IAM

Amazon Resource Names(ARNs)
What is IAM?
IAM Terminology?
Identities - (Users, Groups, Roles)
Idenity Policy and Resource Policy

Amazon Resource Names(ARNs)

- Amazon Resource Names (ARNs) uniquely identify AWS resources.
- ARN are required when we need to specify a resource in IAM policies, Amazon Relational Database Service (Amazon RDS) tags, and API calls.

ARN Format:

```
arn:partition:service:region:account:resource-id
```

- *partition* - aws (AWS Regions) , aws-cn - AWS China Regions , aws-us-gov - AWS GovCloud (US) Regions
- *service* - service identifies the AWS product.
- *region* - is the Region where resource resides in (e.g **us-east-1**)
- *account* - AWS Account id (e.g **123412344321**)
- *resource-id* - Resource Identifier like **user/alex** for IAM User or **instance/i-1212343456qwerty0** for an EC2 instance.

Paths in ARN

- Resource ARNs can include a path.
- E.g For S3, the resource identifier is an object name that can include slashes (/) to form a path.
- ARN are required when we need to specify a resource in IAM policies, Amazon Relational Database Service (Amazon RDS) tags, and API calls.

ARN Format:

`arn:`*`partition`*`:`*`service`*`:`*`region`*`:`*`account`*`:`*`resource-id`*

- *partition* - aws (AWS Regions) , aws-cn - AWS China Regions , aws-us-gov - AWS GovCloud (US) Regions
- *service* - service identifies the AWS product.
- *region* - is the Region where resource resides in (e.g **us-east-1**)
- *account* - AWS Account id (e.g **123412344321**)
- *resource-id* - This part of the ARN can be the name or ID of the resource or a **resource path**.
  Resource Identifier like **user/alex** for IAM User or **instance/i-1212343456qwerty0** for an EC2 instance.

IAM Terminology?

IAM Users

- IAM User is used to authenticate people accessing your AWS account.

IAM Group:

- An IAM group is a collection of IAM users.

IAM role:

- An IAM Role is used to authenticate AWS resources, for example an EC2

IAM Policy:

- An IAM policy is used to define the permissions for a user, group, or role.

IAM Terminology?

Principals:

- A person or application that uses the AWS account root user, an IAM user, or an IAM role to sign in and make requests to AWS.
- the **PRINCIPAL(s)** specifies WHO permissions are given to

Resources:

- The user, group, role, policy, and identity provider objects that are stored in IAM. As with other AWS services, you can add, edit, and remove resources from IAM.
- the **RESOURCE(s)** specifies WHICH properties are being accessed

Actions(s):

- the **ACTION(s)** specifies WHAT is being performed.

What is IAM?

- Enables you to securely control access to AWS services and resources for your users.
- Manage IAM users/groups and their access
- Manage IAM roles and their permissions

With AWS IAM you get to control who can do what in your AWS environment and from where

IAM Introduction

- Manage IAM users/groups and their access.
- Users must be created with proper permissions.
- Policies are written in JSON (JavaScript Object Notation)
- Root account should never be used (and shared)

IAM Introduction

- Root in AWS is the same as Root in Windows/Linux
- Password Policies
- Manage Access Keys
- Fine grained control of users, groups, roles, and permissions to resources
- IAM has a global view.
- MFA (Multi Factor Authentication) can be setup
- IAM has predefined “managed policies”

