Identity & Access Management: The First Step in AWS Security

Reef D'Souza
Security Consultant
AWS Professional Services

aws | Pop-up Loft

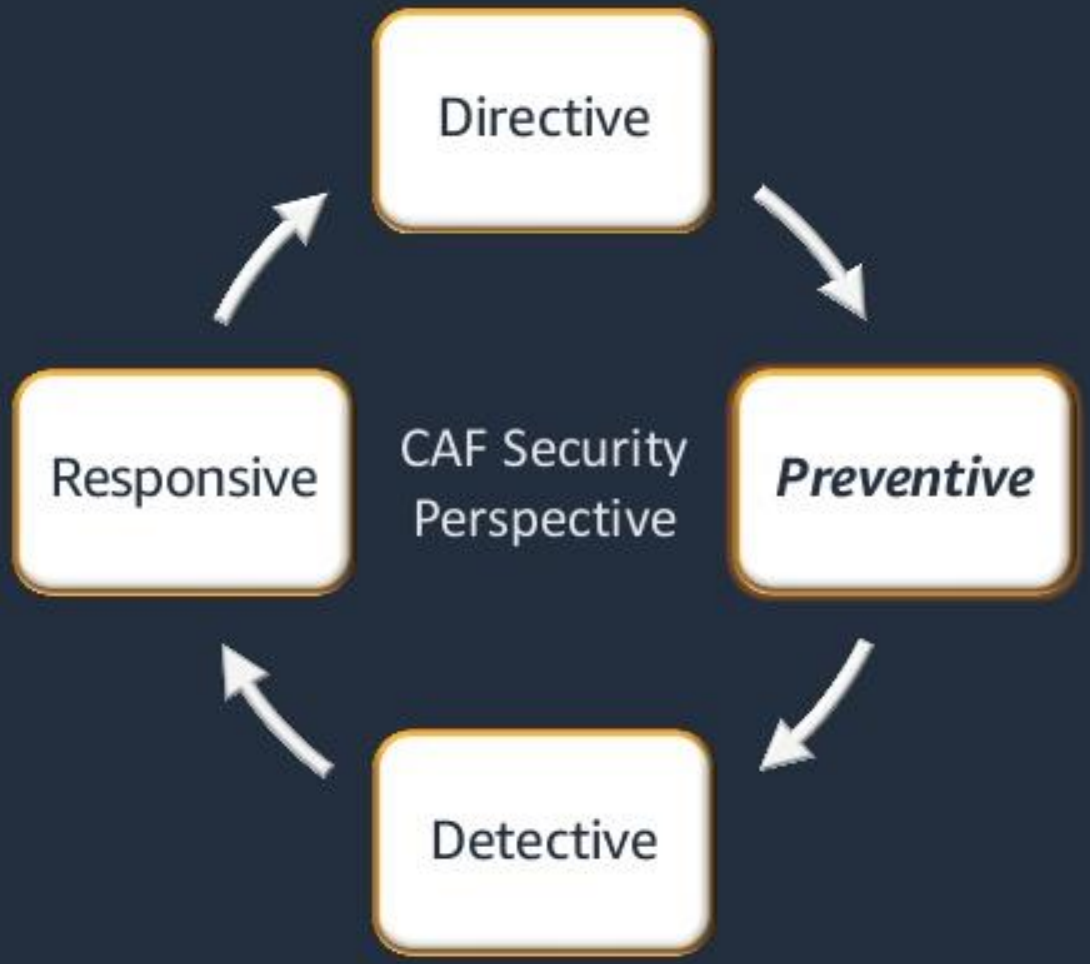
Agenda

- Overview of AWS Identity & Access Management
- IAM Authentication & Authorization
- AWS Identity Federation
- AWS Organizations & Service Control Policies
- Recap of Best Practices

AWS Identity & Access Management

IAM

What is IAM?



- IAM is a preventive security control
- What is a *security control*?
  - Any hardware, software, policy or procedure meant to govern access to IT resources
- A *preventive security control* is one intended to thwart unwanted or unauthorized activity

What is AWS IAM?

