Agenda for Today

1. AWS Regions and Availability Zones
2. Creating an AWS Account
3. Identity and Access Management
4. IAM Identity Centre (Previously called SSO)
5.Advanced IAM Configuration and Best Practices
6. Hands on Lab
7. QA

AWS Regions and Availability Zone

Regional vs Global AWS Services

US Region

India Region

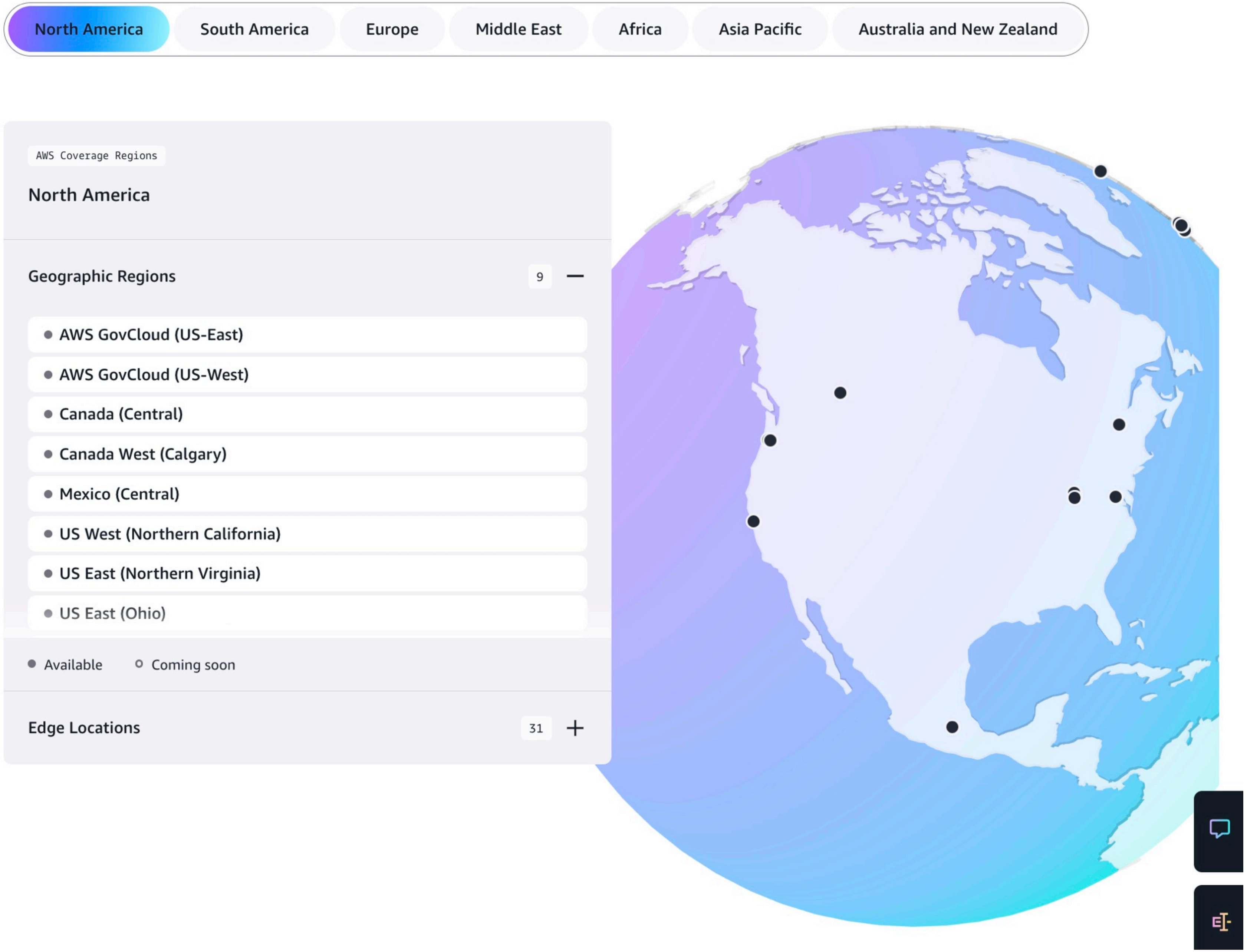
↳ Mumbai
ap-south-1

↳ Hyderabad
ap-south-2

UK Region

```
aws ec2 describe-regions
aws ec2 describe-availability-zones
aws ec2 describe-availability-zones --region us-east-1
```

https://aws.amazon.com/about-aws/global-infrastructure/regions_az/



AWS Coverage Regions

North America

Geographic Regions 9 —

- ● AWS GovCloud (US-East)
- ● AWS GovCloud (US-West)
- ● Canada (Central)
- ● Canada West (Calgary)
- ● Mexico (Central)
- ● US West (Northern California)
- ● US East (Northern Virginia)
- ● US East (Ohio)

