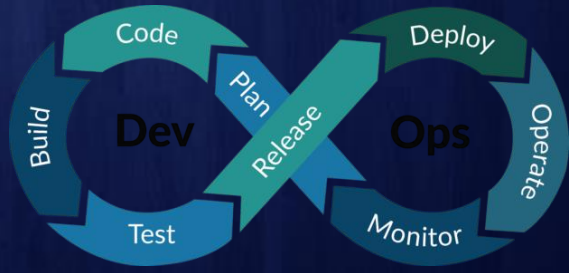
AWS SAA + SysOps + Developer + DevOps Course #Day-9

We will start at 8 AM,
Stay tuned



RAKESH TANINKI
LEARN TO UNLEARN

Recap:

- Levels of Storage
- S3 Service
- Hands on – S3

Today's topics:

- JSON – Introduction & Usage
- IAM Advanced
  - IAM Policies
    - AWS Managed Policies
    - Inline Policies
    - Customer Managed Policies
  - IAM Roles
    - Same account role assumption
    - Cross account role assumption
    - Service linked role
    - Pass Role
- Demos

Java Script Object Notation (JSON)

Java Script Object Notation (JSON)

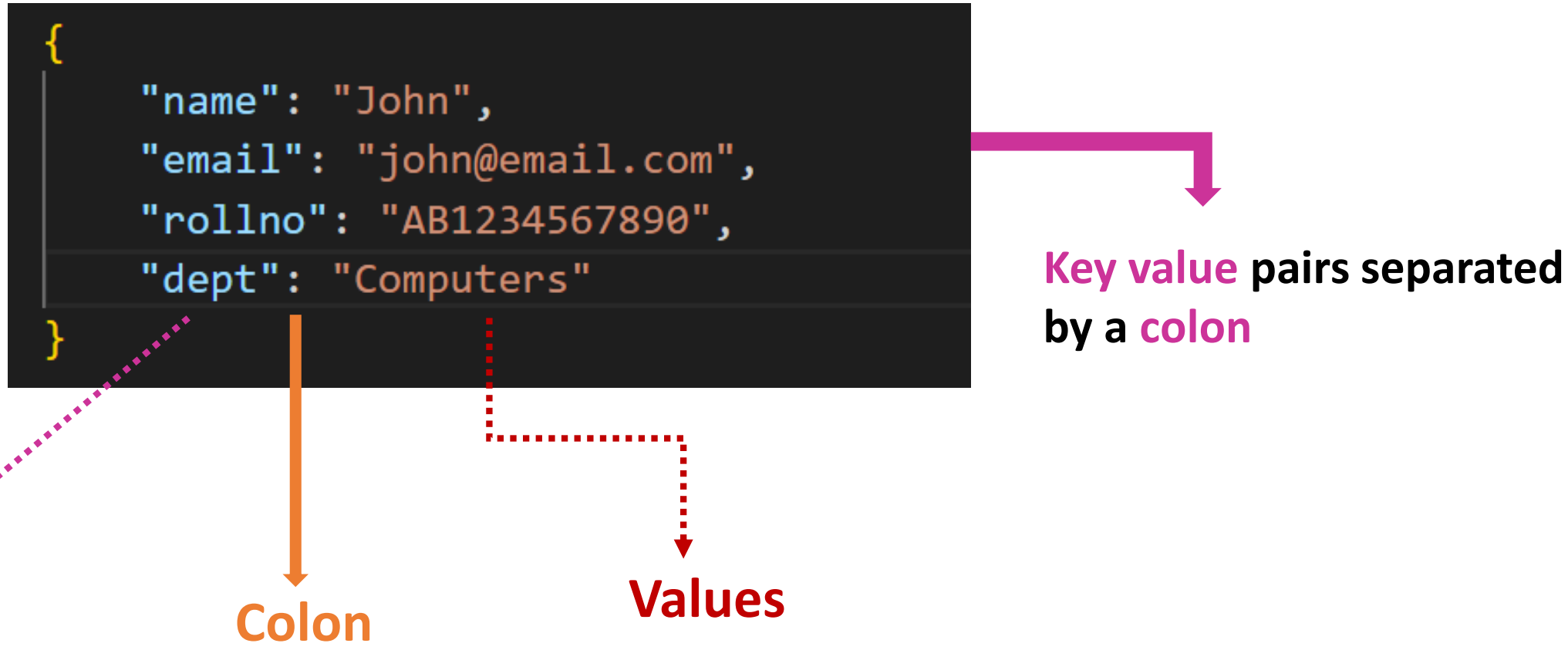
- Easy way for data exchange between servers
- Easy for machines to read / generate / parse
- Successor to XML
- JSON is used in AWS Cloud Formation / IAM Policies / SCP

