IAM Policies

1. Introduction
   1. Defines who is allowed to do what with which AWS resources
   2. Controls access to AWS services and resources.
   3. A set of rules in a JSON Format
   4. Helps keep your AWS environment secure
   5. Can be attached to three types of entities
      1. Users
      2. Groups
      3. Roles
   6. In AWS IAM there are essentially three types of policies
      1. AWS Managed Policies
      2. Customer Managed Policies
      3. Inline Policies
2. AWS Managed Policies
   1. Created and managed by AWS
   2. Designed for common use cases across a wide range of AWS services
   3. Ready to attach to multiple IAM users, groups, or roles within an AWS account.
   4. Automatically updated by AWS to grant access to new services or actions without requiring manual policy updates.
   5. Provide a way to quickly assign necessary permissions based on job function or application need.
   6. Divided into two types
      1. AWS managed policies (general use)
      2. Service-linked policies (specific to AWS services).
   7. Example
      1. Objective — Enable the EC2MasterMind user to fully manage EC2 instances.
      2. Policy Used — AmazonEC2FullAccess
      3. Entity — IAM user named EC2MasterMind
      4. Permissions Included
         1. Launch EC2 instances with chosen
            1. AMIs
            2. Instance types
            3. Configurations.
         2. Perform operations like
            1. Start
            2. Stop
            3. Reboot
            4. Terminate
         3. Handle instance storage options like
            1. Volumes
            2. Snapshots
         4. Configure network settings and security groups for instances
         5. Create and manage SSH key pairs for secure login to instances
         6. Utilize AWS CloudWatch for monitoring instances and setting up alerts
   8. Benefits & Limitations
      1. Benefits
         1. Simplicity
         2. Predefined by AWS
         3. Automatic Updates
         4. Reusable
         5. Best Practices
      2. Limitations
         1. Resource-Specific Access Control
         2. No Customization
         3. Broad Permissions
         4. Dependence on AWS
         5. Understanding Complexity
3. Customer Managed Policies
   1. Introduction
      1. Created and Managed by Users
      2. Tailored for Specific Needs
      3. Attachable to Multiple IAM Entities
      4. Manual Updates Required
      5. Enables Fine-grained Permission Control
      6. Versioning and Rollback
   2. Example
      1. Objective — To configure IAM permissions such that the user "ec2mastermind" is authorized to manage (e.g., start, stop, terminate) only two out of three specific EC2 instances within the AWS environment
      2. Policy Used — Custom Policy
      3. Entity — IAM user named EC2MasterMind
      4. Permissions Included

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
      ],
      "Resource": [
        "arn:aws:ec2:region:account-id:instance/instance-id",
        "arn:aws:ec2:region:account-id:instance/instance-id,
        "arn:aws:ec2:region:account-id:instance/instance-ida"
]
    }
  ]
}
```

      5. Test Policy With Policy Sim
   3. Policy Evaluation
      1. Default Deny — If no pol
      2. Explicit Allow — Allows an action unless explicitly denied
      3. Explicit Deny
         1. Overrides any allow
         2. If a policy explicitly denies an action, that deny takes precedence over any allow.
         3. Example

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ec2:*",
      "Resource": "arn:aws:ec2:region:account-id:instance/instance-id"
    }
  ]
}
```

   4. Maximum Policy Size — The total size of each customer managed policy d … s (including whitespace) in the policy's JSON representation.
4. Inline Policies
   1. Introduction
      1. Direct Association
         1. Inline policies are policies that are directly embedded within a single IAM user, group, or role.
         2. Difference — This contrasts with managed policies, which exist as separate entities in AWS and can be attached to multiple IAM users, groups, or roles.
      2. One-to-One Relationship
         1. Each inline policy is tied to one and only one IAM entity.
         2. If the entity is deleted, the inline policy is also deleted.
         3. Difference — This is in contrast to managed policies, which are independent entities that can be attached to and detached from multiple IAM users, groups, or roles without being deleted when a single association is removed.
      3. Non-Reusable — Since inline policies are embedded in a single IAM entity, they cannot be shared or reused by other IAM entities
      4. No ARN — Inline policies do not have an Amazon Resource Name (ARN) because they are not separate entities within AWS.
   2. Use Case
      1. Specific Job Needs
      2. Keep Things Secure
      3. Short-term Projects
      4. Follow Rules
      5. Easy to Manage for Certain Cases