

INTRODUCTION TO YAML & DOCKER COMPOSE

By VIGNESH S

MARKUP LANGUAGE

A markup language is a way to add instructions and formatting to plain text so that computers can understand how to display or process it. It's like giving special hints or tags to the text to tell a computer what to do with it.

Human and Machine Readable: Importantly, markup languages are designed to be both human-readable and machine-readable. Humans can understand the text with the tags, and computers can interpret the tags to display or process the content correctly.

POPULAR LANGUAGES

HTML (HyperText Markup Language):

HTML is primarily used for creating and structuring content on the World Wide Web.

XML (eXtensible Markup Language):

- Purpose: XML is a versatile markup language used for structuring and exchanging data between different systems. It's not limited to the web and can be used in various applications.**

YAML

"YAML Ain't Markup Language" or sometimes "Yet Another Markup Language," is a human-readable data serialization format. It is often used for configuration files and data exchange between languages with different data structures.

As a Cloud Engineer with an interest in technology, you'll likely encounter YAML frequently, especially when dealing with configuration files for cloud services, containers, and infrastructure as code. YAML's simplicity and readability make it a popular choice in these areas.

Features

- 1. Readability:** YAML is designed to be easy for humans to read and write. It uses indentation and a clean structure of key-value pairs to represent data.
- 2. Syntax:** YAML uses indentation to define the hierarchy of data. It uses colons to separate keys from values and dashes for lists or arrays.
- 3. Data Types:** YAML supports various data types, including strings, numbers, booleans, lists, and dictionaries (key-value pairs). This flexibility allows you to represent complex data structures.
- 4. Comments:** YAML supports comments, which are denoted by the "#" symbol. You can add comments to provide explanations or context within your configuration files.

Data Types

Strings: Think of strings as words or sentences.

Numbers: Numbers are just like the ones you use in math.

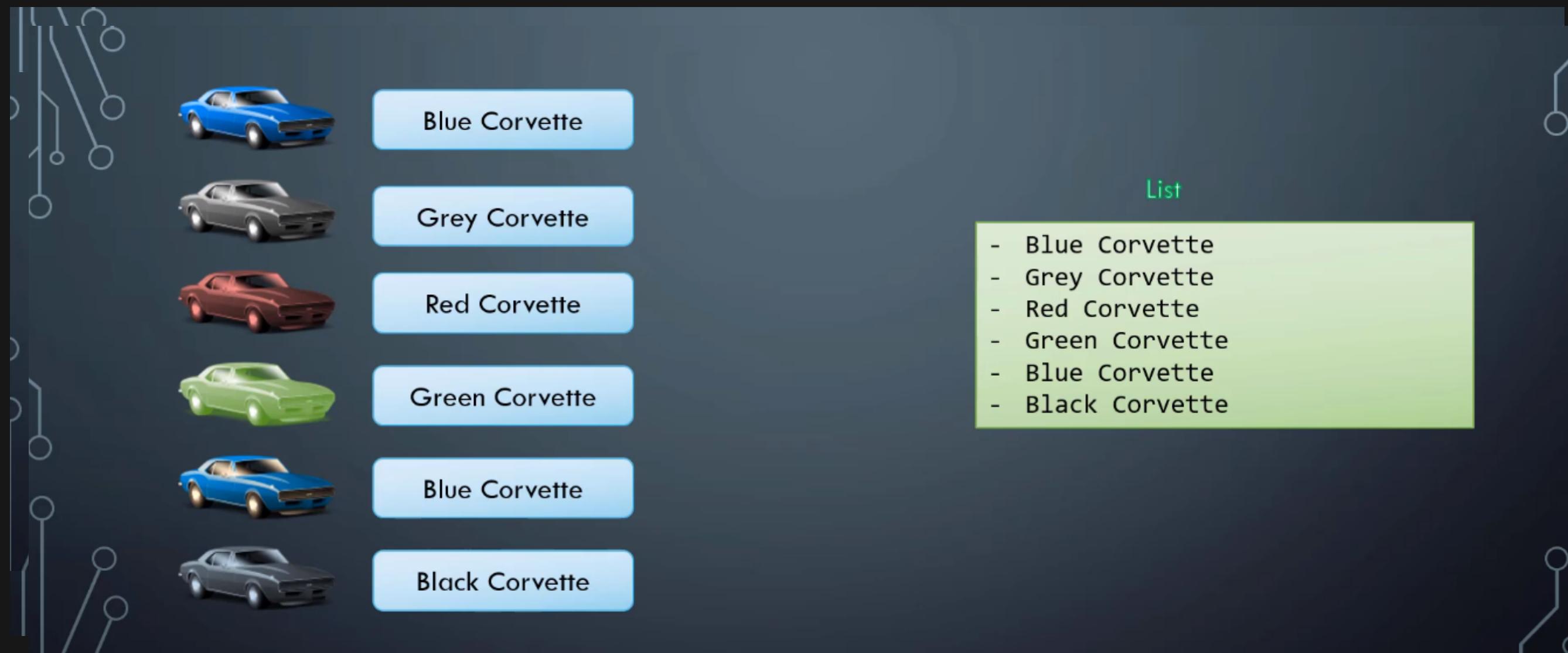
Booleans: Booleans are like simple yes-or-no answers.

