

Infrastructure

Course: Real-Time Backend

Lecturer: Gleb Lobanov

May, 2024



EPIC

Institute of Technology
Powered by epam

Contents

01 Virtualization

02 Docker

03 Nginx

04 Proxies

05 k8s

06 CI/CD

07 Life without ci/cd

08 Podman



EPIC

Institute of Technology
Powered by epam

01

Virtualization



EPIC

Institute of Technology
Powered by epam

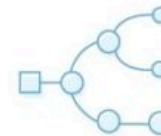
Types of Virtualization



Desktop



Data



Network



Storage



Server



Application



Cloud

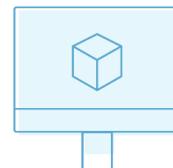
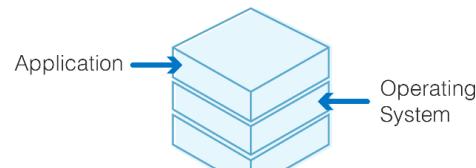


EPIC

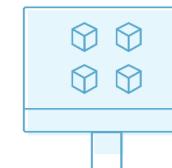
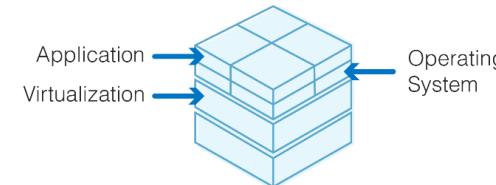
Institute of Technology
Powered by epam

What is Server Virtualization?

Traditional server architecture



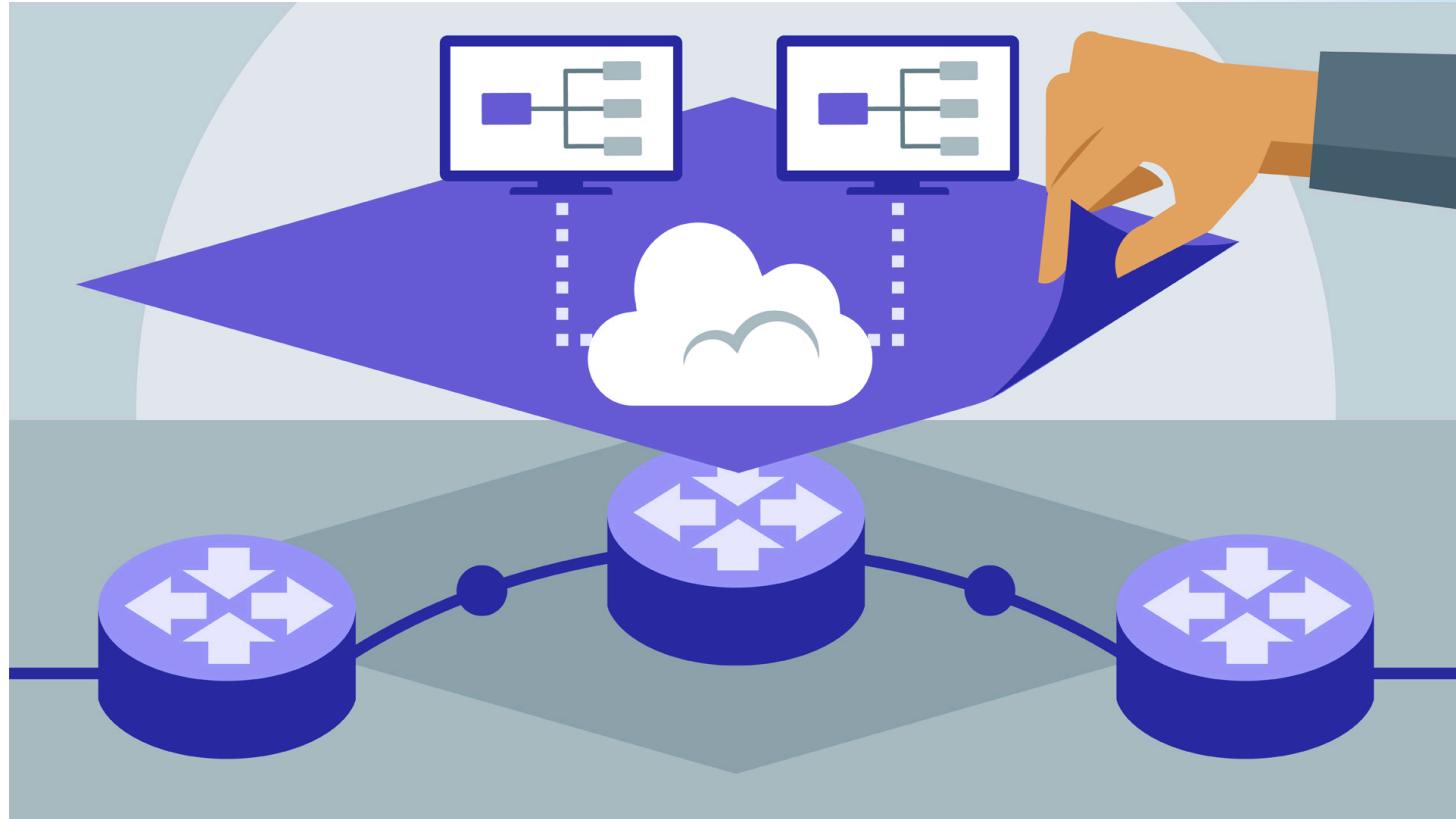
Virtualized server architecture





EPIC

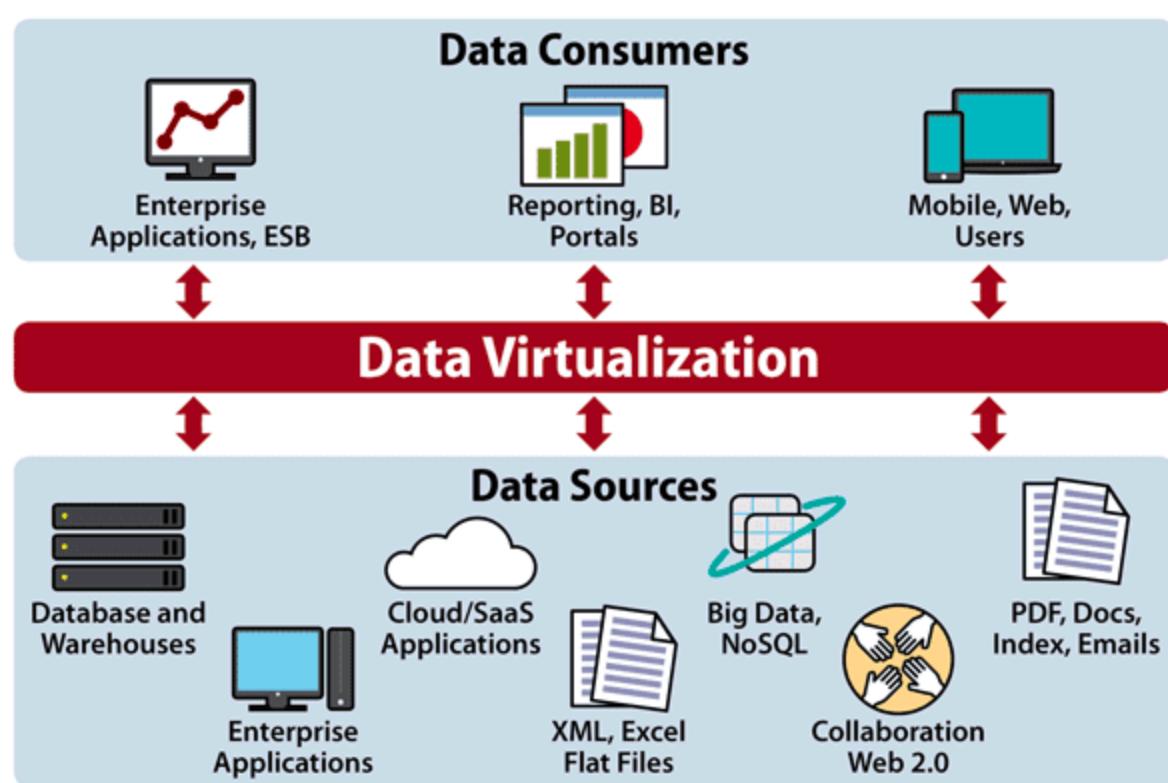
Institute of Technology
Powered by 





EPIC

Institute of Technology
Powered by epam





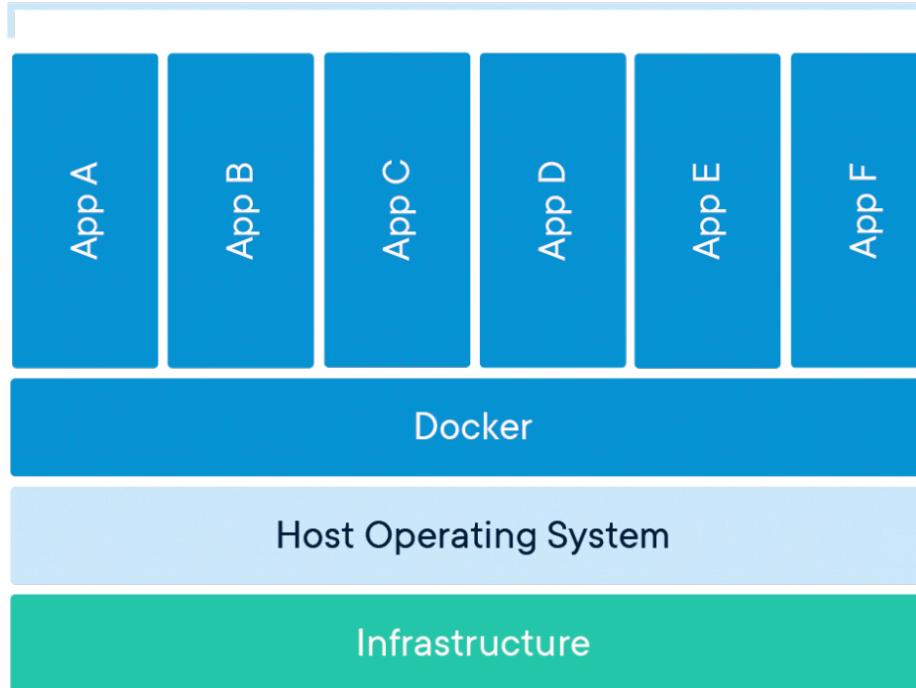
EPIC

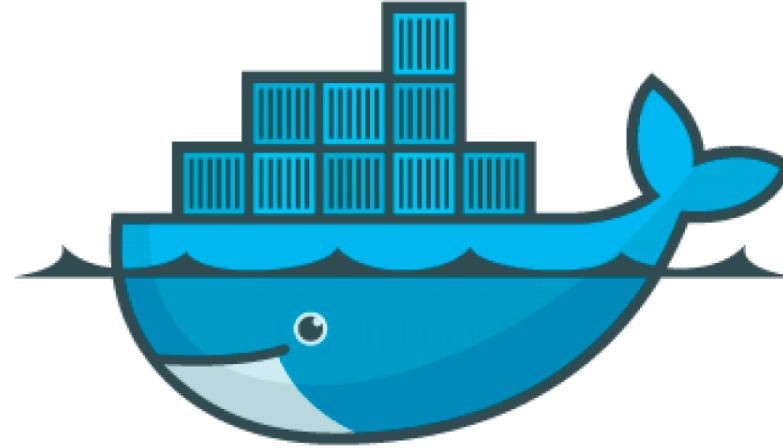
Institute of Technology
Powered by epam

02

Docker

Containerized Applications





docker

Advantages of containers

- 1) Portability
- 2) Isolation
- 3) Efficient use of resources
- 4) Simplified deployment and scaling
- 5) Dependency and configuration management
- 6) Rapid deployment and upgrades
- 7) Isolation and security
- 8) Easy development and testing



