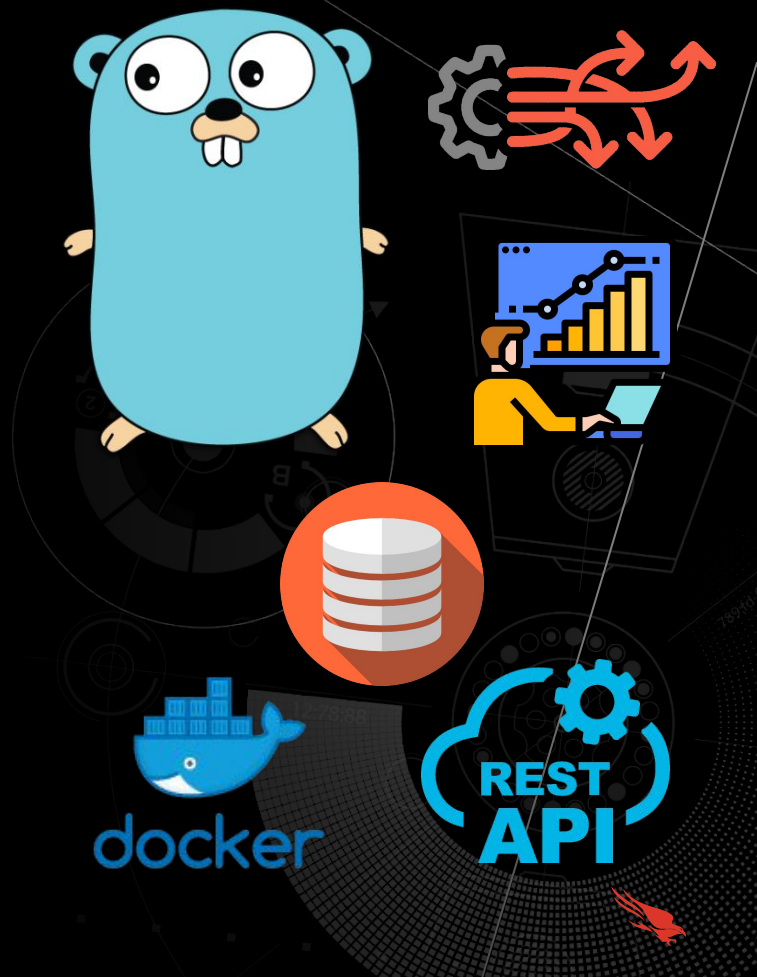
Rest API

CrowdStrike HEROES - Cloud Workshop

CROWDSTRIKE

Date	Topic
July 25 - 16:00	Intro to golang
July 26 - 16:00	Intro to golang (continuation)
July 27 - 16:00	Multithreading
July 28 - 16:00	**Rest API**
July 29 - 16:00	Unit testing, logging and monitoring
August 1 - 16:00	Workshop and Q&A
August 2 - 16:00	Deployments/Docker
August 3 - 16:00	Databases
August 4 - 16:00	Databases extended
August 5 - 13:00	Microservices contest (4h with Awards)



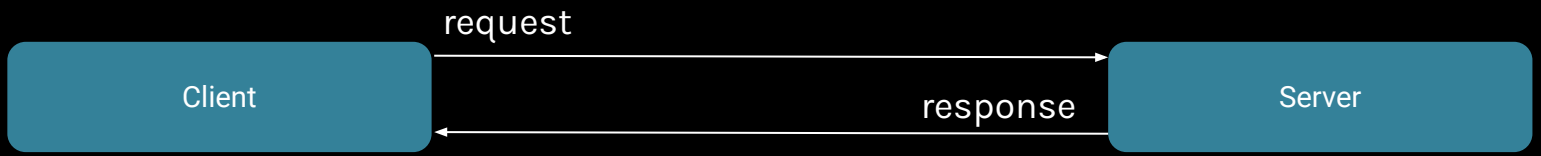
CROWDSTRIKE

What is a microservice

- No consensus on exact definition, generally speaking is a piece of software
- Is deployed independently
- Exposes an API or exposes its data in other ways
- Solves a particular task or a couple of related tasks from a bounded context
- Communication is a key aspect of a microservice

Synchronous Microservices

- Server exposes a synchronous API
- Client sends a request and awaits for Server to respond



CROWDSTRIKE

Types of APIs used in the real world

- REST API
- gRPC
- GraphQL

CROWDSTRIKE

Rest API

- Is a stateless API
- Communicates over HTTP
- Well known and very common on web

Rest API

ex1 : POST https://www.tasks.com/tasks - create task
ex2: PUT https://www.tasks.com/tasks/1234 - update task

- URIs encode the resource address /tasks/1234 => task with id 1234
- HTTP method encodes the action performed on the resource
  - POST - Creating a resource
  - GET - Get a resource information
  - PUT - Updating a resource or creating it
  - PATCH - Partially updating a resource(patching)
  - DELETE - Deleting a resource

