



#Microservices

#Dockers

#Kubernetes

Building Modern Applications -



#Evolve^o

Jargon Busting...

Agile Teams

Continuous Integration

Cloud Architecture

Daily code to customer

Time to Market

Scale

12 Factor App

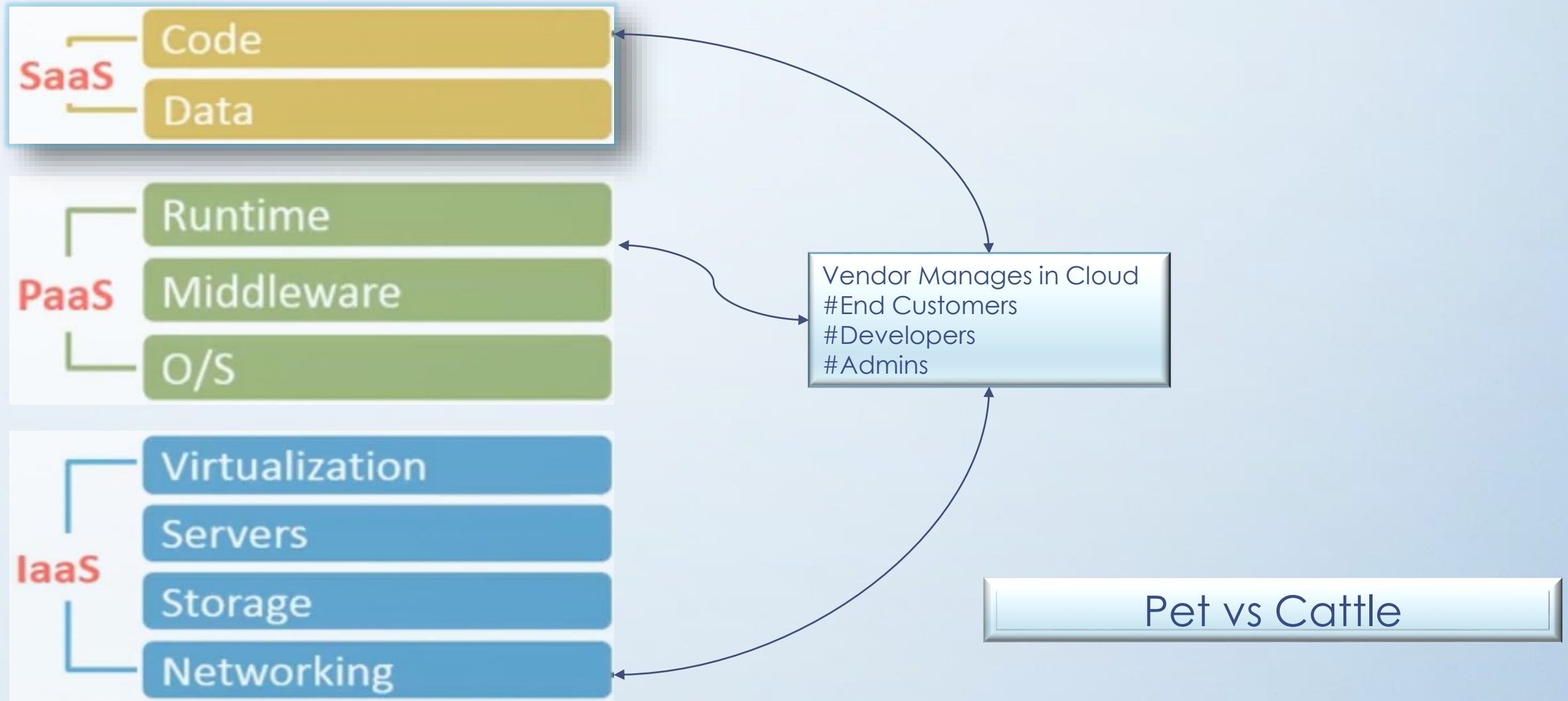
Blue-Green Deployment

A/B Testing

OPEN Source

Developer Friendly

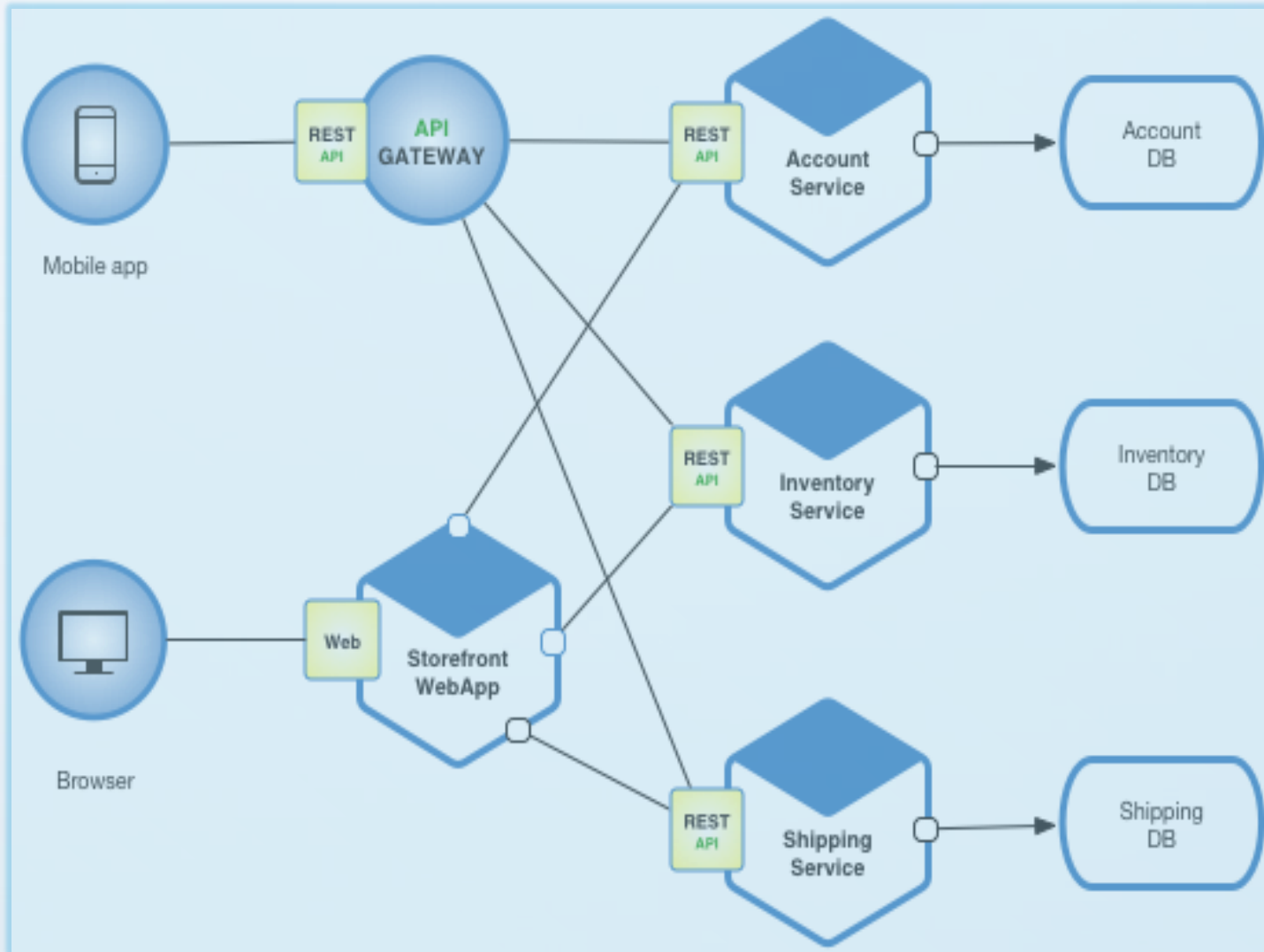
Tradition, on-premises → IaaS, PaaS, SaaS



Think Microservices as...

