

# How to Start with AWS

The Builder's Way

Avinash Dalvi

Head of technology, Nushift Technologies

AWS User group Leader, Bengaluru

AWS Community Builder

Everyone says 'Learn  
AWS'... but

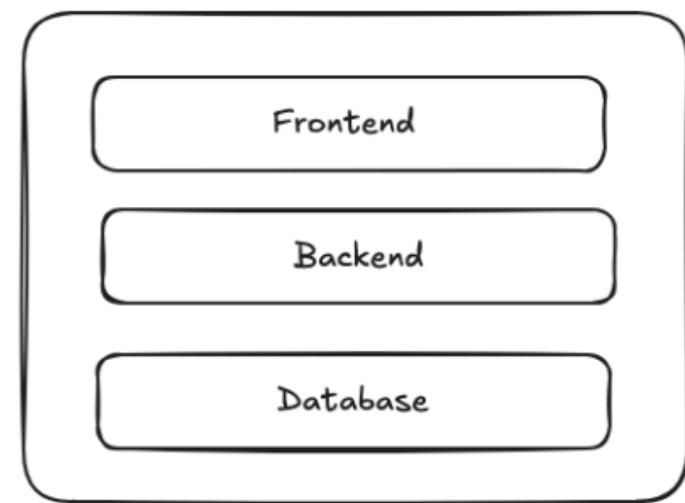




“You don’t start by learning services. You start by understanding how your app grows.”



