Week 5 AWS Identity Access Management

Dr Zhi Zhang

Create Entries

```
aws dynamodb put-item \
  --table-name MusicAlbum \
  --item \ '{"Artist": {"S": "Tom"}, "Song": {"S": "Call Me Today"},
                 "AlbumTitle": {"S": "Somewhat Famous"}}' \
  --return-consumed-capacity TOTAL --endpoint-url=http://localhost:8000


aws dynamodb put-item \
  --table-name MusicAlbum \
  --item '{"Artist": {"S": "Jerry"}, "Song": {"S": "Happy Day"}}' \
--return-consumed-capacity TOTAL  --endpoint-url=http://localhost:8000
```

- Demo: what a table will be like if we create the first entry with 3 attributes and the second entry with 2 attributes?

Overview

- Cryptography
- IAM (Identity Access Management)

Cybersecurity

- It is about the protection of digital information from unauthorised access, harm or misuse.
- This is done by preserving the CIA triad of the information, i.e., Confidentiality, Integrity and Availability.
- **Confidentiality**: keeps sensitive information private and ensures that only authorized individuals or entities have access to it.
- **Integrity**: maintains the accuracy, consistency, and reliability of information.
- **Availability:** ensures that information such as services and data are accessible and operational for authorized users.

Other three cybersecurity terminology

- CIA can be extended to include such as Authentication, Authorization Non-Repudiation, etc.
- **Authentication:** verifies the identity of a user, system, or entity trying to access a resource or system.
- **Authorization:** determines what actions or resources an authenticated user or system is allowed to access or perform.
- **Non-Repudiation:** prevents individuals or entities from denying their involvement in a particular digital transaction.

Cryptography

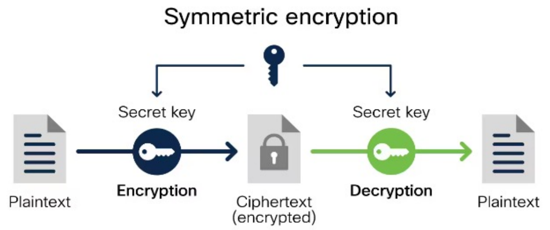
- It is the practice and study of techniques for secure communication and data protection in the presence of adversaries or potential threats.
- It is mainly about the use of mathematical algorithms to transform plain, readable data (i.e., plaintext) into an unintelligible data (i.e., ciphertext) and vise versa
- The transformations involve encryption and decryption.
  - Encryption: takes plaintext as input and converts it into ciphertext
  - Decryption: reverses this process above

Cryptography

- It is the practice and study of techniques for secure communication and data protection in the presence of adversaries or potential threats.
- It is mainly about the use of mathematical algorithms to transform plain, readable data (i.e., plaintext) into an unintelligible format (i.e., ciphertext) and vise versa
- The transformations involve encryption and decryption.
  - Encryption: takes plaintext and converts it into ciphertext
  - Decryption: reverses this process above
- **Caesar cipher**: an old-fashion substitution cipher where each letter in the plaintext is shifted a certain number of positions down the alphabet.
  - ROT3
    - PT : abcdefghijklmnopqrstuvwxyz
    - CT : defghijklmnopqrstuvwxyzabc

Cryptography today

- Symmetric key cryptography: the same key is used for encryption and decryption of data.

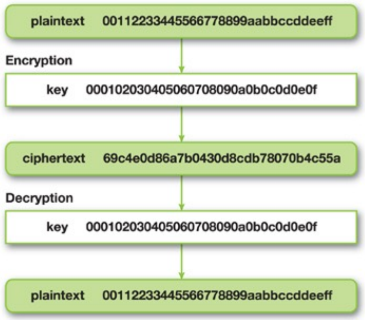


  - Examples: DES, 3DES, **AES**.
  - Applications: data (file, disk, network packets) encryption

https://www.cisco.com/c/en/us/products/security/encryption-explained.html#~encryption-algorithms

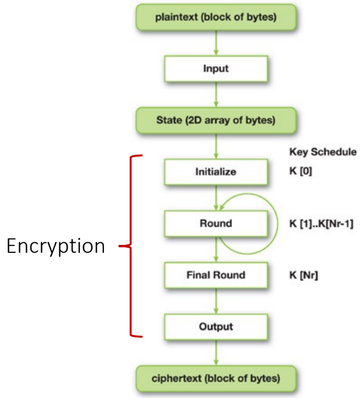
AES (Advanced Encryption Standard)

- AES encrypts a block of 128 bits (16 bytes) at one time.
- Why does the plaintext consist of numeric values only?
  - Plaintext is originally a piece of human readable sentences and can be encoded into blocks of numeric values via mainstream encoders such as ASCII.


plaintext 00112233445566778899aabbccddeeff
Encryption
key 000102030405060708090a0b0c0d0e0f
ciphertext 69c4e0d86a7b0430d8cdb78070b4c55a
Decryption
key 000102030405060708090a0b0c0d0e0f
plaintext 00112233445566778899aabbccddeeff

https://developer.nvidia.com/gpugems/gpugems3/part-vi-gpu-computing/chapter-36-aes-encryption-and-decryption-gpu

