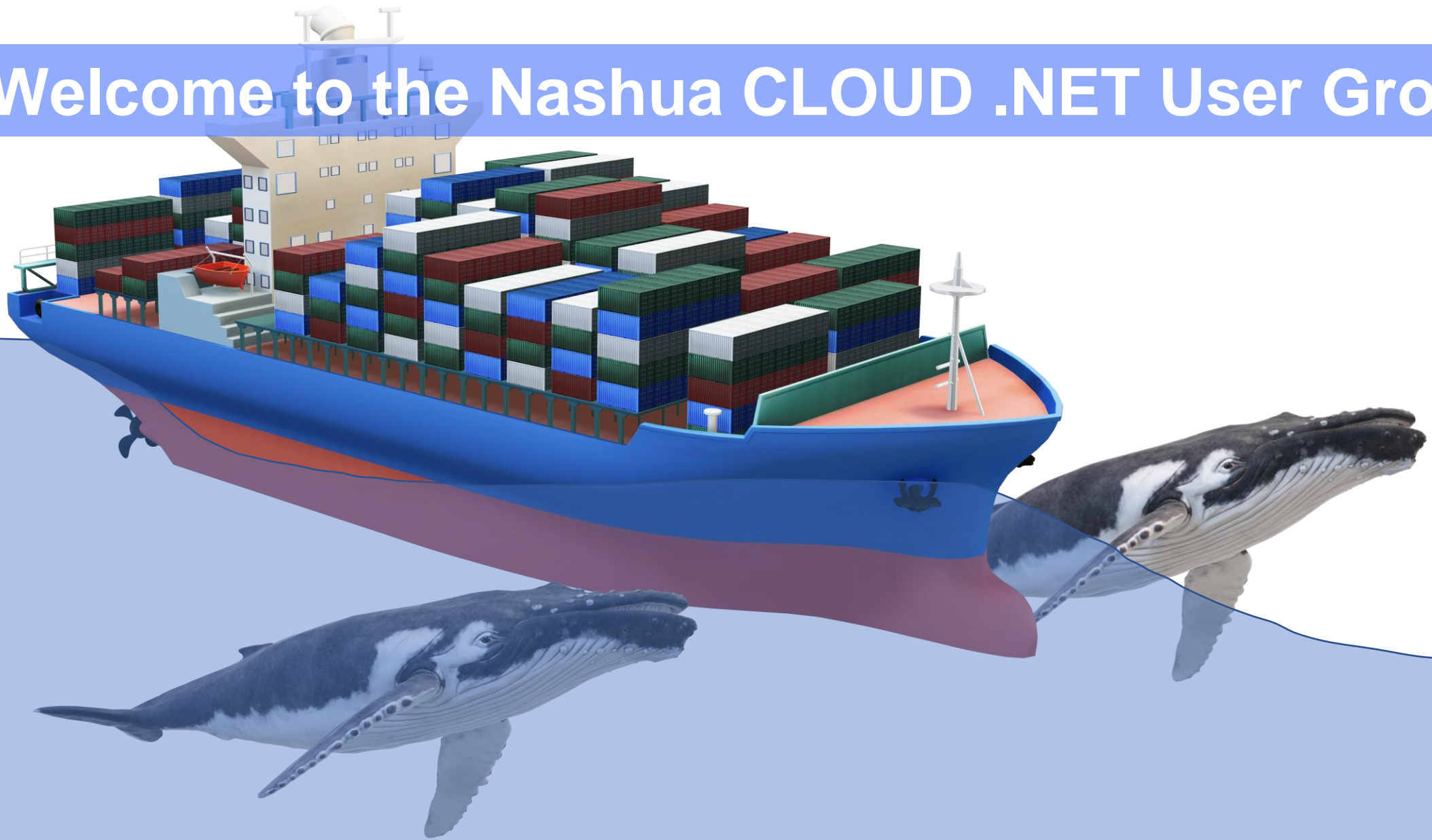


# Welcome to the Nashua CLOUD .NET User Group

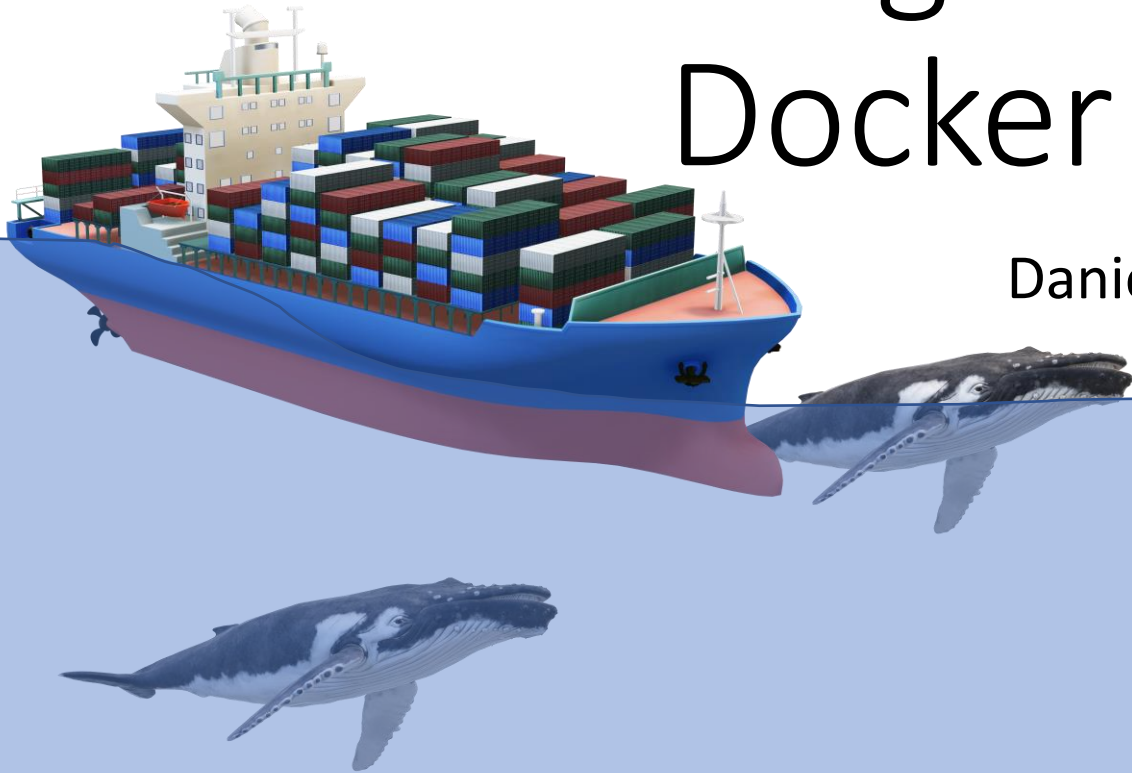


Akumina, Nashua, NH (Online)

