What is AWS IAM

IAM is Identity Access Management.

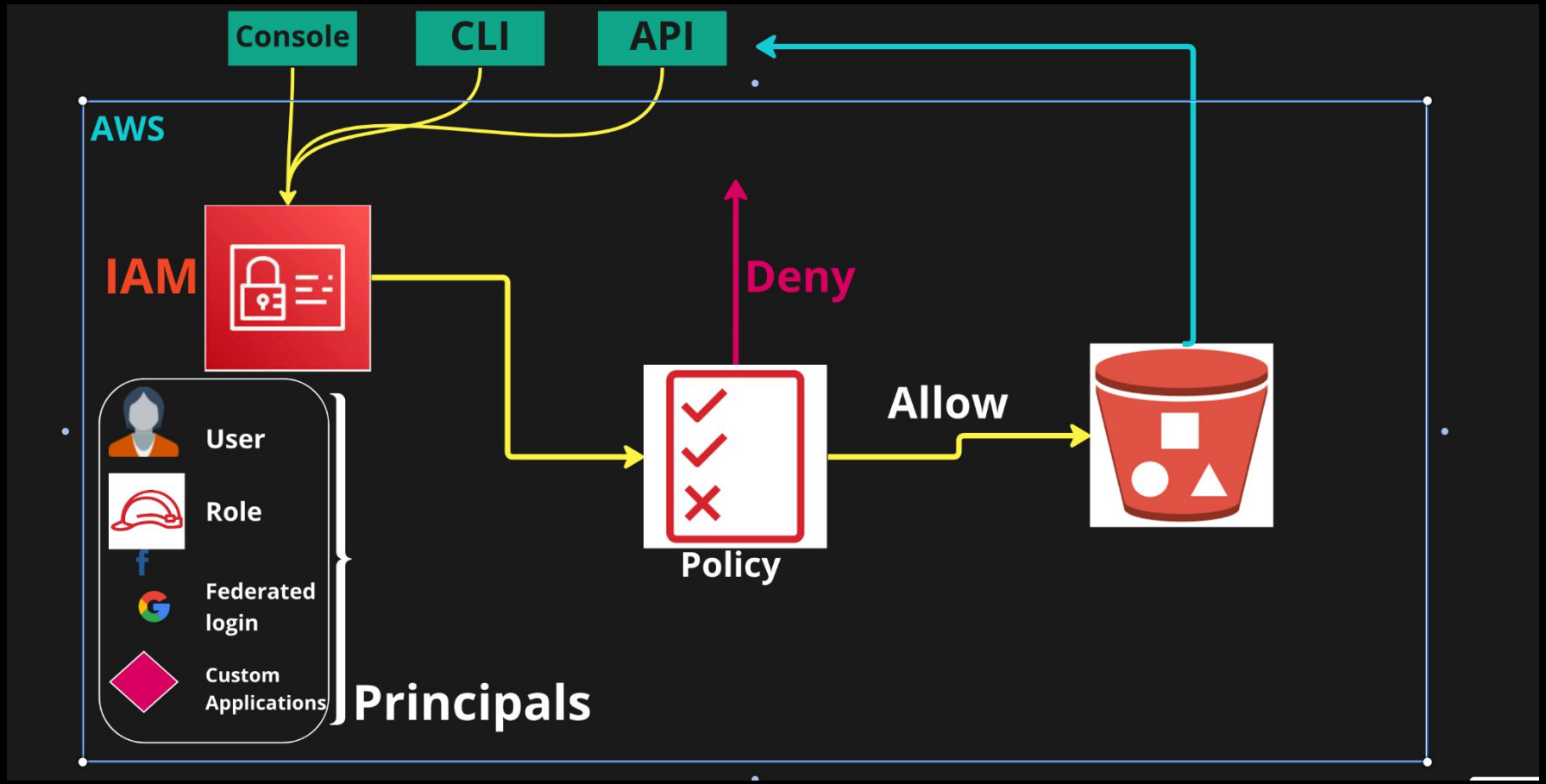
- Authentication
- Authorisation

Agenda

- User, Policy,Role, Groups
- Authentication methods
- Policy (managed or custom)
  - Permission policy
    - Identity Policy, Resource base policy
  - Trust policy

