📦 Cross-account access
- 📦 Federation
- 📦 Mobile users
- 📦 Key rotation for Amazon EC2-based apps

Application Authentication



AWS IAM Authentication and Authorization



Authentication

-  AWS Management Console

 -  User Name and Password

-  AWS CLI or SDK API

 -  Access Key and Secret Key

Authorization

-  Policies



IAM User



IAM Group



IAM Roles

AWS IAM Best Practices



- ❏ Delete AWS account (root) access keys.
- ❏ Create individual IAM users.
- ❏ Use groups to assign permissions to IAM users.
- ❏ Grant least privilege.
- ❏ Configure a strong password policy.
- ❏ Enable MFA for privileged users.



AWS IAM Best Practices (continued)

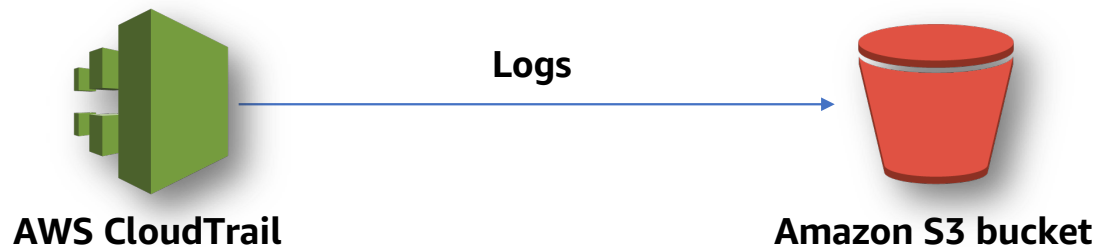


- ❏ Use roles for applications that run on Amazon EC2 instances.
- ❏ Delegate by using roles instead of by sharing credentials.
- ❏ Rotate credentials regularly.
- ❏ Remove unnecessary users and credentials.
- ❏ Use policy conditions for extra security.
- ❏ Monitor activity in your AWS account.

AWS CloudTrail



- Records AWS API calls for accounts.
- Delivers log files with information to an Amazon S3 bucket.
- Makes calls using the AWS Management Console, AWS SDKs, AWS CLI, and higher-level AWS services.



Knowledge Check



❏ Q: Your web application needs to read/write an Amazon DynamoDB table and an Amazon S3 bucket. This operation requires AWS credentials and authorization to use AWS services. What IAM entity should be used?

- ❏ User
- ❏ Group
- ❏ Role
- ❏ Policy

❏ A: **Role**

Demo: AWS IAM



Thanks for participating!

© 2018 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: aws-course-feedback@amazon.com. For all other questions, contact us at: <https://aws.amazon.com/contact-us/aws-training/>. All trademarks are the property of their owners.

