



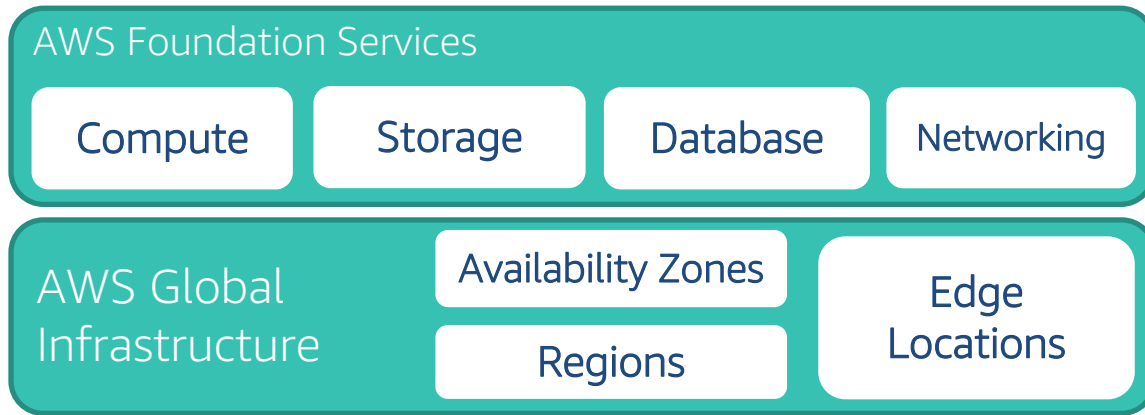
AWSOME DAY

ONLINE CONFERENCE



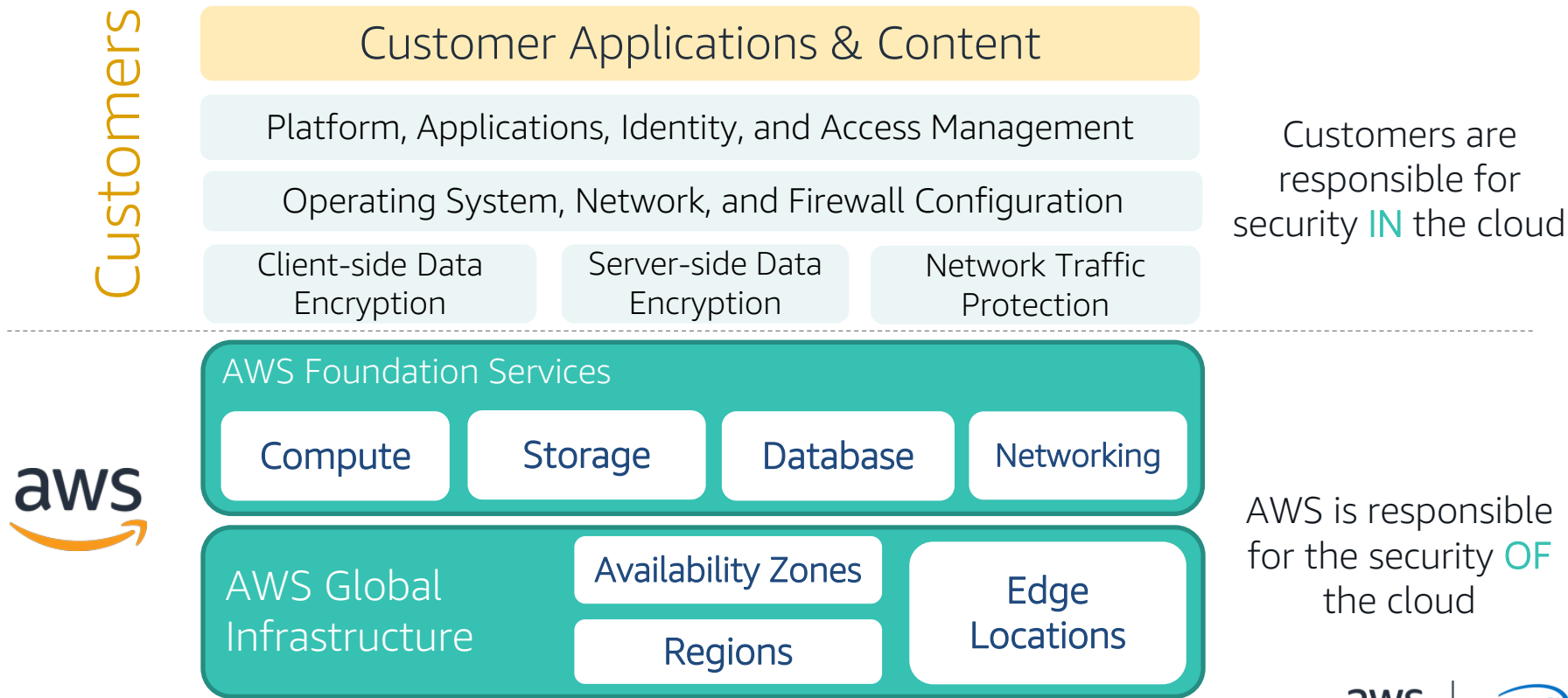
Module 3 Security, Identity, and Access Management