Tuesday, December 13<sup>th</sup>, 2023

# Running Kubernetes on Docker Desktop

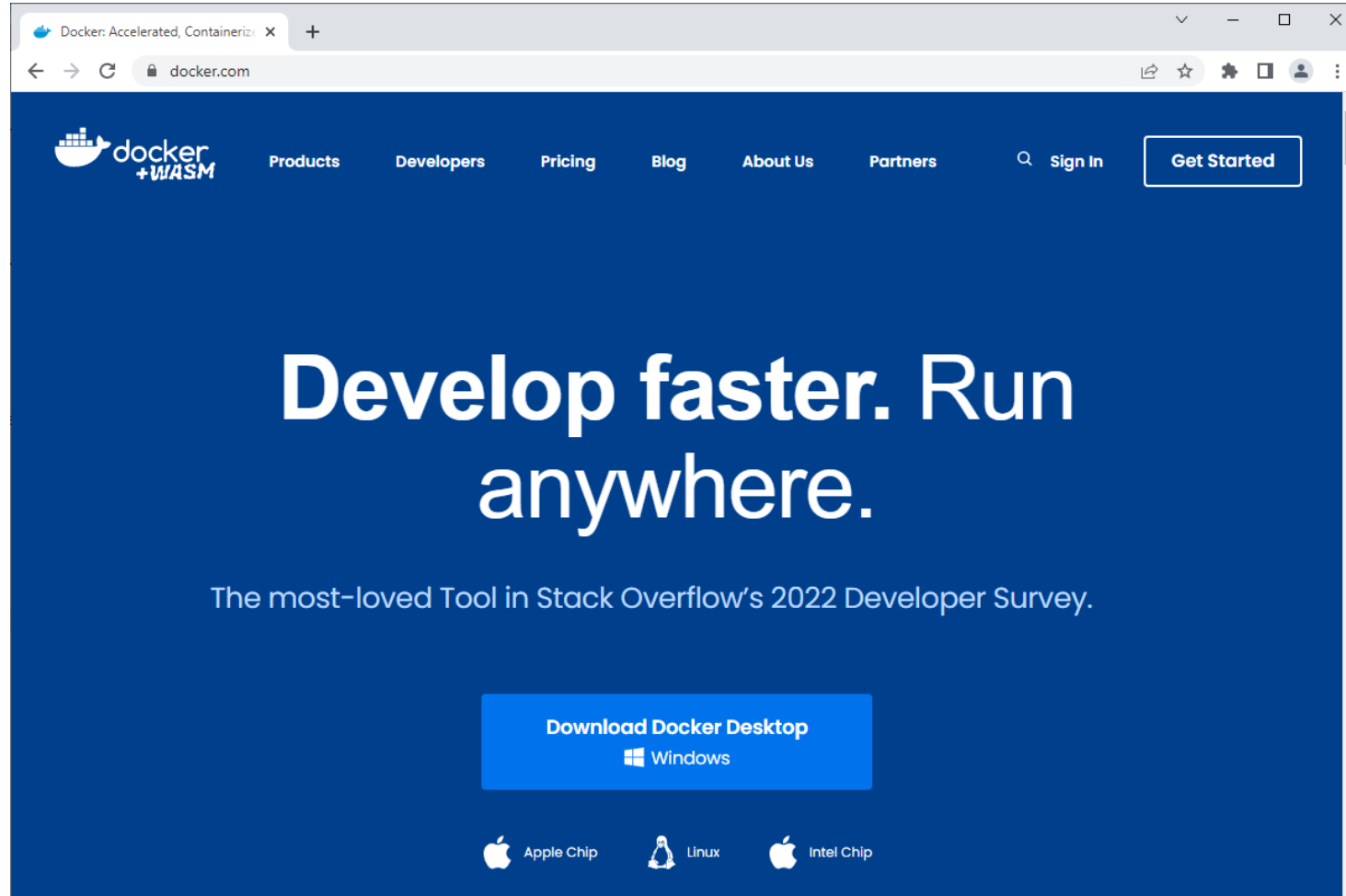
Daniel Colón



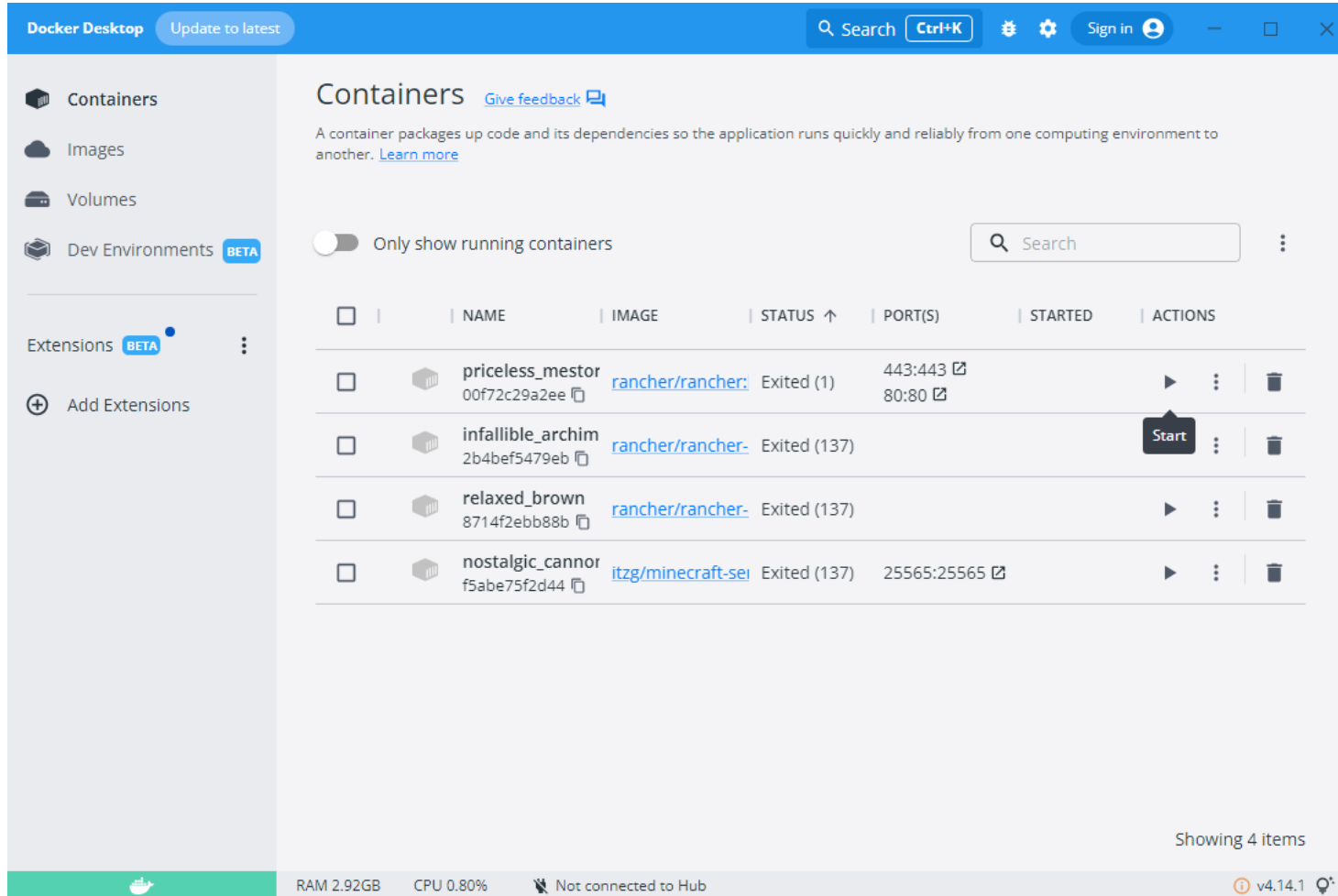
# Agenda

- Docker Desktop Kubernetes Installation/Configuration
- Visual Studio Kubernetes and YAML Extensions
- Brief overview of Docker Desktop
- Brief overview of Kubernetes
- Demos

# Docker Desktop Install



# Docker Desktop



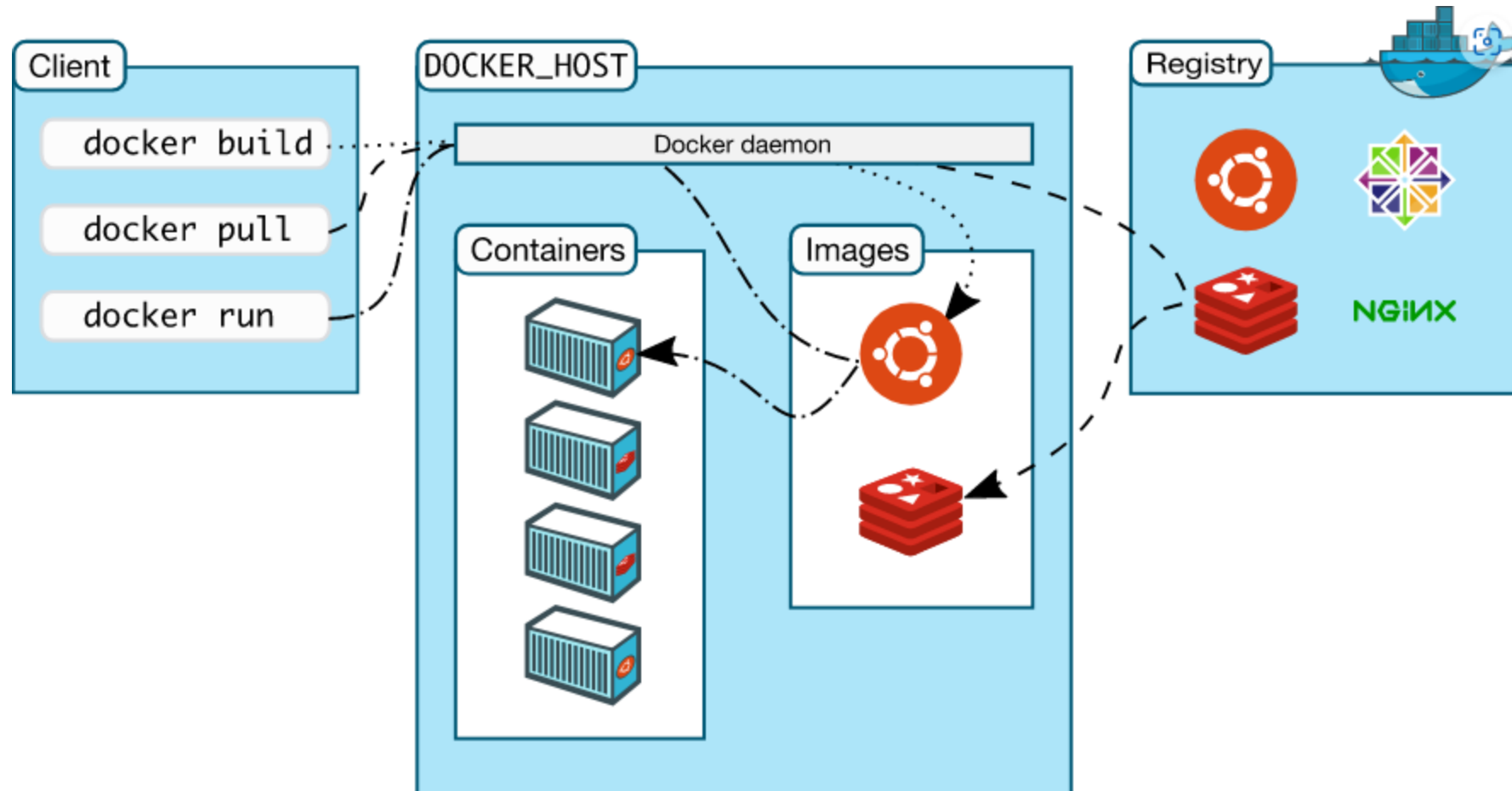
The screenshot shows the Docker Desktop application window. The top bar is blue with the 'Docker Desktop' logo, an 'Update to latest' button, a search bar with 'Ctrl+K', and a 'Sign in' button. The left sidebar contains navigation links for 'Containers', 'Images', 'Volumes', 'Dev Environments' (marked BETA), and 'Extensions' (marked BETA). The main area is titled 'Containers' and includes a toggle for 'Only show running containers' and a search bar. Below this is a table of containers. The table has columns for a checkbox, NAME, IMAGE, STATUS, PORT(S), STARTED, and ACTIONS. Four containers are listed, all with a status of 'Exited'. The first container, 'priceless\_mestor', has a 'Start' button in its actions column. The bottom status bar shows system metrics: RAM 2.92GB, CPU 0.80%, and a connection status 'Not connected to Hub'. The version 'v4.14.1' is also displayed.

