

The background features three vertical stripes on the left side: a light red stripe, a light blue stripe, and a light beige stripe. The right side of the background is white with a pattern of small, light red dots arranged in a grid that tapers off towards the right edge.

# **CLOUD TECHNOLOGIES**

**Presented By :**

**Cameron, Jahnavi, & Montana**

# OVERVIEW

- SaaS IaaS PaaS
- Cloud Computing
- Public cloud
- Bare metal
- Virtual Machine
- Advantages
- Disadvantages
- Cost of cloud
- Containers
- Docker
- Kubernetes
- References

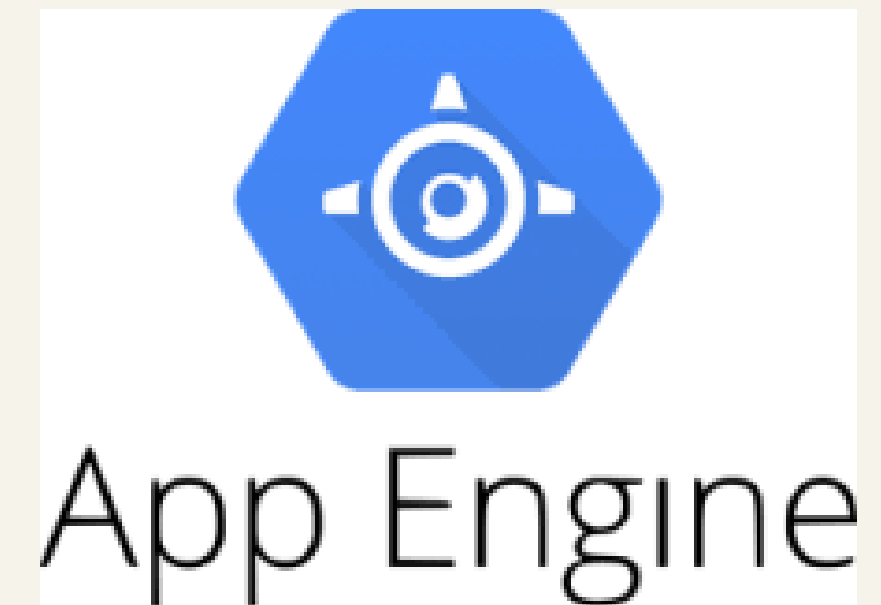
# Software as a Service (SaaS)

- A cloud-based model that allows users to access software applications through the internet instead of on local machines
- frees users from the need to manage software and hardware
- Service provider manages hardware and software
- Service provider will ensure the availability and security of the app and your data
- Either pay-as-you-go or subscription



# Platform as a Service (PaaS)

- 1** Provides a framework that developers can build upon and use to create customized applications
- 2** Third-party provider can manage the servers, storage, and networking to let the developers focus on maintaining the applications
- 3** PaaS can streamline workflows when multiple developers are working on the same development project



# Infrastructure as a Service (IaaS)

- service that offers essential computing, storage, and networking resources
- Allows businesses to purchase resources on demand and as needed instead of having to buy hardware outright
- variable and consumption based cost
- highly scalable service options
- Companies that start experiencing rapid growth tend to choose IaaS.



# SaaS

- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

# PaaS

- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

# IaaS

- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking



# Cloud Computing

**On demand access of computing resources**

- **Physical or Virtual servers**
- **data storage**
- **networking capabilities**
- **application development tools**
- **software**
- **AI-Powered analytic tools**



# Public Cloud

**computing service offered by  
third-party providers**

**Offered over public internet,  
therefore available to all**

**may be free (Gmail) or  
sold on demand (Netflix)**

# Bare Metal


- **Single-tenant server that provides users with direct access to a physical server's hardware, rather than virtualized computing resources**
- **Bare metal offers better security compared to virtual servers as the physical hardware is isolated from other users, reducing the risk of breaches or data leakage**
- **Allows the user to access 100% of the processing power, memory and storage of the physical hardware resources**

# VIRTUAL MACHINES

**A software-based computer that acts like a physical computer, but uses virtual resources instead of physical components**

## Benefits

- **Isolation** - VMs are partitioned from the rest of the system so the software can not interfere with the host computer's primary operating system
- **Portability** - VMs can be moved between machines almost instantly
- **Multiple OS** - VMs allow you to run different operating systems on the same host computer.

The background features a vertical bar on the left with a color gradient from light pink to light blue. The rest of the background is white with a grid of small, light pink dots in the top right and bottom right corners.

# **ADVANTAGES OF CLOUD**

# COST SAVINGS

- 1** Better cost savings compared to maintaining an on premises data center.
- 2** Cloud providers have pay as you go policies for resources.
- 3** reduced maintenance and employee costs since everything is handled by the provider

# COLLABORATION

- **Cloud based tools facilitate real time collaboration between teams, and people across the business wherever they are located**

- **Team members can view and share information easily.**

# AGILITY

**Cloud accelerates time to market for new products, making it ideal for startups and companies with rapid deployment cycles. It allows companies to quickly provision computing resources.**

## Agile development

**Provide tools and services that support agile software development practices.**

## Agile deployment

**teams can iterate on software quickly, deploy updates frequently, and respond rapidly to feedback**



# DISASTER RECOVERY

- **Most cloud providers offer robust data backup and recovery solutions.**
- **Ensures that data is protected and recoverable in case of hardware failures.**
- **This support minimizes data loss and speeds up recovery.**

# OTHER ADVANTAGES

## Scalability

This refers to the ability to increase or decrease IT resources quickly based on the demand.

