API Development

PyRVA - January 12th 2022

Adam Emery

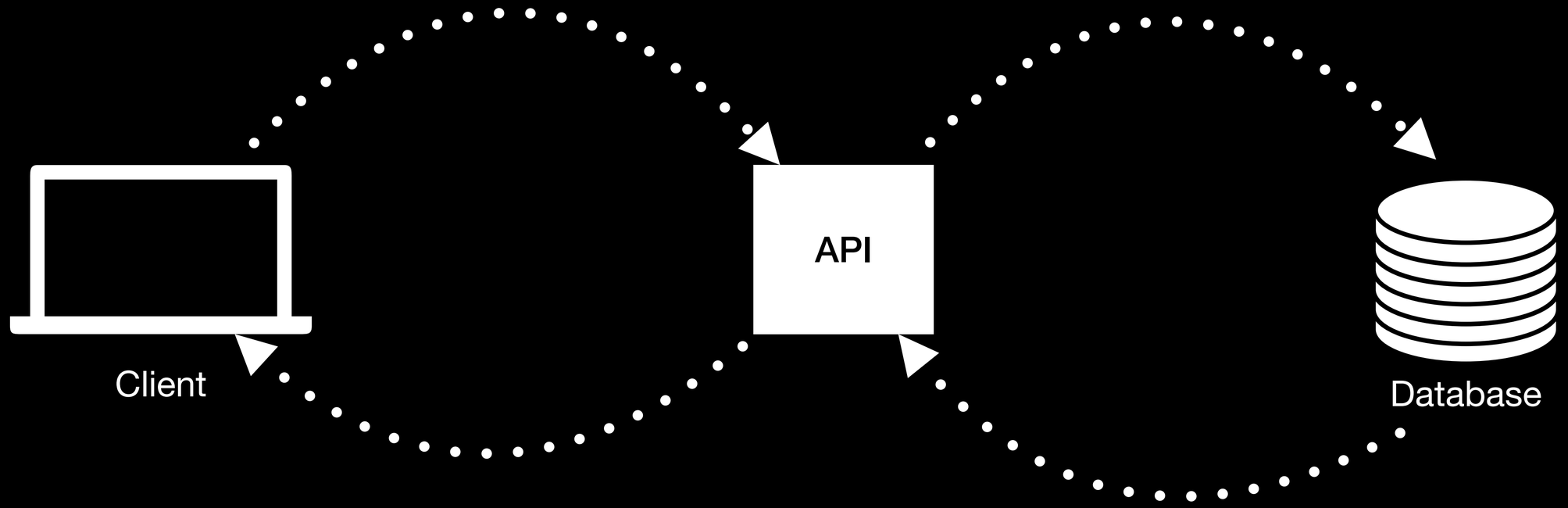
Agenda

- Who am I?
- What is an API?
- Python API development with FastAPI
- Docker and Microservies
- Beyond basic APIs

Who am I?

What is an API?

Application Programming Interface



REST API?

Representational State Transfer? CRUD?

1. Client-Server: The client and server act independently.
2. Stateless: The server does not record the state of the client.
3. Cacheable: The server marks whether data is cacheable.
4. Uniform Interface: The client and server interact in a uniform and predictable way. An important aspect of this is that the server exposes resources.
5. Layered System: The application behaves the same regardless of any intermediaries between the client and server.

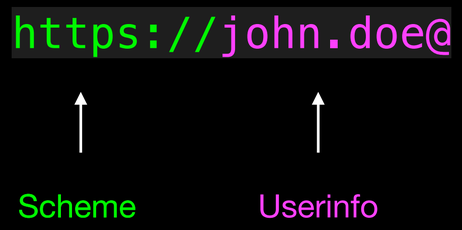
Http and json

GET http://localhost:8000/items/5

```
{
  "id": 5,
  "name": "The Fifth Element"
}
```