	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
<input type="checkbox"/>	priceless_mestor 00f72c29a2ee	<a href="#">rancher/rancher:</a>	Exited (1)	443:443 80:80		<span>Start</span> <span>⋮</span> <span>🗑️</span>
<input type="checkbox"/>	infallible_archim 2b4bef5479eb	<a href="#">rancher/rancher-</a>	Exited (137)			<span>▶</span> <span>⋮</span> <span>🗑️</span>
<input type="checkbox"/>	relaxed_brown 8714f2ebb88b	<a href="#">rancher/rancher-</a>	Exited (137)			<span>▶</span> <span>⋮</span> <span>🗑️</span>
<input type="checkbox"/>	nostalgic_cannor f5abe75f2d44	<a href="#">itzg/minecraft-sei</a>	Exited (137)	25565:25565		<span>▶</span> <span>⋮</span> <span>🗑️</span>

Showing 4 items

RAM 2.92GB CPU 0.80% Not connected to Hub v4.14.1

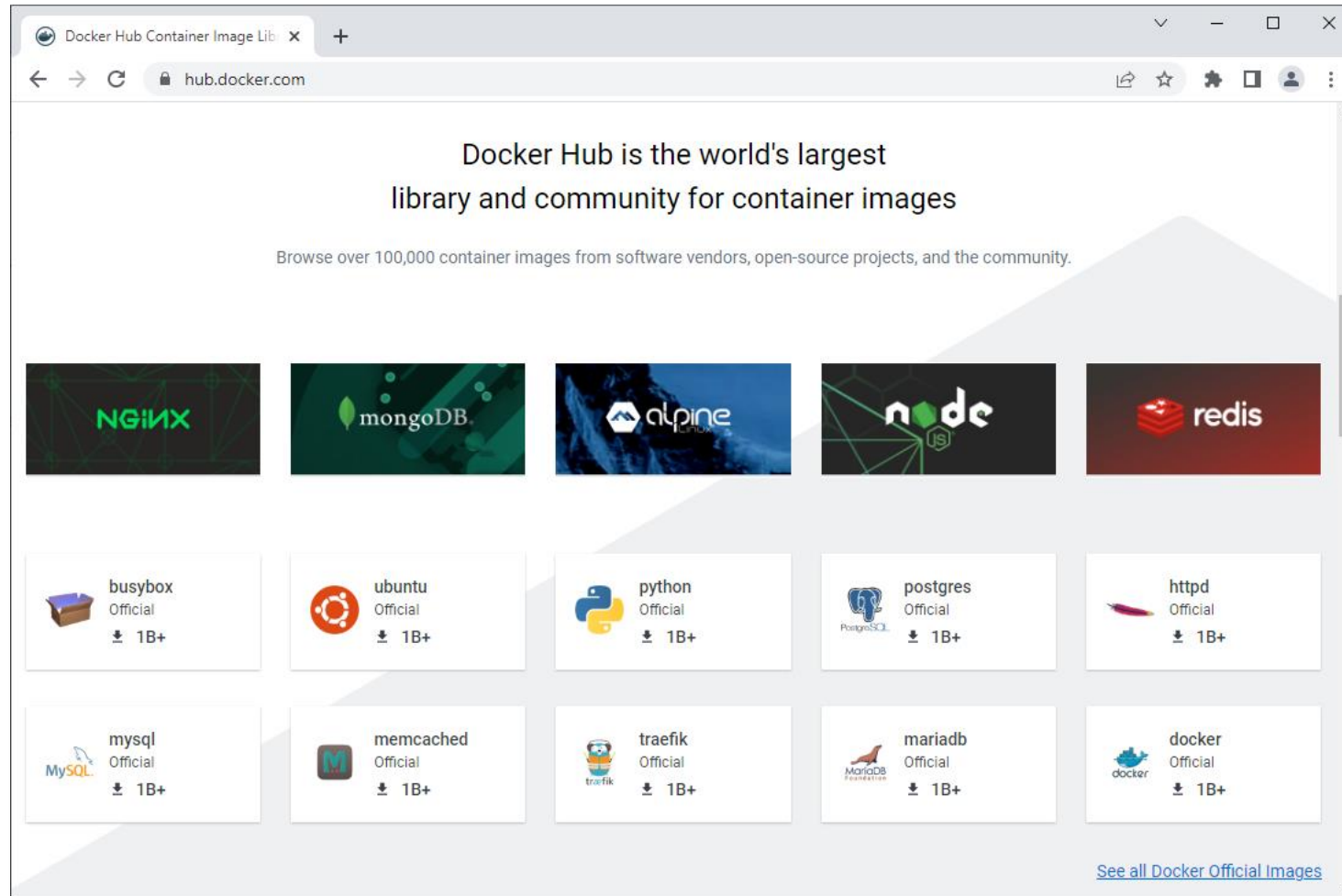
# Docker Architecture



# Running Docker Images

- `docker run hello-world`
- `docker run -d -p 80:80 docker/getting-started`

# Docker Images

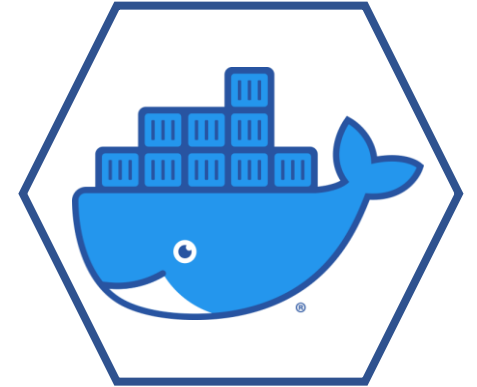




# httpd Image

- `docker run -dit --name my-apache-app -p 8080:80 -v "$PWD":/usr/local/apache2/htdocs/ httpd:2.4`