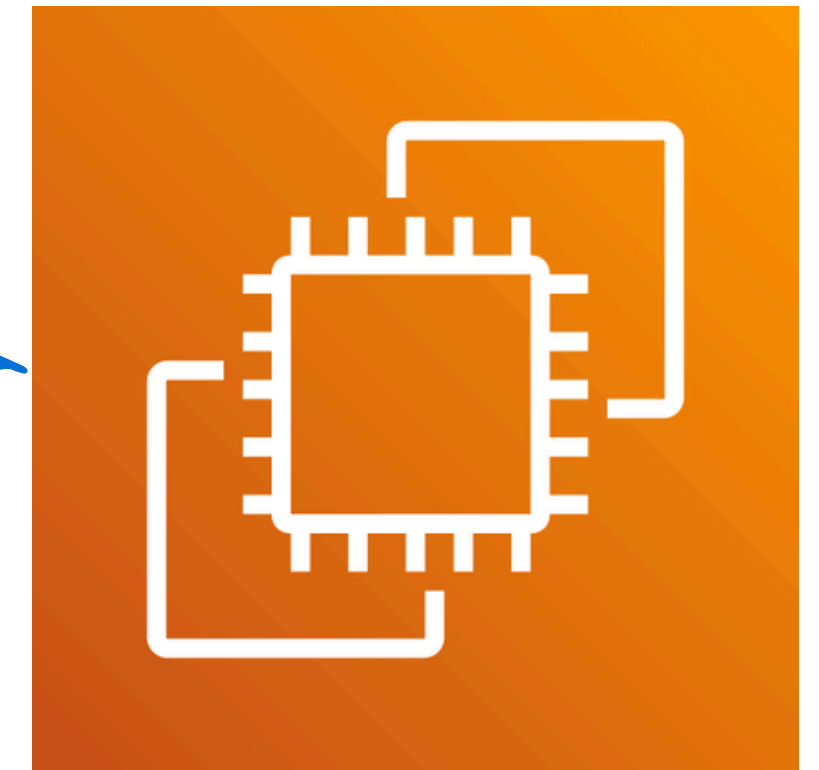
# Stage 1: Everything on One Machine



One codebase