EPIC

Institute of Technology
Powered by epam



A screenshot of a macOS terminal window. The title bar shows an Apple icon, a home icon, and a tilde (~). The status bar at the bottom right shows a video camera icon, the number '1', an 'X' button, the time '00:01:40', and a circular refresh icon. The main pane of the terminal displays the following command and its execution:

```
brew install --cask docker
==> Downloading https://raw.githubusercontent.com/Homebrew/homebrew-cask/44ce0da9aade712
#####
==> Downloading https://desktop.docker.com/mac/main/arm64/106363/Docker.dmg
#####
==> Installing Cask docker
```



EPIC

Institute of Technology
Powered by epam

```
docker ps
CONTAINER ID     IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
70f5ac315c5a: Pull complete
Digest: sha256:fc6cf906cbfa013e80938cdf0bb199fbdbb86d6e3e013783e5a766f50f5dbce0
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (arm64v8)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

**EPIC**Institute of Technology
Powered by epam

```
docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
79d0ea7dc1a8: Pull complete
Digest: sha256:dfd64a3b4296d8c9b62aa3309984f8620b98d87e47492599ee20739e8eb54fbf
Status: Downloaded newer image for ubuntu:latest
root@224fa243b235:/# ls
bin  dev  home  media  opt  root  sbin  sys  usr
boot  etc  lib  mnt  proc  run  srv  tmp  var
root@224fa243b235:/# cat a.cpp
cat: a.cpp: No such file or directory
root@224fa243b235:/# cat /proc/version
Linux version 5.15.49-linuxkit (root@buildkitsandbox) (gcc (Alpine 10.2.1_pre1) 10.2.1 2
0201203, GNU ld (GNU Binutils) 2.35.2) #1 SMP PREEMPT Tue Sep 13 07:51:32 UTC 2022
root@224fa243b235:/# exit
exit
```



EPIC

Institute of Technology
Powered by epam

```
└ docker ps -h
Flag shorthand -h has been deprecated, please use --help

Usage: docker ps [OPTIONS]

List containers

Aliases:
  docker container ls, docker container list, docker container ps, docker ps

Options:
  -a, --all           Show all containers (default shows just running)
  -f, --filter filter Filter output based on conditions provided
  --format string     Format output using a custom template:
                      'table':             Print output in table format with
                      column headers (default)
                      'table TEMPLATE':   Print output in table format using
                      the given Go template
                      'json':              Print in JSON format
                      'TEMPLATE':          Print output using the given Go
                      template.
                      Refer to https://docs.docker.com/go/formatting/ for more
                      information about formatting output with templates
  -n, --last int      Show n last created containers (includes all states)
                      (default -1)
  -l, --latest        Show the latest created container (includes all states)
  --no-trunc          Don't truncate output
  -q, --quiet         Only display container IDs
  -s, --size          Display total file sizes
```

```
docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS		
		POR TS	NAMES			
224fa243b235	ubuntu		"bash"	2 minutes ago	Exited (
0)	About a minute ago		hopeful_mccarthy			
633724e24851	hello-world			"/hello"	3 minutes ago	Exited (
0)	3 minutes ago		hopeful_cerf			
a44725a4c6fb	pomomondreganto/ubuntu_ctf:latest			"/bin/zsh"	14 months ago	Exited (
0)	14 months ago		pensive_shirley			
947b907da5ca	pomomondreganto/ubuntu_ctf:latest			"/bin/zsh"	14 months ago	Exited (
0)	14 months ago		hopeful_varahamihira			

```
docker run --help
Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
Create and run a new container from an image
Aliases:
  docker container run, docker run
Options:
  --add-host list          Add a custom host-to-IP mapping (host:ip)
  -a, --attach list         Attach to STDIN, STDOUT or STDERR
  --blkio-weight uint16     Block IO (relative weight), between 10 and
                           1000, or 0 to disable (default 0)
  --blkio-weight-device list Block IO weight (relative device weight)
                           (default [])
  --cap-add list            Add Linux capabilities
  --cap-drop list           Drop Linux capabilities
  --cgroup-parent string   Optional parent cgroup for the container
  --cgroupns string         Cgroup namespace to use (host|private)
                           'host': Run the container in the Docker
                           host's cgroup namespace
                           'private': Run the container in its own
                           private cgroup namespace
                           ''': Use the cgroup namespace as
                           configured by the
                           default-cgroupns-mode option on
                           the daemon (default)
  --cidfile string          Write the container ID to the file
  --cpu-period int          Limit CPU CFS (Completely Fair Scheduler)
                           period
  --cpu-quota int           Limit CPU CFS (Completely Fair Scheduler) quota
  --cpu-rt-period int       Limit CPU real-time period in microseconds
  --cpu-rt-runtime int      Limit CPU real-time runtime in microseconds
  -c, --cpu-shares int      CPU shares (relative weight)
  --cpus decimal             Number of CPUs
  --cpuset-cpus string      CPUs in which to allow execution (0-3, 0,1)
  --cpuset-mems string      MEMs in which to allow execution (0-3, 0,1)
  -d, --detach               Run container in background and print
                           container ID
  --detach-keys string       Override the key sequence for detaching a
                           container
  --device list              Add a host device to the container
  --device-cgroup-rule list  Add a rule to the cgroup allowed devices list
  --device-read-bps list     Limit read rate (bytes per second) from a
                           device (default [])
  --device-read-iops list    Limit read rate (IO per second) from a
                           device (default [])
  --memory bytes             Memory limit
  --memory-reservation bytes Memory soft limit
  --memory-swap bytes       Swap limit equal to memory plus swap: '-1'
                           to enable unlimited swap
  --memory-swappiness int   Tune container memory swappiness (0 to
                           100) (default -1)
  --mount mount              Attach a filesystem mount to the container
  --name string              Assign a name to the container
  --network network          Connect a container to a network
  --network-alias list       Add network-scoped alias for the container
  --no-healthcheck           Disable any container-specified HEALTHCHECK
  --oom-kill-disable         Disable OOM Killer
  --oom-score-adj int        Tune host's OOM preferences (-1000 to 1000)
  --pid string               PID namespace to use
  --pids-limit int           Tune container pids limit (set -1 for
                           unlimited)
  --platform string          Set platform if server is multi-platform
                           capable
  --privileged               Give extended privileges to this container
  -p, --publish list          Publish a container's port(s) to the host
  -P, --publish-all           Publish all exposed ports to random ports
  --pull string               Pull image before running ("always",
                           "missing", "never") (default "missing")
  -q, --quiet                 Suppress the pull output
  --read-only                Mount the container's root filesystem as
                           read only
  --restart string            Restart policy to apply when a container
                           exits (default "no")
  --rm                       Automatically remove the container when it
                           exits
  --runtime string             Runtime to use for this container
  --security-opt list         Security Options
  --shm-size bytes            Size of /dev/shm
  --sig-proxy                Proxy received signals to the process
                           (default true)
  --stop-signal string        Signal to stop the container
  --stop-timeout int          Timeout (in seconds) to stop a container
  --storage-opt list          Storage driver options for the container
  --sysctl map                 Sysctl options (default map[])
  --tmpfs list                 Mount a tmpfs directory
  -t, --tty                   Allocate a pseudo-TTY
  --ulimit ulimit             Ulimit options (default [])
  --user string                Username or UID (format:
                           <name|uid>[:<group|gid>])
  --userns string              User namespace to use
  --uts string                 UTS namespace to use
  -v, --volume list            Bind mount a volume
  --volume-driver string       Optional volume driver for the container
  --volumes-from list          Mount volumes from the specified container(s)
  -w, --workdir string         Working directory inside the container
```



EPIC

Institute of Technology
Powered by 



	mcaramello Replace author's username with placeholder (#329) ...	def64e2 on Apr 10, 2024
 .github	Update github workflow to latest version	
 flask-app	Fix python error in catnip app (#313)	
 static-site	[ImgBot] Optimize images (#257)	
 tutorial	Replace author's username with placeholder (#329)	
 .firebaserc	fix some lighthouse warnings	
 .gitignore	use environment variable for port	
 LICENSE	Added MIT license	
 README.md	moved to metalsmith	
 nodemon.json	Added nodemon config & fixed the rest	
 package-lock.json	Bump undefsafe from 2.0.2 to 2.0.5 (#330)	
 package.json	Bump handlebars from 4.7.6 to 4.7.7 (#296)	

README.md

Docker Curriculum

Learn to build and deploy your distributed applications easily to the cloud with Docker

Follow the curriculum on docker-curriculum.com



EPIC

Institute of Technology
Powered by epam

```
remote: Compressing objects: 100% (67/67), done.
remote: Total 1682 (delta 55), reused 39 (delta 22), pack-reused 1591
Receiving objects: 100% (1682/1682), 8.93 MiB | 9.46 MiB/s, done.
Resolving deltas: 100% (948/948), done.

~ cd docker-curriculum/flask-app
~/docker-curriculum/flask-app | on master | ✓ | at 23:49:19

~/docker-curriculum/flask-app | on master | ✓ | at 23:49:31
-- pip3 install -r requirements.txt
Collecting Flask==2.0.2
  Downloading Flask-2.0.2-py3-none-any.whl (95 kB)
    95.2/95.2 kB 1.5 MB/s eta 0:00:00
Collecting Werkzeug>=2.0
  Downloading Werkzeug-2.3.4-py3-none-any.whl (242 kB)
    242.5/242.5 kB 3.9 MB/s eta 0:00:00
Collecting Jinja2>=3.0
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
    133.1/133.1 kB 5.5 MB/s eta 0:00:00
Collecting itsdangerous>=2.0
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting click>=7.1.2
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
    96.6/96.6 kB 4.0 MB/s eta 0:00:00
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.2-cp311-cp311-macosx_10_9_universal2.whl (17 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, Werkzeug, Jinja2, Flask
Successfully installed Flask-2.0.2 Jinja2-3.1.2 MarkupSafe-2.1.2 Werkzeug-2.3.4 click-8.1.3 itsdangerous-2.1.2

[notice] A new release of pip is available: 23.0.1 -> 23.1.2
[notice] To update, run: python3.11 -m pip install --upgrade pip

~/docker-curriculum/flask-app | on master | ✓ | at 23:49:50
-- ls
Dockerfile      app.py          templates
Dockerrun.aws.json  requirements.txt

~/docker-curriculum/flask-app | on master | ✓ | at 23:49:52
-- python3 app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.220.61.121:5000
Press CTRL+C to quit
127.0.0.1 - - [23/May/2023 23:50:32] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [23/May/2023 23:50:34] "GET /favicon.ico HTTP/1.1" 404 -
```



EPIC

Institute of Technology
Powered by epam

CAT GIF OF THE DAY



SENORGIF.COM

Courtesy: [Buzzfeed](#)



EPIC

Institute of Technology
Powered by epam

```
~/docker-curriculum/flask-app | on master !1
└─ cat Dockerfile
FROM python:3-onbuild

# set a directory for the app
WORKDIR /usr/src/app

# copy all the files to the container
COPY . .

# install dependencies
RUN pip install --no-cache-dir -r requirements.txt

# tell the port number the container should expose
EXPOSE 5000

# run the command
CMD ["python", "./app.py"]
```



Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?

```
~/.docker-curriculum/flask-app | on master !1
└── docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

~/.docker-curriculum/flask-app | on master !1
└── docker build -t glebodin/example_for_epum .
[+] Building 33.8s (12/12) FINISHED
--> [internal] load build definition from Dockerfile
--> transferring dockerfile: 346B
--> [internal] load .dockerignore
--> transferring context: 2B
--> [internal] load metadata for docker.io/library/python:3-onbuild
--> [internal] load build context
--> transferring context: 3.42kB
--> [1/4] FROM docker.io/library/python:3-onbuild@sha256:76df62c122c910751d8cd3101f8e3da39ef4ee828686b7ff0b5a5b1d967553f
--> resolve docker.io/library/python:3-onbuild@sha256:76df62c122c910751d8cd3101f8e3da39ef4ee828686b7ff0b5a5b1d967553f
--> sha256:a06a975b738391d72254582d5a2537ac0961581af50ec5f7a70fd62e01f96b9 50.19MB / 50.19MB
--> sha256:2095682ba1c8ea76d23358f918685d48abce3f28fe2f04e491ea9fe899b2a4f 39.77MB / 39.77MB
--> sha256:6ee7ce2e42f35f3b5e6cad14f2880c367f6555ca8aab7df9eb00646b2990da93a 2.21kB / 2.21kB
--> sha256:d70a07d587c96c1a7f3d0086d75422996051bcb2369f68079880f77d734e 8.39kB / 8.39kB
--> sha256:98ef1dbac9624e93cfa5a7ac63c8d126bc65202d2536fb411e8506d69e563985f 16.75MB / 16.75MB
--> sha256:76df62c122c910751d8cd3101f8e3da39ef4ee828686b7ff0b5a5b1d967553f 1.41kB / 1.41kB
--> sha256:c9ee6c3d046736819eef47a50e29ff8eaa46725af7ba7f0e970718943de4af 114.09MB / 114.09MB
--> extracting sha256:ra06d975b738391d72254582d5a2537ac0961581af50ec5f7a76df622e01f96b9
--> sha256:7eebaae1fb882d65757680f32f812befaa833a61d91c221fb018524fc35854 2.90MB / 2.90MB
--> sha256:0e613d41dc587e7780d0c446289a306dc31f34e69e900820ef761136ab819 19.09MB / 19.09MB
--> sha256:d60088a689e2215e8a515b258458fc4a018728acf71a71b09331c4a54f7f2 240B / 240B
--> extracting sha256:98ef1dbac9624e93cfa5a7ac63c8d126bc65202d2536fb411e8506d69e563985f
--> extracting sha256:2095682ba1c8ea76d23358f9f685d48abce3f28fe2f04e491ea9fe899b2a4d
--> sha256:eeb64c99ac75ff18ef9a34022570e5fd7ec643e0cbc1f3cc72370e676c3940d4e 1.79MB / 1.79MB
--> sha256:ed03a2e49b38cd6e4b43c2945e3ab2f6995043abeb3588664986885f803875b0 164B / 164B
--> extracting sha256:c9ee6c3d04736819eef47a350cc13ae029f2f6ea46725af7ba7f0e9707189d34e4f
--> extracting sha256:7eebaae1fb882d65750086321812befaa833a61d91c221fb018524fc35854
--> extracting sha256:0e613d41dc587e7780d0e446289a306dc31f34e69e900820ef761136ab819
--> extracting sha256:d60088a689e2215e8a515b258458fc4a018728acf71a71b09331c4a54f7f2
--> extracting sha256:eeb64c99ac75ff18ef9a34022570e5fd7ec643e0cbc1f3cc72370e676c3940d4e
--> extracting sha256:ed03a2e49b38cd6e4b43c2945e3ab2f6995043abeb3588664986885f803875b0
--> [2/4] COPY requirements.txt /usr/src/app/
--> [3/4] RUN pip install --no-cache-dir -r requirements.txt
--> [4/4] COPY . /usr/src/app
--> [5/4] WORKDIR /usr/src/app
--> [6/4] COPY . .
--> [7/4] RUN pip install --no-cache-dir -r requirements.txt
--> exporting to image
--> exporting layers
--> writing image sha256:dc87ff8101af486a62e72be093389e2e7099db1172cae5932629326c5ad1aff3
--> naming to docker.io/glebodin/example_for_epum
```

```
~/.docker-curriculum/flask-app | on master !1
```



Docker Desktop

Upgrade plan

Search for local and remote images, containers, and more...

glebod...

Containers

Images

Volumes

 Dev Environments BETA

Learning Center

Extensions

⋮

Add Extensions

Images Give feedback

Local

Hub

Artifactory EARLY ACCESS

glebodin

⋮

TAGS

OS

VULNERABILITIES

LAST PUSHED

SIZE

glebodin/cheers2019

latest



Inactive

over 3 years ago

1.96 MB

glebodin/my_ubuntu

No tags

glebodin/example_for_e...

latest



Inactive

less than a minute ago

248.37 MB

[View in Hub](#) [Pull](#)

glebodin/example_for_epum

Repositories per page

5

1–3 of 3



RAM 2.37 GB CPU 0.25% Disk 46.87 GB avail. of 58.37 GB Connected to Hub

v4.19.0

```
INT x 01:32:37 ⏺
└─ docker run -p 8888:5000 glebodin/example_for_epum
  * Serving Flask app 'app' (lazy loading)
  * Environment: production
    WARNING: This is a development server. Do not use it in a production deployment.
    Use a production WSGI server instead.
  * Debug mode: off
  * Running on all addresses.
    WARNING: This is a development server. Do not use it in a production deployment.
  * Running on http://172.17.0.3:5000/ (Press CTRL+C to quit)
172.17.0.1 -- [23/May/2023 22:35:50] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [23/May/2023 22:35:51] "GET /favicon.ico HTTP/1.1" 404 -
```



EPIC

Institute of Technology
Powered by epam

CAT GIF OF THE DAY



Courtesy: [Buzzfeed](#)



Docker Desktop Upgrade plan

Search for local and remote images, containers, and more... ⌘K

glebod...

Containers

Images

Volumes

Dev Environments BETA

Learning Center

Extensions

Images Give feedback

Local Hub Artifactory EARLY ACCESS

6.81 GB / 6.24 GB in use 5 images

Last refresh: 8 minutes ago

Search



<input type="checkbox"/>	Name	Tag	Status	Created	Size	Actions
<input type="checkbox"/>	glebodin/example_for_epum dc87f8101af4	latest	In use	22 minutes ago	568.72 MB	
<input type="checkbox"/>	hello-world b038788db22	latest	In use	20 days ago	9.2 MB	
<input type="checkbox"/>	ubuntu 3f5ef9003cef	latest	In use	28 days ago	69.2 MB	
<input type="checkbox"/>	pomondreganto/ubuntu_ctf 2fdb5b3da139	latest	In use	about 3 years	6.04 GB	
<input type="checkbox"/>	prakhar1989/static-site f01030e1dcf3	latest	In use	over 7 years ago	133.88 MB	

Showing 5 items

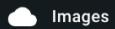


RAM 3.10 GB Disk 46.27 GB avail. of 58.37 GB Connected to Hub

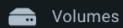
v4.19.0



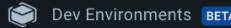
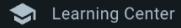
Containers



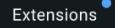
Images



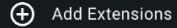
Volumes

Dev Environments BETA

Learning Center



Extensions



Add Extensions

Images Give feedback

Local

Hub

Artifactory EARLY ACCESS

6.81 GB / 6.24 GB in use 5 images

Last refresh: 8 minutes ago

Search



<input type="checkbox"/>	Name	Tag	Status	Created	Size	Actions
<input type="checkbox"/>	glebodin/example_for_epum dc87f8101af4	latest	In use	22 minutes ago	568.72 MB	 View packages and CVEs Pull Push to Hub
<input type="checkbox"/>	hello-world b038788ddb22	latest	In use	20 days ago	9.2 MB	 Pull Push to Hub
<input type="checkbox"/>	ubuntu 3f5ef9003cef	latest	In use	28 days ago	69.2 MB	 Pull Push to Hub
<input type="checkbox"/>	pomondreganto/ubuntu_ctf 2fdb5b3da139 AMD64	latest	In use	about 3 years	6.04 GB	 Pull Push to Hub
<input type="checkbox"/>	prakhar1989/static-site f01030e1dcf3 AMD64	latest	In use	over 7 years ago	133.88 MB	 Pull Push to Hub

