Proactive Serverless Function Resource Management

Sixth International Workshop on Serverless Computing (WoSC6) 2020

Erika Hunhoff, Shazal Irshad, Vijay Thurimella*, Ali Tariq, Eric Rozner

University of Colorado Boulder, *Thrive, Inc

Background

Freshen Design

Evaluation

Discussion

Questions

Outline

Background

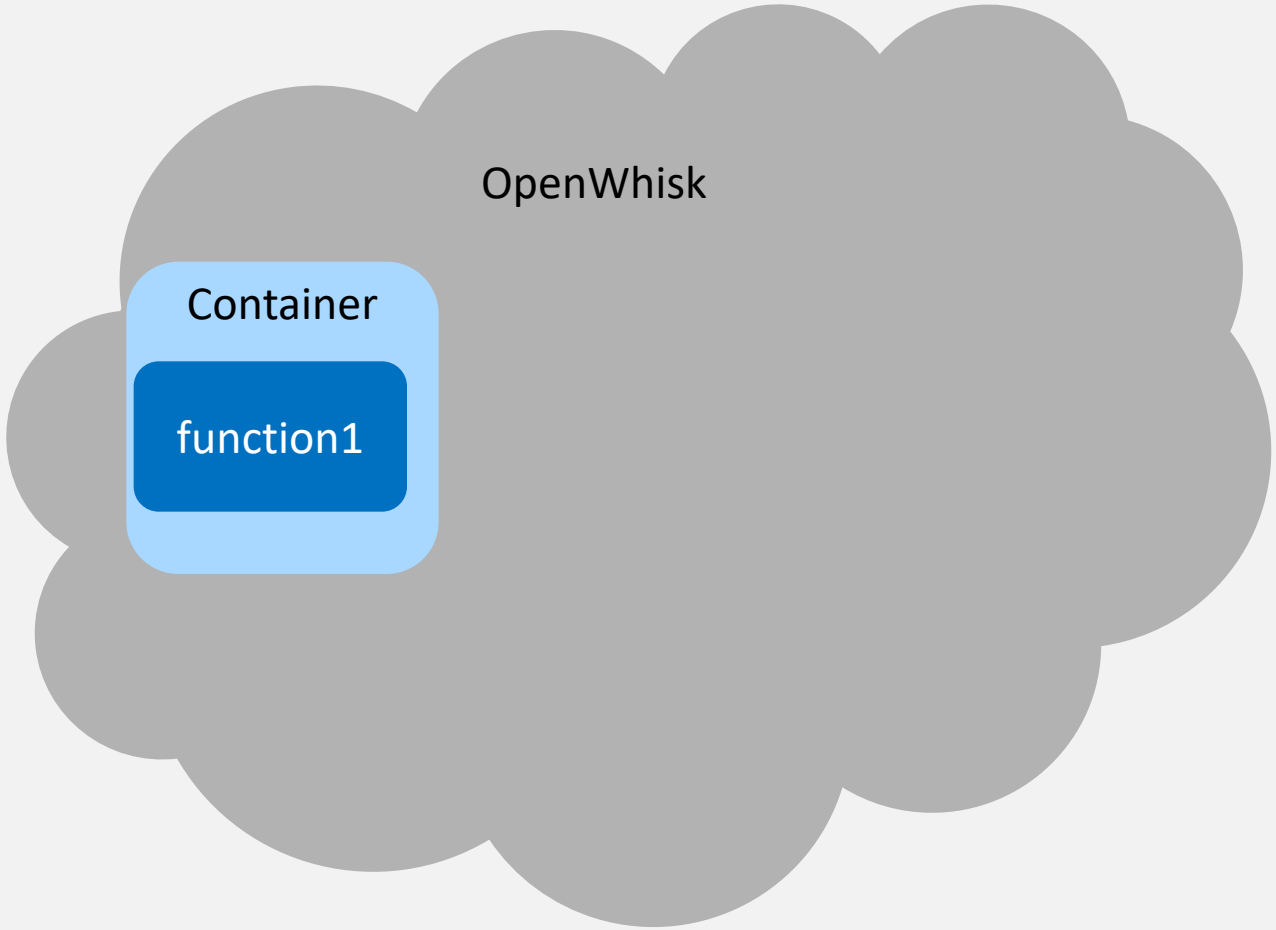