- Create and manage AWS users and groups and use permissions to allow and deny access to AWS resources
- Integrates with Microsoft Active Directory using SAML identity federation and AWS Directory Service
- Roles can be created and assumed to control what operations can be performed by an entity or AWS service (e.g. EC2 instance)

Why use AWS IAM?

- You can specify permissions to control which operations a user or role can perform on AWS resources
- IAM service provides access to the AWS Management Console, AWS API, and AWS Command-Line Interface (CLI)

Note: IAM does *not* provide authentication for your OS or application

What are IAM Users?

- IAM users can be an individual, system, or application requiring access to AWS services
- A user account consists of a unique name and security credentials such as a password, access key, and/or multi-factor authentication (MFA)
- IAM users only need passwords when they access the AWS Management Console

What are IAM Groups?

- IAM Groups are a way to assign permissions to logical and functional units of your organization
- IAM Groups are a tool to help with operational efficiency
  - Bulk permissions management (scalable)
  - Easy to change permissions as individuals change teams (portable)

What are IAM Roles?

- An IAM role is similar to a user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS.
- You can authorize roles to be assumed by humans, Amazon EC2 instances, custom code, or other AWS services

IAM Authentication & Authorization

What are IAM Access Keys?

- IAM access keys are used to make programmatic calls to AWS using the AWS Command Line Interface (CLI), the AWS SDKs, or direct HTTPS calls to the APIs for individual AWS services
- Consists of an Access Key ID and a Secret Access Key
- Example:
  - Access Key ID: AKIA3R7HGUSSI4BOW
  - Secret Access Key: MxQ4QSzT0NsnEO5VNCYjJo

What is the signing process?

- AWS Signature Version 4 is the process to add authentication information to AWS requests.
  - The AWS SDKs or CLI tools will construct, sign, and send requests for you, with the access keys you provide.
  - If you are constructing AWS API requests yourself, you will have to include code to sign the requests.

- http://docs.aws.amazon.com/general/latest/gr/signature-version-4.html

What are IAM passwords?

- IAM passwords are credentials used by IAM users to authenticate to the AWS Management Console.
- A password policy can be configured for the AWS account
- Alphanumeric and common special characters are allowed:

  ! @ # $ % ^ & * ( ) _ + − = [ ] { } | ‘

When should I use Access Keys vs. passwords?

- Depends on how your users will access AWS
  - Console → Password
  - API, CLI, SDK → Access keys
- In either case, make sure to rotate credentials regularly
  - Use Credential Report to audit credential rotation
  - Configure password policy
  - Configure policy to allow access key rotation

aws

What is an IAM principal?

- IAM principal refers to a user, account, service, role, or other entity
- In terms of evaluating authorization, a principal is defined as the entity that is allowed or denied access to a resource.

What are IAM policies?

- IAM policies are JSON-based statements that define access control and permissions
- IAM polices can be “inline” or “managed”

aws

Example IAM policy

```
{   "Version": "2012-10-17",
    "Statement": {
     "Effect": "Allow",
     "Action": "s3:ListBucket",
     "Resource": "arn:aws:s3:::example_bucket”
   }
}
```


aws

What are IAM Policies?

- *Inline policies* are policies that you create and manage, and that are embedded directly into a single user, group, or role.
- *Managed policies* are standalone policies that you can manage separately from the IAM users, groups, or roles to which they are attached
  - AWS managed policies
  - Customer managed polices

Choosing Inline vs Managed Policies

- Use *inline policies* when you need to:
  - Enforce a strict one-to-one relationship between policy and principal
  - Avoid the wrong policy being attached to a principal
  - Ensure the policy is deleted when deleting the principal

Choosing Inline vs Managed Policies

- Use *managed policies* when you need:
  - Reusability
  - Central change management
  - Versioning and rollback
  - Delegation of permissions management
  - Automatic updates for AWS managed policies
  - Larger policy size

What are Resource-based policies?

- To specify resource-based permissions, you attach an inline policy to a supported resource, such as an Amazon SNS topic, an Amazon S3 bucket, an AWS KMS key, or an Amazon Glacier vault.
- Resource-based policies must specify who is allowed to access the resource, known as the Principal.