AWS Shared Responsibility Model



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

AWS Shared Responsibility Model



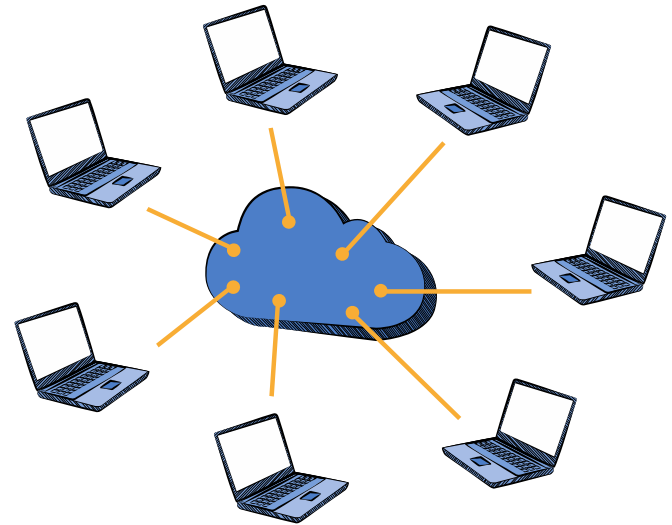
Physical Security

- 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

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

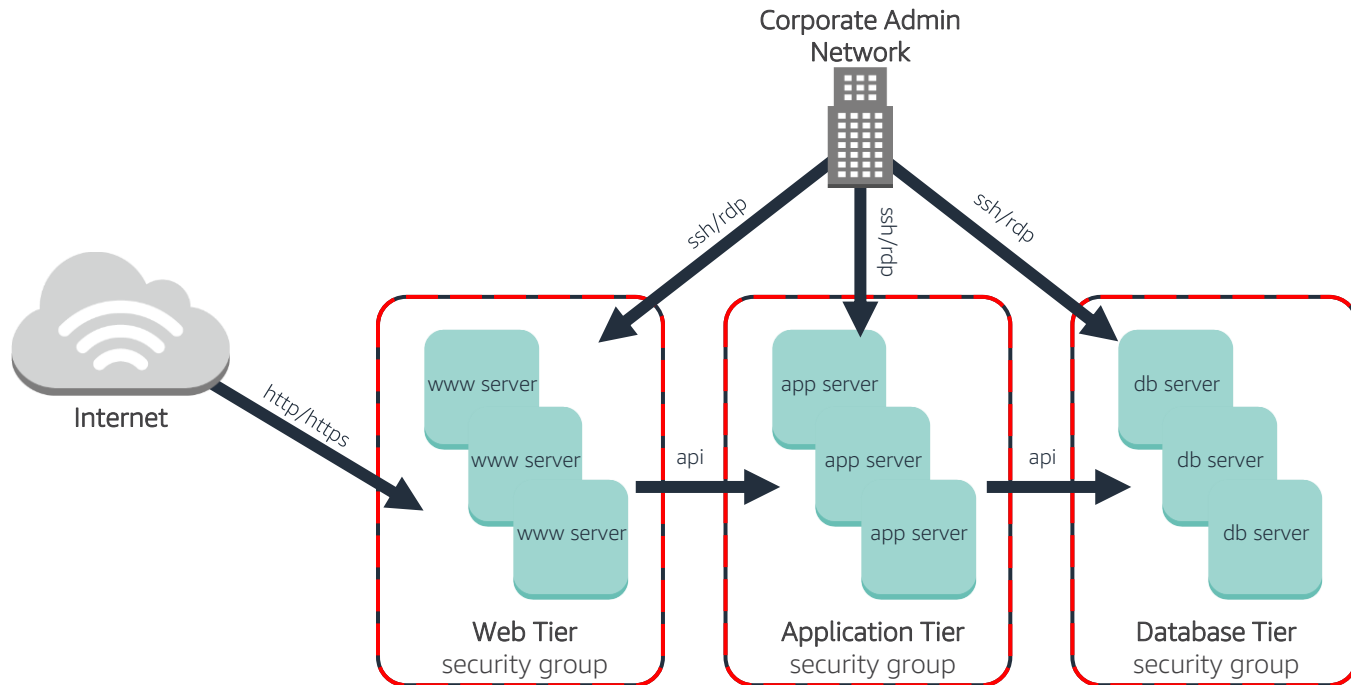
Use security groups to configure firewall rules for instances.