A list is a data structure that holds an ordered collection of elements.

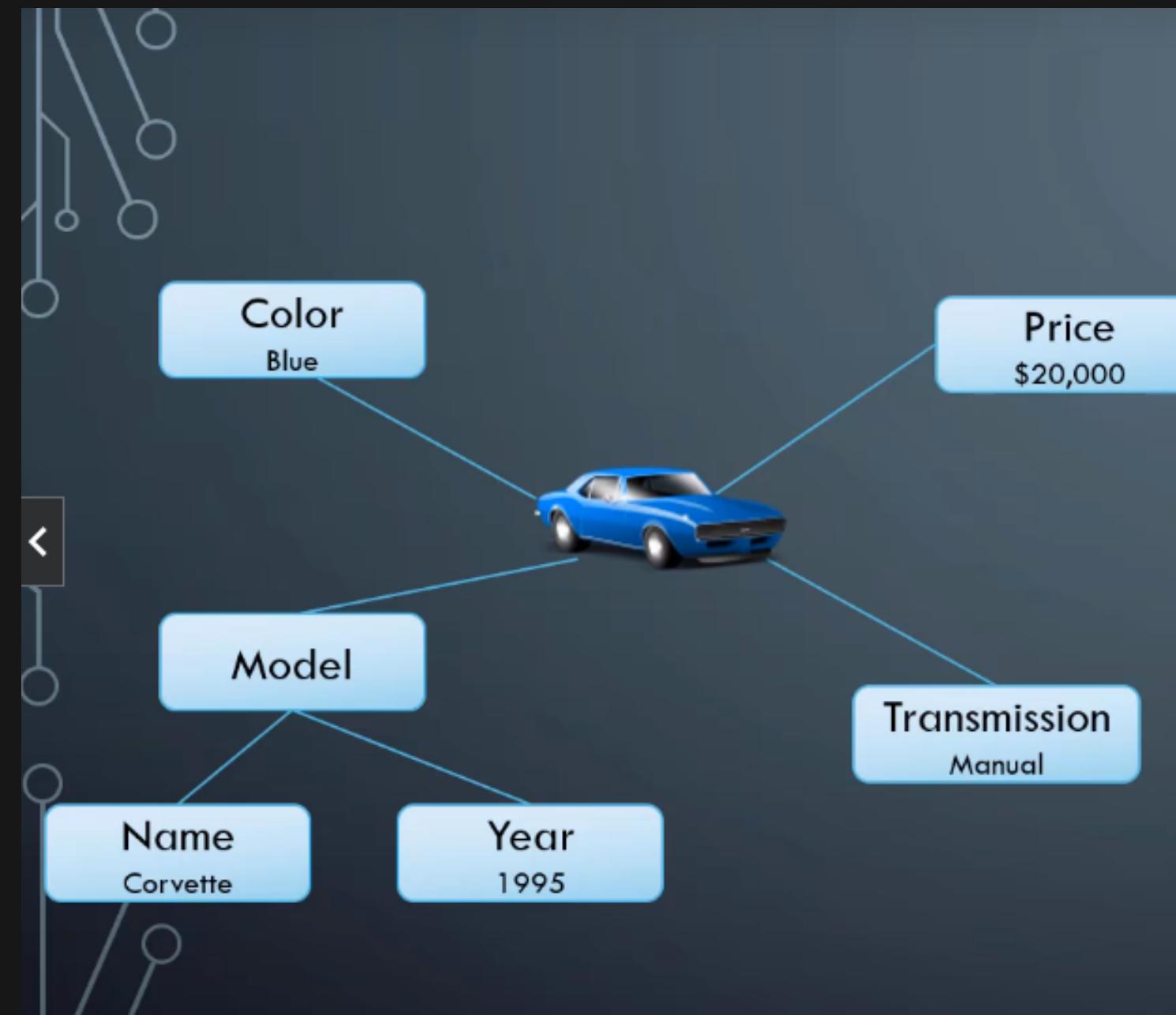
A key-value pair is like a label attached to something, along with its description. Imagine a set of keys and each key has a tag indicating what it unlocks.