Freshen Design

Evaluation

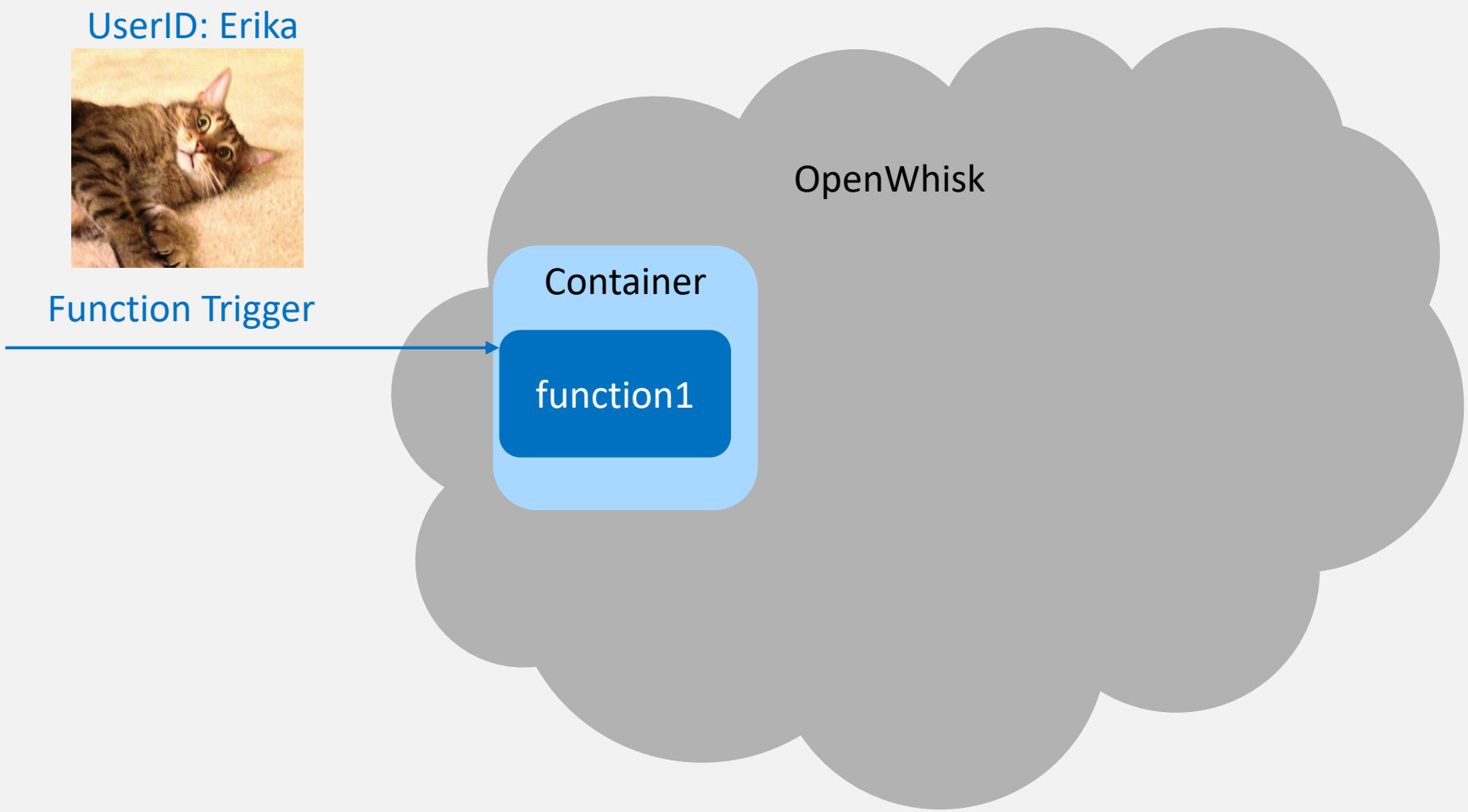
Discussion

Questions

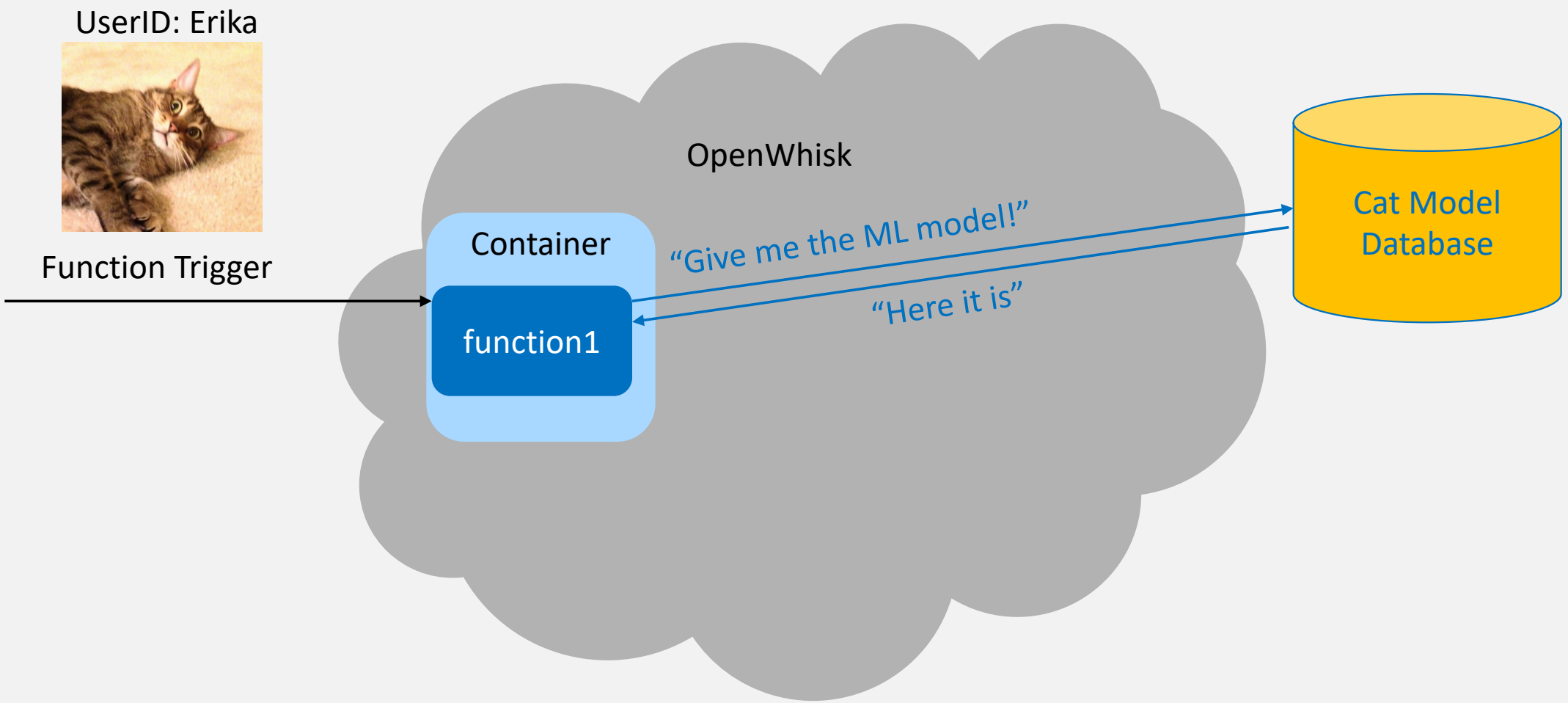
Outline



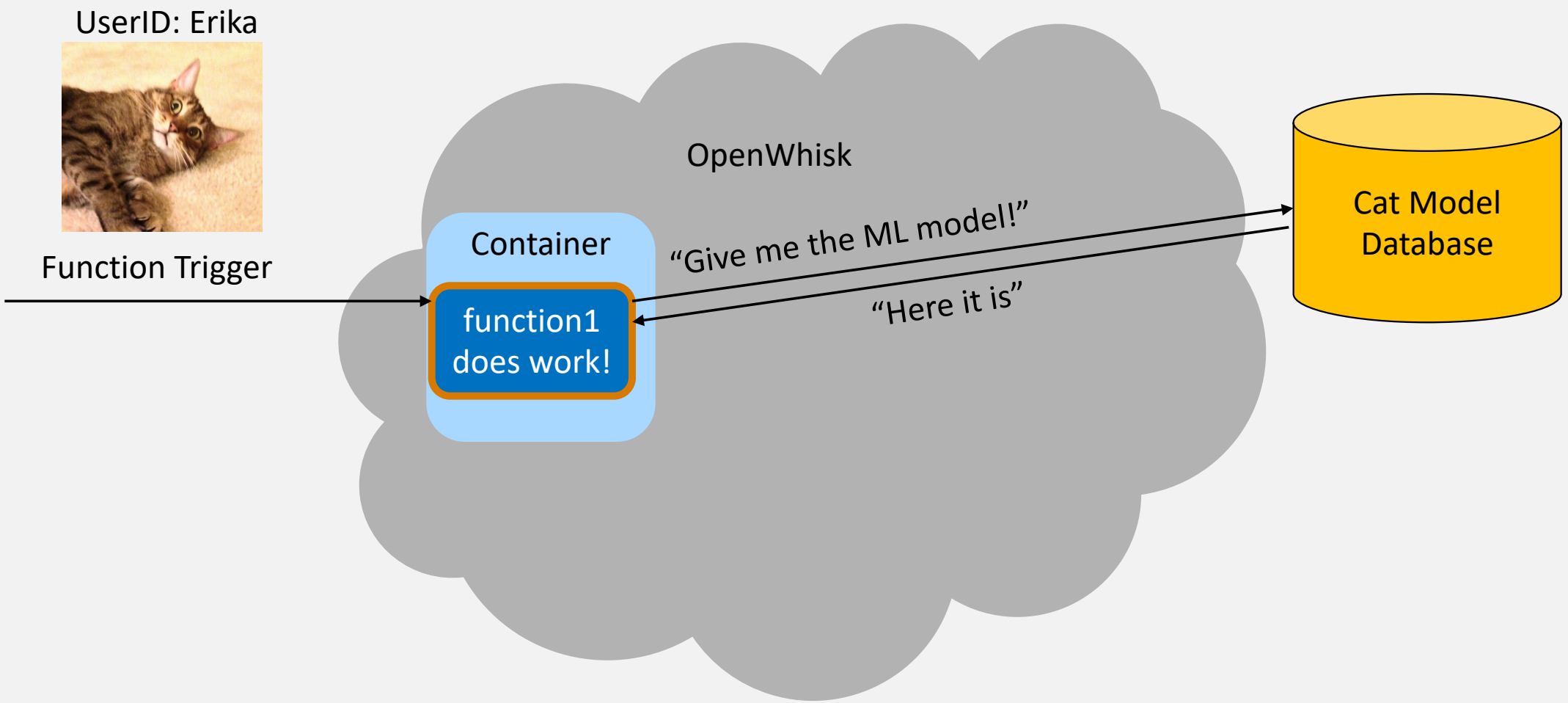
IDCat Serverless Application



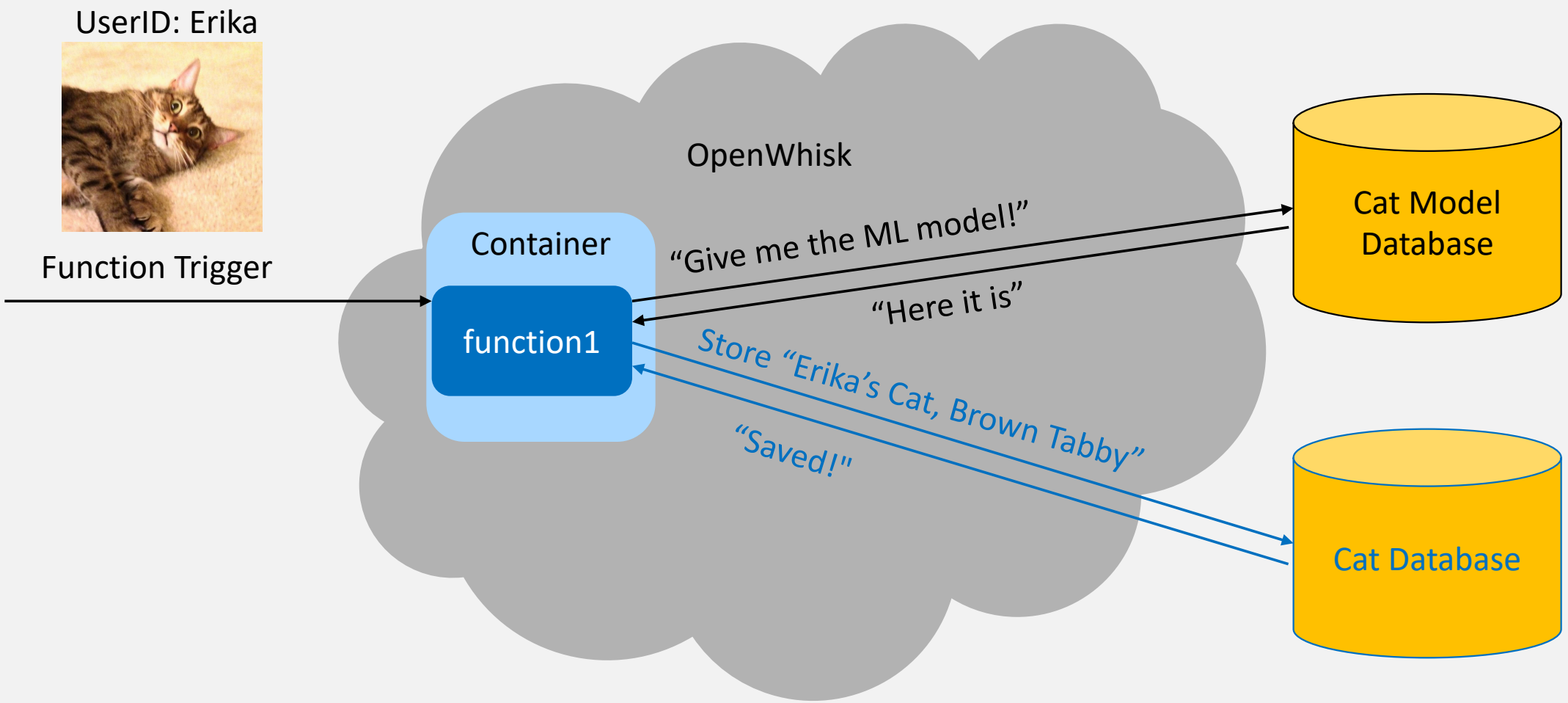
IDCat Serverless Application



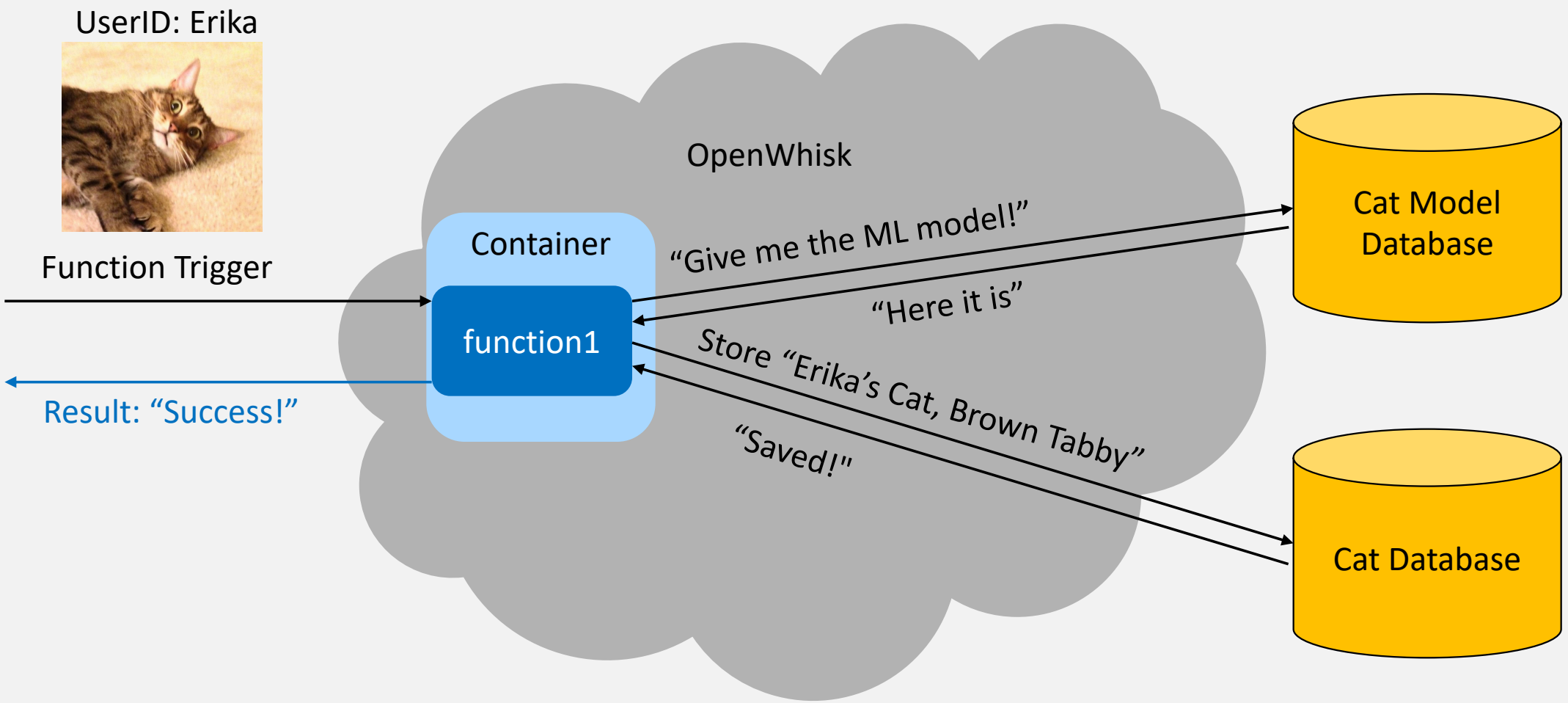
IDCat Serverless Application



