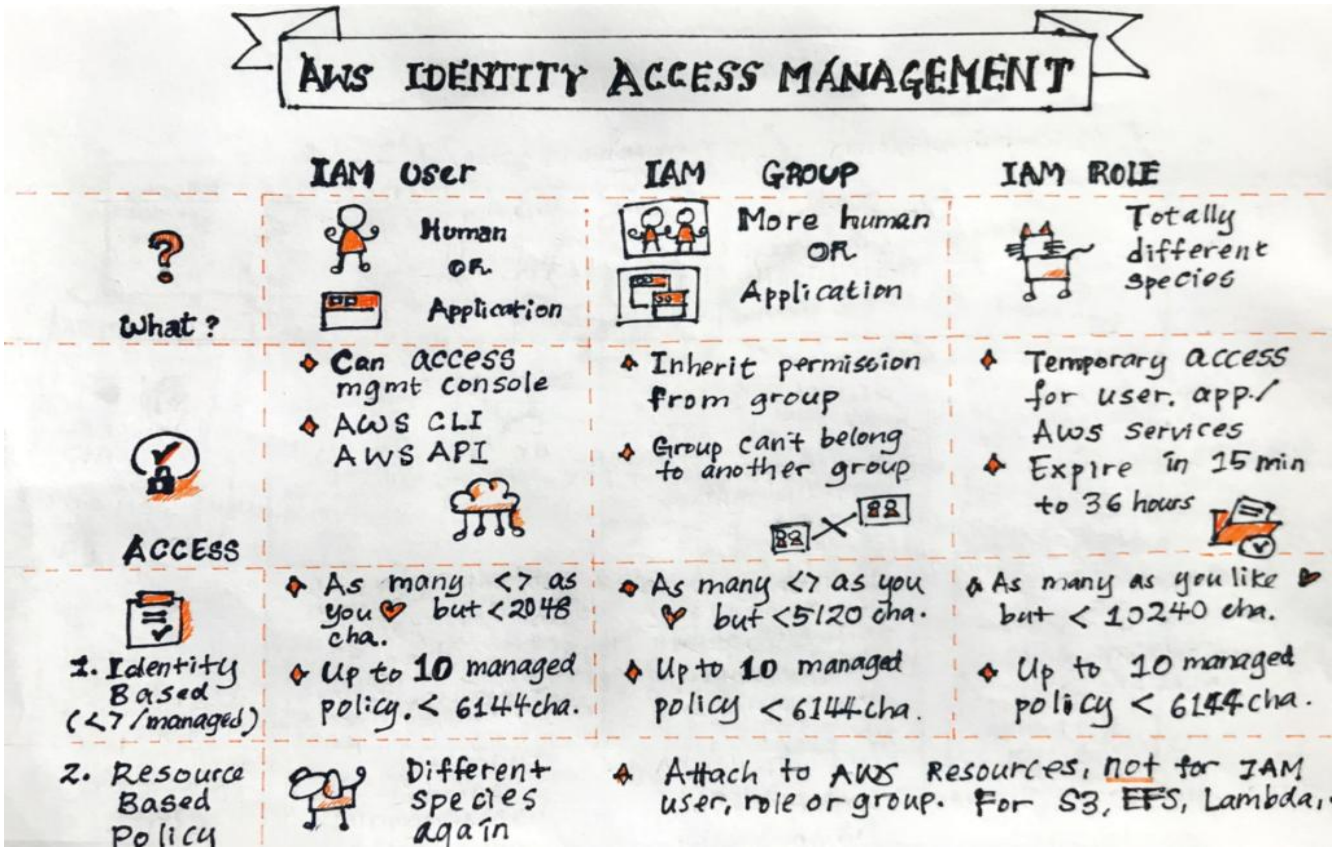
IAM-Identity and Access Management

11. IAM-Identity and Access Management



- AWS identity and access management service (IAM) is a web service that helps you securely control access to AWS resources for your users.
- AWS Identity and Access Management (IAM) provides fine-grained access control across all of AWS.
- With IAM, you can specify who can access which services and resources, and under which conditions.
- With IAM policies, you manage permissions to your workforce and systems to ensure least-privilege permissions.
- IAM is a feature of your AWS account and is offered at no additional charge.
- IAM Provides –
- Shared access to your AWS account.
- Granular Permissions.
- Secure access to AWS resources.
- Integrated with many of the AWS resources.
- Free to use.
- Global service
- 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.
- A user in AWS consists of a name and credentials.
- We can also create a user as a service account.
- An IAM user with administrator permissions is not the same thing as the AWS account root user.

