

HUGE

Hello

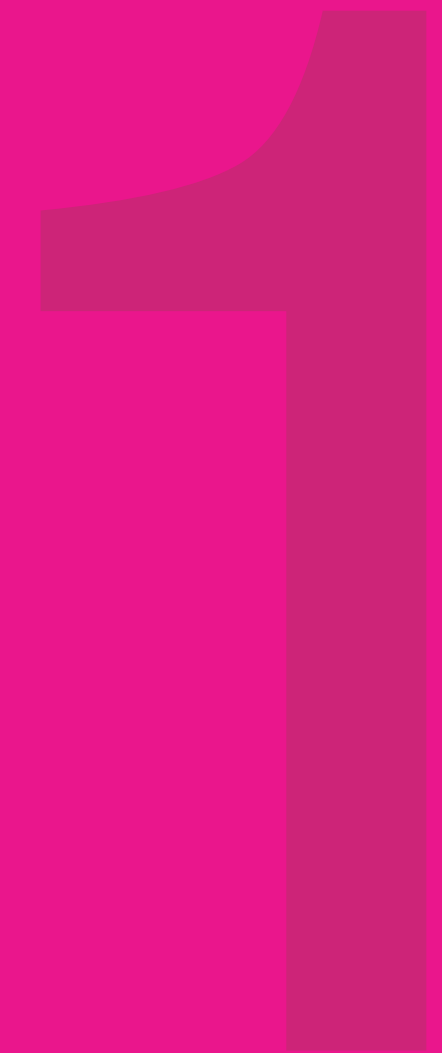
PyDay Cali

Going Serverless with Python

1. What is Serverless?
2. Can I go Serverless?
3. Going Serverless with Python
4. Conclusions

Agenda.

What is Serverless?



What is Serverless?

**“Serverless is
a deployment architecture
where servers are not explicitly
provisioned by the deployer.”**

Taken from <https://www.fullstackpython.com/serverless.html>

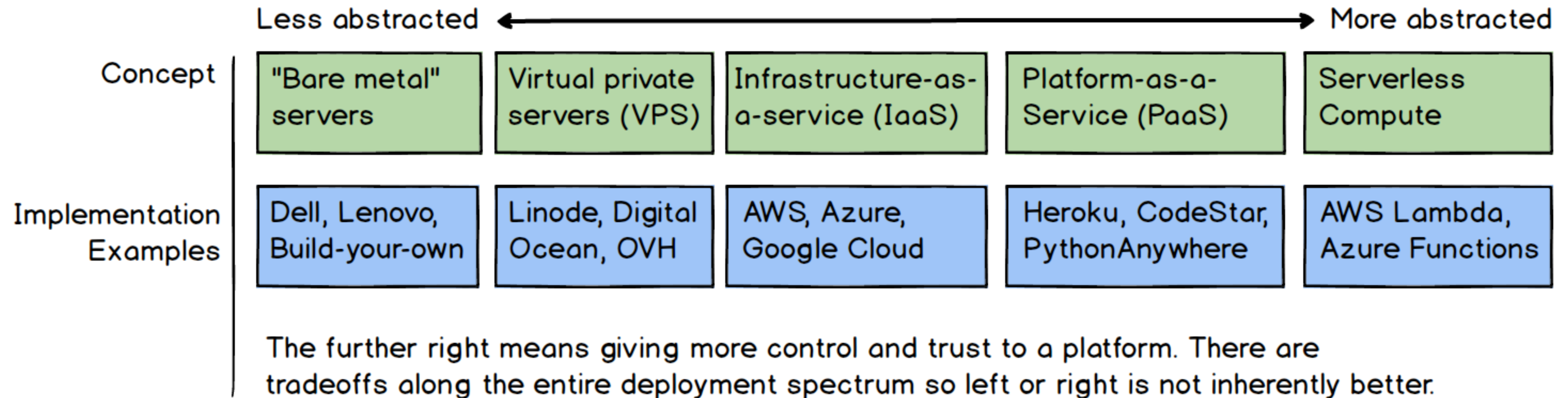
What is Serverless?

“Code is instead executed based on developer-defined events that are triggered...”