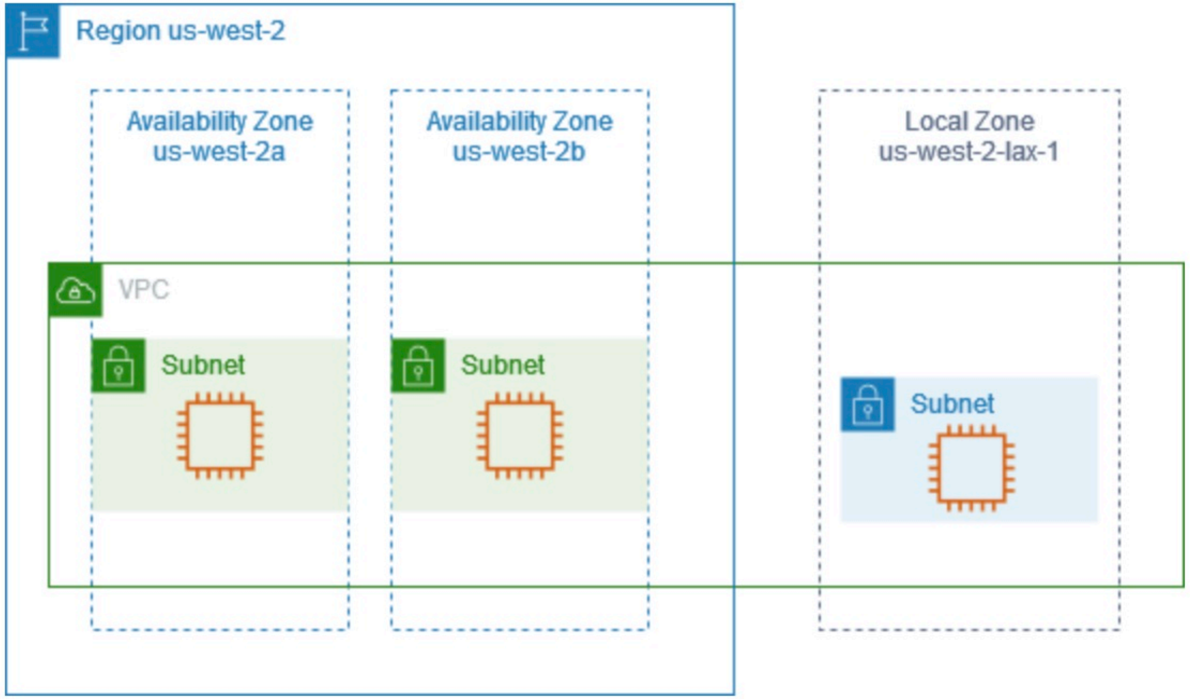
● Available ○ Coming soon

Edge Locations 31 +

A Local Zone is an extension of an AWS Region in geographic proximity to your users. Local Zones have their own connections to the internet and support AWS Direct Connect, so that resources created in a Local Zone can serve applications that require low latency.

To use a Local Zone, you must first enable it. Next, you create a subnet in the Local Zone. Finally, you launch resources in the Local Zone subnet. For more detailed instructions, see Getting started with AWS Local Zones.

The following diagram illustrates an account with a VPC in the AWS Region `us-west-2` that is extended to the Local Zone `us-west-2-lax-1`. Each zone in the VPC has one subnet, and each subnet has one EC2 instance.


Region us-west-2
Availability Zone us-west-2a
Availability Zone us-west-2b
Local Zone us-west-2-lax-1
VPC
Subnet
Subnet
Subnet

IAM

- ↳ Identity and Access Management
- ↳ Creating a user
  - ↳ Permission (Policy)
- ↳ IAM Group
  - ↳ Developer ←
  - ↳ Admin
  - ↳ DBA
  - ↳ Security

