Knative Vs. OpenFaaS

Functions on Kubernetes

Carson Anderson

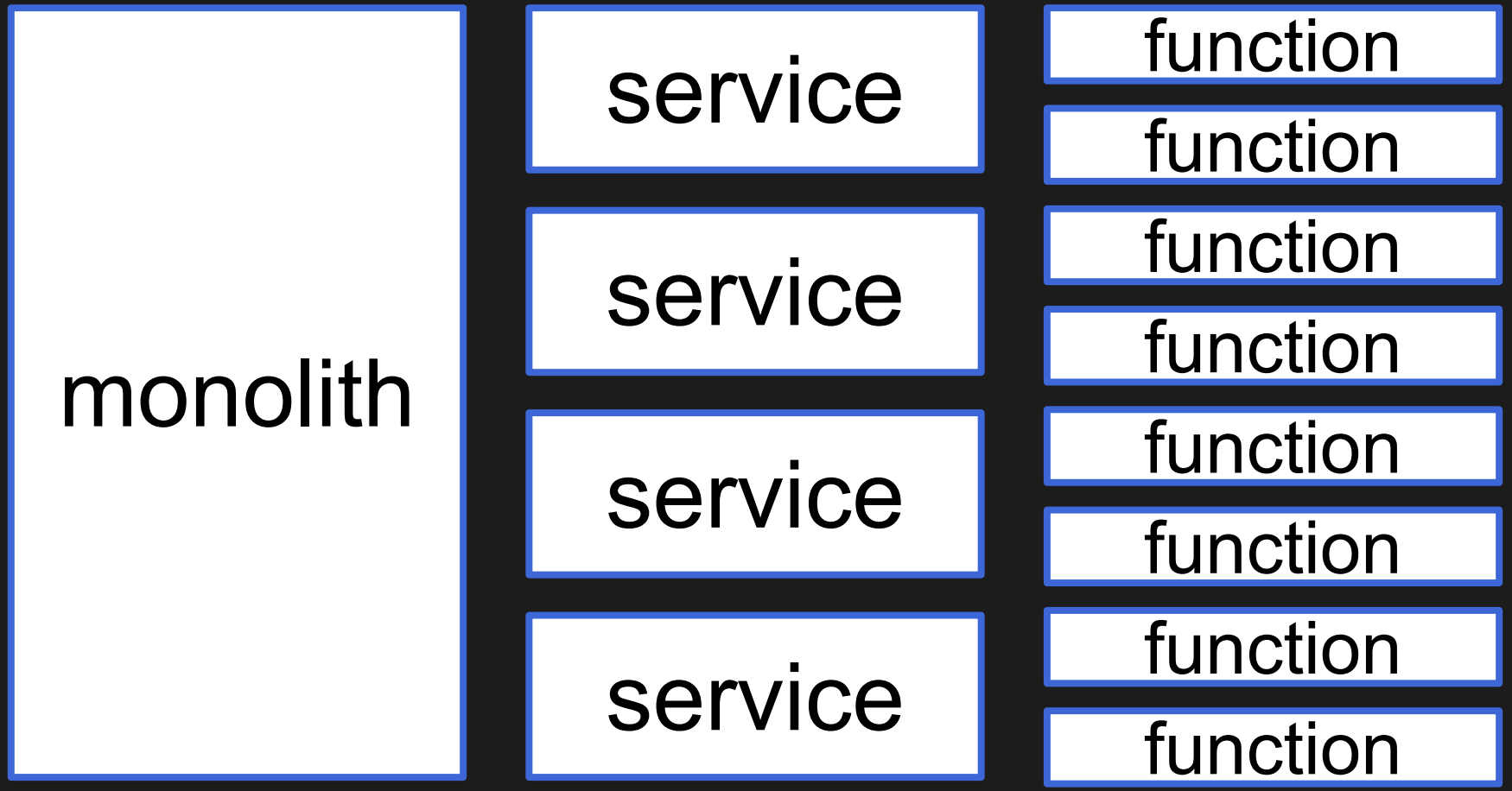
Domo

 @carsonoid

 @carson_ops

Kubernetes & Serverless

- Kubernetes **is** Serverless
- A layer to reduce **developer** complexity
- Functions As A Service narrows the scope of
  - Work
  - Risk
  - Scale


monolith
service
service
service
service
function
function
function
function
function
function
function
function

Let's Come Clean

- Knative is not AWS Lambda
- OpenFaaS is not AWS Lambda
- If you are in AWS and want top quality integration: Use Lambda!

But!

- AWS Lamba != perfect
  - Limits
    - Execution time
    - Persistent disk limits
  - Vendor lock-in
  - VM runtime environment

Lambda Vs (Knative Vs OpenFaaS)

Functions on Things!

Carson Anderson

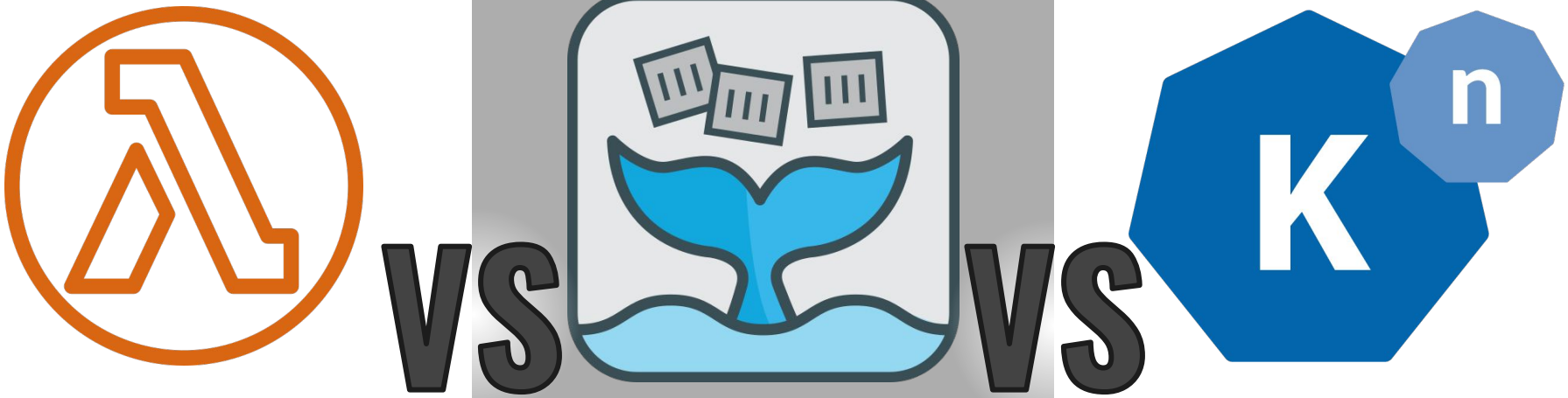
Domo

@carsonoid

@carson_ops

Plan

- Introductions
- Common Components
- Demo
- Platform Comparisons
  - Architecture
  - Installation
- Function Operations
  - Building
  - Deploying
  - Invoking



Introductions

AWS Managed

- AWS managed
- AWS scale
- AWS integrated
- No overhead

OpenFaaS

- Independent
- Open source
- Portable
- Lightweight

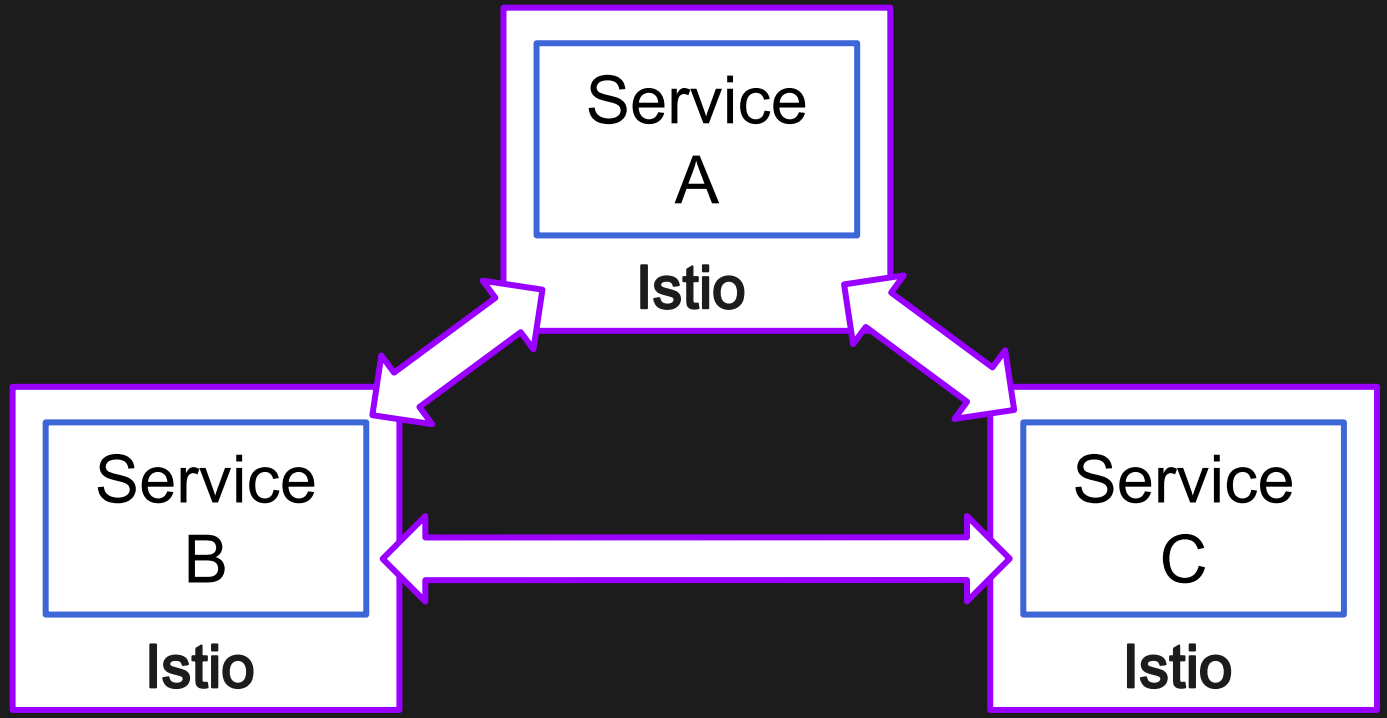
Knative Introduction

- Google backed
- Serverless platform
  - Building
  - Serving
  - Eventing

Knative Caveat

Knative Requires Istio

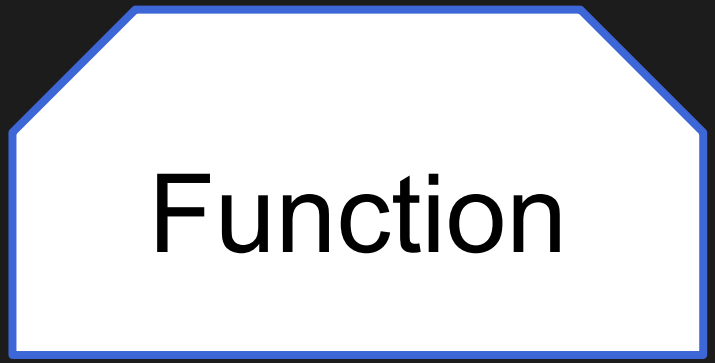
What is a service mesh?