VPC

Network Control

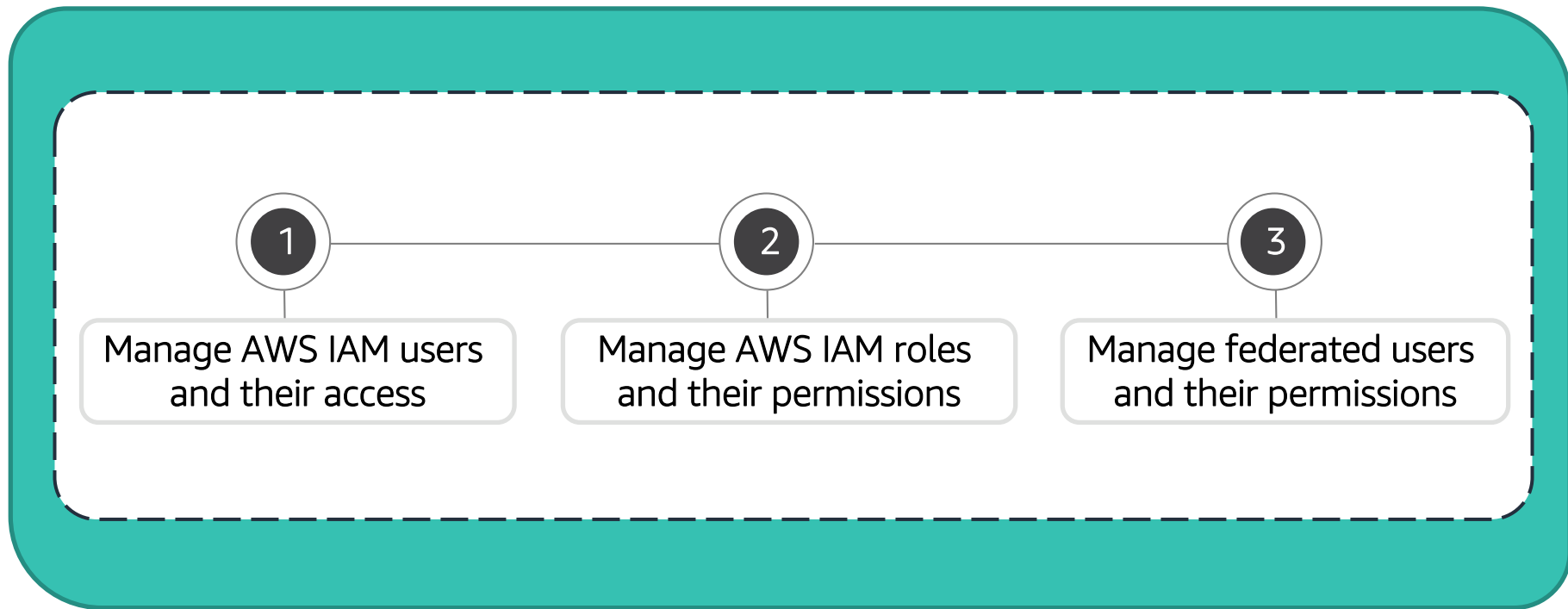
Use public and private subnets, NAT, and VPN support in your virtual private cloud to create low-level networking constraints for resource access.