TechData-Infinity-Devops with MultiCloud

- As a best practice, do not use the AWS account root user for any task where it's not required. Instead, create a new IAM user for each person that requires administrator access.
- Then make those users administrators by placing the users into an "Administrators" user group to which you attach the Administrator-Access managed policy.
- IAM refers to a framework or policies and technologies for ensuring the proper people in an organization have the appropriate access to technology Resources.
  OR
- AWS Identity and access Management is web service that helps you securely control access to AWS Resources you use IAM to control who is Authenticated (signed-in) and authorized (has permission) to use Resources.
- When you create an AWS account, you begin with a single sign-in identity that has completely access to all AWS services and resources in the account.
- This identity is called the AWS account "Root-User" and is accessed by signing-in with the email address and password that you used to create the account.
- AWS Strongly recommends that you do not use the root user for your everyday task, even the administrative ones.
- Use other IAM user accounts to manage the administrative task of your account and securely lock away the root user credentials and use them to perform only a few account and service Management Task.
- IAM user limit is 5000 per AWS Account you can add up to 10 users at one time.
- You are also limited to 300 Groups per AWS account.
- You are limited to 1000 IAM Roles under AWS account.
- Default limits of Managed Policies attached to an IAM role and IAM user is 10.
- IAM users can be member of 10 groups (max).
- We can assign two access keys (max) to an IAM user.
- Shared access to your AWS account.

You can grant other people permission to administer and use Resources in your AWS account without having to share your access Credentials (password or access key).

GRANULAR PERMISSIONS-

- You can Grant different permission to different people for different resources.
- For Instance, you can allow Some users complete access to EC2, S3, dynamo DB, Red shift while for others, you can allow read only access to just some S3 brackets or permission to administer just some EC2 instances or to access your billing Information but nothing else.
- Secure access to AWS resources for applications that run on amazon EC2.
- You can use IAM features to Securely give application that run on EC2 instances the credentials that they need in order to access other AWS Resources Example include S3 Buckets and RDS or DynamoD3 database.

Multifactor Authentication (MFA)-

- You can add two factor authentication to your account and to individual users for extra security you can use physical hardware or virtual MFA (for example. Google Authenticator).

Identity federation-

- You can allow users who already have password elsewhere for example. In your Corporate Network or with an Internet Identity provider to get temporary access to your AWS account.

TechData-Infinity-Devops with MultiCloud

Identity Information for assurance-

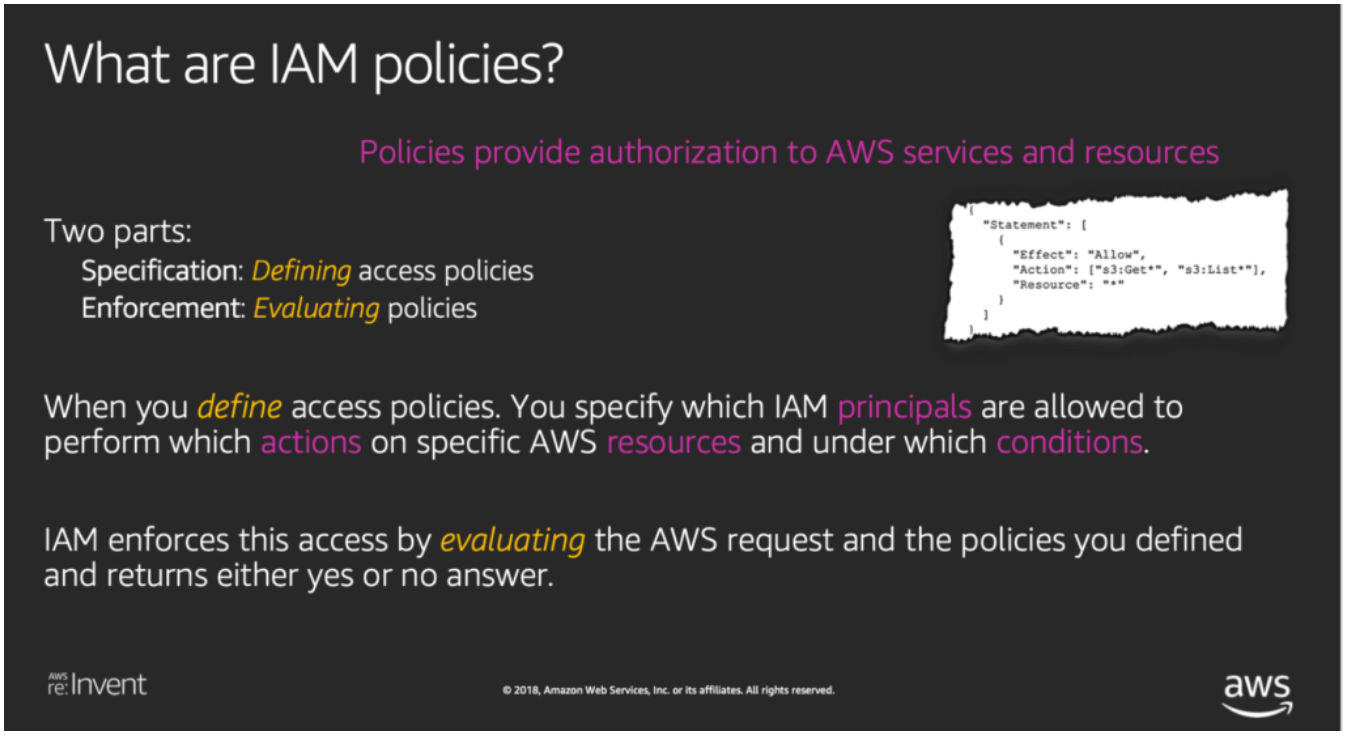
- If you use AWS cloud trail, you receive log Records that include Information about those who made request for resources in your account that information is based on IAM Identities.