URI

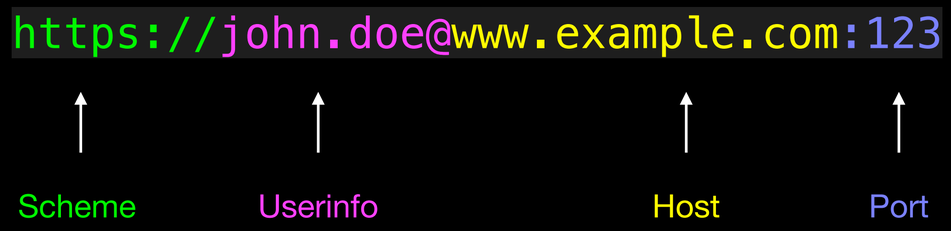
URI



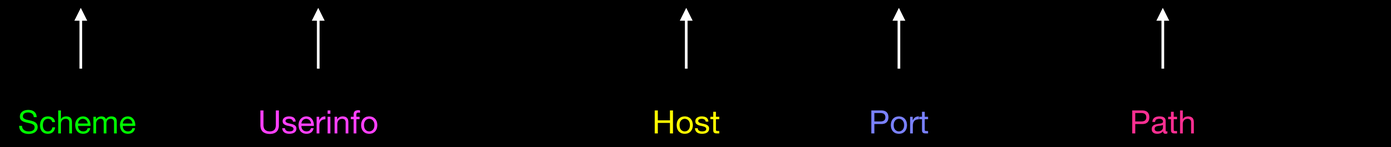
URI

```
https://john.doe@www.example.com
```

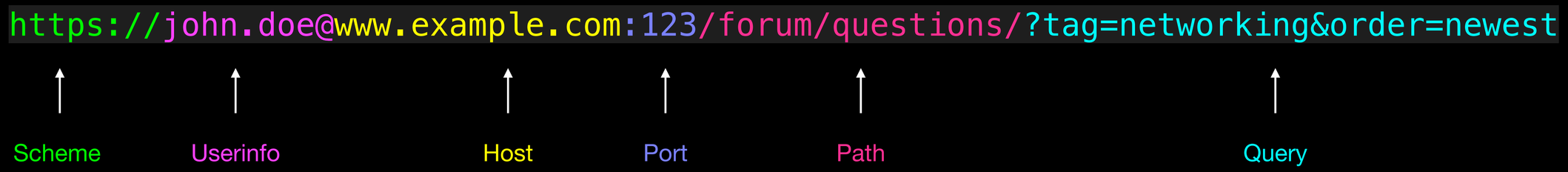
URI


https://john.doe@www.example.com:123
Scheme
Userinfo
Host
Port

URI


https://john.doe@www.example.com:123/forum/questions/
Scheme
Userinfo
Host
Port
Path

URI


https://john.doe@www.example.com:123/forum/questions/?tag=networking&order=newest
Scheme
Userinfo
Host
Port
Path
Query

URI


https://john.doe@www.example.com:123/forum/questions/?tag=networking&order=newest#top
Scheme
Userinfo
Host
Port
Path
Query
Fragment

HTTP Methods

- GET - request data from resource
- POST - send data to create or update resource
- PUT - send data to create or update resource idempotent
- HEAD - request data from resource without body
- DELETE - deletes a resource
- PATCH - send data to modify a resource
- OPTIONS - describes communication options

HTTP Methods

- GET - request data from resource
- POST - send data to create or update resource
- PUT - send data to create or update resource idempotent
- HEAD - request data from resource without body
- DELETE - deletes a resource
- PATCH - send data to modify a resource
- OPTIONS - describes communication options

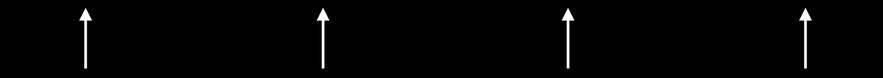
Google API Standards

https://cloud.google.com/apis/design/

Standard Method	HTTP Mapping	HTTP Request Body	HTTP Response Body
List	GET <collection URL>	N/A	Resource* list
Get	GET <resource URL>	N/A	Resource*
Create	POST <collection URL>	Resource	Resource*
Update	PUT or PATCH <resource URL>	Resource	Resource*
Delete	DELETE <resource URL>	N/A	google.protobuf.Empty**

Collections and Resources

/libraries/{lib_id}/books/{book_id}



Collection Resource Collection Resource

Custom Methods

POST https://service.name/v1/books:checkout

GET https://service.name/v1/books:search

Status Codes

HTTP	gRPC	Description
200	OK	No error.
201	Created	No error.
400	INVALID_ARGUMENT	Client specified an invalid argument. Check error message and error details for more information.
401	UNAUTHENTICATED	Request not authenticated due to missing, invalid, or expired OAuth token.
403	PERMISSION_DENIED	Client does not have sufficient permission. This can happen because the OAuth token does not have the right scopes, the client doesn't have permission, or the API has not been enabled.
404	NOT_FOUND	A specified resource is not found.
500	INTERNAL	Internal server error. Typically a server bug.
503	UNAVAILABLE	Service unavailable. Typically the server is down.