# Docker

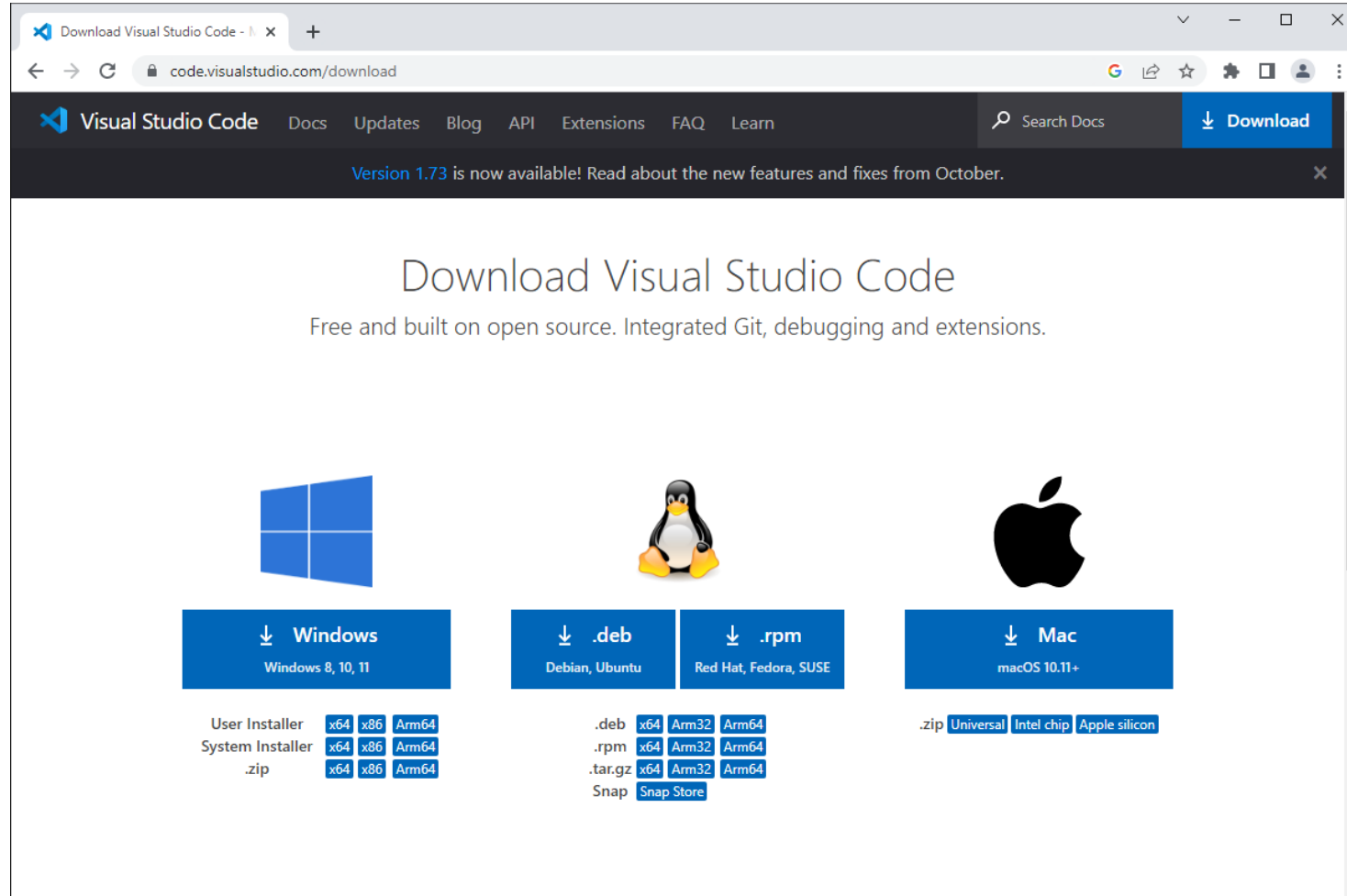


- Docker CLI
  - <https://docs.docker.com/engine/reference/commandline/cli/>
- Common Commands
  - docker version
  - docker pull/push **NAME[:TAG]**
  - docker image/container ls -a
  - docker image rm **Image**
  - docker container create **Container**
  - docker container start/stop/rm **Container**

# Docker CLI

- `docker --version`
- `docker --help`
- `docker ps -a`
- `docker run -dp 80:80 docker/getting-started`
  - `docker run -dp 3002:80 docker/getting-started`
- `docker logs -n 5 Container`
- `docker stop Container`
- `docker images`

# Visual Studio Code



# git

The screenshot shows the Git website homepage. At the top, there's a navigation bar with the Git logo and the tagline "--distributed-is-the-new-centralized". Below this, a search bar is present. The main content area features two paragraphs describing Git as a free and open source distributed version control system, highlighting its speed, efficiency, ease of learning, and tiny footprint. To the right of the text is a diagram illustrating distributed version control with multiple stacks of code connected by lines. Below the main text, there are four sections: "About" (advantages of Git), "Documentation" (command reference, Pro Git book, etc.), "Downloads" (GUI clients, binary releases), and "Community" (bug reporting, mailing list, etc.). On the right side, a monitor displays the latest source release "2.39.0" with a "Download for Windows" button. The browser's address bar shows "https://git-scm.com".