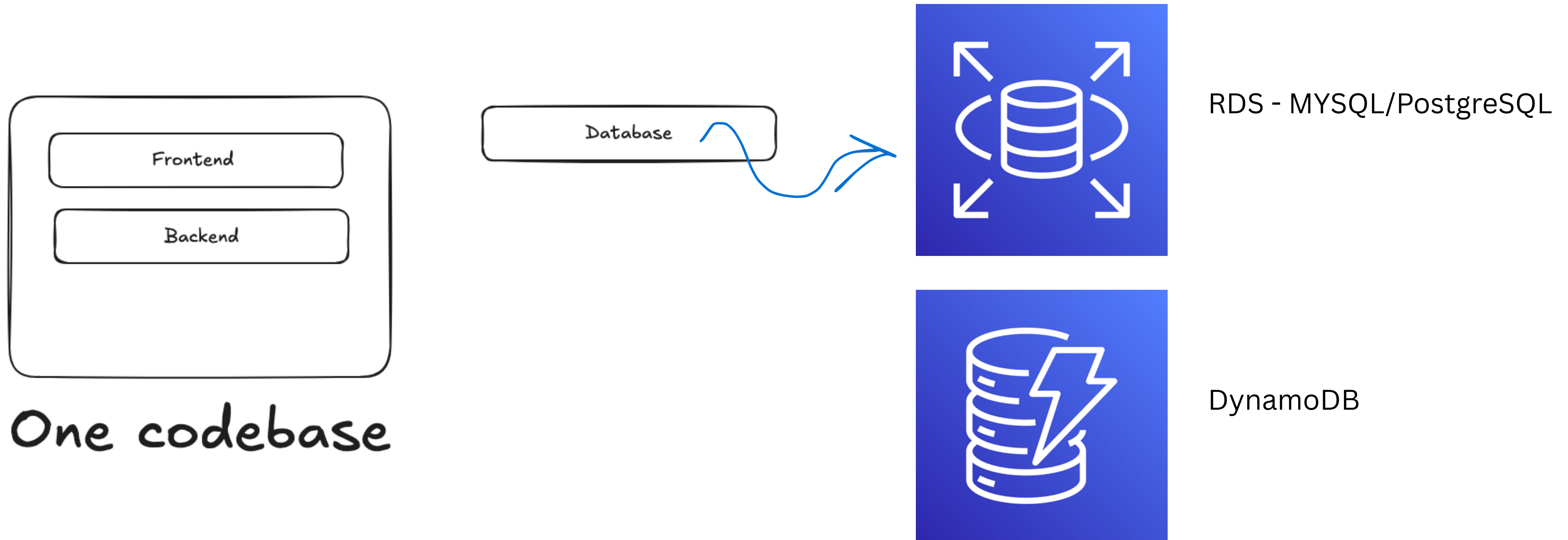


Local server or locally running

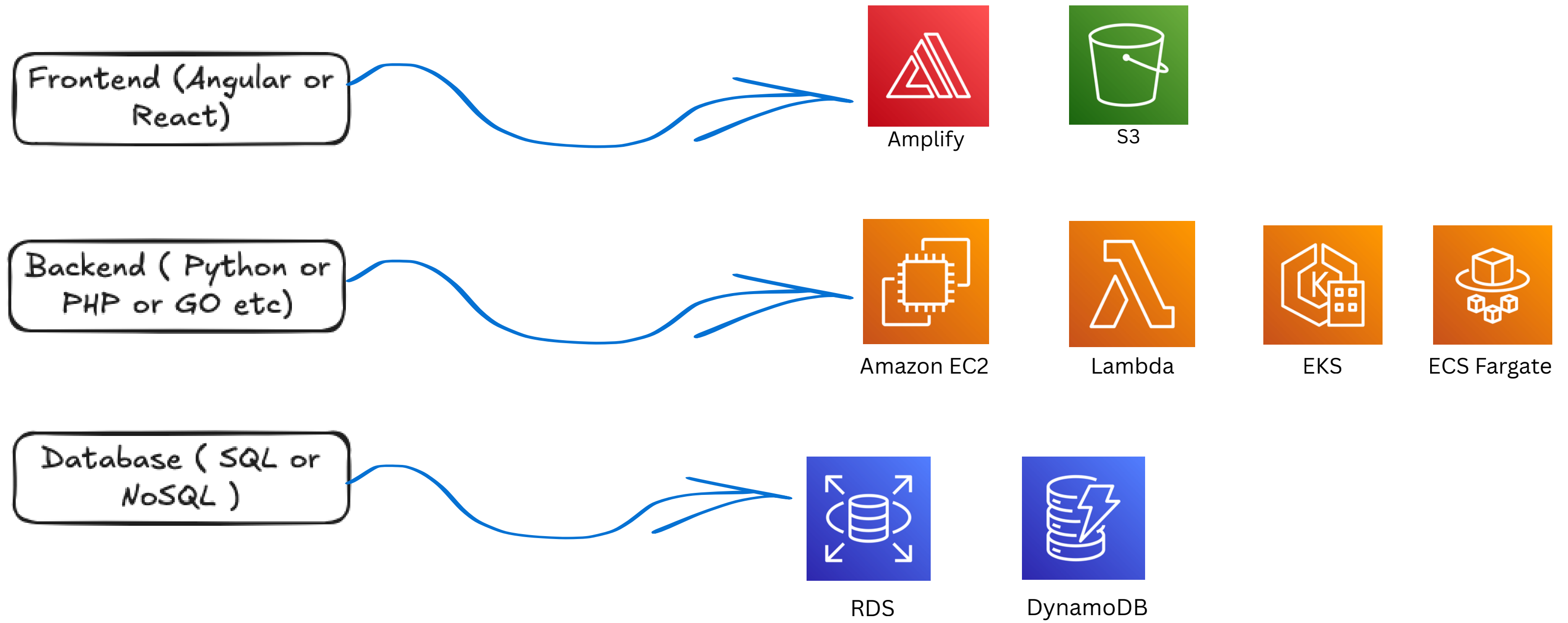


Amazon EC2