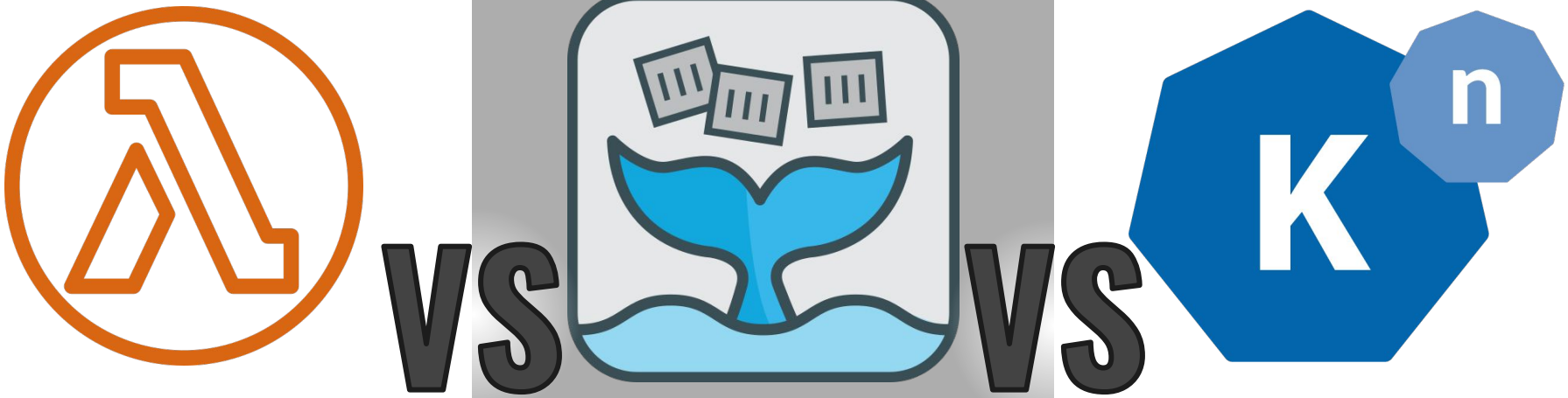
Common Components

1. Function
2. Invoker
3. Queue
4. Ringleader
5. Usher

1. Function



- Imperative
  - Returns something
  - Called by user
  - Called by other function
  - Called by other system
- Reactive
  - Triggered by outside event



Functions

Template: get_tax()

```
# EXPECTS = {"subtotal":INT}
def main():
  subtotal = [IN]

  tax = subtotal * 0.047

  [OUT] { "tax": tax }
```

Implementation Specifics In Red

Lambda: get_tax()

```
# EXPECTS {"subtotal":INT}
def main(event, context):
  subtotal = event["subtotal"]

  tax = subtotal * 0.047

  return {
    'statusCode': 200,
    'body': { "tax": tax }
  }
```

Implementation Specifics In Red

OpenFaaS: get_tax()

```
import sys, json

# EXPECTS {"subtotal":INT}
def handle(req):
  subtotal = json.loads(req)["subtotal"]

  tax = subtotal * 0.047

  print({ "tax": tax })
```

Implementation Specifics In Red

Knative: get_tax()

```
import os, json
from flask import Flask, request

app = Flask(__name__)

@app.route('/', methods=['POST'])
# EXPECTS {"subtotal":INT}
def main():
  subtotal = request.get_json()["subtotal"]
  tax = subtotal * 0.047
  return str(tax)

if __name__ == '__main__':
  app.run(debug=True,host='0.0.0.0',port=8080)
```

Implementation Specifics In Red

Reactive Functions

Template: post_message()

```
import requests

# EXPECTS {"title":STR, "txt":STR}
def main([...]):
  url   = "https://webhook.site/..."
  title = [IN]
  txt   = [IN]

  message = title + "\n" + txt

  requests.post(url, data = message)
```

Implementation Specifics In Red

Lambda: post_message()

```
import requests

# EXPECTS {"title":STR, "txt":STR}
def main(event, context):
  url = "https://webhook.site/..."
  title = event["title"]
  txt   = event["txt"]

  message = data["title"] + "\n" + data["txt"]

  requests.post(url, data = message)
```

Implementation Specifics In Red

OpenFaaS: post_message()

```
import requests
import sys, json

# EXPECTS {"title":STR, "txt":STR}
def main(req):
  url = "https://webhook.site/..."
  data = json.loads(req)
  title = data["title"]
  txt   = data["txt"]

  message = data["title"] + "\n" + data["txt"]

  requests.post(url, data = message)
```

Implementation Specifics In Red

Knative: post_message()

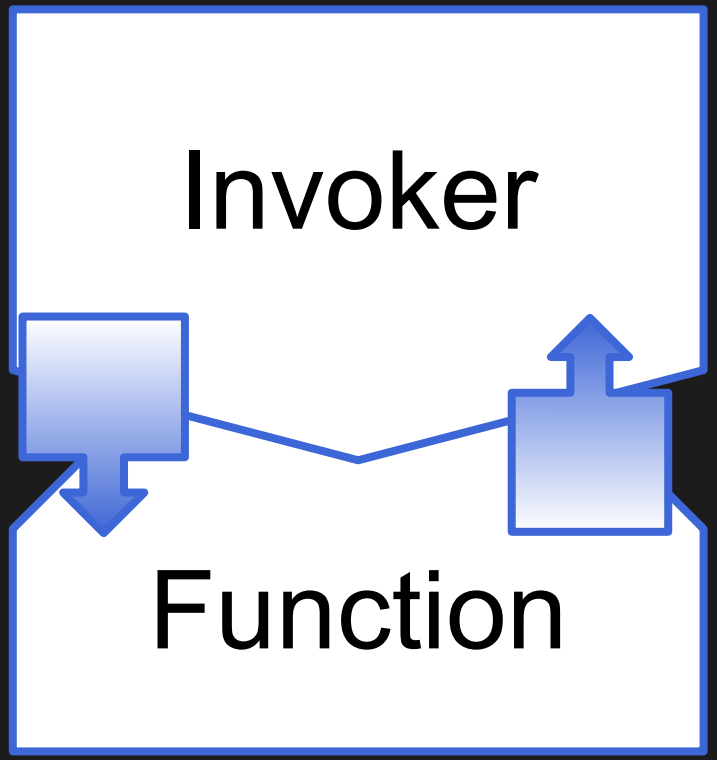
```
import os, json
import requests
from flask import Flask, request

app = Flask(__name__)

@app.route('/', methods=['POST'])
# EXPECTS {"title":STR, "txt":STR}
def main():
  url = 'https://webhook.site/...'
  data = request.get_json()
  title = data["title"]
  txt   = data["txt"]

  message = title + "\n" + txt

  requests.post(url, data = message)

if __name__ == '__main__':
  app.run(debug=True,host='0.0.0.0',port=8080)
```

Implementation Specifics In Red

Common Components

1. Function
2. Invoker
3. Queue
4. Ringleader
5. Usher

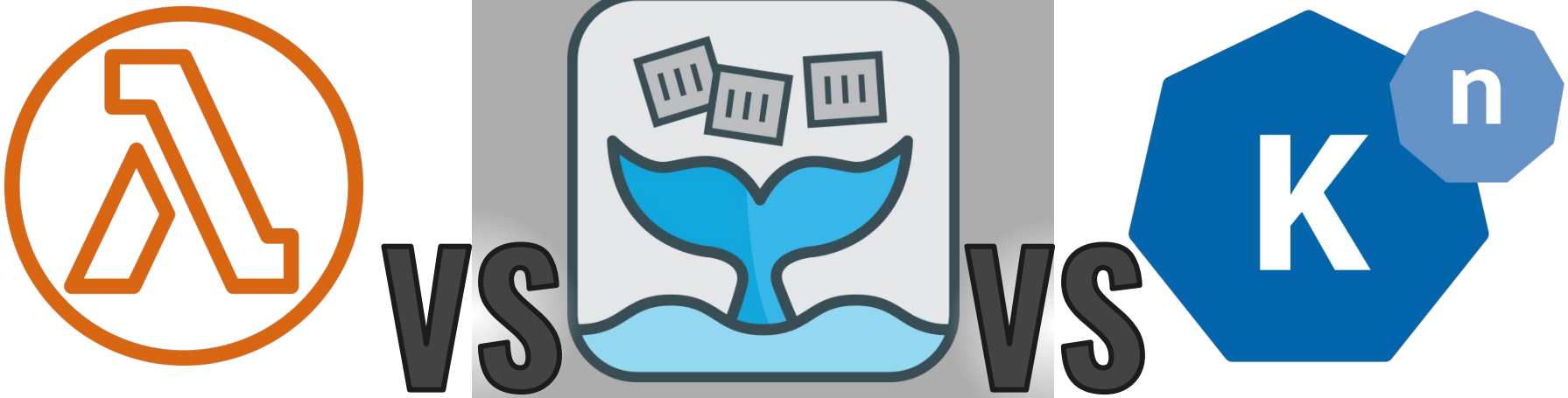
2. Invoker



- Forks or calls the function
- Handles
  - Inputs
  - Outputs
- Usually
  - HTTP server or client
  - Customizable or replaceable

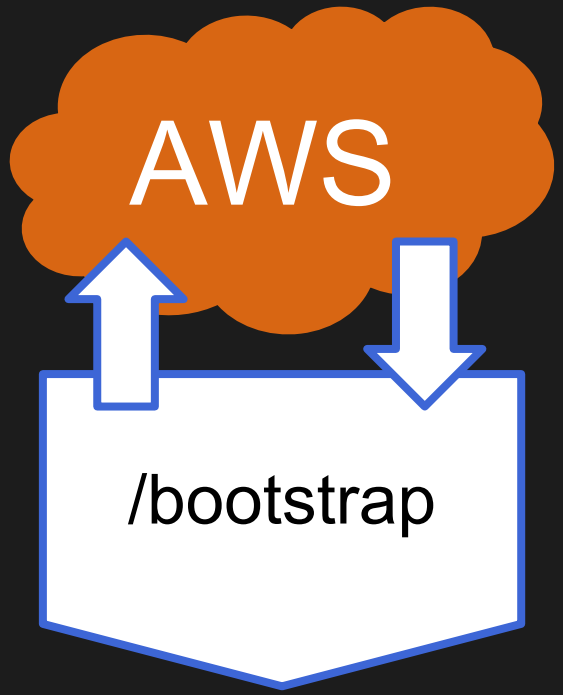
Invokers Matter!

- Synchronous or asynchronous
- Serial or parallel
- For serial
  - Is the execution environment recreated or sanitized for each invocation?
  - Is the execution called via forking or threads?
    - Global variables
    - Connection pools
- For parallel
  - Concurrency?



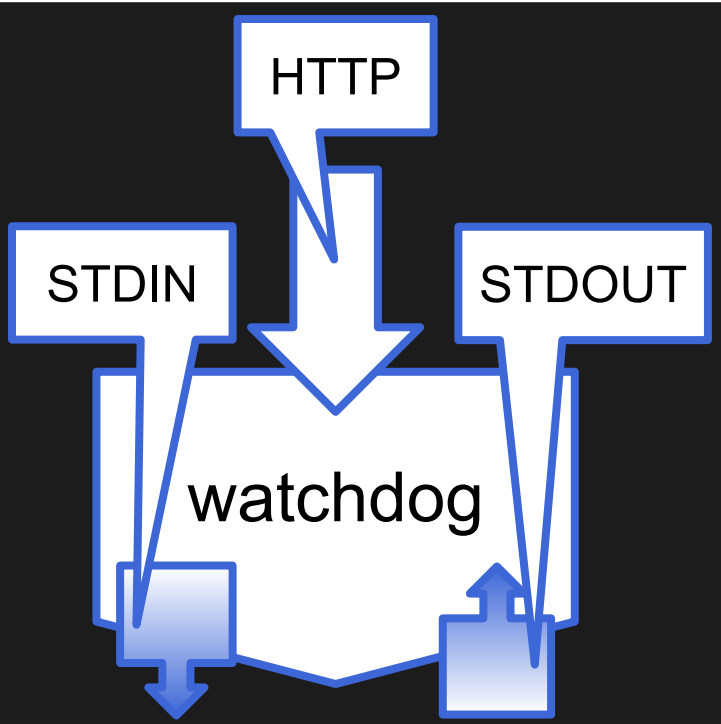
Invokers

/bootstrap script



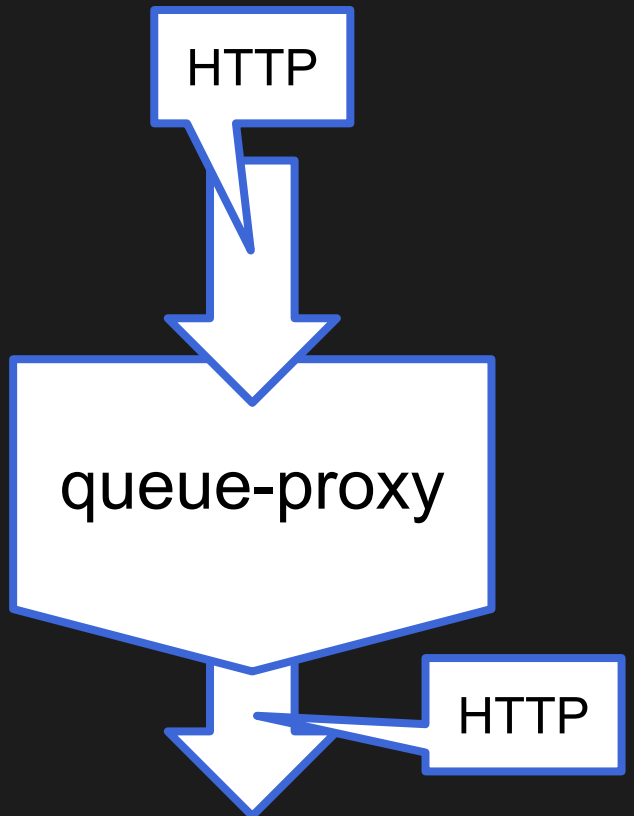
- Method
  - Read events from a "magic" API
  - Call function with event data
  - Return results to API
- Serialized executions
- Invocations run as threads
- Runtimes come with one by default
- Replaceable