**git** --distributed-is-the-new-centralized

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

**About**  
The advantages of Git compared to other source control systems.

**Documentation**  
Command reference pages, Pro Git book content, videos and other material.

**Downloads**  
GUI clients and binary releases for all major platforms.

**Community**  
Get involved! Bug reporting, mailing list, chat, development and more.

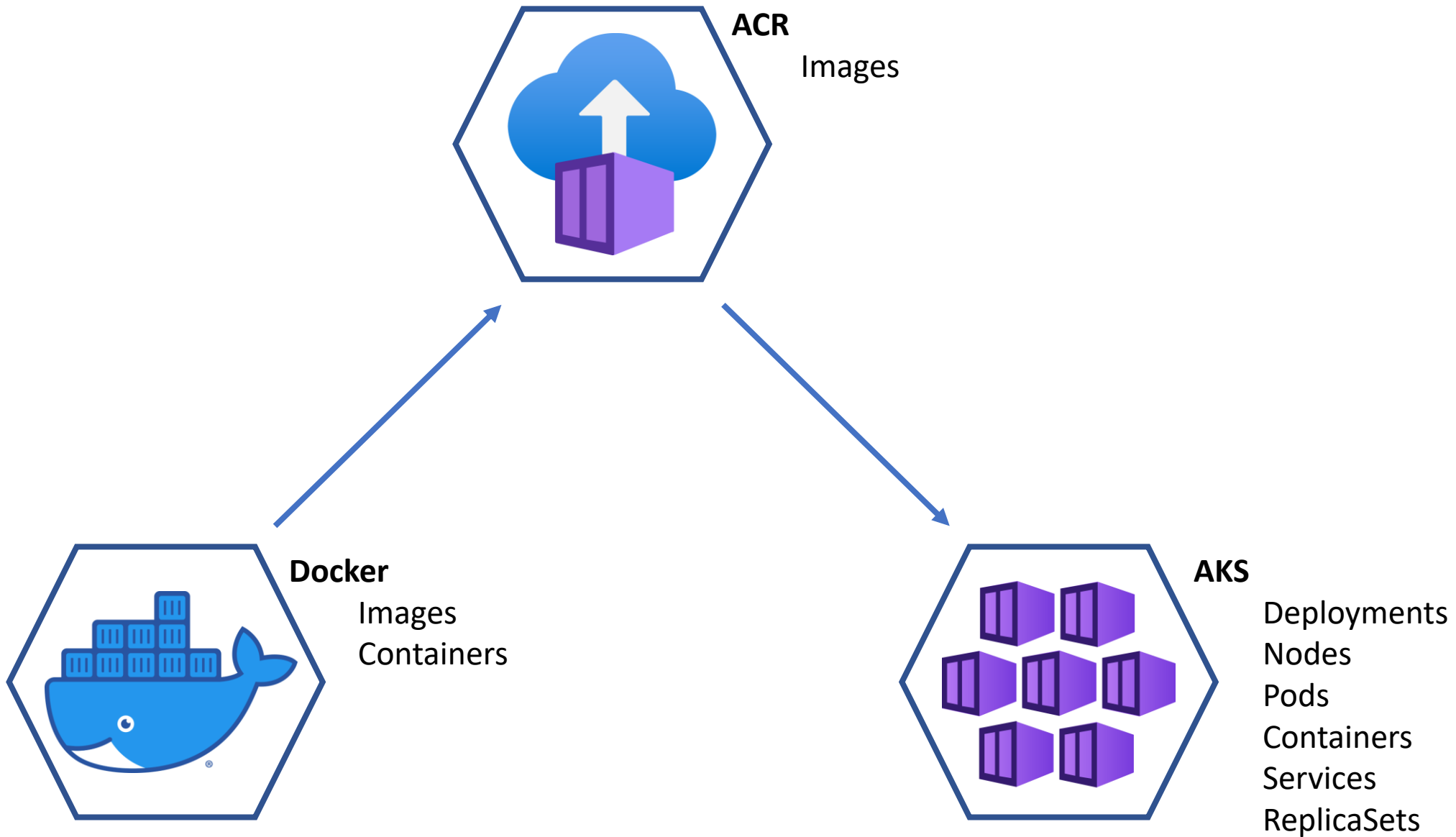
Latest source Release  
**2.39.0**  
[Release Notes \(2022-12-12\)](#)  
[Download for Windows](#)

# Building Docker Images

- `git clone https://github.com/docker/getting-started.git`
- Create Dockerfile

```
# syntax=docker/dockerfile:1
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000
```

- `docker build -t getting-started`
- `docker run -dp 3000:3000 getting-started`



# Kubernetes (K8S)

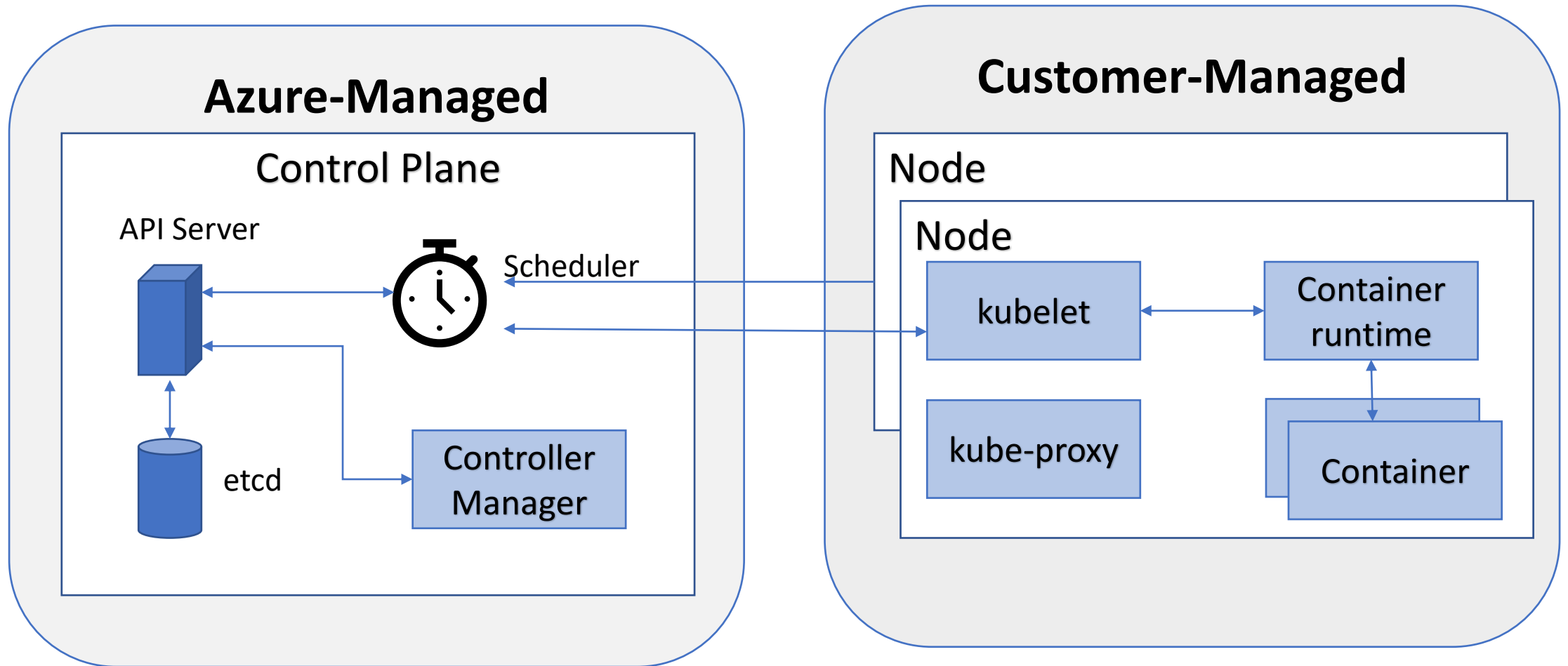
## Container Based Application Management