AWS Mgmt Console

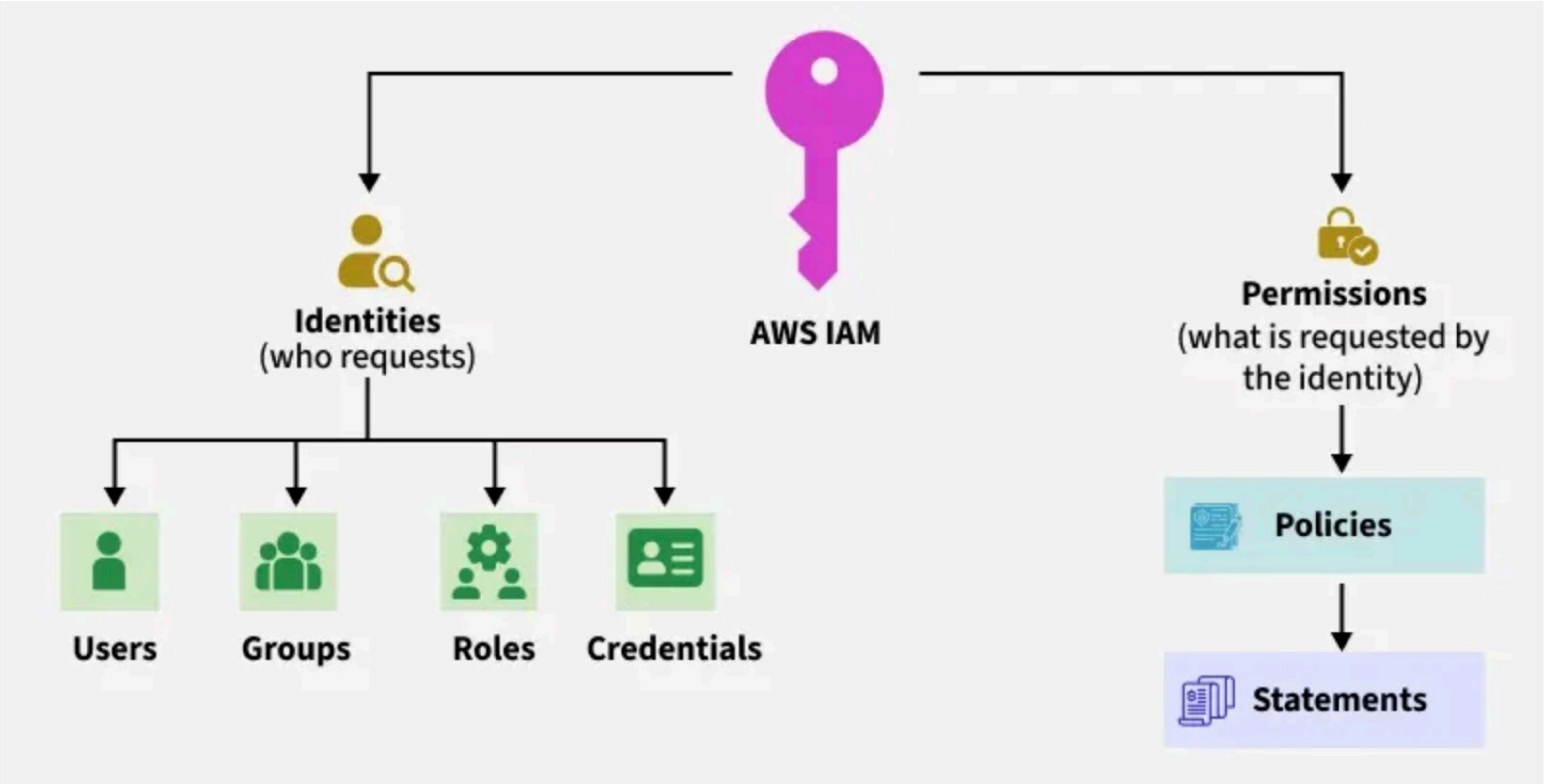
Programmatic Access

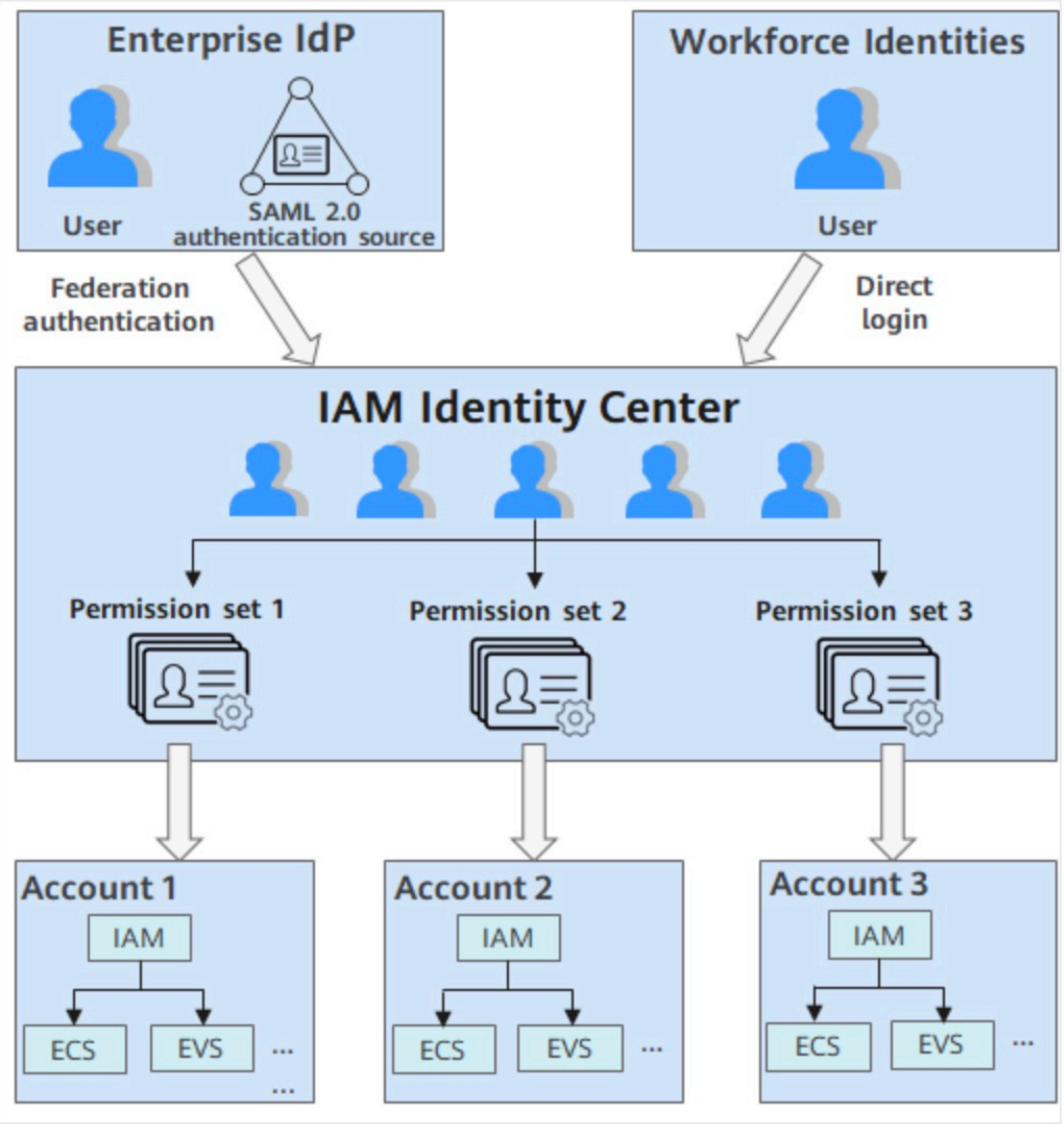
- ↳ CLI
- ↳ SDK
- ↳ Python

Policy

↳ A JSON structure to define the kind of resource access someone can have.

- AWS Managed
- Customer Managed
- Inline policy


AWS IAM
Identities
(who requests)
Users
Groups
Roles
Credentials
Permissions
(what is requested by the identity)
Policies
Statements


Enterprise IdP
User
SAML 2.0
authentication source
Workforce Identities
User
Federation
authentication
Direct
login
IAM Identity Center
Permission set 1
Permission set 2
Permission set 3
Account 1
IAM
ECS
EVS
...
...
Account 2
IAM
ECS
EVS
...
Account 3
IAM
ECS
EVS
...

1. Regions:
AWS Regions are separate geographic areas that host multiple Availability Zones to provide high availability and fault tolerance.

2. Availability Zone (AZ):
An AZ is a physically isolated data center within a region, connected through low-latency links for high availability.

3. Edge Locations:
Edge Locations are endpoints in the AWS global network used by CloudFront and Route 53 to cache content closer to users.

4. Local Zone:
Local Zones extend AWS services closer to large population centers for ultra-low latency use cases like gaming and video editing.

5. Outpost:
AWS Outposts bring AWS infrastructure and services to on-premises locations for a consistent hybrid experience.

6. Wavelength:
AWS Wavelength embeds compute and storage services at telecom providers' 5G networks for ultra-low latency applications.

IAM (Identity & Access Management)

-> An AWS Identity and Access Management (IAM) user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS Services.

-> AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources.

We can use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

-> IAM helps protect against security breaches by allowing administrators to automate numerous user account related tasks.

-> Best practice of using the root user only to create your first IAM user.