Showing 5 items



RAM 3.10 GB Disk 46.27 GB avail. of 58.37 GB Connected to Hub

v4.19.0

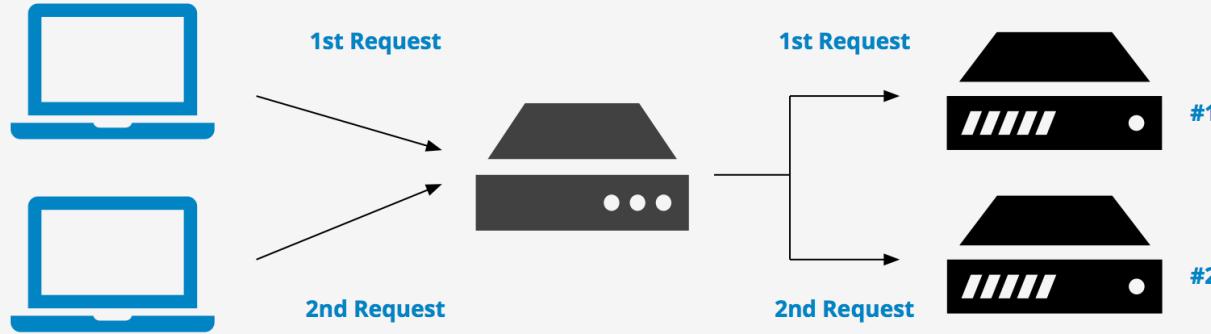
03

Nginx



EPIC

Institute of Technology
Powered by epam



Load Balancing

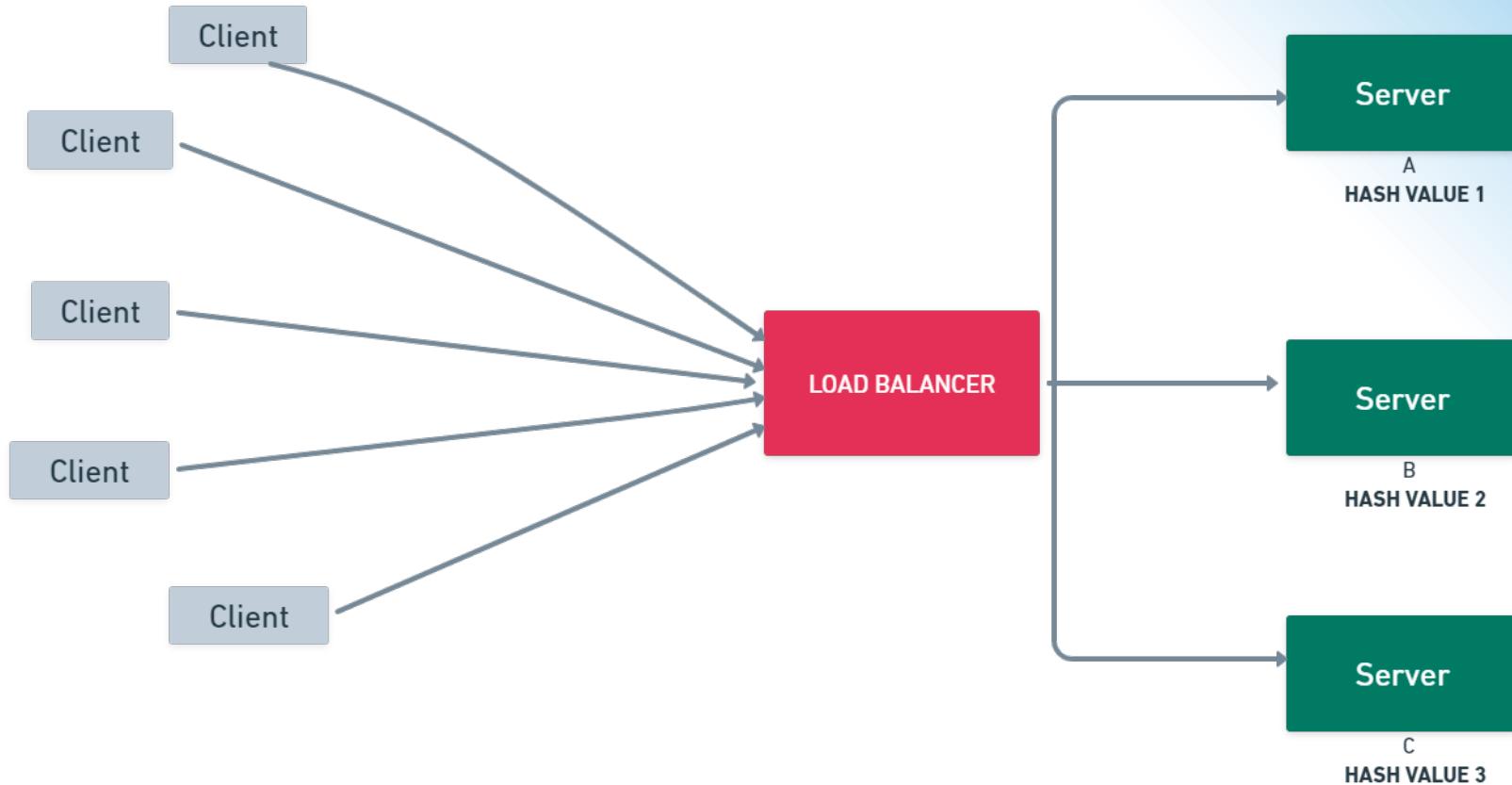
Load Balancer

Server

Client

Methods of balancing

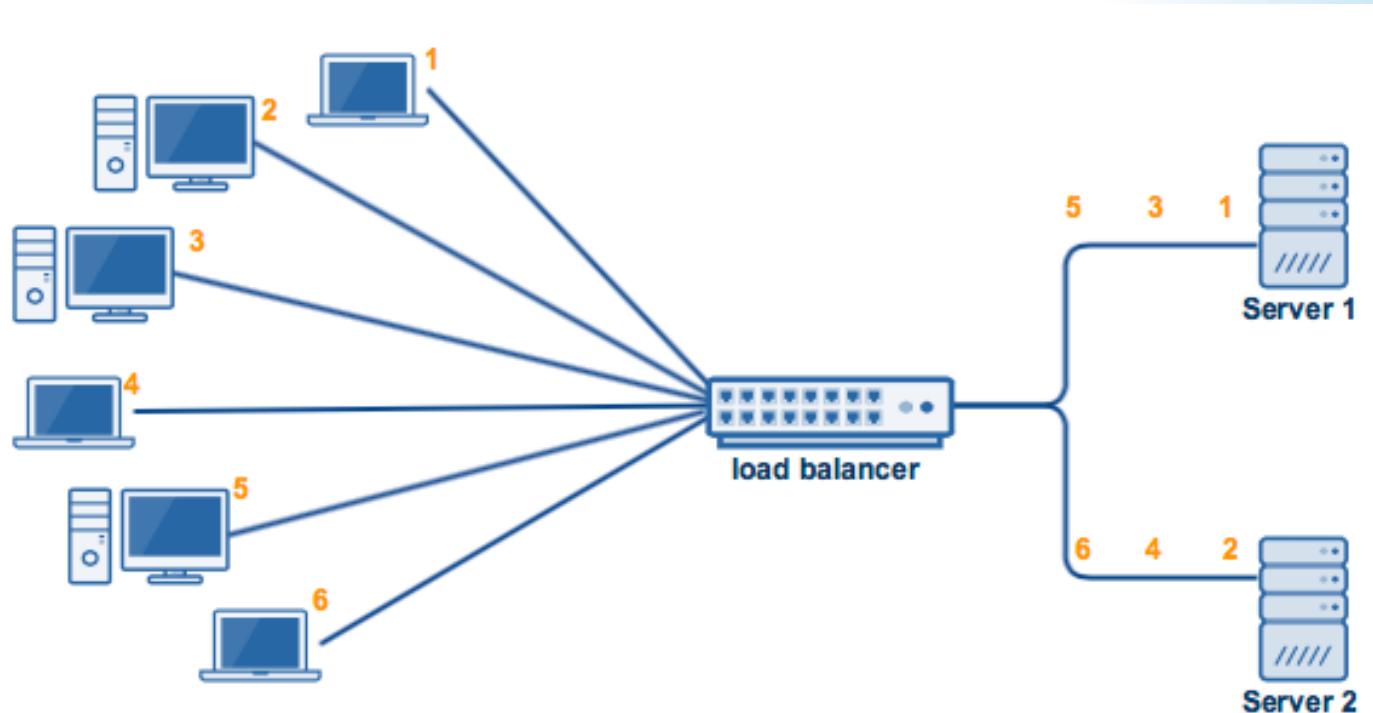
- 1) Round Robin
- 2) Least Connections
- 3) IP Hash
- 4) Least Response Time
- 5) Adaptive
- 6) Geobalancing

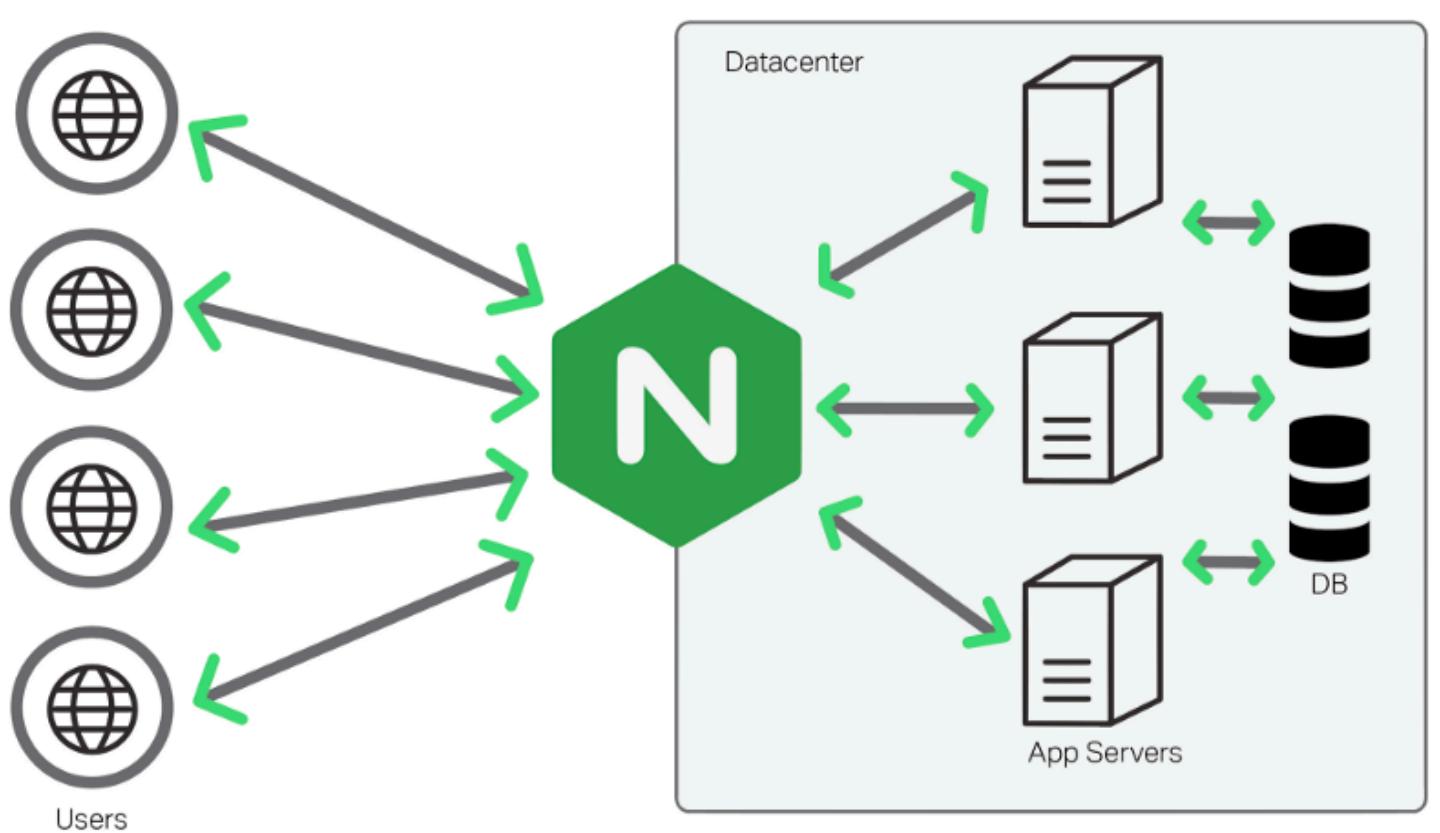


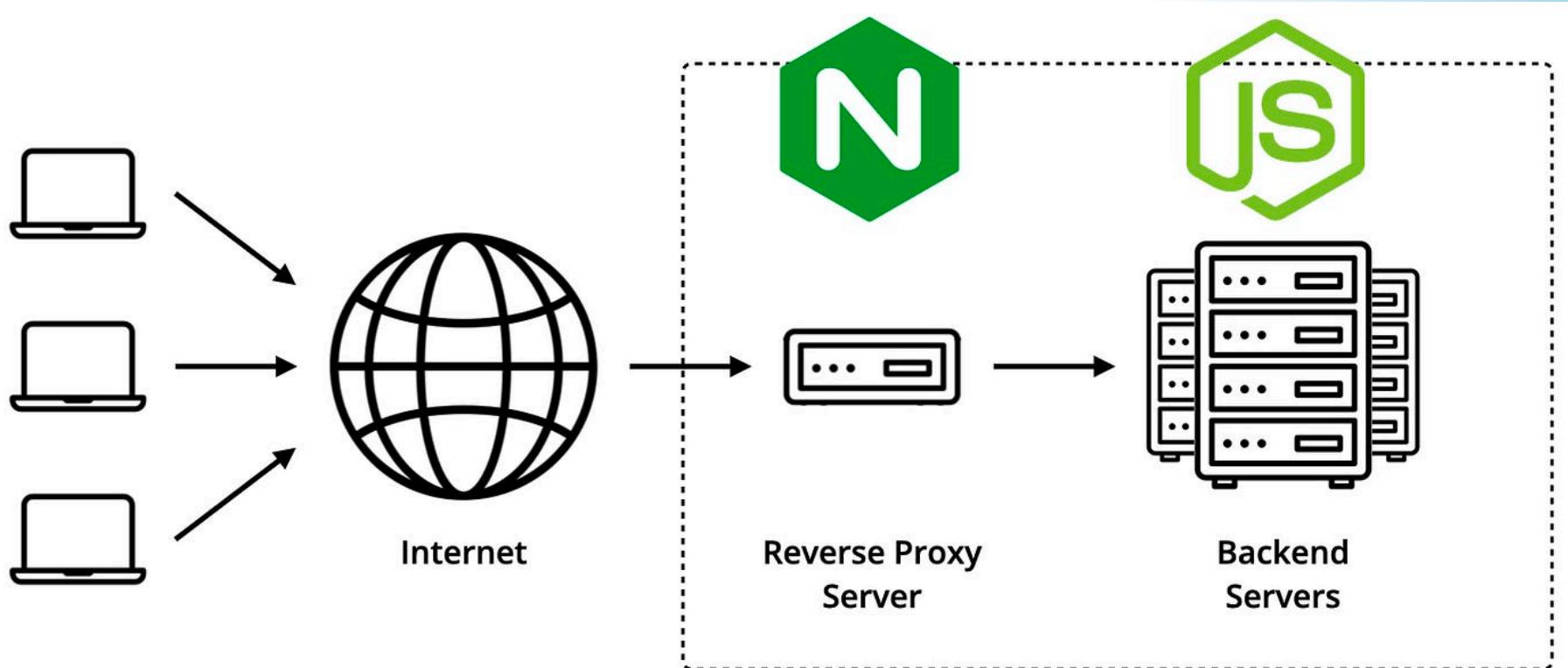


EPIC

Institute of Technology
Powered by epam







The nginx functions

- 1) Web server
- 2) Reverse Proxy
- 3) Load balancer
- 4) Scalability and Resiliency
- 5) WebSocket Proxying
- 6) caching
- 7) SSL/TLS Termination



EPIC

Institute of Technology
Powered by epam

Have you heard? [njs](#) has moved to [GitHub](#). [Read more.](#)

nginx news

2024-06-05 The [njs](#) project has officially moved to [GitHub](#). [Read more.](#)

2024-05-29 [nginx-1.26.1](#) stable and [nginx-1.27.0](#) mainline versions have been released, with fixes for [vulnerabilities in HTTP/3](#) (CVE-2024-32760, CVE-2024-31079, CVE-2024-35200, CVE-2024-34161).

