**Accounts**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AWS Account Type** | **AWS Account Name** | **Email address** | **Sign in URL** | **MFA Enabled?** |
| General/Management Account | AWS-Train-General | Danmonchusi@gmail.com |  | Yes |
| Production Account | AWS-Train-Prod | [Danmonchusi1@gmail.com](mailto:Danmonchusi1@gmail.com) |  | Yes |
| Account3 (Name TBD) |  | Danmonchusi2@gmail.com |  |  |
| Account4 (Name TBD) |  | Danmonchusi3@gmail.com |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**IAM Accounts**

|  |  |  |
| --- | --- | --- |
| **IAM Account Type** | **IAM Account Name** | **Sign in Url** |
| General IAM Account | Iam1-train-general  Uname: iamadmin  P:G\*\*\*\*ng! | https://iam1-train-general.signin.aws.amazon.com/console |
| Production IAM Account | iam1-train-production  Uname: iamadmin  P:G\*\*\*\*ng! | https://iam1-train-production.signin.aws.amazon.com/console |

**CLI profiles**

1. iamadmin-general
2. iamadmin-production

**AWS accounts**

Set up AWS accounts and associate accounts with MFA devices

Use CloudWatch to create billing alarms

**\*AWS Solutions Architect**

What does a solutions architect do? Designs solutions for moving to the cloud – they are a metric, rubric, a tool that allows users to measure the requirements for implementing cloud solutions. So, they gather requirements from customers and identify the best use case and cloud fit scenario for a customer’s needs. In a nutshell aligning a user’s requirements with what AWS has to offer.

*Solutions alignment + requirements = Architecting*

*\*The course provides the student with the skills to become immediately effective*

**SysOps/Devops administrator**– maintains the resources and ensures that resources systems are running optimally Managing an **implementation** of the system designed by the Architect.

*Sysops = manage resources*

**Developer** – Develops the tools used n the solution

**T Shaped skills** – metaphor that describes the abilities of a person in the workforce, the vertical bar represents the depth of the persons skills in a single discipline and the horizontal bar represents the person’s ability to collaborate across disciplines with experts in other areas and to apply knowledge in areas of expertise other than one’s own.

**Multi-Factor Authentication**

Factor – different pieces of evidence that prove identity

Different Factors

* Knowledge -something you know (username, password)
* Possession – something you have (bank card, MFA app)
* Inherent – something you are/personal physical feature (fingerprint, face, voice, iris)
* Location – location, network, or domain

**Principle of Least Privilege**

A user is given the minimum levels of access – or permissions – needed to perform his/her job functions.

**IAM Basics**

**\*Not best practice to access AWS using the root account. Best practice is to use an IAM account**

IAM is an AWS service that allows you to create the following identity objects:

1. IAM Users – Identities which represent humans or applications that need access to your account
2. IAM Groups – collection of related users eg. HR, development or Finance
3. IAM Roles – Can be used by AWS services or for granting external access to your account. Used to grant access to an uncertain number of users, instances or applications.

IAM identities start with no permissions on an AWS account and can be granted permissions (almost) up to those of the Account root user.

Every AWS account has its own running copy of IAM

IAM is a globally resilient service, so any data is secure across all regions

Policies/Policy documents – Used to ALLOW or DENY access to AWS services BUT only if they are attached to User, Group or Roles.

IAM 3 main jobs:

1. Manages Identities – An ID provider (IDP). It lets you create, delete, and modify identities such as users and roles.
2. Authenticates Identities
3. Authorizes users to access AWS resources

No costs involved for use of IAM but there is a limit to the number of users, groups or roles that can be created using IAM service.

Global service/ Globally resilient

ALLOWS or DENIES **its** identities. It can only control users and roles created in IAM.