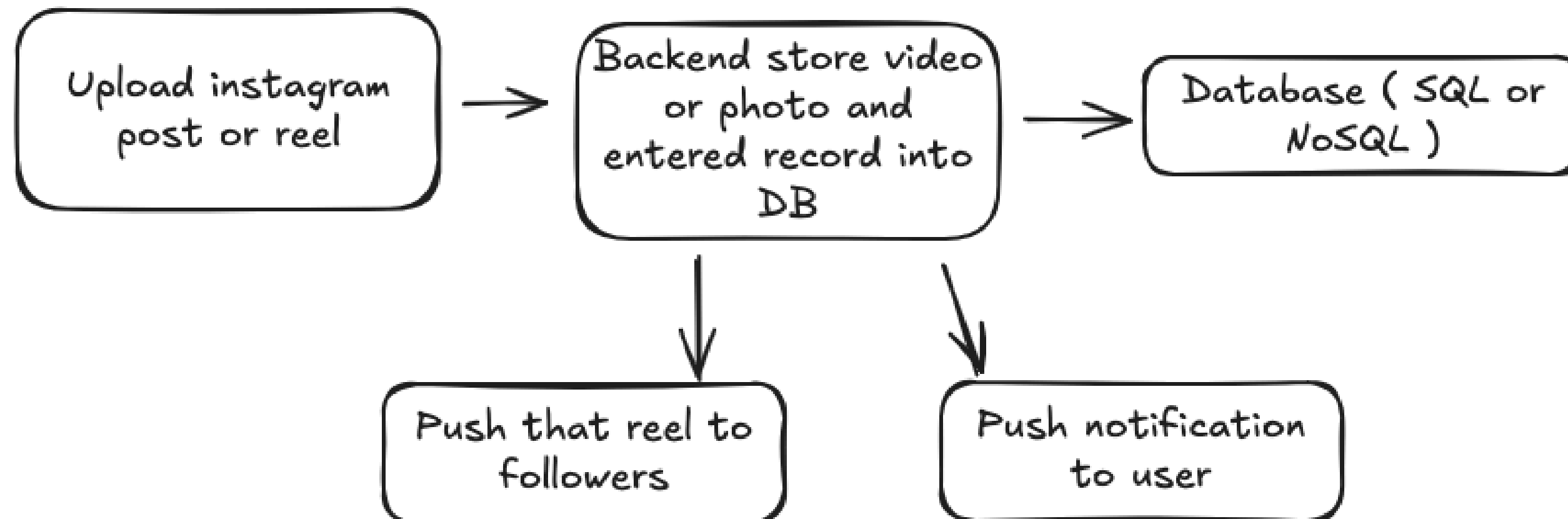
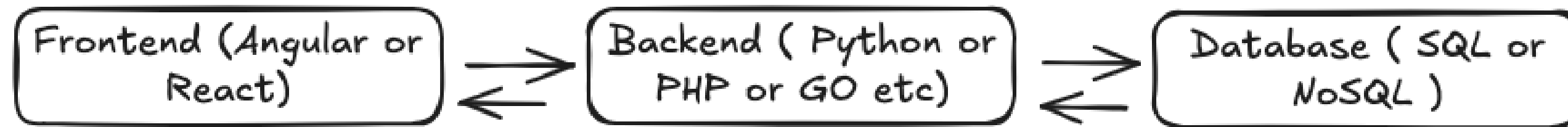
# Stage 2: Separate the Database