Taken from <https://www.fullstackpython.com/serverless.html>

Deployment Abstractions

Deployment Abstractions



Taken from <https://www.fullstackpython.com/serverless.html>

Advantages

- 1. Not server managing.**
- 2. Paying for the compute you use.**
- 3. Horizontal scaling out of the box.**

- 1. Speedy, simpler, singular.**
- 2. Think in concurrency, not in total requests.**
- 3. Share nothing.**
- 4. Assume no hardware affinity.**
- 5. Design for failures and duplicates.**

Can I go
Serverless?

2

No suitable apps for Serverless

- 1. Apps with web-sockets don't fit here.**
- 2. Apps serving statics.**

- 1. Apps with spikes or periods of low activity.**
- 2. Stateless apps (REST APIs, GraphQL API)**
- 3. Workers (sending an email, uploading a file, push a notification)**

Neep help?

<https://servers.lol/>

RESTfull microservices

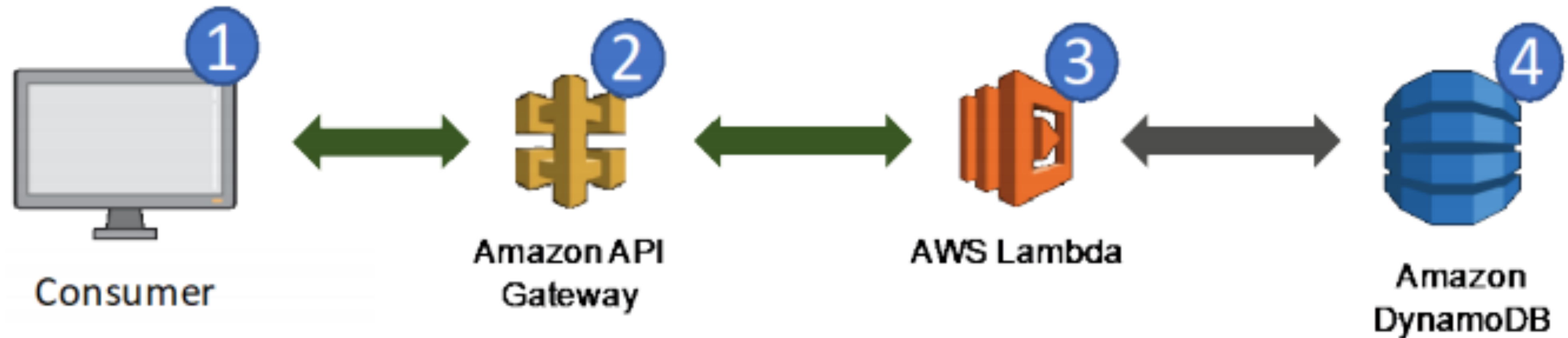


Figure 1: Reference architecture for RESTful microservices

Taken from <https://d1.awsstatic.com/whitepapers/architecture/AWS-Serverless-Applications-Lens.pdf>