AWS Multi-Tier Security Groups



(all other ports are blocked)

AWS Identity and Access Management (IAM)



AWS IAM Authentication

- Authentication
- AWS Management Console
 - User Name and Password



IAM User



Account:

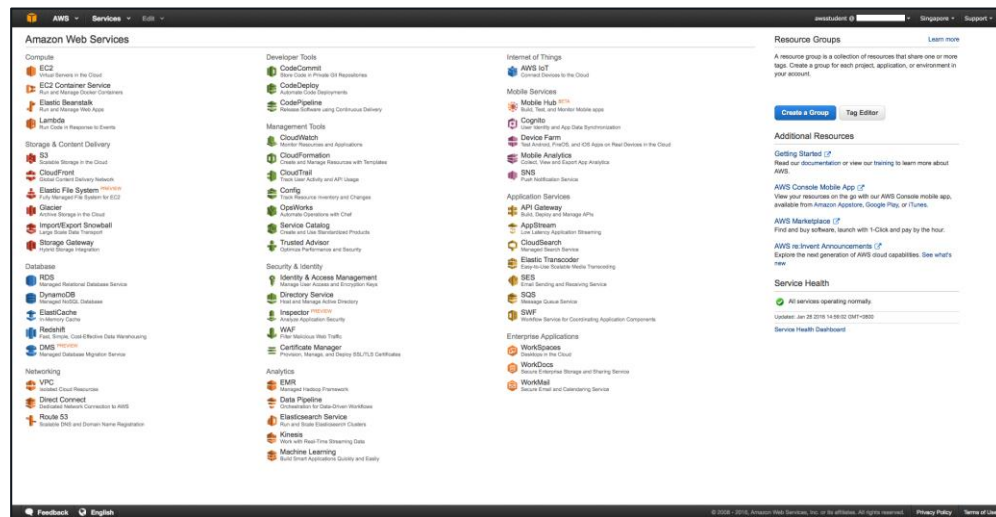


User Name:

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/bPxrRfICYEXAMPLEKEY

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

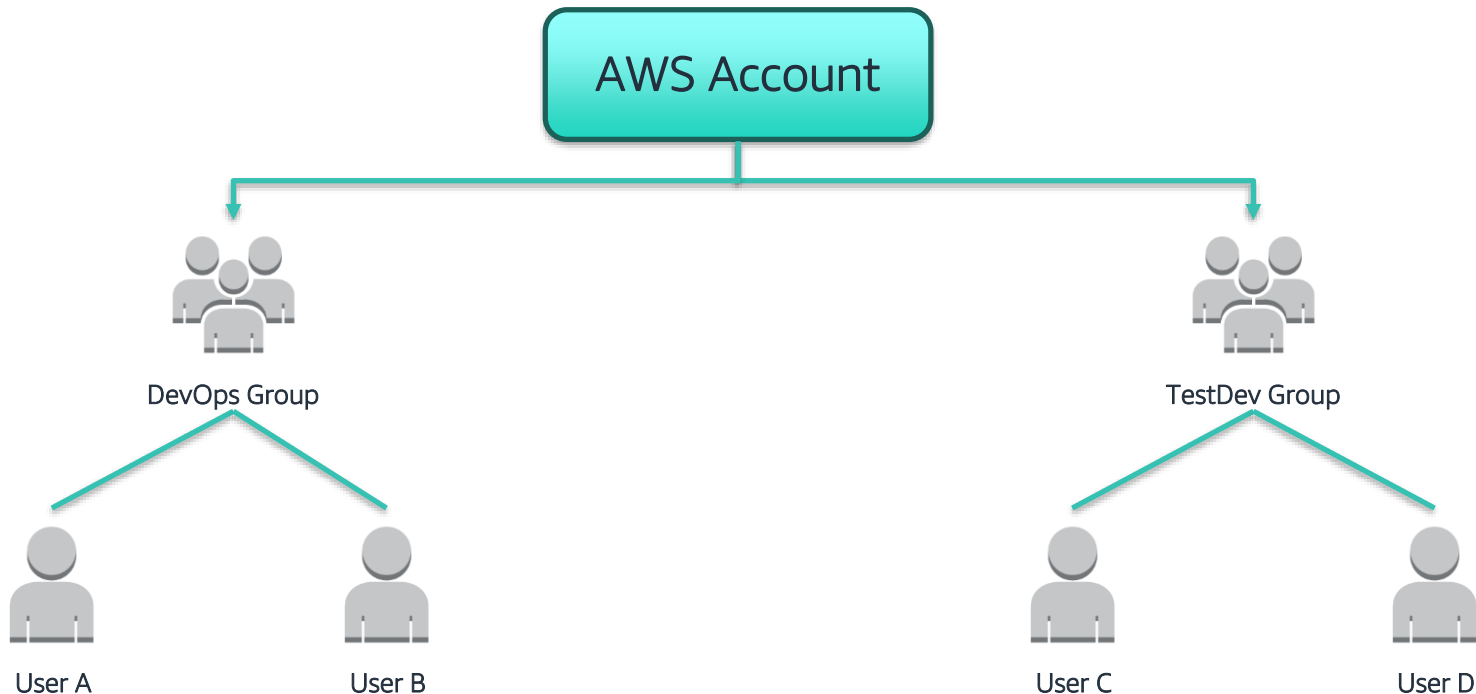


Python



.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

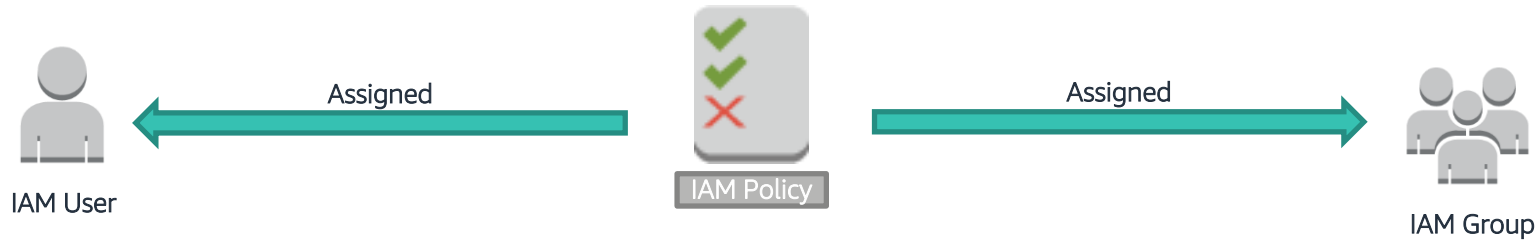
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1453690971587",
      "Action": [
        "ec2:Describe*",
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "IpAddress": {
        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