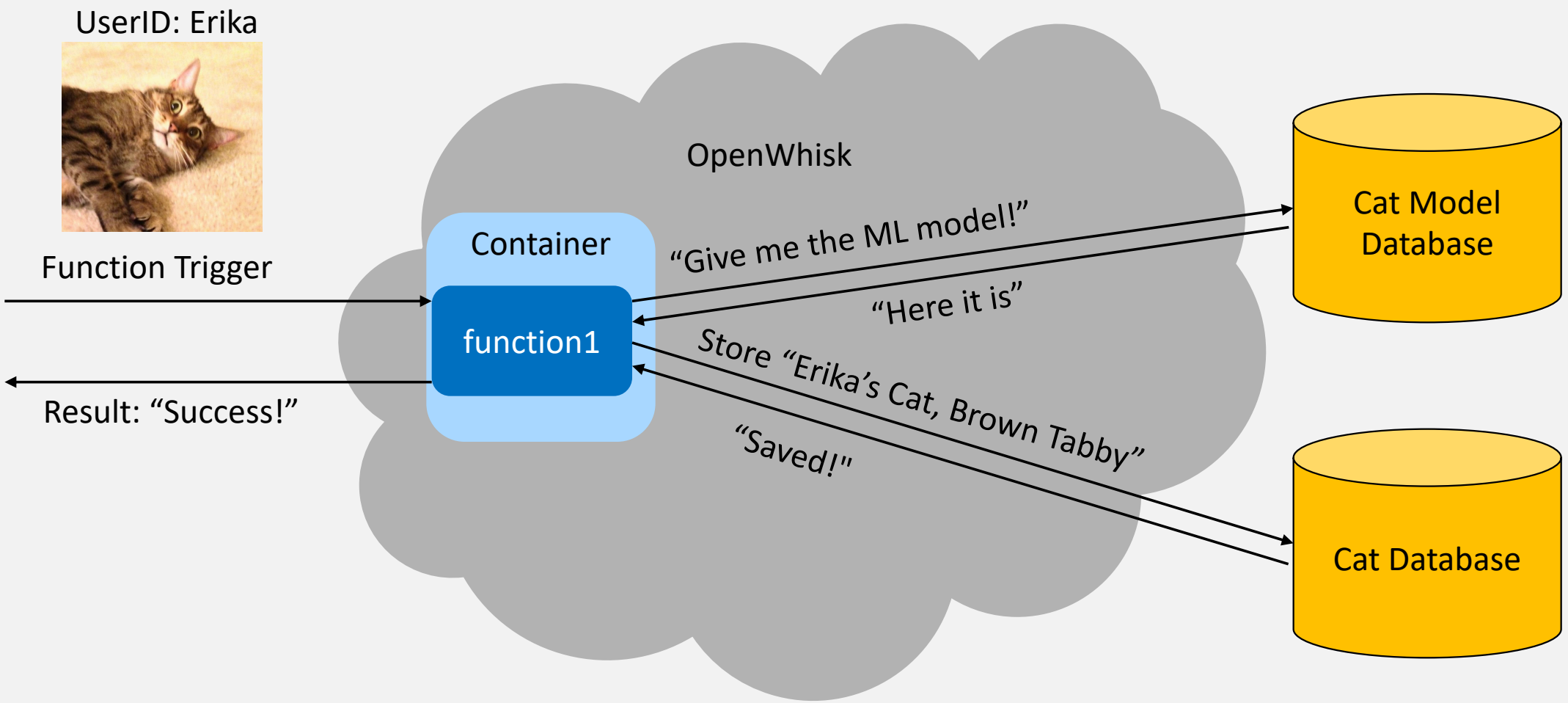
IDCat Serverless Application



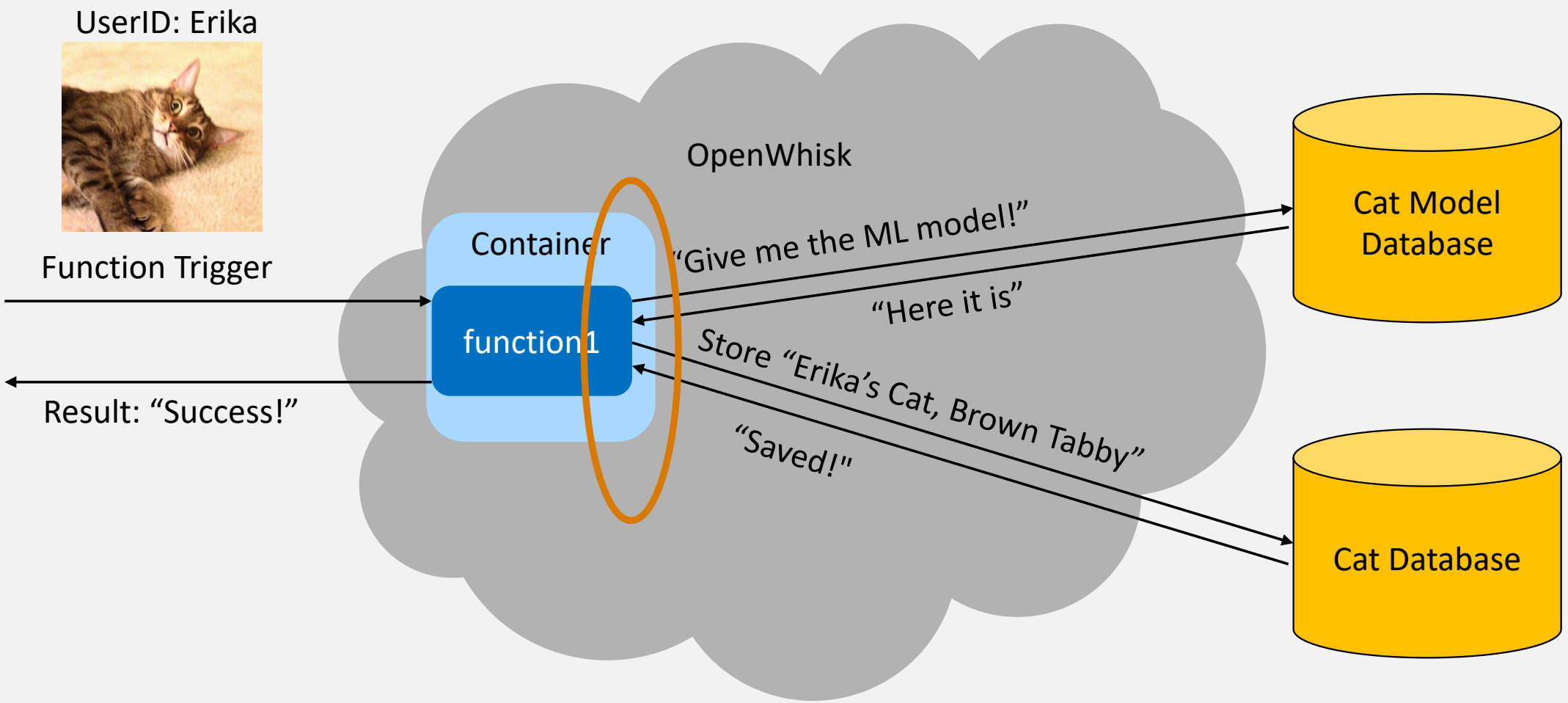
IDCat Serverless Application



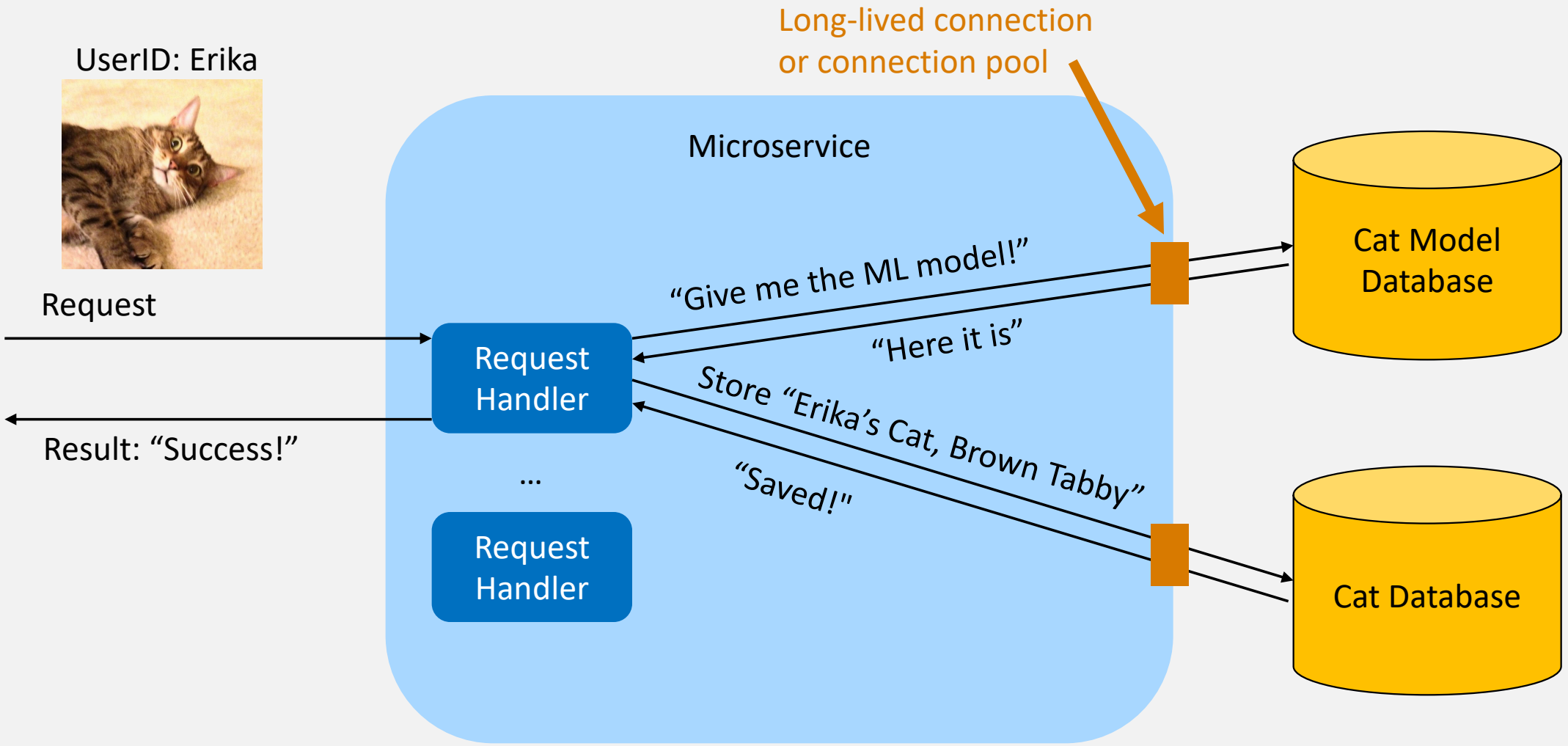
IDCat Serverless Application



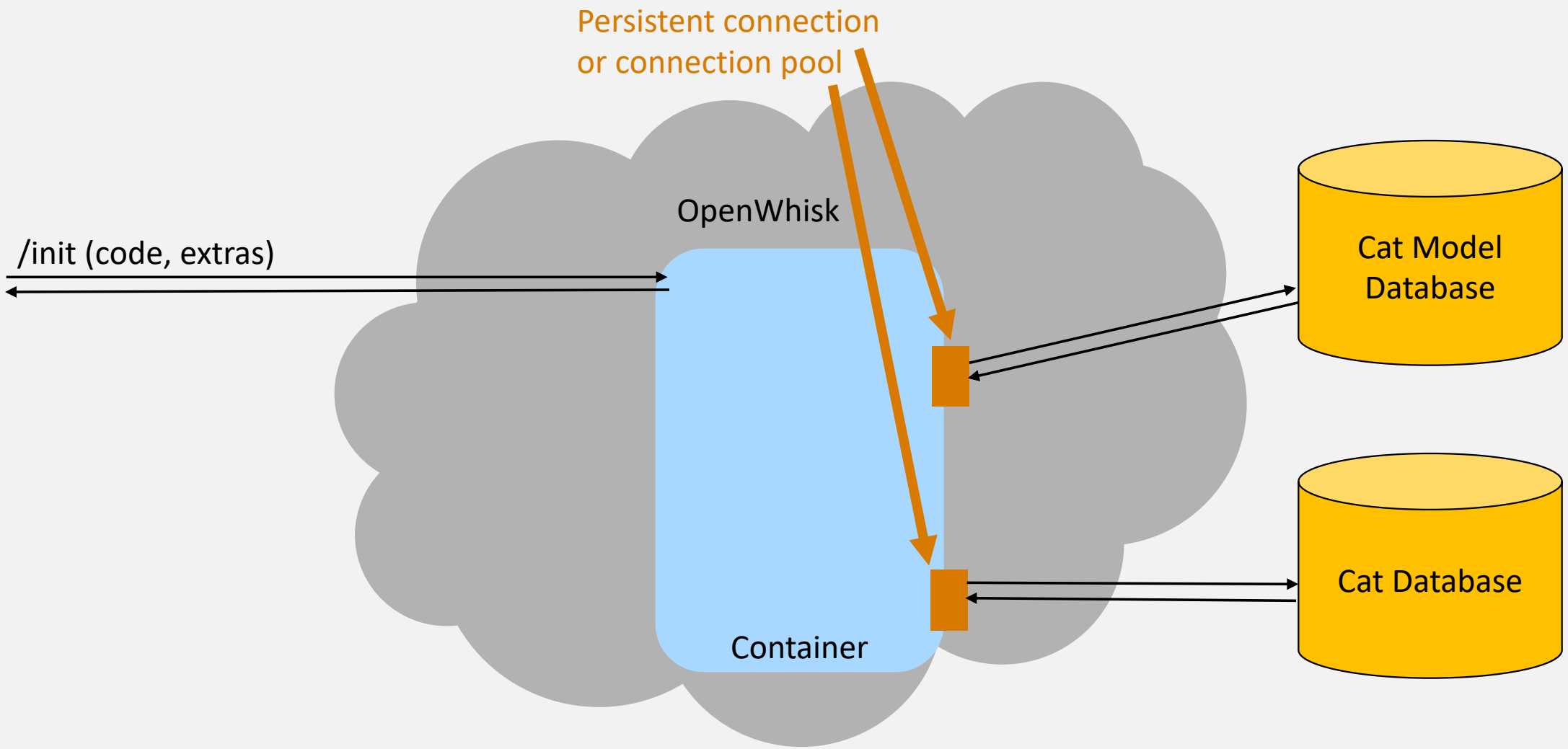
Improve Efficiency?



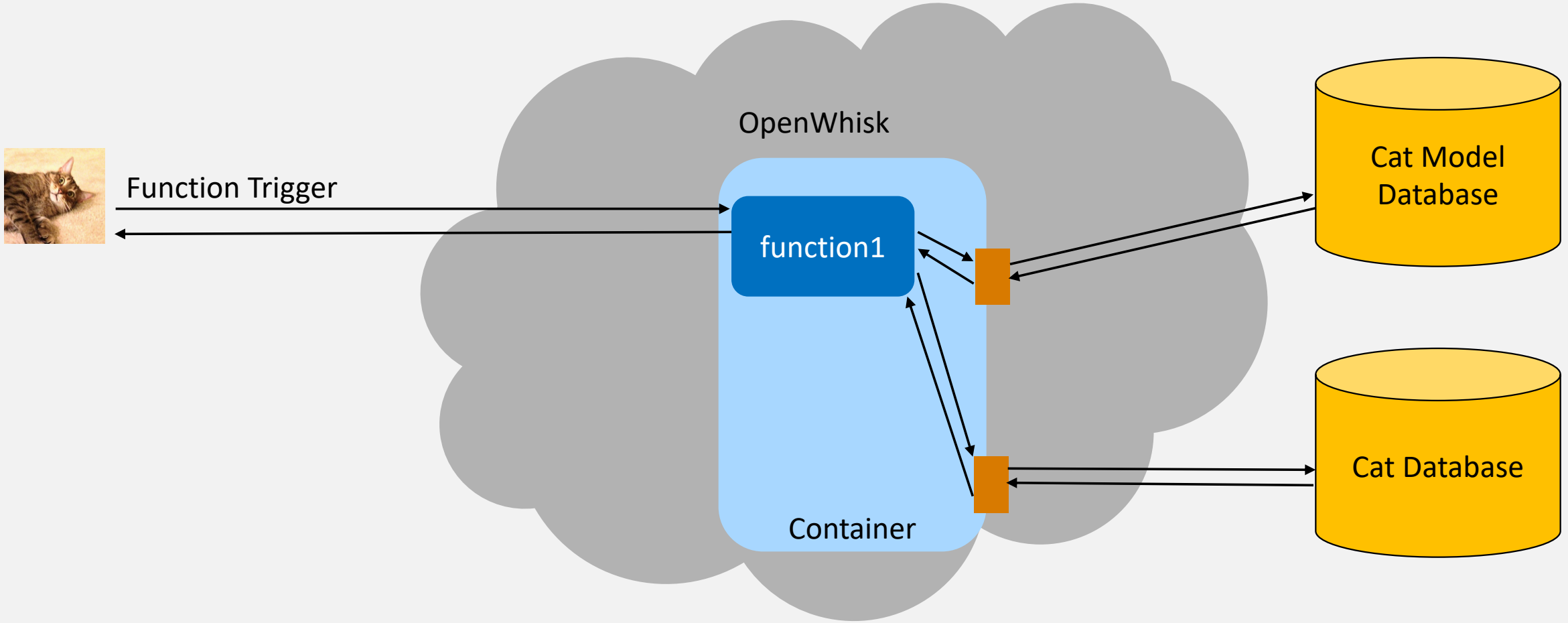
Improve Efficiency?