NO direct control on external accounts or users

IAM makes use of IDENTITY FEDERATION and MFA

**Creating an IAM Admin account**

1. Login to console with root account
2. Access IAM service
3. Customize sign-on url for IAM users for that account
4. Copy link and store in safe place (the link will be used whenever you need to login to the IAM account)
5. Select Add user
6. Set user details
   1. Access Type – Allow Management Console Access
   2. Console Password – Custom Password
   3. Uncheck Require password rest
7. Set Permissions
   1. Attach existing policies directly
   2. Check Administrator Access Policy (This policy provides full access almost up to root user)
8. No tags
9. Review
10. Create user
11. Login as iamadmin and associate account with MFA device
    1. Account Security details
    2. Assign MFA device

**IAM Access Keys**

IAM Access Keys are used to provide authentication for the CLI (Command Line Interface)

A type of long-term credentials – Credentials that are not changed regularly and the user has to explicitly change them.

An IAM user can have only one username and only one Password

An IAM user can have two access keys

Access Keys can be created, deleted or made inactive or active

Access Keys are made of two parts

* + 1. Access key ID 
    2. Secret Access key



**Cloud Computing**

Public Cloud – Using 1 public cloud

Private Cloud – Using on Premises REAL cloud

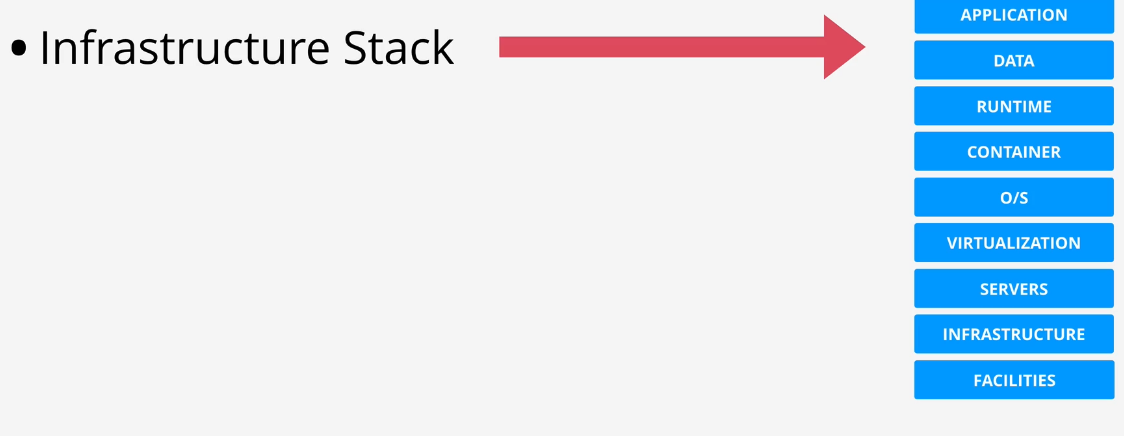
Multi Cloud – Using more than 1 Public cloud

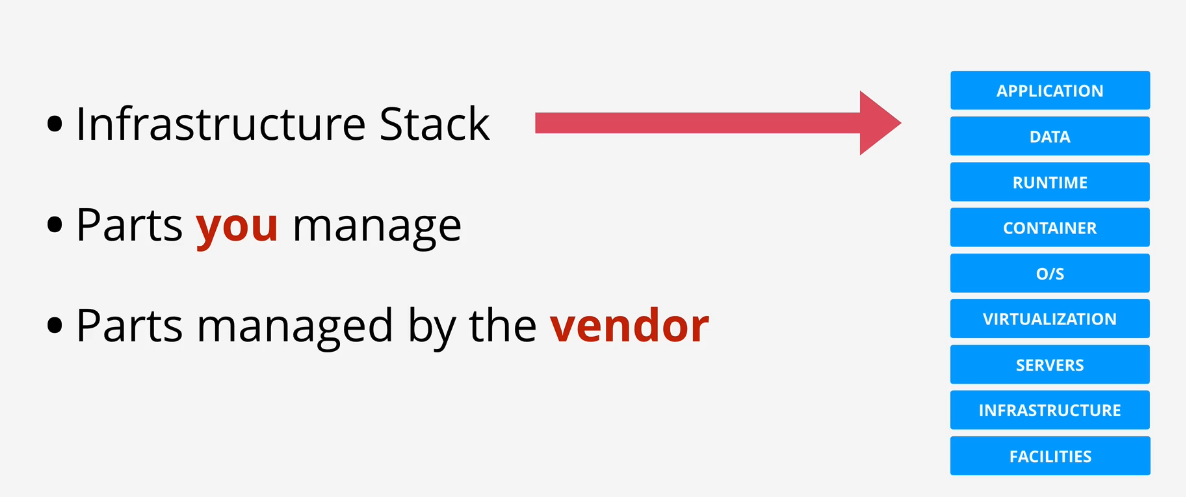
Hybrid Cloud = Private and Public Clouds

