AWS SECURITY SERVICES OVERVIEW

Security, Identity & Compliance

IAM

Inspector

Certificate Manager

Directory Service

WAF & Shield

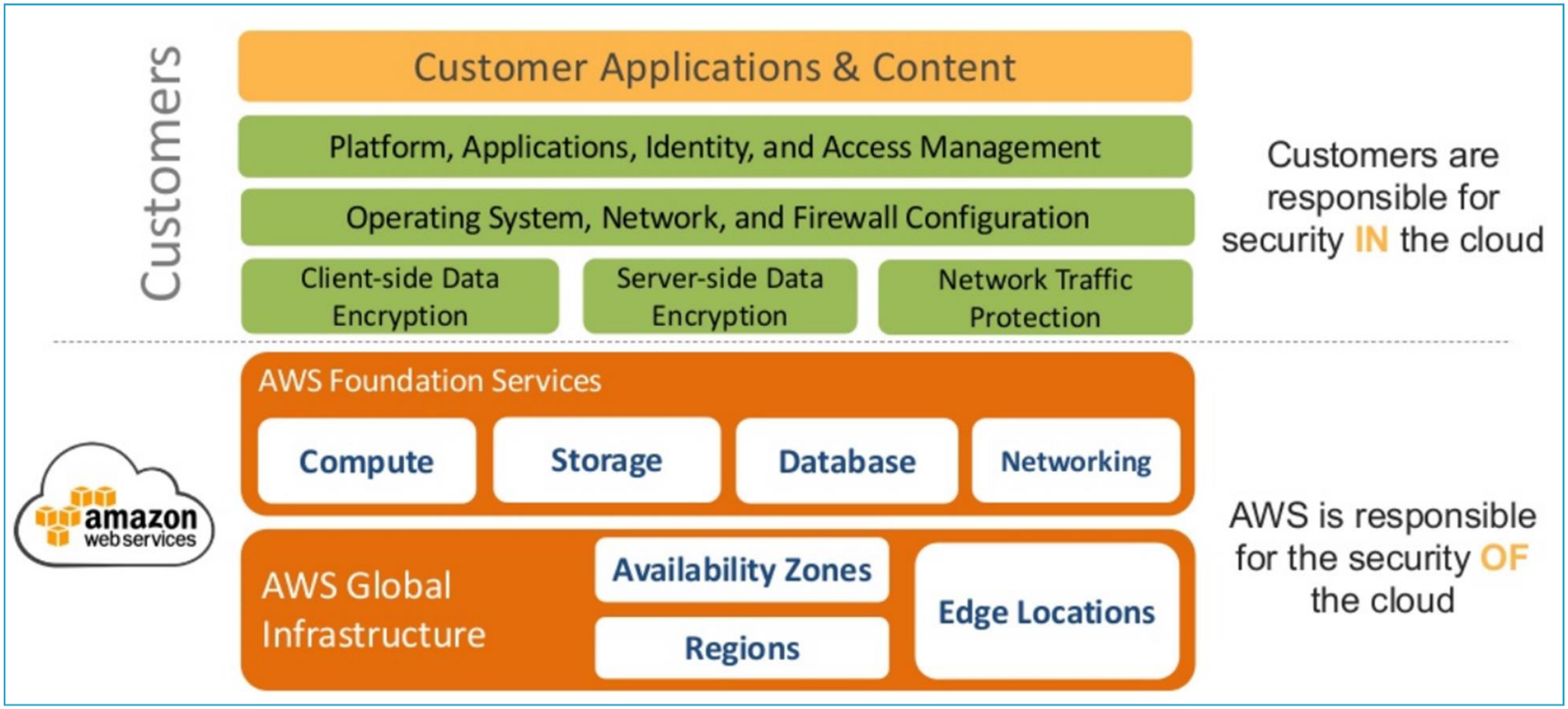
Artifact

Amazon Macie

CloudHSM

AWS Cloud Security

AWS Shared Security 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, Network