Python API Framworks

- Django
- Flask
- FastAPI
- Falcon
- Bottle
- Tornado
- Starlette

FastAPI

Features

- Fast, Async
- OpenAPI
- Validation built-in
- Auto Documentation
- Pydantic data modeling

FastAPI

How fast?

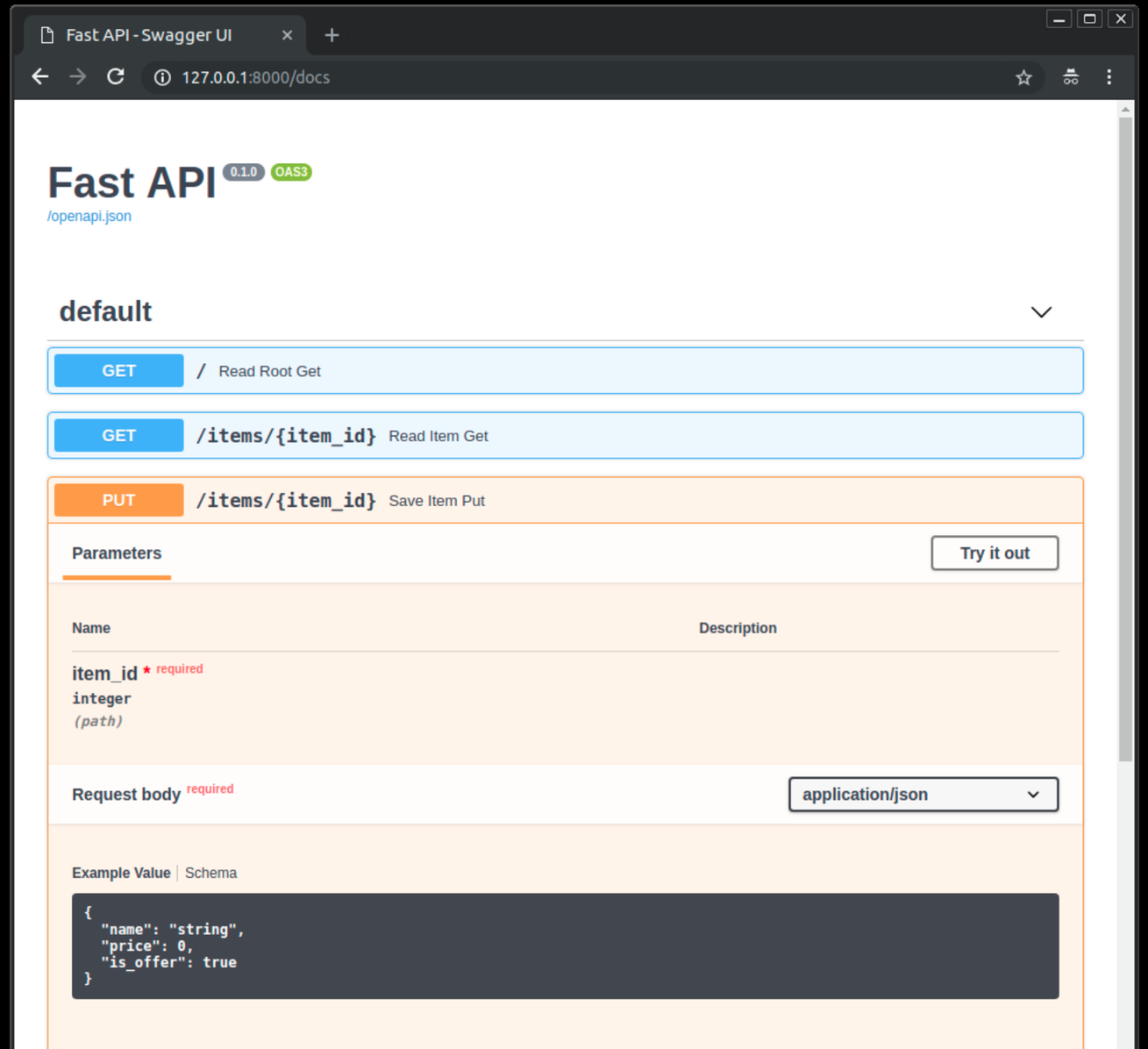
Rnk	Framework	Best performance (higher is better)	
1	uvicorn	71,763	100.0% (17.0%)
2	starlette	65,396	91.1% (15.4%)
3	bottle-raw	35,793	49.9% (8.5%)
4	aiohttp-pg-raw	34,037	47.4% (8.0%)
5	api_hour	27,747	38.7% (6.6%)
6	flask-raw	23,078	32.2% (5.5%)
7	api_hour-mysql	21,304	29.7% (5.0%)
8	flask-pypy2-raw	20,929	29.2% (4.9%)
9	aiohttp	19,528	27.2% (4.6%)
10	weppy-pypy2	18,396	25.6% (4.3%)
11	morepath	18,041	25.1% (4.3%)
12	tornado-pypy2	16,804	23.4% (4.0%)
13	weppy-py3	16,533	23.0% (3.9%)
14	weppy-nginx-uwsgi	16,283	22.7% (3.8%)
15	pyramid-py2	16,253	22.6% (3.8%)
16	pyramid	15,284	21.3% (3.6%)
17	bottle-pypy2	13,670	19.0% (3.2%)
18	django-postgresql	12,059	16.8% (2.8%)
19	bottle	11,857	16.5% (2.8%)
20	turbogears	11,798	16.4% (2.8%)
21	tornado-py3	11,293	15.7% (2.7%)
22	web2py-optimized	10,815	15.1% (2.6%)
23	django	10,687	14.9% (2.5%)
24	flask	10,376	14.5% (2.5%)