watchdog, of-watchdog



- Method
  - Synchronous, parallel
    - Listen for HTTP invocations
    - Fork function code
    - Send STDIN, read STDOUT
  - Asynchronous, serial
    - Read events from queue
    - Fork function code
    - Send STDIN, read STDOUT
- Replaceable, configurable

queue-proxy sidecar

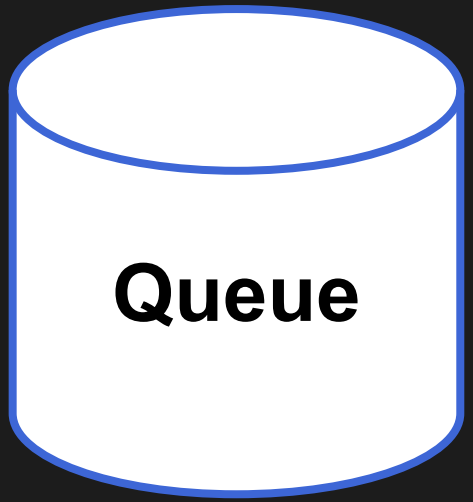


- Method
  - Intercept web traffic destined for function web server container
  - Queue, limit, reroute, trace, etc.
  - Call function web server container
- Serial or parallel based on configuration
- Not easily replaceable

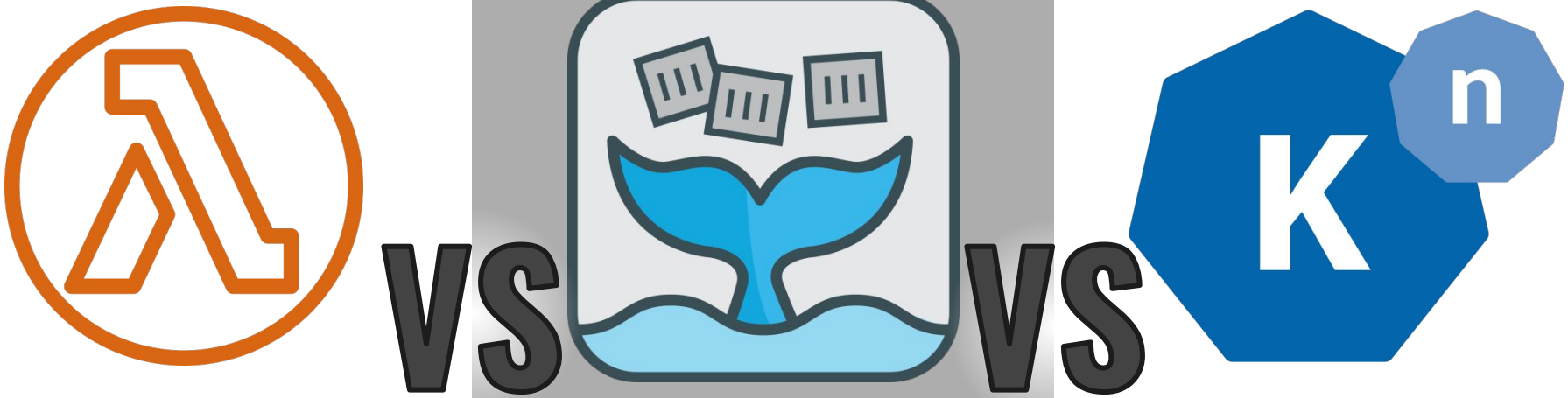
Common Components

1. Function
2. Invoker
3. Queue
4. Ringleader
5. Usher

3. Queue



- Stores
  - Async invocations
  - Pending invocations
- Tracks state
- Reports load



Queues

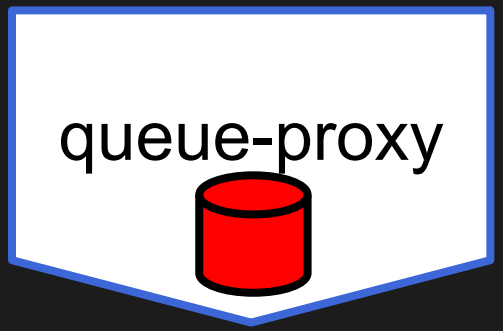
AWS Managed

- SQS?

NATS

- CNCF project
- Distributed
- Lightweight
- Scalable

queue-proxy

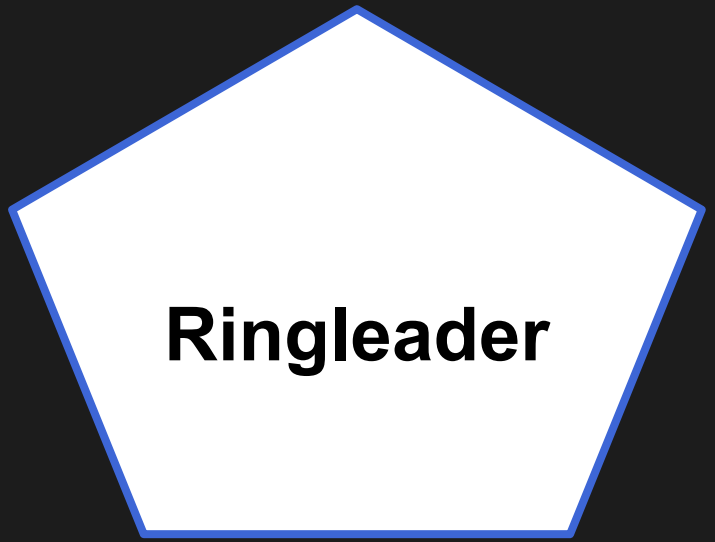


- In memory queue inside queue-proxy
- Distributed between all active function containers
- Not fault tolerant
- Not inspectable

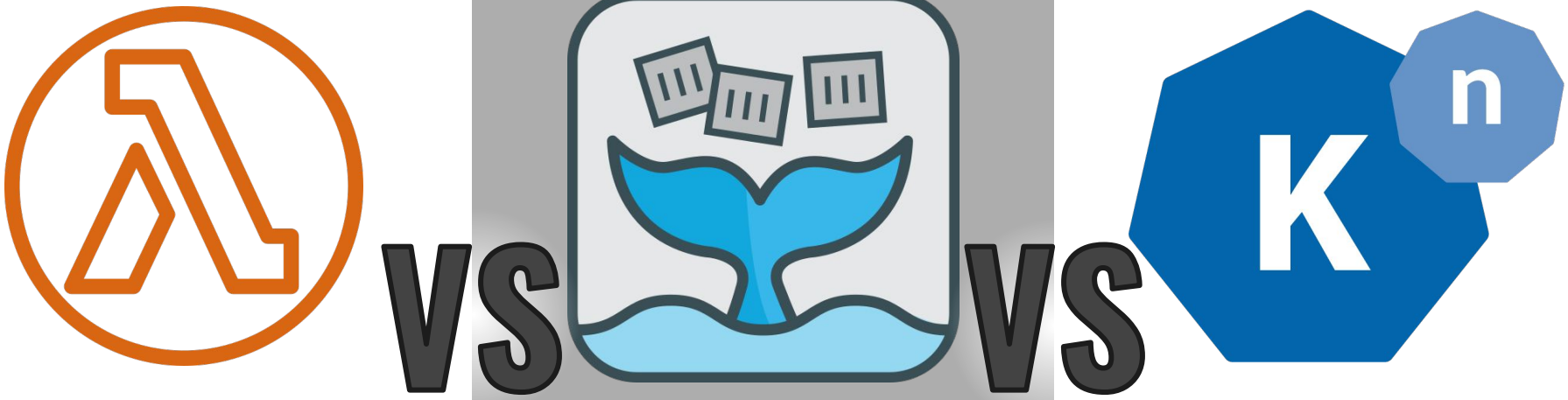
Common Components

1. Function
2. Invoker
3. Queue
4. Ringleader
5. Usher

4. Ringleader



- Creates
  - Functions
  - Infrastructure
- Scales functions
- Reports status



Ringleaders

AWS Managed

- API Gateway?

OpenFaaS Gateway

- Golang
- Open source
- Lightweight
- Full Featured

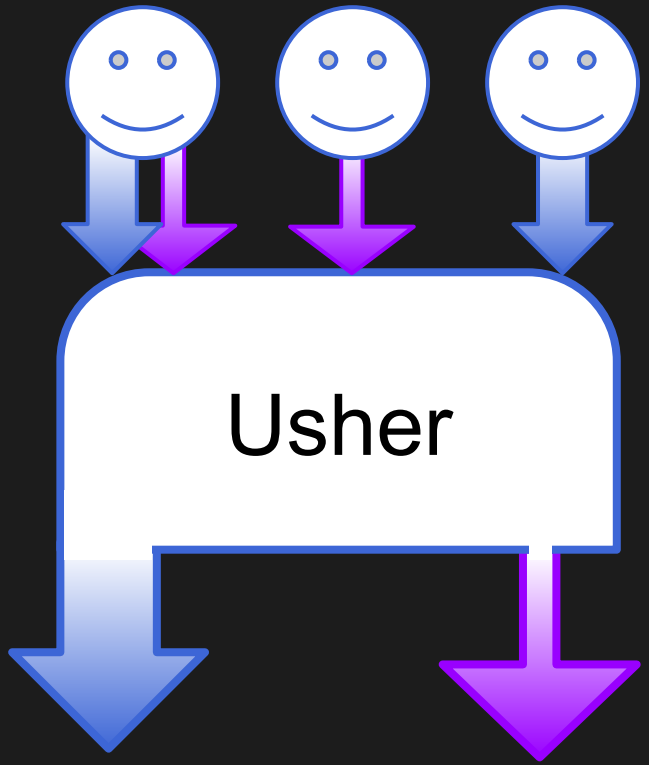
Knative Serving Controller

- Golang
- Open Source
- Follows the K8s "Operator" pattern

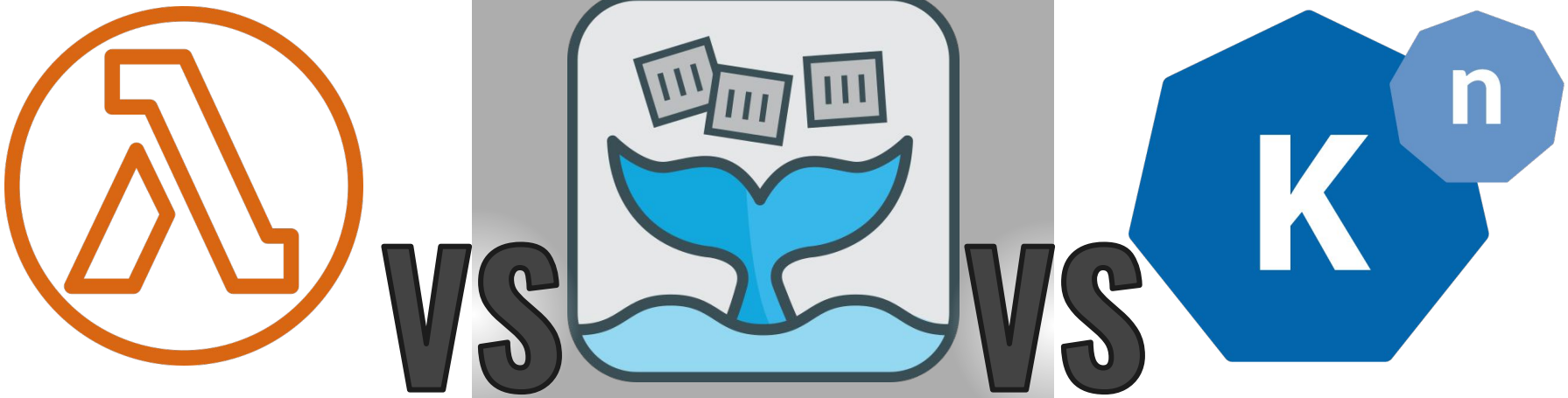
Common Components

1. Function
2. Invoker
3. Queue
4. Ringleader
5. Usher

6. Usher



- Takes in traffic for functions
- Routes requests
- May handle
  - Auth
  - Encryption



Usher

AWS Managed

- ELB?
- ALB?
- API Gateway?