- JSON is enclosed in { }

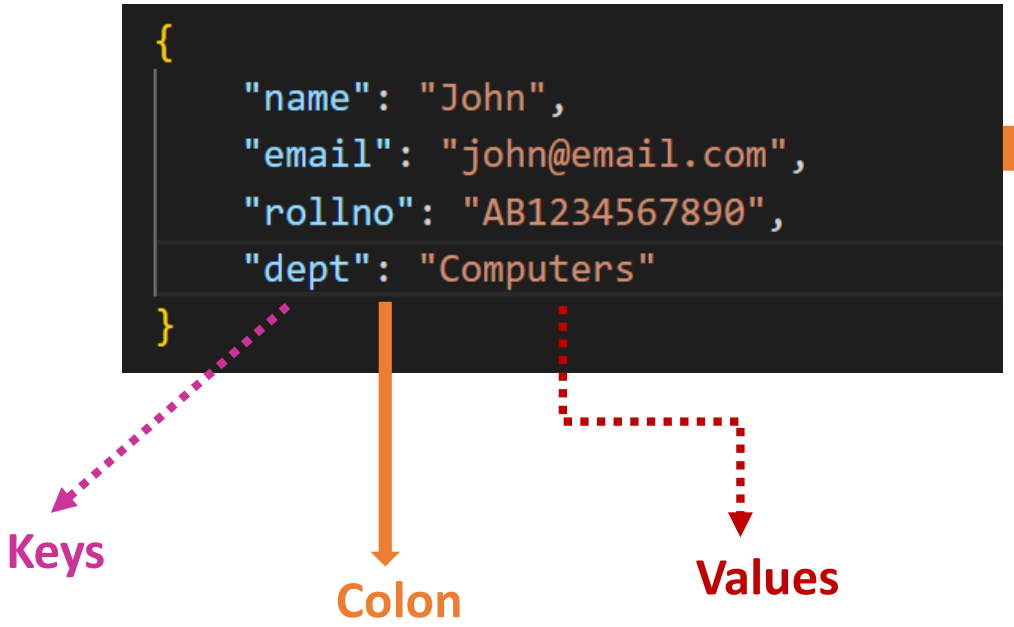
```
{ "name": "John", "email": "john@email.com" }
```

Object

Java Script Object Notation (JSON)



Java Script Object Notation (JSON)




```
{
    "name": "John",
    "email": "john@email.com",
    "rollno": "AB1234567890",
    "dept": "Computers",
    "mobile": "1223456789"
}
```

Java Script Object Notation (JSON)

- Syntax:
  - Keys are always must be a string enclosed in “ ”
  - Values can be of different types

1. String – “”
2. Number – 100 or 21 or 33 etc.
3. JSON – {}
4. Array – []
5. Boolean – true, false
6. null

Java Script Object Notation (JSON)

Address is added as string

Address is added as JSON (nested)

Java Script Object Notation (JSON)

- Number: 1, 2, 3
- Array: [“123456789”, “9876543221”]
- Bool: true / false
- null

Examples – IAM Policy

Examples – Cloud Formation

```
{
    "Resources": {
        "HelloBucket": {
            "Type": "AWS::S3::Bucket",
            "Properties": {
                "BucketName": "MyFirstBucket"
            }
        }
    }
}
```

AWS Policies

AWS Policies

- Policies are used to manage permissions across AWS
- Types:
  1. Identity based policies
  2. Resource based policies
  3. Permission boundaries
  4. Organization SCPs
  5. ACLs
  6. Session Policies

Ref: click here

Identity based policies

- IAM Policies – to manage permissions of IAM identities – Users, Groups, Roles
- Allow / Deny access to identities – Deny has priority over Allow
- For SAA exam, you should be able to understand the policy