https://www.techempower.com/benchmarks/#section=data-r17&hw=ph&test=fortune&l=zijzen-1

Code

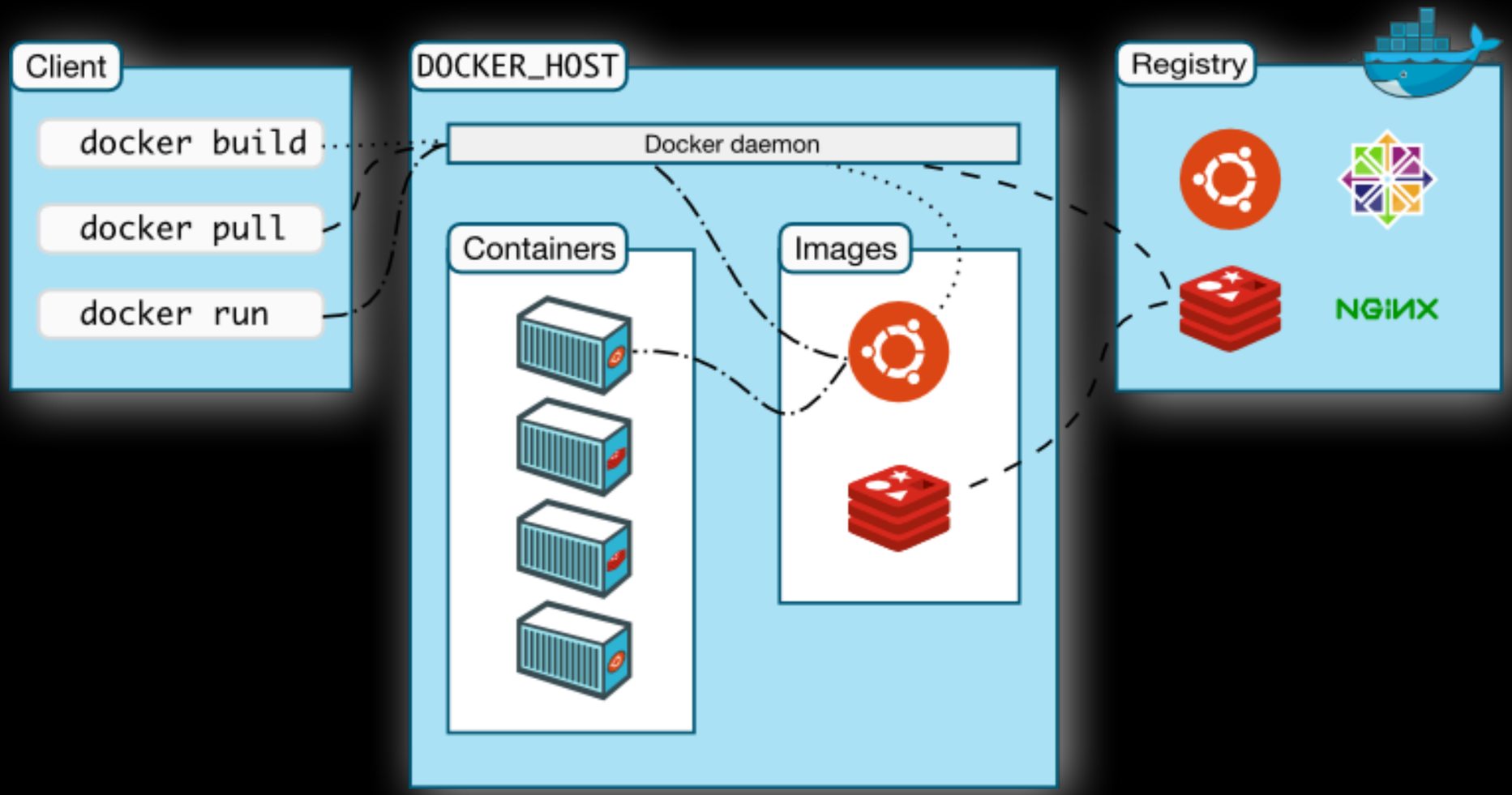
```python
from fastapi import FastAPI

app = FastAPI()


@app.get("/")
def read_root():
    return {"Hello": "World"}
```