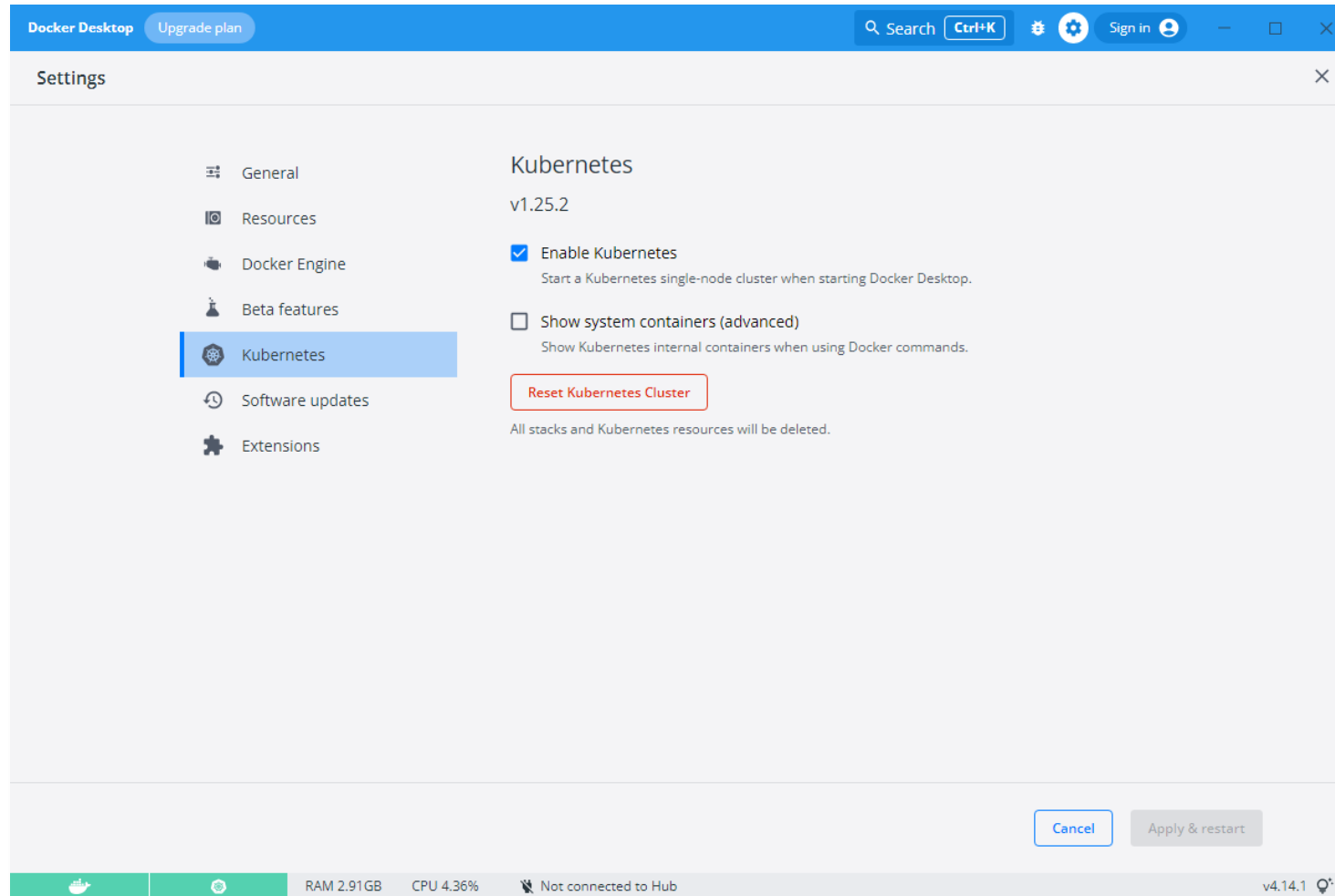
- Containerized infrastructure
- Auto-scalable infrastructure
- Loosely coupled infrastructure
- Higher density of resource utilization
- Declarative configuration