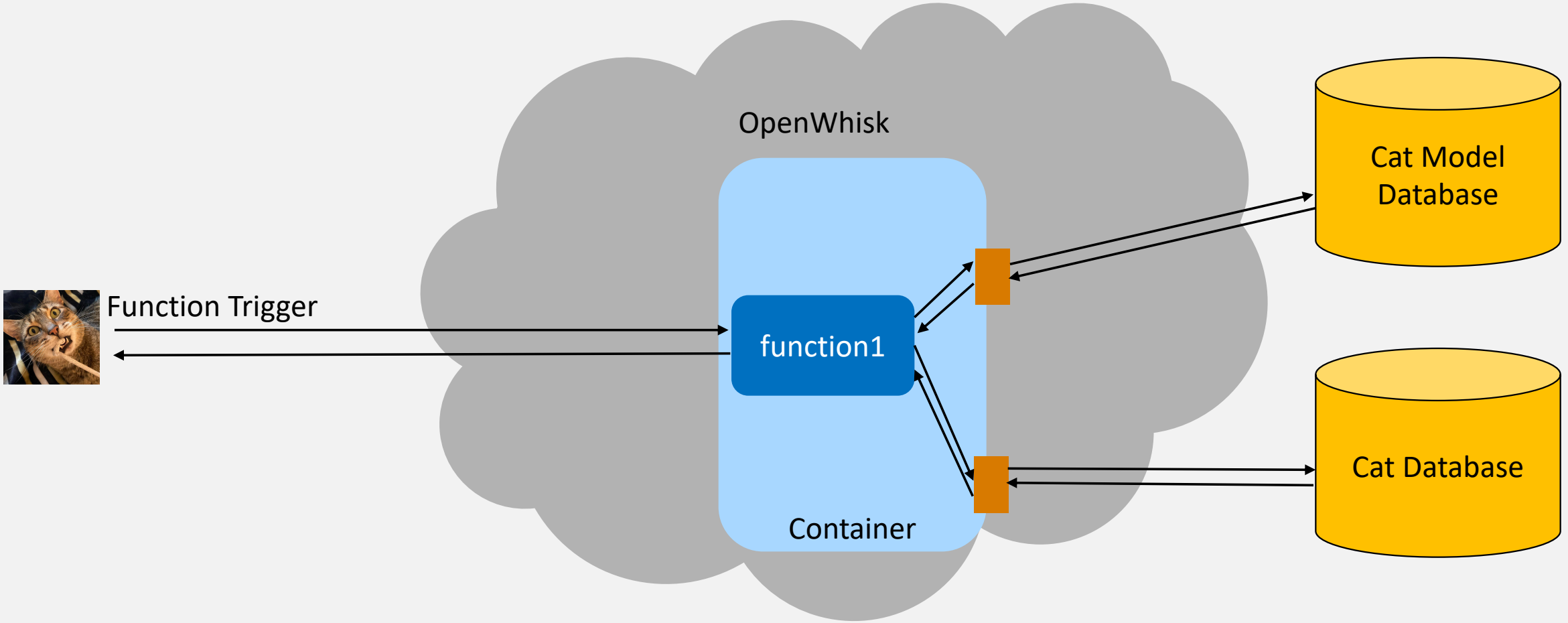
IDCat Microservice



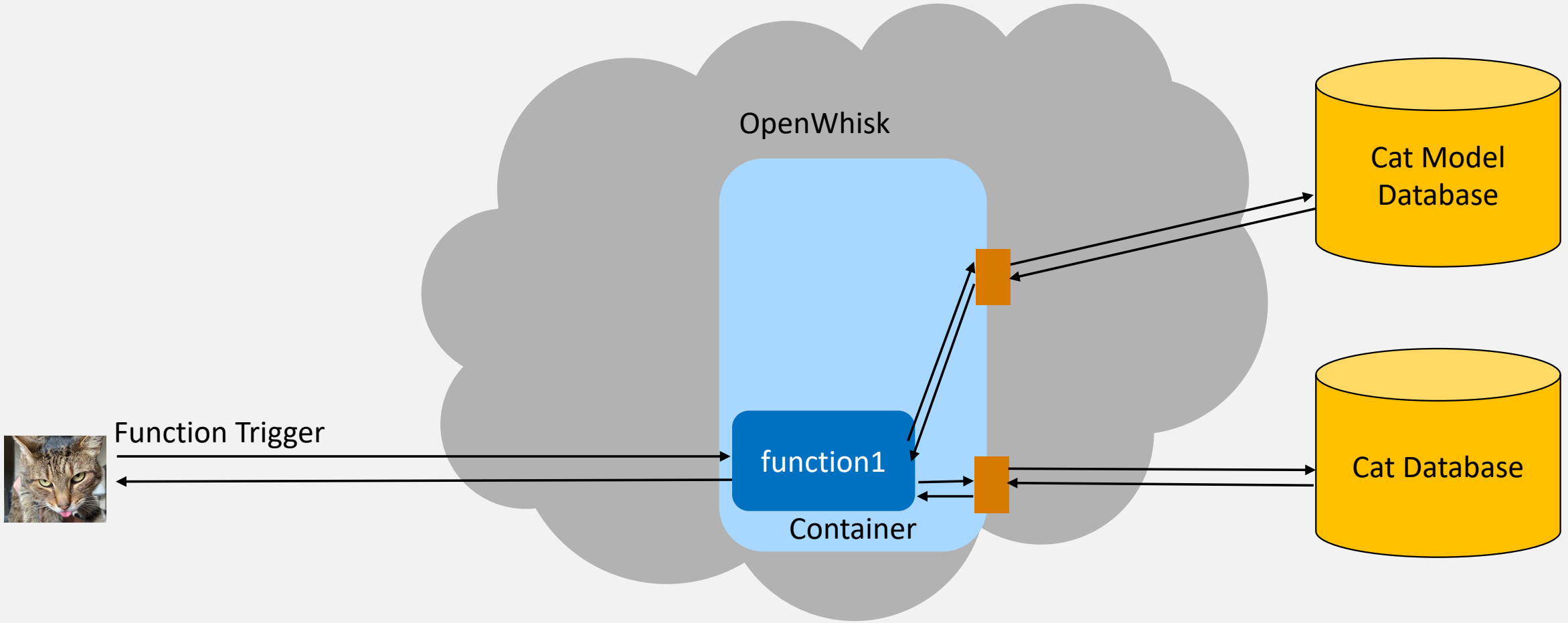
Runtime Reuse in Serverless



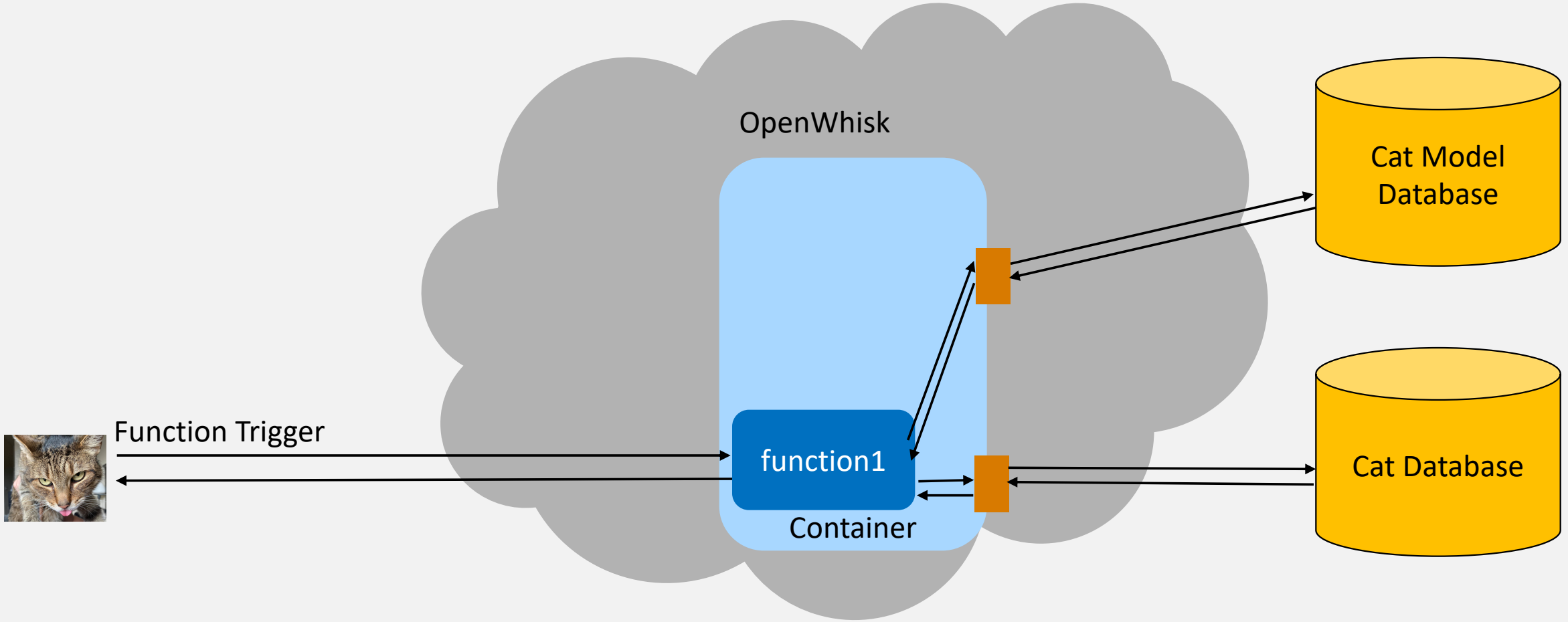
Runtime Reuse in Serverless



Runtime Reuse in Serverless



Runtime Reuse in Serverless



Runtime Reuse in Serverless – **Room for Improvement?**

Background

***Freshen* Design**

Evaluation

Discussion

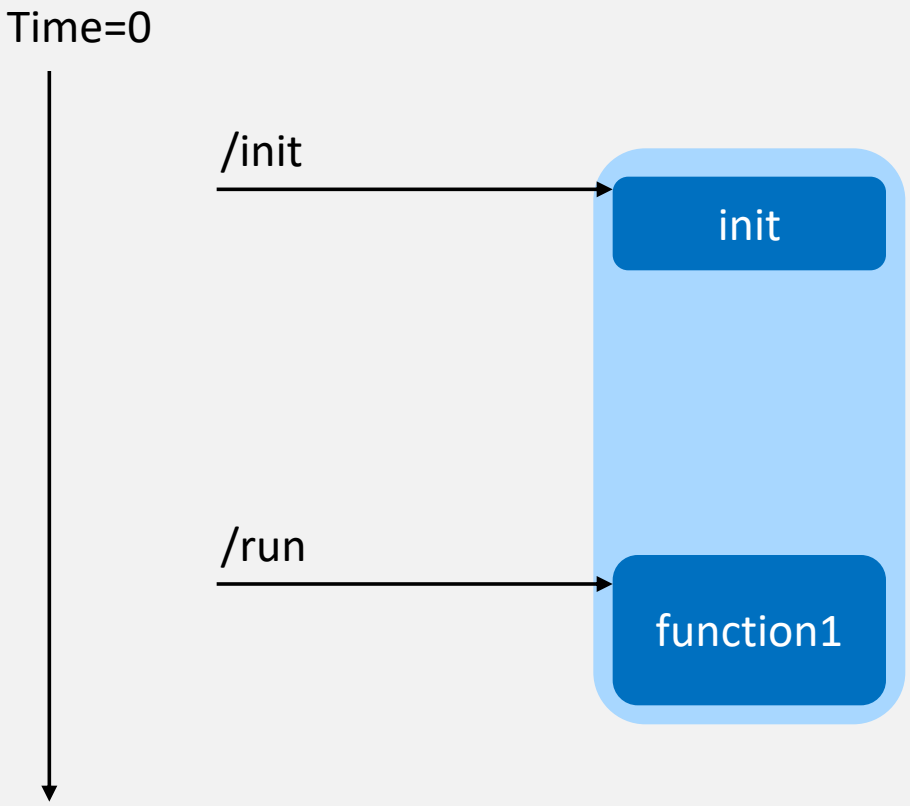
Questions

Outline

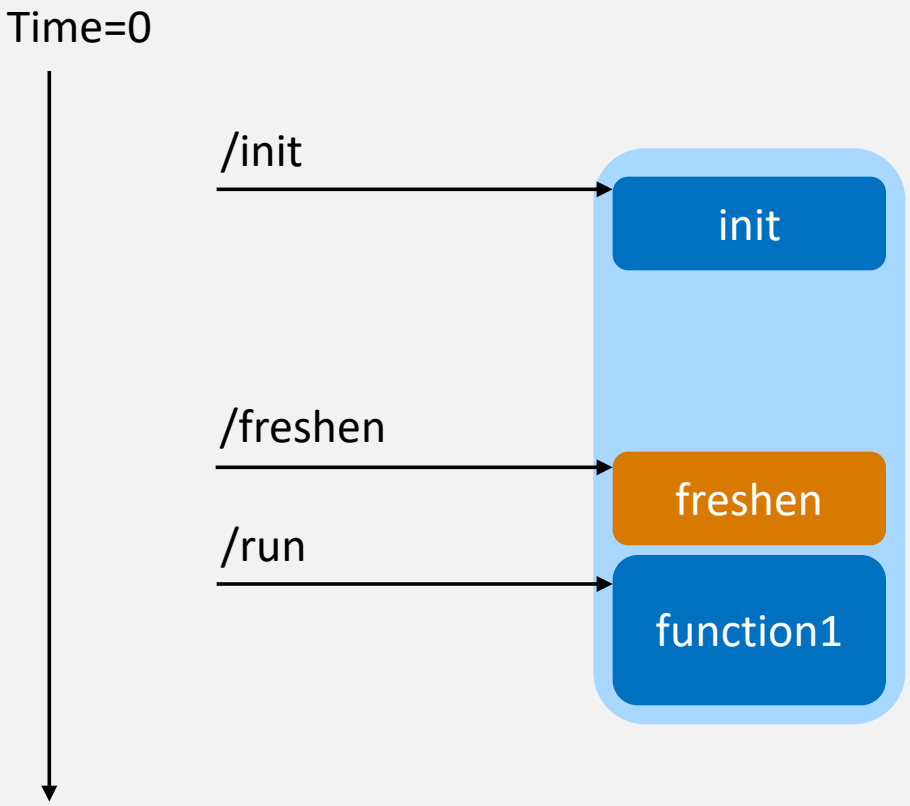
Overview