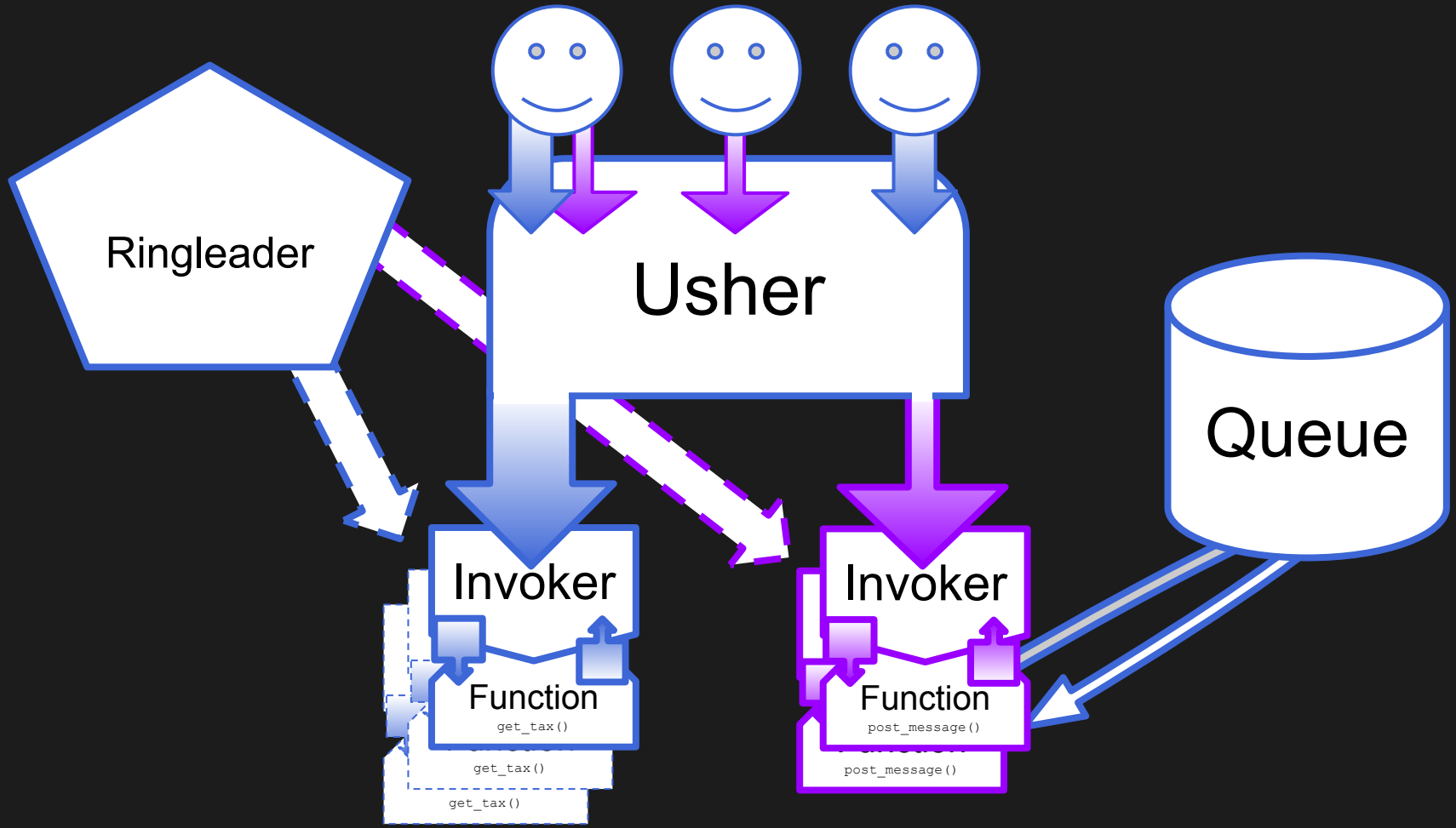
OpenFaaS Gateway

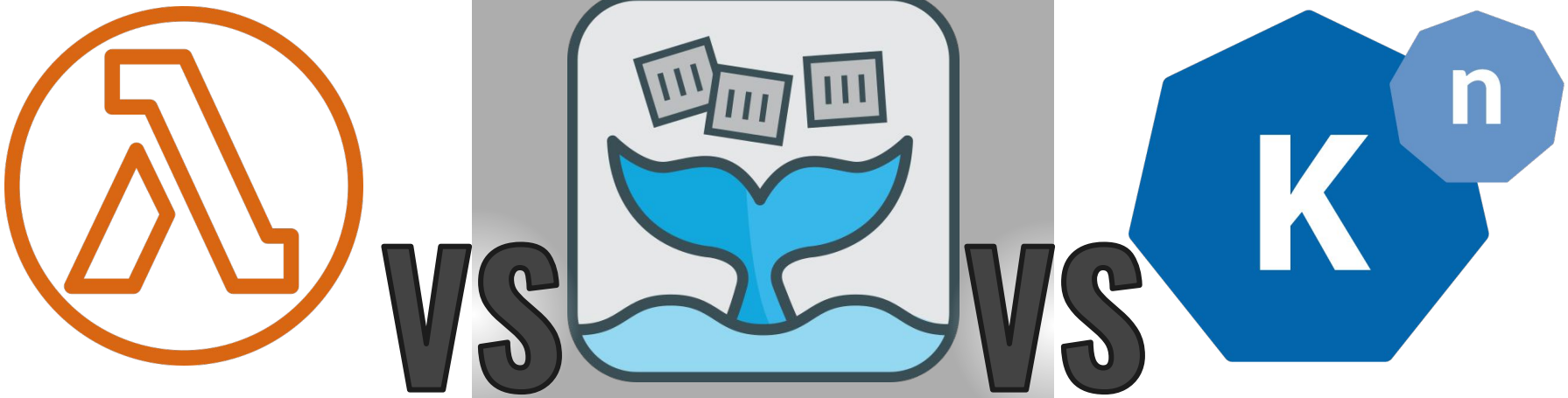
- Hosts
  - Admin GUI
  - Management API
- Proxies
  - Sync calls to function pods
  - Async calls to NATS queue

Istio Ingress Provider

istio-ingress/
Gloo

- Choices
  - istio-ingress
  - gloo
- Service mesh!


Ringleader
Usher
Queue
Invoker
Function
get_tax()
get_tax()
get_tax()
Invoker
Function
post_message()
post_message()

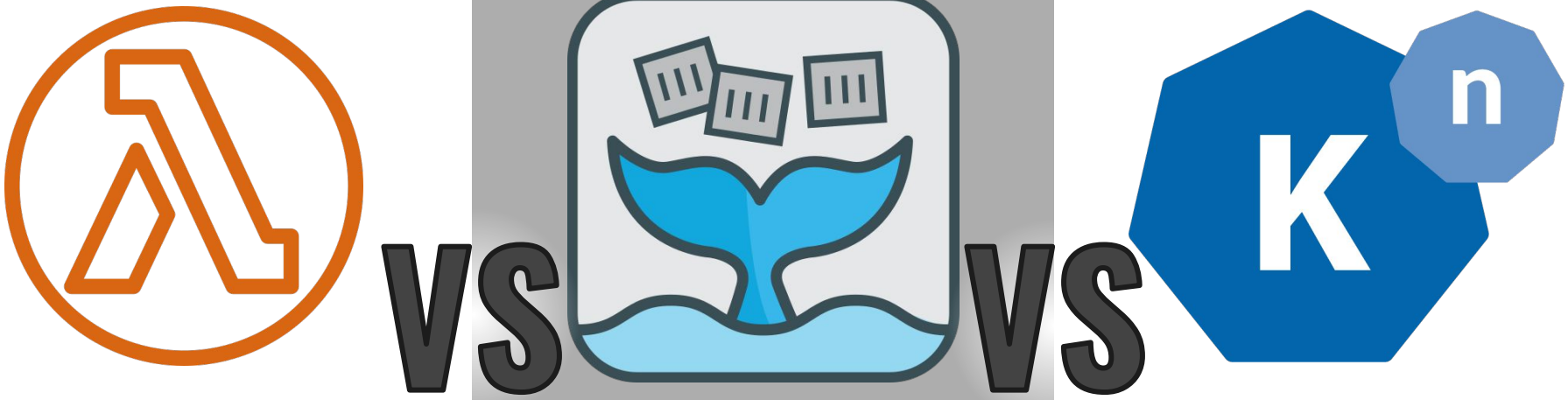


Demo

Why Only One Demo?

- OpenFaas
  - Lightweight
  - Quick to start
- Lambda
  - Account Required
- Knative
  - Does not work in k3s (no sidecar injection)