Policy structure

Identity-based policy

```
{
 "Statement":[{
   "Effect":"effect",
   "Action":"action",
   "Resource":"arn",
   "Condition":{
     "condition":{
       "key":"value" }
     }
    }
   ]
 }
```

Resource-based policy

```
{
 "Statement":[{
   "Effect":"effect",
   "Principal":"principal",
   "Action":"action",
   "Resource":"arn",
   "Condition":{
     "condition":{
       "key":"value" }
     }
    }
   ]
 }
```


aws

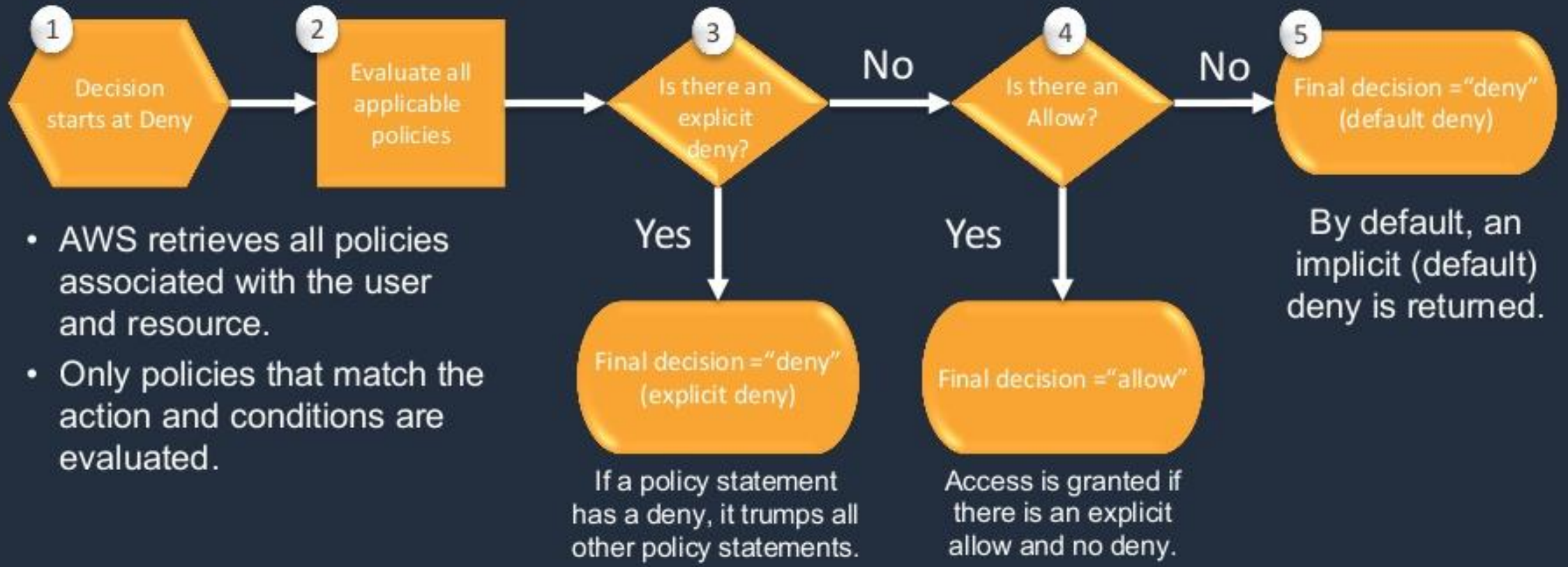
What is the key difference?

- With Identity-based Policies, the Principal is *implicit*.
- With Resource-based Policies, the Principal is *explicit*.

IAM Evaluation Logic

- By default, the decision is default deny
- An allow overrides any default denies
- An explicit deny overrides any allows
- The order in which the policies are evaluated is not important

IAM Policy Evaluation Logic



- AWS retrieves all policies associated with the user and resource.
- Only policies that match the action and conditions are evaluated.

By default, an implicit (default) deny is returned.

If a policy statement has a deny, it trumps all other policy statements.