PCI-DSS Compliance-

- IAM support the processing storage and transmission of credit card by merchant or service provider, and has been validated as being compliant with payment card Industries (PCI) Data Security Standard (DSS).

Eventually Consistent-

- If a request to change some data is successful the change is committed & Safely stored However the change must be Replicated across IAM which can take some time.
- IAM achieves high availability by Replicating data across multiple servers within AWS data center around the world.
- Free to Use.
- AWS IAM is feature of your AWS account offered at no additional charges.
- You will be charged only for use of others AWS products by your IAM users.



TechData-Infinity-Devops with MultiCloud

IAM Terms-

Principal-

- A principal is a person or application that can make a request for an action or operation on an AWS Resources.
- Your administrative IAM user your first principal.
- You can allow users and services to assume a Role.
- IAM users, Roles, Federated users and application are all AWS Principals.
- You can support federated user or Programmatic access to allow an application to access yours AWS Account.

Request-

When a principle tiers to use the AWS management Console, the AWS API, or the AWS CLI, that principal sends a Request to AWS the Request includes the following information.

- Actions
- Resources
- Principal
- Environment data
- Resources Data
- Actions- That principal wants to perform.
- Resources- Upon which the actions are performed.
- Principal- Information including the environment from which the Request was made.

Authentication-

- A principal sending a request must be authenticated (signed in to AWS) to send a Request to AWS.
- Some AWS services, Like AWS S3, allow request from anonymous users, they are exception to the role.
- To authenticate from the console as a root user, you must sign in with your user name and password.
- To authenticate from the API to CLI, you must provide your access key and secret key.
- You might also require to provide additional Security Information like MFA (example Google Authentication).

Authorization-

- To authorize Request, IAM users value from the Request content to check for matching policies and determine whether to allow or deny the request.
- IAM policies are stored in IAM as JSON documents and specify the permission that are allowed or denied.
- User (Identity) based policies specify permission allowed Denied for principal.

Note – By default, only the AWS root user has access to all the resources in that account.

- Resource base policies- specify permission allowed/ denied for Resources popular for granting Gross account permissions.
- IAM checks each policy that matches the context of your Request.
- If a single policy includes a denied action, IAM Denies the entire Request and stop evaluating this is called Explicit deny.
- The evaluation logic follows these Rules.
- By default all request are denied.
- An explicit allow overrides this default.

- An explicit deny overrides any allows.

- You can create a new IAM policy in the AWS management console using one of the following ways-
- JSON- you can create your own JSON syntax.
- Visual- you can construct a new policy from scratch in the visual editor if you use the visual editors, you can do not have to understand JSON syntax.
- Import- you can Import a managed policy within your account and then edit the policy to customize it to your specific Requirement.

Actions

- Actions are defined by a service, and are the things that you can do to a resources Such as Viewing, Creating, Editing and Deleting that Resources.
- IAM supports approx. 40 actions for a user Resource including create user, Deleting user etc.
- Any action or resources that are not explicitly allowed are denied by default.
- After your request has been authenticated and authorized, AWS approves the actions in your Request.

Resource

- A resource is an entity that exist with is a service.
- Examples are EC2 instances, S3 bucket, IAM user, Dynamo DB table.
- Each WAS service defines a set of actions that can be performed on each resource.
- After AWS approves the action in your request those actions can be performed on the Related Resource within your account.
- If you create a request to perform an unrelated action on a Resource, that Request is denied.
- When you provide permission using an identity based policy in IAM than you provide permissions to access resources only within the same account.

Identity Federation

- If your account user already have a way to be authenticated such as authentication through your corporate network.
- You can federate those user Identities in to AWS.
- A user who has already logged to the corporate using their corporate Identity.
- The corporate can replace their existing identity in your AWS account.
- This user can work in the AWS management console.
- Similarly, an application that the user is working with can make programmatic request using permissions that you define.

Federation is particularly use in these cases

- If your corporate directory is compatible with security Assertion Markup Language (2.0)
- You can configure your corporate directory to provide sign-on (SSO) access to the AWS Management for your users.
- If your corporate directory is not compatible with SAML (2.0)
- You can create Identity Broker application to provide single sign-on (SSO) access to the AWS management console for your user.

