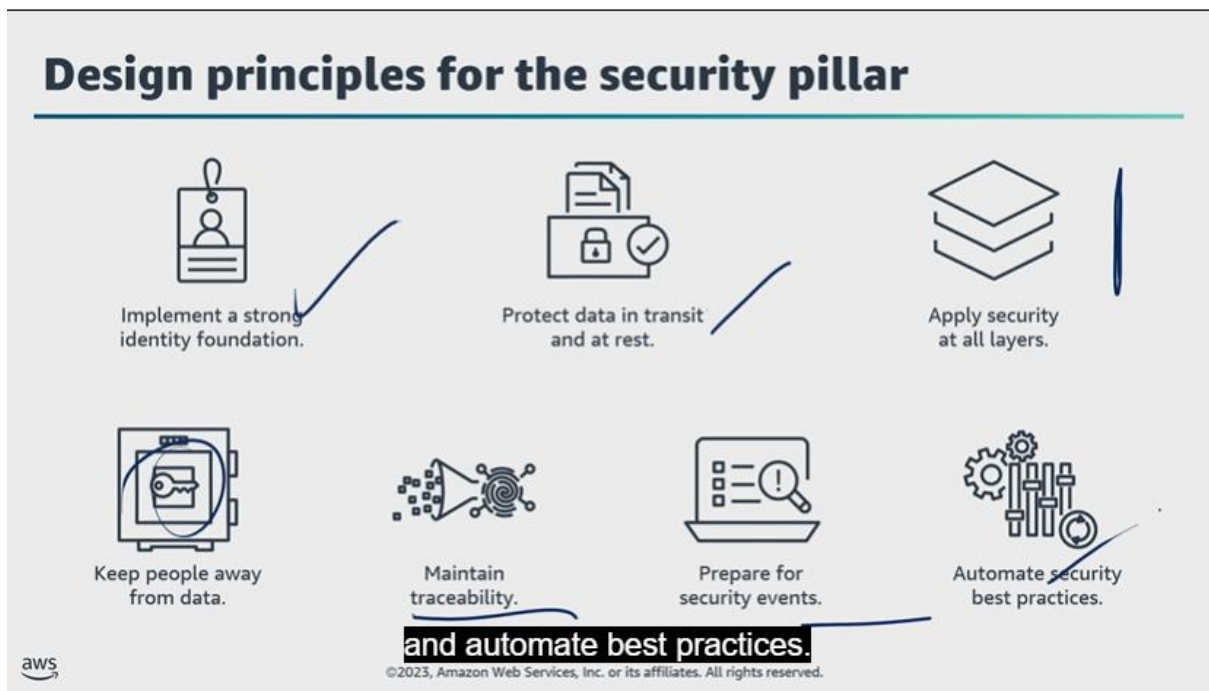


Securing Access

AWS = Security of the cloud

You = security in the cloud

Security ada di 6 pilar yang lalu.



S3 Bucket = permission diberikan tergantung ke orangnya.

Cryptographic protocol.

Amazon S3 = yang bisa encryption.

Key takeaways: Security principles



- Security and compliance are shared responsibilities between AWS and customers.
- The security pillar of the Well-Architected Framework provides design principles to architect secure solutions.
- The principle of least privilege is a key part of implementing a strong identity foundation.
- Another key security design principle is protecting data at rest and in transit. Encryption is one important mechanism for protecting data.

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.



The Security pillar of the

Authenticating and securing access

Authentication = memastikan siapa saja yang bisa masuk ke server.

Authorization = apa yang diizinkan ke user.

IAM resource = Things stored in IAM

IAM entity = Resource can used for authenticated

IAM identity = Can be authorized to do something

IAM credentials for authentication

| Action | Credentials needed |
|--|-----------------------|
| Sign in to the AWS Management Console | Username and password |
| Run commands from the AWS Command Line Interface (AWS CLI) | AWS access key* |
| Make programmatic calls to AWS | AWS access key* |

*An AWS access key is a combination of an access key ID and a secret key.