-> Enable Multi Factory Authentication (MFA) for Root Usr

-> By using Google Authenticator App we can configure "Virtual MFA"

Best Practices:

- When we login AWS using 'email' and 'password', that has complete access to all AWS services and resources in the account (Root account).

- Strongly recommended that you do not use the "root user" for your everyday tasks, even the administrative ones.

- Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

- IAM user is truely global, i.e, once IAM user is created it can be accessible in all the regions in AWS.

- Amazon S3 also considered as Global but, it is not truely global. When we create a bucket in S3 it displays all the buckets of other regions in one place , so that is the reason we are calling AmazonS3 is Global (but partly global).

- But IAM is 100% Global. Once you create IAM user you can use it anywhere in all the regions.

1. Main things in IAM is
    - Roles
    - Users
    - Policies / Permissions
    - Groups

2. IAM users can be accessible by the following 3 ways.
    - through AWS console
    - CLI (Command Line Interface)
    - API (fast glaciers)

3. In MNCs , permissions will not be provided for individual users. Create the Groups and add the users into it.

Users & Groups are for the Endusers.
Roles are for the AWS Services.

Steps:
====

```
1. Create an IAM user
   Services - Security, Identity, & Compliance - IAM
   Users---<Add user>
      User name* = Iamuser1
      Access type  = 'select' both "Programmatic Access"
                       "AWS Management Console access"


      Console password = 'select'
                     custom Password =  (********somepassword eg:test1234)

            click <NextPermissions>
      (Note: we are not providing any permissions as of now, just <create user>)

      Once the IAM user has been created.
            AccessKeyID =AKIAIEJH7Z3FDKH36YWQ
            Secretaccesskey=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

      (Note: Once you close this window, AccessKeyID and Secret Accesskey has gone,
so save it somewhere)

  - Best Practice is never give an individual permissions to the user, as users
will be changed frequently, when they left the organization.
    So Need to create the Groups and assign the users to it.

2. Group
      <create new group>
      Groupname =admins
      (Note: no need add any policy now).
      <creategroup>

3. Add user to this group

      click on newly create group 'admins'
      <Add users to Group>

      GroupARN =arn:aws:iam::540105522204:group/admins
```

-Always add the permissions to the 'Groups' level not to the 'users' level. Its a Best Practice in the real-time.

****
Policies:
****

- When we want to add the permissions to the the groups is through the 'Policies'.
- Default AWS Policies are appear in'Orange color Icons'
- One disadvantage of AWS Default Policies are , we can't customize the policies to apply to the Groups.

- To provide customized policies to apply to Groups, we need to create the new one and apply to the Groups.

4. Now, we will add 'Administrator Access' Permissions to the user(Iamuser1) we create.

Groups -Admins-tab<permissions> ---<AttachPolicy>---'select' Administrator Access----<AttachPolicy>

-Dashboard -Customize the IAM link replacing the ID with any name. To Hide the ID need to customize.

IAM user sigin in

https://4234324234.signin.aws.amazon.com/console

After Customize

https://classroomuser.signin.aws.amazon.com/console

- Open the new tab in the browser
https://classroomuser.signin.aws.amazon.com/console

IAMuser =Iamuser1
password=test1234

5. Now need to login using the IAM user, which we created.

Once login , we can launch an EC2 instance.As this user(Iamuser1) is provided with Admin access.

=================
Requirement:
=================

I got an requirement to create a new user and he should be able to do only 'stop' and 'start' , 'reboot' select instances only.

He should not have the permissions to terminate the EC2 Instances.

He should not have the permissions to create the new EC2 Instance.

1. Login to your AWS Console with your root login.

2. IAM -Create another user
User name* = Iamuser2
Access Type ='select' "AWS Management Console access"

'select' CustomPassword ="<somepassword>"
<NextPermissions>
Not selecting any group here
<createuser>

3. Signout and Login using the 'Iamuser2' and its credentials

Open browser
https://classroomuser.signin.aws.amazon.com/console
login with Iamuser2 credentials

Services ---EC2
you will get an 'Authorization Error'

4. To view EC2 instances need to provide read permission to the user 'Iamuser2'.
- using Tags, we can provide permissions to this user.

```
 Login using the Root user
  EC2 Instances
  Select the Running Instance
   click on tab <Tags>
        add new tag
           Key =user
           Value=Iamuser2
            <save>
```

5. Using this we can restrict the user to create EC2 instances. We can allow him to do only 'stop' and 'start' Instances.
For this, need to write the custom scripts.