## Accessibility

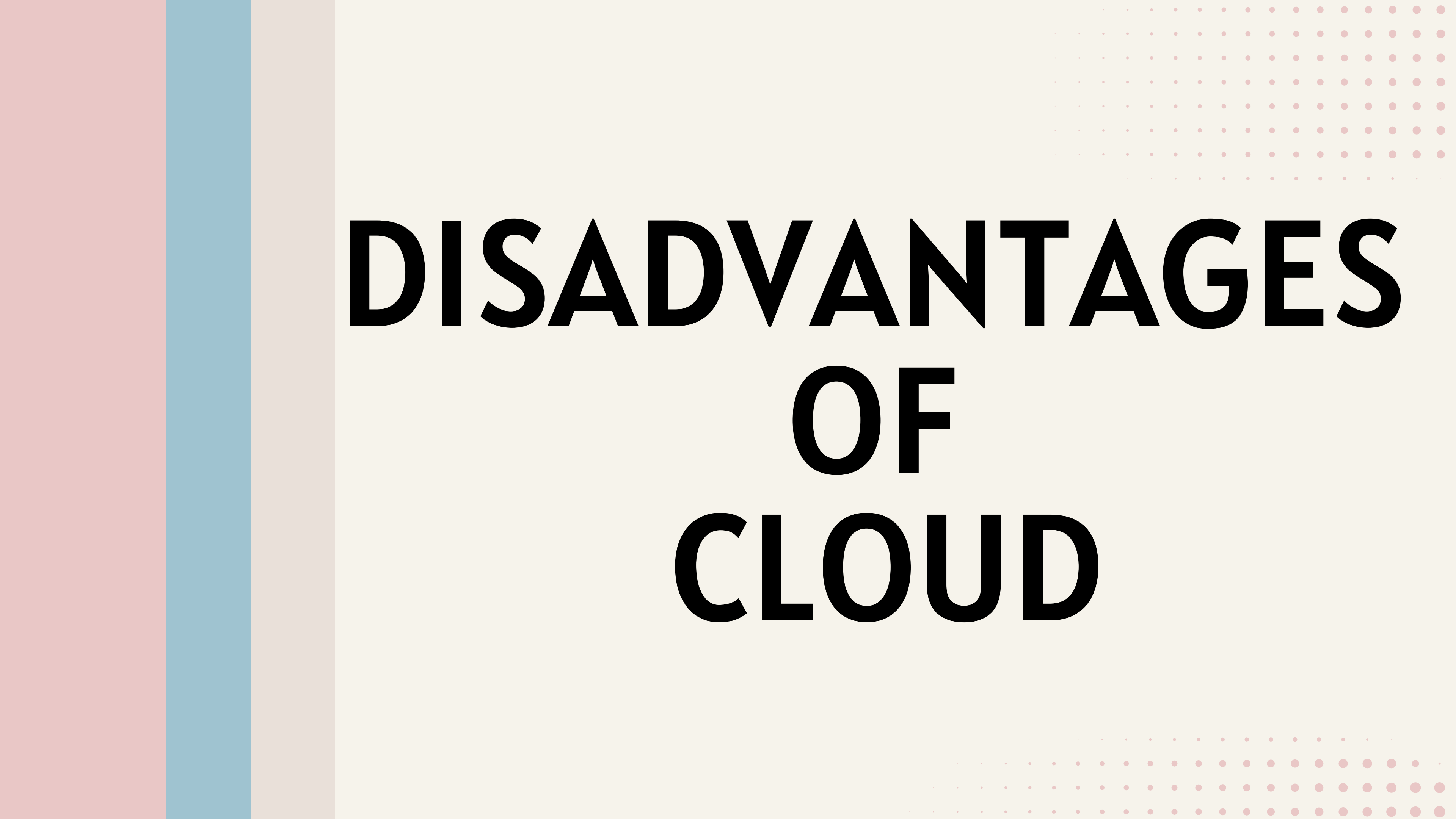
Cloud services can be accessed from anywhere with an internet connection, enabling remote work.

## Quality

Cloud providers offer high levels of service availability, performance and reliability. they ensure regular updates and maintenance.

## Advanced Security

Cloud providers offer robust security features, including data encryption access control, compliance with industry standards and regulations.

The background features three vertical stripes on the left side: a light red stripe, a light blue stripe, and a light beige stripe. The right side of the background is white with a pattern of small, light red dots arranged in a grid that tapers towards the right edge.

# **DISADVANTAGES OF CLOUD**

# DOWNTIME

- **When cloud services go offline due to power outage, human error, hardware failure, cyber attack etc.**
- **The service provider can plan downtime or it can happen without notice. This causes financial loss**

## Solution

**Consider providers downtime history and SLA guarantees.  
Have a backup system with minimal downtime.**

# LIMITED CONTROL & FLEXIBILITY

## ● Why?

End-user license agreement and management policies impose limits on what the customer can do with their deployments.

