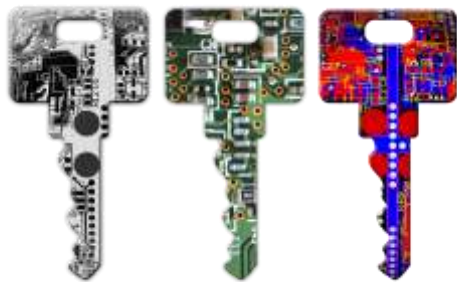


Module 3

Security, Identity, and Access Management

Physical & Environmental Security

- Lock your data center.
- Only provide access to those who need it.
- Keep track of access.
- Mount servers on racks with locks.
- Have redundant utilities.
- Build your data center with security in mind.



Network Security

- Identification & Authentication
- Firewalls
- Patching
- Virus Protection
- Encryption

Shared Responsibility – AWS

Customer

Customer Data

Platform, Applications, Identity and Access Management

Operating System, Network and Firewall Configuration

**Client-side Data Encryption
and Data Integrity
Authentication**

**Server-side Encryption
(File System and/or Data)**

**Network Traffic Protection
(Encryption/Integrity/Identity)**

AWS

Foundation Services

Compute

Storage

Database

Network

**AWS Global
Infrastructure**

Availability Zones

Regions

**Edge
Locations**

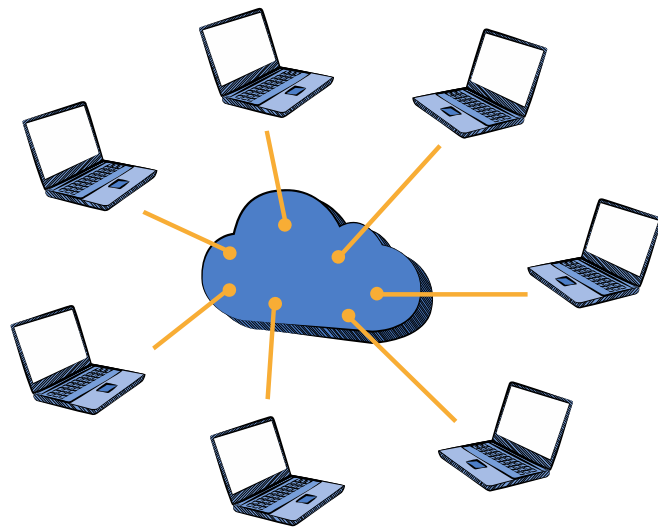
Physical Security

- 📦 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



Certifications and Accreditations

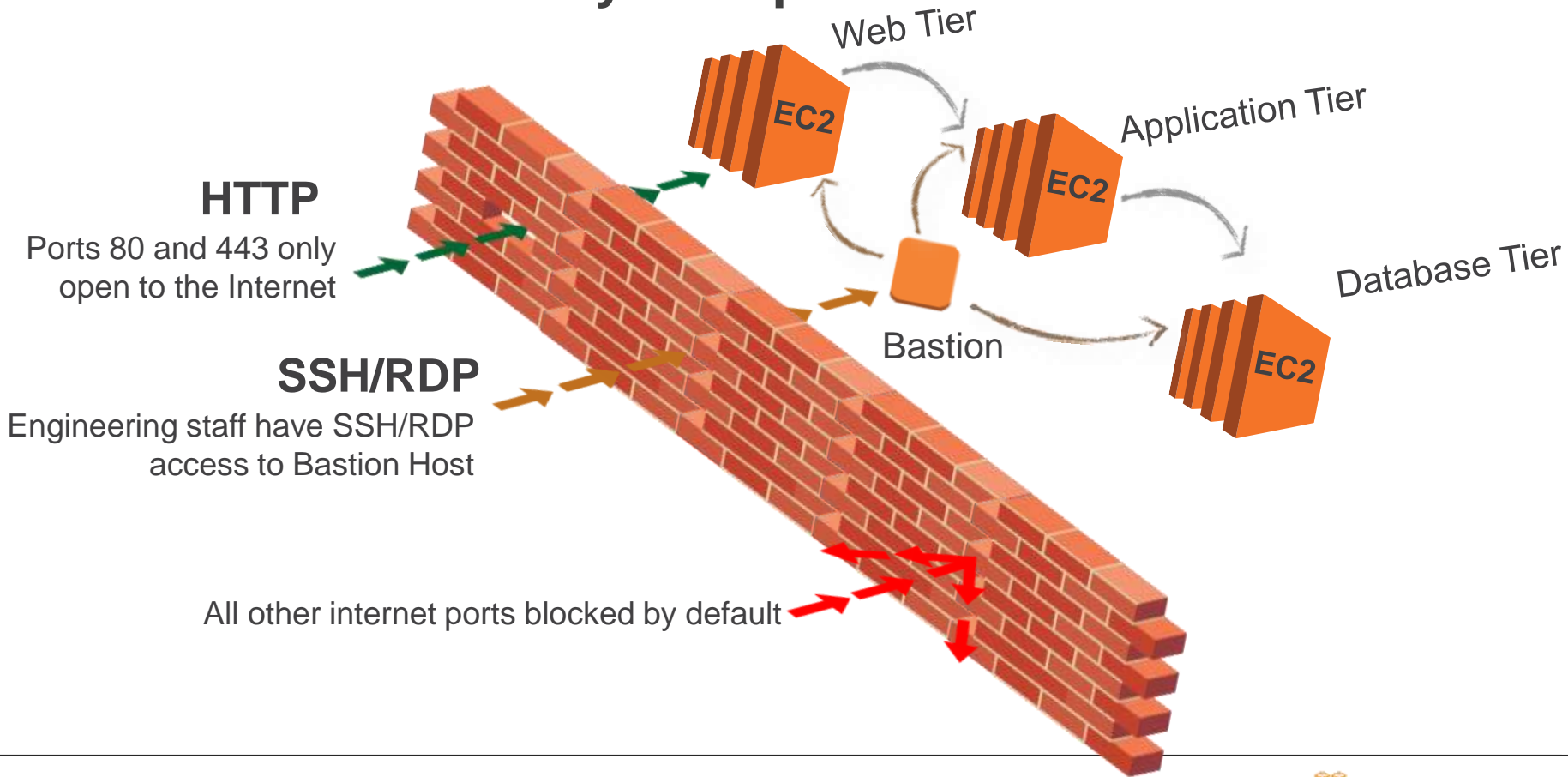


ISO 9001, ISO 27001, ISO 27017, ISO 27018, IRAP (Australia), MLPS Level 3 (China), MTCS Tier 3 Certification (Singapore) and more ...