Ref: click here

Identity based policies

1st part

2nd part

Statements

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3 Policy",
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "*"
        },
        {
            "Sid": "Deny Policy",
            "Effect": "Deny",
            "Action": ["s3:Delete*", "s3:List*"],
            "Resource": ["arn:aws:s3:::example_bucket", "arn:aws:s3:::example_bucket/*"]
        }
    ]
}
```

Ref: click here

Statements - Identity based policies

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "S3 Policy",
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": "*"
      },
      {
        "Sid": "Deny Policy",
        "Effect": "Deny",
        "Action": ["s3:Delete*", "s3:List*"],
        "Resource": ["arn:aws:s3:::example_bucket", "arn:aws:s3:::example_bucket/*"]
      }
    ]
}
```

Allow / Deny

String or Array (actions)

String or Array (ARNs)

Ref: [click here]

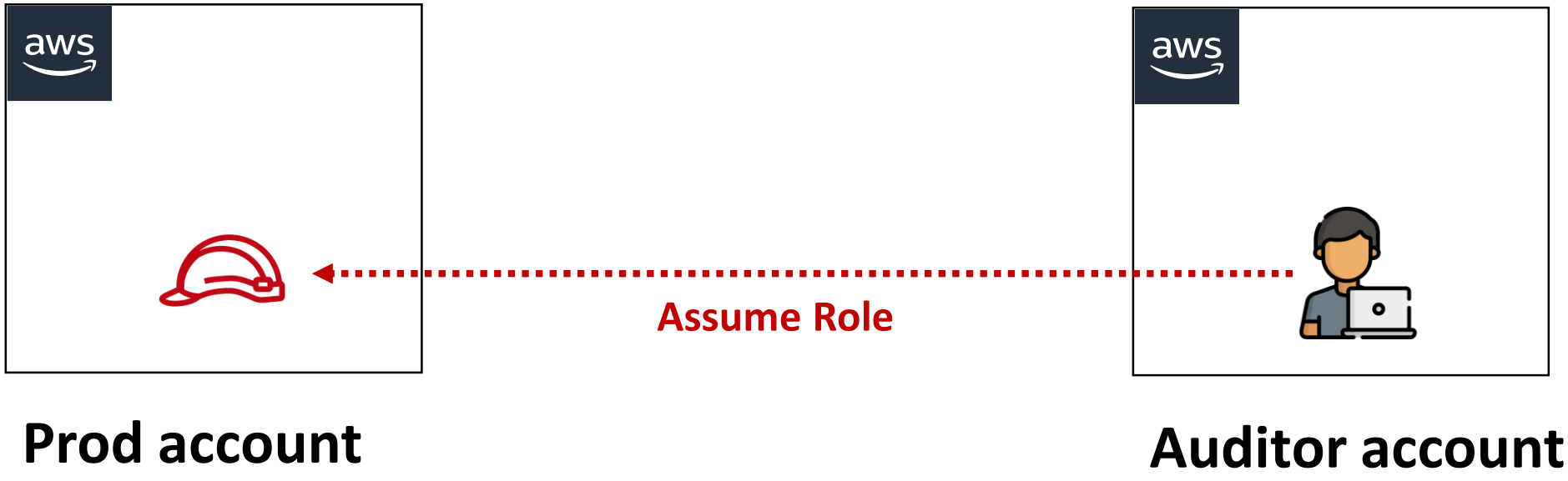
Statements - Identity based policies

The information in a statement is contained within a series of elements.

