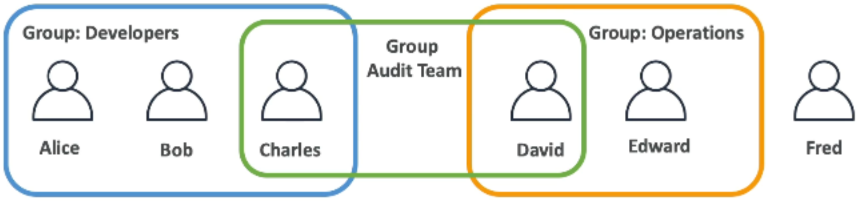
Identity and Access Management (IAM)

Student - Christina Grys
Instructor - Stephane Maarek

IAM

IAM: Users & Groups

- IAM = Identity and Access Management, Global service
- Root account created by default. No sharing.
- Users are people within your organization, they can be grouped to one or multiple
- Groups only contain users, not other groups
- Users don't have to belong to a group. User can belong to multiple groups



IAM: Permissions

- Users or Groups can be assigned JSON documents called policies
    - These policies define permissions of the users
- In AWS, you apply the least privilege principle: don't give more permissions than a user need.

IAM Policies

- Able to create own policies, add & remove users/groups
- MFA options with devices are accessible to that specific user
    - Able to change password policies under "Account settings"
    - Manage MFA device – virtual, device, QR scan code

- How can users access AWS?
    - 3 Options:
        - AWS Management Console (protected password + MFA)
        - AWS Command Line Interface (CLI): protected by access keys (displayed in console) *Never share these!*
        - AWS Software Developer Kit (SDK): for code: protected by access keys

What's AWS CLI? – Command-line Interact

- A tool that enables to interact with AWS services using commands in command-line shell
- Direct access to public APIs of AWS services
- Able to create our own scripts to manage the resources https://github.com/aws/aws-cli

What's AWS SDK? - AWS Software Development Kit

- Language-specific APIs (set of libraries)
- Enables you to access and manage AWS services programmatically
- Embedded with your application

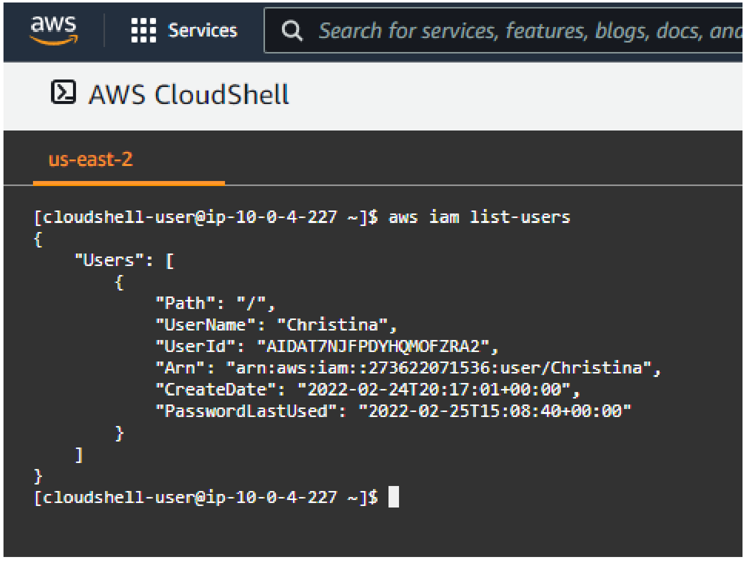
    - Supports many different programming languages:
        - SDKs (JavaScript, Python, PHP, .NET, Ruby, Java, Go, Node.js, C++)
        - Mobile SDKs (Android, IOS)
        - IoT Device SDKs (Embedded C, Arduino)
            - Ex: AWS CLI is built on AWS SDK for Python

Be sure to install aws cli on MacOS [terminal: script]

```
christinagrys@Christinas-MacBook-Pro ~ % aws configure
AWS Access Key ID [****************ZP4K]: AKIAT7NJFPDYD6YO6BTU
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-2
Default output format [None]:
christinagrys@Christinas-MacBook-Pro ~ % aws iam list-users
{
  "Users": [
    {
      "Path": "/",
      "UserName": "Christina",
      "UserId": "AIDAT7NJFPDYHQMOFZRA2",
      "Arn": "arn:aws:iam::273622071536:user/Christina",
      "CreateDate": "2022-02-24T20:17:01+00:00",
      "PasswordLastUsed": "2022-02-25T15:08:40+00:00"
    }
  ]
}
christinagrys@Christinas-MacBook-Pro ~ %
```

AWS CloudShell - terminal in the cloud of AWS (only available in some regions)

- Whenever using CLI, it is going to return you an API call



```
[cloudshell-user@ip-10-0-4-227 ~]$ aws iam list-users
{
    "Users": [
        {
            "Path": "/",
            "UserName": "Christina",
            "UserId": "AIDAT7NJFPDYHQMOFZRA2",
            "Arn": "arn:aws:iam::273622071536:user/Christina",
            "CreateDate": "2022-02-24T20:17:01+00:00",
            "PasswordLastUsed": "2022-02-25T15:08:40+00:00"
        }
    ]
}
[cloudshell-user@ip-10-0-4-227 ~]$
```

IAM Roles for AWS Services

- There will be a few AWS services that will need to perform actions on your behalf.
    - Assign permissions to AWS services with IAM Roles
    - Common Roles:
        - EC2 Instance Roles
        - Lambda Function Roles
        - Roles for CloudFormation

IAM Security Tools

IAM Credentials Report (account-level)

- A report that lists all your account's users and the status of their various credentials

IAM Access Advisor (user-level)

- Access advisor shows the service permissions granted to a user and when those services were last accessed.
- This information can be used to revise the policies too.
- Users > name > Access Advisor > see list on last accessed column

IAM Guidelines & Best Practices

1. Don't use the root account except for AWS account setup
2. One physical user = one AWS user
3. Assign users to groups and assign permissions to groups
4. Create strong password policy
5. Use and enforce the use MFA
6. Create and use Roles for giving permissions to AWS services
7. Use Access Keys for Programmatic Access (CLI/SDK)
8. Audit permissions of your account with the IAM Credentials Report
9. Never share IAM users and Access Key

IAM SUMMARY SECTION:

- **Users:** mapped to a physical user, has a password for AWS Console
- **Groups:** contains users only
- **Policies:** JSON document that outlines permissions for users or groups
- **Roles:** for EC2 Instances or AWS services
- **Security:** MFA + Password Policy
- **Access Keys:** access AWS using the CLI or SDK
- **Audit:** IAM Credential Reports & IAM Access Advisor