AES (encryption)



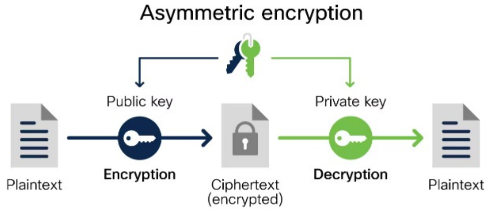
Key Length	Number of Rounds
128	10
192	12
256	14

- AES-128, AES-192, AES-256
- A longer key provides stronger security

https://developer.nvidia.com/gpugems/gpugems3/part-vi-gpu-computing/chapter-36-aes-encryption-and-decryption-gpu

Cryptography today

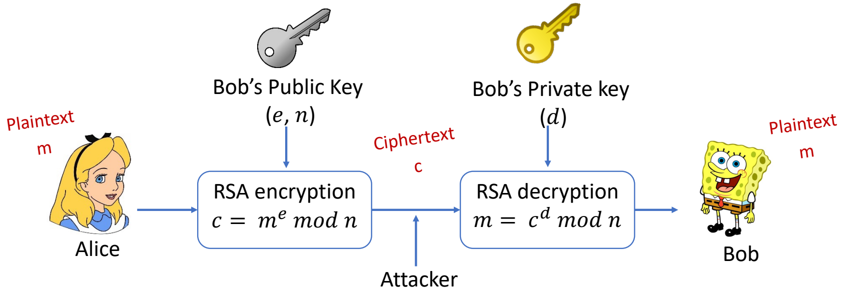
- Symmetric key cryptography: the same key is used for encryption and decryption of data.
- Asymmetric key cryptography (public key cryptography): a pair of distinct keys is used for encryption and decryption.



- Examples: Diffie-Hellman key exchange, ECC, **RSA**
- Applications: remote access (e.g., SSH communication), authentication (e.g., digital signatures), etc.

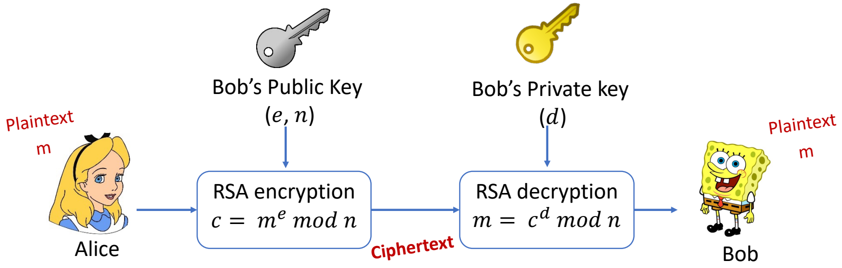
https://www.cisco.com/c/en/us/products/security/encryption-explained.html#~encryption-algorithms

RSA



- $n = p * q$ where p and q are two large prime numbers
- As d is based on p and q, RSA's security WILL be broken if n can be factorized into $p * q$

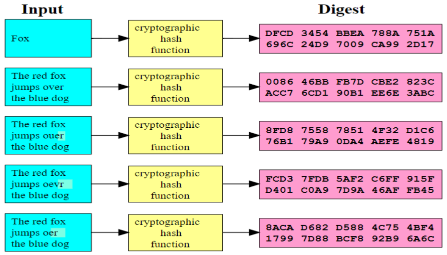
RSA



- Symmetric key cryptography is **much faster** than asymmetric key cryptography. When asymmetric key cryptography achieves key exchange, symmetric key cryptography is in place for secure data transmission.

Cryptography today

- Symmetric key cryptography, Asymmetric key cryptography,
- Hash functions: take an input (e.g., a large block of text) and transform it into a fixed-size value (i.e., hash digest/checksum). The hash value serves as a 'fingerprint' of the input.



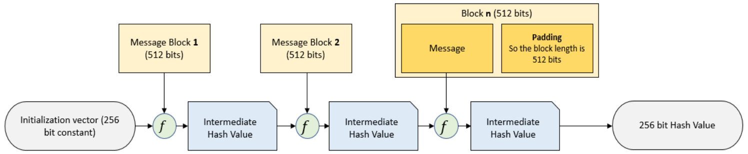
Input
Digest
Fox
cryptographic hash function
DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17
The red fox jumps over the blue dog
cryptographic hash function
0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC
The red fox jumps ouer the blue dog
cryptographic hash function
8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819
The red fox jumps oevr the blue dog
cryptographic hash function
FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45
The red fox jumps oer the blue dog
cryptographic hash function
8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C

- Examples: MD5, SHA-1, SHA-2 (e.g., **SHA-256**)

https://upload.wikimedia.org/wikipedia/commons/2/2b/Cryptographic_Hash_Function.svg

SHA256 (Secure Hash Algorithm 256-bit)

- It is a series of mathematical operations that takes an input message and produces a fixed-size 256-bit hash value.