2024-04-23 [nginx-1.26.0](#) stable version has been released, incorporating new features and bug fixes from the 1.25.x mainline branch — including experimental [HTTP/3 support](#), HTTP/2 on a [per-server](#) basis, [virtual servers](#) in the stream module, [passing](#) stream connections to listen sockets, and more.

2024-04-16 [nginx-1.25.5](#) mainline version has been released.

2024-04-16 [njs-0.8.4](#) version has been [released](#).

2024-03-26 [unit-1.32.1](#) bugfix version has been [released](#).

2024-02-27 [unit-1.32.0](#) version has been [released](#), featuring the WebAssembly Component Model and Unit variables access from njs.

2024-02-14 [nginx-1.25.4](#) mainline version has been released, with fixes for [vulnerabilities in HTTP/3](#) (CVE-2024-24989, CVE-2024-24990).

2024-02-07 [njs-0.8.3](#) version has been [released](#).

2023-10-24 [nginx-1.25.3](#) mainline version has been released.

NGINX

english

[русский](#)

news

[2023](#)

[2022](#)

[2021](#)

[2020](#)

[2019](#)

[2018](#)

[2017](#)

[2016](#)

[2015](#)

[2014](#)

[2013](#)

[2012](#)

[2011](#)

[2010](#)

[2009](#)

[about](#)

[download](#)

[security](#)

[documentation](#)

[faq](#)

[books](#)

[community](#)



EPIC

Institute of Technology
Powered by epam

```
brew install nginx
Running `brew update --auto-update`...
==> Auto-updated Homebrew!
Updated 3 taps (mongodb/brew, homebrew/core and homebrew/cask).
==> New Formulae
apk0           libpaho-mqtt          tailwindcss-language-server
libfastjson    melange              shodan
libmemcached   ==> New Casks
chatbox        dintch               filemonitor        processmonitor
copilot        eu                   firefly-shimmer    tea
craft          eusamanager         lasso
Warning: Calling plist_options is deprecated! Use service.require_root instead.
Please report this issue to the yandex/arc tap (not Homebrew/brew or Homebrew/homebrew-core), or even better, submit a PR to fix it:
/opt/homebrew/Library/Taps/yandex/homebrew-arc/arc-launcher.rb:14

You have 44 outdated formulae installed.

==> Fetching nginx
==> Downloading https://ghcr.io/v2/homebrew/core/nginx/manifests/1.25.0
#####
==> Downloading https://ghcr.io/v2/homebrew/core/nginx/blobs/sha256:f3442e5a2286570d9689
==> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256:f34
#####
==> Pouring nginx--1.25.0.arm64_big_sur.bottle.tar.gz
==> Caveats
Docroot is: /opt/homebrew/var/www

The default port has been set in /opt/homebrew/etc/nginx/nginx.conf to 8080 so that
nginx can run without sudo.
```



EPIC

Institute of Technology
Powered by epam

```
brew services start nginx
==> Successfully started `nginx` (label: homebrew.mxcl.nginx)

nginx -t
nginx: the configuration file /opt/homebrew/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /opt/homebrew/etc/nginx/nginx.conf test is successful

brew services list
Warning: Calling plist_options is deprecated! Use service.require_root instead.
Please report this issue to the yandex/arc tap (not Homebrew/brew or Homebrew/homebrew-core), or even better, submit a PR to fix it:
/opt/homebrew/Library/Taps/yandex/homebrew-arc/arc-launcher.rb:14

nginx          started      glebodin ~/Library/LaunchAgents/homebrew.mxcl.nginx.p
list
```

```
nginx -h
nginx version: nginx/1.25.3
Usage: nginx [-?hvVtTq] [-s signal] [-p prefix]
            [-e filename] [-c filename] [-g directives]

Options:
  -?, -h      : this help
  -v          : show version and exit
  -V          : show version and configure options then exit
  -t          : test configuration and exit
  -T          : test configuration, dump it and exit
  -q          : suppress non-error messages during configuration testing
  -s signal   : send signal to a master process: stop, quit, reopen, reload
  -p prefix   : set prefix path (default: /opt/homebrew/Cellar/nginx/1.25.3/)
  -e filename : set error log file (default: /opt/homebrew/var/log/nginx/error.log)
  -c filename : set configuration file (default: /opt/homebrew/etc/nginx/nginx.conf)
  -g directives : set global directives out of configuration file

nginx -s reopen
```



EPIC

Institute of Technology
Powered by 

04

k8s



EPIC

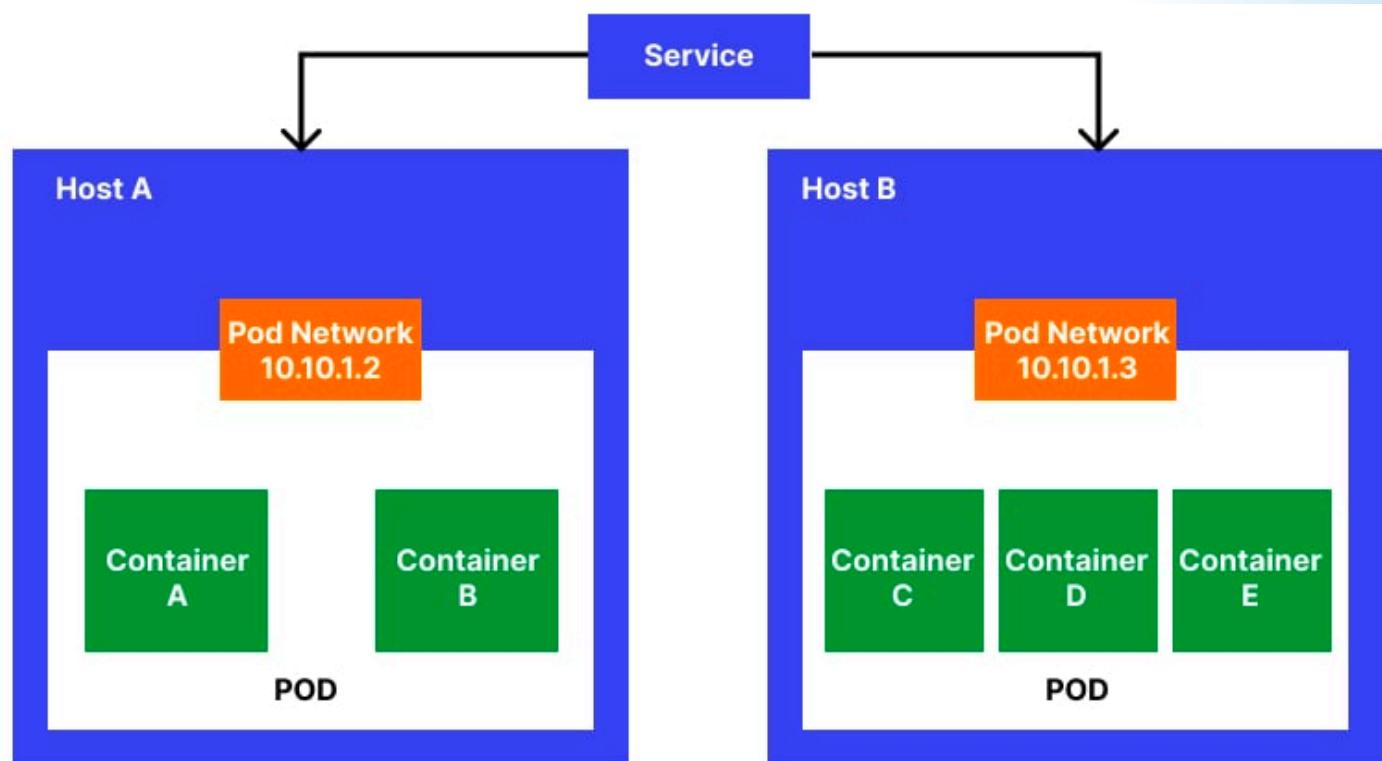
Institute of Technology
Powered by 



Advantages of k8s

- 1) Orchestration
- 2) Scalability
- 3) Self-healing
- 4) Monitoring and journaling
- 5) Declarative configuration

Kubernetes pod architecture



Features of pod

- 1) Containers
- 2) Overall life cycle
- 3) Shared resources
- 4) Unique IP address
- 5) Lifecycle Management
- 6) Management object



2GB dynamic
allocated for free



3GB reserved



2GB dynamically added



3GB reserved



EPIC

Institute of Technology
Powered by epam

If you have [go](#) 1.16+ and [docker](#) or [podman](#) installed `go install sigs.k8s.io/kind@v0.19.0 && kind create cluster` is all you need!

```
$ time kind create cluster
Creating cluster "kind" ...
✓ Ensuring node image (kindest/node:v1.16.3) 
✓ Preparing nodes 
✓ Writing configuration 
✓ Starting control-plane 
✓ Installing CNI 
✓ Installing StorageClass 
Set kubectl context to "kind-kind"
You can now use your cluster with:

kubectl cluster-info --context kind-kind

Not sure what to do next? 😊 Check out https://kind.sigs.k8s.io/docs/user/quick-start/

real      0m21.890s
user      0m1.278s
sys       0m0.790s
```



EPIC

Institute of Technology
Powered by epam

YAML

0.24 KB | None |

0

0

```
1. apiVersion: kind.x-k8s.io/v1alpha4
2. kind: Cluster
3. name: for-nginx
4. nodes:
5.   - role: control-plane
6.     extraPortMappings:
7.       - containerPort: 30080
8.         hostPort: 30080
9.         listenAddress: "0.0.0.0"
10.        protocol: TCP
11.   - role: worker
12.   - role: worker
```

[source](#)

```
kind create cluster --config kind.yaml
Creating cluster "for-nginx" ...
✓ Ensuring node image (kindest/node:v1.27.1) 
✓ Preparing nodes   
✓ Writing configuration 
✓ Starting control-plane 
✓ Installing CNI 
✓ Installing StorageClass 
✓ Joining worker nodes 
Set kubectl context to "kind-for-nginx"
You can now use your cluster with:

kubectl cluster-info --context kind-for-nginx

Not sure what to do next? 😊 Check out https://kind.sigs.k8s.io/docs/user/quick-start/
```

```
✓ 39s ✘ kind-for-nginx * ( 22
  kubectl get nodes
NAME           STATUS   ROLES      AGE    VERSION
for-nginx-control-plane Ready   control-plane 43s   v1.27.1
for-nginx-worker     Ready   <none>    22s   v1.27.1
for-nginx-worker2   Ready   <none>    19s   v1.27.1
```