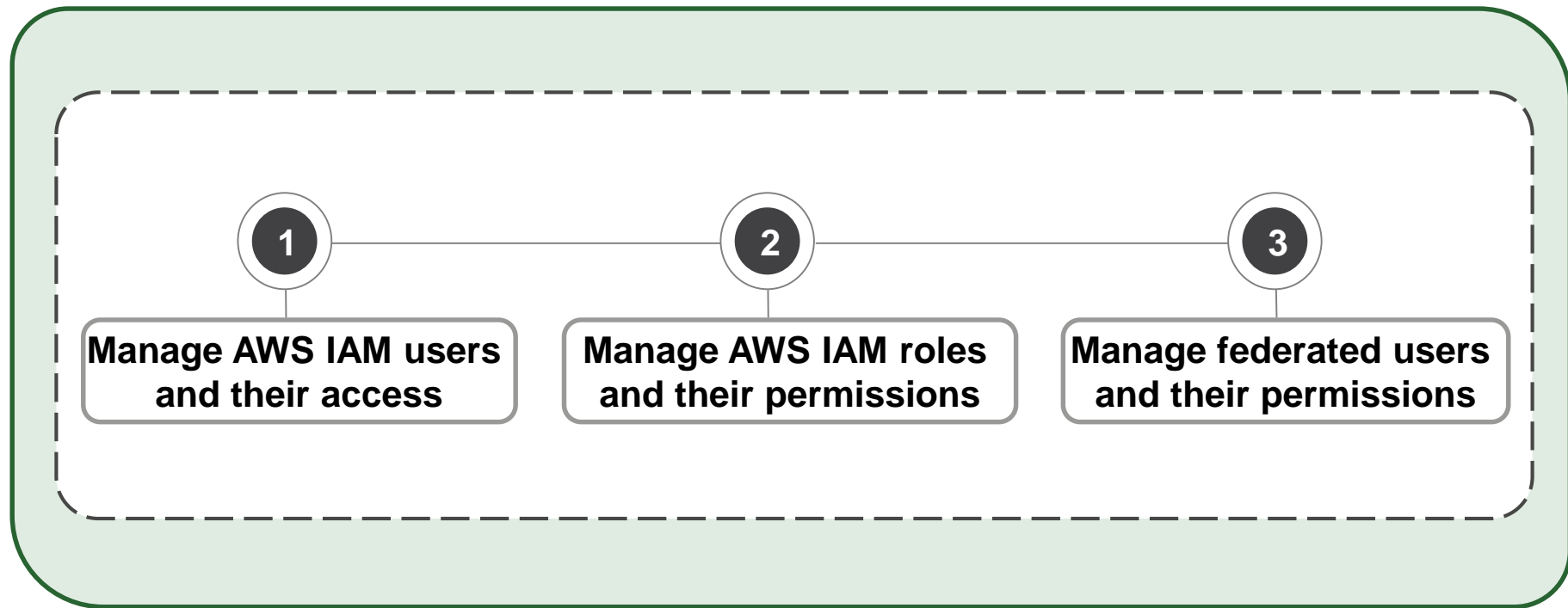
SSL Endpoints

SSL Endpoints	Security Groups	VPC
Secure Transmission Establish secure communication sessions (HTTPS) using SSL/TLS.	Instance Firewalls Configure firewall rules for instances using Security Groups.	Network Control In your Virtual Private Cloud, create low-level networking constraints for resource access. Public and private subnets, NAT and VPN support.

AWS Multi-Tier Security Groups



AWS Identity and Access Management (IAM)



AWS IAM Authentication



Authentication

AWS Management Console

➤ User Name and Password



IAM User

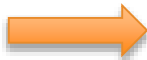
Account:

User Name:

Password:

MFA users, enter your code on the next screen.