- ≡ Decomposing
- ≡ Single-Function
- ≡ Well-Defined Interfaces
- ≡ Independent
- ≡ Small Teams
- ≡ Entire lifecycle
- ≡ Minimizing communication
- ≡ Reduce the scope of change
- ≡ Design for failure

Microservice Architecture



- △ Codebase
- △ Dependencies
- △ Backing Services
- △ Build, Release, Run
- △ Stateless Processes
- △ Disposability
- △ Dev/Prod Parity
- △ Logs


Design Patterns for Microservices

Aggregator Pattern

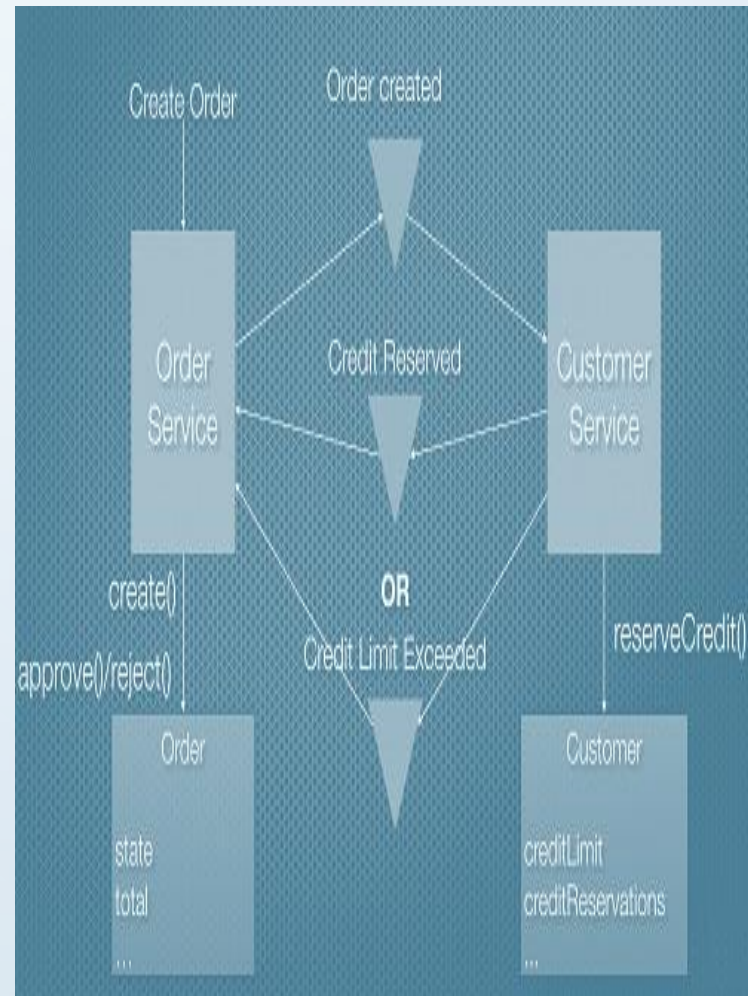
Defining the Warehouse API

This part of the configuration first defines the valid URIs for the Warehouse API and then defines a common policy for handling requests to the Warehouse API.

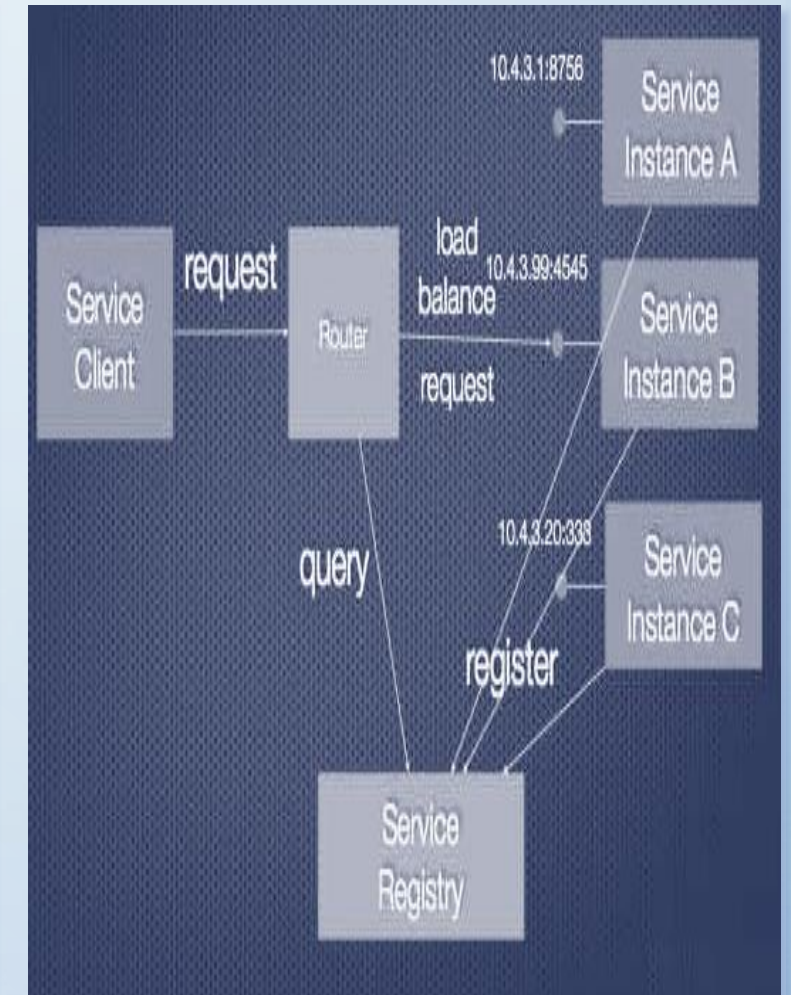
```
1 # API definition
2 #
3 location /api/warehouse/inventory {
4     set $upstream warehouse_inventory;
5     rewrite ^ /_warehouse last;
6 }
7
8 location /api/warehouse/pricing {
9     set $upstream warehouse_pricing;
10    rewrite ^ /_warehouse last;
11 }
12
13 # Policy section
14 #
15 location = /_warehouse {
16     internal;
17     set $api_name "Warehouse";
18
19     # Policy configuration here (authentication, rate limiting, logging, more...)
20 }
```