Mobile Backend

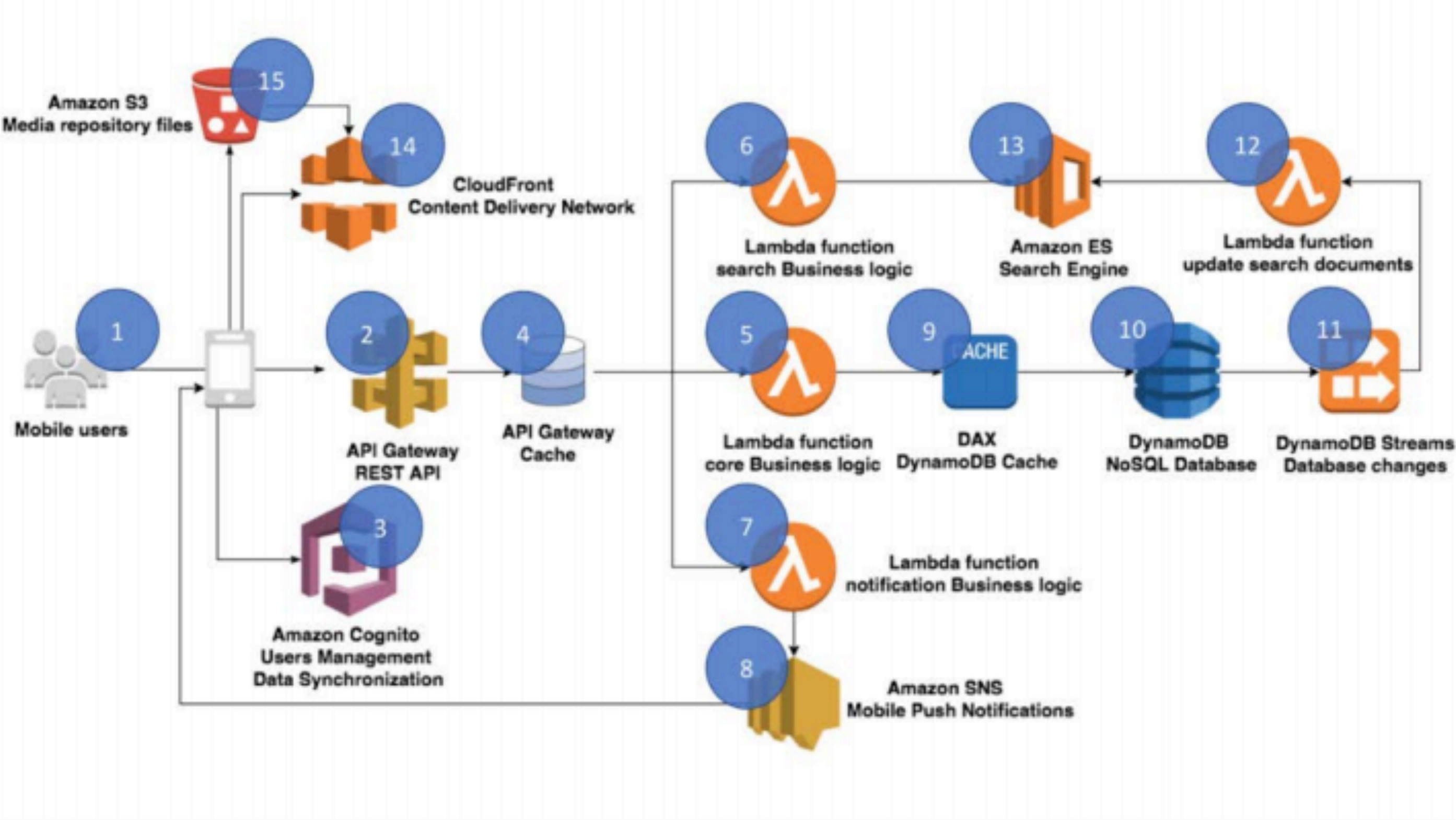


Figure 2: Reference architecture for a mobile backend

Taken from <https://d1.awsstatic.com/whitepapers/architecture/AWS-Serverless-Applications-Lens.pdf>