## ● Solution

Opt for a provider which will provide more control and customization.  
Enlist help of a cloud partner to run, implement and support CS

# VENDOR LOCK IN

- **Hard time switching between cloud service providers as there are differences in the platform.**
- **changing platforms can lead to configuration complexity which can lead to gaps.**

## Solution 1

Use a backup and recovery system for the data to ensure that there is a copy that can be retrieved

## Solution 2

employ multi cloud vendor strategies to avoid vendor lock-ins. Although this might add deployment complexity

The background features a vertical bar on the left with a color gradient from light pink to light blue. The rest of the background is light beige with a pattern of small, light pink dots in the top right and bottom right corners.

# CLOUD BILLING MODELS

Cloud Technologies

# PAY AS YOU GO

## flexible Scaling

- PAYG allows organizations to scale resources as needed.
- Makes it suitable for variable workload

## Budgeting Challenges

- User can select tools and pay for them as they go.
- Lack of predictability in costs more budgeting complex.



## Cost Optimization

- Users are only required to pay for the services they use.
- allows user to avoid unnecessary costs.



# SUBSCRIPTION BASED

- **Predictable Costs**  
Fixed monthly or yearly that makes budgeting and planning easier.
- **Discounts for Commitments**  
Longer term commitments come with discount.
- **Inflexibility**  
May lead to unused resources if needs are overestimated.

# RESERVED INSTANCES

1

**COST SAVINGS:** reserved instances offer substantial discounts compared to on demand pricing.

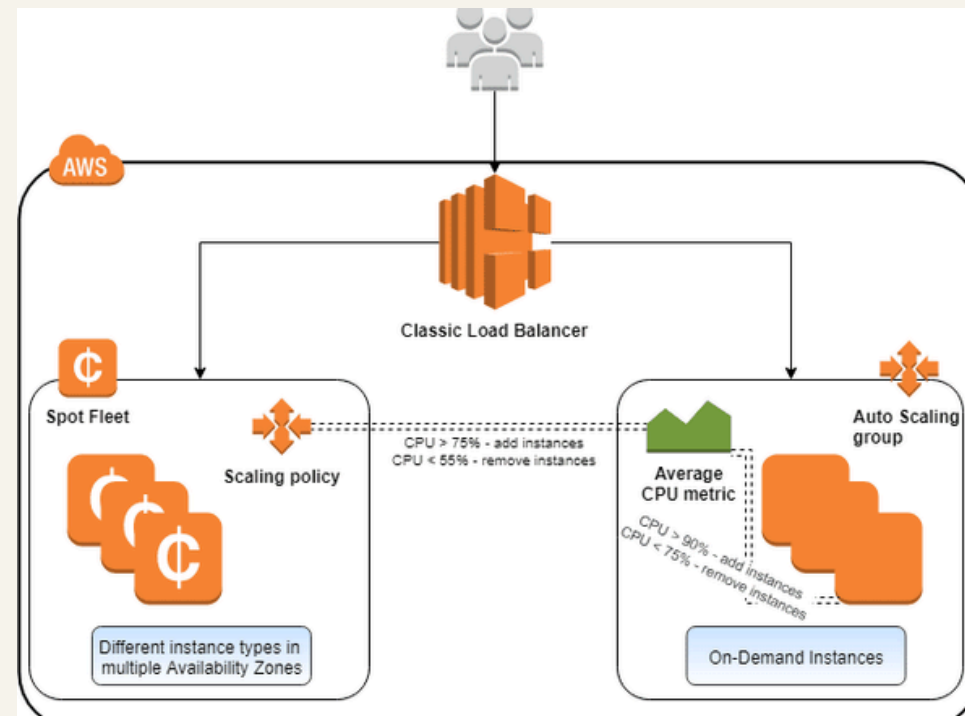
2

**CAPACITY ASSURANCE:** Provides guaranteed access to resources when needed

3

**UPFRONT COMMITMENT:** Has upfront payment and fixed term length.

# SPOT INSTANCES



## Intermittent Availability

These are subject to interruptions when spot price exceeds the bid price or when capacity is needed by other users.

## Cost Optimization

Offer significant cost savings, often at a fraction of the cost of on-demand instances.

## Suitable Workloads

Best suited for non-critical, fault tolerant workloads which can handle interruptions.