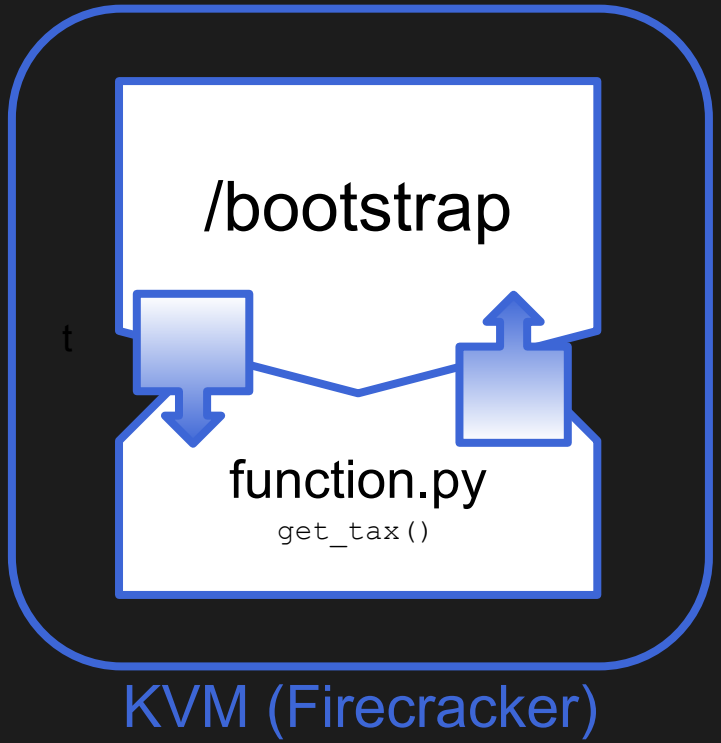
OpenFaaS Demo



Platform Architecture

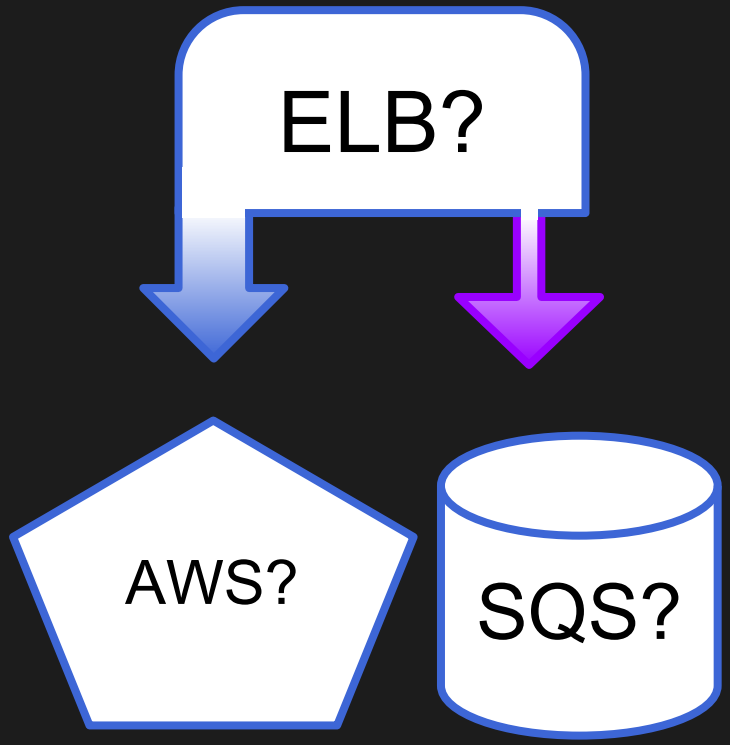
AWS Lambda Architecture

Function & Invoker

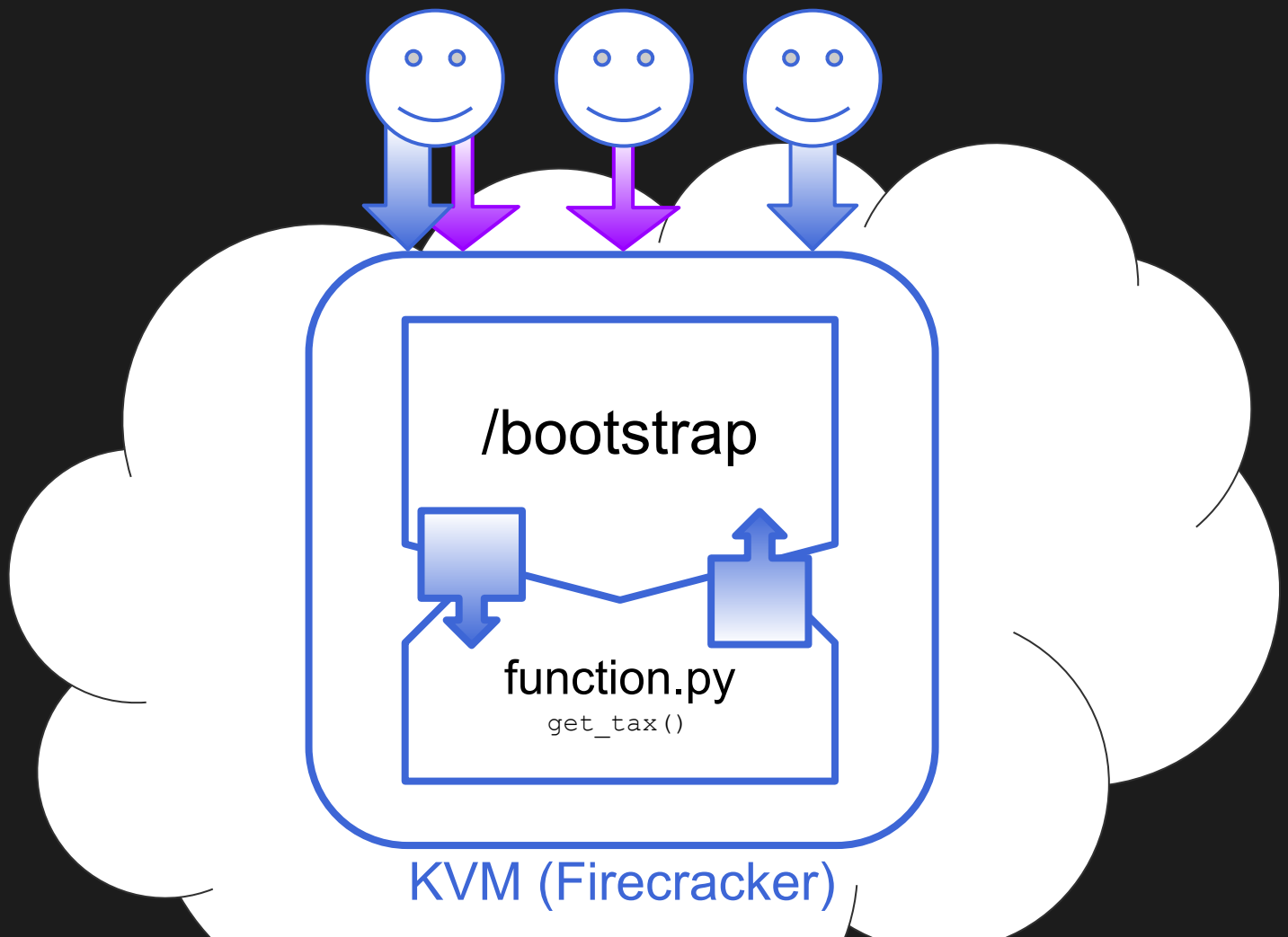


- NOT Containerized
- May need dedicated build vms
- Functions distributed as zip files built specifically for Amazon Linux
- Invocations are always serial
- A single VM may process many function invocations

Queue, Ringleader, & Usher



- ELB?
- SQS?
- AWS?
- MAGIC?


/bootstrap
function.py
get_tax()
KVM (Firecracker)

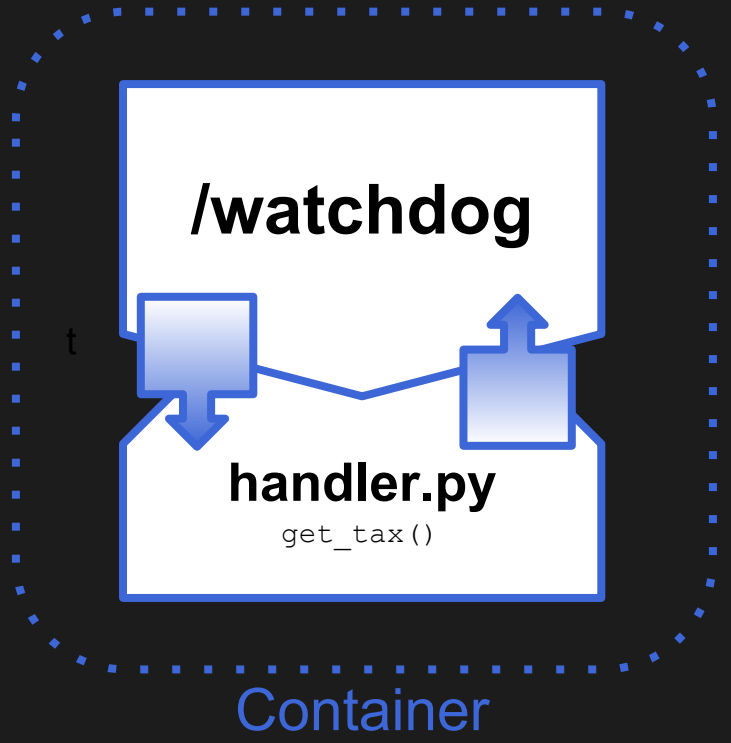
More Lambda Info

https://www.youtube.com/watch?v=eOBq__h4OJ4

http://amzn.to/2i1K7cE

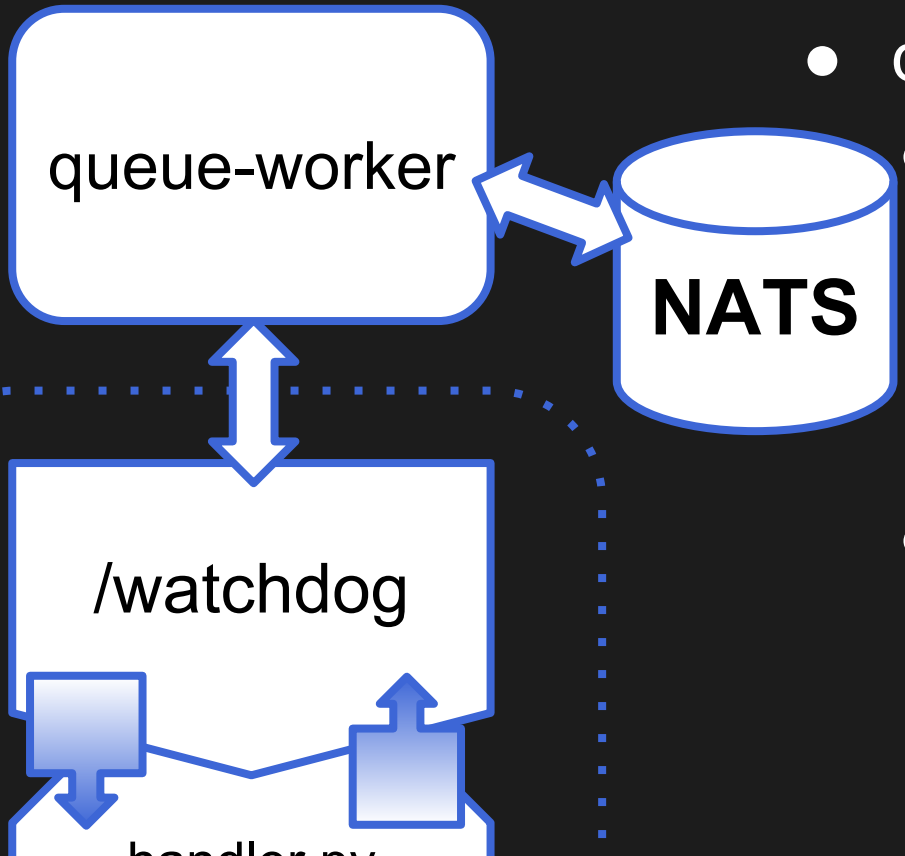
OpenFaaS Architecture

Function & Invoker



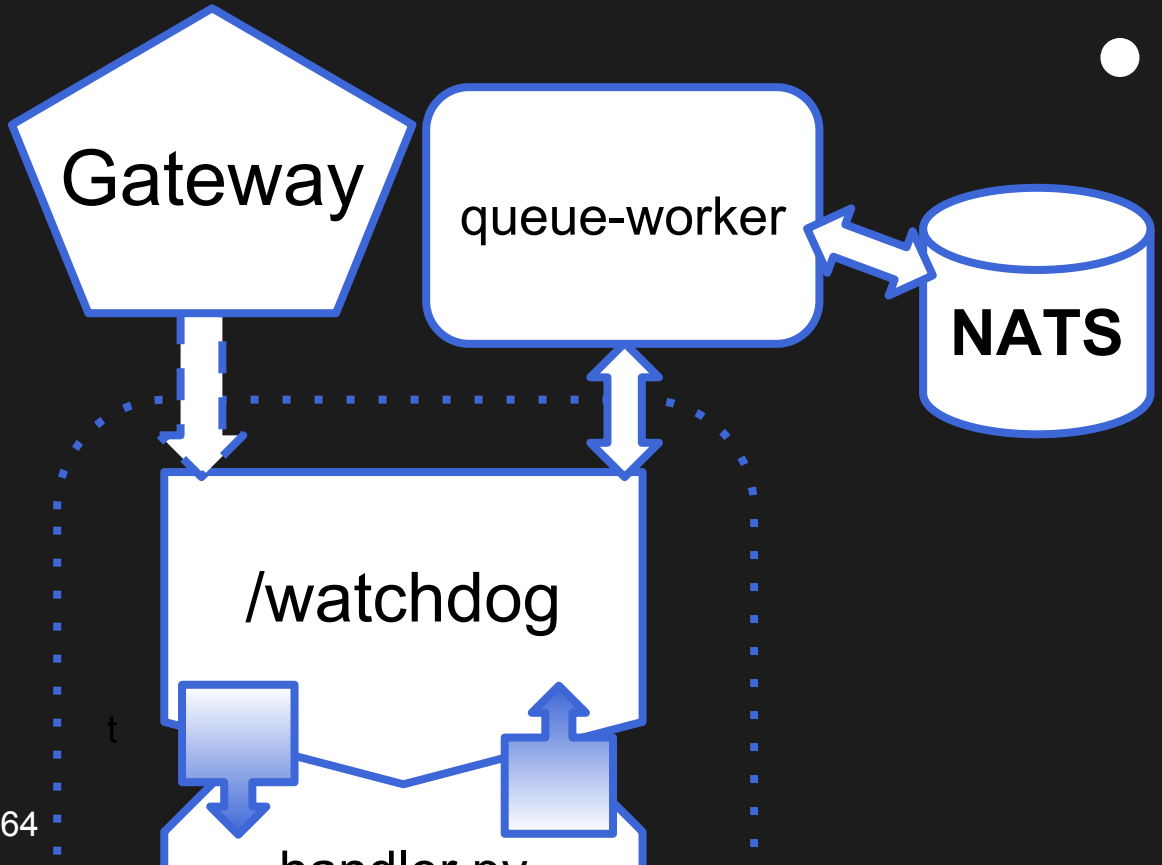
- Functions are run in containers
- Watchdog invoker
  - Proxies all requests
  - Forks handler.py
    - Sends HTTP data to STIN
    - Returns STDOUT
- Invocations are parallel by default