Sign In



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

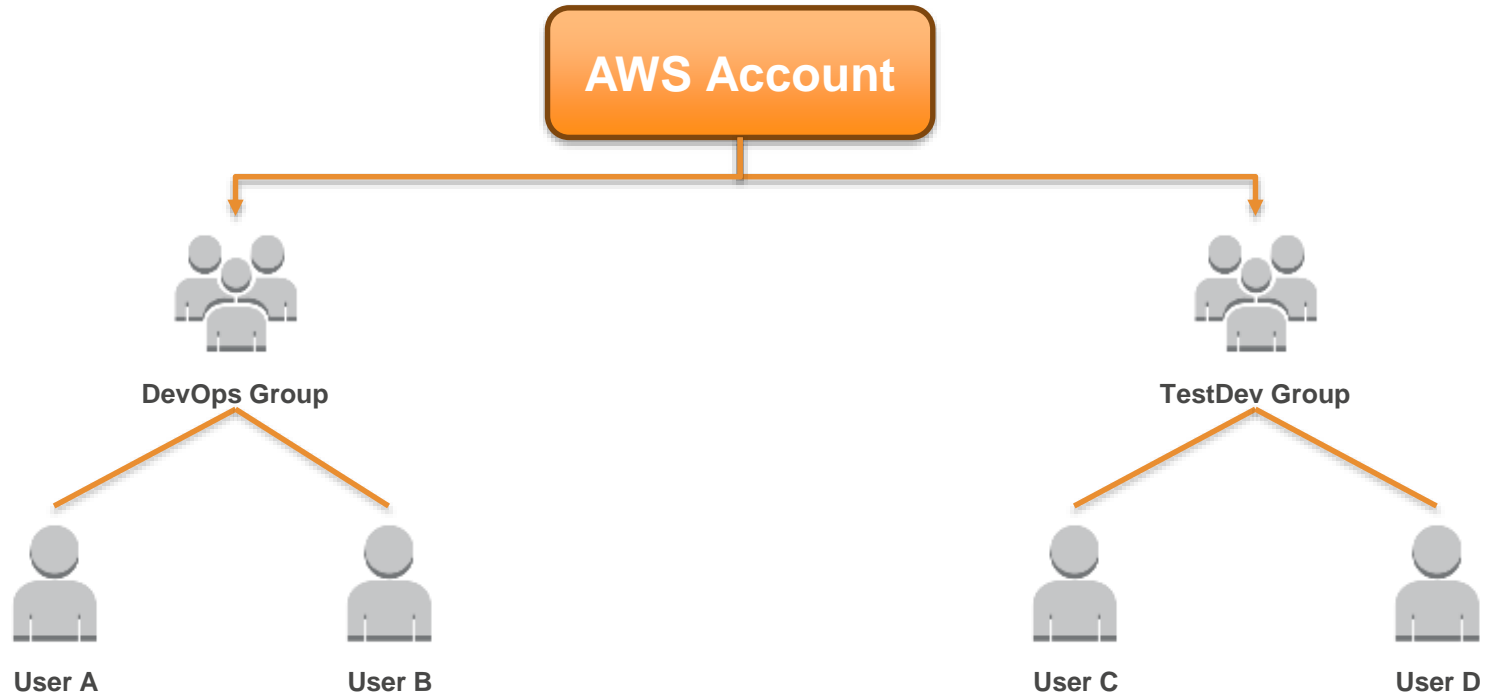


Python



.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 Roles - Instance Profiles



Amazon EC2



1

Create Instance

Select IAM Role

2

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: ☐ On-Demand Instances ☐ Request Spot Instances

Network: Create new VPC

Subnet: Create new subnet

Auto-assign Public IP:

Domain join directory: Create new directory

IAM role: Create new IAM role

Shutdown behavior:

Enable termination protection:

Monitoring: ☐ Enable CloudWatch detailed monitoring Additional charges apply

Tenancy: Additional charges will apply for dedicated tenancy

Amazon S3



Application interacts with S3

4



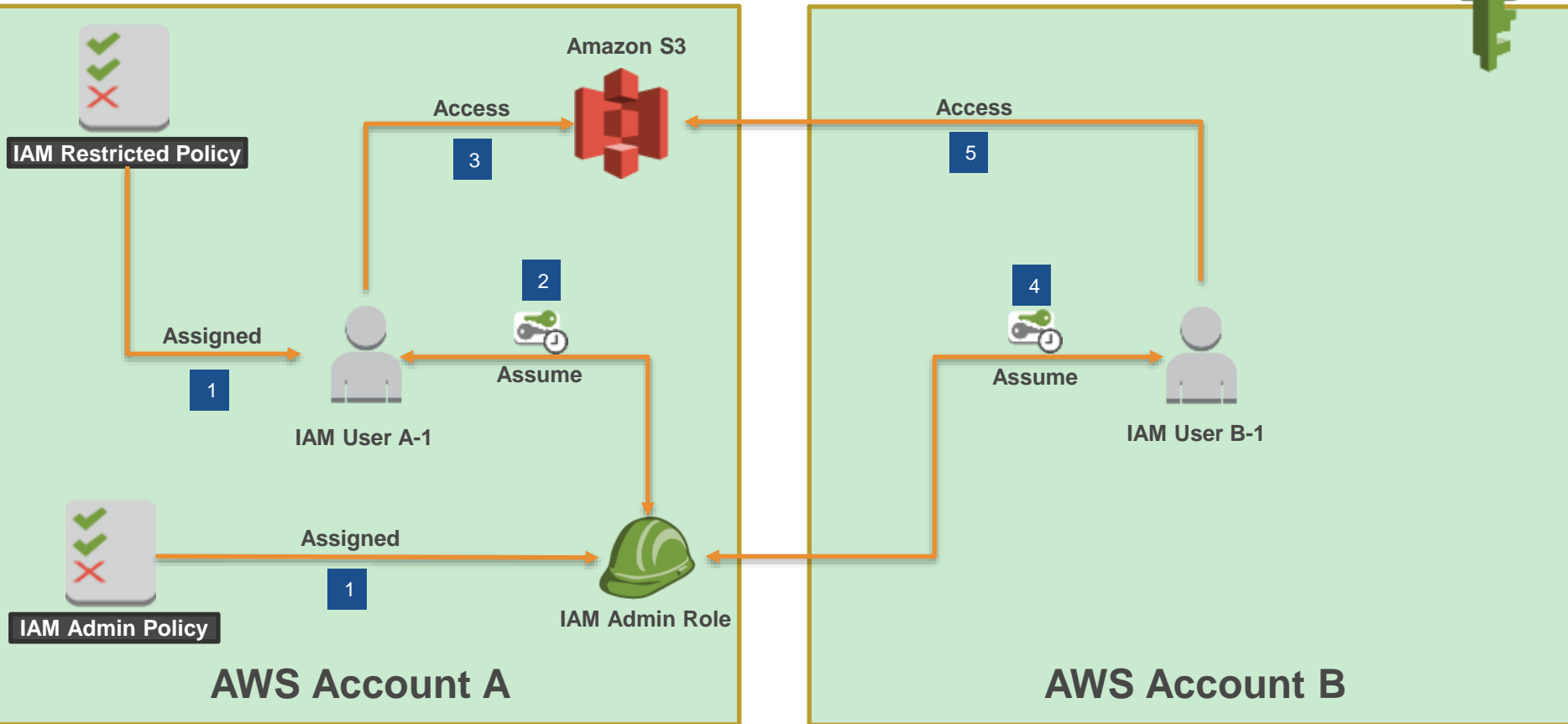
App &



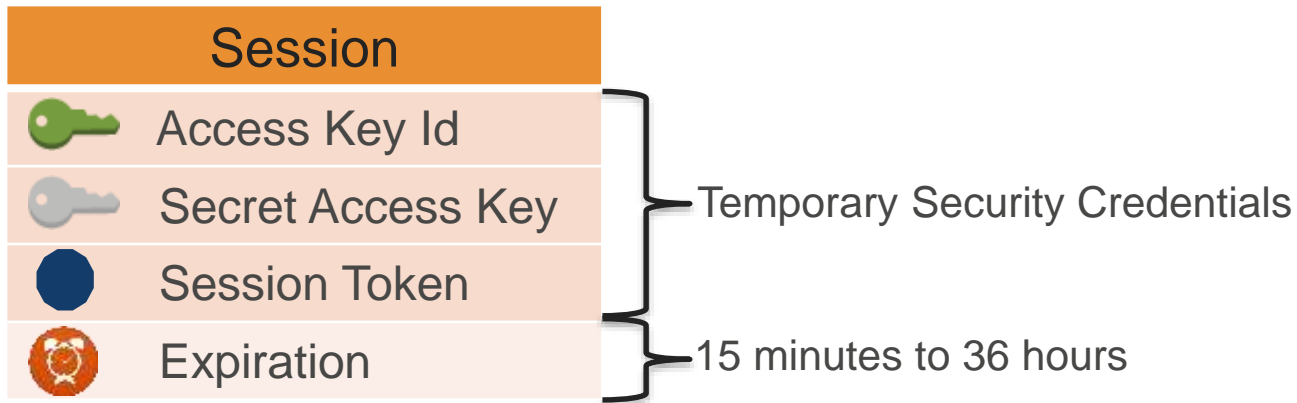
3

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)