Documentation

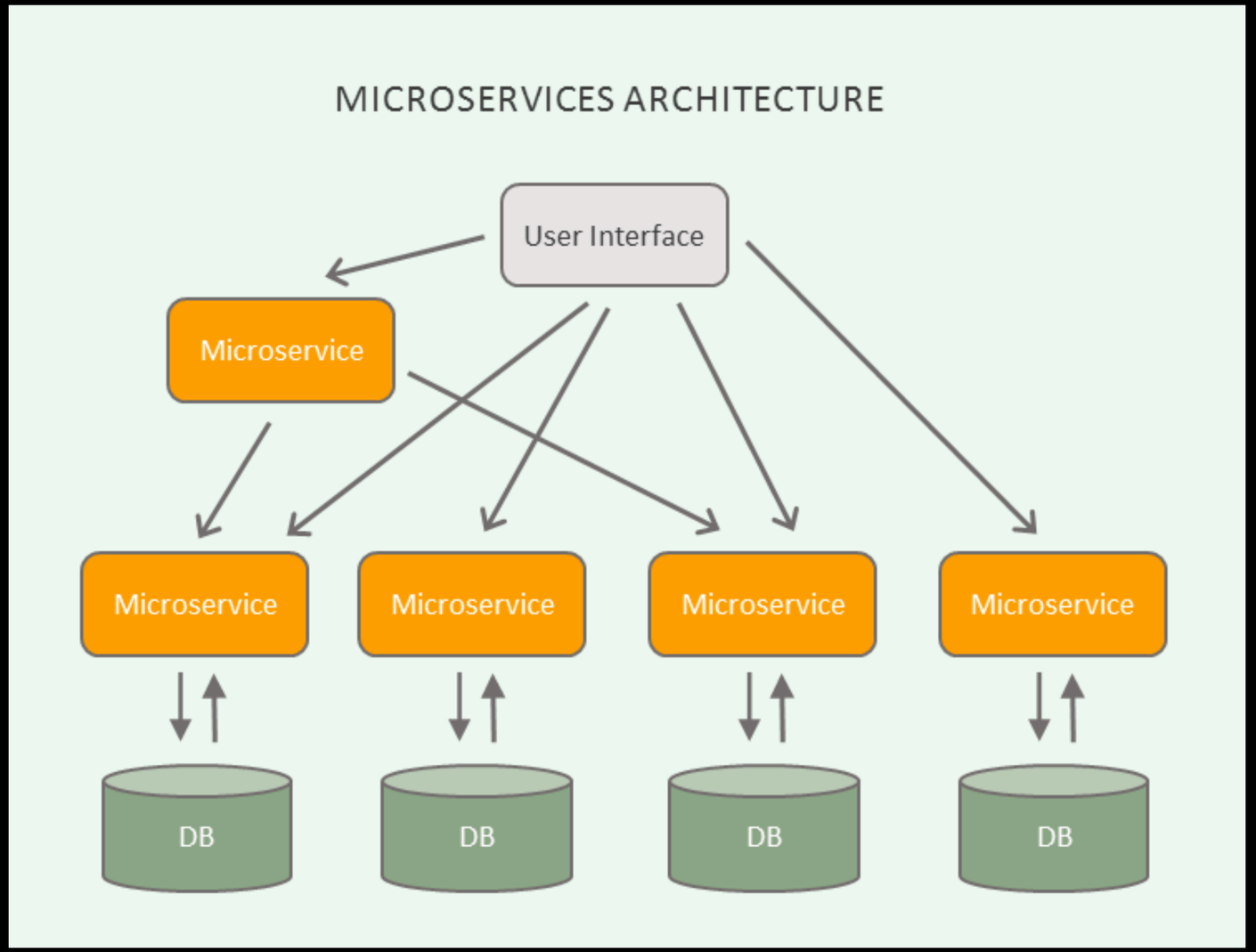


Demo

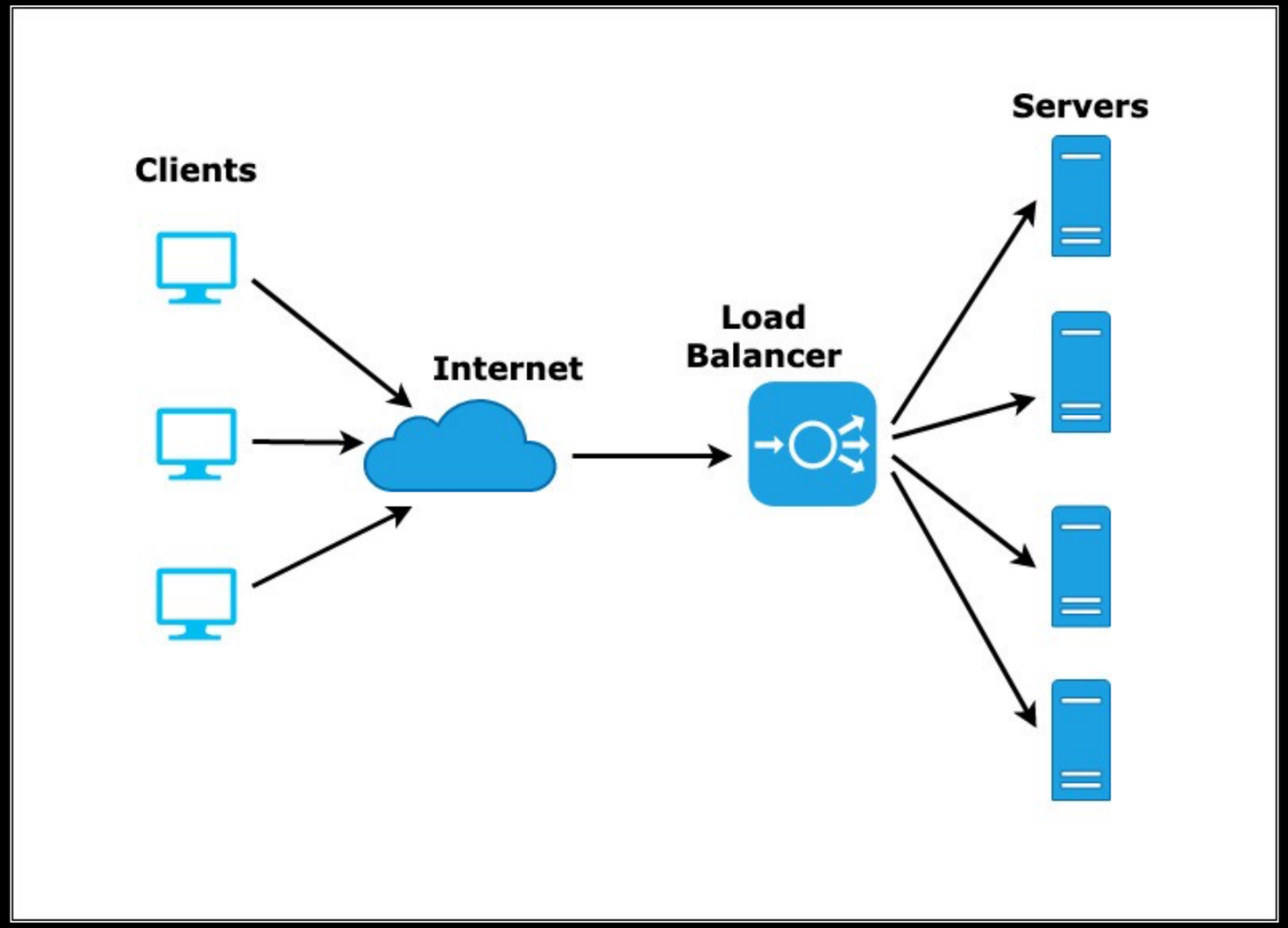
Docker


Client
docker build
docker pull
docker run
DOCKER_HOST
Docker daemon
Containers
Images
Registry
NGINX