- We propose a new serverless runtime primitive, ***freshen***, as a mechanism to enable proactive serverless function resource management.

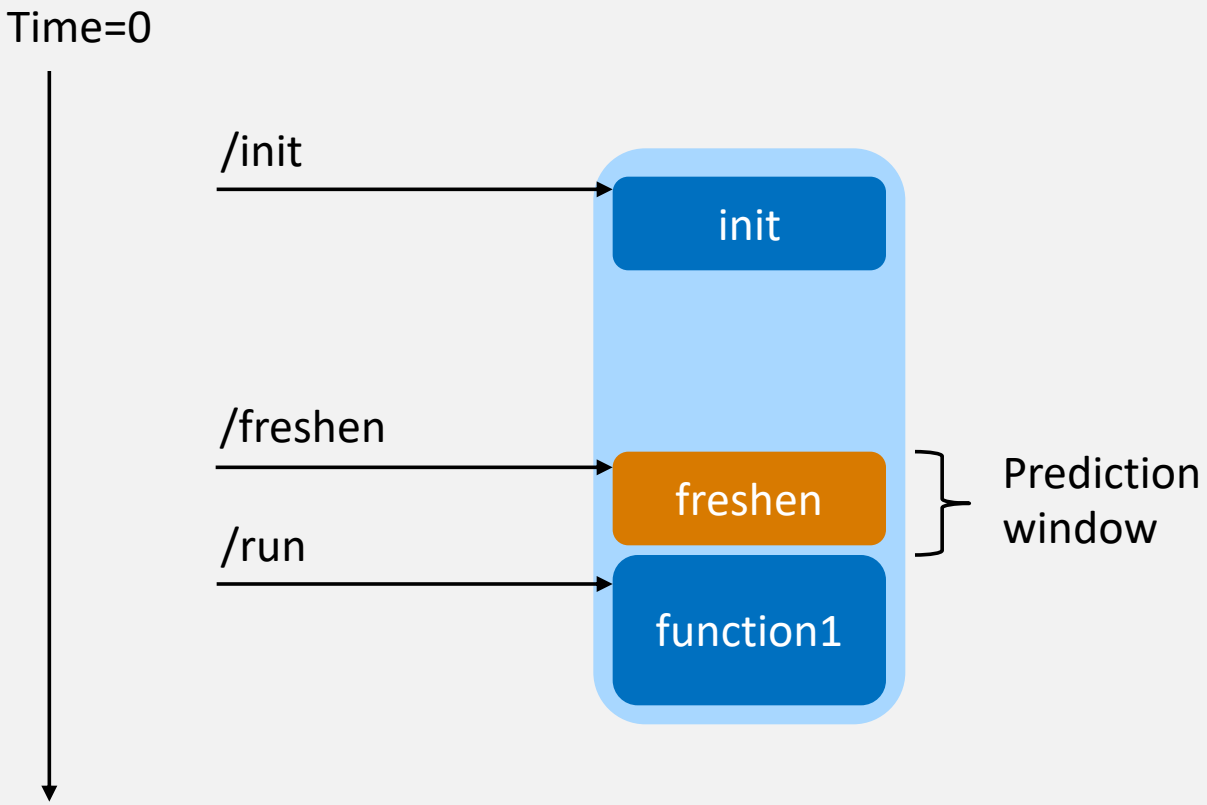
- Goals:
  - Build mechanism for dynamic resource management into serverless lifecycle
  - Reduce function latency
  - Dynamically manage state without cluttering function code



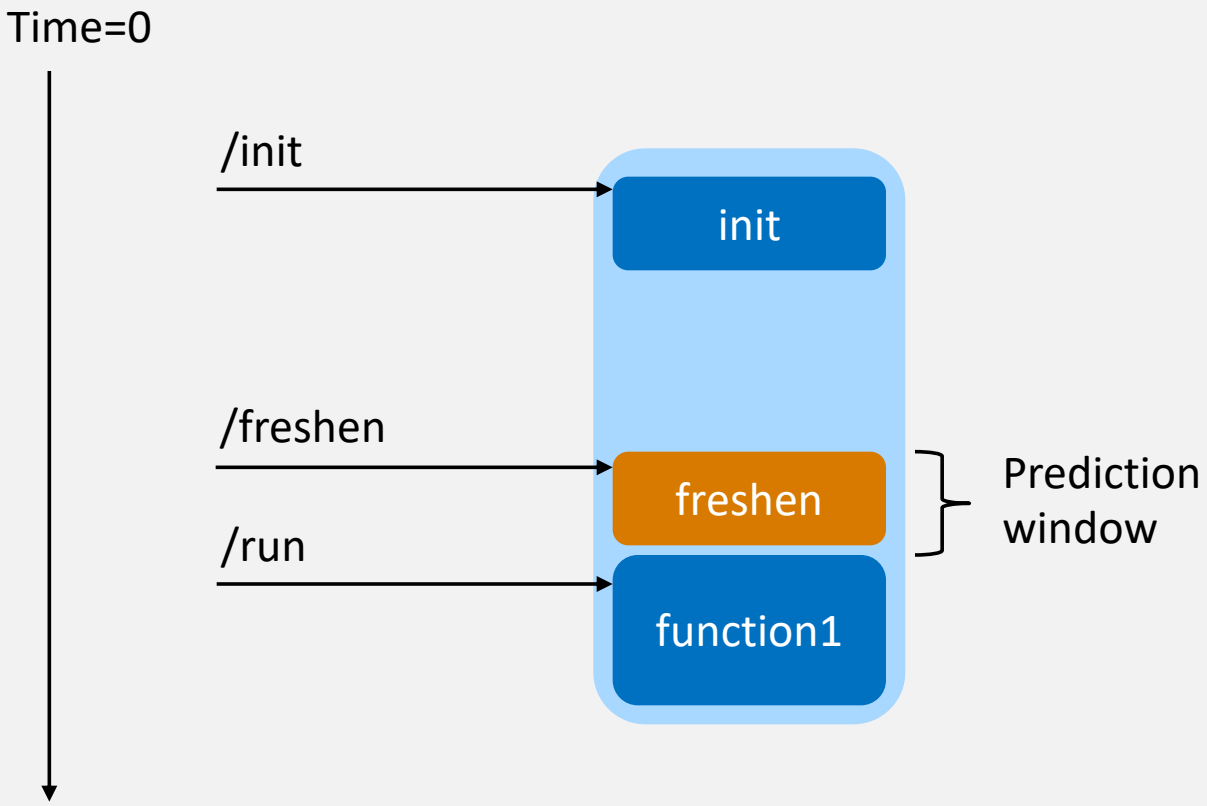
Freshen Design



Freshen Design



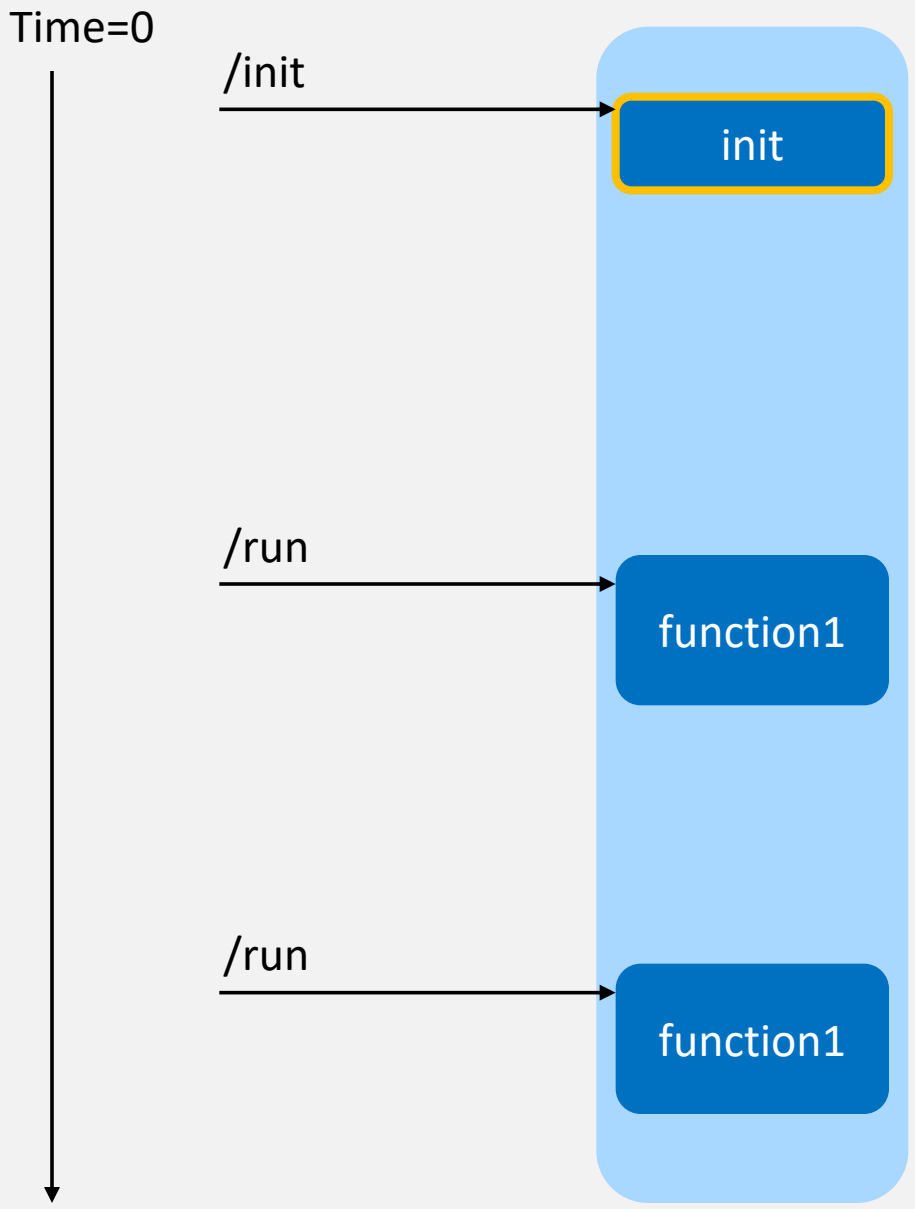
Freshen Design



Policy Options:

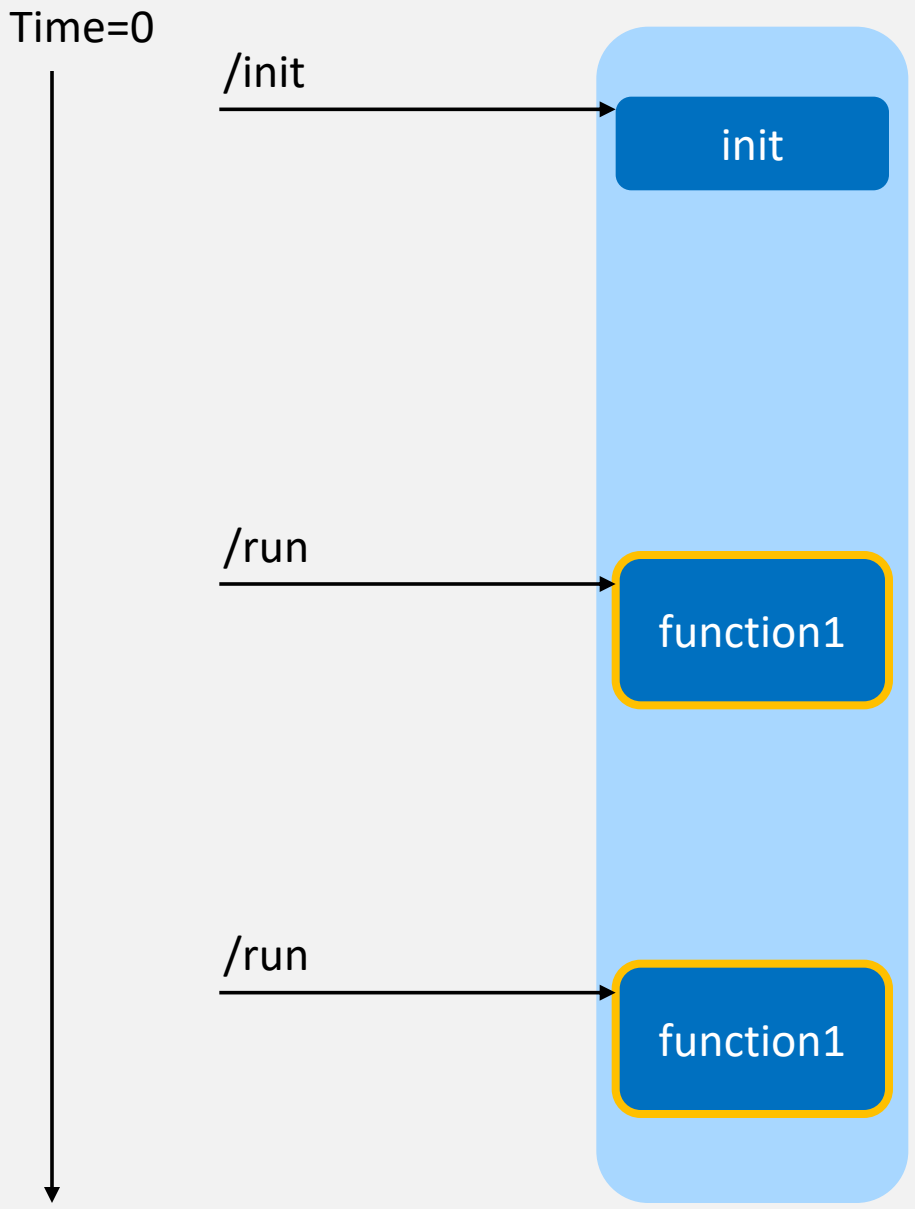
- Prediction
- Concurrency
- Forced blocking

Freshen Design



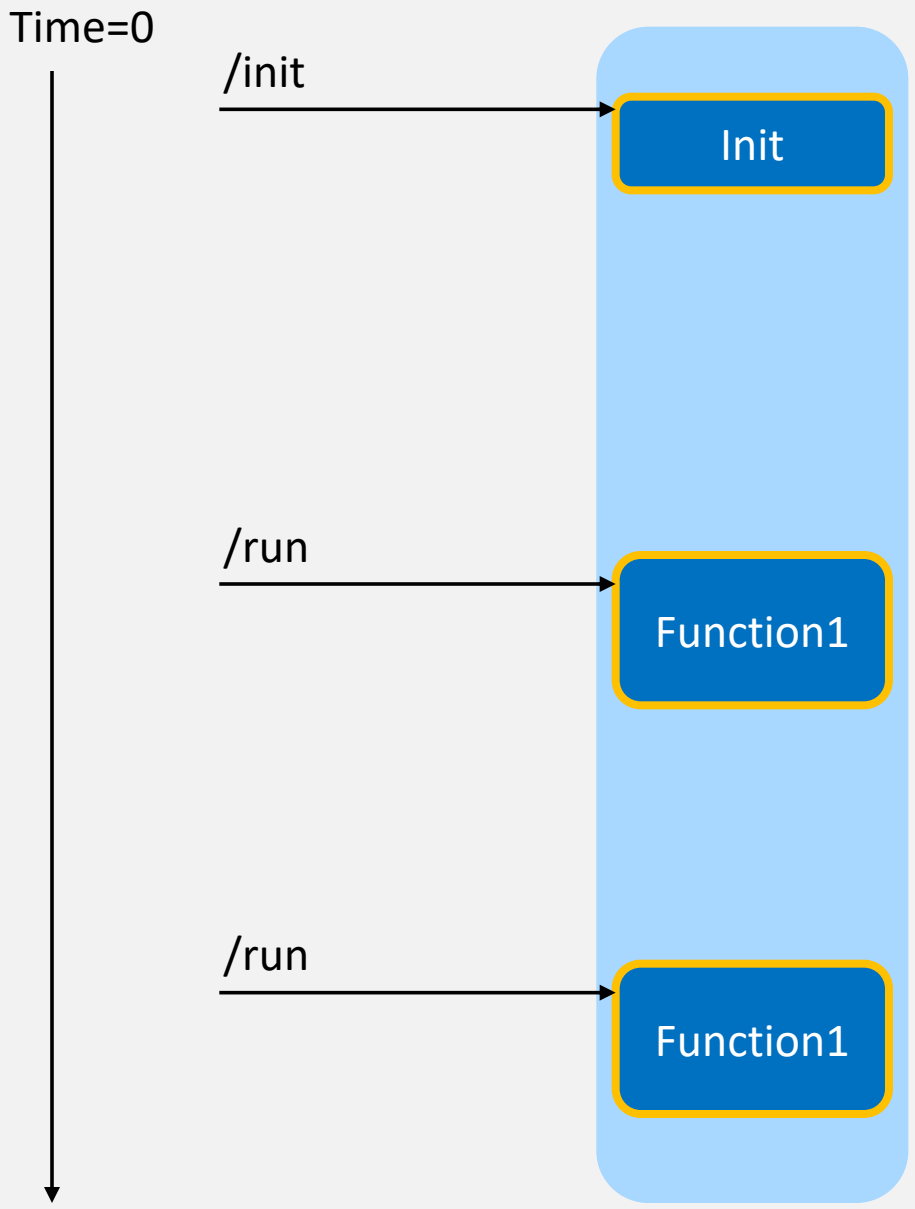
Serverless Resource Management Overview

	Reuse	Dynamic State	Proactive
Init Phase	✓	✗	✗
Function Code			
Runtime Reuse			
Freshen			



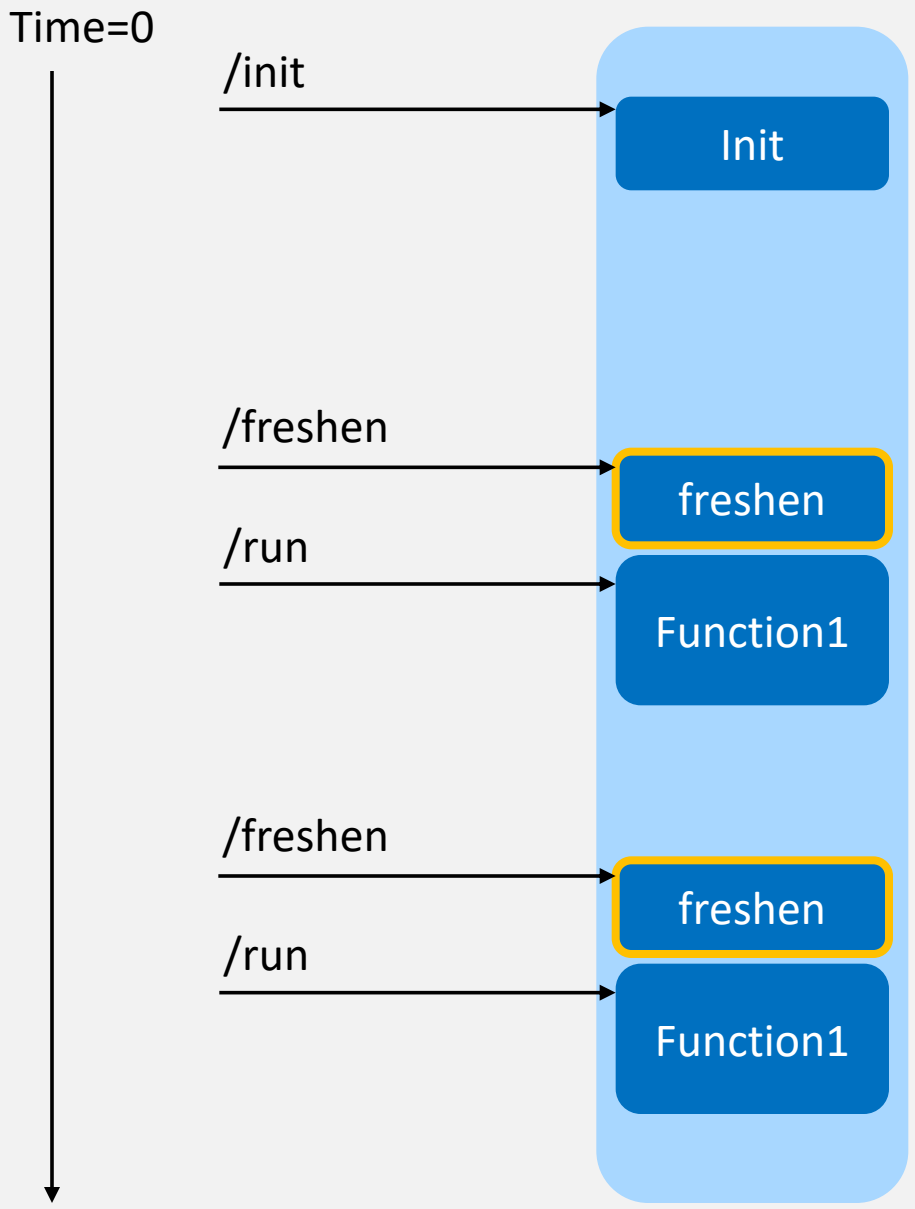
Serverless Resource Management Overview

	Reuse	Dynamic State	Proactive
Init Phase	✓	✗	✗
Function Code	✗	✓	✗
Runtime Reuse			
Freshen			



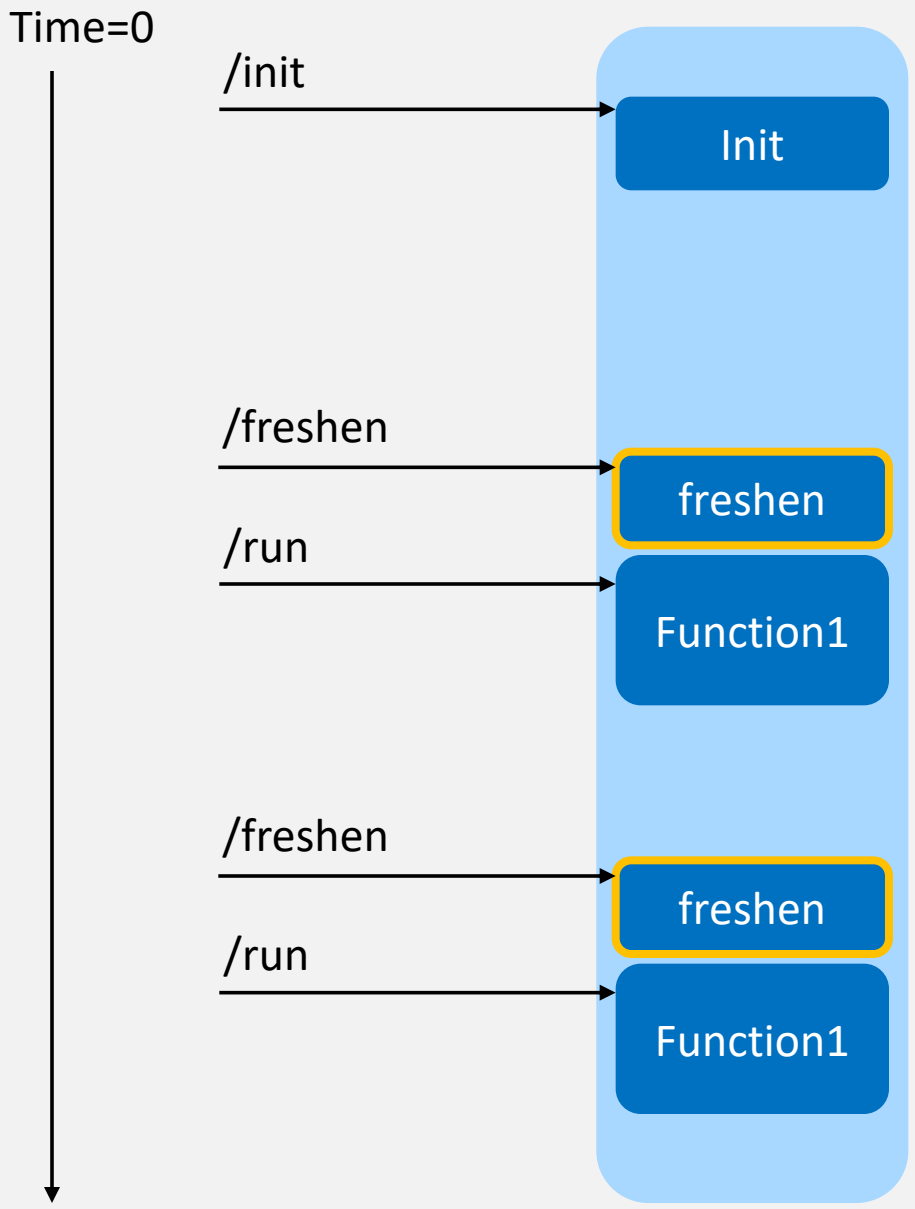
Serverless Resource Management Overview

	Reuse	Dynamic State	Proactive
Init Phase	✓	✗	✗
Function Code	✗	✓	✗
Runtime Reuse	✓	✓	✗
Freshen			



Serverless Resource Management Overview

	Reuse	Dynamic State	Proactive
Init Phase	✓	✗	✗
Function Code	✗	✓	✗
Runtime Reuse	✓	✓	✗
Freshen	✓	✓	✓



Serverless Resource Management Overview

	Reuse	Dynamic State	Proactive
Init Phase	✓	✗	✗
Function Code	✗	✓	✗
Runtime Reuse	✓	✓	✗
Freshen	✓	✓	✓

By **proactively** supporting **reuse** and **dynamic state maintenance**, freshen removes the latency cost of those tasks from function execution.