**Cloud Service Models (xAAS)**

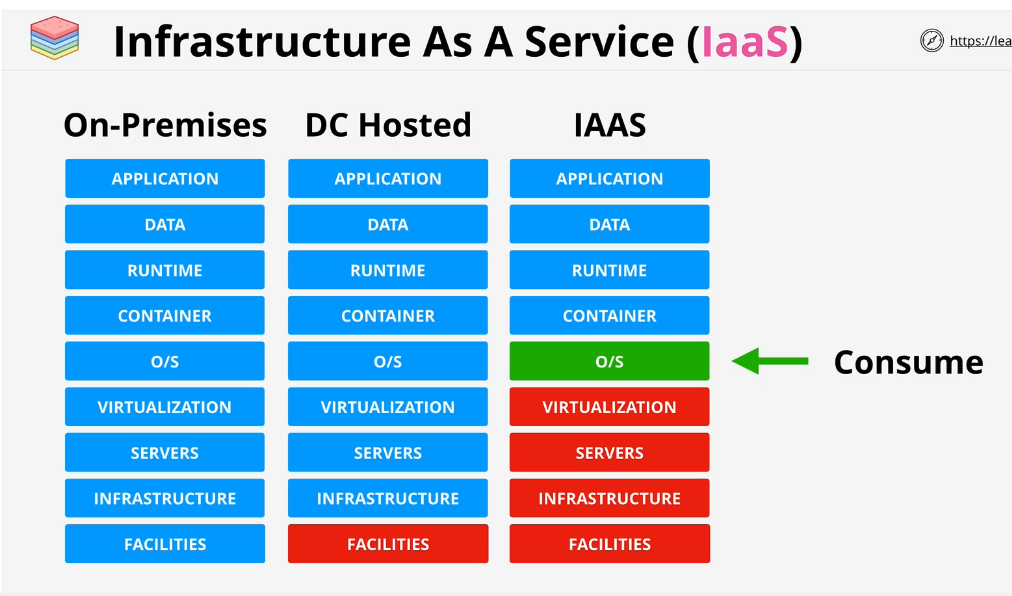
When an application is deployed, it makes use of an infrastructure stack

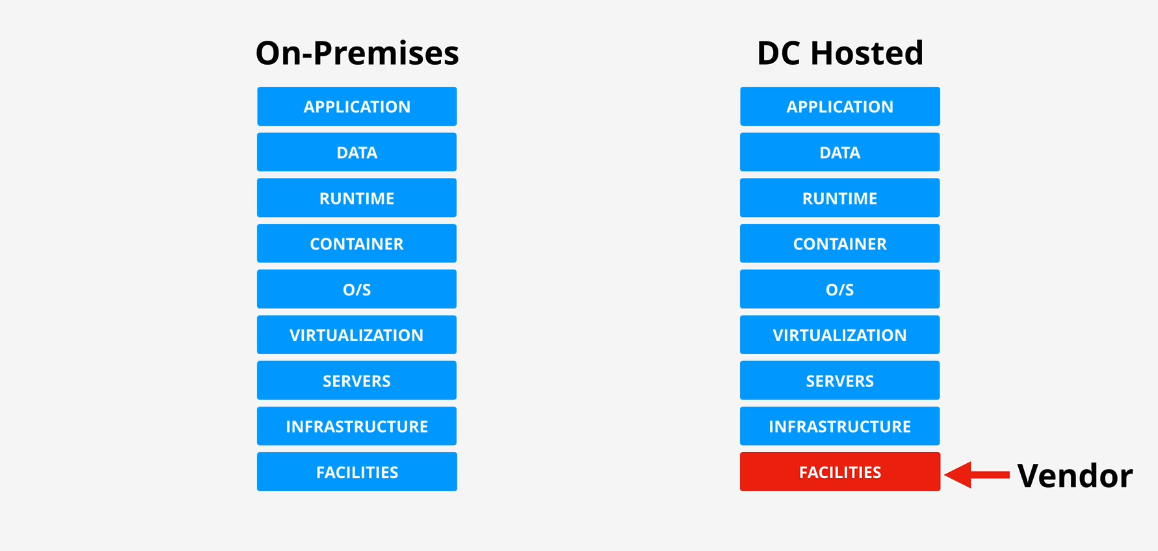
Infrastructure stack are sets of hardware and services combined together that make information available on a webpage or or to deploy an application.