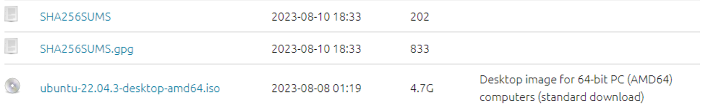


Message Block **1** (512 bits)
Message Block **2** (512 bits)
Block **n** (512 bits)
Message
Padding So the block length is 512 bits
Initialization vector (256 bit constant)
f
Intermediate Hash Value
f
Intermediate Hash Value
f
Intermediate Hash Value
256 bit Hash Value

https://steemit.com/cryptocurrency/@f4tca7/introduction-to-the-sha-256-hash-function

SHA256

- A real-world example: verifying file integrity



SHA256SUMS	2023-08-10 18:33	202	
SHA256SUMS.gpg	2023-08-10 18:33	833	
ubuntu-22.04.3-desktop-amd64.iso	2023-08-08 01:19	4.7G	Desktop image for 64-bit PC (AMD64) computers (standard download)

- SHA256SUMS: contains a checksum/hash digest for the iso image to verify the image's integrity.
- SHA256SUMS.gpg: contains a signature for the SHA256SUMS file to verify the image's authenticity.

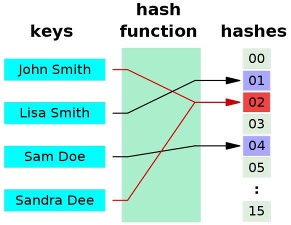
Properties of hash functions

- The same message results in the same hash digest
- Small changes to a message result in large changes to its hash digest

Hash collision

- While two different messages are very unlikely to generate the same hash, such a possibility still exists, so-called **hash collision** (e.g., MD5 and SHA-1)

Why?


keys
hash function
hashes
John Smith
Lisa Smith
Sam Doe
Sandra Dee
00
01
02
03
04
05
:
15

https://en.wikipedia.org/wiki/Hash_collision

Pigeonhole principle

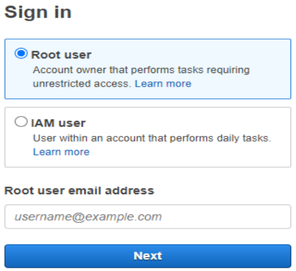
- if n items are put into m containers, with $n > m$, then at least one container must contain more than one item.
- e.g., pigeons in holes

https://en.wikipedia.org/wiki/Pigeonhole_principle

What is IAM (identity access management)?

- It is a web service that helps us securely control access to AWS resources.
- It is used to control who is authenticated (signed in) and authorized (has permissions) to use AWS resources.

Root user: complete access to all AWS services and resources in the account


Sign in

● **Root user**
Account owner that performs tasks requiring unrestricted access. Learn more

○ **IAM user**
User within an account that performs daily tasks. Learn more

Root user email address

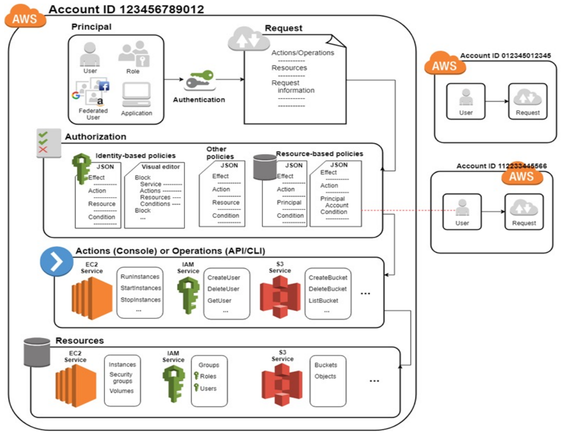
username@example.com

IAM identity

- **IAM user**: an identity within a root user account that has specific permissions for a single person or application:
  - Each user has an ARN:
    e.g., arn:aws:iam::489389878001:user/12345678@student.uwa.edu.au
- **IAM user group**: an identity that specifies a collection of IAM users:
  - Users within the same group are given the same set of permissions.
  - Users can belong to different groups.
  - Each group has an ARN, e.g., arn:aws:iam::489389878001:group/admins
- **IAM role**: an identity that has specific permissions, similar to IAM user but not relevant to a specific person/application.
  - Any users/applications can assume a role to complete a specific task.
    - User case: an IAM role grants permissions to applications running on EC2 instances
  - Each role has an ARN, e.g., arn:aws:iam:: 489389878001 :role/apps4ec2

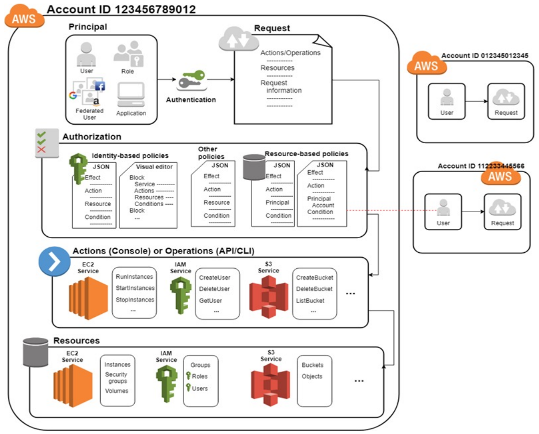
How IAM works

- Step 1: Authenticate a principal.
  - **Principal**: a person or application that uses an IAM user, a root user, or an IAM role to sign in and make requests to AWS.