# Stage 3: Frontend, Backend, Database

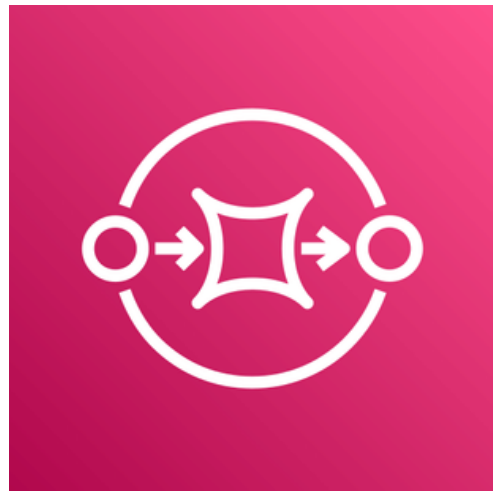


# Stage 4: When Your App Starts Talking to Itself





# How AWS Handles Events



Simple Queue  
Service



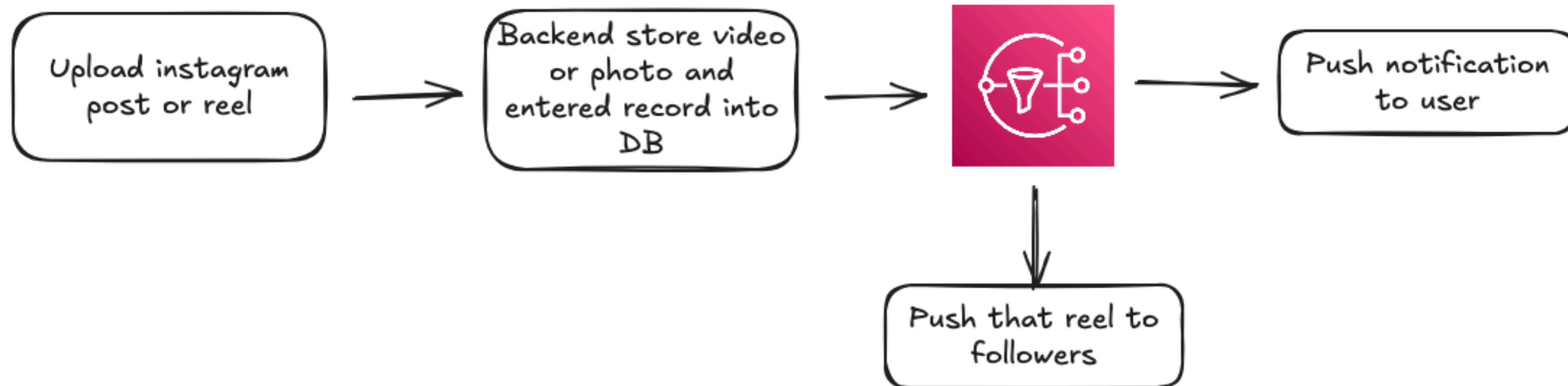
Simple Notifications  
Service



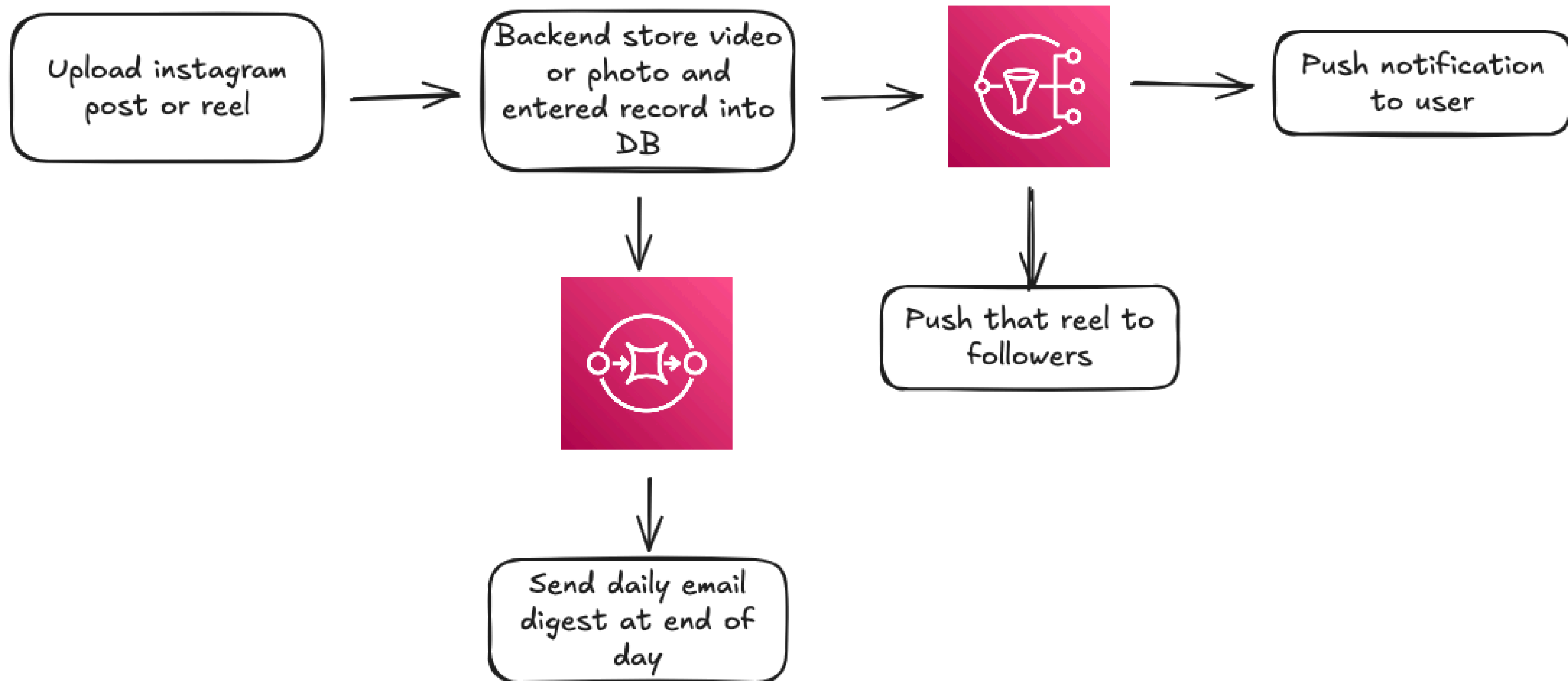
Steps functions

Many More...