Open the browser search for ='restrict aws user ec2 instance'
https://aws.amazon.com/premiumsupport/knowledge-center/restrict-ec2-iam/

copy the script and open in any editor and customize it.
arn:aws:ec2:us-east-1:111122223333:instance/*"
(Note: For every service we have arn (amazon resource name), but for EC2 there is no arn naming)

InterviewQuestion:If anyone ask you , arn is not displaying for the EC2 instances?
Ans:Simply say that, ARN is not visible for the EC2 instances, but for the other services like S3, we have ARN url.

copy the script

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:RebootInstances"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Owner": "Bob"
                }
            },
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:instance/*"
            ],
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:Describe*",
            "Resource": "*"
        }
    ]
}
```

After Customization
===================

```
      {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:RebootInstances"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/user": "Iamuser2"
                }
            },
            "Resource": [
                "arn:aws:ec2:us-east-1:449938344550:instance/*"
            ],
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:Describe*",
            "Resource": "*"
        }
    ]
}
```

=============

Note:
449938344550 = Root AccountID

6. copy the script after customization
IAMUser
Policies ---<createPolicy>---'select' JSON tab
paste the customized script.

<ReviewPolicy>

7. Review Policy
Name ='UserRestrictEC2Instance'
<createpolicy>

8. Now, need to add this policy to the user or groups.
select Users
'Iamuser2' ---Permissions(tab)-- Add Permissions ---AttachExisting Policies directly
Filter policies ='UserRestrictEC2Instance'

Select the policy(UserRestrictEC2Instance') ---<Review>--<AddPermissions>