A dictionary (or associative array) is a data structure that stores key-value pairs. It allows you to map keys to their corresponding values, making it easy to retrieve data by its unique key.

XML	JSON	YAML
<pre><Servers> <Server> <name>Server1</name> <owner>John</owner> <created>12232012</created> <status>active</status> </Server> </Servers></pre>	<pre>{ Servers: [{ name: Server1, owner: John, created: 12232012, status: active, }] }</pre>	<pre>Servers: - name: Server1 owner: John created: 12232012 status: active</pre>



	<p>Color: Grey Model: Name: Corvette Model: 1995 Transmission: Manual Price: \$22,000</p>	<p>Transmission : Manual Price: \$20,000</p> <ul style="list-style-type: none">- Color: Grey Model: Name: Corvette Model: 1995 Transmission: Manual Price: \$22,000
	<p>Color: Red Model: Name: Corvette Model: 1995 Transmission: Automatic Price: \$20,000</p>	<ul style="list-style-type: none">- Color: Red Model: Name: Corvette Model: 1995 Transmission: Automatic Price: \$20,000
	<p>Color: Green Model: Name: Corvette Model: 1995 Transmission: Manual Price: \$23,000</p>	<ul style="list-style-type: none">- Color: Green Model: Name: Corvette Model: 1995 Transmission : Automatic Price: \$20,000
	<p>Color: Blue Model: Name: Corvette Model: 1995 Transmission: Manual Price: \$20,000</p>	<ul style="list-style-type: none">- Color: Blue- Color: Green Model: Name: Corvette Model: 1995 Transmission : Manual Price: \$23,000



Dictionary In Dictionary

Color: Blue
Model:
 Name: Corvette
 Year: 1995
Transmission: Manual
Price: \$20,000

List Of Dictionaries

- Color: Blue
Model:
 Name: Corvette
 Model: 1995
 Transmission : Manual
 Price: \$20,000
- Color: Grey
Model:
 Name: Corvette
 Model: 1995
 Transmission: Manual
 Price: \$22,000
- Color: Red
Model:
 Name: Corvette
 Model: 1995
 Transmission: Automatic
 Price: \$20,000
- Color: Green
Model:
 Name: Corvette
 Model: 1995
 Transmission: Manual
 Price: \$23,000
- Color: Blue
Model:
 Name: Corvette
 Model: 1995
 Transmission: Manual
 Price: \$20,000



Color: Blue
Model:
 Name: Corvette
 Model: 1995
 Transmission: Manual
 Price: \$20,000



