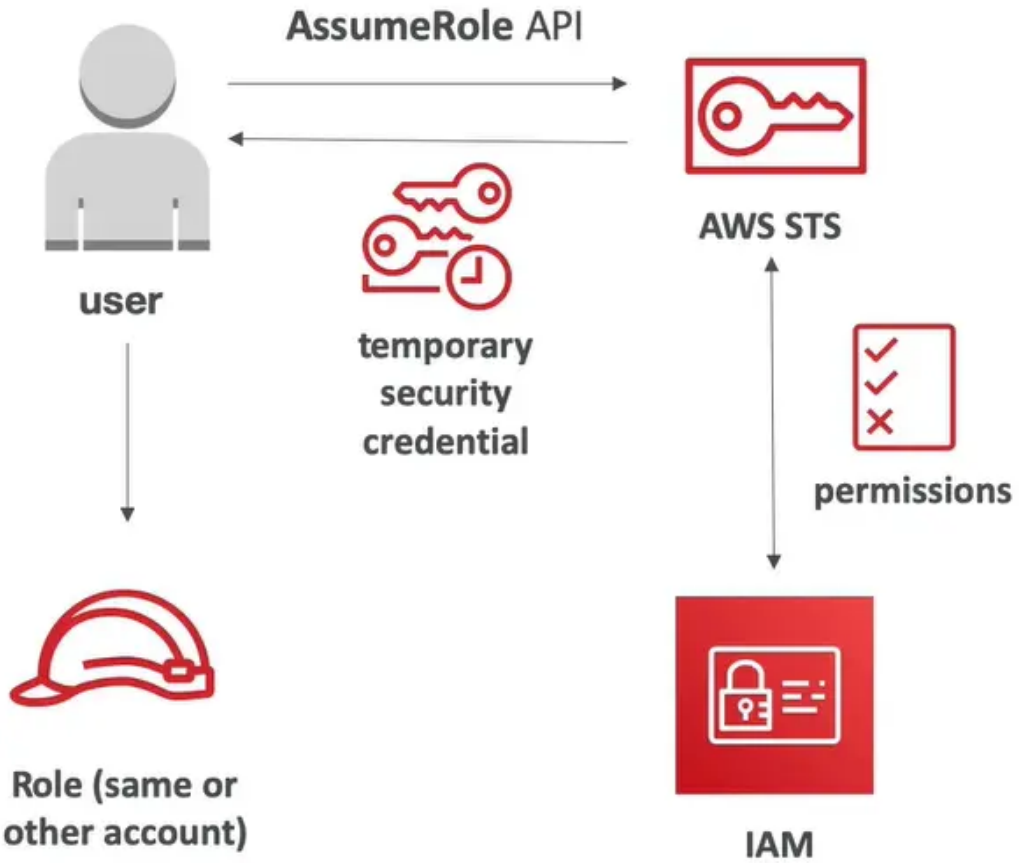
Advanced Identity

AWS STS – Security Token Service

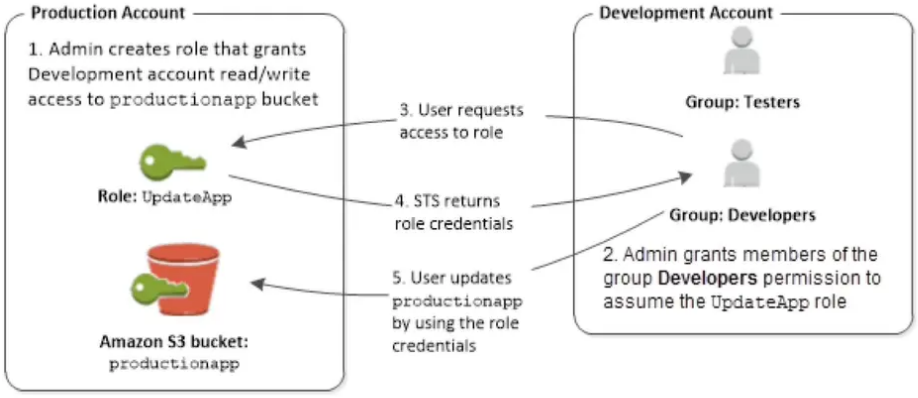
- allows to grant limited and temporary access to AWS resource (up to 1 hour)
- AssumeRole: Assume [假定] roles within your account or cross account
- AssumeRoleWithSAML : return credentials for users logged with SAML
- AssumeRoleWithWebIdentity
  - return creds for users logged with an IdP (facebook login, google login, OIDC compatible...)
  - AWS recommends against using this, using Cognito Identity Pools instead
- GetSessionToken : for MFA, from a user or AWS account root user
- GetFederationToken : obtain temporary creds for a federated user
- GetCallerIdentity : return details about the IAM user or role in the API call
- DecodeAuthorizationMessage : decode error message when an AWS API is denied

using STS to Assume a Role



- define an IAM Role within your account or cross-account
- define which principals can access this IAM Role
- use AWS STS (Security Token Service) to retrieve credentials and impersonate the IAM Role you have access to (AssumeRole API)
- temporary credentials can be valid between 15 minutes to 1 hour

cross account access with STS



https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_common-scenarios_aws-accounts.html