Saga Pattern



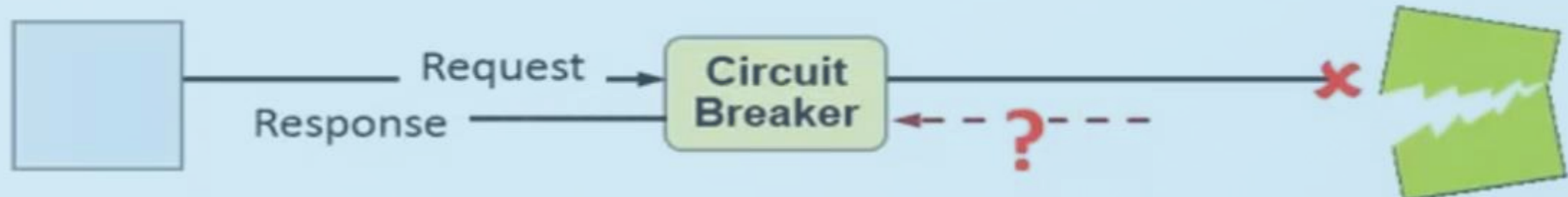
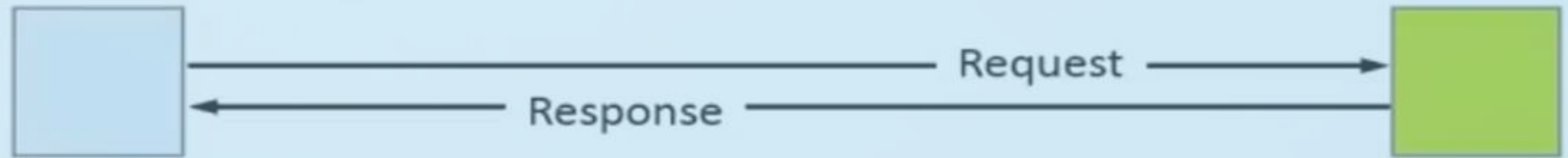
Discovery Pattern



