

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

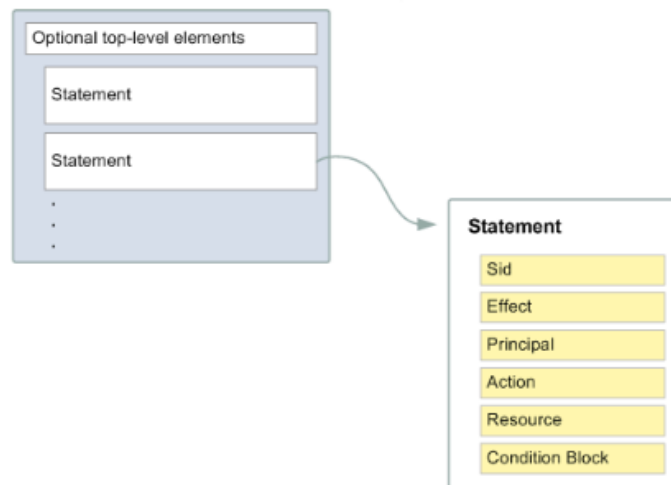


Figure 2 - Example of JSON document structure

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 warning about updating 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 displays account details like ID, Alias, and Sign-in URL. Below that are 'Quick Links' for security credentials and a 'Tools' section with a policy simulator.

IAM resources				
User groups	Users	Roles	Policies	Identity providers
1	2	14	1	0

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 'Enabled by default' status. Below this, a section titled 'IAM User and Role Access to Billing Information' explains the 'Activate IAM Access' setting. It states that this setting allows IAM users and roles access to Billing and Cost Management console pages. A list of affected console pages and SDK APIs is provided. At the bottom, the 'Activate IAM Access' checkbox is checked, and there are 'Update' and 'Cancel' buttons.

US West (N. California) Enabled by default

US West (Oregon) Enabled by default

▼ IAM User and Role Access to Billing Information

Use the **Activate IAM Access** setting to allow IAM users and roles access to pages of the Billing and Cost Management console. This setting alone doesn't grant IAM users and roles the necessary permissions for these console pages. In addition to activating IAM access, you must also attach the required IAM policies to those users or roles. For more information, see [Granting access to your billing information and tools](#).

If this setting is deactivated, then IAM users and roles in this account can't access the Billing and Cost Management console pages, even if they have administrator access or the required IAM policies.

The **Activate IAM Access** setting does not control access to:

- The console pages for AWS Cost Anomaly Detection, Savings Plans overview, Savings Plans inventory, Purchase Savings Plans, and Savings Plan cart
- The Cost Management view in the AWS Console Mobile Application
- The Billing and Cost Management SDK APIs (AWS Cost Explorer, AWS Budgets, and AWS Cost and Usage Report APIs)
- The Customer Carbon Footprint Tool on the Cost & Usage Reports console page

☒ **Activate IAM Access**

Update Cancel

▼ Reserved Instance Marketplace Settings

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

The screenshot shows the AWS IAM console for the user 'admin-terraform'. The left sidebar lists navigation options under 'Identity and Access Management (IAM)' and 'Access management'. The main content area shows the user's summary, including their ARN, console access status (disabled), and access keys. Below the summary, the 'Permissions' tab is selected, showing a table of permissions policies. One policy, 'AdministratorAccess', is listed as being attached directly to the user.

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 for the user 'admin-terraform @ conceptis-art'. The dashboard displays the current month's total forecast, current MTD balance, and prior month's total for the same period with trend. It also shows the total number of active services, active AWS accounts, and active AWS Regions. The 'Highest cost' section highlights the service with the highest spend, 'Elastic Compute Cloud', and shows its trend compared to the prior month and current MTD balance.

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

Highest cost			
Viewing highest service spend.			
Service name	Trend compared to prior month	Current MTD balance	Prior month for the same period
Elastic Compute Cloud	↓ 0.0%	USD 0.63	No data to display

Figure 6 - Billing dashboard from my IAM User

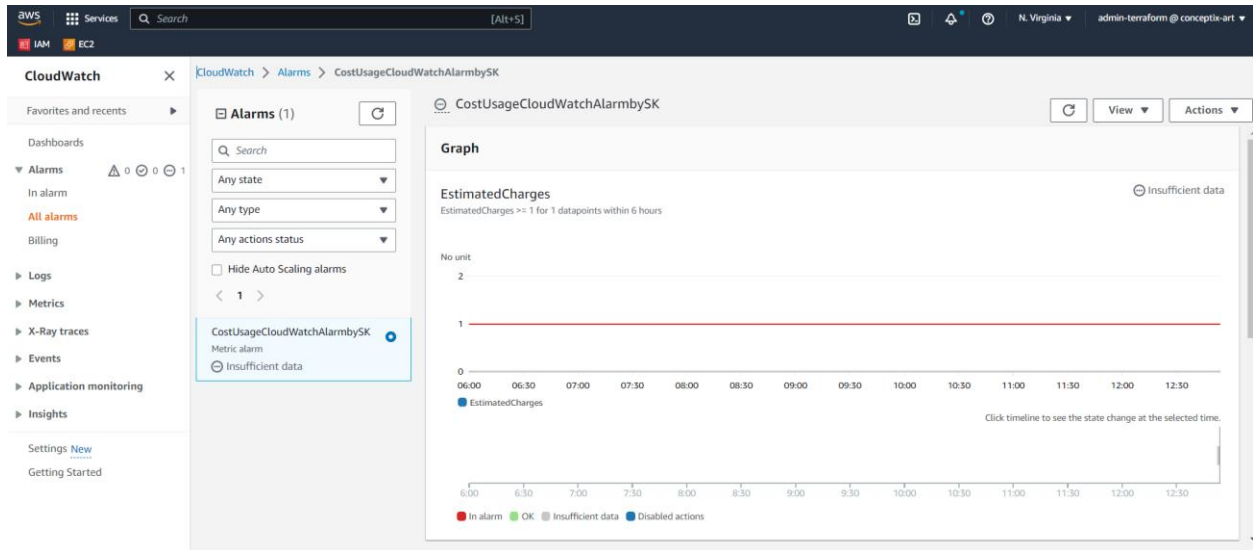


Figure 7 - CloudWatch Alarm for Cost Usage (IAM User)