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

AWS Security Certifications & Accreditations

AWS Security Compliance

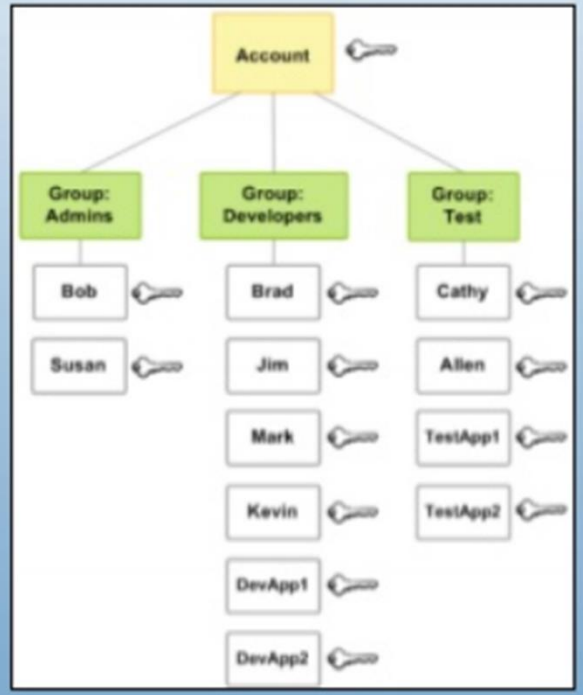
IAM

AWS Identity and Access Management (IAM)

IAM is a web service for securely controlling access to AWS services. Centrally manage users, security credentials such as access keys, and permissions that control which AWS resources, users and applications can access.

AWS Identity and Access Management (IAM)

- Each account has root identity plus Users, Groups, Roles
  - Account-level: password complexity policies
- Unique security credentials for each user
  - Login/password (optional)
  - Access / secret keys (for APIs) (optional)
  - (V)MFA devices (optional)
- Policies control access to AWS APIs
- Deeper integration into some Services
  - S3: policies on objects and buckets
  - Simple DB: domains
- AWS Management Console supports IAM user log on
- Not for Operating Systems or Applications
  - use LDAP, Active Directory/ADFS, etc...