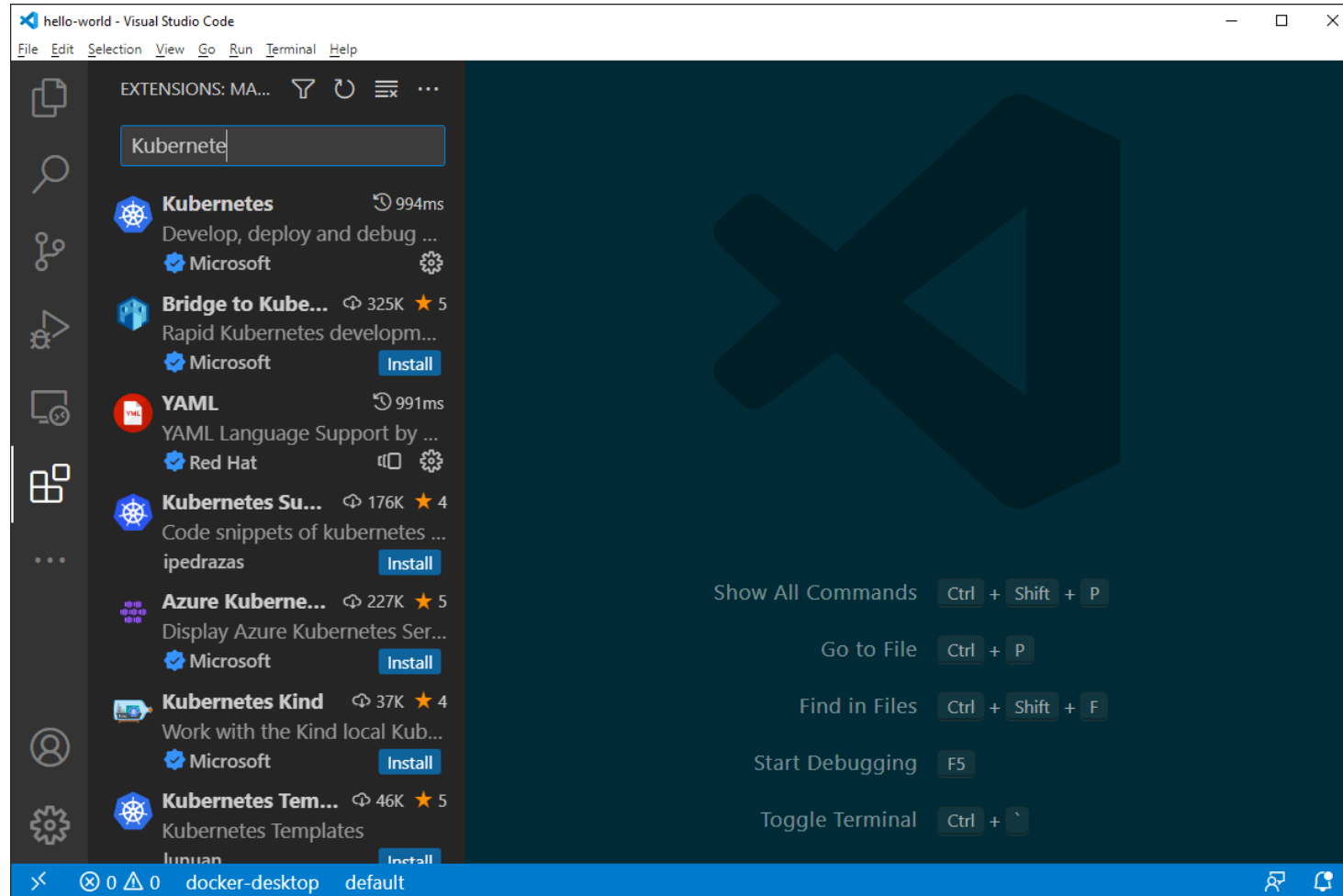
# Kubernetes Cluster Architecture



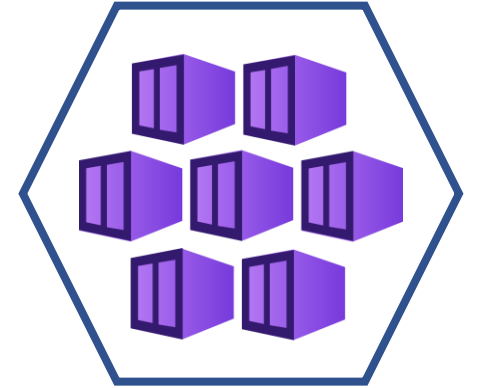
# Enable Kubernetes



# Visual Studio Code Extensions



# kubectl



- kubectl Installation
  - <https://kubernetes.io/docs/tasks/tools/install-kubectl/>
- Common Commands
  - kubectl get nodes/pods/deployments/svc
  - kubectl get node/pod/deployment/svc **NAME**
  - kubectl apply -f **file.yaml**

# kubectl Command Use Examples

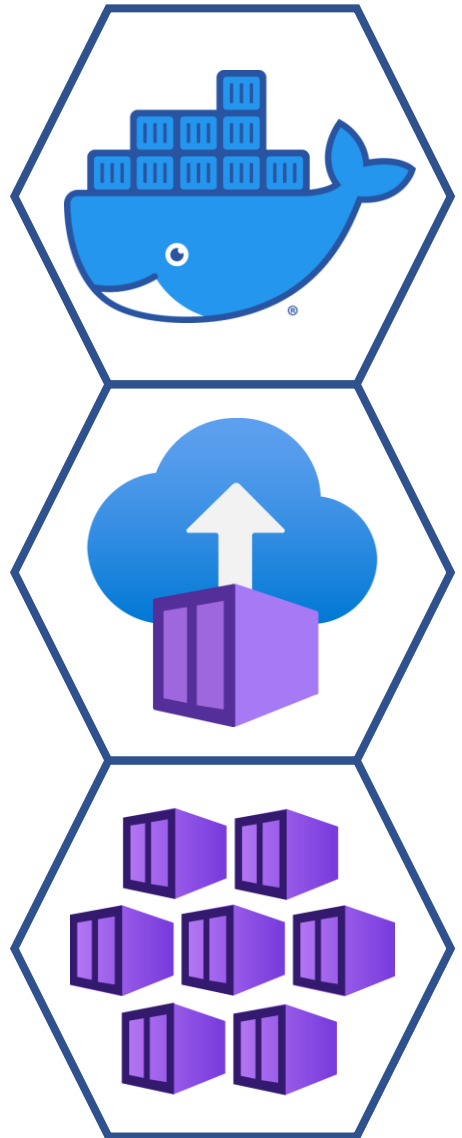
- `kubectl get nodes`
- `kubectl get pods`
- `kubectl get svc`

# Deploy Docker Images to Kubernetes

- `docker pull docker/getting-started`
- Grab bb.yaml from <https://docs.docker.com/get-started/kube-deploy/>
  - Modify targetPort from 3000 to 80
  - Modify replicas from 1 to 2
- `kubectl create -f bb.yaml`

# Summary

- Why use Docker Containers
  - Isolate app and its dependencies
  - More efficient use of system resources
  - Enables faster software delivery cycles
- Why use ACR, ECR, or other Image Repository
  - Automate Development
  - Collaborate with Team
  - Secure Docker Images
- Why use K8S
  - Minimize infrastructure maintenance
  - Automate repair, and scaling
  - Use only the resource you need



# About me!



Daniel Colón

<https://www.linkedin.com/in/danielecolon/>

A+, Security+, Azure Solutions Architect Expert

New Hampshire Cloud User Group

<https://www.meetup.com/nashuaug>



New Hampshire Cloud User Group





# Resources

- Docker Desktop  
<https://www.docker.com/products/docker-desktop/>
- Docker Images  
<https://hub.docker.com/search/?q=&type=image>
- Docker CLI  
<https://docs.docker.com/engine/reference/commandline/cli/>
- Kubernetes - Docker  
<https://www.docker.com/products/kubernetes/>