Queue



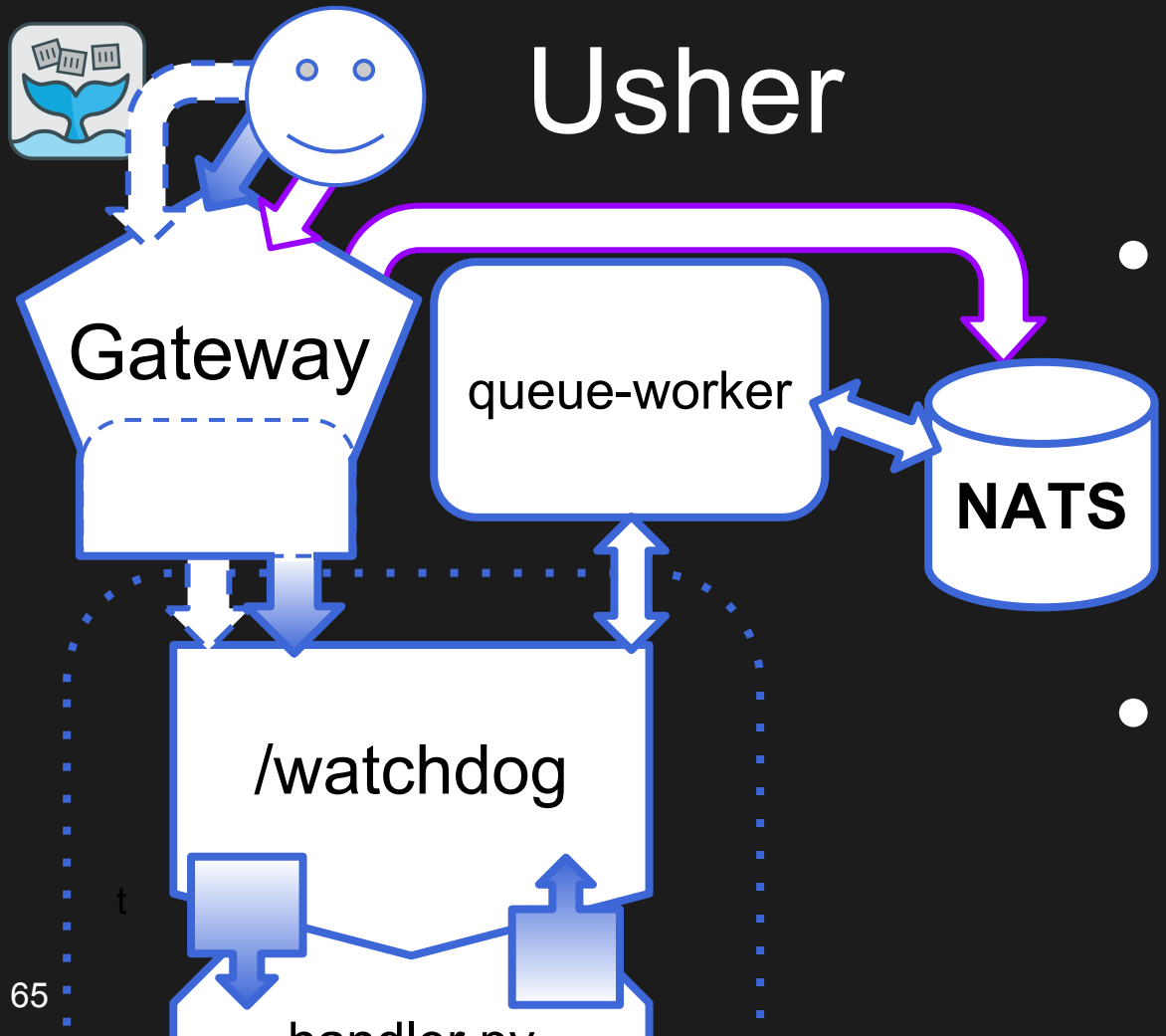
- Uses NATS to handle Async function calls
- queue-worker
  - Watches NATS topic and invokes function calls one at a time for each event
  - Returns results to NATS for later client retrieval

Ringleader

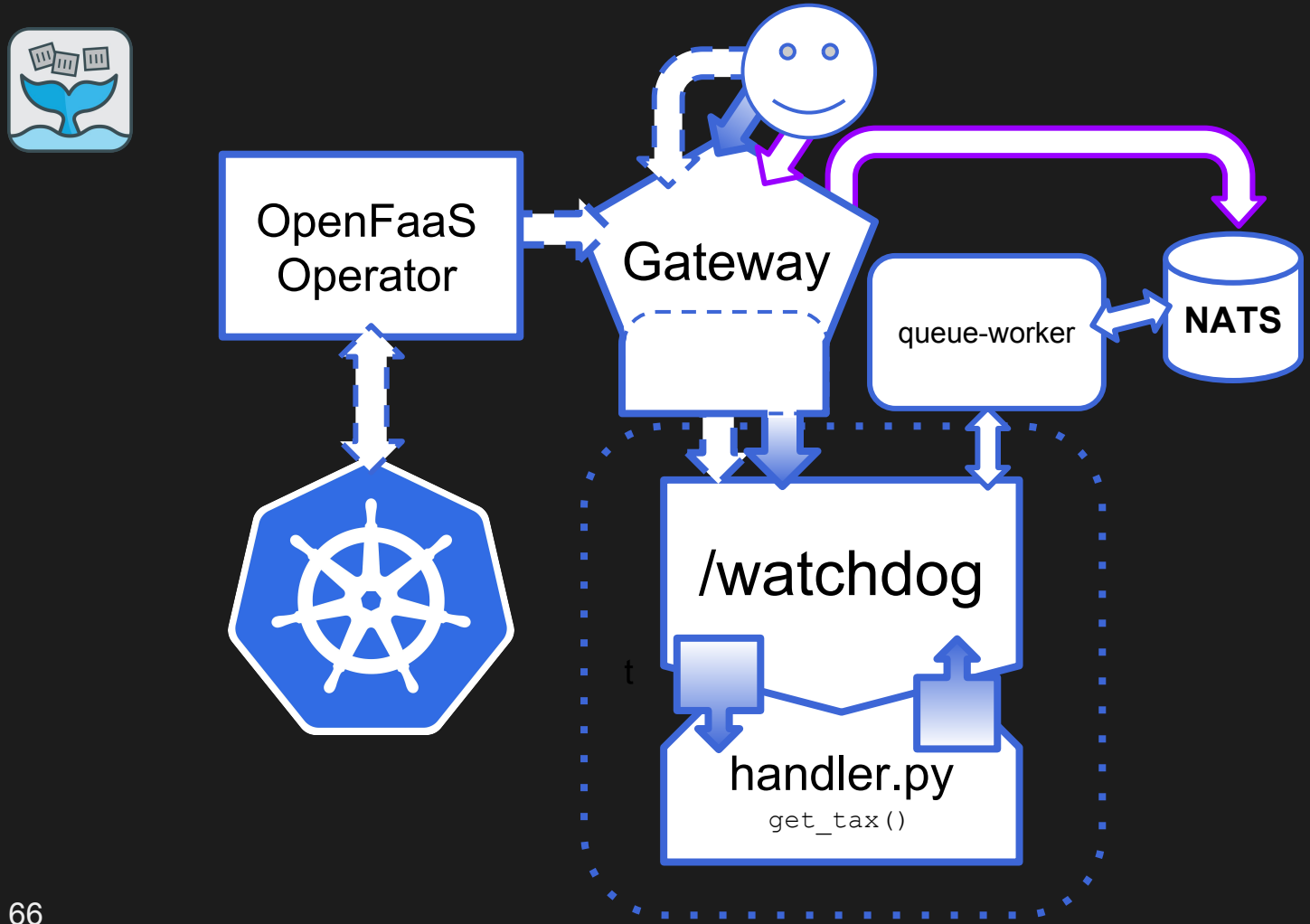


- Gateway hosts
  - Admin API
  - Admin GUI
  - prometheus metrics

Usher



- Gateway is also the Usher
  Proxies calls to
  - Functions (Sync)
  - NATS (Async)
- Scales from zero


OpenFaaS
Operator
Gateway
queue-worker
NATS
/watchdog
handler.py
get_tax()

More OpenFaaS Info

https://blog.alexellis.io/introducing-functions-as-a-service/

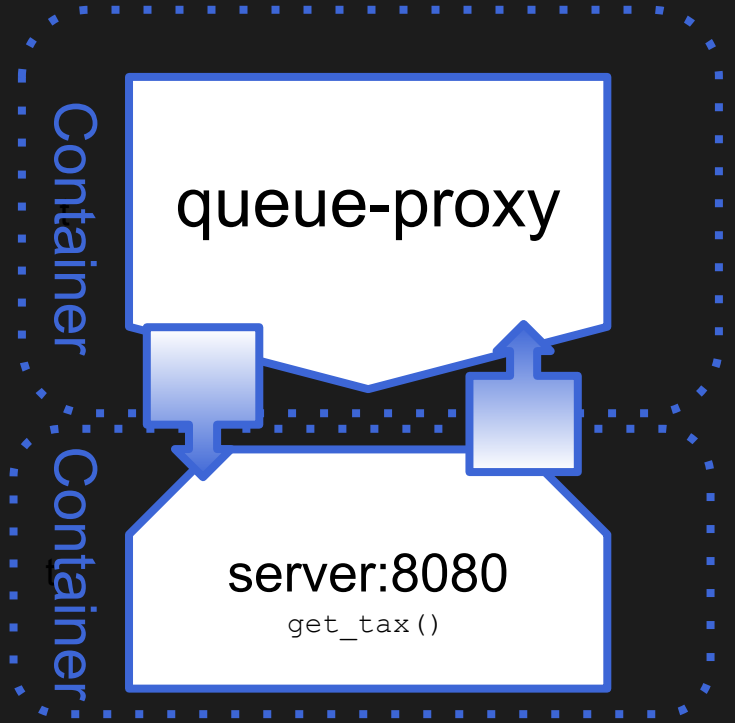
https://docs.openfaas.com/#presentations