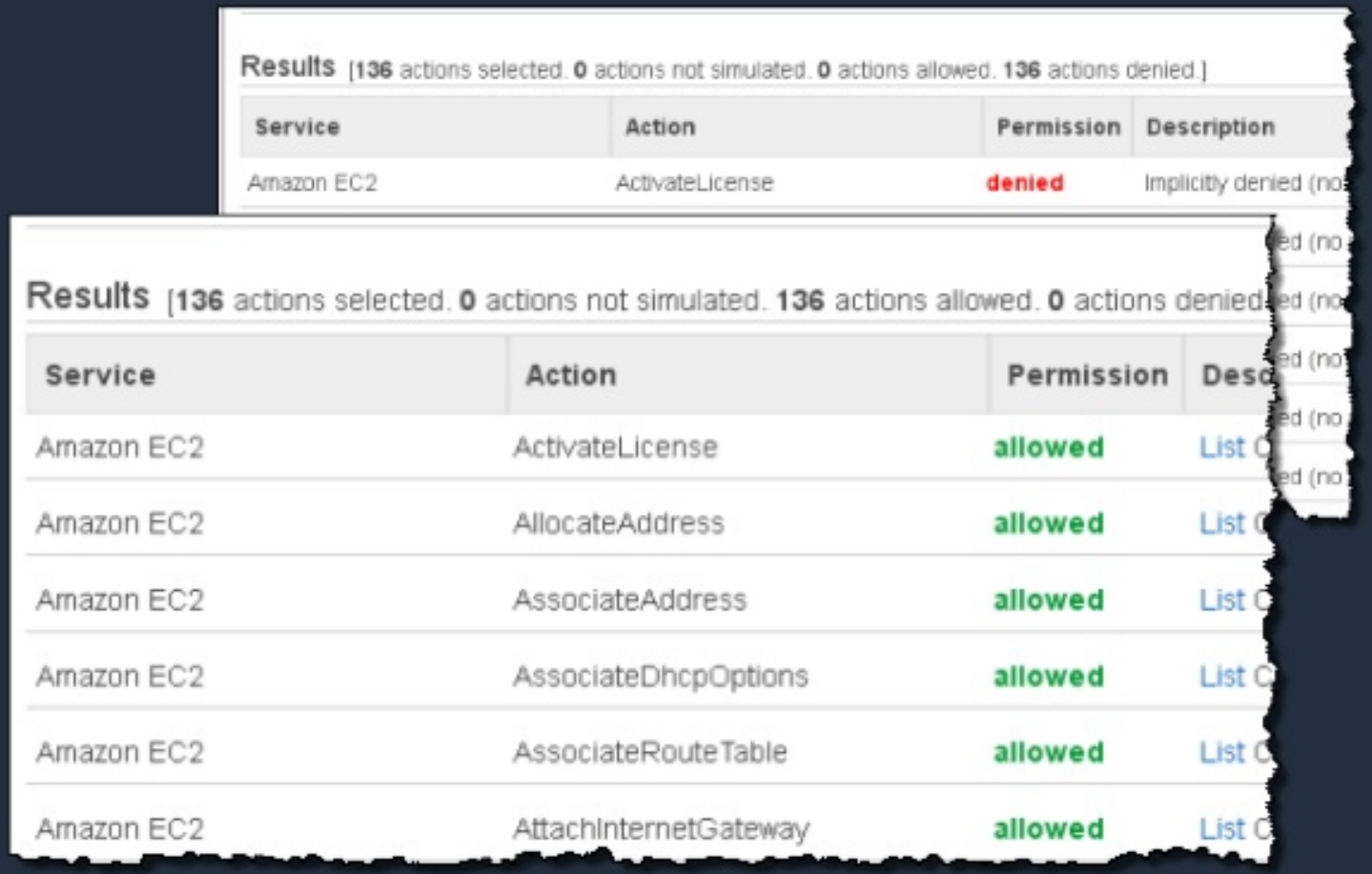
Access is granted if there is an explicit allow and no deny.

