

Assignment Name: **Week 02**
Student Name: **Sylvain S. Kamdem**

TOPIC: AUTHENTICATION AND AUTHORIZATION WITH AWS IDENTITY AND ACCESS MANAGEMENT.

Knowledge Summary:

- A. IAM is used for managing user access to AWS resources** - IAM allows to create and manage AWS users and groups, and assign them permissions to access specific AWS resources.
- B. IAM uses policies to control access** - IAM policies define what actions a user or group can perform on AWS resources. Policies can be attached to users, groups, and roles.

Policy type	Function
Identity-based	Attach managed and inline policies to IAM identities (users, groups to which users belong, or roles)
Resource-based	Attach inline policies to resources
Permission boundaries	Use a managed policy as the permissions boundary for an IAM entity (user or role)
Organizations SCPs	Use an AWS Organizations service control policy (SCP) to define the maximum permissions for account members of an organization or organizational unit (OU)
Access Control Lists (ACLs)	Use ACLs to control which principals in other accounts can access the resource to which the ACL is attached. Don't use JSON policy document structure. They are cross-account permission policies.
Session	Pass advanced session policies when you use the AWS CLI or AWS API to assume a role or a federated user

JSON: Most policies are stored in AWS as **JSON** (JavaScript Object Notation) documents:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "FirstStatement",
      "Effect": "Allow",
      "Action": ["iam:ChangePassword"],
      "Resource": "*"
    },
    {
      "Sid": "SecondStatement",
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    },
    {
      "Sid": "ThirdStatement",
      "Effect": "Allow",
      "Action": [
        "s3:List*",
        "s3:Get*"
      ],
      "Resource": [
        "arn:aws:s3:::confidential-data",
        "arn:aws:s3:::confidential-data/*"
      ],
      "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
    }
  ]
}

```

Figure 1 - Example of JSON document

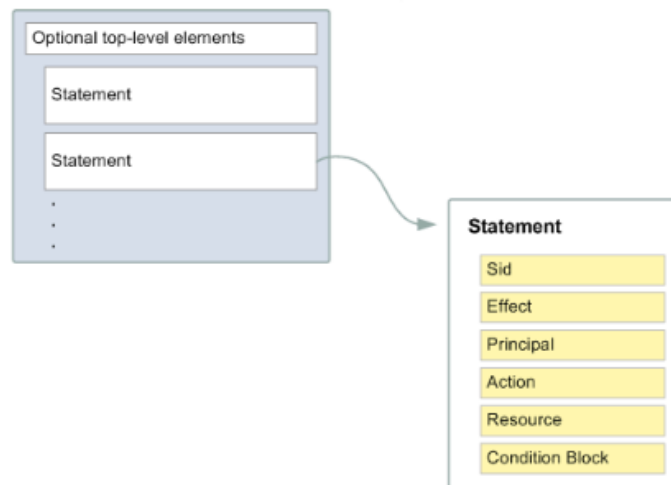


Figure 2 - Example of JSON document structure

Version	version of the policy language, use the latest 2012-10-17 version
Statement	container for the following elements
Sid (Optional)	optional statement ID to differentiate between your statements
Effect	Allow or Deny to indicate whether the policy allows or denies access
Principal (sometimes required)	If creating a resource-based policy , you must indicate the account, user, role, or federated user to which you would like to allow or deny access.

	If creating an IAM permissions policy to attach to a user or role, you cannot include this element . The principal is implied as that user or role.
Action	Include a list of actions that the policy allows or denies
Resource (sometimes required)	If creating an IAM permissions policy , you must specify a list of resources to which the actions apply. If you create a resource-based policy , this element is optional .
Condition (Optional)	Specify the circumstances under which the policy grants permission

- C. **IAM provides security best practices** - IAM provides a number of security best practices, such as requiring strong passwords, enabling MFA, and rotating access keys. When you create IAM policies, follow the standard security advice of **granting *least privilege***, or granting only the permissions required to perform a task.
- D. **IAM is a free service** - There is no additional cost to use IAM, and you can create as many users, groups, and roles as you need.
- E. **IAM has granular access control** - IAM allows you to grant users and groups permissions at a granular level, allowing you to provide access to specific resources or actions.