Color: Grey
Model:
 Name: Corvette
 Model: 1995
 Transmission: Manual
 Price: \$22,000



Color: Red
Model:
 Name: Corvette
 Model: 1995
 Transmission: Automatic
 Price: \$20,000



Color: Green
Model:
 Name: Corvette
 Model: 1995
 Transmission: Manual
 Price: \$23,000



Color: Blue
Model:
 Name: Corvette
 Model: 1995
 Transmission: Manual
 Price: \$20,000

EXAMPLE

name: John Doe

age: 30

hobbies:

- Reading
- Running

languages:

- English
- French

contact:

email: john@example.com

phone: +1 123-456-7890

YAML

Defining configurations for containerization. The primary use of YAML in Docker is through Docker Compose files, which allow you to specify the services, containers, and configurations for multi-container applications.

Kubernetes uses YAML extensively for defining and managing resources such as pods, services, deployments, and more. Kubernetes YAML files describe the desired state of your application and infrastructure.

YAML

In cloud services like AWS, Azure, and Google Cloud, YAML is used for defining infrastructure as code (IaC) using tools like AWS CloudFormation (YAML/JSON), Azure Resource Manager (ARM templates), and Google Cloud Deployment Manager.

AML is a key component in modern DevOps and cloud-native practices. It's used to define configurations, resources, and infrastructure across various technologies, making it easier to automate, manage, and version control complex systems and applications.

Environmental variable

Often referred to as "env" is a dynamic value that can affect the behavior of processes running in an operating system or within a software application.

In the context of Docker containers and Kubernetes pods, environmental variables can be used to configure containerized applications, specify database connection strings, define API keys, and much more.

Environmental variable

Storage of Information: Environmental variables store information in the form of key-value pairs. Each variable has a name (key) and a corresponding value.

Security: Some sensitive information, like passwords or secret keys, should not be stored in environmental variables because they can be accessed by other processes.

CMD & ENTRYPOINT

In Docker, both the CMD and ENTRYPOINT instructions are used to specify what command should be run when a container is started. However, they serve slightly different purposes and have different behaviors

CMD

- The **CMD instruction in a Dockerfile specifies the default command and/or arguments to be executed when a container is started if no other command is provided at runtime.**
- You can have only one **CMD instruction in a Dockerfile**.
- If a **command is provided when starting the container (e.g., docker run my-image some-command)**, it will override the **CMD instruction in the Dockerfile**.
- If no **CMD is specified in the Dockerfile, Docker will use the default system command, typically /bin/sh -c, as the command**.

EXAMPLES

```
FROM ubuntu
```

```
CMD ["echo", "Hello, World!"]
```

```
FROM ubuntu
```

```
ENTRYPOINT ["echo", "Hello,"]
```

```
CMD ["World!"]
```

In summary, the ENTRYPOINT instruction sets the main command for the container, while the CMD instruction provides default arguments that can be overridden at runtime.