- SCP, Permission boundaries
- Cross account access
- Access control using
  - RBAC (Role-based access control)
  - ABAC (Attribute-based access control)

How IAM Works



User, Policy, Role, Groups

- User is an account for a user
- No permission in default
- 5000 user accounts are allowed per an account
- Policy
  - Permission policy
    - Identity-based policies apply to the principles
    - Resource-based policies apply to resources like S3
  - Trust policy
- Demo

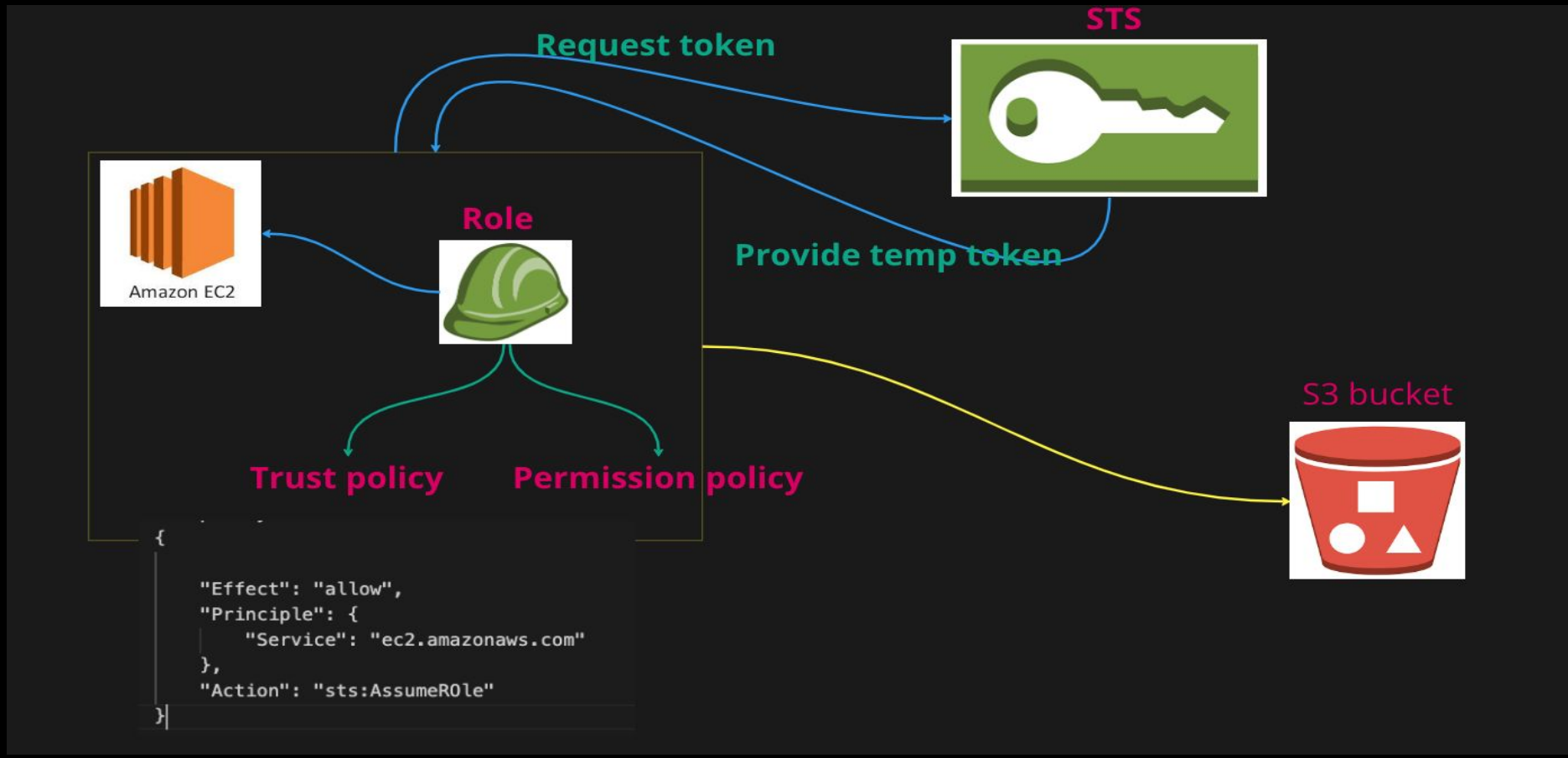