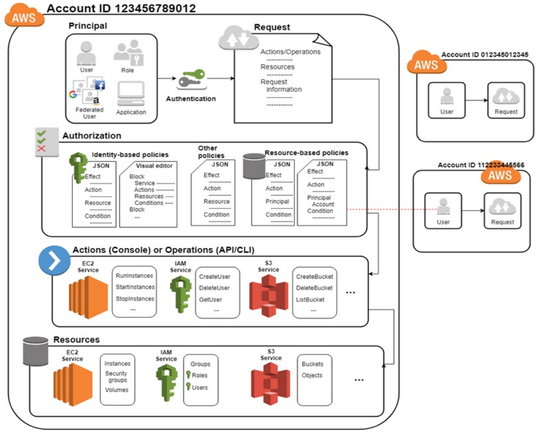
How IAM works

- Step 1: Authenticate a principal.
- Step 2: Authorize a principal.



How IAM works

- Step 1: Authenticate a principal.
- Step 2: Authorize a principal.
- Step 3: Take actions/operations on AWS resources.



Main features of IAM

- Shared access to AWS root user account
  - Grant other people permission to use resources in our root user account without having to share our password or access key.
- Granular permissions
  - Grant different permissions to different people for different resources.
    - e.g., some users have complete access to specified EC2 instances while some have read-only access to specified S3 buckets.

Policies and permissions

- Access permissions (authorization) are managed by creating policies and attaching them to IAM identities (users, groups of users, or roles) or AWS resources.
- Note: IAM policies only define permissions for an action regardless of the method that we use to perform the action
  - e.g., if a policy allows the GetUser action, then a user with that policy can get user information with all three methods.
- Policy types (most frequently used):
  - Identity-based policy
  - Resource-based policy
  - permissions boundary

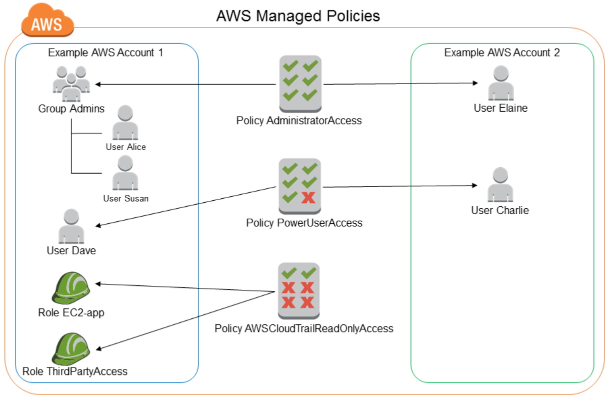
Identity-based policy

- It’s in a JSON format that controls what actions an identity can perform.
- **Managed policy**: standalone identity-based policy that we can attach to multiple users, groups, and roles.
  - AWS managed policy: created and managed by AWS
  - Customer managed policy: created and managed by AWS users.
- **Inline policy**: it maintains a strict one-to-one relationship between a policy and an identity. If the identity is deleted, the policy is deleted as well.

AWS managed policy

- full-access managed policy: defines permissions for administrators by granting full access to services.
- power-user managed policy: provides full access to services and resources, but disallows managing users and groups, i.e., a subset of full-access managed policy.
- partial-user managed policy: provides specific access to specified services, i.e., a subset of power-user managed policy.

AWS managed policy



https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html#aws-managed-policies

AdministratorAccess

Version: indicates the language version of the policy language.

Statement: represents a permission rule.

Effect: what the effect will be when a user requests the specific action—this can be either **’Allow**’ or **’Deny**’.

Action: defines a set of resource operations a user/application is allowed (or denied) to perform.

Resource: specifies AWS resources for which a user is allowed or denied to take actions. ARN is often used.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": "*",
         "Resource": "*"
      }
   ]
}
```

PowerUserAccess

- Organizations: are a service that allows us to consolidate multiple AWS accounts into an organizational structure.

- This policy allows actions against all resources except management of IAM, organizations and account.

```
{
   "Version": "2012-10-17",
   "Statement": [
     { "Effect": "Allow",
        "NotAction": [
          "iam:*",
          "organizations:*",
          "account:*"
        ],
        "Resource": "*"
     },
     { "Effect": "Allow",
        "Action": [
          "iam:ListRoles",
          "organizations:DescribeOrganization",
          "account:GetAccountInformation"
        ],
        "Resource": "*"
     }
   ]
}
```

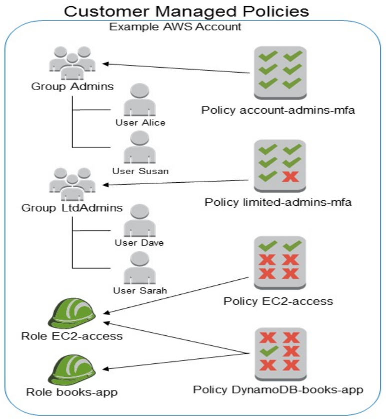
AWSCloudTrail_ReadOnlyAccess

- CloudTrail is a service that provides visibility into user activity and resource usage.
- records and stores AWS Management Console actions, AWS SDK calls, AWS CLI commands, and other AWS service activity.
- A trail records the resources to be monitored, the storage locations for log files, and other log data.
- e.g., GetTrail, DescribeTrails, ListTrails

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "cloudtrail:Get*",
            "cloudtrail:Describe*",
            "cloudtrail:List*",
         ],
         "Resource": "*"
      }
   ]
}
```

https://docs.aws.amazon.com/awscloudtrail/latest/APIReference/API_Operations.html