Prevent Privilege Escalation

- Any principal with full IAM permissions is effectively a super user, even if no other permissions have been granted.
- Why? The principal can modify its own permissions at any time.
- To prevent this, deny access to at least these IAM actions:
  - `iam:AttachUserPolicy`
  - `iam:AttachGroupPolicy`
  - `iam:AttachRolePolicy`
  - `iam:PutUserPolicy`
  - `iam:PutGroupPolicy`
  - `iam:PutRolePolicy`

What is the IAM Policy Simulator?

The IAM policy simulator allows you to test policies against resources in your account



What is the IAM Policy Simulator?

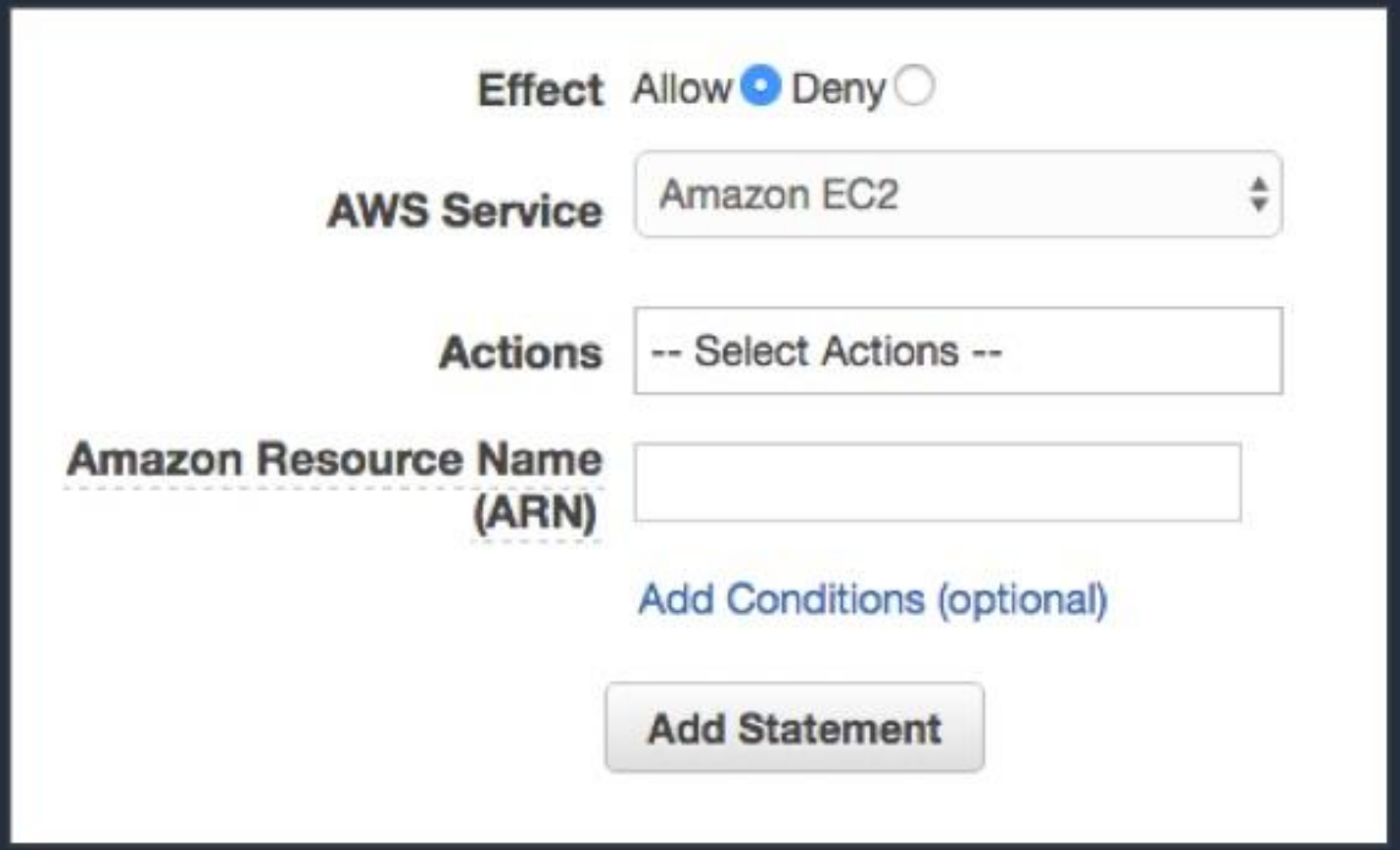
- Test policies that are attached to IAM users, groups, or roles in your AWS account.
- Test policies that are attached to AWS resources, such as Amazon S3 buckets, Amazon SQS queues, Amazon SNS topics, or Amazon Glacier vaults
- Test new policies that are not yet attached to a user, group, or role by typing or copying them into the simulator.