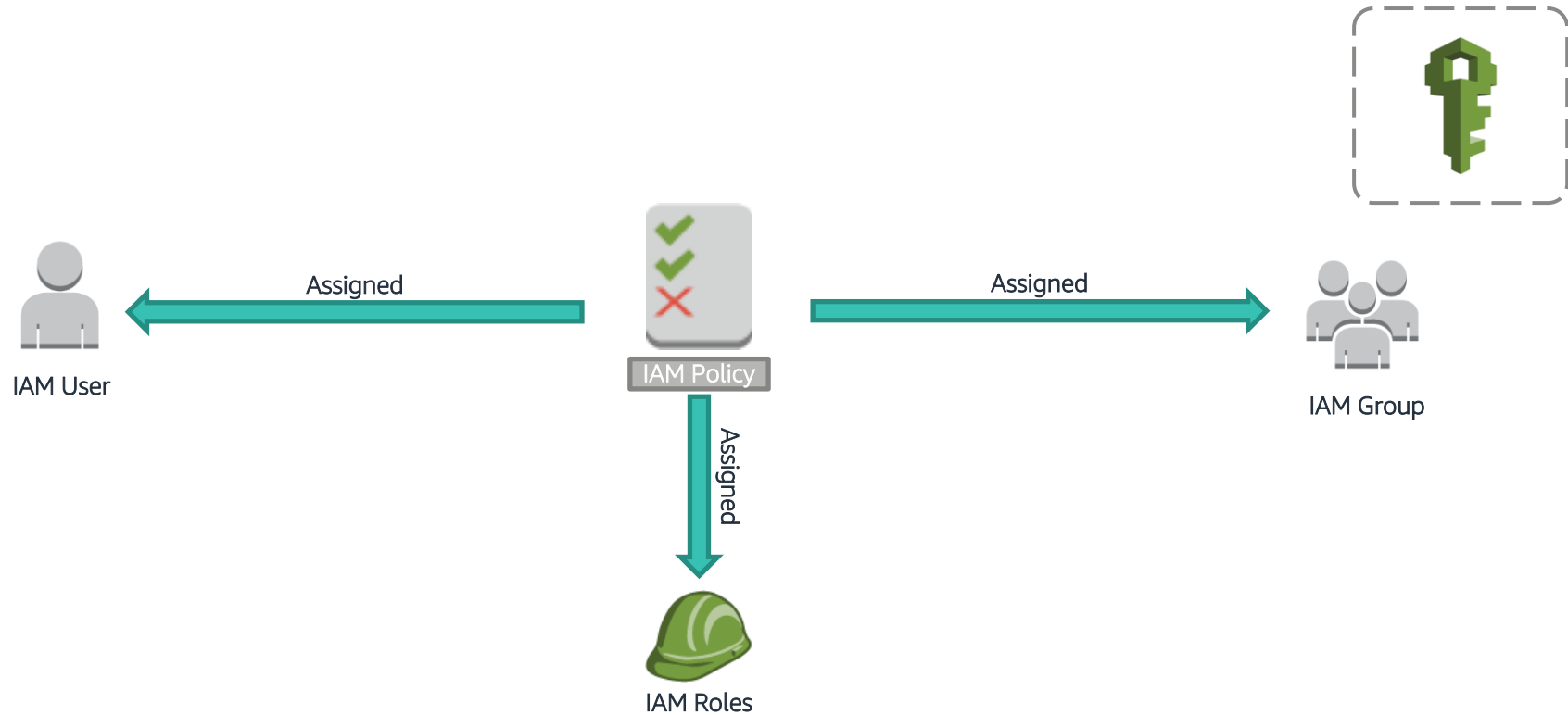
IAM Policy



AWS IAM Policy Assignment (1)



AWS IAM Policy Assignment (2)



