Notes

aws

AWS Project Documentation

Cloud Securitywith AWS IAM

IAM

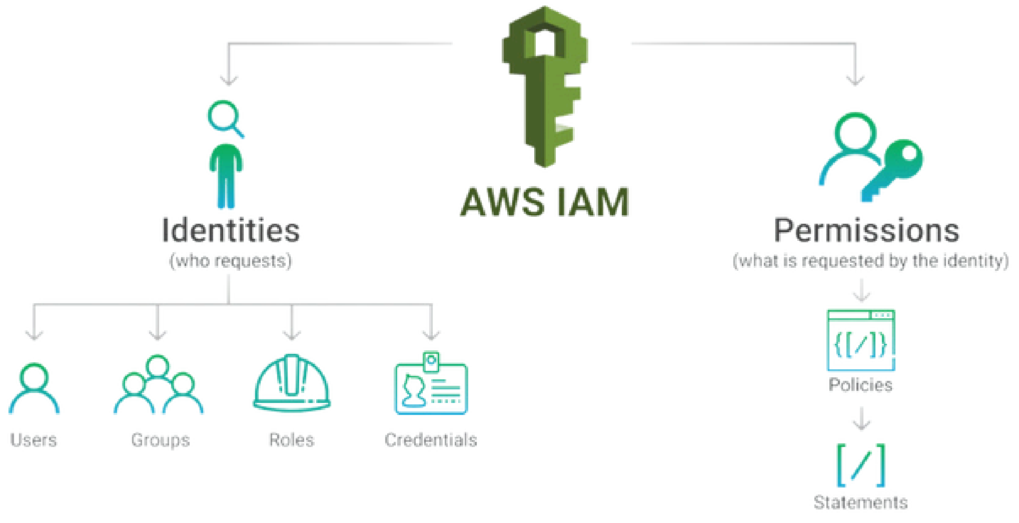
@sAnandha

Notes

AWS IAM (IDENTITY ACCESS MANAGEMENT)

Features of AWS IAM :

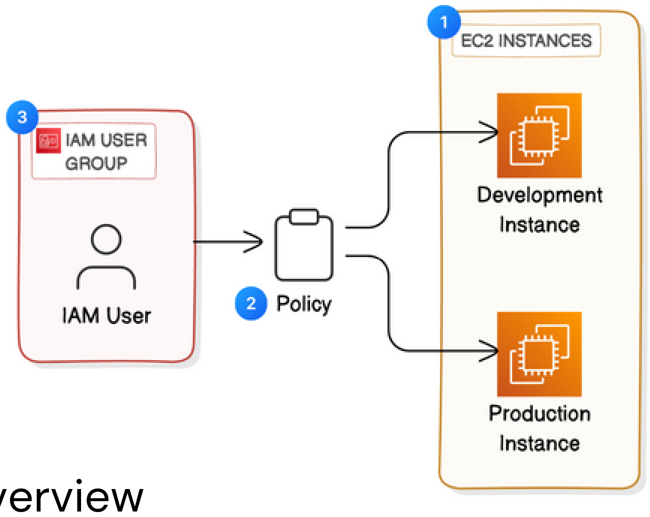
1. Fine-grained Permissions
2. You can assign specific permissions to users, groups, or roles, ensuring minimal privilege access.
3. Authentication and Authorization
4. IAM verifies the identity of users and determines their access rights to AWS services.
5. Manage access for AWS users, applications, federated users (via SSO), and third-party systems.
6. Role-Based Access Control (RBAC)
7. Assign roles to users or services for temporary or cross-account access.
8. Policy-based Management
9. Use JSON-based policies to define what actions are allowed or denied for a specific resource.
10. Integration with Other AWS Services



@sAnandha

Notes

Architecture Diagram :



Architecture Overview

1. **EC2 Instances (Right-Side):**
    - These are the compute instances (Development and Production environments) running on AWS. Access to these instances is controlled by IAM policies.
    - Development Instance: Used for testing and staging applications.
    - Production Instance: Used for hosting the live application.
2. **Policy (Center):**
    - The policy acts as the permission document defining what the user or group can do. For example, it might allow users to start, stop, or terminate instances.
    - Policies can be applied to individual users or groups.
3. **IAM User and Group (Left-Side):**
    - **IAM User:** Represents a specific individual (e.g., a developer or admin) who needs access to AWS services.
    - **IAM Group:** A collection of users. Instead of assigning policies individually, permissions are granted by attaching policies to the group.
    - Each user in the group inherits the group's permissions.