# HYBRID MODELS

1

**TAILORED APPROACH:** Combine different billing methods to best suit an organizations requirements.

2

**FLEXIBILITY AND OPTIMIZATION:** By combining models, organizations can balance cost, performance and scalability.

3

**INCREASES COMPLEXITY:** Implementing and managing different hybrid models can be challenging.

# CONTAINERS

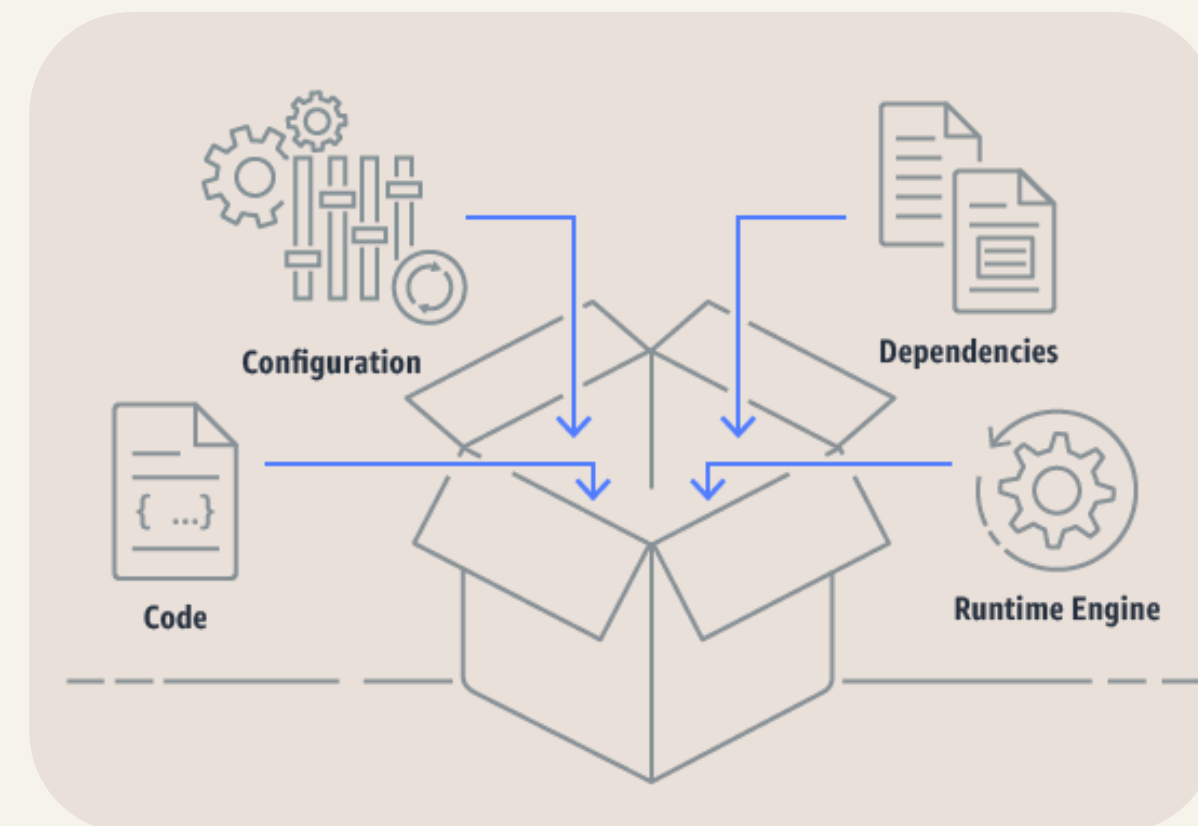
## What is a Container?

Containers are a software package that contain all the necessary elements like libraries to be run in any environment.

Containers virtualize the operating system and run anywhere from a private data center to the public cloud or even a developer's personal laptop.

## Example

Gmail is a Google application which is containerized and runs on the Google cloud server.



# USES OF CONTAINERS

1

**Continuous Integration and Continuous deployment :** Helps create applications that can be deployed, scaled and integrated without interruption.

2

**Batch Processes :** Sharing of libraries and OS make it great for deploying and executing batch processes.

3

**Microservices :** Uses several containers to build and deploy a single application which creates a cluster of containers.

# ADVANTAGES

- **Portability Across environments**
- **Resource efficiency**
- **Scalability**
- **Rapid deployment**
- **Streamlined DevOps**
- **Isolation**
- **Version control and rollbacks**
- **Ecosystem support & tooling**
- **Security**