AWS IAM Authorization

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

IAM User

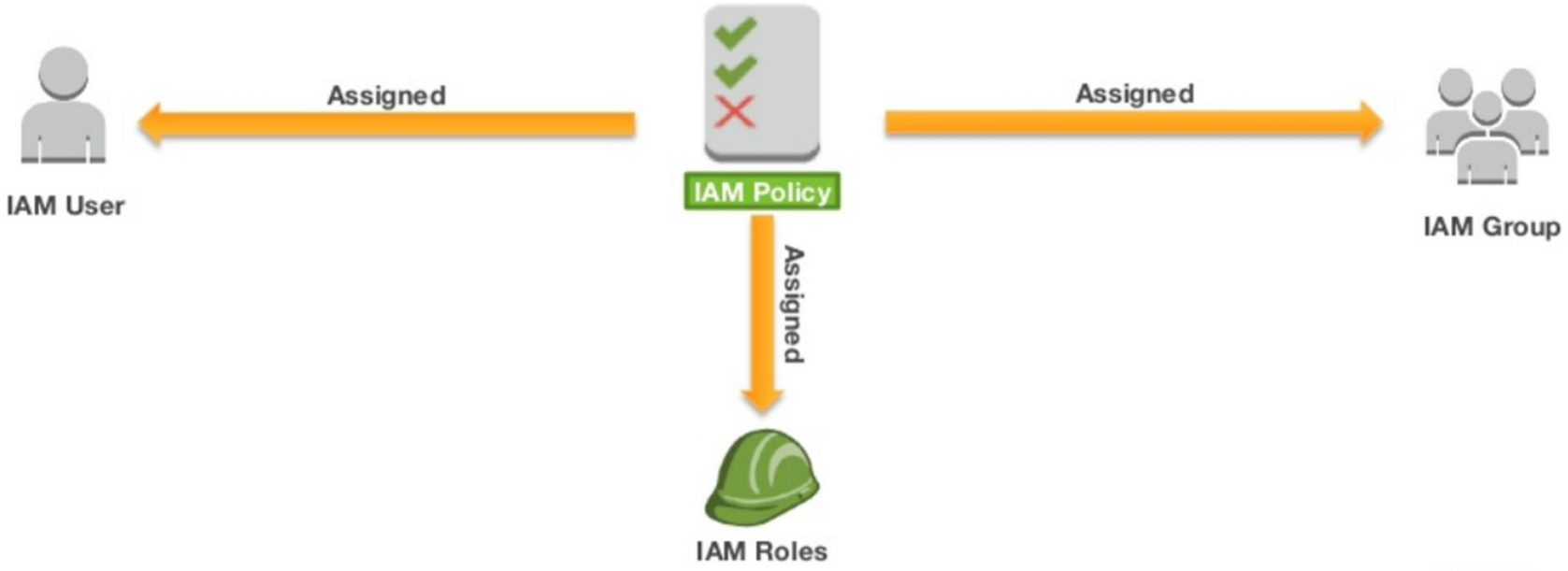
IAM Group

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

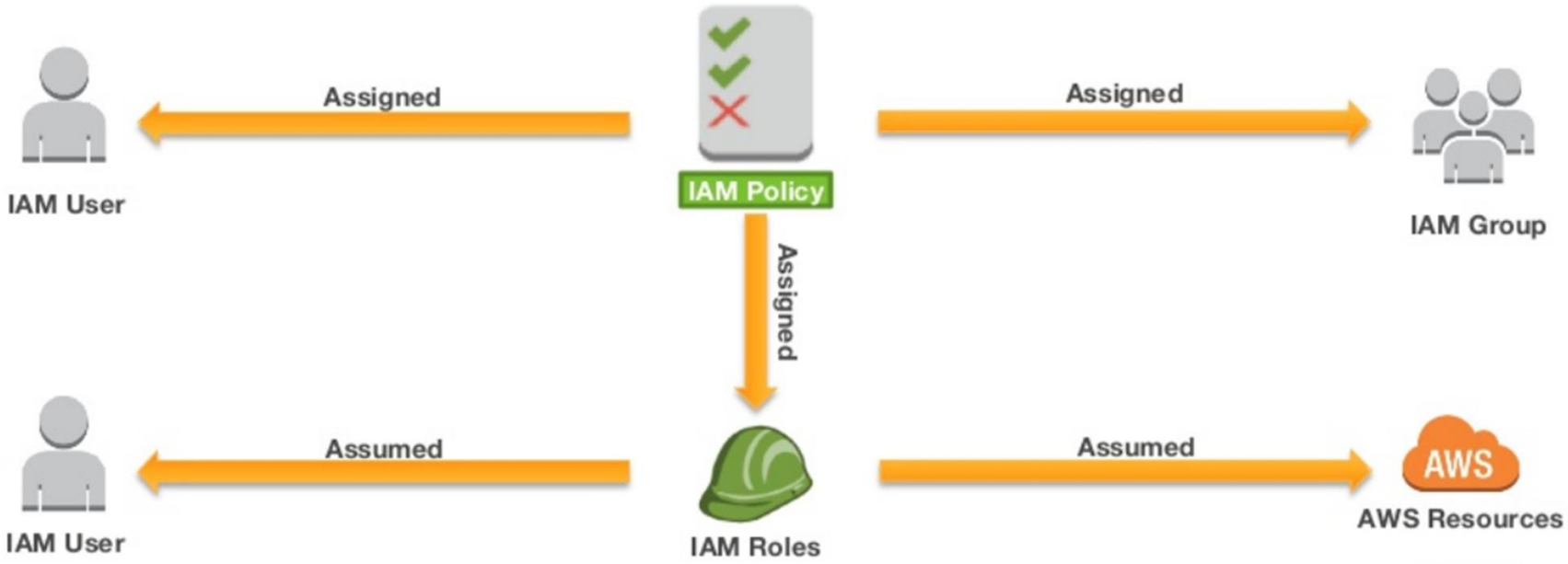