- If your corporate directory is Microsoft active Directory, you can use AWS directory services to establish trust between your corporate directory and yours AWS account.
- If you are creating a mobile app or web based app that can let uses identity themselves through an internet identity provider like login with Amazon, Facebook, Google or any open id connect (OIDC) compatible identity provider the app can use web federation to access AWS.
- AWS recommends to use AWS cognito for identity federation.

IAM Users and SSO

- IAM users in your account have access only to the AWS Resources that you specify in the policy that is attached to the users or to an IAM Group that the user belongs to.
- To work in the console user must have permission to perform the action that the console performs such as listing and creating AWS resources.

- IAM Identities-
- Users
- Groups.
- Role.
- IAM identities is that you create under your AWS account to provide authentication for people application and process in your AWS account.
- Identities represent the user and can be authenticated and the authorized to perform action in AWS.
- Each of this can be associated with one or more policies to determine what actions are user role or member of a group can do with which resources and under what conditions.
- IAM group is a collection of IAM users.
- IAM roll is very similar to IAM user.

IAM Users

- IAM user is an entity that you create in AWS it represent the person or service who uses the IAM user to interact with AWS.

- You can create 10 users at a time.
- IAM user can represent and actual person or an application that requires AWS access to perform actions on AWS resources.
- A primary use for IAM users is to give people the ability to sign in to the AWS management console for interactive task and to make programmatic request to AWS services using the API or CLI.

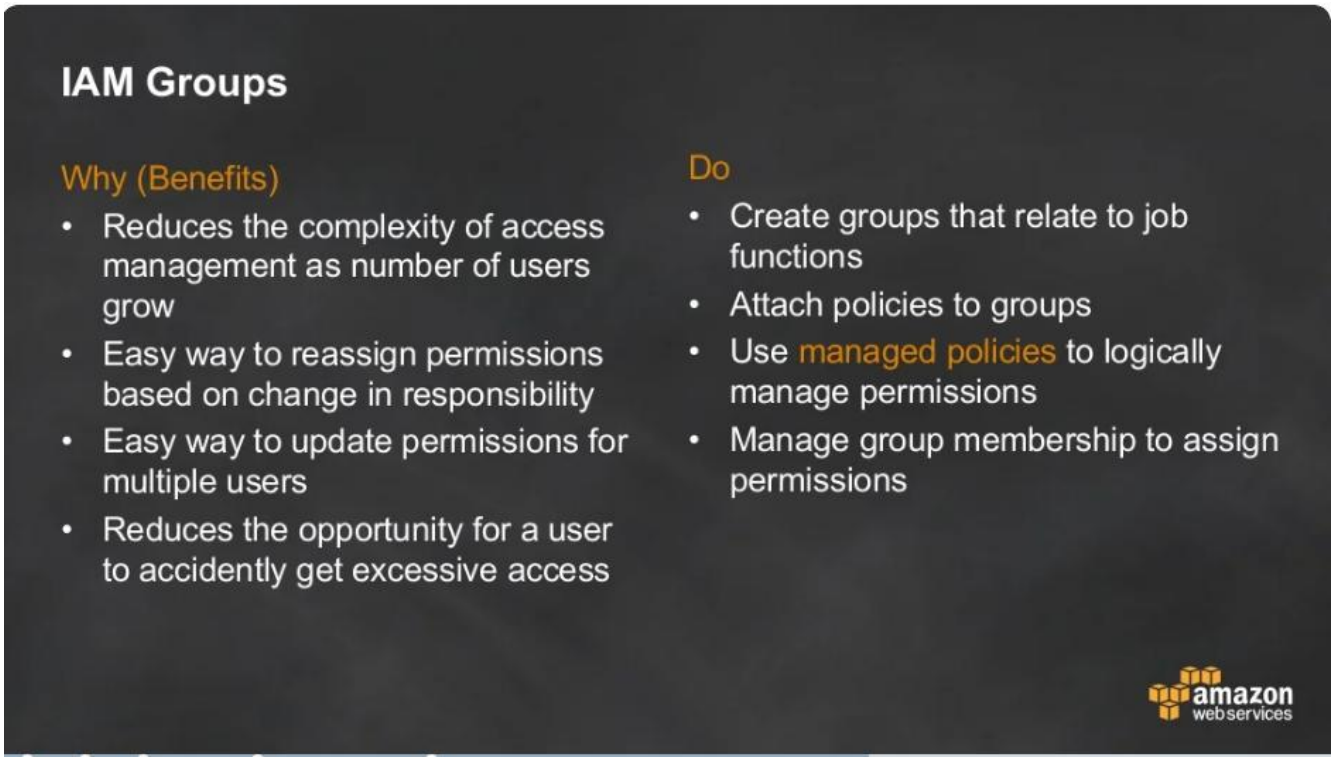
For any Users you can assign them-

- A username and password to access the AWS console.
- An access key id and secret key that they can use for programmatic access.
- The newly created IAM users have no password and no excess key you need to create the user password.
- Each IAM user is associated with one and only one AWS account.
- Users can define within your account so users do not have do payment bill would be pay by the parent account.