EPIC

Institute of Technology
Powered by epam

YAML

0.28 KB | None | 0 0

```
1. apiVersion: v1
2. kind: Pod
3. metadata:
4.   creationTimestamp: null
5.   labels:
6.     run: nginx
7.   name: nginx
8. spec:
9.   containers:
10.    - image: nginx
11.      name: nginx
12.      ports:
13.        - containerPort: 80
14.      resources: {}
15.      dnsPolicy: ClusterFirst
16.      restartPolicy: Always
17.      status: {}
```

[source](#)

```
✓ kind-for-nginx * 22:57:34 ⓘ
└ kubectl apply -f nginx.yml
pod/nginx created

✓ kind-for-nginx * 22:57:43 ⓘ
└ kubectl get pods
NAME      READY   STATUS            RESTARTS   AGE
nginx    0/1     ContainerCreating   0          8s

✓ kind-for-nginx * 22:57:51 ⓘ
└ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx    1/1     Running   0          22s
```



EPIC

Institute of Technology
Powered by epam

YAML

0.28 KB | None | 0 0

```
1. apiVersion: v1
2. kind: Service
3. metadata:
4.   creationTimestamp: null
5.   labels:
6.     run: nginx
7.   name: nginx-nodeport
8. spec:
9.   ports:
10.    - port: 80
11.      protocol: TCP
12.      targetPort: 80
13.      nodePort: 30080
14.   selector:
15.     run: nginx
16.   type: NodePort
17. status:
18.   loadBalancer: {}
19.
```

[source](#)

**EPIC**Institute of Technology
Powered by epam

✓ kind-for-nginx * 23:02:05
└─ kubectl apply -f nginx-nodeport.yml
service/nginx-nodeport configured

✓ kind-for-nginx * 23:02:18
└─ kubectl get svc nginx-nodeport

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
nginx-nodeport	NodePort	10.96.74.63	<none>	80:30080/TCP	3m49s