Authentication methods

- Username/password
- Access key/ Secret
- Signing certificate
  - Amazon EC2
- SSH key/HTTPS GIT credentials
  - AWS CodeCommit/AWS CodePipeline

STS (Secure Token Service)

- Responsible for creating short-lived token
  - EC2 instance access S3 bucket
  - Trust policy is used here

STS




```
{
    "Effect": "allow",
    "Principle": {
        "Service": "ec2.amazonaws.com"
    },
    "Action": "sts:AssumeROle"
}
```

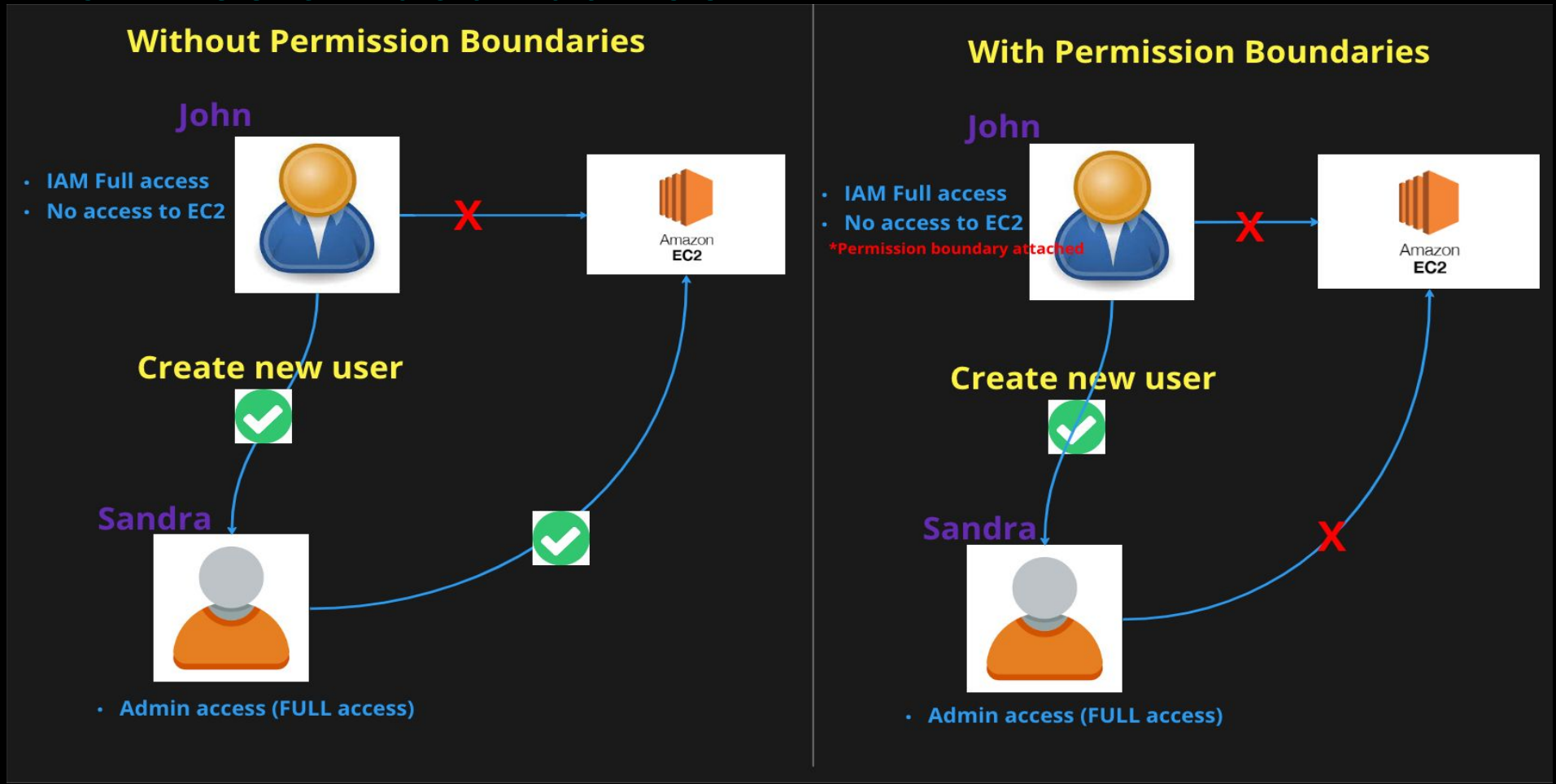
SCP (Service control policy)