IAM Groups

- An IAM group is a collection of IAM users.
- It is a way to assign permission or policies to multiple users at a time.
- Use groups to specify permissions for a collection of users which can make those permissions easier to manage for those users for example you could have a group called HR and that group the types of permissions that HR department typically needs.
- Any user that groups automatically has the permission that are assigned to the group.
- If a new user joins your organization and should have HR privileges you can assign the appropriate permissions by adding the user to that group.
- If a person changes a job in your organization instead of editing that uses permissions you can remove him or her from the old groups and add him or her to the appropriate new groups.

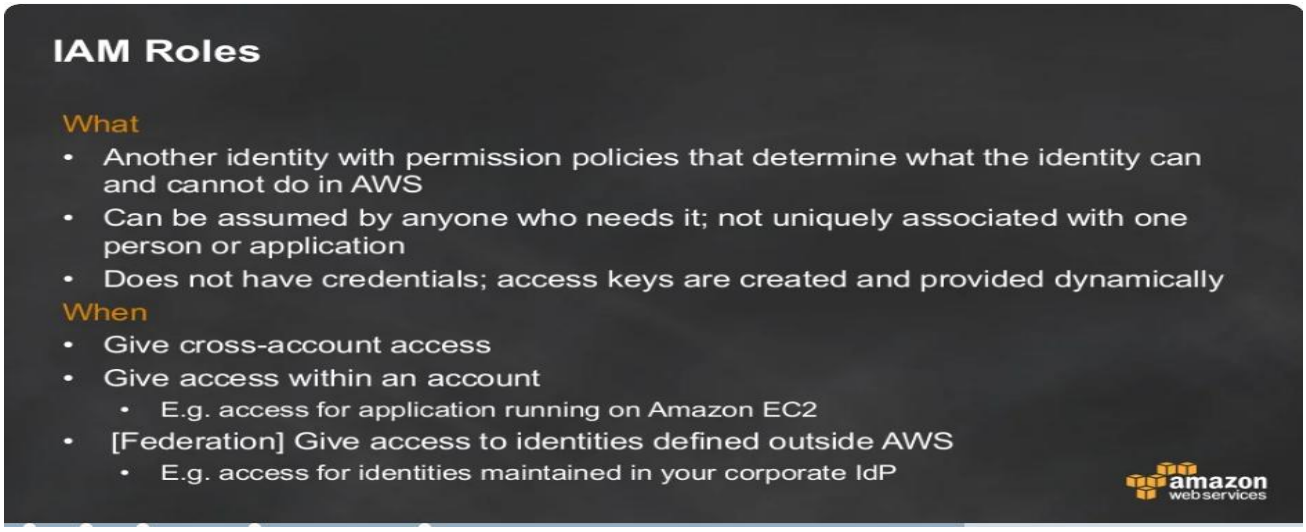
TechData-Infinity-Devops with MultiCloud



IAM Groups Limitations

- A group is not truly and identity and IIM because it cannot be identified as a principal is a permission policy.
- Groups can't be nested.
- You have a limit of 300 groups is an AWS account.
- A user can be member of up to 10 IAM groups.

IAM Roles



TechData-Infinity-Devops with MultiCloud

- IAM role is very similar to user in that it is an identity with permission policies that determine What the Identity can and cannot do in AWS.
- An IAM role does not have any credentials password or access key associated with it.
- Instead of being unequally associated with one person a role is intended to be assumable by anyone who needs it.
- An IAM user can assume a roll to temporary take an different permissions for a specific task.
- An IAM role can be assigned to a federated user who signs in by using an external identity provider instead IAM.