The first is a username and password that you can use

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS access key = **Access key id** and **secret key id**

Protect root user (pakai root yang tidak bisa dipakai orang lain)

Authorizing Users

Policy defines the permissions (allowed/denied) default = denied

- Identity based (per user)
- Resource-based (per resource)

Identity-based and resource-based policies

Identity-based
(Attached to an *IAM user, group, or role*)

What does a particular identity have access to?

| Carlos | Resource | Read | Write | List |
|----------|------------|-------|-------|-------|
| | Resource X | Allow | Allow | Allow |
| Richard | Resource | Read | Write | List |
| | Resource Y | Allow | N/A | N/A |
| | Resource Z | Allow | N/A | N/A |
| Managers | Resource | Read | Write | List |
| | Resource X | N/A | N/A | Allow |
| | Resource Y | N/A | N/A | Allow |
| | Resource Z | N/A | N/A | Allow |

Resource-based
(Attached to an *AWS resource*)

Who has access to a particular resource?

| Resource X | User | Read | Write | List |
|------------|-------|-------|-------|-------|
| | Ana | Allow | Allow | Allow |
| | Akua | Allow | Allow | Allow |
| Resource Y | User | Read | Write | List |
| | Paulo | Allow | Allow | Allow |
| | Nikki | Allow | N/A | N/A |
| | Mateo | N/A | Allow | Allow |

and grant specific users access to each resource.

Policy: get, put, list / read, write, list

Parts of IAM Policy (stored in JSON documents) = effect, action, resources.

IAM policy document structure

| Element | Information |
|------------------|---|
| Version | Version of the policy language that you want to use |
| Statement | Defines what is allowed or denied based on conditions |
| Effect | Allow or deny |
| Principal | For a resource-based policy, the account, user, role, or federated user to allow or deny access to. For an identity-based policy, the principal is implied as the user or role that the policy is attached to. |
| Action | Action that is allowed or denied Example: "Action": "s3:GetObject" |
| Resource | Resource or resources that the action applies to Example: "Resource": "arn:aws:sqs:us-west-2:123456789012:queue1" (ARN = AWS resource name) |
| Condition | Conditions that must be met for the rule to apply or deny the permissions. |

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example: resource-based policy

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": ["dynamodb:*", "s3:*"],
    "Resource": [
      "arn:aws:dynamodb:region:account-number-without-hyphens:table/course-notes",
      "arn:aws:s3:::course-notes-web",
      "arn:aws:s3:::course-notes-mp3/*"
    ]
  },
  {
    "Effect": "Deny",
    "Action": ["dynamodb:*", "s3:*"],
    "NotResource": [
      "arn:aws:dynamodb:region:account-number-without-hyphens:table/course-notes",
      "arn:aws:s3:::course-notes-web",
      "arn:aws:s3:::course-notes-mp3/*"
    ]
  }
]
```

Explicitly allow any (*) DynamoDB or S3 action on the DynamoDB table course-notes, the S3 bucket course-notes-web and any object in the S3 bucket course-notes-mp3.

Deny any (*) DynamoDB or S3 action on tables or S3 buckets except for those listed under NotResource.

except for those listed under NotResource.

Example: Identity-based policy

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "iam:*LoginProfile*",
      "iam:*AccessKey*",
      "iam:*SSHPublicKey*"
    ],
    "Resource": [
      "arn:aws:iam::account-id-without-hyphens:user/${aws:username}"
    ]
  }]
}
```

The Action element lists all the actions that are allowed by the Effect: Allow.

The Resource element lists the AWS resources that the allowed actions can be performed on.



that might be attached to a user, group or role.

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example: Cross-account, resource-based policy

Policy created by account A

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "AccountBAccess1",
    "Principal": {"AWS": "111122223333"},
    "Effect": "Allow",
    "Action": "s3:*",
    "Resource": [
      "arn:aws:s3::DOC-EXAMPLE-BUCKET",
      "arn:aws:s3::DOC-EXAMPLE-BUCKET/*"
    ]
  }
}
```

Account number of Account B

Allow account B to take any S3 action on the DOC-EXAMPLE-BUCKET.