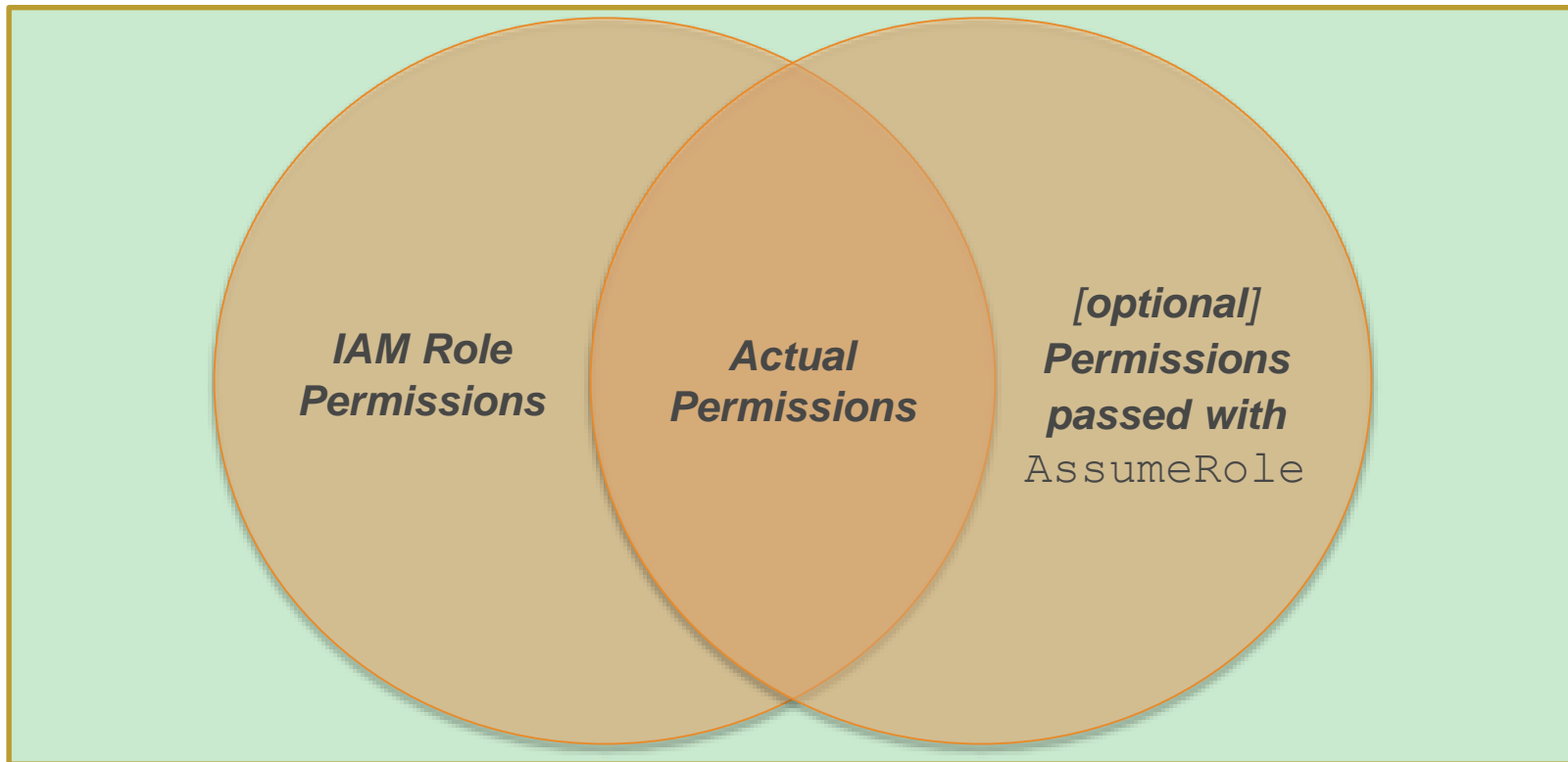


Use Cases

-  Cross account access
-  Federation

-  Mobile Users
-  Key rotation for Amazon EC2-based apps

sts:AssumeRole



AWS IAM Federation



❏ IAM federation may be used for federated access to:

- AWS Management Console
- AWS APIs

❏ Supported Identities:

- AWS Directory Service
- Microsoft Active Directory
- OpenID Connect (OIDC) such as Amazon Cognito and Login with Amazon
- SAML 2.0



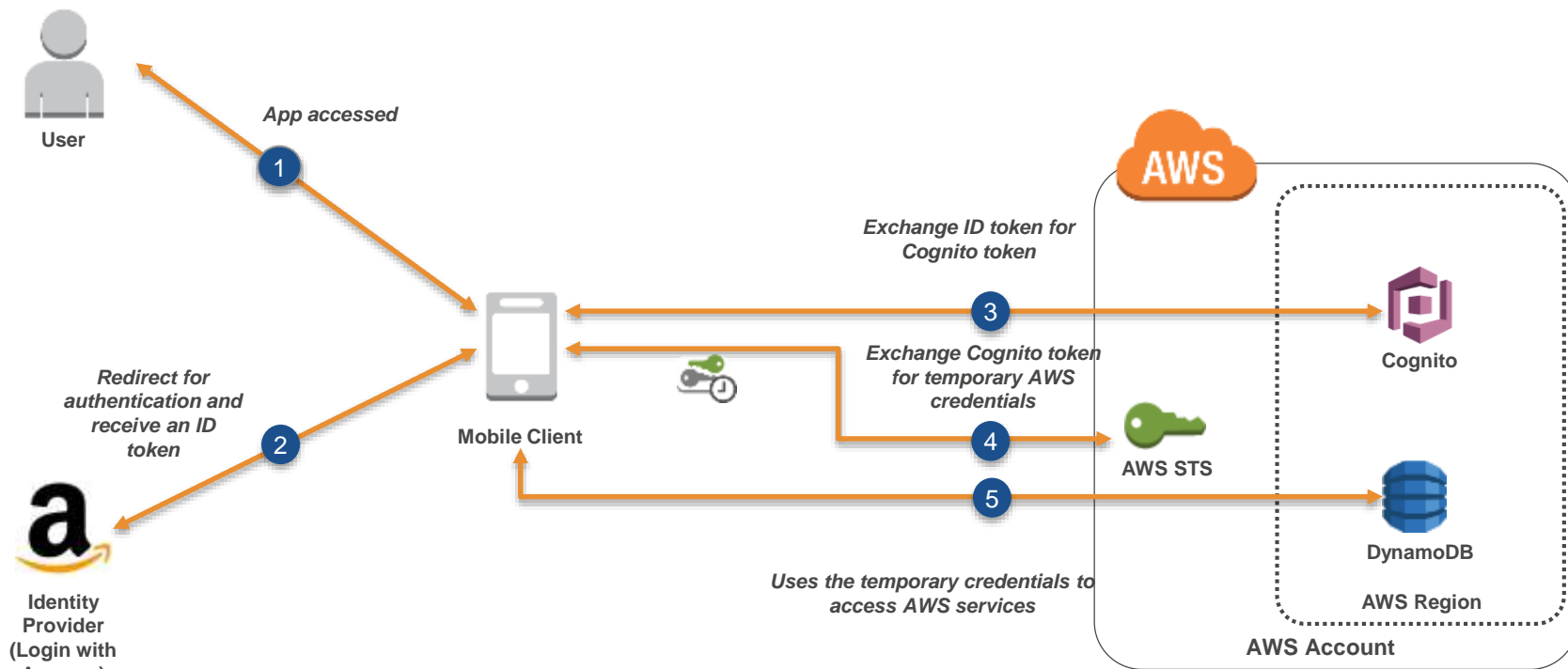
AWS Directory Service



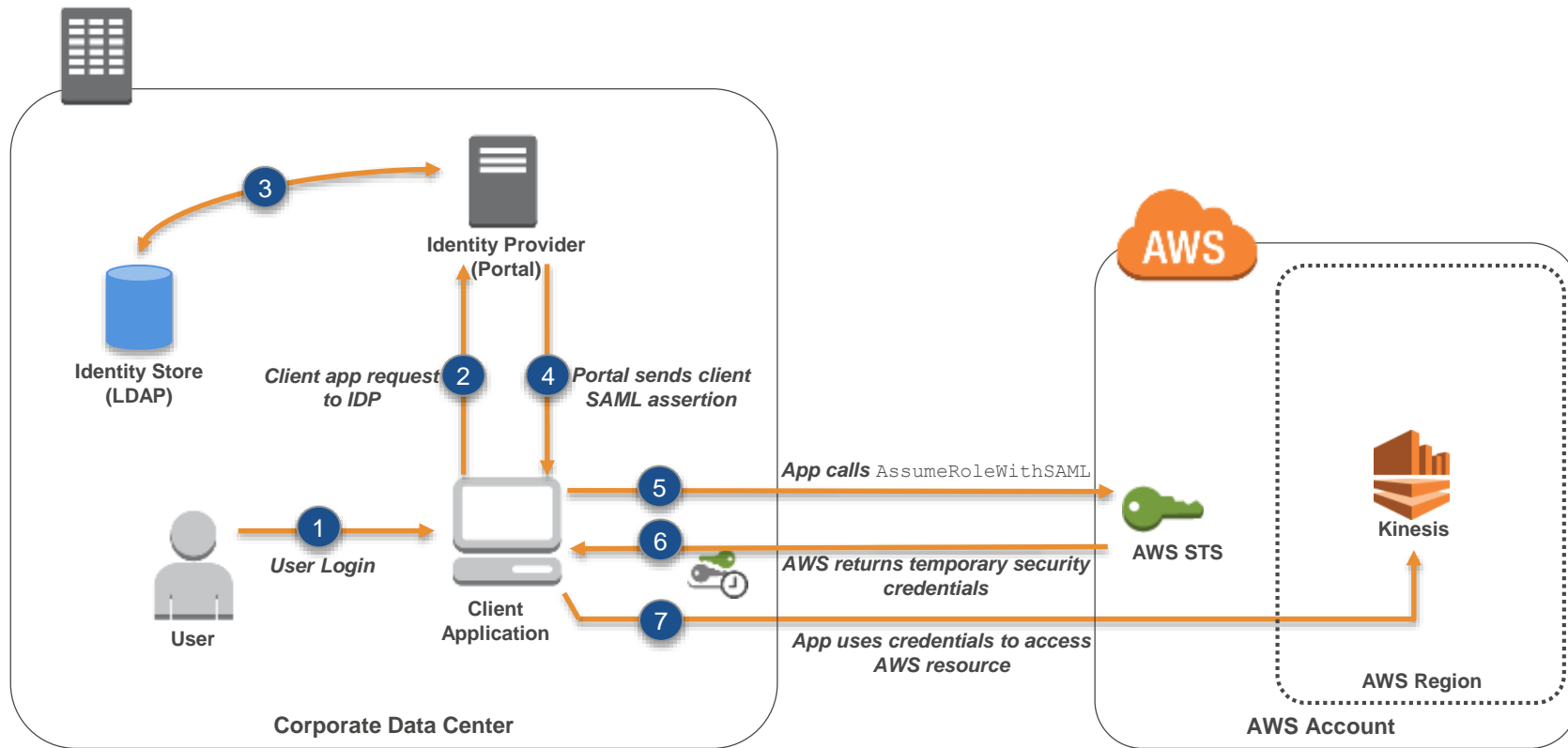
Amazon Cognito



Amazon Cognito Federation for Mobile Applications



AWS IAM Federation using SAML 2.0



Application Authentication



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 (cont.)



- 📦 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 Resource-Based Policies

- 📦 Are an alternative to IAM and supported by some services.
- 📦 Grant cross-account access to your resources.
- 📦 Use a principal to uniquely identify account in the policy.
- 📦 Supported AWS services include :
 - Amazon S3 Bucket Policy
 - Amazon SNS Topic Policy
 - Amazon SQS Queue Policy
 - Amazon Glacier Vault Policy
 - AWS OpsWorks Stack Policy
 - AWS Lambda Function Policy

© 2016 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.

Errors or corrections? 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.