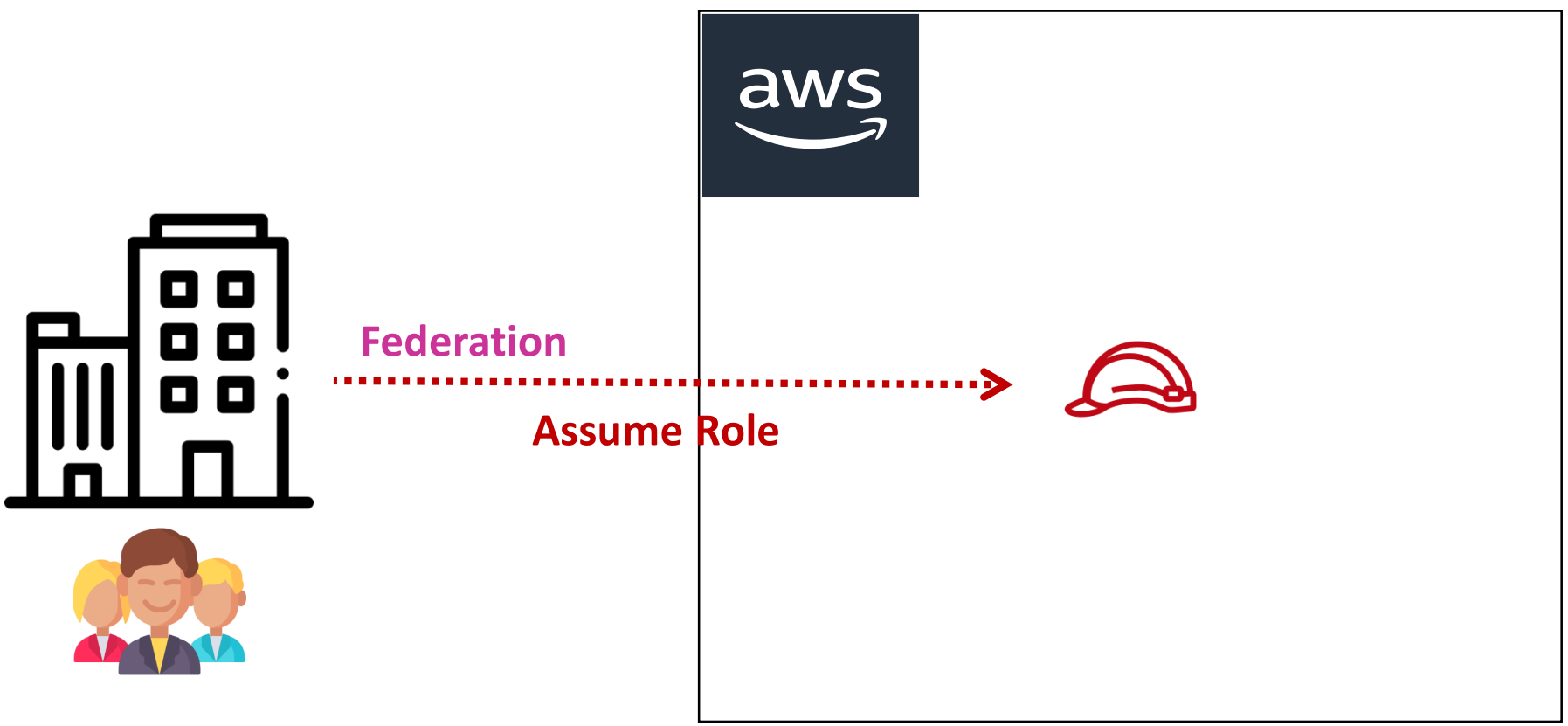
- **Version** – Specify the version of the policy language that you want to use. We recommend that you use the latest `2012-10-17` version. For more information, see IAM JSON policy elements: Version
- **Statement** – Use this main policy element as a container for the following elements. You can include more than one statement in a policy.
- **Sid** (Optional) – Include an optional statement ID to differentiate between your statements.
- **Effect** – Use `Allow` or `Deny` to indicate whether the policy allows or denies access.
- **Principal** (Required in only some circumstances) – If you create a resource-based policy, you must indicate the account, user, role, or federated user to which you would like to allow or deny access. If you are creating an IAM permissions policy to attach to a user or role, you cannot include this element. The principal is implied as that user or role.
- **Action** – Include a list of actions that the policy allows or denies.
- **Resource** (Required in only some circumstances) – If you create an IAM permissions policy, you must specify a list of resources to which the actions apply. If you create a resource-based policy, this element is optional. If you do not include this element, then the resource to which the action applies is the resource to which the policy is attached.
- **Condition** (Optional) – Specify the circumstances under which the policy grants permission.

Ref: click here

AWS Roles



AWS Roles



AWS Roles

Web Federation

Assume Role

AWS IAM Demo - Advanced

- IAM Policies – on users, groups, roles
- Allow / Deny priorities
- IAM role trust policy / permissions policy
- Same account role login
- Cross account role login

Thank you, will meet in tomorrow's session