Customer managed policy



https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html#aws-managed-policies

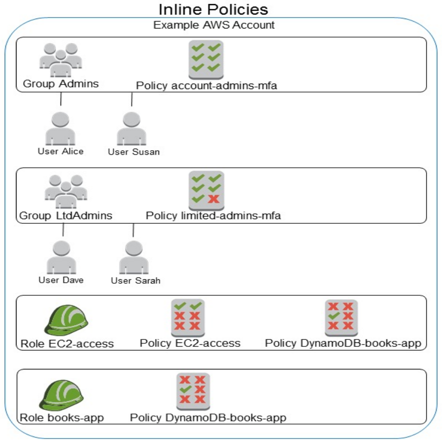
cits5503StudentPolicy

- Based on **PowerUserAccess**.

```
{
        "Effect": "Allow",
        "Action": [
           "iam:CreateAccessKey",
           "iam:DeleteAccessKey",
           "iam:ListAccessKeys",
           "iam:UpdateAccessKey",
           "iam:GetAccessKeyLastUsed",
           "iam:DeleteSSHPublicKey",
           "iam:GetSSHPublicKey",
           "iam:ListSSHPublicKeys",
           "iam:UpdateSSHPublicKey",
           "iam:UploadSSHPublicKey",
            "account:ListRegions",
           "account:GetAccountInformation",
            ],
        "Resource": "*"
}
```

Inline policy

- The DynamoDB-books-app policy is used by both roles. Is it shared?



https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html#aws-managed-policies

Resource-based policy

- It’s in a JSON format that grants **specified principals specific permissions** to perform **specific actions** on **specific resources** under **specific conditions**.
- Note: it is an inline policy.
- e.g., bucket policy:

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": "*",
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3::: cits5503-123456-lecture /*"
    }]
}
```

Permissions boundary

- It is an advanced feature for using a managed policy to set the **maximum permissions** that an identity-based policy can grant.
- e.g., The permissions boundary is attached to an IAM user named Alice.

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
       "Effect": "Allow",
       "Action": [
         "s3:*",
         "ec2:*"
       ],
       "Resource": "*"
     }
   ]
}
```

Permissions boundary

identity-based policy

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:CreateUser",
    "Resource": "*"
  }
}
```

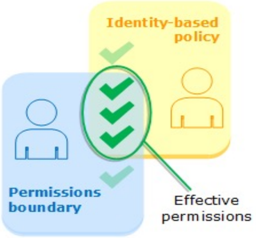
Permissions boundary

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
        "Effect": "Allow",
        "Action": [
           "s3:*",
           "ec2:*"
        ],
        "Resource": "*"
     }
   ]
}
```

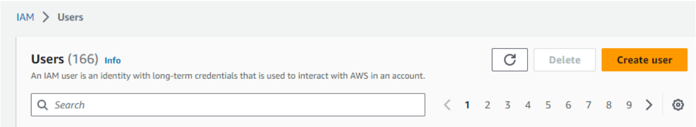
- Both policies are attached to Alice.
- Can Alice really create a user?
- Can Alice really create S3 buckets and EC2 instances?

Permissions boundary

- Both answers are NO.
- Effective permissions are in the intersection of Identity-based policies and permissions boundaries.



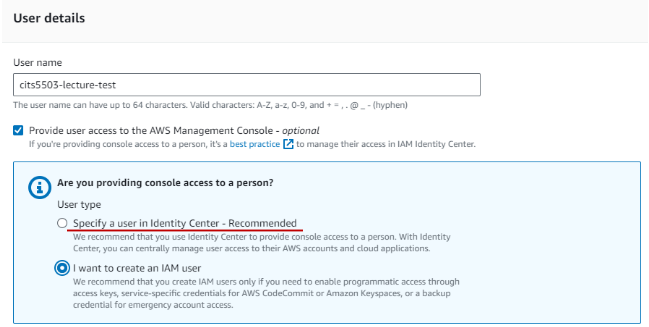
Attach customer managed policy to an IAM user


IAM > Users

Users (166) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Specify user details


User details

User name

cits5503-lecture-test

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☑ Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.

Are you providing console access to a person?