- SCP can be used to control access in AWS account level
- https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_scps.html

Permission boundaries

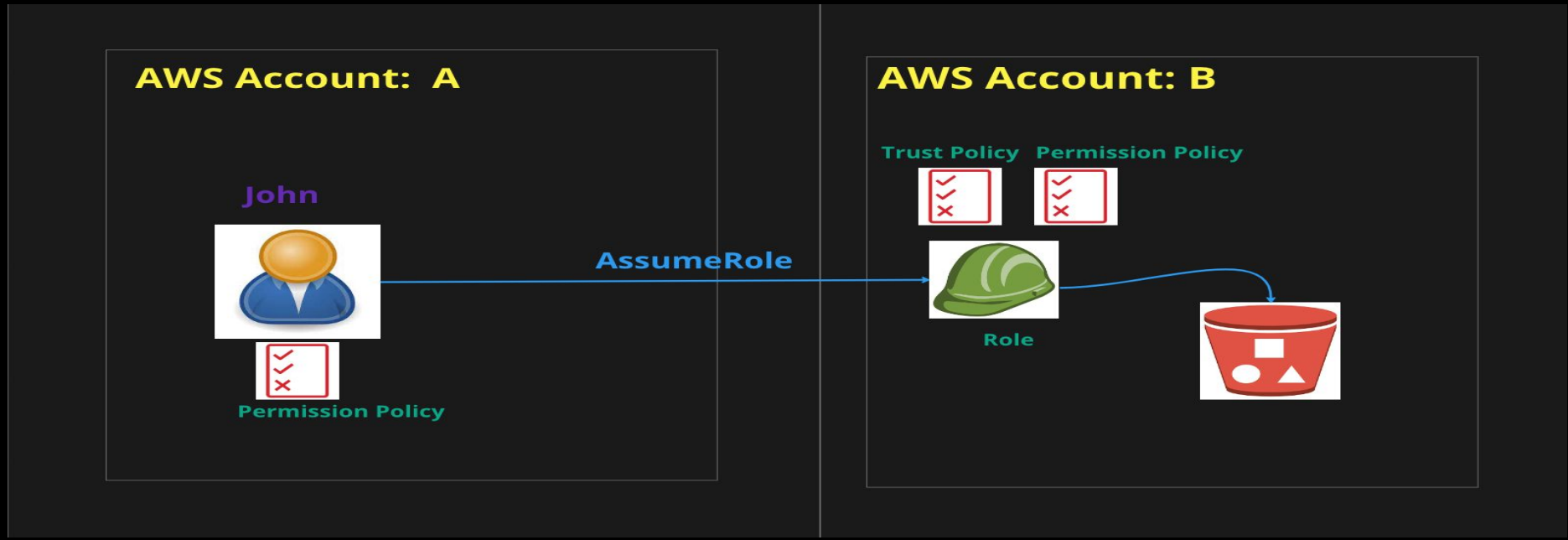
- Avoid privilege escalation
- If a user has permission boundary, this user cannot create another user without adding permission boundary
- The user cannot create another user with more privilege than what he/she currently has

Permission boundaries



Cross account access

- Accessing a resource from another AWS account



RBAC, ABAC

- https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction_attribute-based-access-control.html
- Demo