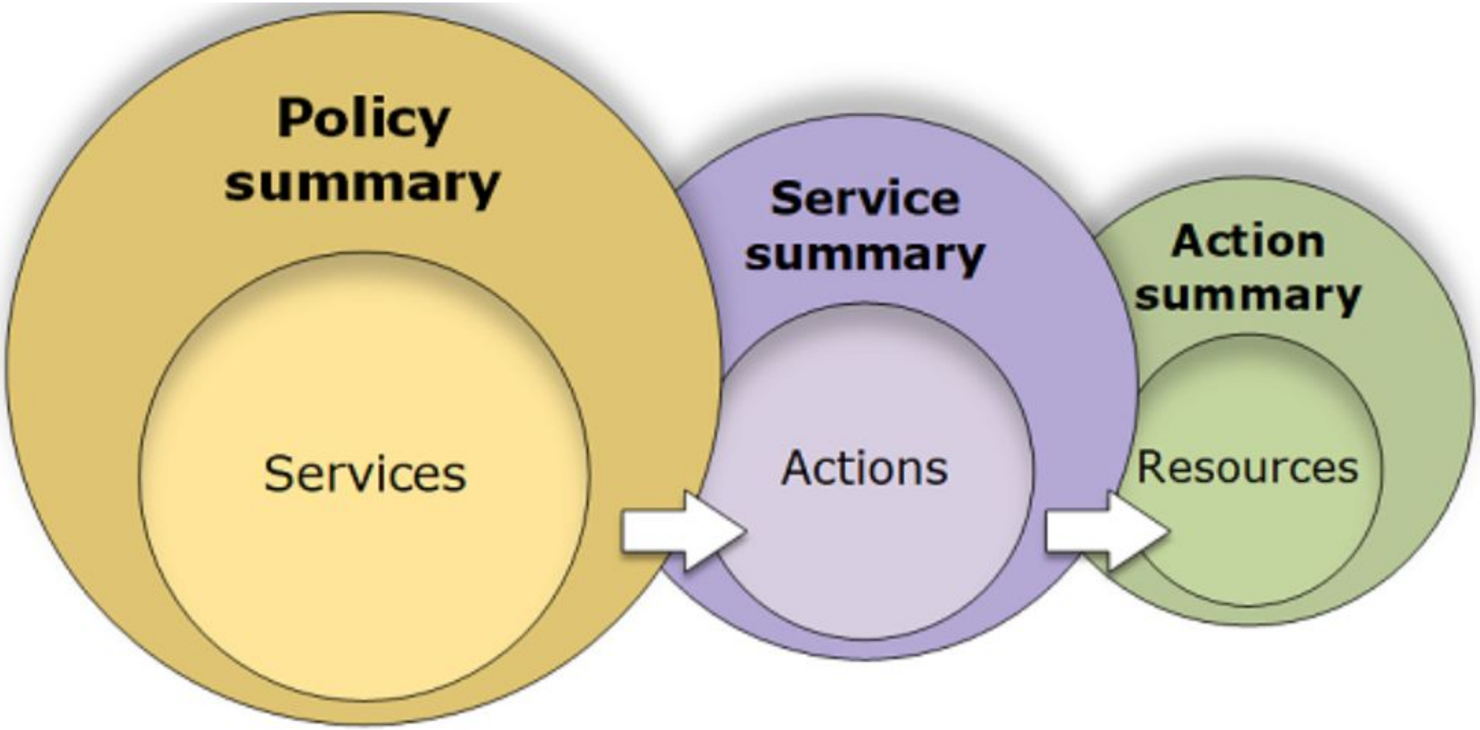
Identities - (Users, Groups, Roles)

- The AWS Account Root User
- IAM Users
- IAM Groups
- IAM Roles

What is an AWS IAM Policy?

- A set of rules that, under the correct conditions, define what actions the policy principal or holder can take to specified AWS resources.
- Most permission policies are JSON policy documents in AWS that, when attached to an identity or resource, define their permissions.
- Policies can be granted in a number of ways:
  - Attaching a managed policy. AWS provides a list of pre-defined policies such as **AmazonS3ReadOnlyAccess**.