AWS IAM Roles

- 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

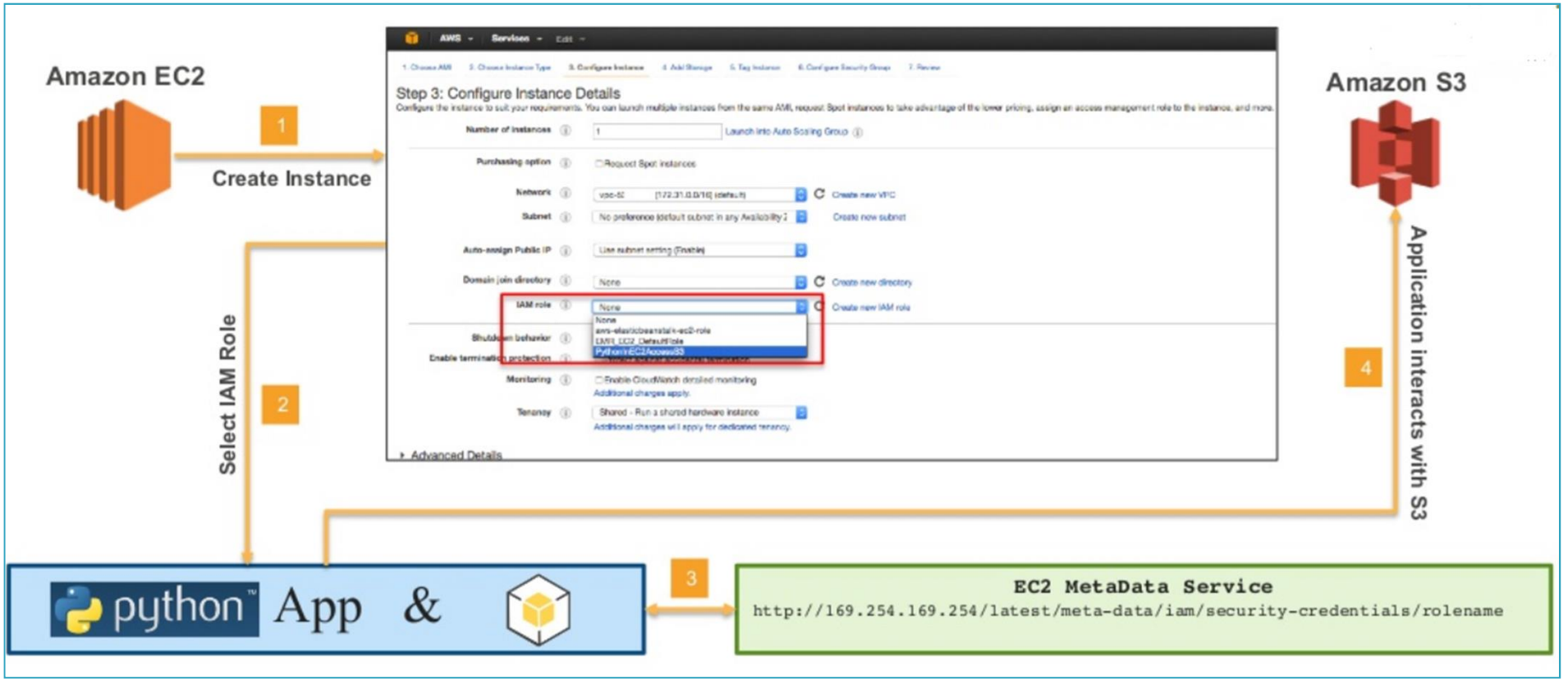


Example: Application Acess to AWS Resources

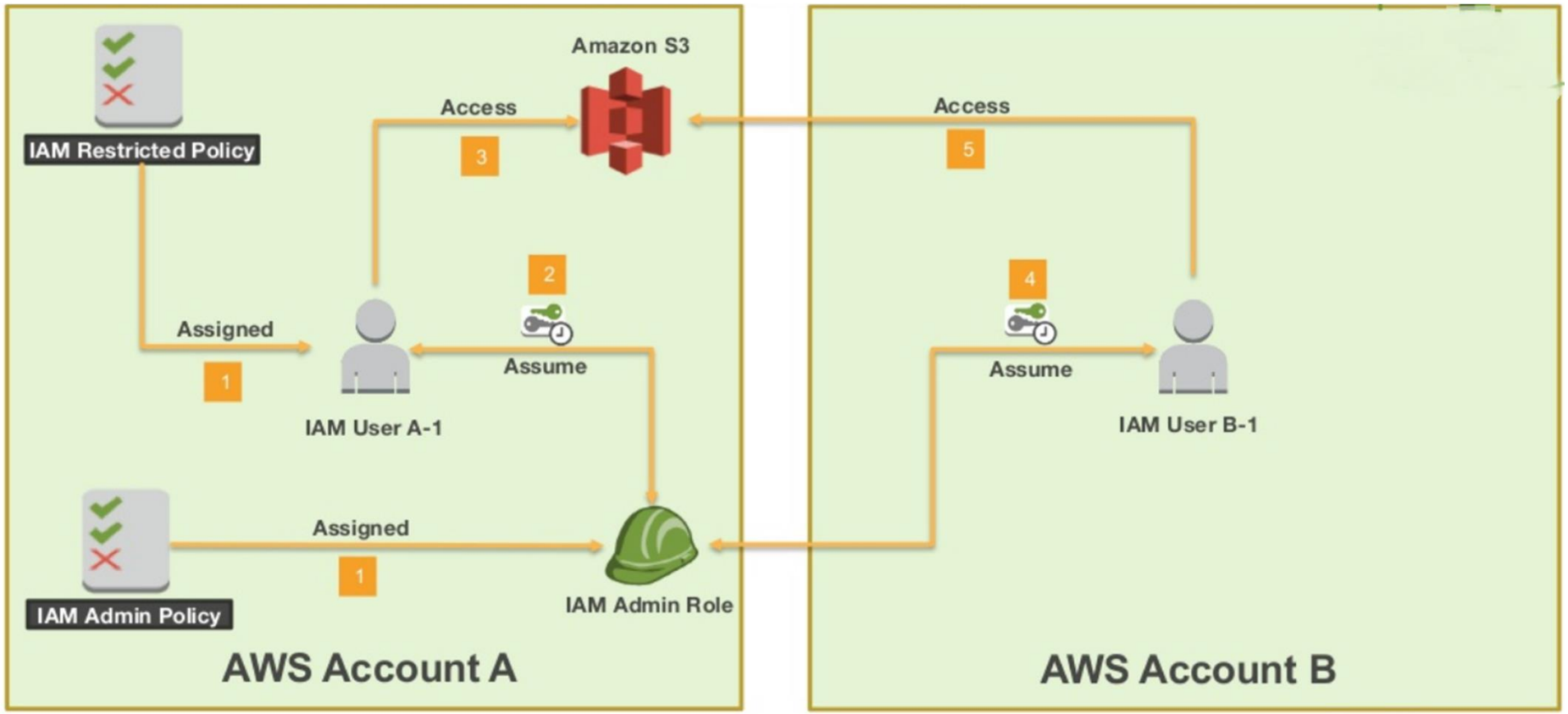
- Python application hosted on an Amazon EC2 Instance needs to interact with Amazon S3.
- AWS credentials are required:
  - ~~Option 1: Store AWS Credentials on the Amazon EC2 instance~~.
  - Option 2: Securely distribute AWS credentials to AWS Services and Applications.