# DOCKER

**Docker is an open platform for developing, shipping, and running applications.**

## **Useful for..**

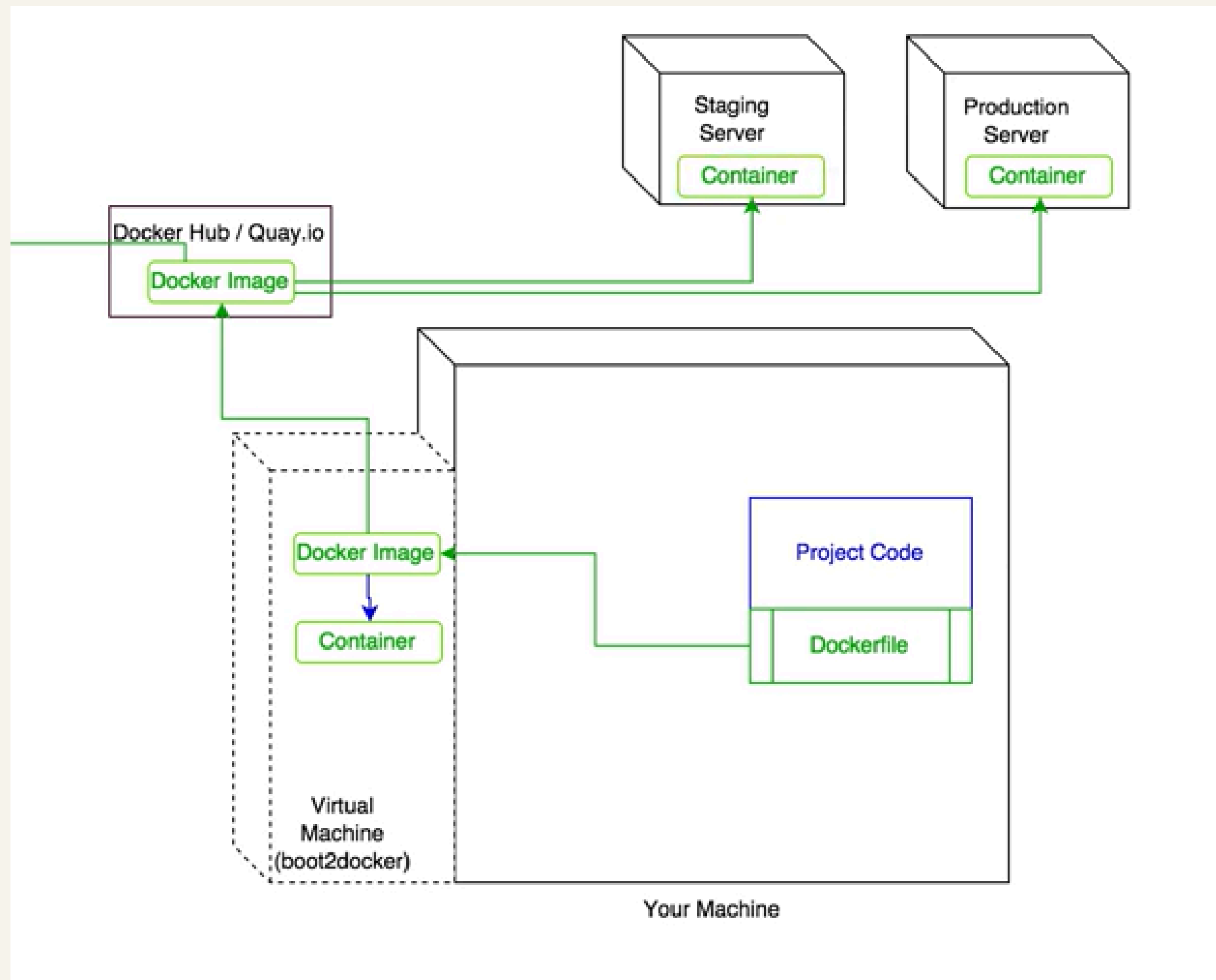
- **Fast, consistent delivery of your applications**
- **Responsive deployment and scaling**
- **Running more workloads on the same hardware**

## **What can it do?**

**provides the ability to package and run an application in a loosely isolated environment called a container. The isolation and security lets you run many containers simultaneously on a given host.**



# HOW IT WORKS



1. Dockerfile builds a docker image, which contains project code, installations needed to run the code, and any installments of other programs needed
2. Docker image runs as container
3. Push docker image up to Docker Hub/Quay.io
4. Other machines can run the docker image with complete environment, no installation needed

# DOCKERFILE

```
FROM python:3.12
WORKDIR /usr/local/app

# Install the application dependencies
COPY requirements.txt ./
RUN pip install --no-cache-dir -r requirements.txt

# Copy in the source code
COPY src ./src
EXPOSE 5000

# Setup an app user so the container doesn't run as the root user
RUN useradd app
USER app

CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8080"]
```

**A Dockerfile is a text-based document that's used to create a container image. It provides instructions to the image builder on the commands to run, files to copy, startup command, and more.**

# STARTING A DOCKER CONTAINER

**Build docker image:**

```
docker build -t my-username/my-image .
```

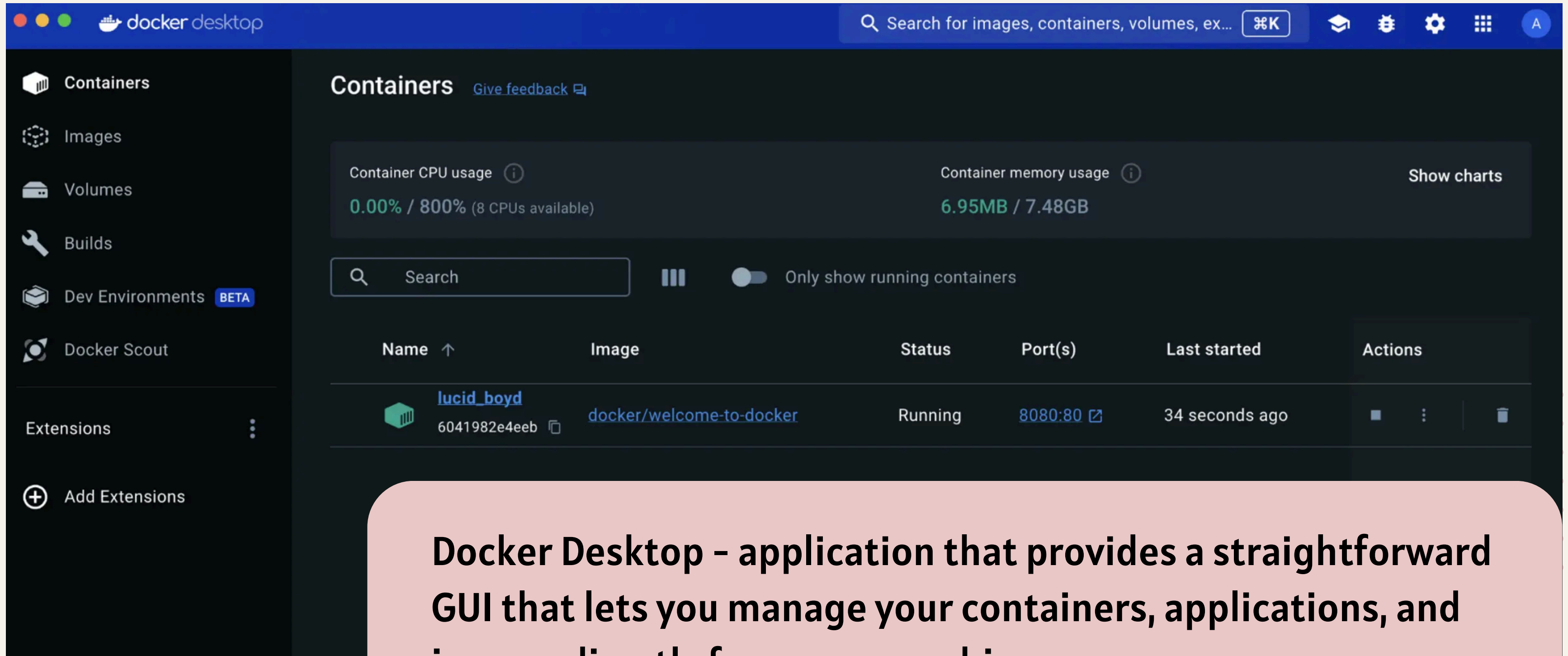
**Push image to registry (optional):**

```
docker push my-username/my-image
```