EPIC

Institute of Technology
Powered by epam

```
curl http://localhost:30080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>. <br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>. </p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Usage of Containers

- 1) CI/CD
- 2) microservices
- 3) testing
- 4) security

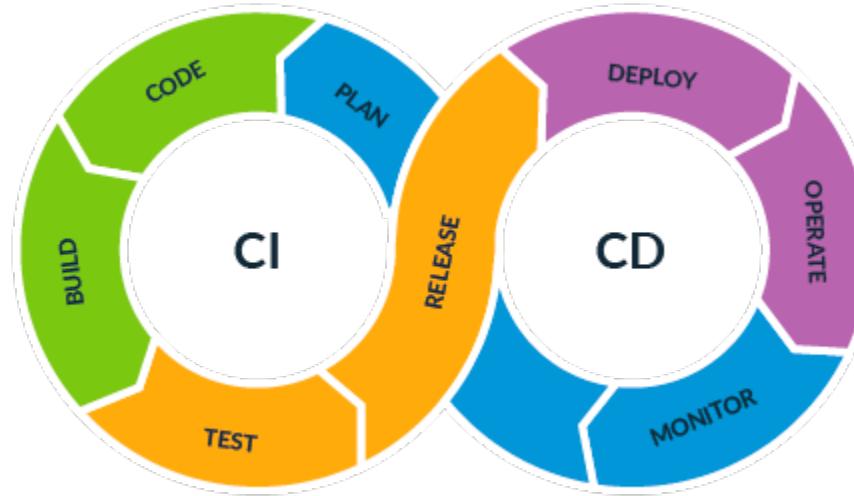
05

CI/CD



EPIC

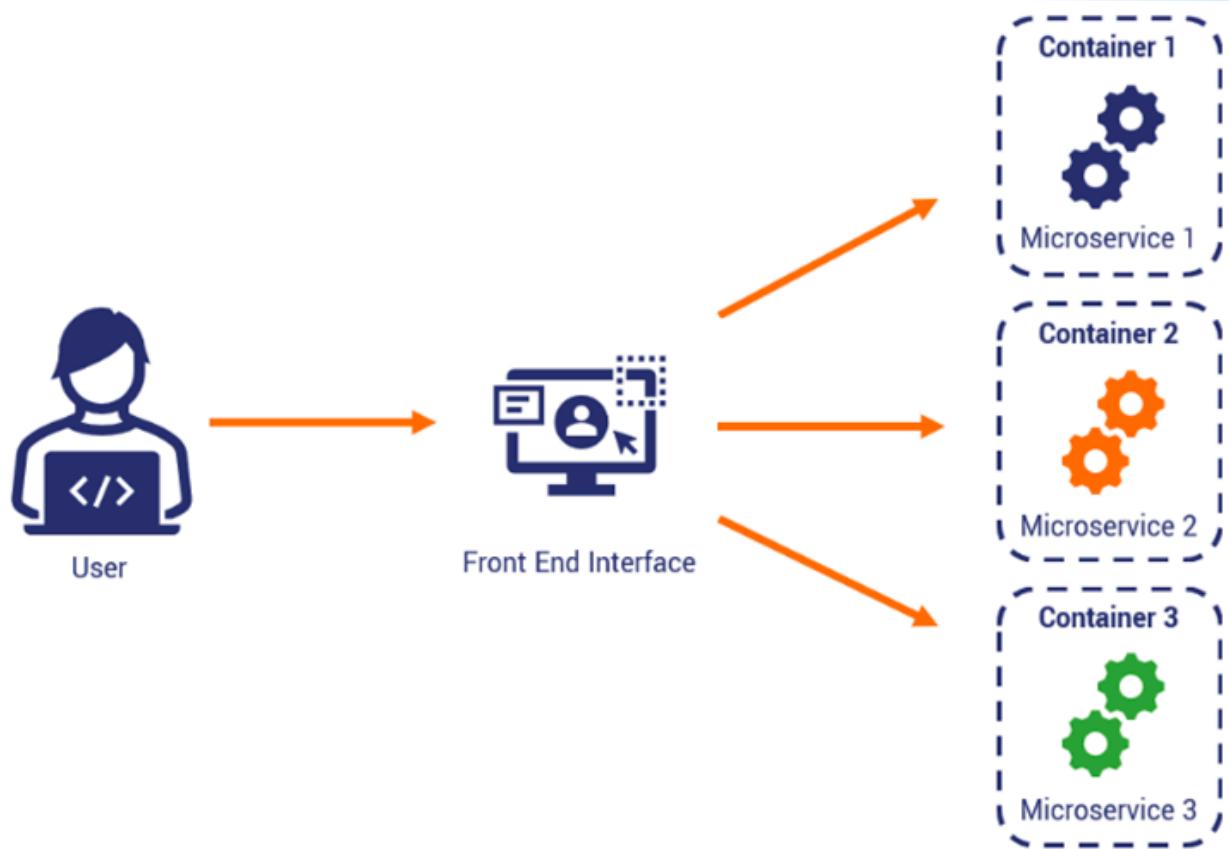
Institute of Technology
Powered by epam

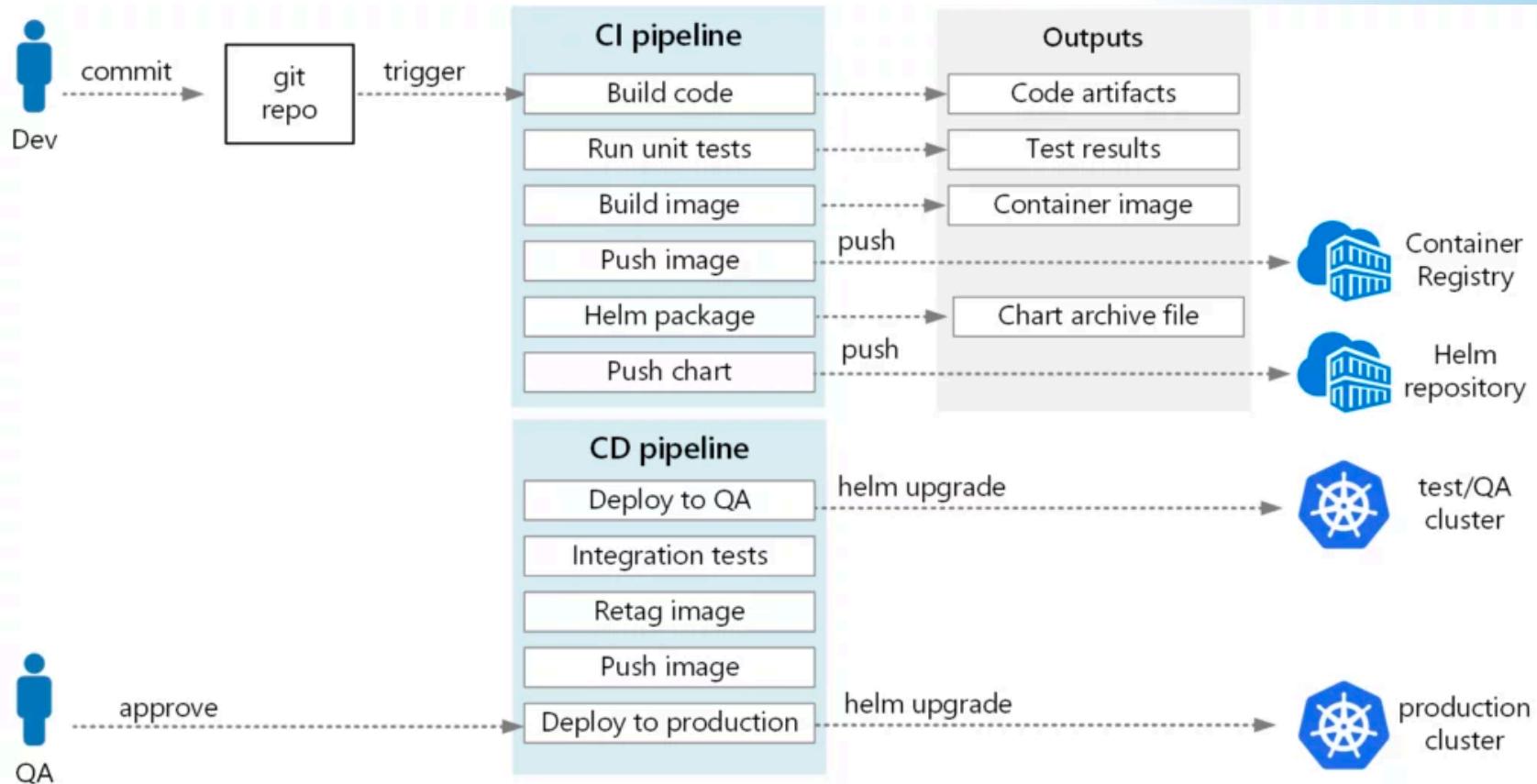


Stages

- 1) Assembling the container image
- 2) Testing
- 3) Image storage and management
- 4) Deployment and Delivery
- 5) Monitoring and Logging
- 6) Automation

Containerized Microservices







EPIC

Institute of Technology
Powered by epam

06

Life without ci/cd

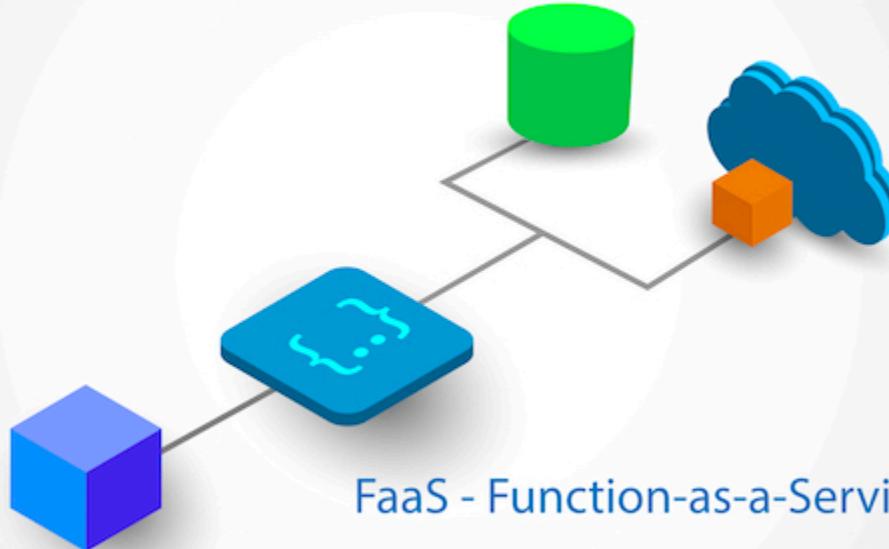


EPIC

Institute of Technology
Powered by epam

FAAS

Serverless



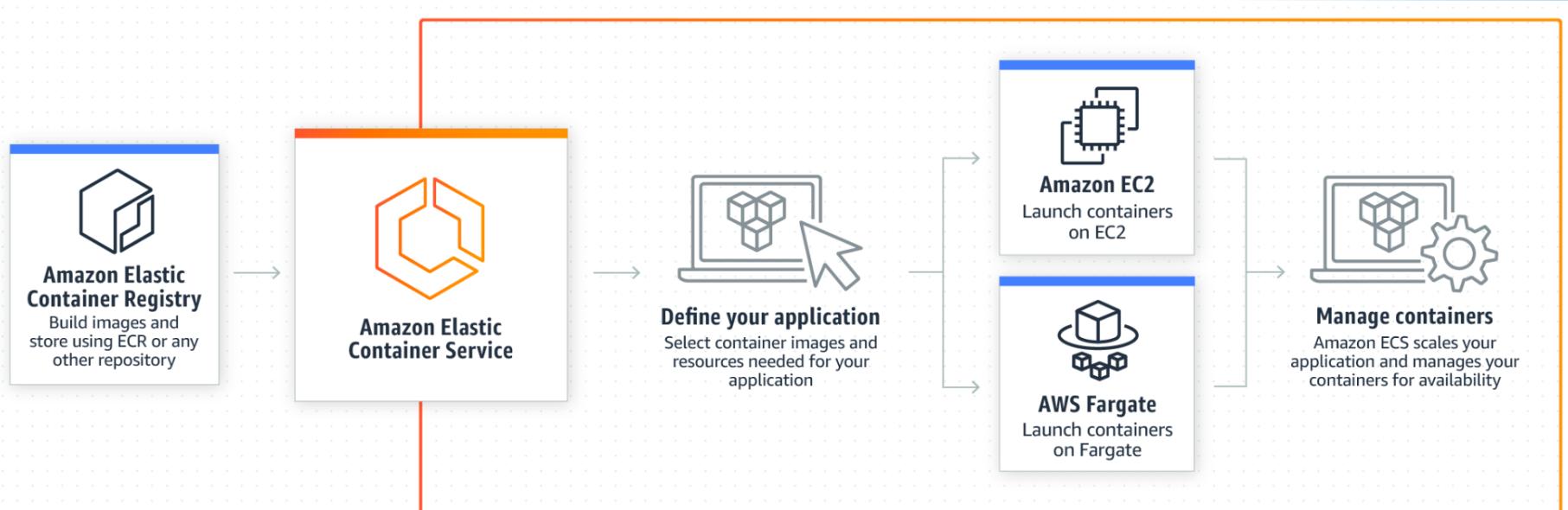
Serverless vs Containers

- 1) Infrastructure management
- 2) Scalability
- 3) Costs
- 4) Response time
- 5) Flexibility



EPIC

Institute of Technology
Powered by epam



**EPIC**Institute of Technology
Powered by **Yandex Cloud**[Services](#) [Solutions](#) [Why Yandex Cloud](#) [Blog](#) [Pricing](#) [Documentation](#) [Contact us](#)

Search



Console

[Serverless solutions](#)[Solutions](#) [Guides](#) [Services](#) [Success stories](#) [Partners](#) [All Solutions](#)

Serverless

Serverless cloud computing models allow you to develop applications, store your data, and configure integration with other platforms without creating VMs or maintaining infrastructure.

Yandex Cloud provides a fault-tolerant and automatically scalable environment for your code and data in the serverless computing ecosystem, taking care of maintenance, security, and continuous operation.

[Talk to an expert](#)

What makes Yandex Cloud the right choice?

Fully managed services
<https://cloud.yandex.com>

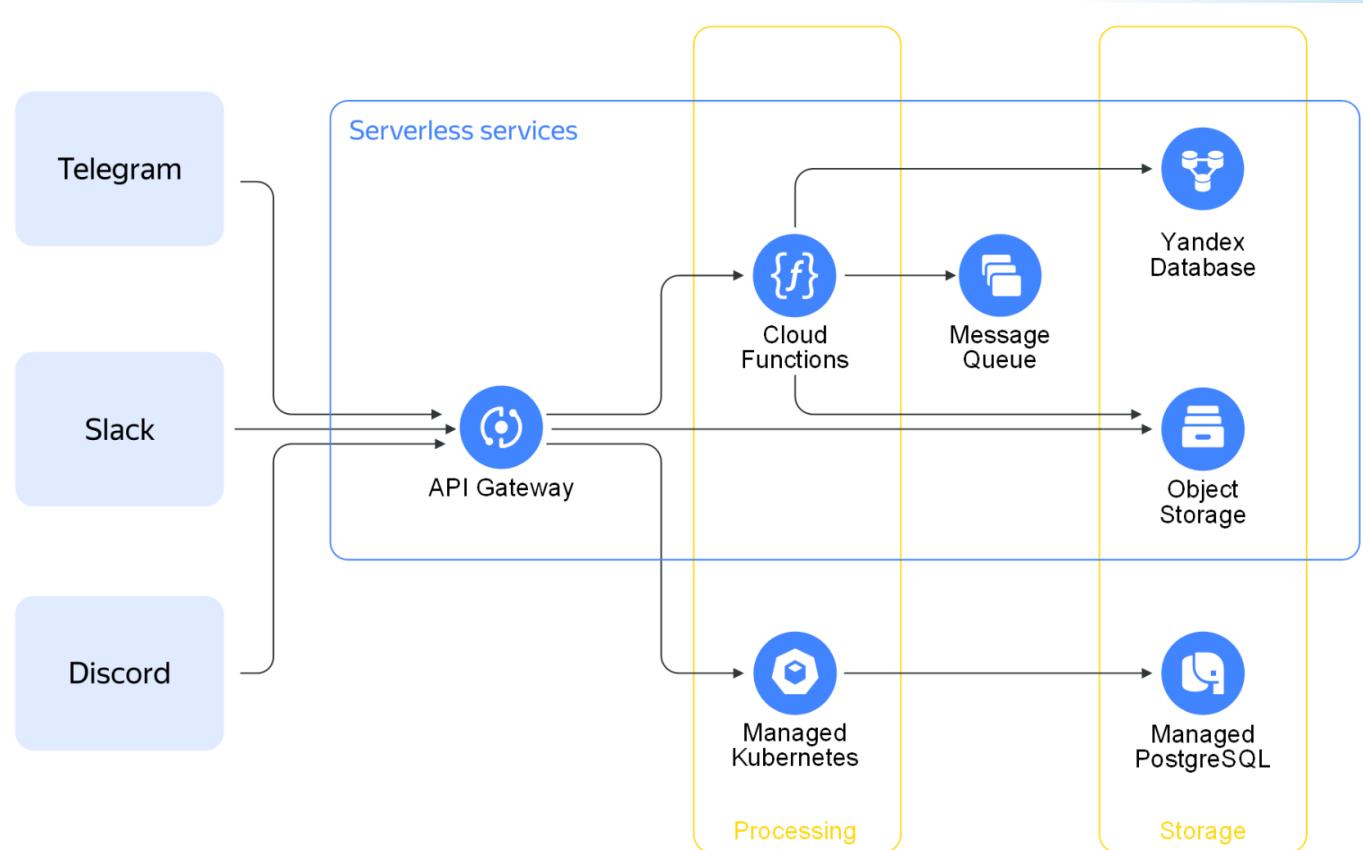
Special terms and conditions: Free tier

Only pay for resources used



EPIC

Institute of Technology
Powered by epam



**EPIC**Institute of Technology
Powered by epam

Cloud Functions

Solutions Q&A

Yandex Cloud Functions

The service enables you to run code as a function in a secure, fault-tolerant, and automatically scalable environment without creating or maintaining VMs.

[Console](#)[Documentation](#)

Serverless computing

Run your applications without creating VMs. We'll take care of the installation, maintenance, and administration of software and runtime environments.

Triggers Preview

You can use triggers to set up integration between Cloud Functions and other services without writing any integration code.

[Learn more >](#)

Flexible scaling

As the number of function calls increases, the service automatically creates additional instances of your function. All functions run in parallel.

Different programming languages

You can write code in popular programming languages, including Python, Node.js, Bash, Go, and PHP. This list is continuously updated.

High availability

The runtime environment is hosted in three availability zones. This ensures availability even if a zone fails.



EPIC

Institute of Technology
Powered by epam

Get started in minutes

Deploy an Edge Function Template

[View All Templates](#)

JWT Authentication

JWT is a standard JSON-based access token used to securely exchange information between two parties before they log in, authenticate, and interact with each other.

Your requests to our API will now require the `JWT-Auth` header.

Don't worry, we've added `JWT-Auth` to all tokens, so when the final API request is made, the token will be passed to the edge, they'll validate it.

```
const handle = (req, res) => {
  const token = req.headers['jwt-auth'];
  if (!token) {
    return res.status(401).json({ error: 'No token provided' });
  }
  // ...
}
```

JWT Authentication

Learn how to do JWT authentication at the edge.



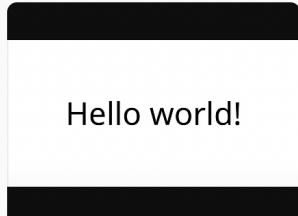
CORS in Edge Functions

Handle CORS at the edge.