User type

○ Specify a user in Identity Center - Recommended

We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

● I want to create an IAM user

We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

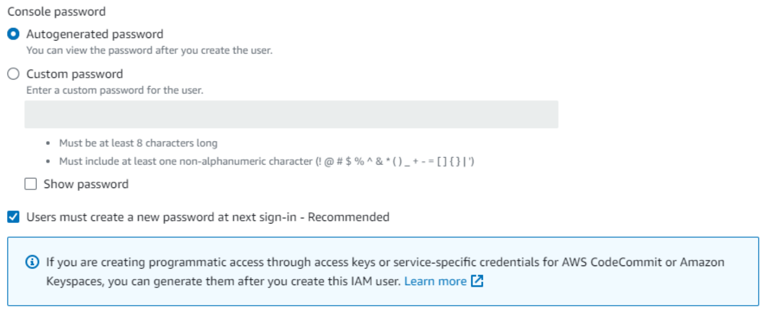
IAM identity center

- It is a place where an administrator can create or connect workforce users and centrally manage their access across all their AWS accounts and applications.
  - Workforce users/identities refer to users who are members within the same organization.
- The admin can use **multi-account permissions** to assign their workforce users access to multiple AWS accounts.

IAM user

- It is an identity **within a root user account** that has specific permissions for a single person or application.
- It is unlikely for an IAM user to have multi-account access unless explicitly specified.

Specify user details



Console password

● Autogenerated password
You can view the password after you create the user.

○ Custom password
Enter a custom password for the user.

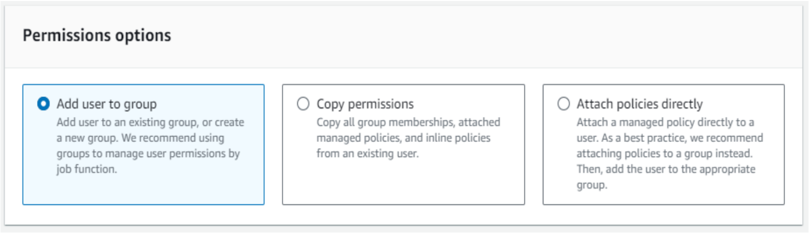
- Must be at least 8 characters long
- Must include at least one non-alphanumeric character (! @ # $ % ^ & * ( ) _ + - = [ ] { } | ')

□ Show password

☑ Users must create a new password at next sign-in - Recommended

ⓘ If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more

Set permissions



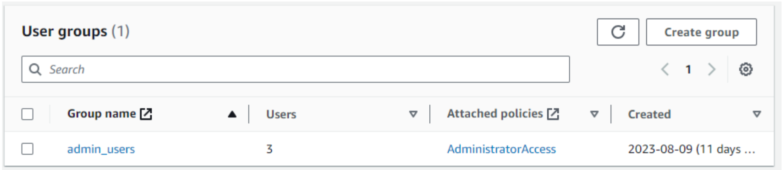
Permissions options

● Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

○ Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

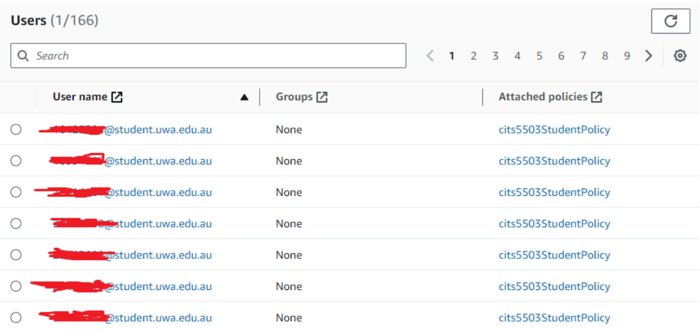
Add user to group



User groups (1)

□	Group name	Users	Attached policies	Created
□	admin_users	3	AdministratorAccess	2023-08-09 (11 days ...

Copy permissions



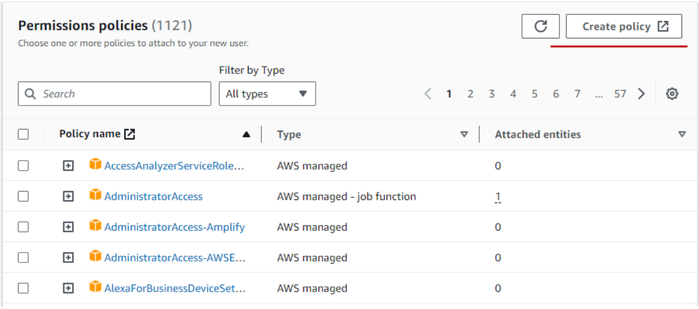
Users (1/166)

Search

< 1 2 3 4 5 6 7 8 9 >

	User name	Groups	Attached policies
○	[redacted]@student.uwa.edu.au	None	cits5503StudentPolicy
○	[redacted]@student.uwa.edu.au	None	cits5503StudentPolicy
○	[redacted]student.uwa.edu.au	None	cits5503StudentPolicy
○	[redacted]student.uwa.edu.au	None	cits5503StudentPolicy
○	[redacted]student.uwa.edu.au	None	cits5503StudentPolicy
○	[redacted]student.uwa.edu.au	None	cits5503StudentPolicy
○	[redacted]student.uwa.edu.au	None	cits5503StudentPolicy

Attach policies directly



Permissions policies (1121)

Choose one or more policies to attach to your new user.