**Start a new container:**

```
docker run -d -p 8080:80 -t my-username/my-image .
```

# ACCESSING A DOCKER CONTAINER



The screenshot displays the Docker Desktop application window. The top navigation bar includes a search field for images, containers, and volumes, along with icons for help, Docker Hub, settings, and a user profile. The left sidebar contains navigation options: Containers, Images, Volumes, Builds, Dev Environments (marked BETA), Docker Scout, and Extensions. The main content area is titled 'Containers' and shows system metrics for CPU usage (0.00% / 800%) and memory usage (6.95MB / 7.48GB). Below these metrics is a search bar and a toggle for 'Only show running containers'. A table lists the active containers:

Name	Image	Status	Port(s)	Last started	Actions
lucid_boyd 6041982e4eeb	docker/welcome-to-docker	Running	8080:80	34 seconds ago	[Stop] [More] [Delete]

**Docker Desktop - application that provides a straightforward GUI that lets you manage your containers, applications, and images directly from your machine.**

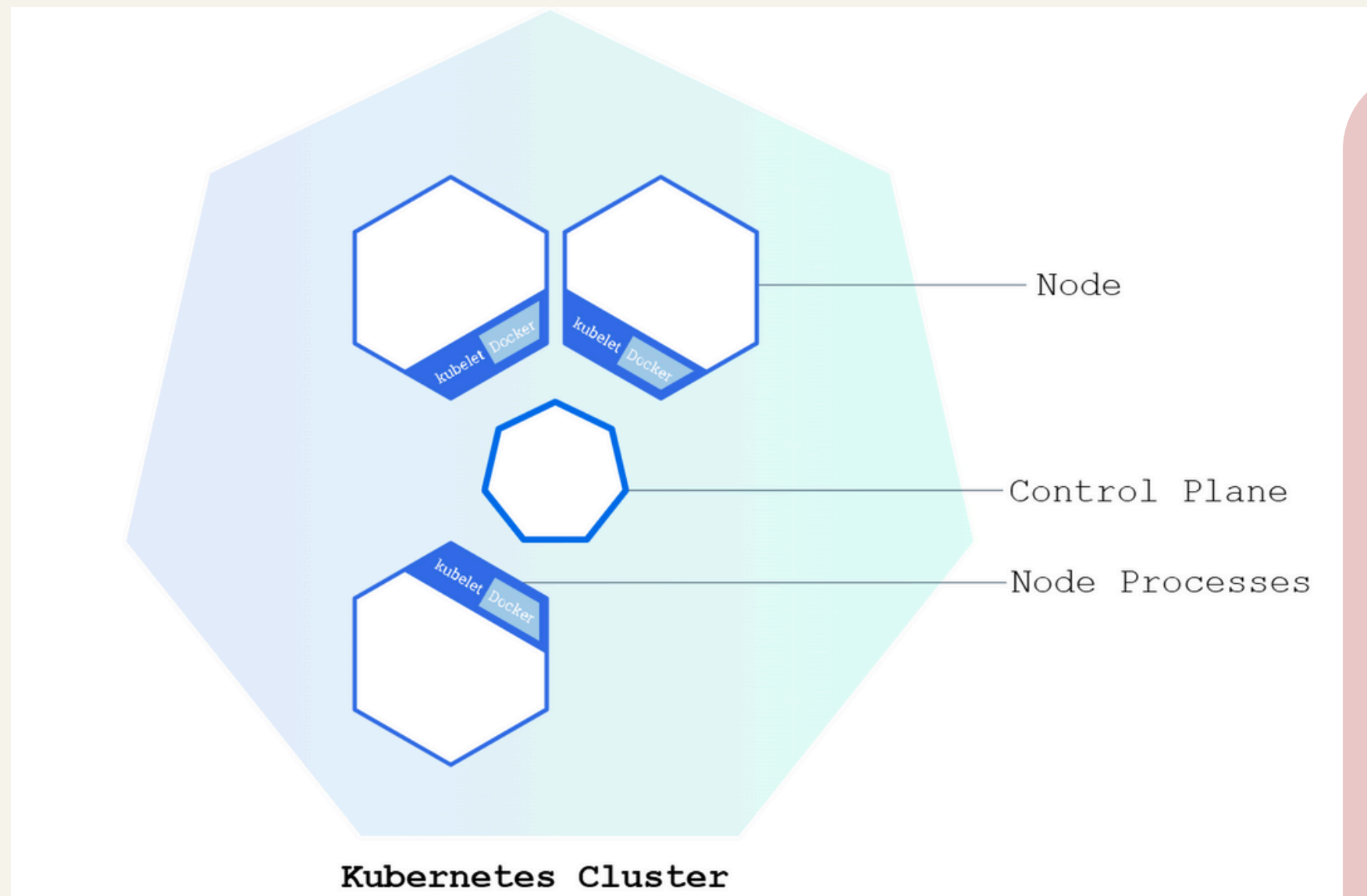
# KUBERNETES

Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services

## What can it do?

- run distributed systems resiliently - automates scaling and ensures failover for the application
- self-healing - restarts containers that fail, replaces containers, kills containers that aren't responding
- automated rollouts and rollbacks - adjusts container states to match the desired configuration defined

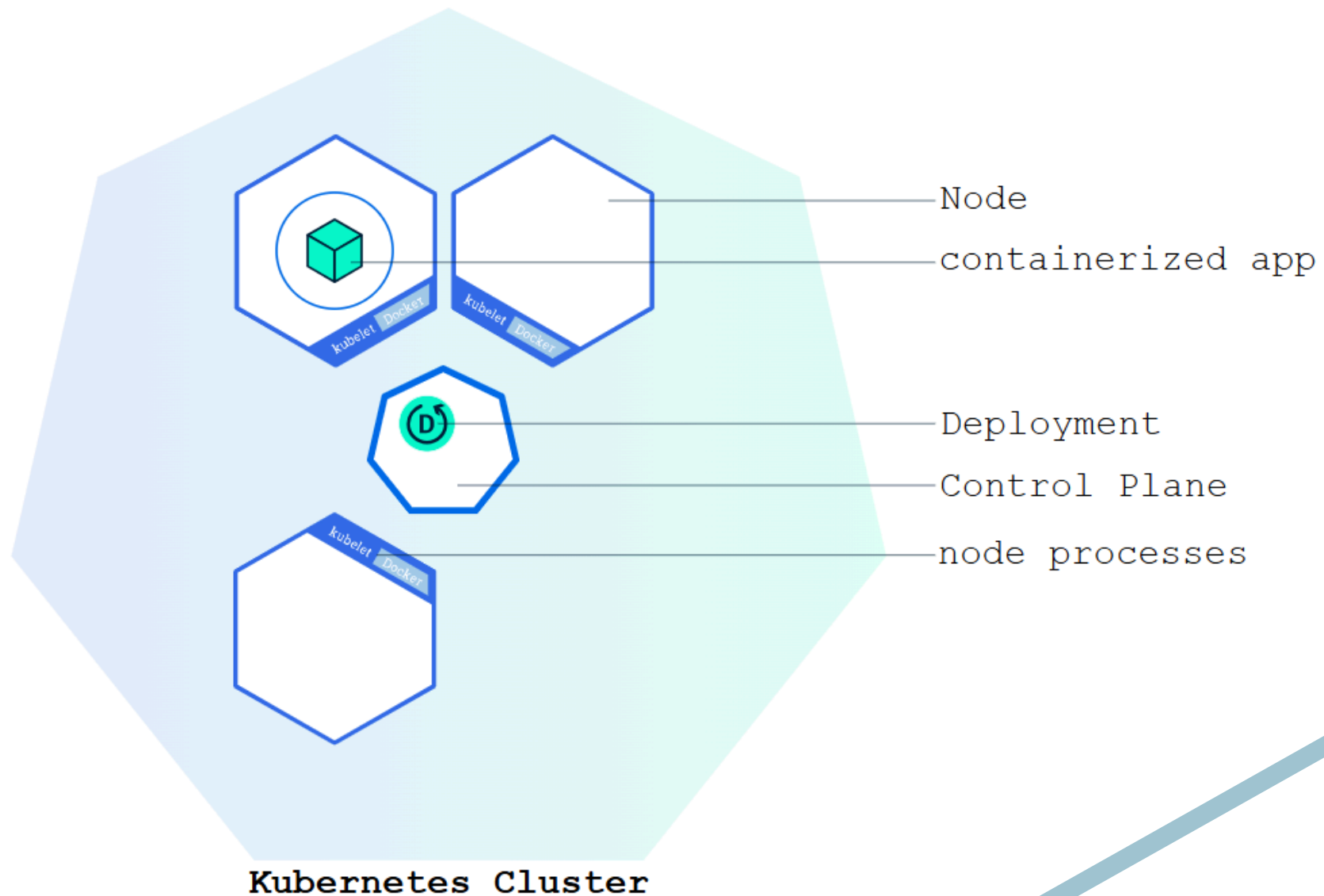
# HOW TO DEPLOY



**Create a Kubernetes cluster - runs containerized applications in distributed manner**

- **Control Plane: coordinates the cluster**
- **Nodes: a VM or physical computer that serves as workers that run containerized applications**
  - **Kubelet - agent for managing node and communicating to control plane using Kubernetes API**