What is the IAM Policy Simulator?

- Test the policies with selected services, actions, and resources.
- Simulate real-world scenarios by providing context keys, such as an IP address or date, that are included in Condition elements in the policies being tested.
- Identify which specific statement in a policy results in allowing or denying access to a particular resource or action.

What is the IAM Policy Generator?

The AWS IAM Console has a simple GUI that helps you build your IAM policies.



AWS Identity Federation

AWS Identity Federation

	Users	Security	Compliance
			
Before:	Unique Credentials	Long Lived Keys	One-off
After:	Single sign-on	Short-term tokens	Naturally aligned

IAM Users vs. Federated Users

- IAM supports users managed in AWS's identity management system
- Users managed outside of AWS in your corporate directory are referred to as "federated users"
  - Examples of corporate directories, include Microsoft Active Directory, PingFederate, Okta Univeral Directory, etc.

What are IAM Identity Providers?

- IAM identity providers allow you to use identities managed outside of AWS instead of creating IAM users in your AWS account
- You don’t have to create custom sign-in code or manage your own identities
- This is helpful if your organization already has its own identity system, such as a corporate user directory like Microsoft Active Directory

What is AWS Security Token Service (STS)?

- The AWS Security Token Service (STS) is a web service that enables you to request temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) users or for users that you authenticate (federated users)

What is AWS Security Token Service (STS)?

- The AWS Security Token Service (STS) is a web service that enables you to request temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) users or for users that you authenticate (federated users)

Ideas on Identity & Federation

- Do IAM users require a password or access key to call an AWS API?
  - Access key, as passwords are only used to log into the AWS Management Console
- Should code running on an EC2 instance use an access key or IAM role to access DynamoDB?
  - IAM role, as access keys can be challenging to manage
- Which IAM permission is more powerful, “iam:CreatePolicy” or “iam:AttachRolePolicy”?
  - iam:AttachRolePolicy, since you can escalate permissions by attaching any policy to a role

aws

AWS Organizations & Service Control Policies

What is AWS Organizations?

- Service that enables customers to centrally manage policies across multiple AWS accounts
- Create accounts and invite existing accounts to join your organization and organize them into groups called organizational units (OUs)
- Attach policy-based controls called Service Control Policies (SCPs) to centrally control AWS service use across multiple AWS accounts
- Simplify billing for multiple accounts by enabling a single payment method through consolidated billing

aws

AWS Organizations Policies

- A policy is a “document” with one or more statements that define the controls that you want to apply to a group of AWS accounts.
- Currently, AWS Organizations provides a Service Control Policy (SCP).
  - An SCP defines the AWS service actions, such as Amazon EC2 RunInstances, that are available for use in different accounts within an organization.

What is a Service Control Policy (SCP)?

- Service Control Policies (SCPs) allow you to control which AWS service actions are accessible to principals (account root, IAM users, and IAM roles)
- Authorization on a principal in an account that has an SCP attached is the intersection of what is allowed explicitly in the SCP and what is allowed explicitly in the IAM permissions attached to the principal.