@sAnandha

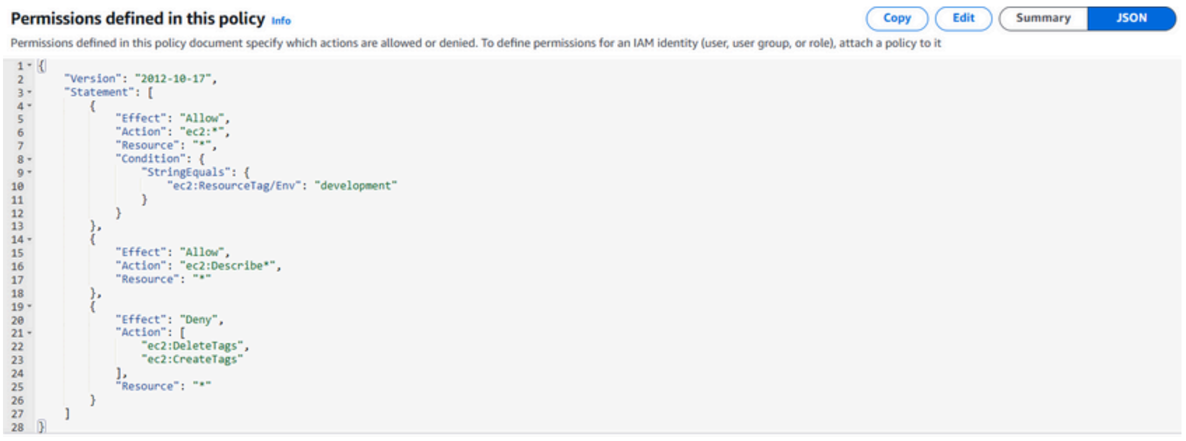
Notes

Process Flow to Implement

1. Step 1: **Create EC2 Instances**
    - Launch EC2 instances for Development and Production environments.
    - Ensure the required security groups and key pairs are configured for SSH or application access.
2. Step 2: **Define a Policy**
    - Write an IAM policy to control access to the EC2 instances. For example:
        - Allow access to start and stop specific instances.
        - Deny access to terminate production instances.
3. Step 3: **Create IAM Groups**
    - Create groups based on roles (e.g., Developers, Admins).
    - Attach the policy to the relevant group.
4. Step 4: **Add IAM Users**
    - Create individual IAM users for your team members (e.g., developer1, admin1).
    - Assign each user to the appropriate group (e.g., Developers group).
5. Step 5: **Test Access**
    - Users log in with their AWS credentials (via AWS CLI, SDK, or Console).
    - Test permissions by trying to access the EC2 instances. For example:
        - Developers should only access Development instances.
        - Admins can access both Development and Production instances.

Notes

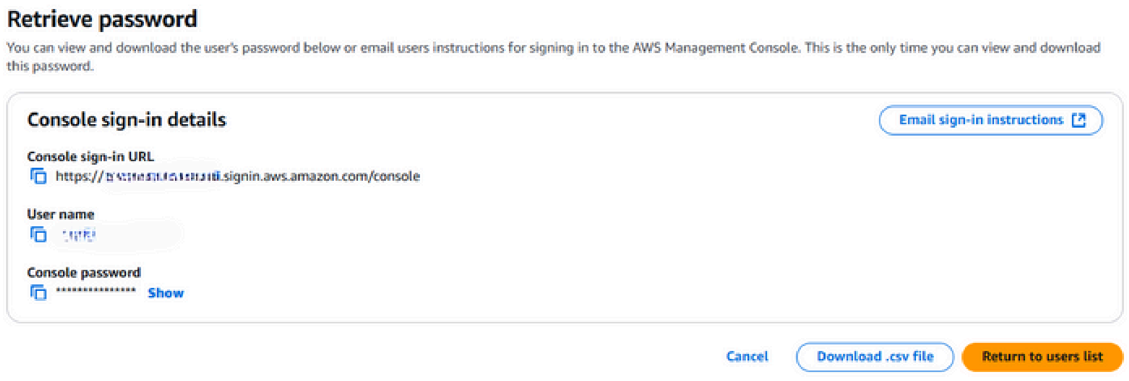
Sample Output:



Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Env": "development"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ec2:DeleteTags",
                "ec2:CreateTags"
            ],
            "Resource": "*"
        }
    ]
}
```



Retrieve password

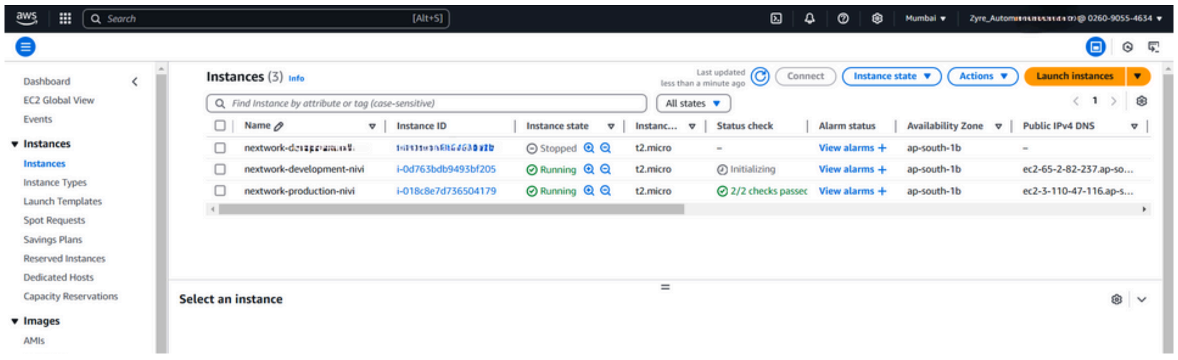
You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

Console sign-in URL
https://[redacted].signin.aws.amazon.com/console

User name
[redacted]

Console password
*************** Show



Zyre_Autom[redacted]@ 0260-9055-4634

Instances (3) Info

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instanc...	Status check	Alarm status	Availability Zone	Public IPv4 DNS
nextwork-d[redacted]	[redacted]	Stopped	t2.micro	-	View alarms +	ap-south-1b	-
nextwork-development-nivi	i-0d763bdb9493bf205	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-65-2-82-237.ap-so...
nextwork-production-nivi	i-018c8e7d736504179	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-3-110-47-116.ap-s...

Select an instance

For Detialed Description:
https://learn.nextwork.org/projects/aws-security-iam

@sAnandha