STS with MFA

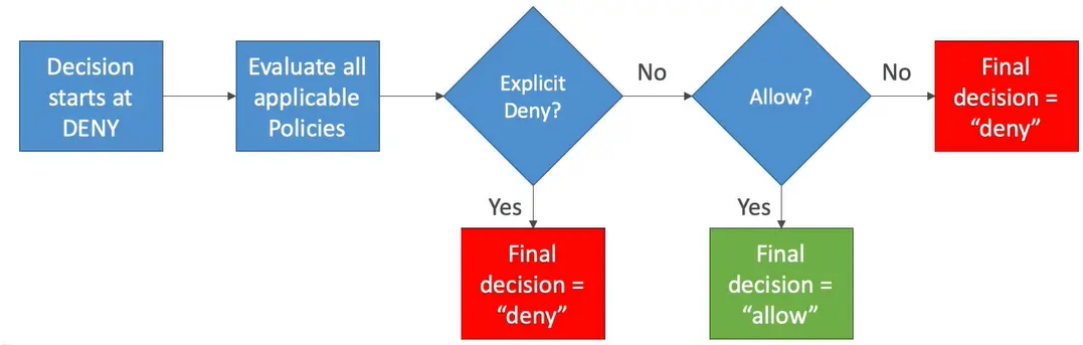
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StopInstances",
                "ec2:TerminateInstances"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Bool": {
                    "aws:MultiFactorAuthPresent": "true"
                }
            }
        }
    ]
}
```

- use GetSessionToken from STS
- appropriate [合适的] IAM policy using IAM Conditions
- aws:MultiFactorAuthPresent:true
- Reminder,GetSession Token returns
  - Access ID
  - Secret Key
  - Session Token
  - Expiration date

Advanced IAM – Authorization Model Evaluation [评估] of Policies, simplified

1. if there's an explicit [明确的] DENY, end decision and DENY

2. if there's an ALLOW, end decision with ALLOW
3. Else DENY



IAM Policies & S3 Bucket Policies

- IAM Policies are attached to users, roles, groups
- S3 Bucket Policies are attached to buckets
- when evaluating if an IAM Principal can perform an operation X on a bucket, the union of its assigned IAM Polices and S3 Buckets Policies will be evaluated

Example 1

- IAM Role attached to EC2 instance, authorizes RW to "my_bucket"
- No S3 Bucket Policy attached

- => EC2 instance can read and write to "my_bucket"

Example 2

- IAM role attached to EC2 instance, authorizes RW to "my_bucket"
- S3 bucket Policy attached ,explicit deny to the IAM Role

- => EC2 instance cannot read and write to "my_bucket"

Example 3

- IAM Role attached to EC2 instance, no S3 bucket permissions
- S3 bucket policy attached, explicit RW allow to the IAM Role

- => EC2 instance can read and write to "my_bucket"

Example 4

- IAM role attached to EC2 instance, explicit deny S3 bucket permissions
- S3 bucket policy attached, explicit RW allow to the IAM Role

- => EC2 instance cannot read and write to "my_bucket"

Dynamic Policies with IAM

- how do you assign each user a /home/<user> folder in an S3 bucket ?
- Option 1
  - create an IAM policy allowing georges to have access to /home/georges
  - create an IAM policy allowing sarah to have access to /home/sarah
  - create an IAM policy allowing matt to have access to /home/matt
  - ..on policy per user
  - this doesn't scale
- Options 2
  - create on dynamic policy with IAM
  - leverage the special policy variable ${aws:username}
  -
```
{
   "Sid": "AllowAllS3ActionsInUserFolder",
   "Action":["s3:*"],
   "Effect":"Allow",
   "Resource": ["arn:aws:s3:::my-company/home/${aws:username}/*"]
}
```

Inline vs Managed Polices

- AWS Managed Policy
  - maintained by AWS
  - good for power users and administrators
  - updated in case of new services / new APIs
- Customer Managed Policy
  - best practice, re-usable, can be applied to many principals
  - version controlled + rollback, central change management
- Inline
  - strice one-to-one relationship between policy and principal
  - policy is deleted if you delete the IAM principal

Granting a User Permissions to Pass a Role to an AWS Service

- to configure many AWS services,you must **pass** an IAM role to the service (this happens only once during setup)
- the service will later assume the role and perform actions
- example of passing a role
  - to an EC2 instance
  - to a lambda function
  - to an ECS task
  - to CodePipeline to allow it to invoke other services
- for this, you need the IAM permission **iam:PassRole**
- it often comes with iam:GetRole to view the role being passed

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/S3Access"
        }
    ]
}
```