https://docs.openfaas.com/architecture/gateway/

https://docs.openfaas.com/architecture/watchdog/

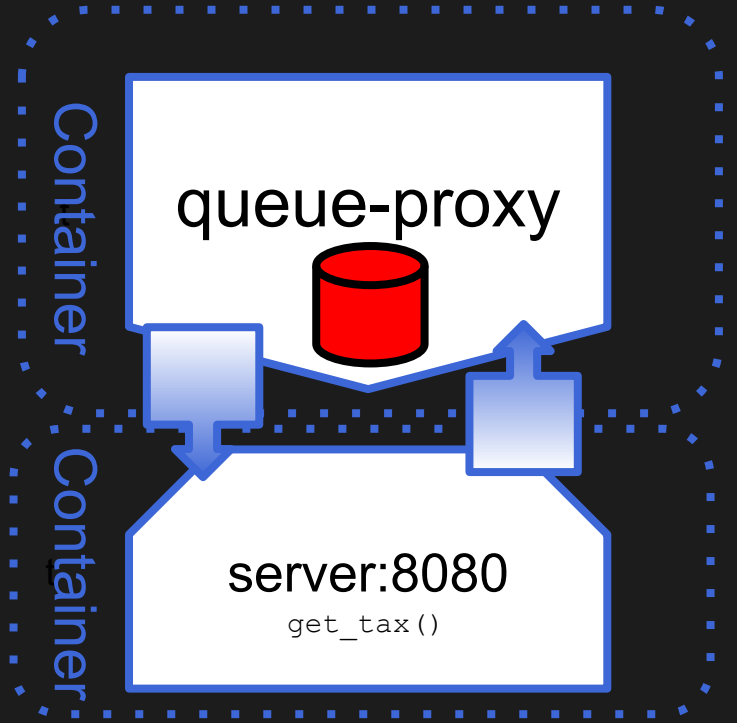
Knative Architecture

Function & Invoker



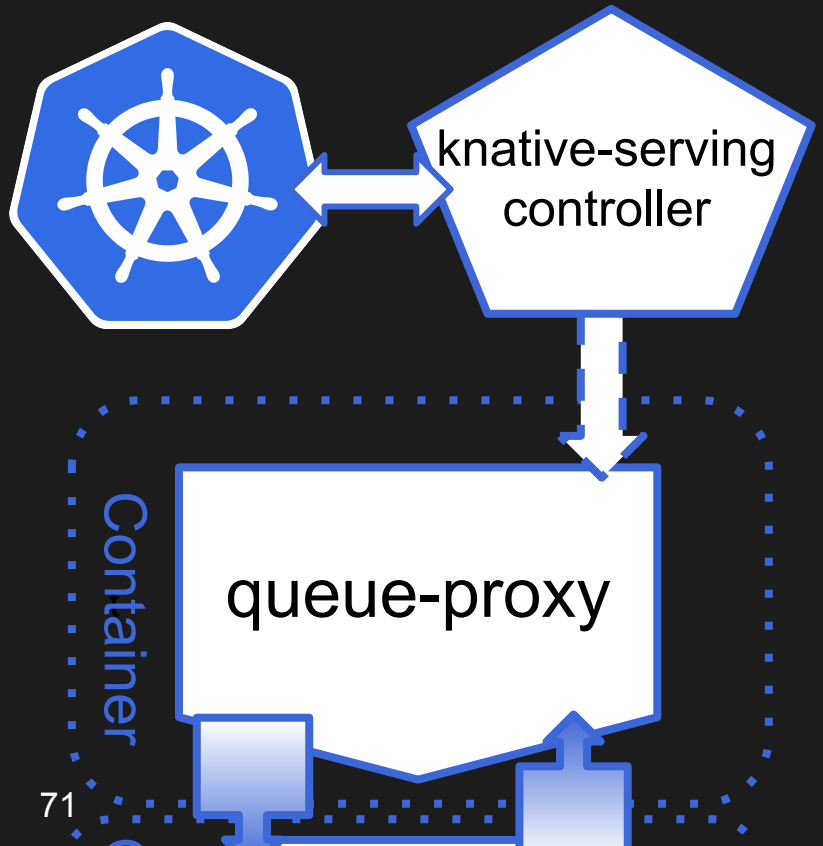
- Functions are web servers
  - Must process event data from web calls manually
- Queue-proxy intercepts traffic and proxies to server container
- Concurrency
  - Automatic
  - Serial
  - Parallel, 2-N

Queue



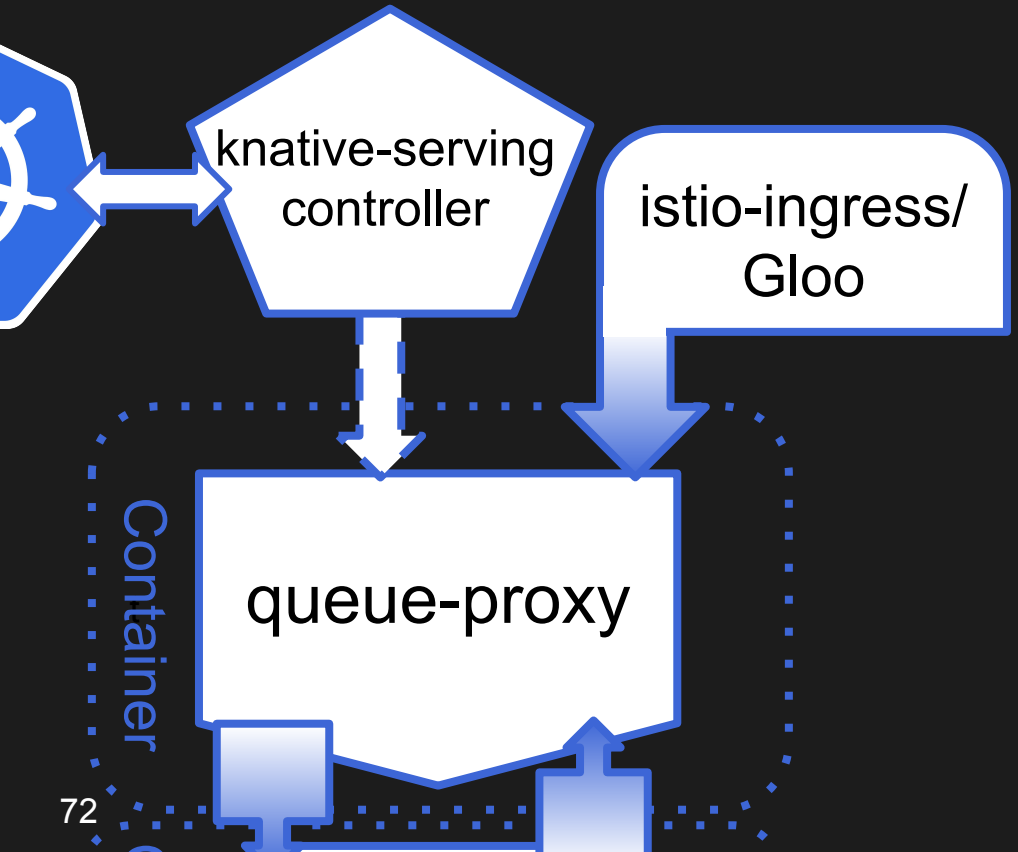
- queue-proxy has an internal queue for pending invocations
- No central queue system
- Lost containers lose all queued invocations

Ringleader



- Serving operator watches Kubernetes custom resources
  - Creates immutable revisions
  - Switches load to new revision

Usher



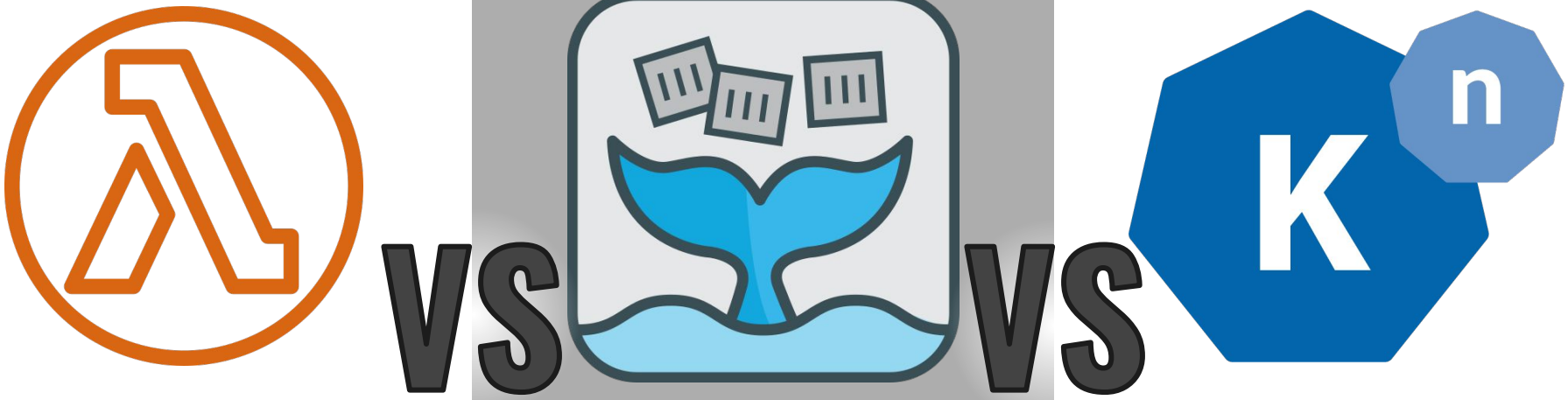
- Service Mesh power!
  - Canary deployments
  - Full roll-back

More Knative Info

https://youtu.be/LtELzpw1l1M

https://www.knative.dev/docs/

https://www.knative.dev/docs/serving/



Platform Installation

AWS Lambda Installation

- Get AWS Account
- Give AWS Money

OpenFaaS Installation

- Install with HELM
- Install with raw YAML

OpenFaaS Footprint

Namespaces
- openfaas
- openfaas-fn

Roles
- openfaas-operator-rw
- openfaas-prometheus

RoleBindings
- openfaas-operator-rw
- openfaas-prometheus

ConfigMaps
- alertmanager-config
- prometheus-config

CustomResourceDefinitions
- functions.openfaas.com

Ingresses
- openfaas-ingress

OpenFaaS Footprint Cont.

Services

- alertmanager
- gateway
- gateway-external
- nats
- prometheus

Deployments

- alertmanager
- faas-idler
- gateway
- nats
- prometheus
- queue-worker

Knative Installation Prereq

- Install Istio (Full support)
  - Install with raw YAML
    - Comprehensive Install
    - Limited Install
    - Custom Install

Knative (Istio) Footprint

Namespaces
- istio-system

ClusterRoles
- cluster-local-gateway-istio-system
- istio-citadel-istio-system
- istio-cleanup-secrets-istio-system
- istio-egressgateway-istio-system
- istio-galley-istio-system
- istio-ingressgateway-istio-system
- istio-mixer-istio-system
- istio-pilot-istio-system
- istio-sidecar-injector-istio-system

ClusterRoleBindings
- cluster-local-gateway-istio-system
- istio-citadel-istio-system
- istio-cleanup-secrets-istio-system
- istio-egressgateway-istio-system
- istio-galley-admin-role-binding-istio-system
- istio-ingressgateway-istio-system
- istio-mixer-admin-role-binding-istio-system
- istio-pilot-istio-system
- istio-sidecar-injector-admin-role-binding-istio-system

ConfigMaps
- istio
- istio-galley-configuration
- istio-security-custom-resources
- istio-sidecar-injector
- istio-statsd-prom-bridge

Jobs
- istio-cleanup-secrets

Kubernetes.config.istio.io
- attributes

AttributeManifests
- istioproxy
- kubernetes

Destinationrule.networking.istio.io
- istio-policy
- istio-telemetry

Gateway.networking.istio.io
- istio-autogenerated-k8s-ingress

Knative (Istio) Footprint Cont.