Create policy

Filter by Type: All types

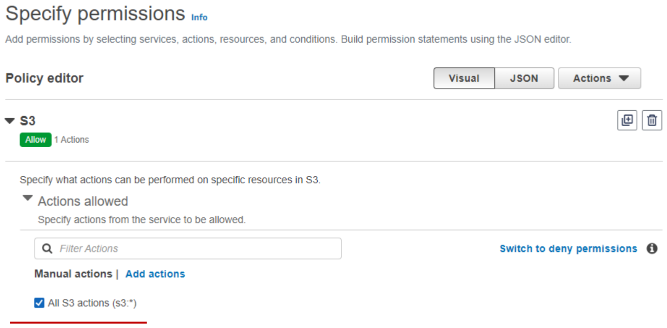
Search

< 1 2 3 4 5 6 7 ... 57 >

□	Policy name	Type	Attached entities
□	AccessAnalyzerServiceRole...	AWS managed	0
□	AdministratorAccess	AWS managed - job function	1
□	AdministratorAccess-Amplify	AWS managed	0
□	AdministratorAccess-AWSE...	AWS managed	0
□	AlexaForBusinessDeviceSet...	AWS managed	0

Create customer managed policy

- A policy allows the IAM user to access a specified S3 bucket only.



Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

Visual | JSON | Actions

S3

Allow 1 Actions

Specify what actions can be performed on specific resources in S3.

Actions allowed

Specify actions from the service to be allowed.

Filter Actions

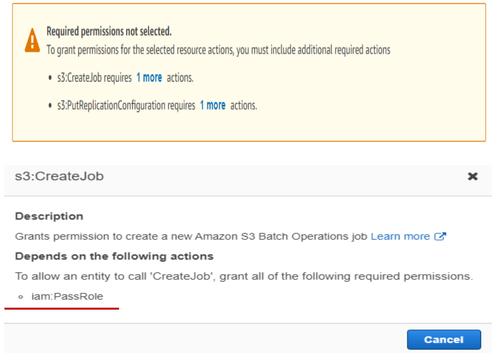
Switch to deny permissions

Manual actions | Add actions

☑ All S3 actions (s3:*)

Create customer managed policy

- A policy allows the IAM user to access a specified S3 bucket only.



Required permissions not selected.

To grant permissions for the selected resource actions, you must include additional required actions

- s3:CreateJob requires 1 more actions.
- s3:PutReplicationConfiguration requires 1 more actions.

s3:CreateJob

Description

Grants permission to create a new Amazon S3 Batch Operations job Learn more

Depends on the following actions

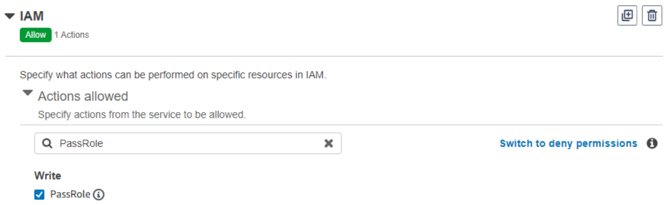
To allow an entity to call 'CreateJob', grant all of the following required permissions.

- iam:PassRole

Cancel

Create customer managed policy

- A policy allows the IAM user to access a specified S3 bucket only.



▾ IAM

Allow 1 Actions

Specify what actions can be performed on specific resources in IAM.

▾ Actions allowed

Specify actions from the service to be allowed.

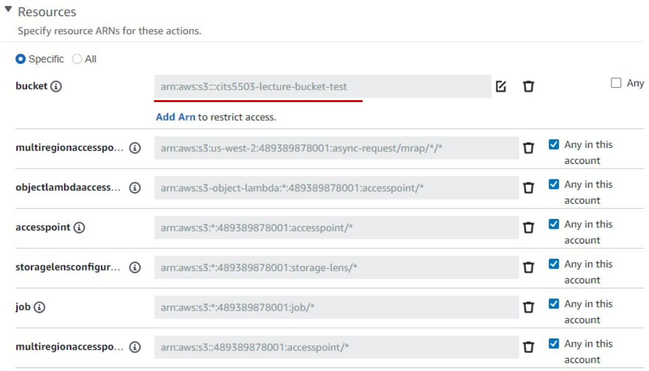
PassRole

Write

☑ PassRole

Create customer managed policy

- A policy allows the IAM user to access a specified S3 bucket only.



▾ Resources

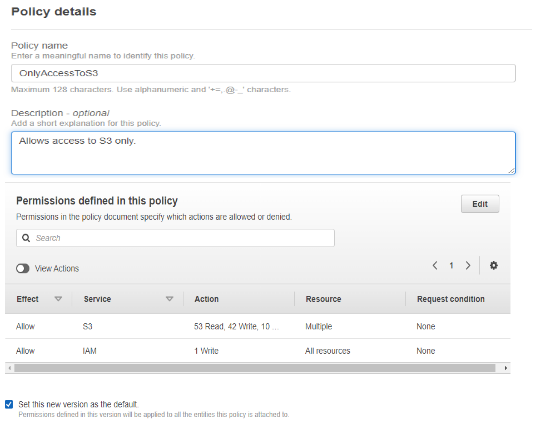
Specify resource ARNs for these actions.