```
9. Login to IAM user console
      Iamuser2/password

   - Now Try to Terminate the EC2 Instance. It throws an error
   - Try to Launch an EC2 instance , it throws an error.
     Like this we can restrict the user by creating some policies and apply to it.
     AWS provides the readymade(default) policies we need to customize as per our
requirement.
```

What is IAM ?
What is Root Account ?
How to enable MFA for root account

What is IAM account
How to create IAM account
Programmatic Access Vs Console Access
Attaching Policies to User
Creating Custom Policy
Creating User Group
Adding Users to Group
Adding Policies to User Group
What is IAM Role

Interview Questions on AWS Regions and Availability Zone

Basic Level

What is an AWS Region?
What is an Availability Zone (AZ) in AWS?
What is the purpose of Edge Locations in AWS?
How is a Local Zone different from an Availability Zone?
How many Availability Zones can a Region have?
Can a single Region have multiple Local Zones?
What services typically use Edge Locations?
Why does AWS recommend spreading resources across multiple AZs?
Which AWS services are global in scope and not tied to a Region?
How do you choose the right AWS Region for your application?

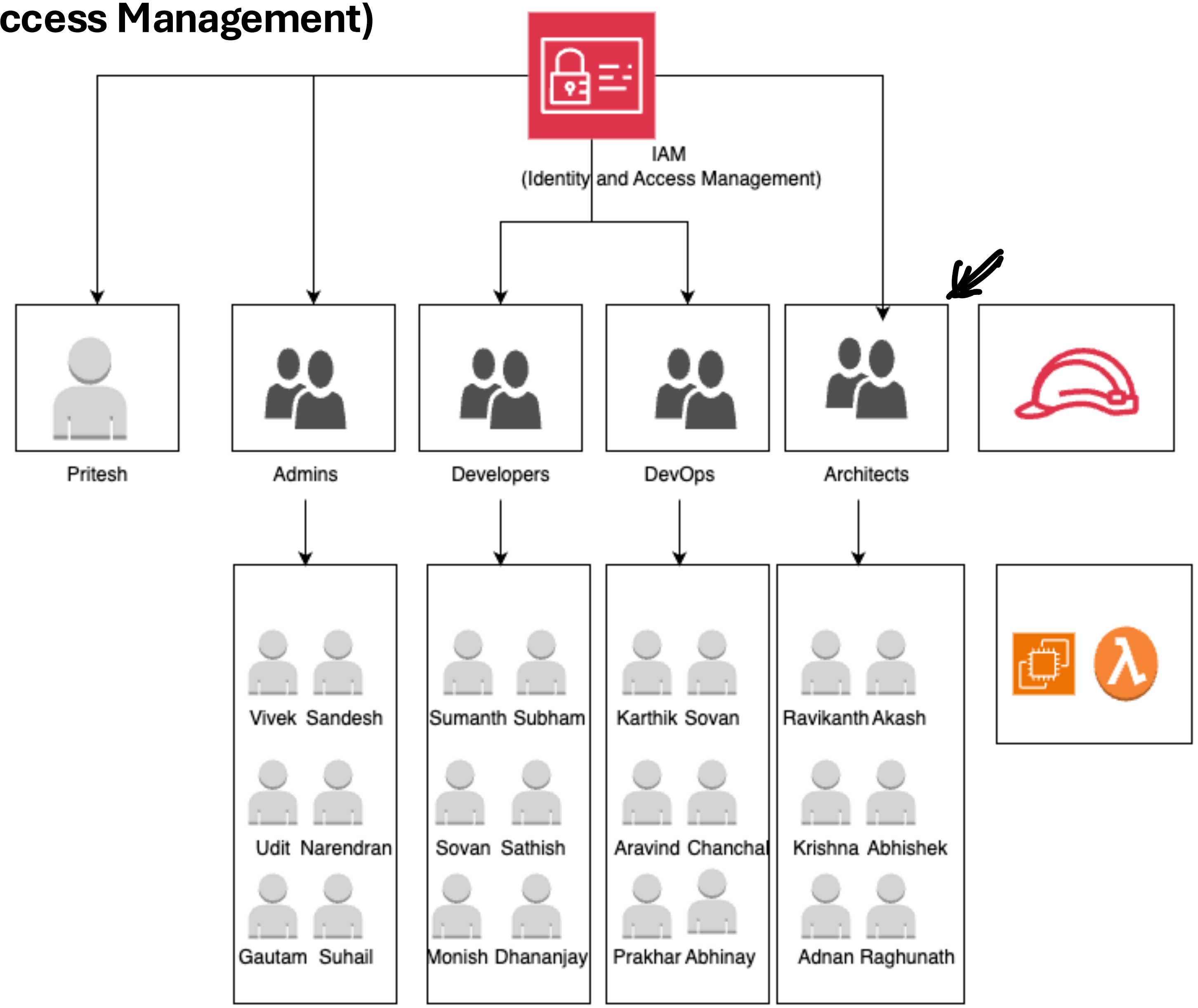
Intermediate Level

Explain the difference between a Region and an Availability Zone with an example.
How do Edge Locations improve latency for AWS services?
What's the role of Local Zones in hybrid cloud scenarios?
Which AWS services support deployment in Local Zones?
Can you run an EC2 instance in a Local Zone? How?
What is the impact of choosing a single-AZ deployment vs multi-AZ?
How does Route 53 use Edge Locations to resolve DNS queries faster?
How do Regions, AZs, and Edge Locations affect disaster recovery (DR) planning?
What considerations would you make when deploying in multiple Regions?
What are the cost implications of using Local Zones or Edge Locations?

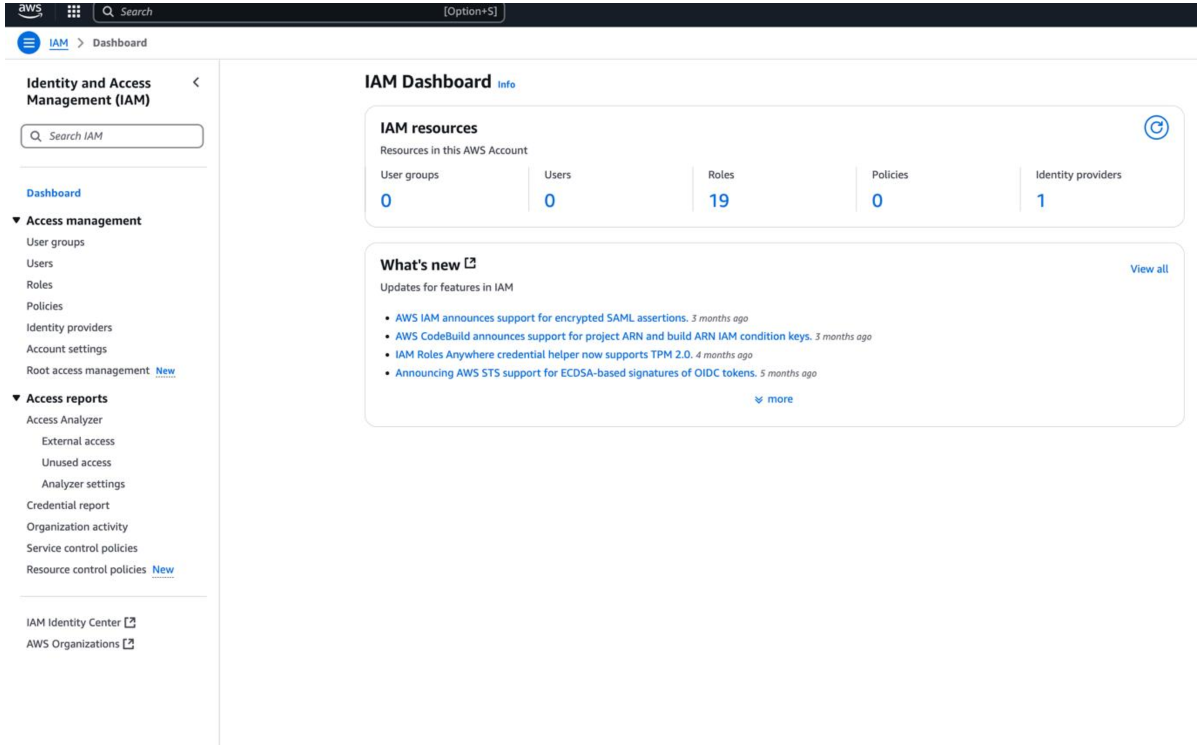
Architect Level

Design a high-availability architecture using multiple AZs. What factors do you consider?
How do AWS Outposts, Local Zones, and Wavelength differ in terms of on-premise and low-latency requirements?
What are the trade-offs of using Edge Computing at Edge Locations vs Local Zones?
Describe a scenario where using a Local Zone significantly improves application performance.
How do you secure data replication across multiple Regions?
What tools or services would you use to monitor performance across Regions and AZs?
What is Regional Isolation and why is it important in AWS architecture design?
How would you architect a globally distributed app using Route 53, CloudFront, and multiple Regions?
What is the latency difference typically observed between Edge Locations and Local Zones?
How can you enforce compliance policies across different AWS Regions?

IAM (Identity and Access Management)



IAM Terminologies



IAM Dashboard Info

IAM resources

Resources in this AWS Account