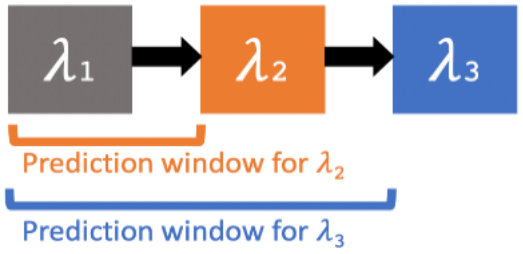
Serverless Function Prediction

Prediction useful for many reasons (scheduling, resource utilization, coldstart avoidance, etc.)

Serverless Function Prediction

Prediction useful for many reasons (scheduling, resource utilization, coldstart avoidance, etc.)

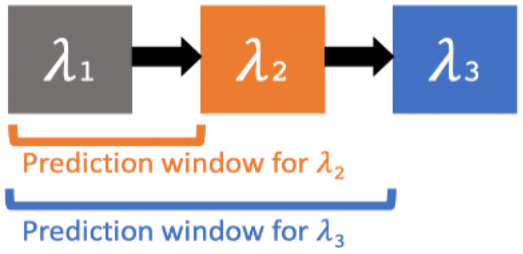
Some cases are easier to predict, e.g., chained functions



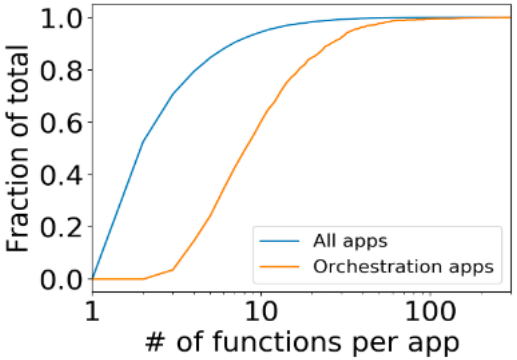
Serverless Function Prediction

Prediction useful for many reasons (scheduling, resource utilization, coldstart avoidance, etc.)

Some cases are easier to predict, e.g., chained functions



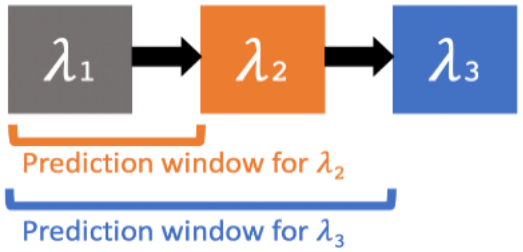
Many applications consist of multiple functions



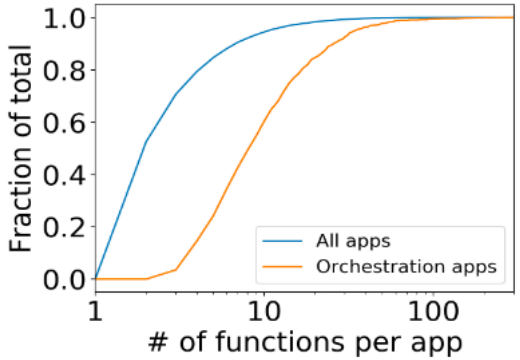
Serverless Function Prediction

Prediction useful for many reasons (scheduling, resource utilization, coldstart avoidance, etc.)

Some cases are easier to predict, e.g., chained functions



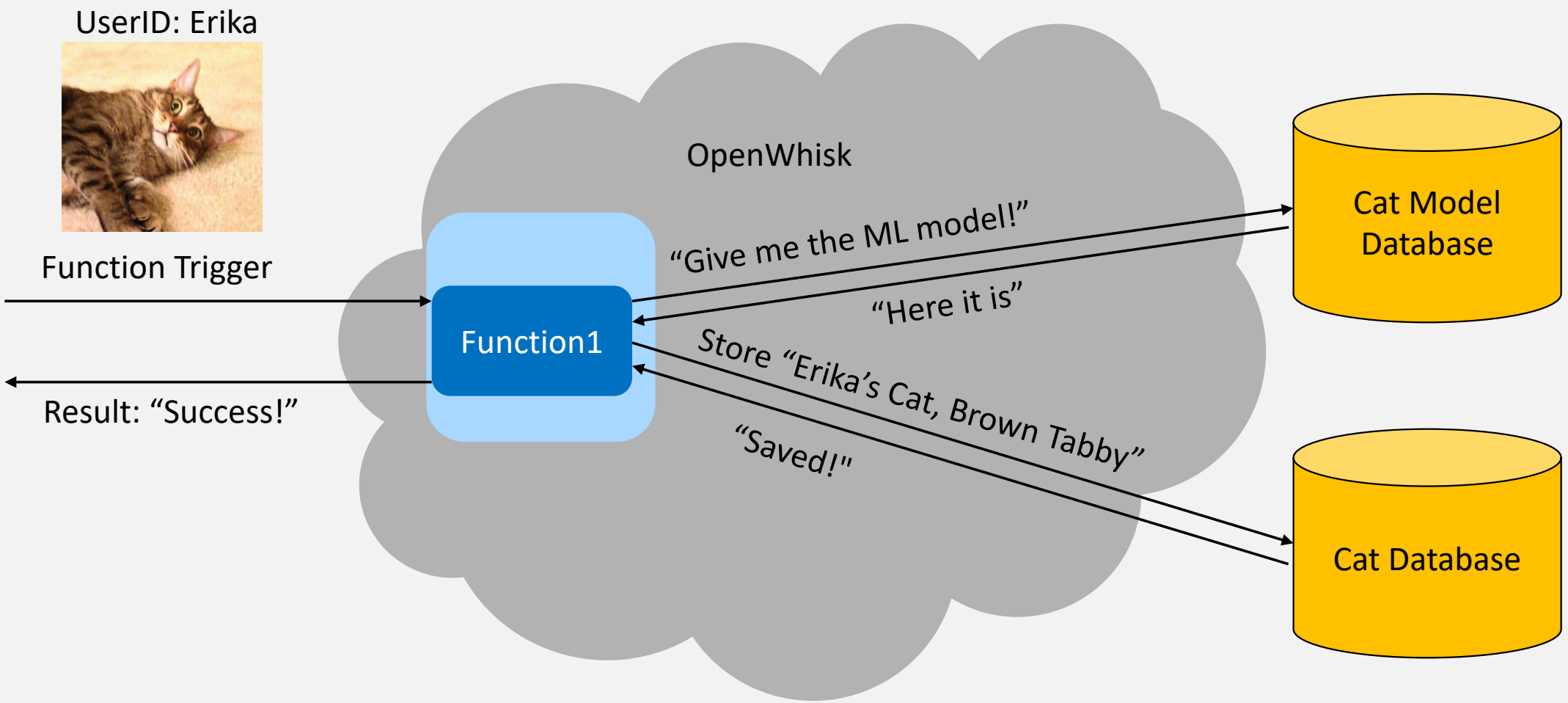
Many applications consist of multiple functions



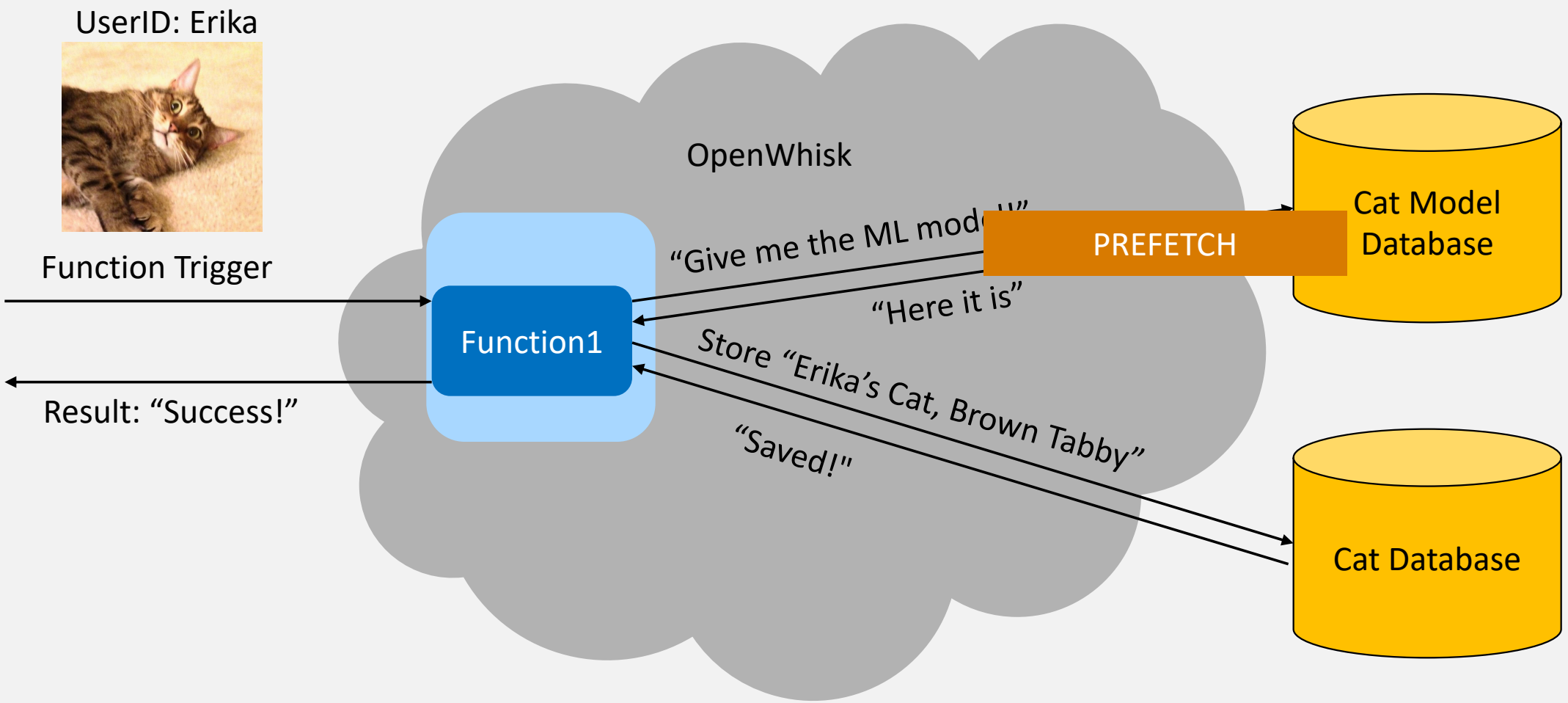
May be infrastructure overheads

Trigger Service	Delay (s)
Step Functions	0.064
Direct (Boto3)	0.060
SNS Pub/Sub	0.253
S3 bucket	1.282