aws

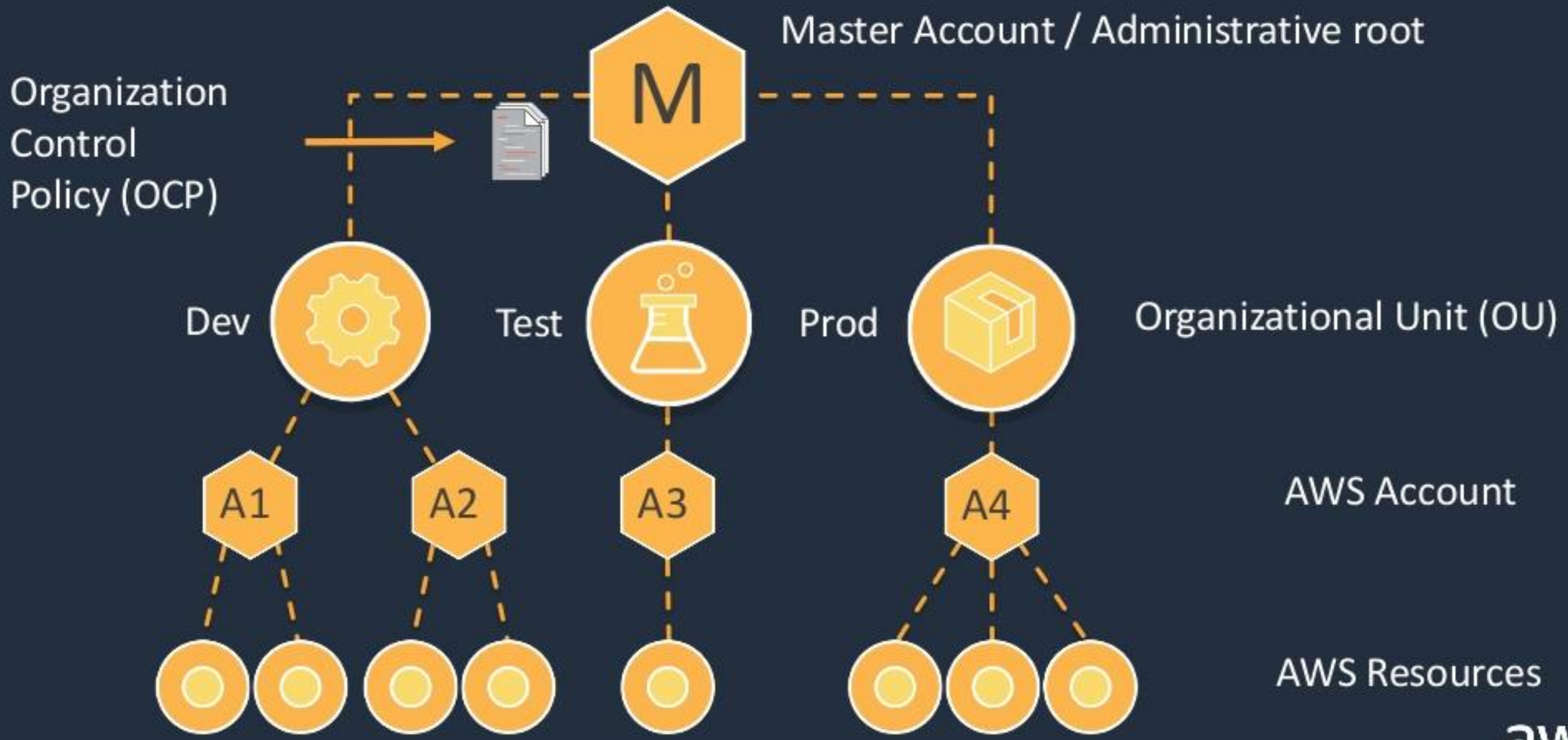
Service Control Policy (SCP) Example

- If a **SCP** applied to an account states that the only ALLOWs Amazon EC2 actions
- And if a **IAM policy** on a principal in the same AWS account ALLOWs both EC2 actions and Amazon S3 actions