IAM Temporary Credentials-

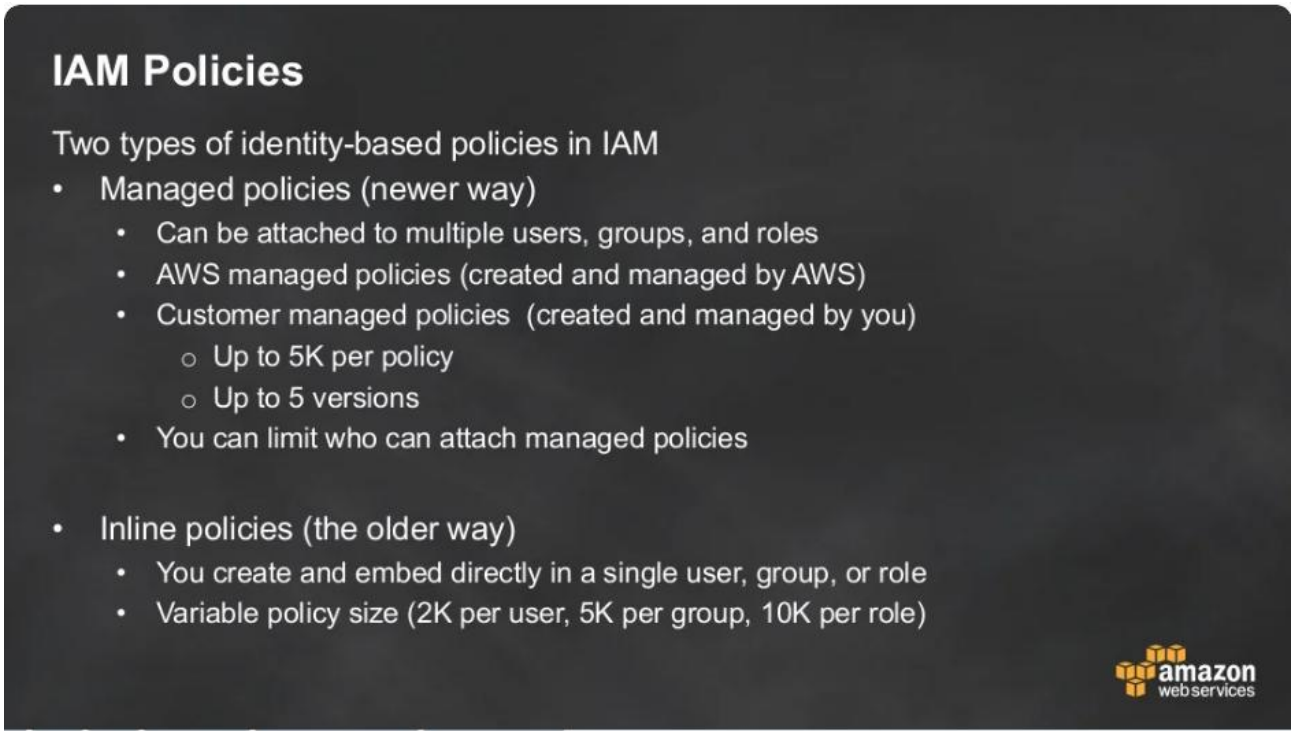
- Temporary credentials are primarily used with I am Rose but there are also other uses.
- You can request temporary credentials that have a more restricted set of permissions than you are standard IAM users.
- This present you from accidentally performing task that are not permitted by the more restricted credentials.
- A benefit of temporary credentials is that the expires automatically after a set of time.

Permissions and Policies

The access management portion of AWS identity and access management IAM helps you get define what are you this are there in the di is allowed to do in and around dolphin the referred to as authorization.

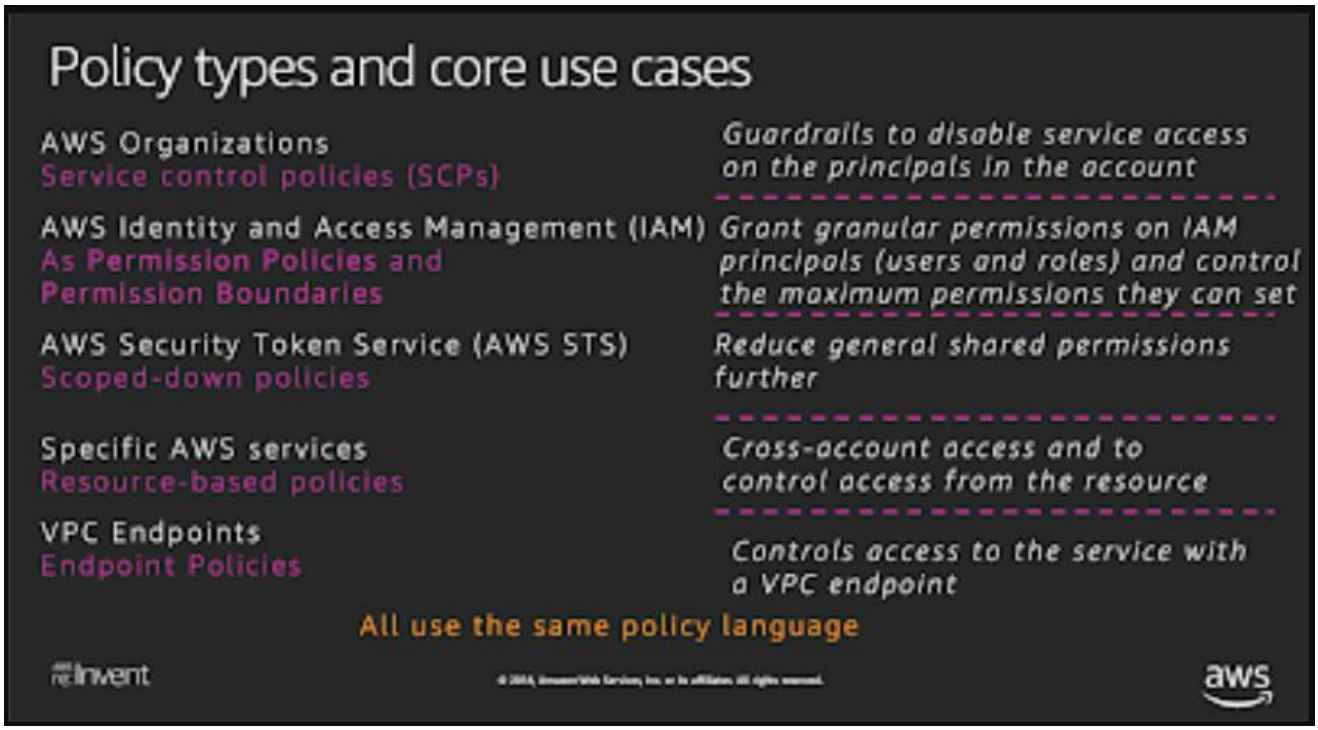
Permission our granted through policies that are created on then attach to users groups or roles.