IAM Policy


Policy summary
Services
Service summary
Actions
Action summary
Resources

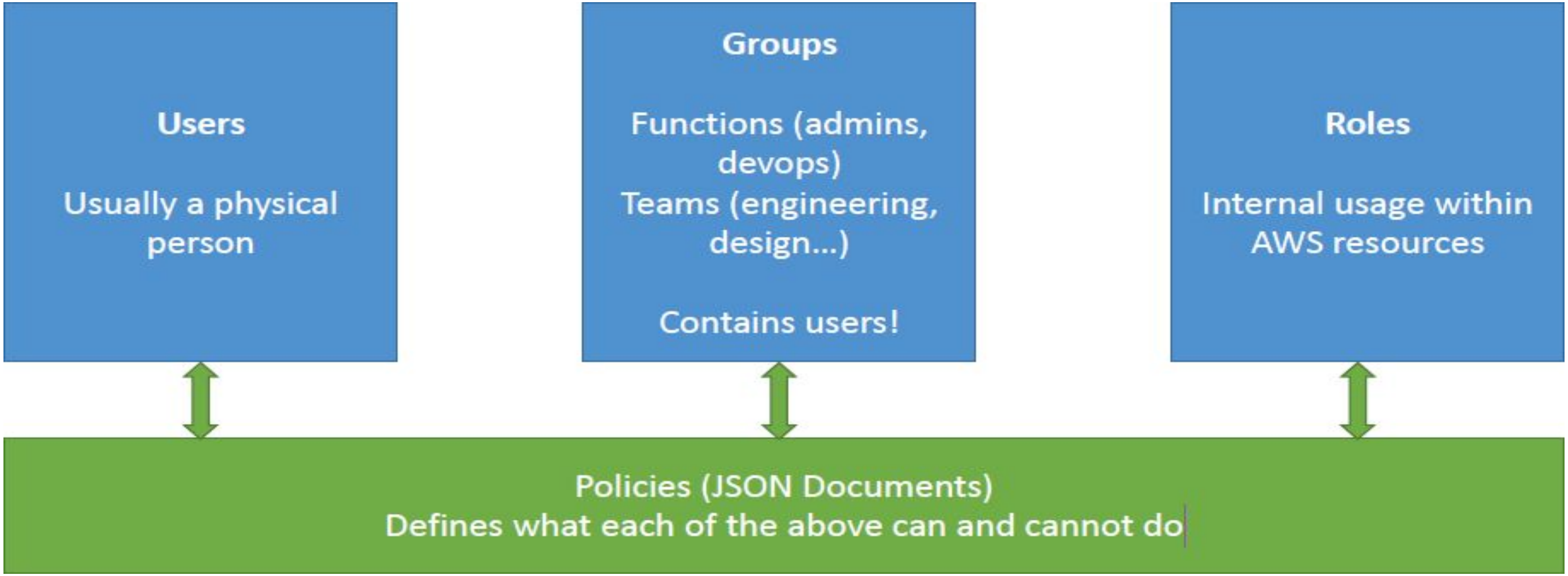
Permissions and Policies

- The access management portion of AWS Identity and Access Management (IAM) helps you define what a user or other entity is allowed to do in an account. This process is often referred to as authorization.
- Permissions are categorized as permissions policies and permissions boundaries.

Policies and Users

- IAM users are identities in the service.
- When you create an IAM user, they can't access anything in your account until you give them permission.
- You give permissions to a user by creating an identity-based policy, which is a policy that is attached to the user.

IAM Introduction



Users

Usually a physical person

Groups

Functions (admins, devops)
Teams (engineering, design...)

Contains users!

Roles

Internal usage within AWS resources

Policies (JSON Documents)
Defines what each of the above can and cannot do

Policy Evaluation Logic

Deny evaluation -

- By default, all requests are denied. This is called an **implicit deny.**
- In all policies, the enforcement code looks for a Deny statement that applies to the request. This is called an **explicit deny**.
- If the code finds even one explicit deny that applies, the code returns a final decision of Deny. If there is no explicit deny, the code continues.
- **Deny** rule for a similar action will take precedence.