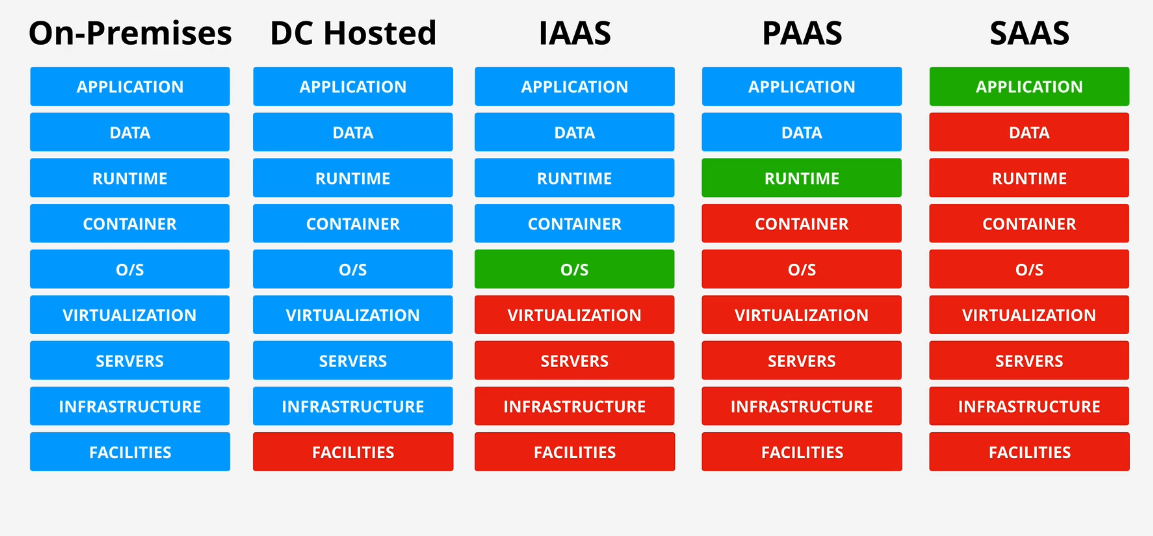


**Units of consumption**





**Service models**



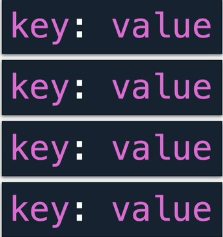
Unit of Consumption