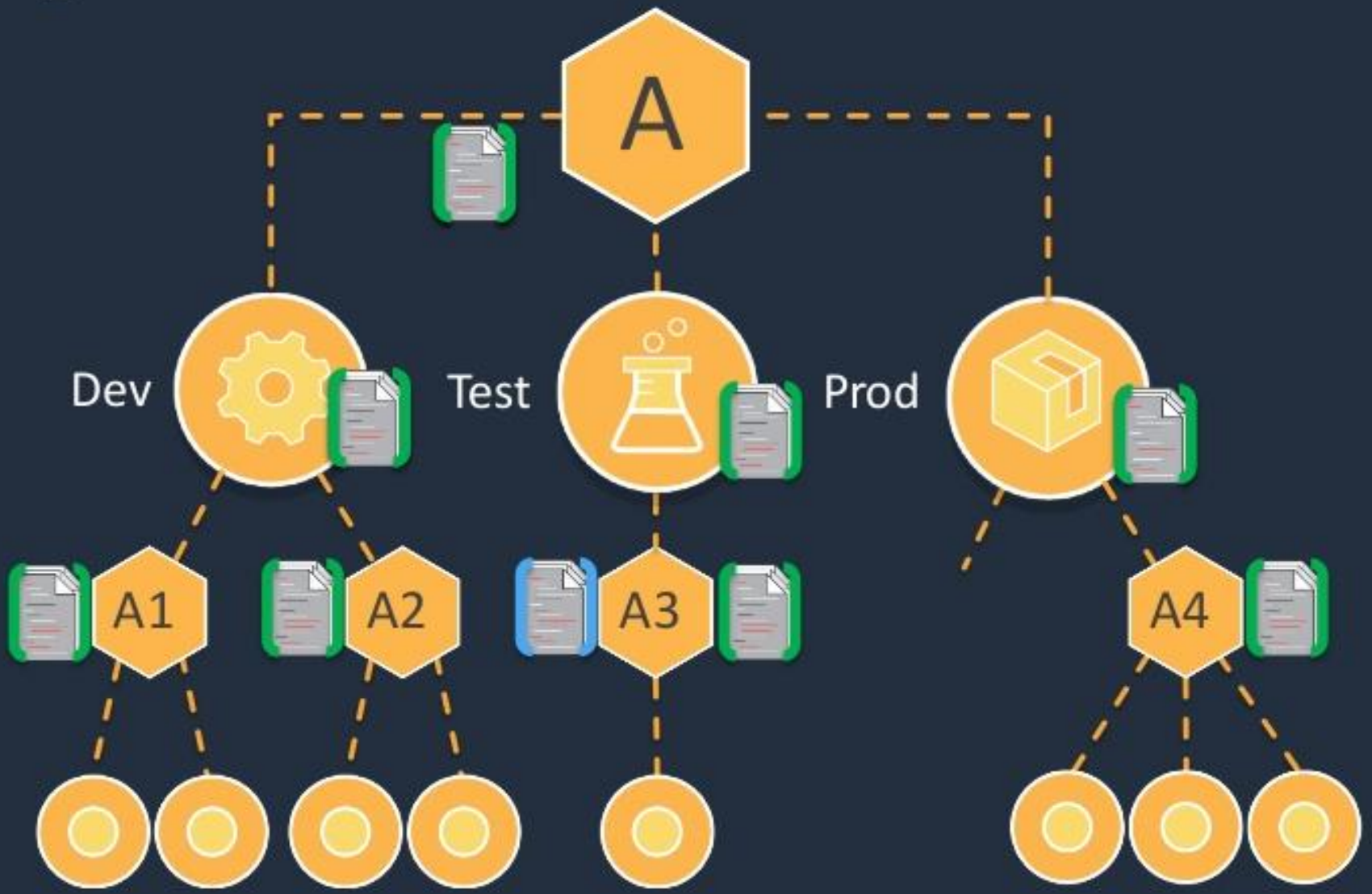
- Then the principal is able to access ***only*** the EC2 actions

aws

AWS Organizations



AWS Organizations Service Control Policies



IAM Best Practices

- Lock away your AWS account (root) access keys
- Create individual IAM users
- Use groups to assign permissions to IAM users
- Grant least privilege
- Configure a strong password policy for your users
- Enable MFA for privileged users

IAM Best Practices

- Use roles for applications that run on Amazon EC2 instances
- Delegate by using roles instead of by sharing credentials
- Rotate credentials regularly
- Remove unnecessary credentials
- Use policy conditions for extra security
- Monitor activity in your AWS account

- http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

Everything and Anything Startups Need to Get Started on AWS

aws.amazon.com/activate