```
const cors = require('cors');
const express = require('express');
const app = express();
app.use(cors());
app.get('/', (req, res) => {
  res.json({ message: 'Hello from the edge!' });
});
app.listen(3001, () => {
  console.log(`Hello from the edge!`);
});
```

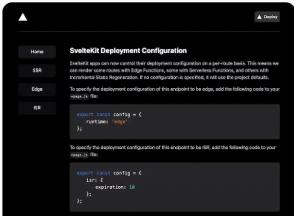
Using Crypto in Edge Middleware and Edge Functions

Learn to utilize the crypto Web APIs at the edge.



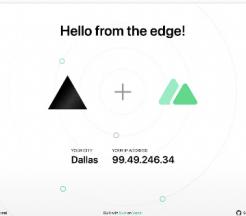
Open Graph Image Generation

Compute and generate dynamic social card images with React components.



SvelteKit Route Config

This template shows how to configure Edge Functions, Serverless Functions, and ISR in SvelteKit applications on a per-route basis.



Nuxt on the Edge

Vue based SSR on the edge, powered by Nuxt 3, Nitro, and Vercel Edge Functions.



EPIC

Institute of Technology
Powered by epam

Serverless Computing	Edge Computing
Serverless computing offers a pay-as-per-usage feature, i.e., one only pays as per the functions that are used.	Edge computing refers to executing functions at the edge of the network where data originates (or as close as possible).
Serverless has a long cold start as it stops dormant functions.	Edge has a lower latency and faster response time.
Serverless is more cost-effective as almost no infrastructure cost is required.	Edge computing depends on the device for its computational power and hence is not very cost-effective.
Serverless offers easy scalability and development support to shorten the time to market.	There are almost no scalability provisions in Edge computing.
It has the capability to do in-depth analysis and data-driven workload.	Edge is limited to the computational power of the devices.
Most of the web and mobile applications and services run on this. E.g. Payment Gateway, Photo processing on a site, Live streaming a game, etc.	Most IoT devices and sensors have this architecture to ensure faster response time and quick decision-making. E.g. Self-driving cars, Sensors on solar panels, motion detection cameras, etc



	Serverless Functions	Edge Functions
Release phase	GA	GA
Cold starts	Yes	No
Regions	Single <small>i</small>	Deployed globally, or in a specified region
Runtime	Node.js , Go, Ruby, Python	Edge Runtime
Node.js APIs	Yes	Limited
Max size	50MB	Hobby: 1MB, Pro: 2MB, Ent: 4MB
Max execution time	Hobby: 10s, Pro: 60s, Ent: 900s	30s (to begin returning a response)
Max memory	1024MB <small>i</small>	128MB
Max environment variable size	64KB	64KB
Max request body size	4.5MB	4MB
Geolocation data	Yes	Yes
Access request headers	Yes	Yes
Cache responses	Yes	Yes



EPIC

Institute of Technology
Powered by epam

07

Podman



EPIC

Institute of Technology
Powered by epam



podman

Features Get Started Community Development Blog Documentation GitHub ☰

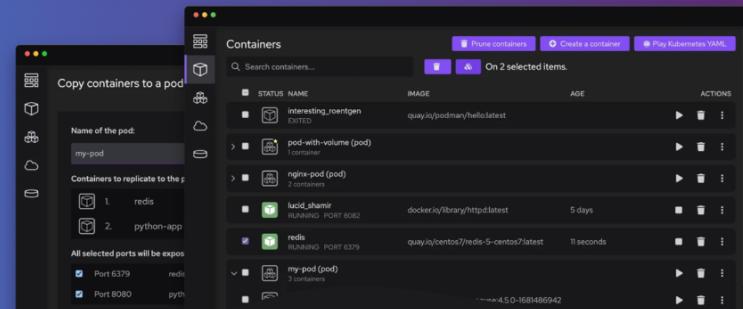
The best free & open source container tools

Manage containers, pods, and images with Podman. Seamlessly work with containers and Kubernetes from your local environment.

[Get Started](#)

[Download ▾](#)

Latest stable Podman [5.1.1](#) - Latest stable Podman Desktop [1.10.2](#) - [Apache License 2.0](#)





EPIC

Institute of Technology
Powered by epam

Get started with Podman Desktop



Welcome to Podman Desktop v1.0.1!

Podman desktop supports many container engines and orchestrators, such as:

 Podman	 Docker	 Lima
 OpenShift Local	 Kind	 Developer Sandbox

Configure these and more under [Settings](#).

Telemetry: Help Red Hat improve Podman Desktop by allowing anonymous usage data to be collected. [Read our privacy statement](#)

You can always modify this preference later in [Settings > Preferences](#)

[Go to Podman Desktop](#)



EPIC

Institute of Technology
Powered by epam

```
└── podman machine init
  Downloading VM image: fedora-coreos-38.20230514.2.0-qemu.aarch64.qcow2.xz: done
  Extracting compressed file
  Image resized.
  Machine init complete
  To start your machine run:

      podman machine start

└── podman machine start
  Starting machine "podman-machine-default"
  Waiting for VM ...
  Mounting volume... /Users:/Users
  Mounting volume... /private:/private
  Mounting volume... /var/folders:/var/folders

This machine is currently configured in rootless mode. If your containers
require root permissions (e.g. ports < 1024), or if you run into compatibility
issues with non-podman clients, you can switch using the following command:

      podman machine set --rootful

API forwarding listening on: /Users/glebodin/.local/share/containers/podman/machine/qemu/podman.sock

The system helper service is not installed; the default Docker API socket
address can't be used by podman. If you would like to install it run the
following commands:

      sudo /opt/homebrew/Cellar/podman/4.5.0/bin/podman-mac-helper install
      podman machine stop; podman machine start

You can still connect Docker API clients by setting DOCKER_HOST using the
following command in your terminal session:

      export DOCKER_HOST='unix:///Users/glebodin/.local/share/containers/podman/machine/qemu/podman.sock'

Machine "podman-machine-default" started successfully
```

**EPIC**Institute of Technology
Powered by epam

```
[/] podman search httpd --filter=is-official
```

NAME	DESCRIPTION
docker.io/library/httpd	The Apache HTTP Server Project

```
[/] podman pull docker.io/library/httpd
```

Trying to pull docker.io/library/httpd:latest...

Getting image source signatures

Copying blob sha256:eeb4169f3e8d4be64783603f97f19023b836627cf589f053fe48821692dfd81

Copying blob sha256:d981f2c20c93e1c57a46cd87bc5b9a554be5323072a0d0ab4b354aabbd237bbcf

Copying blob sha256:faa1c382eb6325c33bb2adb532d9d3042d30b0c3a13ae5b0dd77e641281e233a

Copying blob sha256:9c93073f2b50bbc731a20466388f0505f4b1a003f2320d70072383acadb6c4ec

Copying blob sha256:c�푸fb8eef81ed17a36a60ff93f08c3882f894b68406819865b3cb72ce7ada5cbb

Copying config sha256:d59e5e963fe998098421293b7b29d9dd9aba9a38d09da13b61a3ef14dd8650ed

Writing manifest to image destination

Storing signatures

d59e5e963fe998098421293b7b29d9dd9aba9a38d09da13b61a3ef14dd8650ed

```
[/] podman images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.io/library/httpd	latest	d59e5e963fe9	5 days ago	143 MB



EPIC

Institute of Technology
Powered by epam

```
└── []   ~
    podman run -dt -p 8080:80/tcp docker.io/library/httpd
    cf045f8cee392490b790be12000f42941f195f3d37b95cbf85a3388daea91a0d

└── []   ~
    podman ps
    CONTAINER ID  IMAGE                      COMMAND           CREATED          STATUS          PORTS          NAMES
    cf045f8cee39  docker.io/library/httpd:latest  httpd-foreground  6 seconds ago  Up 6 seconds  0.0.0.0:8080->80/tcp  serene_vaughan

└── []   ~
    curl http://localhost:8080
    <html><body><h1>It works!</h1></body></html>
```

**EPIC**Institute of Technology
Powered by epam

```
[INT ✘ 22:09:05 ⓘ] podman logs cf045f8cee39
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 10.88.0.2. Set the 'ServerName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 10.88.0.2. Set the 'ServerName' directive globally to suppress this message
[Sun May 28 19:07:54.301089 2023] [mpm_event:notice] [pid 1:tid 281472874418192] AH00489: Apache/2.4.57 (Unix) configured -- resuming normal operations
[Sun May 28 19:07:54.301343 2023] [core:notice] [pid 1:tid 281472874418192] AH00094: Command line: 'httpd -D FOREGROUND'
10.88.0.2 - - [28/May/2023:19:08:05 +0000] "GET / HTTP/1.1" 200 45
[✓ 22:09:12 ⓘ] podman stop cf045f8cee39
cf045f8cee39
[✓ 22:09:24 ⓘ] podman ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
```



EPIC

Institute of Technology
Powered by epam

```
└─ podman search example_for_epum
NAME                                     DESCRIPTION
docker.io/glebodin/example_for_epum

└─ podman pull example_for_epum
Resolving "example_for_epum" using unqualified-search registries (/etc/containers/registries.conf.d/999-podman-machine.conf)
Trying to pull docker.io/library/example_for_epum:latest...
Error: initializing source docker://example_for_epum:latest: reading manifest latest in docker.io/library/example_for_epum: requested access to the resource is denied

└─ podman pull glebodin/example_for_epum
Resolving "glebodin/example_for_epum" using unqualified-search registries (/etc/containers/registries.conf.d/999-podman-machine.conf)
Trying to pull docker.io/glebodin/example_for_epum:latest...
Getting image source signatures
Copying blob sha256:a06a975b738391d72254582d5a2537ac0961581af50ec5f7a76df622e01f96b9
Copying blob sha256:00ef1dbac9624e93cf5a7ac63c8df26bc6520d2536fb411e8506d69e563985f
Copying blob sha256:2095682ba1c8ea76d23358f9f8685d48abce3f28fe2f04e491eab9fe089b2a4d
Copying blob sha256:c9ee6c3d04736819ef47a350cc13ae8292f6ea46725afba7f0e0707189d34e4f
Copying blob sha256:7eebaae1fb823d65757608f32f812befaa833ae61d9d1c221fb018524fc35854
Copying blob sha256:0e613d4fdc607e77800d0c446289a4306dc31f34e69e900820e76761136ab819
Copying blob sha256:d600888a689e22f15e8a515b258458fc4a018728acf71a71b89331c4a54f7f2
Copying blob sha256:eed64c99a75ff8ef9a34022570e5fd7ec643e0cbc1f3cc72370e676c3940d4e
Copying blob sha256:ed03a2e49b38cd6e4b43c2945e3ab2f6995043abeb3588664986885f803875b0
Copying blob sha256:8de2bd8a54e1bd909ac5c6c35bed80c960a0bd837cb543daa4be3853d77320a1
Copying blob sha256:d2e7056ef04df17f946e53f9a020f04289f0ac01b53daec004eb709244fc4
Copying blob sha256:86d1b02fb038fe6b37e32821fc5e03c436f9be3e3369ba4068c38aedb3cae17a
Copying blob sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1
Copying blob sha256:86d1b02fb038fe6b37e32821fc5e03c436f9be3e3369ba4068c38aedb3cae17a
Copying blob sha256:28c03c3cf4e8b7558b5dcf94cc4a96d0e36b2ca289f9d6d2a9ee616721a86590
Copying config sha256:dc87f8101af486a62e72be093389e2e70999db1172caee5932629326c5ad1aff3
Writing manifest to image destination
Storing signatures
WARNING: image platform (linux/arm) does not match the expected platform (linux/arm64)
dc87f8101af486a62e72be093389e2e70999db1172caee5932629326c5ad1aff3
```



EPIC

Institute of Technology
Powered by epam

A screenshot of a Mac OS X desktop environment. At the top, there's a dark menu bar with standard Apple, Home, and User icons. To the right of the menu bar is a red notification badge with the number '7' and an 'x' button. Next to it is the current date and time, '22:18:35', followed by a battery icon. The main area is a terminal window with a black background and white text. It shows the command 'podman run -dt -p 8080:80/tcp glebodin/example_for_epum' being entered. Below the command, a warning message is displayed: 'WARNING: image platform (linux/arm) does not match the expected platform (linux/arm64)' followed by a long image ID.



EPIC

Institute of Technology
Powered by epam



ARCHITECTURE

ROOT ACCESS

MULTIPLATFORM

LOCAL REPOSITORY

CLIENT-SERVER (Daemon)

REQUIRED

LINUX, MAC &
WINDOWS

/var/lib/docker

FORK-EXEC (Daemonless)

ROOTLESS

LINUX
ONLY

/var/lib/containers



EPIC

Institute of Technology
Powered by 

That's All Folks!