Stream Processing

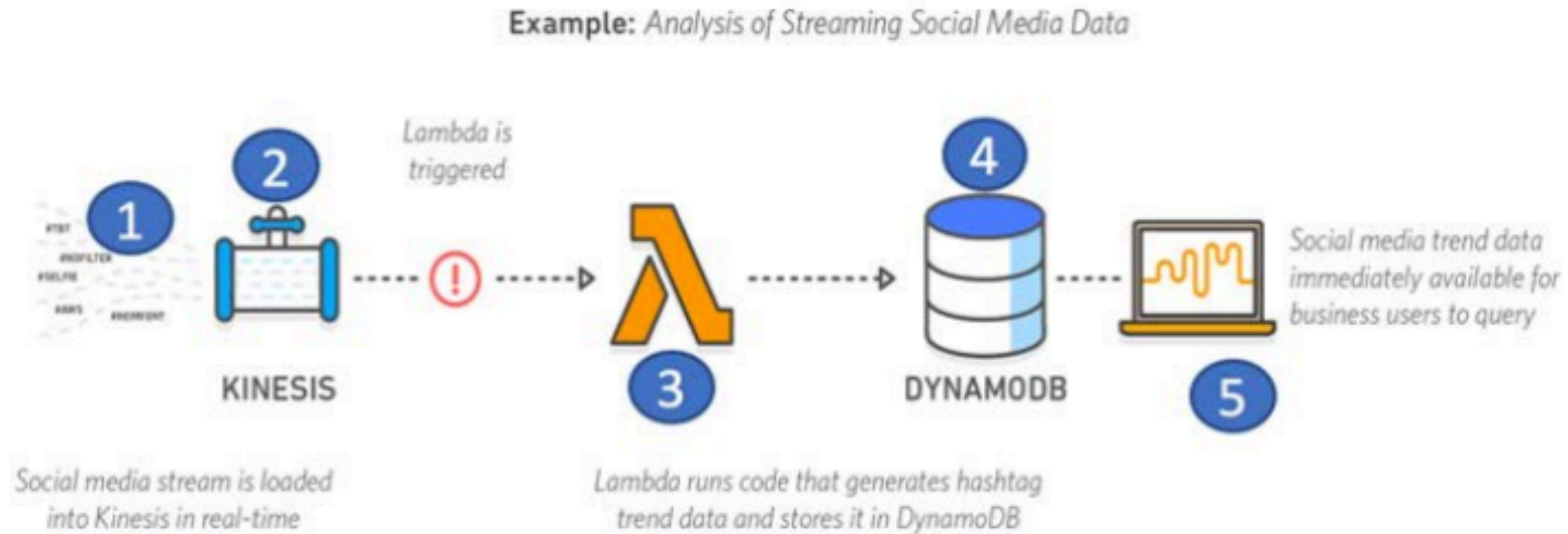


Figure 3: Reference architecture for stream processing

Taken from <https://d1.awsstatic.com/whitepapers/architecture/AWS-Serverless-Applications-Lens.pdf>