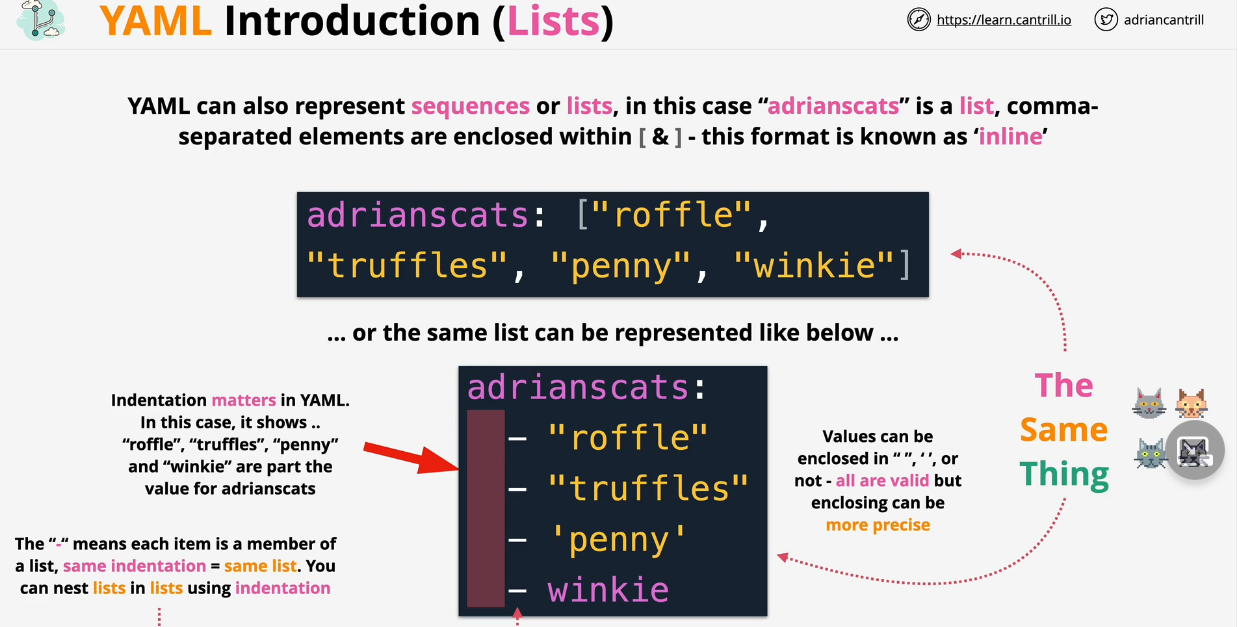
**YAML (YAML ain’t Markup language)**

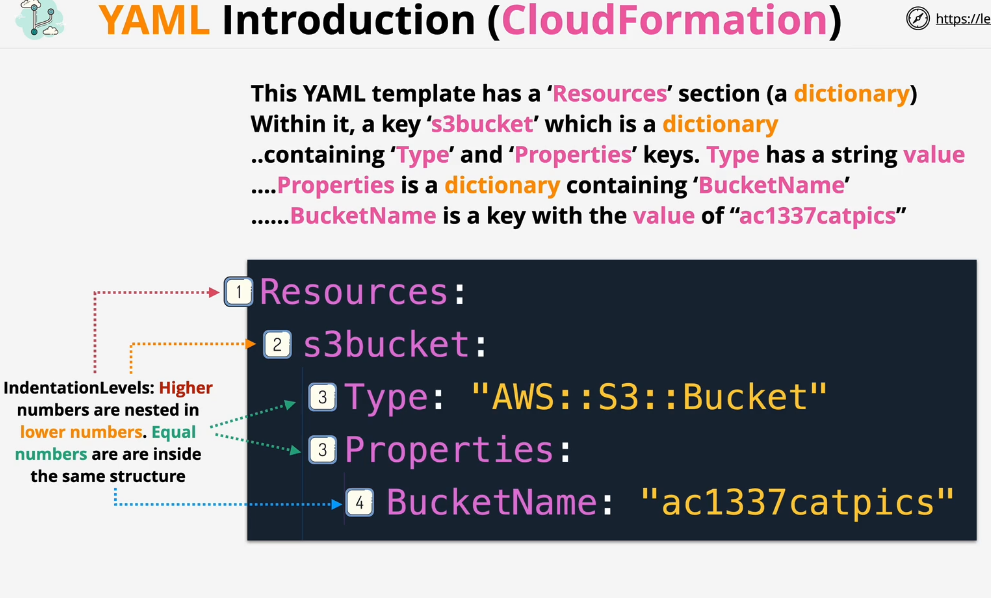
One of the languages cloud formation uses for its templates