Policies and Users.



TechData-Infinity-Devops with MultiCloud

- By default I am uses can't access anything in your account.
- You grant permissions to a user by creating a policy which is document that defines the effect actions resource and optional conditions.
- Any actions are resources that are not explicitly allowed by default.



IAM multiple policies

- User or groups can have multiple policies attached to them that grant different permission.
- In case of multiple policies attach to user or a group the users permission are calculated based on the combination of policies.

Federated users and Roles

- Federated user don't have permanent identities in your AWS account the way that IAM users do.
- To assign permission to federated uses you can create an integrity referred to as a role and define permission for the role.
- When a federated user sign in to AWS the user is associated with the role and is granted the permission that are defined in the role.

Resource Based Policies-

- In some cases like S3 bucket you can attach a policy to resource in addition to attaching into a group or user this is called a resource based policy.
- A resource best policy contents slightly different information then a user based policy in a resource based policy you specify what action are permitted and what resources is affected.
- You also explicitly list who is allowed access to the resource a principal.

- Resource base policies include a principal element that specifies who is granted the permission.

IAM User-

- When you first create an AWS account you create account or root user identity which you use to sign in to AWS account.
- The account root user credentials are the email address used to create the account and a password which can be used to sign in to the AWS management console as the root user.
- When you sign in as the root user you have complete unrestricted access to all resources in your AWS account including access to your billing information and the ability to change your password.
- The level of access is necessary when you initially set up the account.
- It is not possible to restrict the permission that are granted to the AWS account.
- An IAM user is an entity that you create in AWS it represents the person or service who access the IAM user to interact with AWS.
- An IAM user can represent and actual person or an application that requires AWS access to perform action on a w s resources.
- IAM user are global entities like an AWS account is today no reason is required to be specified when you define users permissions users can you use AWS services in any geographically region.

For any user you can assign them–

- A Username and password to access the AWS console.
- An access key (access key id and secret key) that they used for programmatic access (issue in request) to your AWS services using API and CLI.
- You assigned either or both best on the user activities and needs.
- You can view and download you are secret access key only when you create the access key.
- You cannot view recover a secret access key letter.
- If you lose you are secret access key you can create a new access key.
- Each I am user is associated with one AWS account.

AWS recommends

- AWS recommends that you don't use root user credentials everyday access.
- Also AWS recommends that you do not share your root user credentials with anyone because doing so gives them unrestricted access to your account.
- Create an IAM user for yourself and then assign yourself administrative permission for your account.
- You can then sign in as that user to add more users as needed.
- An IAM user with administrator permission is not the same thing as the AWS account root user.

By default a new IAM user

- A new IAM user has no permission to do anything.
- Has no password and no access key neither and access key ID nor a secret access key it means no credentials of any kind.
- You must create the type of potentials for and IAM uses best on what the user will be doing.

- You can grant user permissions by attaching I am policies to them directly or making them members of I am group where they inherit the group policies or permissions.
- You can have up to 5000 users per AWS account.

IAM Role

- An IAM Role Is a set of permission that grant excess to actions and resources in AWS.
- This permissions are attached to the role not to an I am user or group instead of being unequally associated with one person or role is intended to be assumable by anyone who needs it.
- A Role does not have standard long term credentials password or access keys associated with it.
- If a user assumes a role temporary security credentials are created dynamically and provided to the user.
- An IAM user in the same AWS account.
- An IAM used in the different AWS account.
- A web service offered by AWS such as Amazon elastic complete cloud.

There are two things to use role-

- Interactively in the IAM console.
- IAM uses in your account using I am console can switch to a roll to temporary use that permissions of the role in the console.
- The user give up there original permission and take on the permission assigned to the role.
- When they use exits the role their original permission are restored.