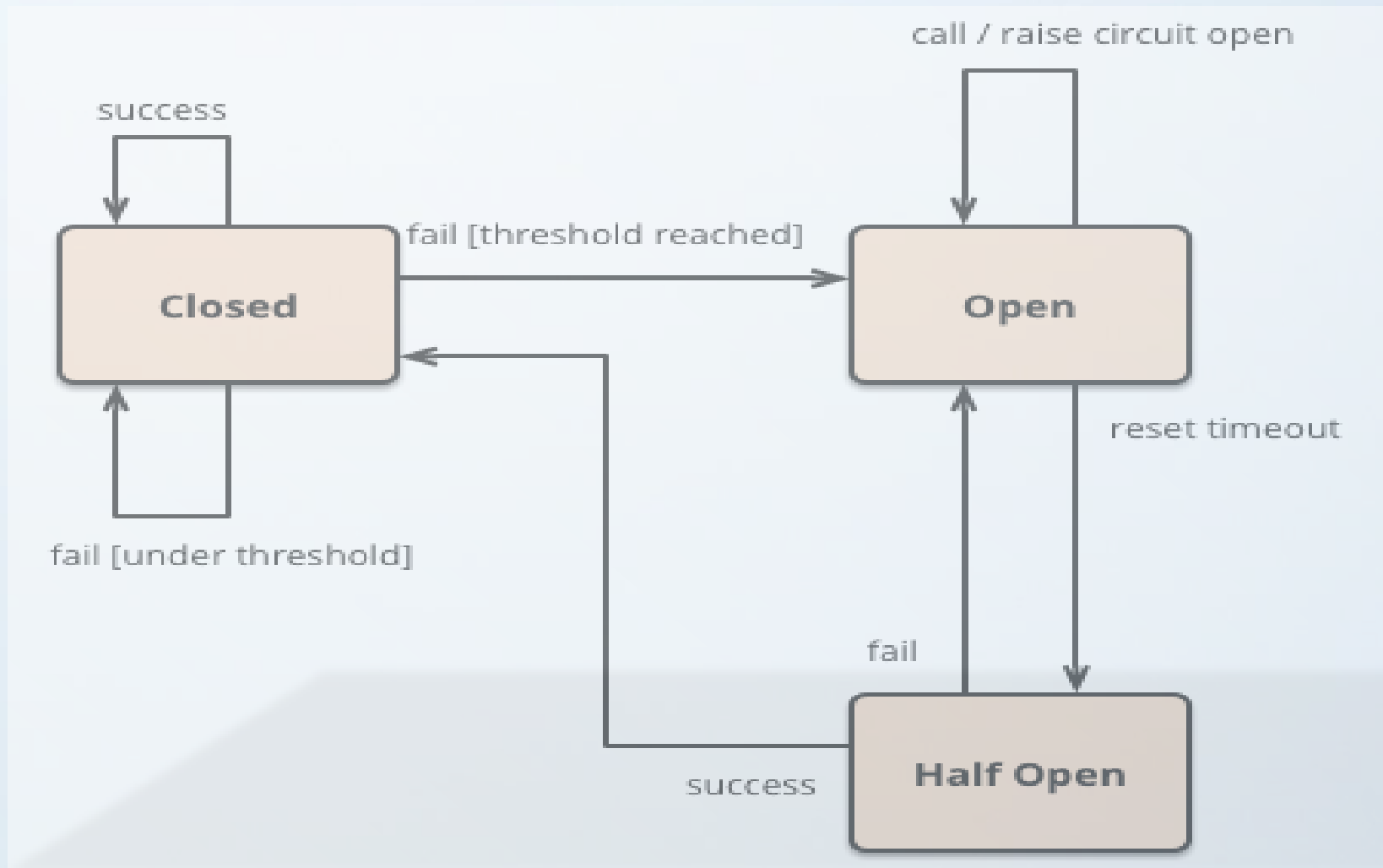
Circuit breaker

Microservice1

Microservice2



Circuit Breaker Pattern



Decomposing Monolith – Strangler Pattern

- Refactor the code:
 - Each REST service is potentially a Microservice (*Have Domain Perspective*)
 - Keep each Microservice persistent data private to that service and accessible only via its API
 - A set of tables that is only used by one service
 - Blobs to be stored in NoSQL Database (*Designed for scale, cost effectiveness*)
 - Redis to be used for Key-Value Store
 - Use Independent DevOps (CI/CD) for each service to be deployed in production