# HOW TO DEPLOY



**Kubectl** - Kubernetes command line interface that allows you to create and manage a deployment.  
**Pod** - Kubernetes abstraction that represents a group of one or more application containers and their shared resources

**Creates deployment and names it**

**Specifies which container image to use**

```
kubectl create deployment kubernetes-bootcamp --image=gcr.io/google-samples/kubernetes-bootcamp:v1
```

# HOW TO CONFIGURE

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
        - name: my-app
          image: my-app:v1
          ports:
            - containerPort: 80
```

Kubernetes deployment specifications are defined using a YAML (shown) or JSON file and customize the behavior of an application

- **apiVersion** - the Kubernetes API version
- **kind** - the resource type
- **metadata** - information about the deployment (names and labels)
- **spec** - desired state of the deployment
  - number of replicas (pods) to be running
  - container image

To apply Kubernetes deployment:

```
kubectl apply -f deployment.yaml
```



# HOW TO MANAGE

Kubectl commands can be utilized to manage both Kubernetes clusters and deployments. Here are some common commands used:

## Managing Deployments:

List all deployments in a namespace

```
kubectl get deployments
```

Update deployment by changing the container image

```
kubectl set image deployment/<deployment-name> <container-name>=<new-image>
```

Scale deployment to adjust number of replicas (pods)

```
kubectl scale deployment/<deployment-name> --replicas=<number>
```

# HOW TO MANAGE

Kubectl commands can be utilized to manage both Kubernetes clusters and deployments. Here are some common commands used:

## Managing Clusters:

**View cluster information**

```
kubectl cluster-info
```

**Show metrics for all nodes**

```
kubectl top node
```

**Get all running pods in the namespace**

```
kubectl get pods --field-selector=status.phase=Running
```

# REFERENCES

- <https://docs.docker.com/get-started/docker-overview/>
- <https://docs.docker.com/get-started/docker-concepts/building-images/writing-a-dockerfile/>
- <https://www.youtube.com/watch?v=pGYAg7TMmp0>
- <https://docs.docker.com/get-started/docker-concepts/building-images/build-tag-and-publish-an-image/#next-steps>
- <https://docs.docker.com/get-started/docker-concepts/the-basics/what-is-a-registry/>
- <https://docs.docker.com/desktop/#:~:text=Docke%20Desktop%20is%20a%20one,run%20containerized%20applications%20and%20microservices.>
- <https://docs.docker.com/get-started/docker-concepts/running-containers/publishing-ports/>
- <https://www.docker.com/resources/what-container/>
- <https://cloud.google.com/learn/what-are-containers>
- <https://kubernetes.io/docs/concepts/overview/>
- <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>
- <https://kubernetes.io/docs/concepts/configuration/overview/>
- <https://kubernetes.io/docs/tutorials/kubernetes-basics/deploy-app/deploy-intro/>
- <https://kubernetes.io/docs/concepts/workloads/pods/>
- <https://kubernetes.io/docs/tutorials/kubernetes-basics/explore/explore-intro/>
- <https://kubernetes.io/docs/tutorials/kubernetes-basics/create-cluster/cluster-intro/>
- <https://kubernetes.io/docs/tutorials/kubernetes-basics/explore/explore-intro/>
- <https://labex.io/tutorials/kubernetes-how-to-configure-kubernetes-deployment-parameters-414651>
- <https://kubernetes.io/docs/concepts/overview/kubernetes-api/>
- <https://kubernetes.io/docs/concepts/overview/#why-you-need-kubernetes-and-what-can-it-do>
- <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>
- <https://kubernetes.io/docs/reference/kubectl/quick-reference/>
- <https://www.geeksforgeeks.org/advantages-of-cloud-computing/>
- <https://cloud.google.com/learn/advantages-of-cloud-computing>
- <https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.html>
- <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-saas>
- <https://blog.equinix.com/blog/2023/08/16/what-is-bare-metal-is-it-a-server-or-a-cloud/#:~:text=The%20term%20bare%20metal%20refers,full%20control%20of%20the%20server.>
- <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-virtual-machine>
- <https://www.cloudwards.net/disadvantages-of-cloud-computing/>
- <https://blog.centrotechnologies.com/overcoming-disadvantages-of-cloud-computing>
- <https://cloudzy.com/blog/cloud-pricing-model/>
- <https://www.cloudflare.com/learning/cloud/what-is-hybrid-cloud/>

The background features three vertical stripes on the left side: a wide light pink stripe, a narrower teal stripe, and a narrow light beige stripe. The right side of the background is white with a grid of small, light pink dots that fades out towards the right edge.

**THANK YOU**

**Presented By : Cameron, Jahnavi, & Montana**