IAM Roles

IAM IAM Roles – Instance Profiles



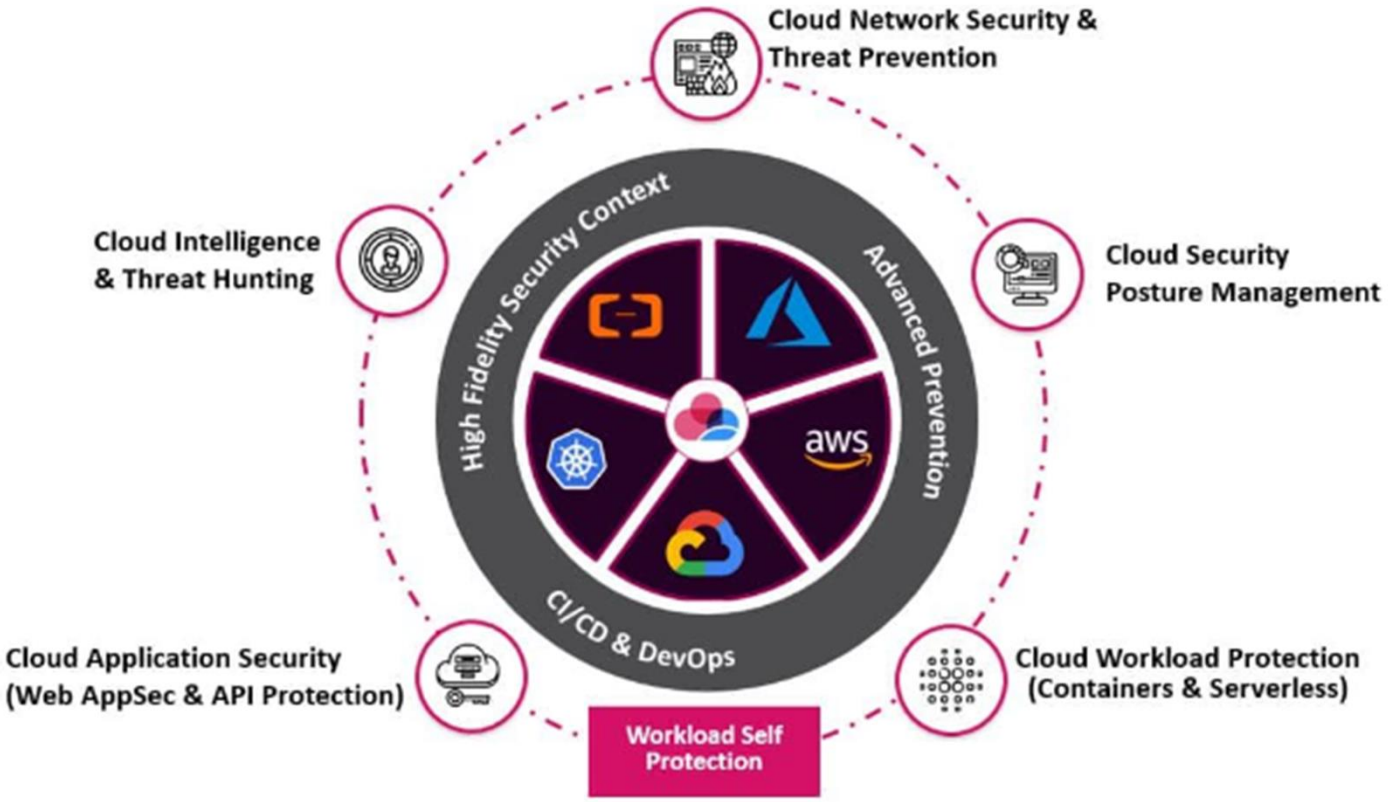
IAM IAM Roles – Assume Role



IAM Best Practices

- Lock away your AWS account root user access keys
- Create individual IAM users
- Use groups to assign permissions to IAM users
- Grant least privilege
- Get started using permissions with AWS managed policies
- Use customer managed policies instead of inline policies
- Use access levels to review IAM permissions
- Configure a strong password policy for your users
- Enable MFA
- Use roles for applications that run on Amazon EC2 instances
- Use roles to delegate permissions
- Do not share access keys
- Rotate credentials regularly
- Remove unnecessary credentials
- Use policy conditions for extra security
- Monitor activity in your AWS account

Cloud Security Pillars



Thank You!