User groups	Users	Roles	Policies	Identity providers
0	0	19	0	1

What's new

Updates for features in IAM

View all

- AWS IAM announces support for encrypted SAML assertions. *3 months ago*
- AWS CodeBuild announces support for project ARN and build ARN IAM condition keys. *3 months ago*
- IAM Roles Anywhere credential helper now supports TPM 2.0. *4 months ago*
- Announcing AWS STS support for ECDSA-based signatures of OIDC tokens. *5 months ago*

more

arn:partition:service:region:account:resource

Interview Questions on AWS Regions and Availability Zone

Basic Level (IAM)

1.What is IAM in AWS, and why is it important?
2.What are IAM Users, Groups, Roles, and Policies?
3.How do IAM policies work? What are the main elements in a policy document?
4.What is the difference between an IAM Role and an IAM User?
5.What is the use of IAM Groups?
6.What is a managed policy vs an inline policy?
7.What is the purpose of the Deny statement in IAM?
8.What does the Resource element define in an IAM policy?
9.How can you enforce MFA (Multi-Factor Authentication) in IAM?
10.What is a service-linked role in IAM?

Intermediate Level (IAM)

11.Explain the policy evaluation logic in IAM (Deny vs Allow precedence).
12.How would you restrict access to an S3 bucket to only one user?
13.What is an IAM trust policy, and where is it used?
14.How can you use IAM roles for cross-account access?
15.What is a permissions boundary in IAM, and how does it differ from a policy?
16.How do you rotate IAM user credentials securely?
17.What is the difference between AWS Organizations SCPs and IAM Policies?
18.How do you audit IAM policy changes?
19.What tools can be used to simulate or test IAM permissions?
20.What are best practices for IAM role permissions in EC2 instances?

Basic Level (IAM Identity Center)

21.What is IAM Identity Center (formerly AWS SSO)?
22.What is the difference between IAM Identity Center and traditional IAM?
23.What are permission sets in IAM Identity Center?
24.How does IAM Identity Center simplify user access management in multi-account environments?
25.What identity sources can IAM Identity Center integrate with?
26.How is user access managed across AWS accounts with IAM Identity Center?
27.What are the benefits of using IAM Identity Center with AWS Organizations?

Advanced Level (IAM Identity Center & Integration)

28.Explain how IAM Identity Center works with Azure AD or Okta as an external identity provider.
29.How does IAM Identity Center improve security compared to managing IAM users directly?
30.How do permission sets map to IAM roles in target AWS accounts?
31.How would you troubleshoot an access issue for a user using IAM Identity Center?
32.What is SCIM, and how is it used in IAM Identity Center?
33.Describe a real-world use case for replacing IAM users with IAM Identity Center.
34.What are the limitations of IAM Identity Center compared to IAM?
35.Can IAM Identity Center enforce fine-grained permissions like IAM policies?