CROWDSTRIKE

REST API Resources

- Resources represent an entities exposed by the API
- Resources depend on the use case and problem domain
- REST API hides the resource implementation details

CROWDSTRIKE

REST API Resource examples

- For example a car renting API will have entities like car, user, reservation
- Ex: GET https://cars.com/cars to get a list of cars
- Ex: GET https://cars.com/reservations to get a list of reservations
- Ex: GET https://cars.com/users/1234/cars to get the cars rented by user 1234

REST APIs HTTP Methods

- POST - Creating a resource
  ex: POST https://cars.com/cars to create a new car
- GET - Get a resource information
  ex: GET https://cars.com/cars/1234 to get the car with id 1234
- PUT - Updating a resource
  ex: PUT https://cars.com/cars/1234 to update the car with id 1234
- PATCH - Partially updating a resource(patching)
  ex: PATCH https://cars.com/cars/1234 to update the car with id 1234
- DELETE - Deleting a resource
  ex: DELETE https://cars.com/cars/1234 to delete the car with id 1234

CROWDSTRIKE

Request Body

- Ex of JSON Body: {"name":"product1", "description":"My awesome product"}
- Only POST/PUT/PATCH HTTP methods accept request body
- Request body size is important

HTTP Response

- Status Code - contains information if the request was processed successfully
- Response Body - contains the actual response data
- Ex of Response Body: {"name":"product1", "description":"My awesome product"}
- All HTTP methods can return response bodies
- Response body size is important

CROWDSTRIKE

Important HTTP Response status codes

- 200 - OK, a general code meaning that request was processed correctly
- 201 - Created, a specific code meaning that resource was created correctly
- 400 - Bad Request, the server could not process the request data
- 401 - Unauthorized, the client is missing necessary authorization
- 404 - Not Found, the requested resources is missing
- 422 - Unprocessable entity, the request failed validation
- 500 - Internal Server Error, a general code for unexpected server side errors

CROWDSTRIKE

Scaling REST APIs

- Vertical Scaling - spend more money get bigger machines
- Horizontal scaling - increase the machines count

CROWDSTRIKE

Vertical scaling

- Handle bigger load right now if you have the money for bigger machines
- Cost grows exponentially
- If the machine fails the service is down
- You will have to scale horizontally eventually anyway

CROWDSTRIKE

Horizontal Scaling advantages

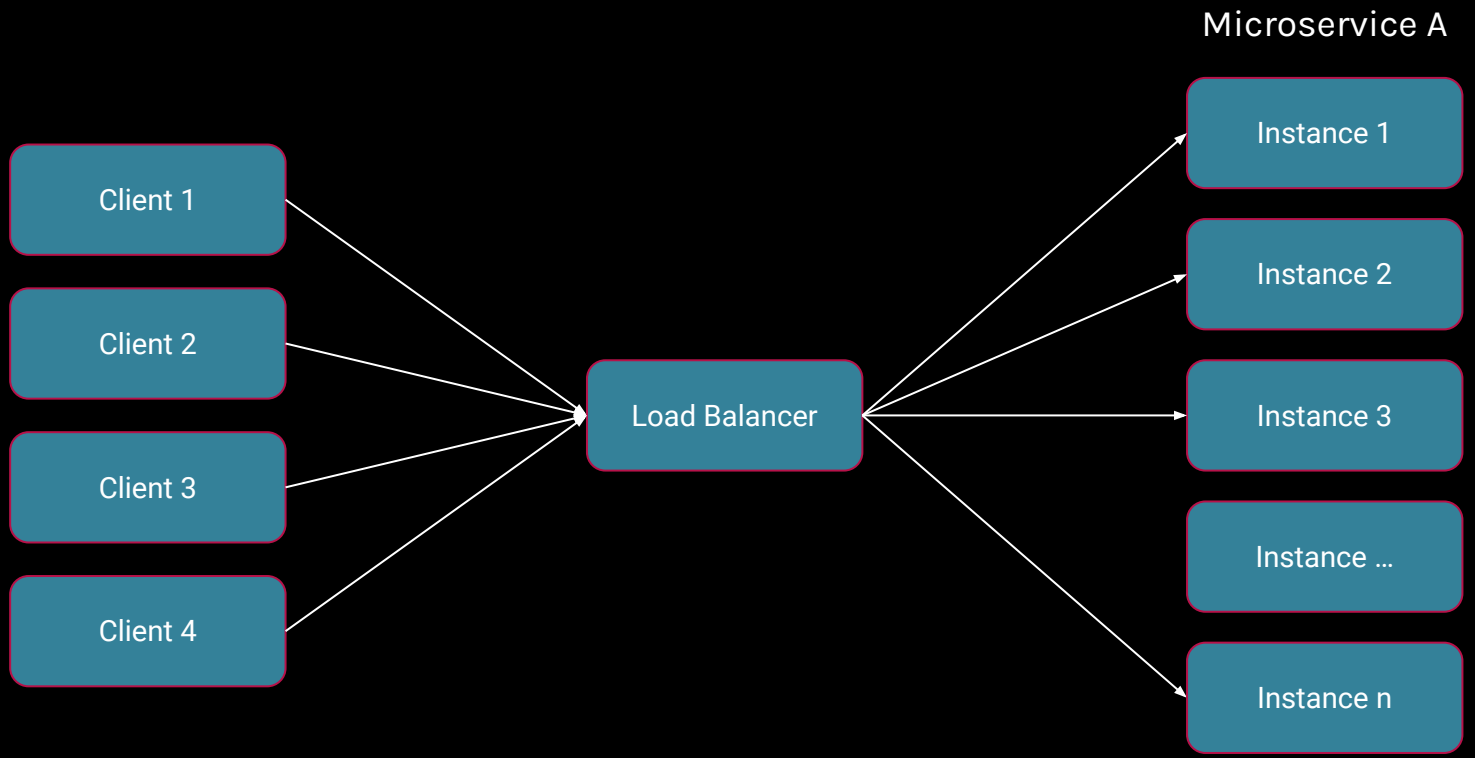
- Better network usage
- You can scale dynamically depending on load
- Cost grows linearly
- Besides good performance you get as a bonus
  - distribution
  - resilience