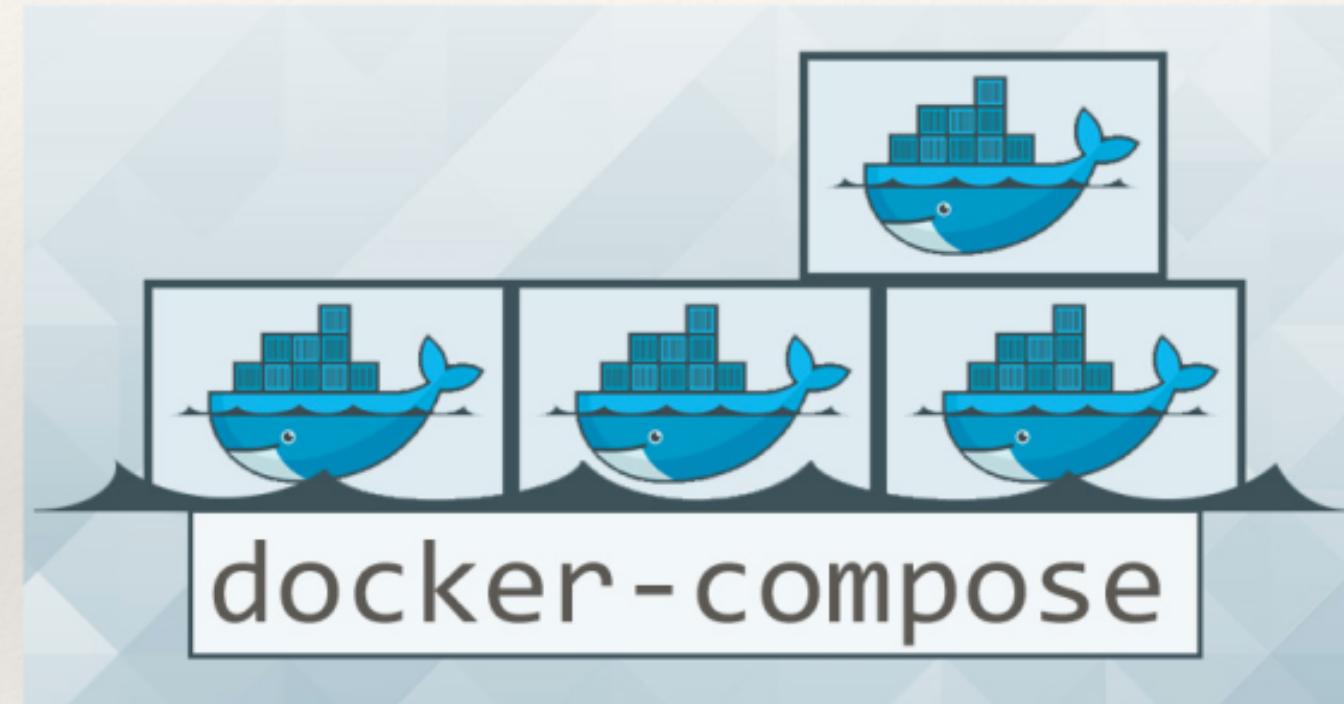
Docker Compose

An open-source tool that allows you to define and run multi-container Docker applications. It simplifies the process of defining, managing, and running complex applications that consist of multiple interconnected containers.

Docker Compose uses a YAML (Yet Another Markup Language) file to define the services, networks, and volumes required for your application.

Service Definition: In the docker-compose.yml file, you can specify each service your application requires.

Creating multiple Docker containers



- Step 1.** Create a docker-compose.yml file (or docker-compose.yaml file)
- Step 2.** Execute “docker-compose up -d”
- Step 3.** Execute “docker-compose logs” from another shell (but from same dir)
- Step 4.** Execute “docker-compose down”

Command	Description
docker-compose up	(Re)build services
docker-compose kill	Kill the containers
docker-compose logs	Show the logs of the containers
docker-compose down	Stop and remove images, containers, volumes and networks
docker-compose rm	Remove stopped containers

```
Add version: '3'  
services:  
  web-app:  
    image: my-web-app:1.0  
    ports:  
      - "8080:80"  
    depends_on:  
      - database  
database:  
  image: postgres:13  
environment:  
  POSTGRES_DB: mydb  
  POSTGRES_USER: myuser  
  POSTGRES_PASSWORD: mypassword  
subheading
```

Lets Dive into a EXAMPLE

Running Mongo DB and Mongo Express with
the Help of a DOCKER-COMPOSE File

docker run command

```
docker run -d \
--name mongodb \
-p 27017:27017 \
-e MONGO_INITDB_ROOT_USERNAME=admin \
-e MONGO_INITDB_ROOT_PASSWORD=password \
--net mongo-network \
mongo
```

mongo-docker-compose.yaml

```
version: '3'
services:
  mongodb:
    image: mongo
    ports:
      - 27017:27017
    environment:
      - MONGO_INITDB_ROOT_USERNAME=admin
      - MONGO_INITDB_ROOT_PASSWORD=password
```