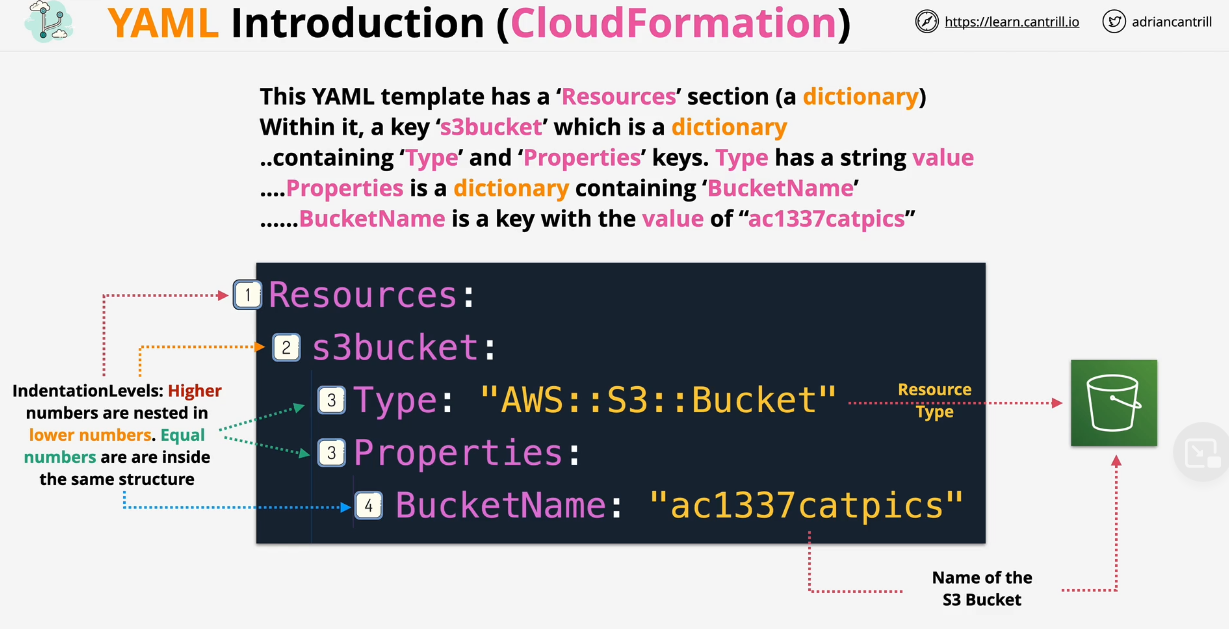
A human readable language design for data serialization i.e its used for defining data

A YAML document is an ordered collection of key:value pairs. Each key has a value