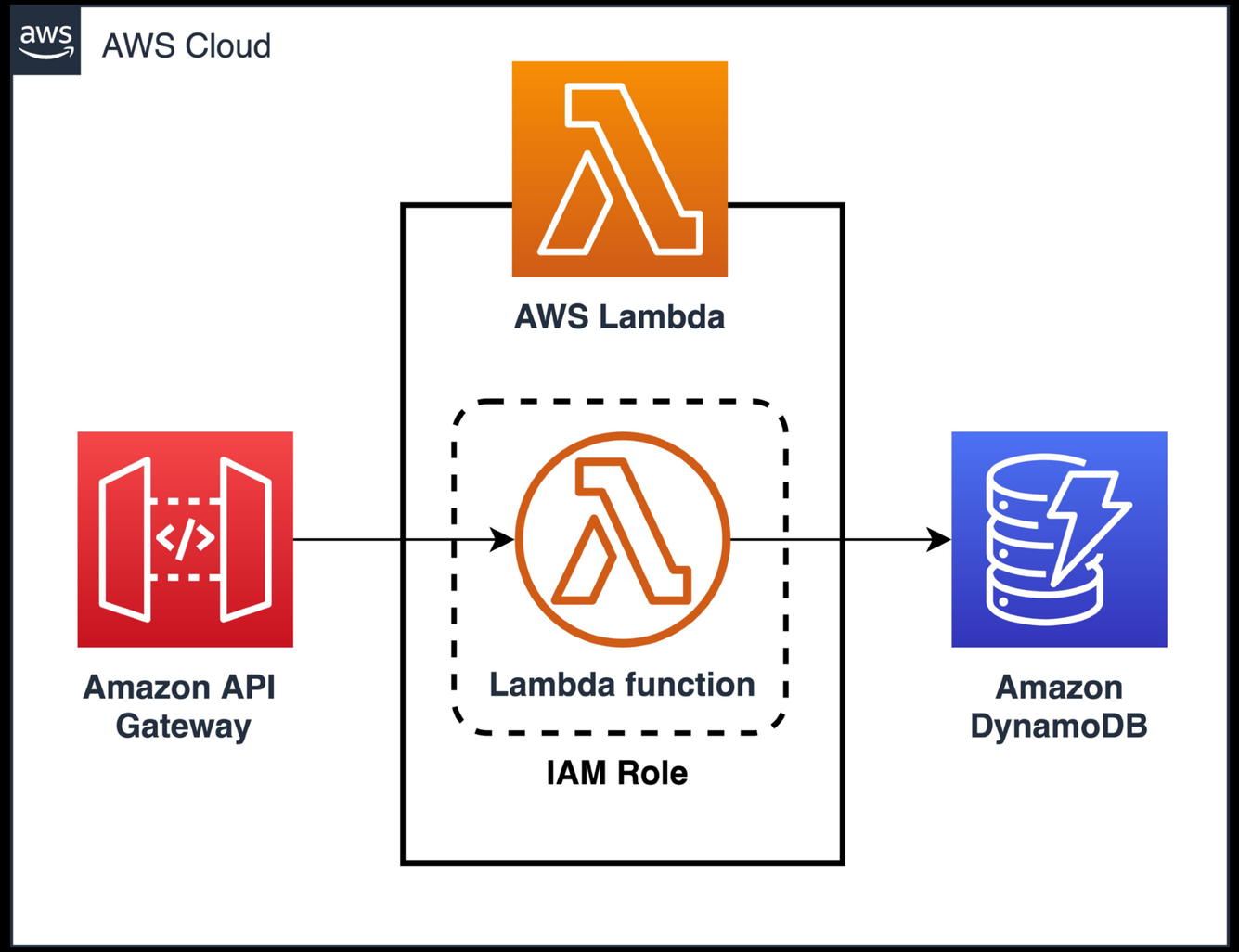
Microservices



Scaling



AWS Serverless



Web Sockets and GraphQL

Demo