# How AWS Handles Events



# How AWS Handles Events



# Stage 5: When Your APP deployment need to automate



CodePipeline

# Stage 6: When Your Infra need to maintain as code



CloudFormation

```
1  Ec2Instance:
2    Type: AWS::EC2::Instance
3    Properties:
4      InstanceType:
5        Ref: InstanceTypeParameter
6      ImageId: ami-0ff8a91507f77f867
```

## Infrastructure as Code



# Stage 7: App need caching content



CloudWatch

# Stage 6: App need logs to checkout



CloudWatch

Screenshot of the AWS CloudWatch console showing the 'API-Gateway-Execution-Logs' log group. The interface includes a sidebar with navigation options (1), a search bar (2), a query editor (4), a 'Run query' button (5), a visualization chart, and a table of log records. The 'Insights' section in the sidebar is highlighted with a blue circle (1). The search bar contains 'API-Gateway-Execution-Logs' (2). The query editor shows a query: 'fields @timestamp, @message | sort @timestamp desc' (4). The 'Run query' button is labeled (5). The visualization shows a bar chart of log volume over time. The table below the chart displays log records with columns for timestamp and message.

Query help [Learn more](#)

Commands

- fields
- filter
- stats
- sort
- limit
- parse

Discovered fields

Search for a field

#	@timestamp	@message
1	2019-07-01T10:33:05.977+02:00	(d90fae30-9bda-11e9-b8ea-bb82be7f51a7) Received response. Status: 200, Integrati
2	2019-07-01T10:33:05.977+02:00	(d90fae30-9bda-11e9-b8ea-bb82be7f51a7) Endpoint response headers: {Date=Mon, 01
3	2019-07-01T10:33:05.977+02:00	(d90fae30-9bda-11e9-b8ea-bb82be7f51a7) Endpoint response body before transformat
4	2019-07-01T10:33:05.977+02:00	(d90fae30-9bda-11e9-b8ea-bb82be7f51a7) Method response body after transformator
5	2019-07-01T10:33:05.977+02:00	(d90fae30-9bda-11e9-b8ea-bb82be7f51a7) Method response headers: {X-Amzn-Trace-Id
6	2019-07-01T10:33:05.977+02:00	(d90fae30-9bda-11e9-b8ea-bb82be7f51a7) Successfully completed execution
7	2019-07-01T10:33:05.977+02:00	(d90fae30-9bda-11e9-b8ea-bb82be7f51a7) Method completed with status: 204
8	2019-07-01T10:33:05.977+02:00	(d90fae30-9bda-11e9-b8ea-bb82be7f51a7) AWS Integration Endpoint RequestId : 5dd6
9	2019-07-01T10:33:05.977+02:00	(d90fae30-9bda-11e9-b8ea-bb82be7f51a7) X-ray Tracing ID : 1-5d19c541-d2a21ade52c
10	2019-07-01T10:33:05.902+02:00	(d90fae30-9bda-11e9-b8ea-bb82be7f51a7) Endpoint request URI: https://lambda.us-e

# Stage 8: App need Security



Identity Access  
Management



Don't Learn Services. Learn Components.



Think in Components → Map to AWS Services.

“Don’t fall into the trap of memorizing 200 AWS services. Identify what your app needs, then find which AWS service fits that role.”



# The Complete Builder's Roadmap

- EC2 for everything
- Add RDS
- Split frontend → S3/Amplify
- Backend → Lambda/Fargate
- CI/CD + CloudFormation
- Event-Driven → SNS / SQS / Step Functions

Don't Learn  — Build with 

# Thank you!



 <https://www.internetkatta.com/>