Web App

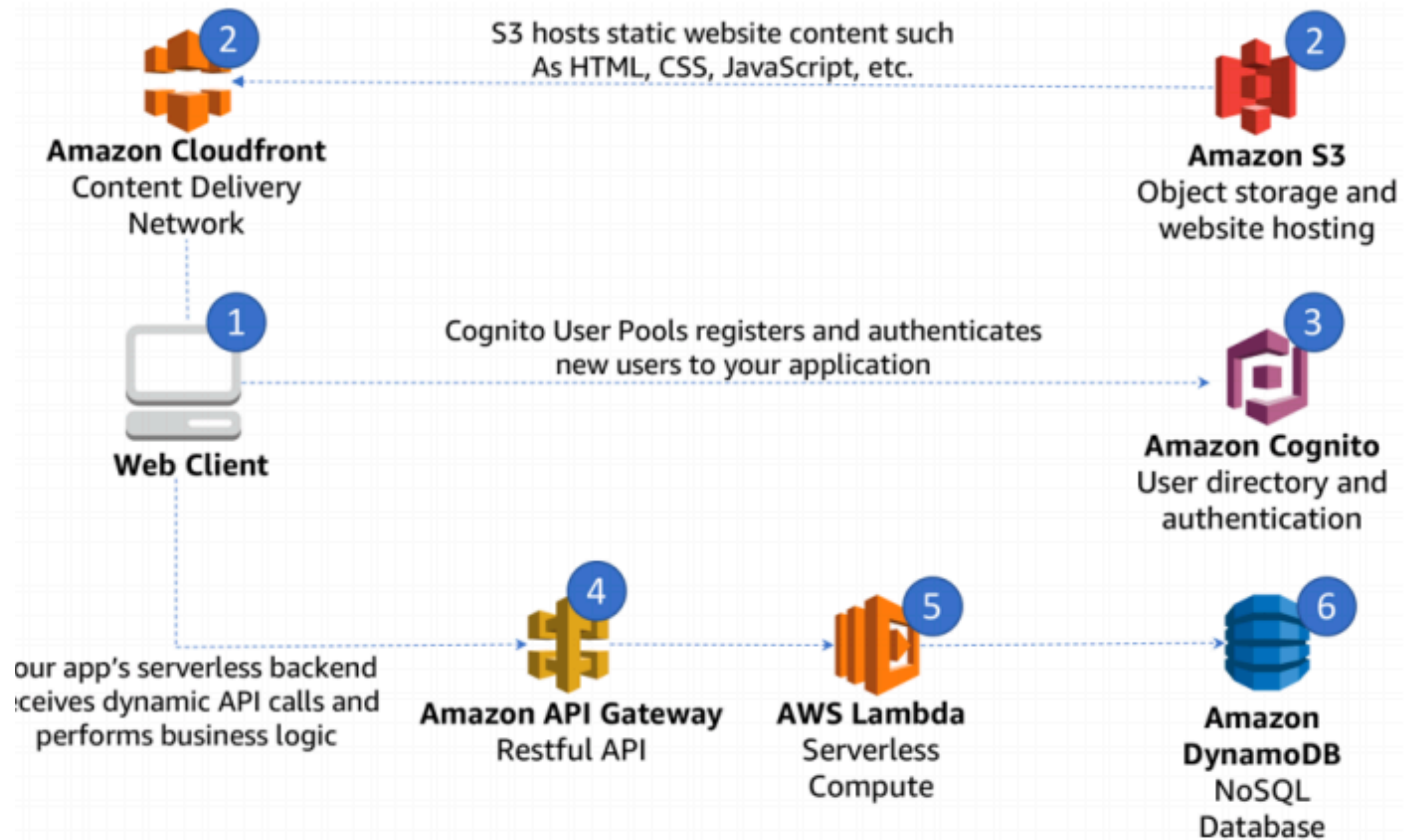


Figure 4: Reference architecture for a web application

Taken from <https://d1.awsstatic.com/whitepapers/architecture/AWS-Serverless-Applications-Lens.pdf>

- 1. Vendor lock-in.**
- 2. Cold starts.**
- 3. Vendor restrictions (time, size, etc)**

Going Servers with Python

3

Providers

- 1. AWS Lambda.**
- 2. Azure Functions.**
- 3. Google Cloud Functions.**

On-premise solutions

1. **OpenFaaS.**
2. **OpenWhisk.**
3. **Fission.io.**

1. Manual Creation
2. Serverless.
3. Zappa.
4. Chalice.

Manual Creation


Functions (1) ↻

🔍

Filter by tags and attributes or search by keyword

	Function name
<input type="radio"/>	say_hello

Manual Creation in AWS

Function Apps


Location:

All locations

NAME ▾

hugecaferakidata

Manual Creation in Azure (no Python)

Cloud Functions

☰

Filter functions

<input type="checkbox"/>	Name ^	Region
<input type="checkbox"/>) say_hello	us-central1

Manual Creation in GCP

Serverless framework

```
sudo npm install -g serverless
```

```
serverless config credentials --provider aws --key <ACCESS KEY ID> --secret <SECRET KEY>
```

```
serverless create --template aws-python3 --name hello-world
```

```
export AWS_PROFILE="serverless"
```

```
serverless deploy
```

```
serverless deploy -v
```

4

Conclusions

- 1. Serverless is another architectural approach for apps.**
- 2. No all apps fit on this approach.**
- 3. Serverless allows to save money if it is used wisely.**
- 4. It autoscale.**

References

1. <https://www.fullstackpython.com/serverless.html>
2. <https://read.iopipe.com/the-right-way-to-do-serverless-in-python-e99535574454>
3. <https://d1.awsstatic.com/whitepapers/architecture/AWS-Serverless-Applications-Lens.pdf>
4. <https://servers.lol/>
5. <https://www.iopipe.com/what-is-amazon-lambda/>
6. <https://read.iopipe.com/understanding-aws-lambda-coldstarts-49350662ab9e>
7. <https://thenewstack.io/serverless-architecture-five-design-patterns/>
8. <https://serverless.com/blog/flask-python-rest-api-serverless-lambda-dynamodb/>
9. <https://martinfowler.com/articles/serverless.html>
10. <https://code.roygreenfeld.com/comparisons/zappa-vs-serverless-framework.html>
11. <https://hackernoon.com/aws-lambda-serverless-framework-python-part-1-a-step-by-step-hello-world-4182202aba4a>

HUGE

Done.

Going Serverless with Python