CROWDSTRIKE

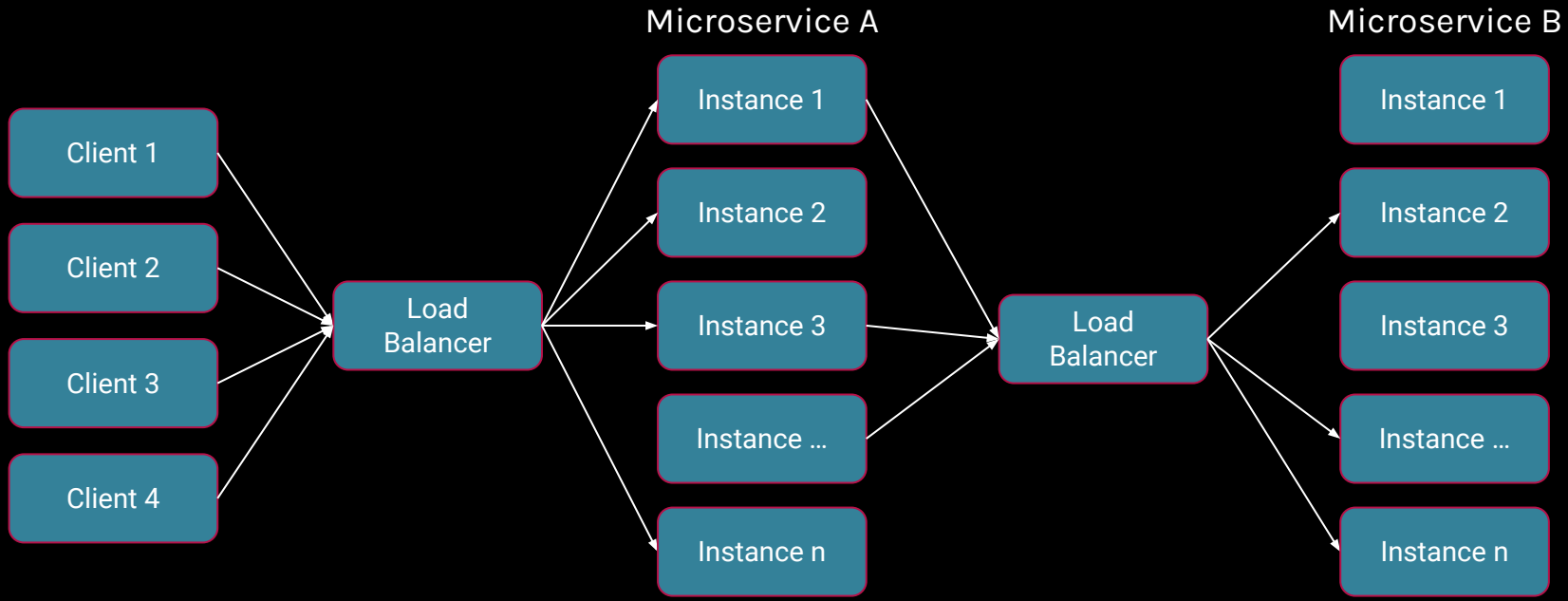
Load Balancers

- Decouples the client from the microservice by redirecting the requests
- Redirect the client requests to multiple microservice instances
- The client only needs to know the Load Balancer address

Load Balancers



Load Balancers



CROWDSTRIKE

HTTP (Hypertext Transfer Protocol) very short recap

Method URL VERSION

HEADERS

BODY

CROWDSTRIKE

Any questions?