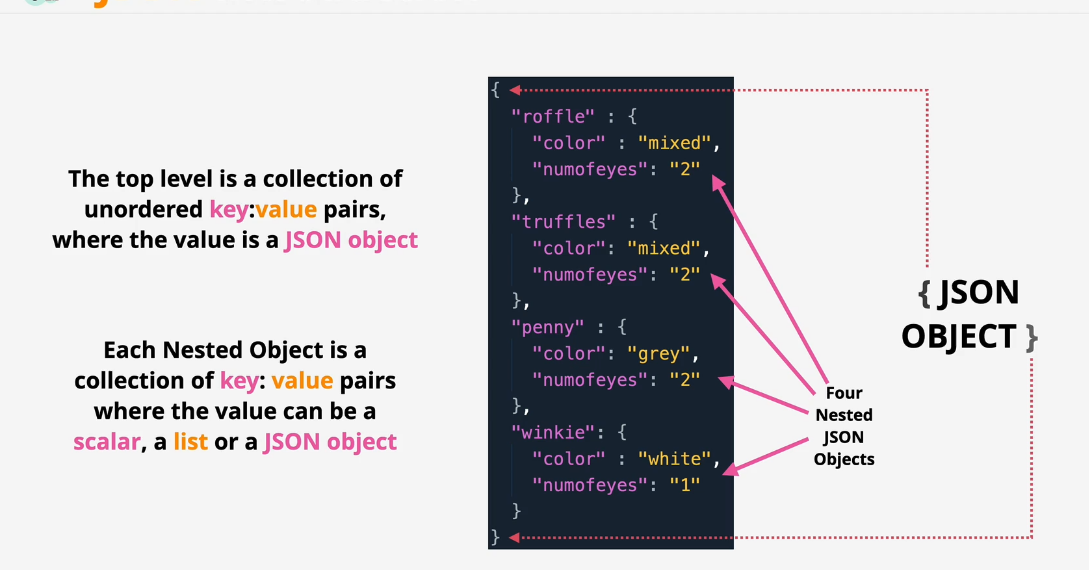
**JSON (JavaScript Object Notation)**

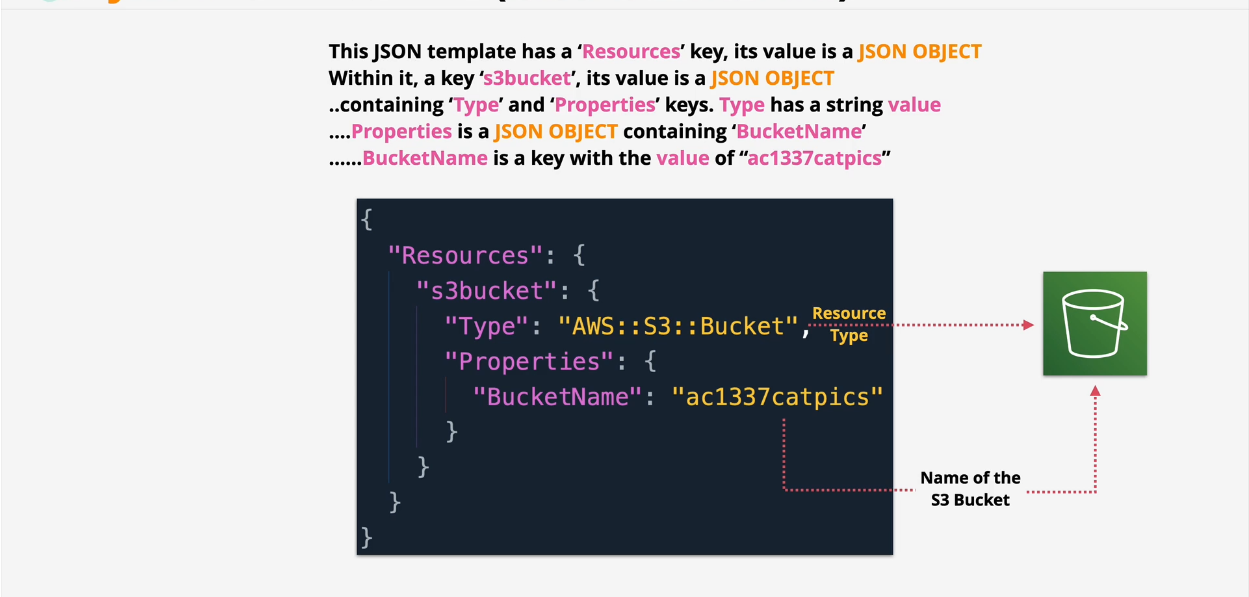
A lightweight data-interchange format

Easy for humans to read and write and easy for machines to parse and generate.

Indentation in Json does not matter because everything is placed inside curly brackets

JSON is used in CloudFormation and identity policies within AWS whereas YAML is used only in CloudFormation.





**Differences between YAML and JSON are:**

|  |  |
| --- | --- |
| **YAML** | **JSON** |
| Comments are denoted with a hash/number sign. | Comments are not allowed. |
| Hierarchy is denoted by using double space characters. Tab characters are not allowed. | Objects and Arrays are denoted in braces and brackets. |
| String quotes are optional but it supports single and double quotes. | Strings must be in double quotes. |
| Root node can be any of the valid data types. | Root node must either be an array or an object. |

**Networking**

**OSI Model**

Conceptual model that breaks down networking into 7 distinct layers and describes how the software transfers data between the layers.

Media Layers – Conceptualy these deal with how data is transferred from point A to point B and could be in the same network or across the world.

Host Layers – Deal with how data and re-assembled for transport and formatted so that it’s understandable to both sides of a network connection.