IAM Policy Variables

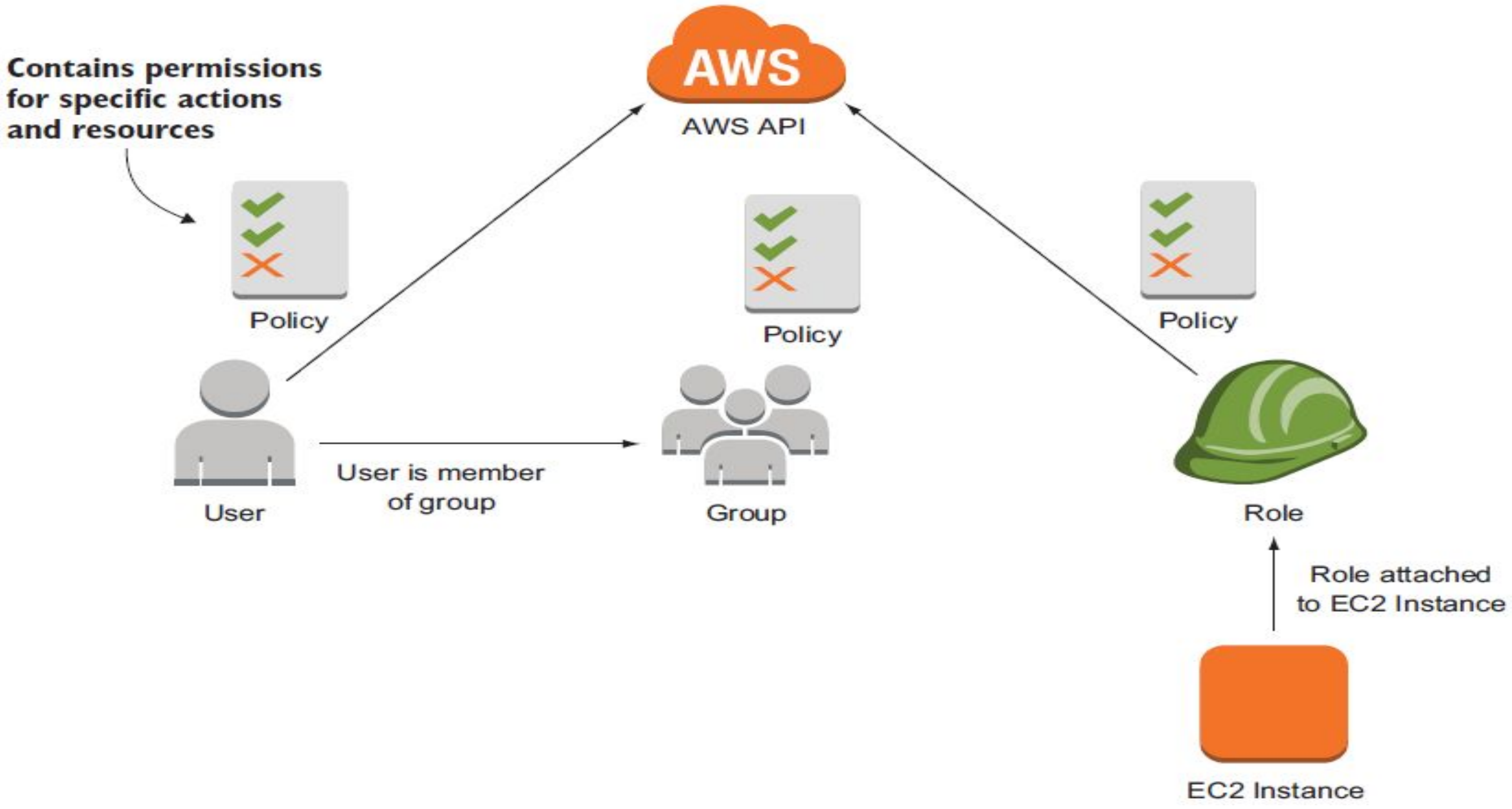
Policies contain keys whose values you can use as policy variables.

- `aws:username` This is a string containing the friendly name of the current user—see the chart that follows.
- `aws:SourceIp` This is the requester's IP address, for use with IP address conditions.

IAM Features

- Shared access to your AWS account
- Granular permissions
- Multi-factor authentication (MFA)
- Free to use

IAM Features


Contains permissions for specific actions and resources
AWS
AWS API
Policy
Policy
Policy
User
User is member of group
Group
Role
Role attached to EC2 Instance
EC2 Instance

IAM Federation

- Big enterprises usually integrate their own repository of users with IAM
- This way, one can login into AWS using their company credentials
- Identity Federation uses the SAML standard (Active Directory)

IAM Brain Dump

- Never use **ROOT IAM** Credentials
- One IAM User per **PHYSICAL PERSON**
- One **IAM Role** per Application
- IAM credentials should **NEVER BE SHARED**
- Never use the **ROOT** account except for initial setup.
- Never write IAM credentials in code.

ANY Questions?