● Specific ○ All

bucket	arn:aws:s3:::cits5503-lecture-bucket-test Add Arn to restrict access.	□ Any
multiregionaccesspo…	arn:aws:s3:us-west-2:489389878001:async-request/mrap/*/*	☑ Any in this account
objectlambdaaccess…	arn:aws:s3-object-lambda:*:489389878001:accesspoint/*	☑ Any in this account
accesspoint	arn:aws:s3:*:489389878001:accesspoint/*	☑ Any in this account
storagelensconfigur…	arn:aws:s3:*:489389878001:storage-lens/*	☑ Any in this account
job	arn:aws:s3:*:489389878001:job/*	☑ Any in this account
multiregionaccesspo…	arn:aws:s3::489389878001:accesspoint/*	☑ Any in this account

Create customer managed policy

- Review.



Policy details

Policy name
Enter a meaningful name to identify this policy.
OnlyAccessToS3
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Description - *optional*
Add a short explanation for this policy.
Allows access to S3 only.

Permissions defined in this policy

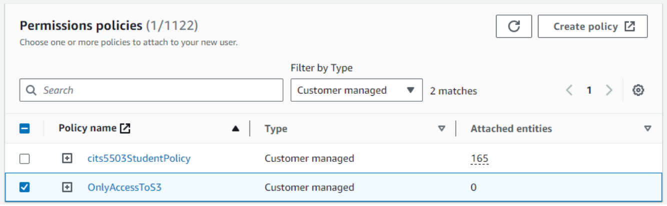
Permissions in the policy document specify which actions are allowed or denied.

Effect	Service	Action	Resource	Request condition
Allow	S3	53 Read, 42 Write, 10 …	Multiple	None
Allow	IAM	1 Write	All resources	None

☑ Set this new version as the default.
Permissions defined in this version will be applied to all the entities this policy is attached to.

Attach policies directly

- Select permission policy.



Permissions policies (1/1122)

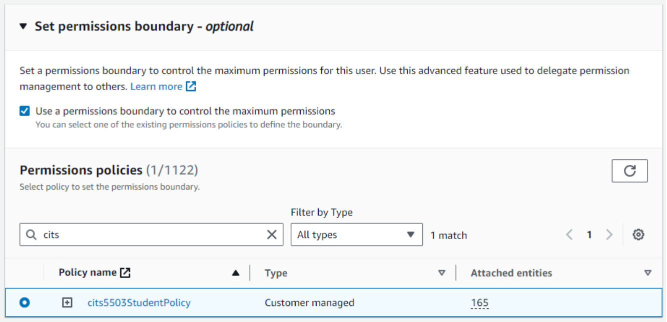
Choose one or more policies to attach to your new user.

Filter by Type: Customer managed ▾ 2 matches

	Policy name	Type	Attached entities
□	cits5503StudentPolicy	Customer managed	165
☑	OnlyAccessToS3	Customer managed	0

Attach policies directly

- Set permissions boundary.



▼ **Set permissions boundary - *optional***

Set a permissions boundary to control the maximum permissions for this user. Use this advanced feature used to delegate permission management to others. Learn more

☑ Use a permissions boundary to control the maximum permissions
You can select one of the existing permissions policies to define the boundary.

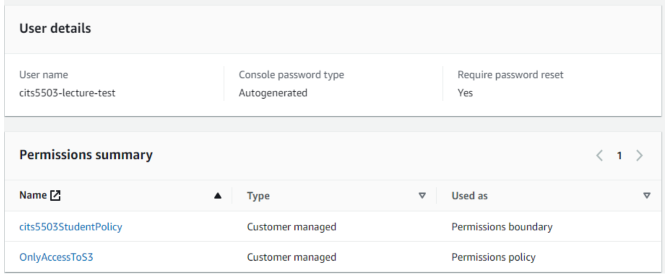
Permissions policies (1/1122)
Select policy to set the permissions boundary.

Search: cits
Filter by Type: All types
1 match

	Policy name	Type	Attached entities
●	cits5503StudentPolicy	Customer managed	165

Attach policies directly

- Review.



User details

User name	Console password type	Require password reset
cits5503-lecture-test	Autogenerated	Yes

Permissions summary

Name	Type	Used as
cits5503StudentPolicy	Customer managed	Permissions boundary
OnlyAccessToS3	Customer managed	Permissions policy

Practice Questions

- [6 marks] Q1: Name 3 of the keys in a Policy. Explain their role. An example of a key is “Version” that specifies the version of the policy syntax and is normally “Version”: “2012-10-17”

- [2 marks] **Statement**: represents a permission rule.
- [2 marks] **Effect**: what the effect will be when a user requests the specific action—this can be either **Allow** or **Deny**.
- [2 marks] **Action**: defines a set of resource operations a user/application is allowed (or denied) to perform.
- [2 marks] **Resource**: specifies AWS resources for which a user is allowed or denied to take actions.