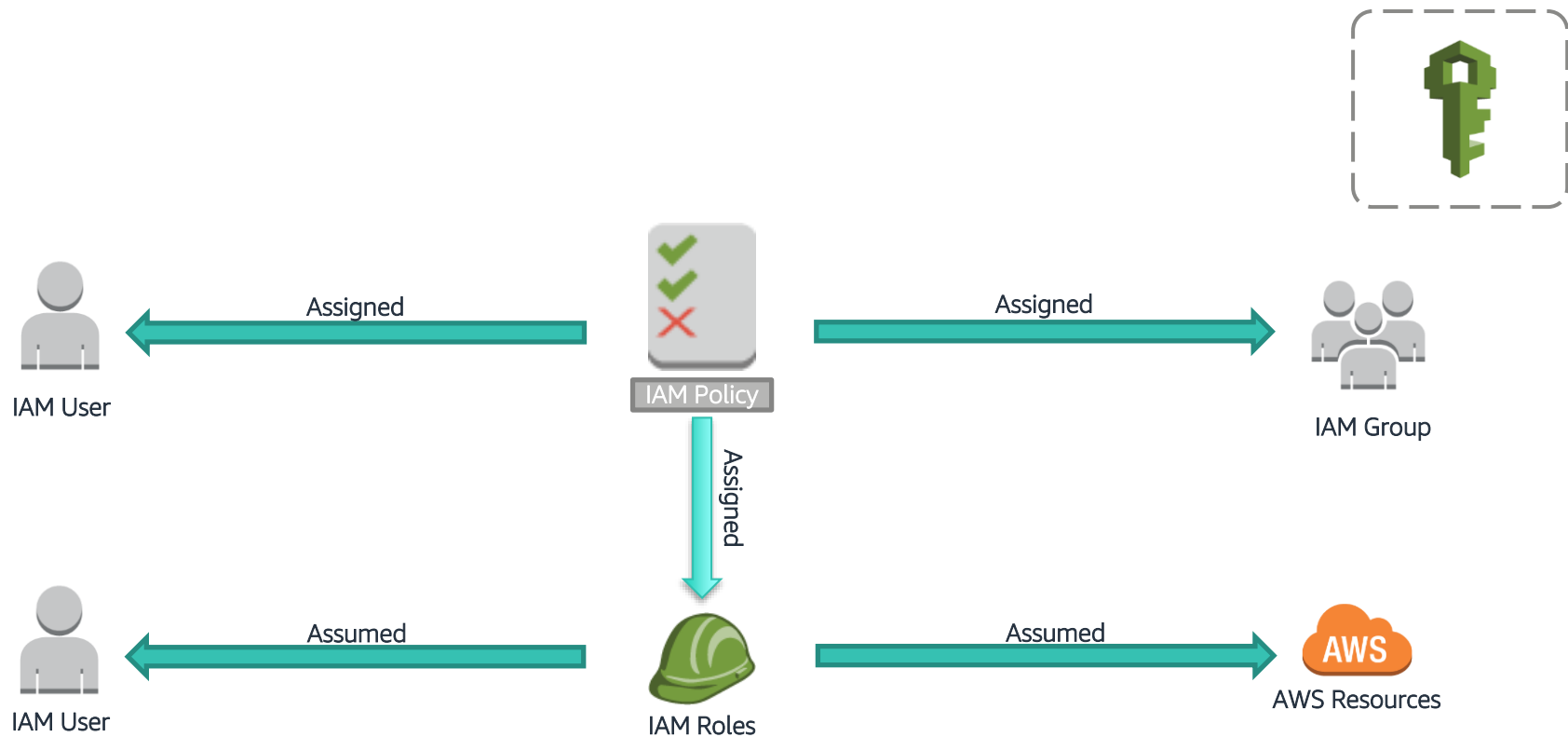
AWS IAM Policy Roles

- 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.



IAM Roles

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-5f... (172.31.0.0/16) (default) [Create new VPC](#)

Subnet: No preference (default subnet in any Availability Z) [Create new subnet](#)

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

Domain join directory: None [Create new directory](#)

IAM role: **None** [Create new IAM role](#)

- None
- aws-elasticbeanstalk-ec2-role
- EMR_EC2_DefaultRole
- Python2GP2AccessS3**
- ProtonTaggingEC2DefaultTermination

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

Enable termination protection: ☐ Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.

Monitoring: ☐ Enable CloudWatch detailed monitoring
Additional charges apply.

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

[Advanced Details](#)



Amazon S3



AWS IAM Roles – Instance Profiles

Amazon EC2



Create Instance

Select IAM Role

AWS Services

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

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-5f1a1b1c (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: **PythonEC2AccessS3** (selected from dropdown: None, aws-elasticbeanstalk-ec2-role, EMR_EC2_DefaultRole, PythonEC2AccessS3) [Create new IAM role](#)

Shutdown behavior: [Additional charges apply.](#)

Enable termination protection: ☐

Monitoring: ☐ Enable CloudWatch detailed monitoring [Additional charges apply.](#)

Tenancy: Shared - Run a shared hardware instance [Additional charges will apply for dedicated tenancy.](#)

[Advanced Details](#)

Amazon S3



App &



AWS IAM Roles – Instance Profiles

Amazon EC2



Create Instance

Select IAM Role

AWS Services

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

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-5f172310 (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: **PythonEC2AccessS3** (selected from dropdown: None, aws-elasticbeanstalk-ec2-role, EMR_EC2_DefaultRole, PythonEC2AccessS3) [Create new IAM role](#)

Shutdown behavior: ☐ Enable CloudWatch detailed monitoring (Additional charges apply)

Enable termination protection: ☐

Monitoring: ☐ Enable CloudWatch detailed monitoring (Additional charges apply)

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

[Advanced Details](#)

Amazon S3



App &



EC2 MetaData Service

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

AWS IAM Roles – Instance Profiles

Amazon EC2



Create Instance

Select IAM Role

2

AWS Services

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

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-5f (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: [None](#) [Create new IAM role](#)

- None
- aws-elasticbeanstalk-ec2-role
- EMR_EC2_DefaultRole
- PythonEC2AccessS3

Shutdown behavior: [None](#)

Enable termination protection: ☐

Monitoring: ☐ Enable CloudWatch detailed monitoring
Additional charges apply.

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

[Advanced Details](#)