- F. **IAM is integrated with other AWS services** - IAM integrates with other AWS services, such as S3 and EC2, allowing you to control access to these services using IAM.
- G. **IAM has a learning curve** - IAM can be complex, especially for beginners. It's important to take the time to learn IAM and best practices for managing user access to AWS resources.

Overall, AWS IAM is a powerful tool for managing access to AWS resources and ensuring the security of your cloud infrastructure. As a beginner, it's important to take the time to learn IAM and best practices for managing user access to AWS resources.

Lab:

The screenshot shows the AWS IAM dashboard. On the left is a navigation menu with sections: Identity and Access Management (IAM), Access management, Access reports, and Credential report. The main content area is titled 'IAM dashboard' and includes 'Security recommendations' with two green checkmarks: 'Root user has MFA' and 'Root user has no active access keys'. Below this is a red warning icon for 'Update your access permissions for AWS Billing, Cost Management, and Account consoles'. At the bottom, a table shows 'IAM resources': 1 User group, 2 Users, 14 Roles, 1 Policy, and 0 Identity providers. On the right, the 'AWS Account' section shows the account ID and alias, and the 'Quick Links' section includes 'My security credentials' and 'Tools'.

Figure 3 - My Root Account with MFA Enabled

The screenshot shows the 'IAM User and Role Access to Billing Information' settings page. It lists two regions: 'US West (N. California)' and 'US West (Oregon)', both with the status 'Enabled by default'. Below this, the 'IAM User and Role Access to Billing Information' section explains that the 'Activate IAM Access' setting allows IAM users and roles access to Billing and Cost Management console pages. It lists the console pages affected: AWS Cost Anomaly Detection, Savings Plans overview, Savings Plans inventory, Purchase Savings Plans, and Savings Plan cart. The 'Activate IAM Access' checkbox is checked, and there are 'Update' and 'Cancel' buttons. Below this is the 'Reserved Instance Marketplace Settings' section.

Figure 4 - Enabling Billing access to my IAM Users (from my root)

The screenshot shows the AWS IAM console interface. The left sidebar contains navigation links for Identity and Access Management (IAM), Access management, and Access reports. The main content area displays the details of a selected IAM user. The 'Summary' section shows the user's ARN, console access status (Disabled), and two access keys. The 'Permissions' section shows that the user has the 'AdministratorAccess' policy attached directly.

Summary

ARN	Console access	Access key 1
[Redacted]	Disabled	[Redacted] - Active
Created	Last console sign-in	Used Yesterday. 31 days old.
[Redacted]	-	Access key 2
		Not enabled

Permissions policies (1)

Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Directly

Figure 5 - My IAM User with Admin Access

The screenshot shows the AWS Billing Dashboard. The 'AWS summary' section provides a high-level overview of billing metrics. The 'Highest cost' section details the highest service spend, which is for Elastic Compute Cloud.

AWS summary

Metric	Value
Current month's total forecast	USD 0.81
Current MTD balance	USD 0.67
Prior month for the same period with trend	No data to display ↓ 0.0%
Total number of active services	3
Total number of active AWS accounts	1
Total number of active AWS Regions	2

Highest cost

Metric	Value
Service name	Elastic Compute Cloud
Trend compared to prior month	↓ 0.0%
Current MTD balance	USD 0.63
Prior month for the same period	No data to display

Figure 6 - Billing dashboard from my IAM User

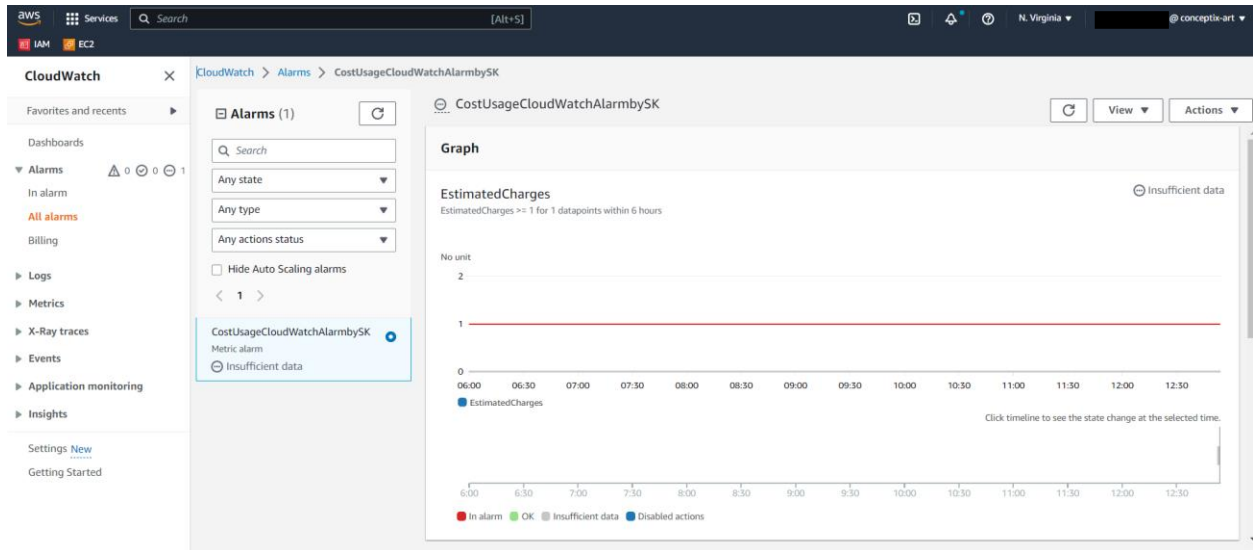


Figure 7 - CloudWatch Alarm for Cost Usage (IAM User)