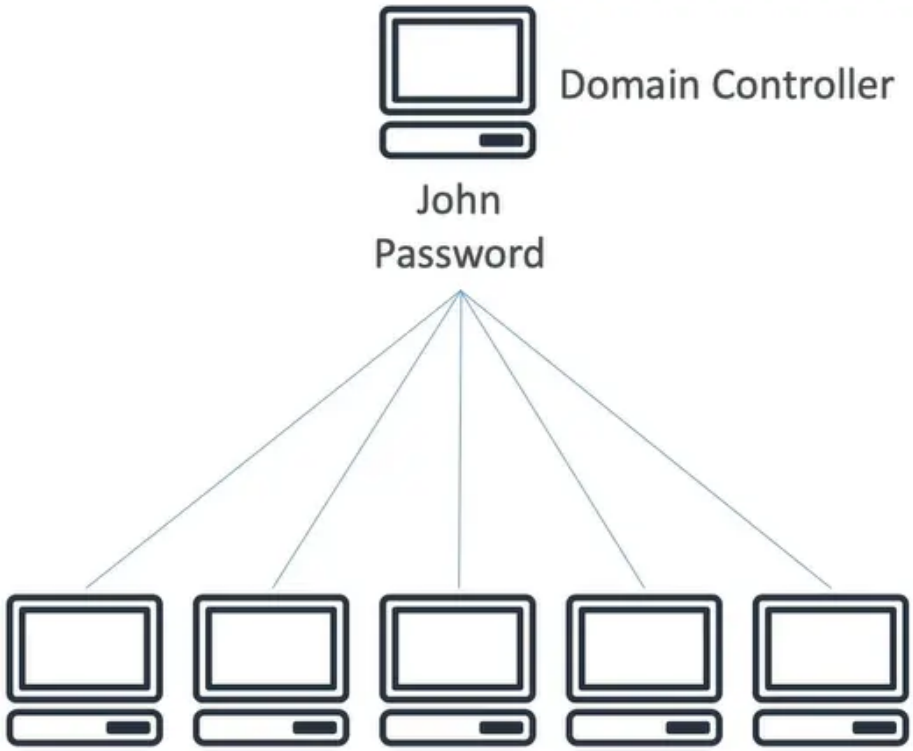
can a role be passed to any service ?

- No: Roles can only be passed to what their trust allows
- *a trust policy for the role that allows the service to assume [承担] the role*

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Sid": "TrustPolicyStatementThatAllowsEC2ServiceToAssumeTheAttachedRole",
        "Effect": "Allow",
        "Principal": {
            "Service": "ec2.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
}
```

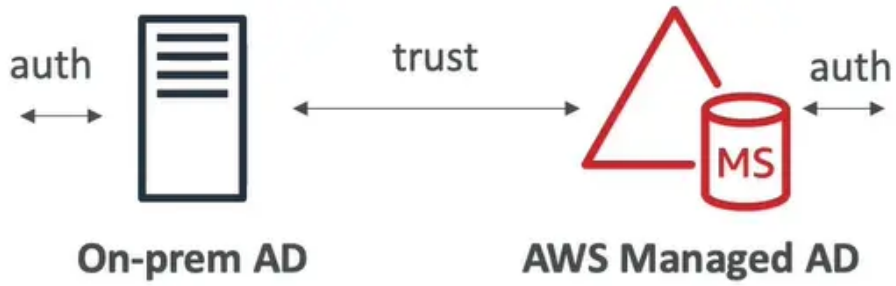
AWS Directory [目录] Services

what is Microsoft Active Directory (AD) ?

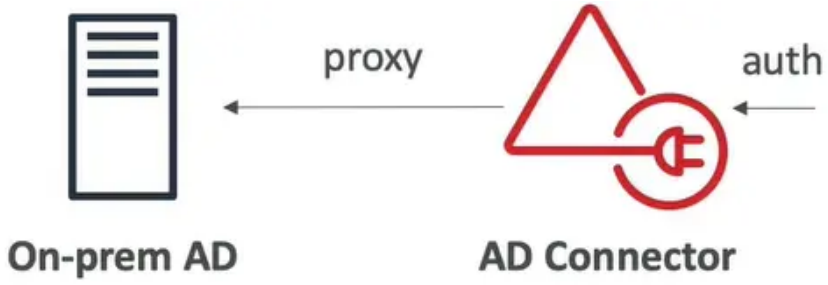


- found on any Windows Server with AD Domain Services
- Database of objects : user accounts, computers, printers,file shares, security groups
- centralized [统一]security management, create account, assign permissions
- objects are organized in trees
- a group of tree is a forest

AWS Directory Services



- AWS Managed Microsoft AD
  - create your own AD in AWS, manage uers locally, supports MFA
  - establish ”trust” connections with your on-permise AD



- AD connector
  - directory gateway (proxy) to redirect to on-premise AD, supports MFA
  - users are managed on the on-premise AD
- Simple AD

Simple AD

- AD-compatible managed directory on AWS
- cannot be joined with on-premise AD