Amazon S3



Application interacts with S3

4



App &

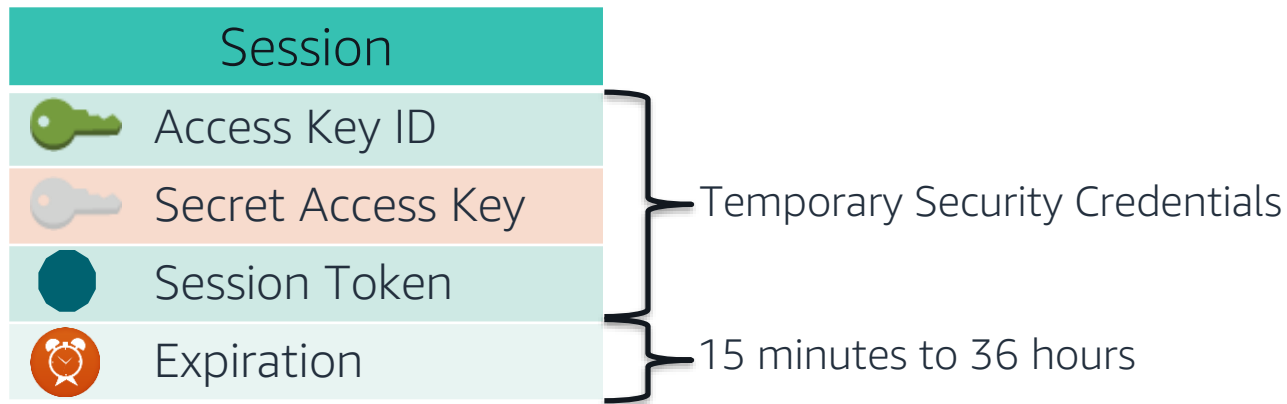


3

EC2 MetaData Service

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

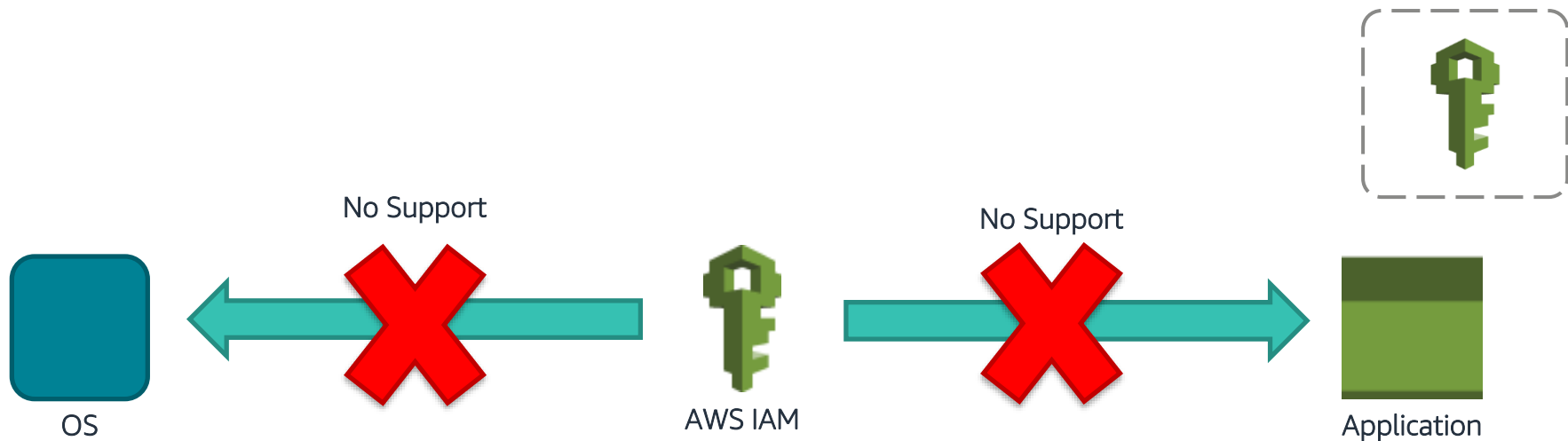
Temporary Security Credentials (AWS STS)



Use Cases

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

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.



DEMO TIME

Learn from AWS experts. Advance your skills and knowledge. Build your future in the AWS Cloud.



Digital Training

Free, self-paced online
courses built by AWS
experts



Classroom Training

Classes taught by
accredited AWS instructors



AWS Certification

Exams to validate
expertise with an industry-
recognized credential

Ready to begin building your cloud skills?
Get started at: <https://www.aws.training/>

Thank You for Attending AWSome Day Online Conference

We hope you found it interesting! A kind reminder to **complete the survey**.
Let us know what you thought of today's event and how we can improve
the event experience for you in the future.



aws-apac-marketing@amazon.com



twitter.com/AWSCloud



facebook.com/AmazonWebServices



youtube.com/user/AmazonWebServices



slideshare.net/AmazonWebServices



twitch.tv/aws