**It's time to embrace 'you build it, you run it'
- Jeff Bezos**



Containers



Dockers

Dockerfile

 Copy

```
FROM microsoft/dotnet:2.1-sdk
WORKDIR /app

# copy csproj and restore as distinct layers
COPY *.csproj ./
RUN dotnet restore

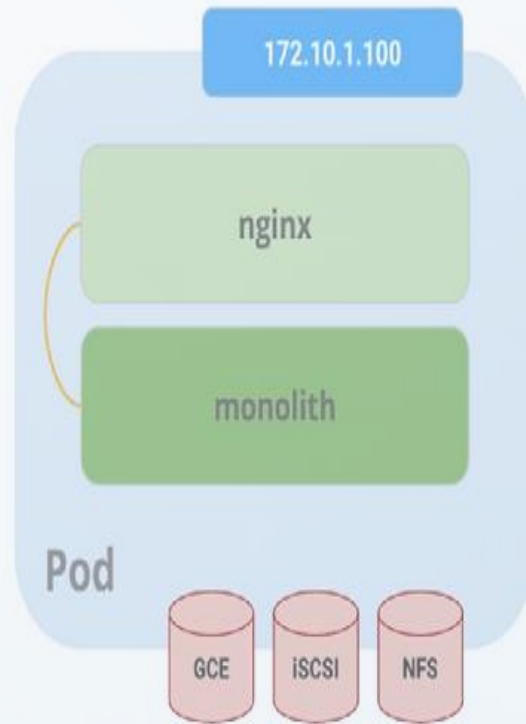
# copy and build everything else
COPY . ./
RUN dotnet publish -c Release -o out
ENTRYPOINT ["dotnet", "out/Hello.dll"]
```

Kubernetes - DevOps

Pods

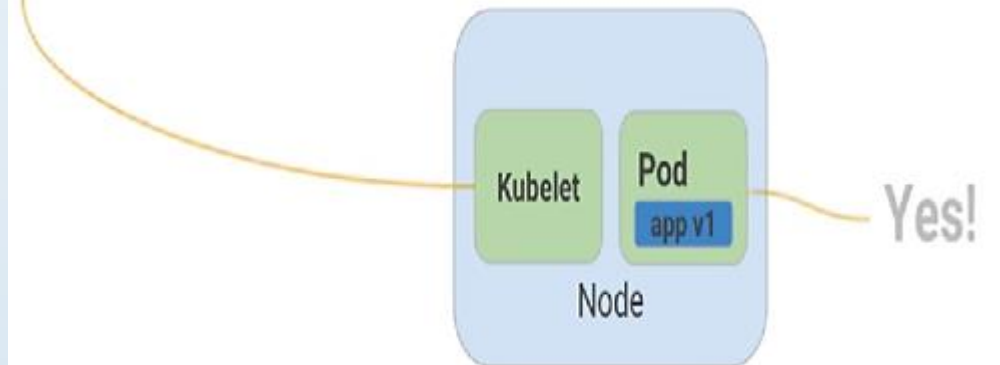
Logical Application

- One or more containers and volumes
- Shared namespaces
- One IP per pod



Monitoring and Health Checks

Hey, app v1... You alive?



@Stake #BrandImage #MarketCompetition



Twitter Bug May Have Sent Messages From 3 Million Users To Developers

The bug ran from May 2017 and within hours of discovering it on September 10, Twitter said it fixed the bug to prevent data from being unintentionally sent to the incorrect developer.

World | Indo-Asian News Service | Updated: September 22, 2018 10:53 IST

#An existing issue of Twitter has to be “Fixed”, “Impact Tested”, “Deployed” with-in hours

#Netflix, handles over a billion calls per day to their video streaming API from over 800 different kinds of devices

Source:

<https://9to5mac.com/2019/01/28/facetime-bug-hear-audio/>

<https://www.ndtv.com/world-news/3-million-twitter-users-messages-sent-to-unknown-developers-due-to-bug-1920396>

now I'll open the floor for questions

Credits Learning pathway:

- ✓ <https://www.coursera.org/learn/intro-ibm-microservices/home/info>
- ✓ <https://in.udacity.com/course/scalable-microservices-with-kubernetes--ud615>
- ✓ <https://dzone.com/articles/design-patterns-for-microservices>
- ✓ <https://martinfowler.com/articles/microservices.html>
- ✓ <https://www.youtube.com/watch?v=94PxlbuiZCU>
- ✓ <https://www.youtube.com/watch?v=pGYAg7TMmp0>
- ✓ <https://www.youtube.com/watch?v=YFI2mCHdv24>
- ✓ <https://www.youtube.com/watch?v=4ht22ReBjno>
- ✓ <https://www.youtube.com/watch?v=2vMEQ5zs1ko>
- ✓ <https://www.youtube.com/watch?v=Q5POuMHxW-0>
- ✓ <https://youtu.be/V9IJj4MzZBc>
- ✓ <https://12factor.net/>
- ✓ <https://microservices.io/patterns/data/saga.html>

Thank You...