- Programmatically with the AWS CLI tools for windows PowerShell or API.
- An application or service offered by AWS (like Amazon EC2) can assume a role by requesting temporary security credentials for a role with which to make programmatic request to AWS.
- You use a role this way so that you didn't have to share or maintain long term security credentials for each entity that require access to a resource.

Difference between IAM role and resource based policy

- Unlike a user based policy a resource best policy specifies who can access that resource.
- Cross account access with a resource based policy has an advantage over a role with a resource that is excess through the resource best policy the user still work in the trusted account and does not have to give up his or her user permission in place of the role permission.
- In other words the user continues to have access to resource in the trusted account at the same time as he or she has excess to the resource in the trusting account.
- This is useful for task such as copying information to or from the shared resource in the other account.
- Note that not all services support resource based policy.

IAM role delegation

- Delegation is the granting of permission to someone to allow to resources that you control.

TechData-Infinity-Devops with MultiCloud

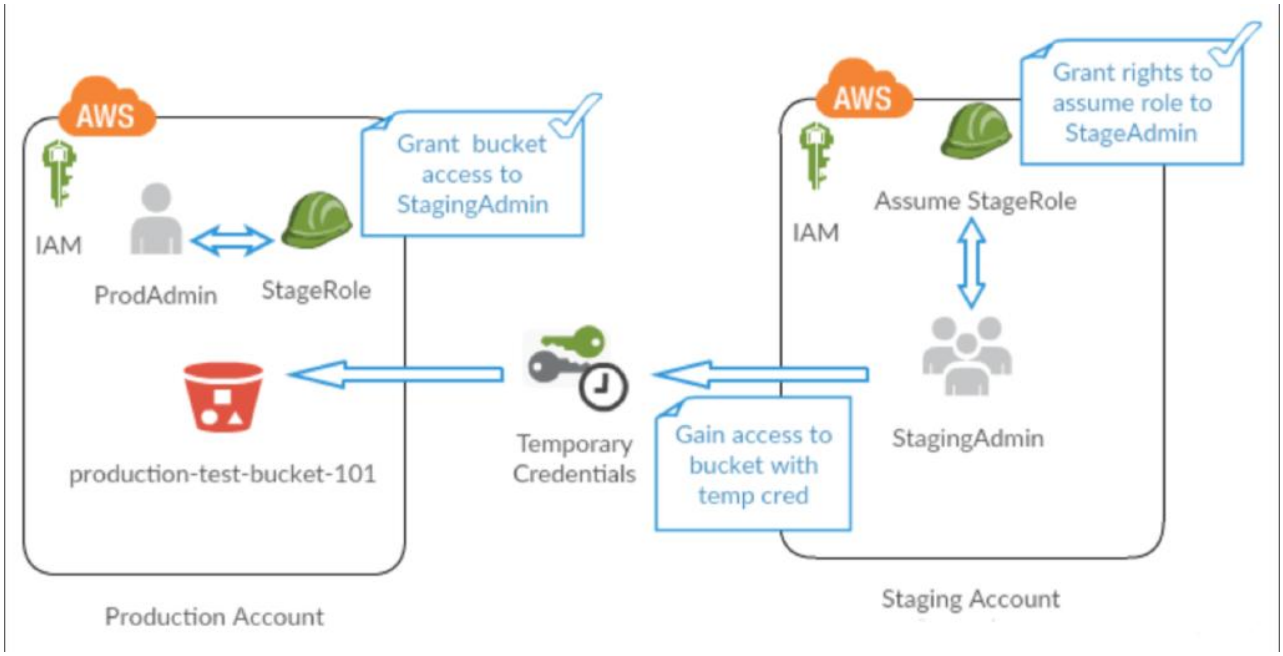
- Delegation involves setting up a trust between the account that owns the resource the trusting account and the account that contains the users that need to access the resource the trusted account.

- The trusted and trusting account can be any of the following.
- The same account.
- To accounts that are both under your organizations control.
- To account owned by different organization.
- To delegate permission to access resource you create an IAM role that has two policies attached.
  1. The trust policy
  2. The permission policy

- The trusted entity is included in the policy as the principle element in the documents.
- When you create a trust policy you cannot specify a wild card as a principal.

Cross account permissions

- You might need to allow users from another AWS account to access resources in your AWS account if so don't share security credentials such as access keys between accounts in state use IAM roles.
- You can define a role in the trusting account that specifies what permissions the IAM user in the other account are allowed.
- You can also designate which AWS account have the IAM users that are allowed to assume the role we do not define users here rather AWS account.



TechData-Infinity-Devops with MultiCloud

Role for Cross account

- Granting access to resource in on account to a trusted principle in different account.
- Roles are the primary way to grant cross account excess
- However with some of the web services offered by AWS you can attach a policy directly to a resource there are called resource best policy you can use them to grant principles in another AWS account access to the resource.
- The following services support resource best policy
- Amazon S3
- Amazon Simple Notification Service
- Amazon Simple Queue Service
- Amazon Glacier Vault