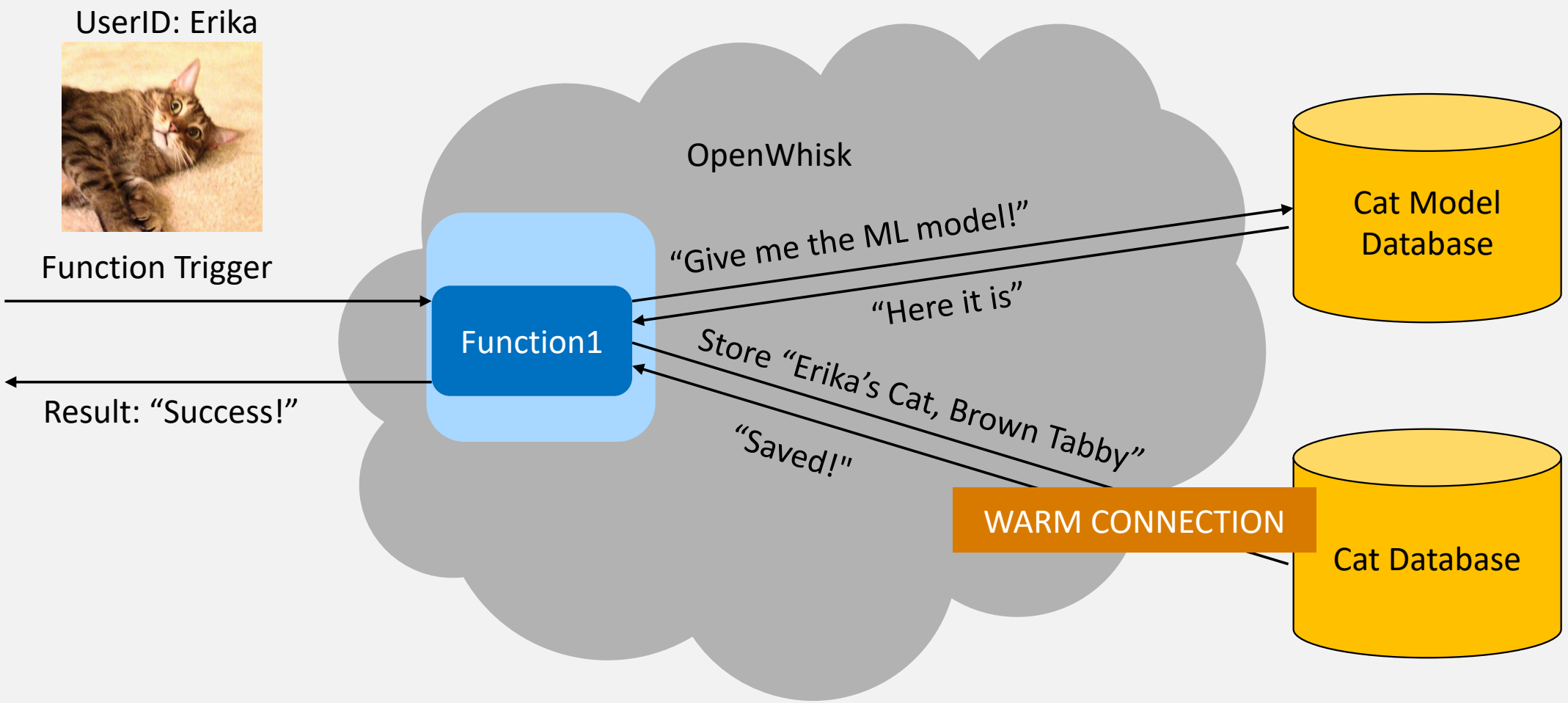
What Can ***Freshen*** Do?



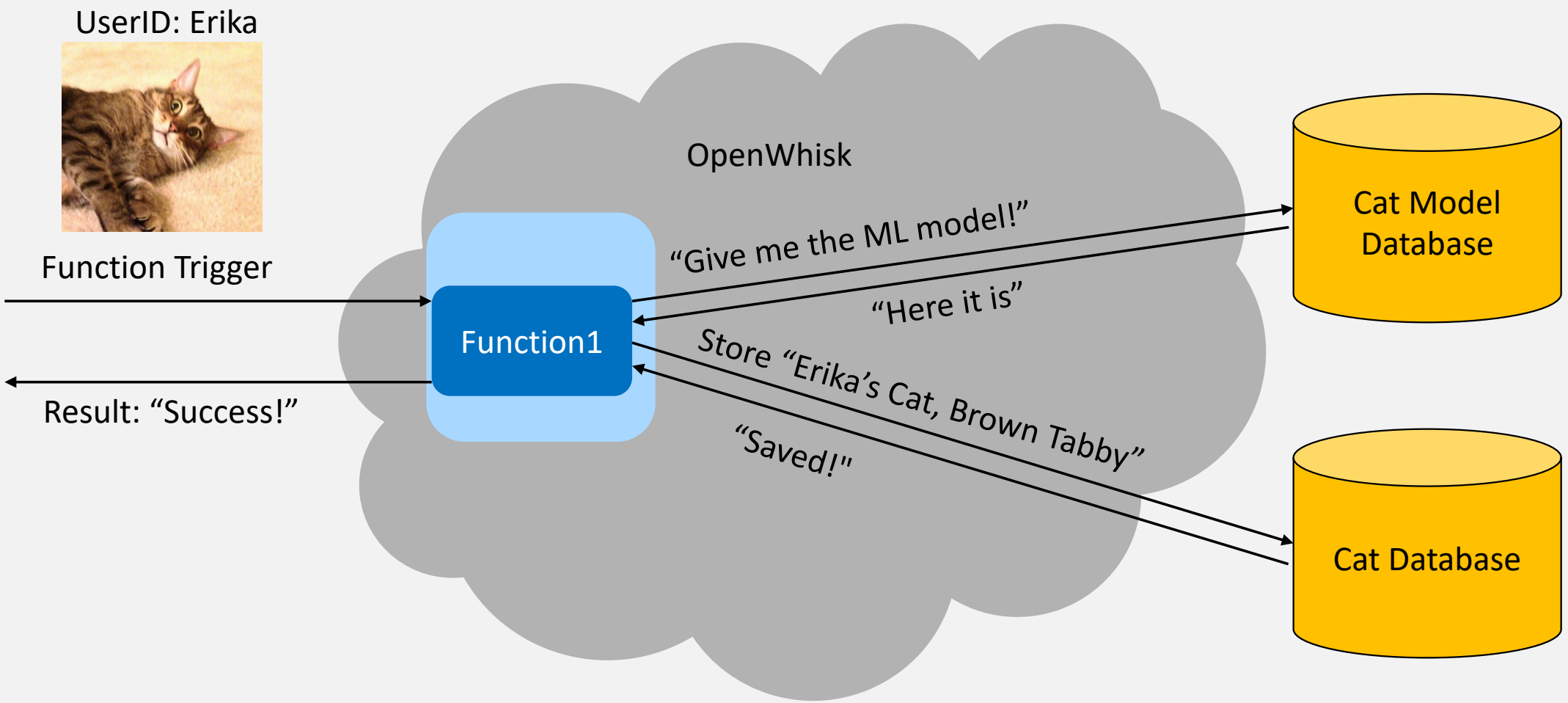
What Can ***Freshen*** Do?



What Can ***Freshen*** Do?



What Can ***Freshen*** Do?



What Can ***Freshen*** Do?

MORE?

Background

Freshen Design

Evaluation

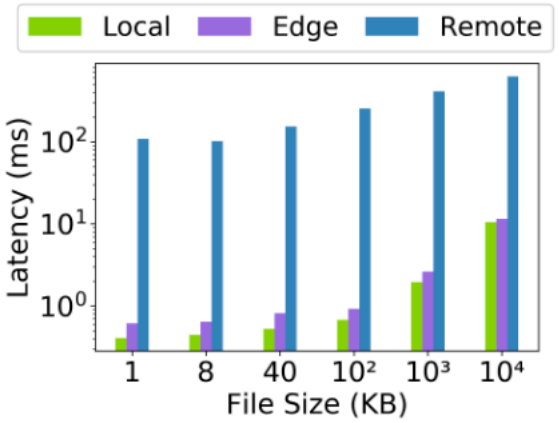
Discussion

Questions

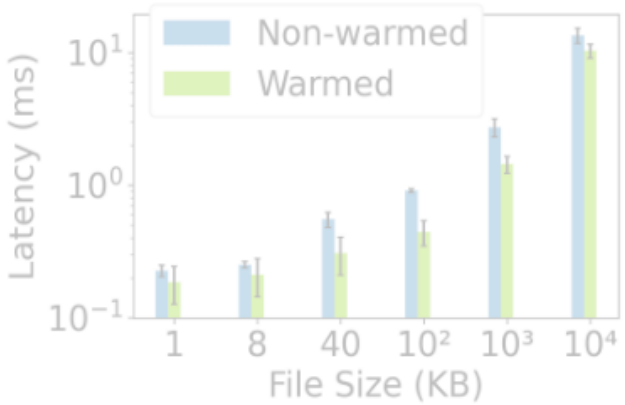
Outline

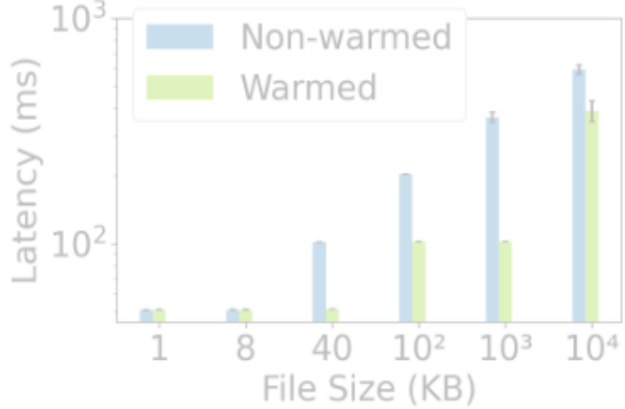
Freshen Motivation

PREFETCH



WARM CONNECTION

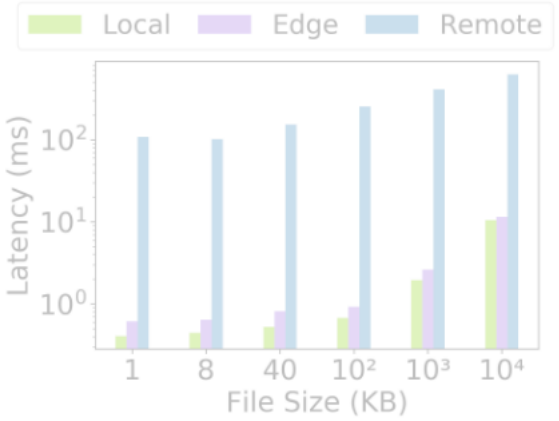




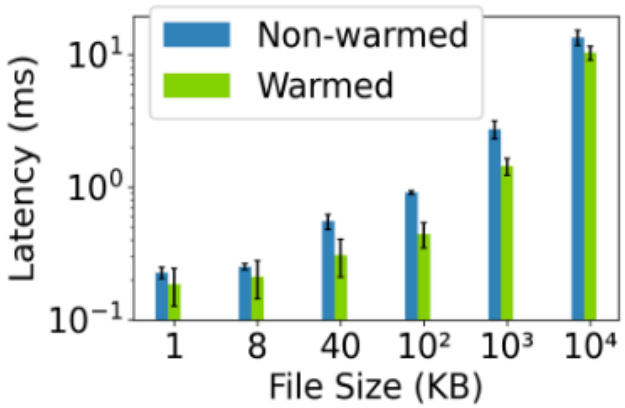
Reduces latency to access data

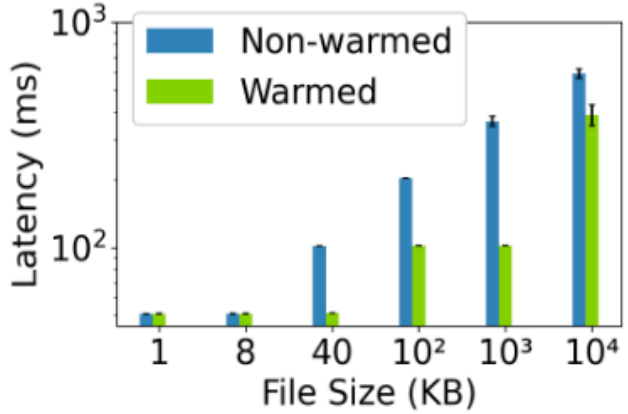
Freshen Motivation

PREFETCH



WARM CONNECTION





Warmed TCP connections send traffic more efficiently

Background

Freshen Design

Evaluation

Discussion

Questions

Outline

Discussion

- Connection state manipulation
- Function prediction
- Who is responsible for ***freshen***?
- Other ***freshen*** actions

How to access connection state?

Beyond TCP?

Discussion

Connection state manipulation

Function prediction

Who is responsible for ***freshen***?

Other ***freshen*** actions

Discussion

Connection state manipulation

Function prediction

Who is responsible for ***freshen***?

Other ***freshen*** actions

Developer written

Libraries

Inference

Discussion

Connection state manipulation

Function prediction

Who is responsible for ***freshen***?

Other ***freshen*** actions

Memory allocation?

Prepopulating Caches?

Things we have not yet thought of?

Background

Freshen Design

Evaluation

Discussion

Questions

- We propose a new serverless runtime primitive, ***freshen***, as a mechanism to enable proactive serverless function resource management.

Erika Hunhoff
erika.hunhoff@colorado.edu

https://www.serverlesscomputing.org/wosc6/#p11