CustomResourceDefinitions

- adapters.config.istio.io
- adapters.config.istio.io
- apikeys.config.istio.io
- apikeys.config.istio.io
- attributemanifests.config.istio.io
- attributemanifests.config.istio.io
- authorizations.config.istio.io
- authorizations.config.istio.io
- bypasses.config.istio.io
- bypasses.config.istio.io
- checknothings.config.istio.io
- checknothings.config.istio.io
- circonuses.config.istio.io
- circonuses.config.istio.io
- deniers.config.istio.io
- deniers.config.istio.io
- destinationrules.networking.istio.io
- destinationrules.networking.istio.io
- edges.config.istio.io
- edges.config.istio.io
- envoyfilters.networking.istio.io
- envoyfilters.networking.istio.io
- fluentds.config.istio.io
- fluentds.config.istio.io
- gateways.networking.istio.io
- gateways.networking.istio.io
- handlers.config.istio.io
- handlers.config.istio.io
- httpapispecbindings.config.istio.io
- httpapispecbindings.config.istio.io
- httpapispecs.config.istio.io
- httpapispecs.config.istio.io
- instances.config.istio.io
- instances.config.istio.io
- kubernetesenvs.config.istio.io
- kubernetesenvs.config.istio.io
- kuberneteses.config.istio.io
- kuberneteses.config.istio.io
- listcheckers.config.istio.io
- listcheckers.config.istio.io
- listentries.config.istio.io
- listentries.config.istio.io
- logentries.config.istio.io
- logentries.config.istio.io
- memquotas.config.istio.io
- memquotas.config.istio.io
- meshpolicies.authentication.istio.io
- metrics.config.istio.io
- metrics.config.istio.io
- noops.config.istio.io
- noops.config.istio.io
- opas.config.istio.io
- opas.config.istio.io
- policies.authentication.istio.io
- prometheuses.config.istio.io
- prometheuses.config.istio.io
- quotas.config.istio.io
- quotas.config.istio.io
- quotaspecbindings.config.istio.io
- quotaspecbindings.config.istio.io
- quotaspecs.config.istio.io
- quotaspecs.config.istio.io
- rbacconfigs.rbac.istio.io
- rbacconfigs.rbac.istio.io
- rbacs.config.istio.io
- rbacs.config.istio.io
- redisquotas.config.istio.io
- redisquotas.config.istio.io
- reportnothings.config.istio.io
- reportnothings.config.istio.io
- rules.config.istio.io
- rules.config.istio.io
- servicecontrolreports.config.istio.io
- servicecontrolreports.config.istio.io
- servicecontrols.config.istio.io
- servicecontrols.config.istio.io
- serviceentries.networking.istio.io
- serviceentries.networking.istio.io
- servicerolebindings.rbac.istio.io
- servicerolebindings.rbac.istio.io
- serviceroles.rbac.istio.io
- serviceroles.rbac.istio.io
- signalfxs.config.istio.io
- signalfxs.config.istio.io
- solarwindses.config.istio.io
- solarwindses.config.istio.io
- stackdrivers.config.istio.io
- stackdrivers.config.istio.io
- statsds.config.istio.io
- statsds.config.istio.io
- stdios.config.istio.io
- stdios.config.istio.io
- templates.config.istio.io
- templates.config.istio.io
- tracespans.config.istio.io
- tracespans.config.istio.io
- virtualservices.networking.istio.io
- virtualservices.networking.istio.io

Knative (Istio) Footprint Cont.

deployment.extensions
- cluster-local-gateway
- istio-citadel
- istio-egressgateway
- istio-galley
- istio-ingressgateway
- istio-pilot
- istio-policy
- istio-sidecar-injector
- istio-statsd-prom-bridge
- istio-telemetry

horizontalpodautoscalers
- cluster-local-gateway
- istio-egressgateway
- istio-ingressgateway
- istio-pilot
- istio-policy
- istio-telemetry

kubernetesenv.config.istio.io
- handler

logentry.config.istio.io
- accesslog
- tcpaccesslog

metric.config.istio.io
- requestcount
- requestduration
- requestsize
- responsesize
- tcpbytereceived
- tcpbytesent

mutatingwebhookconfiguration
- istio-sidecar-injector

prometheus.config.istio.io
- handler

rule.config.istio.io
- kubeattrgenrulerule
- promhttp
- promtcp
- stdio
- stdiotcp
- tcpkubeattrgenrulerule

service
- cluster-local-gateway
- istio-citadel
- istio-egressgateway
- istio-galley
- istio-ingressgateway
- istio-pilot
- istio-policy
- istio-sidecar-injector
- istio-statsd-prom-bridge
- istio-telemetry

serviceaccount
- cluster-local-gateway-service-account
- istio-citadel-service-account
- istio-cleanup-secrets-service-account
- istio-egressgateway-service-account
- istio-galley-service-account
- istio-ingressgateway-service-account
- istio-mixer-service-account
- istio-pilot-service-account
- istio-sidecar-injector-service-account

stdio.config.istio.io
- handler

Knative Installation

- Install Knative
  - Install with raw YAML

Knative Footprint

Namespaces
- default
- istio-system
- knative-serving

Services
- activator-service
- autoscaler
- controller
- webhook

Deployments
- activator
- autoscaler
- controller
- webhook

ClusterRoles
- knative-serving-admin
- knative-serving-core

ClusterRoleBindings
- knative-serving-controller-admin

ConfigMaps
- config-autoscaler
- config-controller
- config-domain
- config-gc
- config-istio
- config-logging
- config-network
- config-observability

CustomResourceDefinitions
- clusteringresses.networking.internal.knative.dev
- configurations.serving.knative.dev
- images.caching.internal.knative.dev
- podautoscalers.autoscaling.internal.knative.dev
- revisions.serving.knative.dev
- routes.serving.knative.dev
- services.serving.knative.dev

Deployments
- activator
- autoscaler
- controller
- webhook

ConfigMaps
- config-autoscaler
- config-controller
- config-domain
- config-gc
- config-istio
- config-logging
- config-network
- config-observability

Knative Footprint Cont.

gateway.networking.istio.io
- **cluster-local-gateway**
- **knative-ingress-gateway**

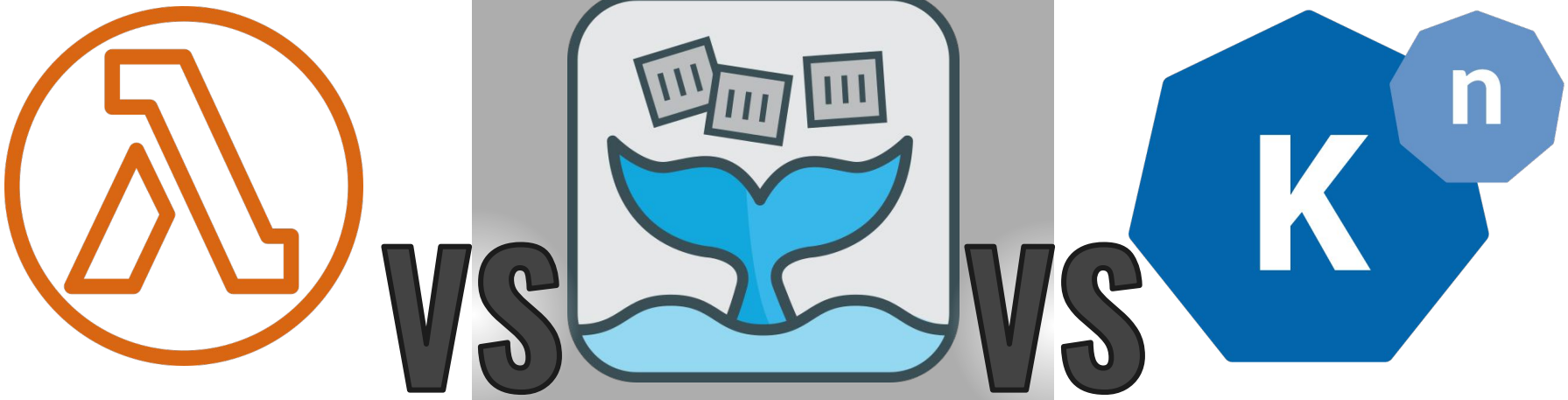
image.caching.internal.knative.dev
- **queue-proxy**

Services
- **activator-service**
- **autoscaler**
- **controller**
- **webhook**

ServiceAccount
- **controller**

Alternative: Knative/Gloo Installation

- Install Gloo (Partial support)
  - `"glooctl install knative"`



Function: Building

Build Process

- Create the Function
  - mkdir myfunc
  - cd myfunc
  - vi function.py
- Install Dependencies
- Submit to AWS
  - zip -r function.zip .
  - aws lambda update-function-code

Function Runtimes

- AWS Provided:
  - Node 6.10, 8.10
  - Python 2.6, 3.6, 3.7
  - Ruby 2.5
  - Java 8
  - Go 1.X
  - .Net Core 1.0, 2.0, 2.1
- Runtimes are Customizable

Language Dependencies

- KVM (Amazon Linux)
  - Native libraries (ex: cython) may require a dedicated build machine
- Must package all dependencies in a zip file
  - Python
    - pip install --target .
    - virtualenv
  - Node
    - node_modules
  - Ruby
    - bundle install --path vendor/bundle

Build Process

- Generate a Docker image
  - faas-cli new --lang python myfunc
  - Edit - requirements.txt
  - Edit - handler.py
  - faas-cli build -f ./myfunc.yml

Faas-cli Tool

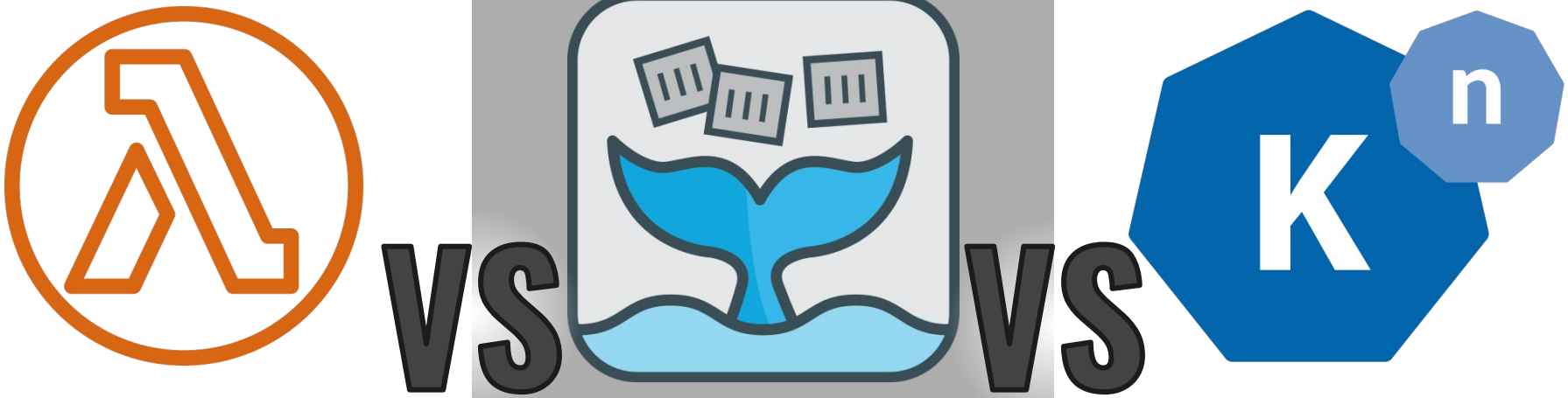
- Function
  - Generate
  - Build
  - Deploy
  - Invoke
- Secrets management
- Config management
- Template store

faas-cli template store ls

```
NAME                    SOURCE             DESCRIPTION
csharp                  openfaas           Official C# template
dockerfile              openfaas           Official Dockerfile template
go                      openfaas           Official Golang template
java8                   openfaas           Official Java 8 template
node                    openfaas           Official NodeJS 8 template
php7                    openfaas           Official PHP 7 template
python                  openfaas           Official Python 2.7 template
python3                 openfaas           Official Python 3.6 template
ruby                    openfaas           Official Ruby 2.5 template
node10-express          openfaas-incubator NodeJS 10 Express template
ruby-http               openfaas-incubator Ruby 2.4 HTTP template
python27-flask          openfaas-incubator Python 2.7 Flask template
python3-flask           openfaas-incubator Python 3.6 Flask template
node8-express           openfaas-incubator NodeJS 8 Express template
golang-http             openfaas-incubator Golang HTTP template
golang-middleware       openfaas-incubator Golang Middleware template
python3-debian          openfaas-incubator Python 3.6 Debian template
powershell-template     openfaas-incubator Powershell Core Ubuntu:16.04 template
powershell-http-template openfaas-incubator Powershell Core HTTP Ubuntu:16.04 template
```

Containerized Builds

- Knative build system
  - Triggered by source changes
    - Build
    - Push
    - Rollout
- Build your own Docker image from scratch
  - Must be a full web server



Function: Deployment

AWS Lambda

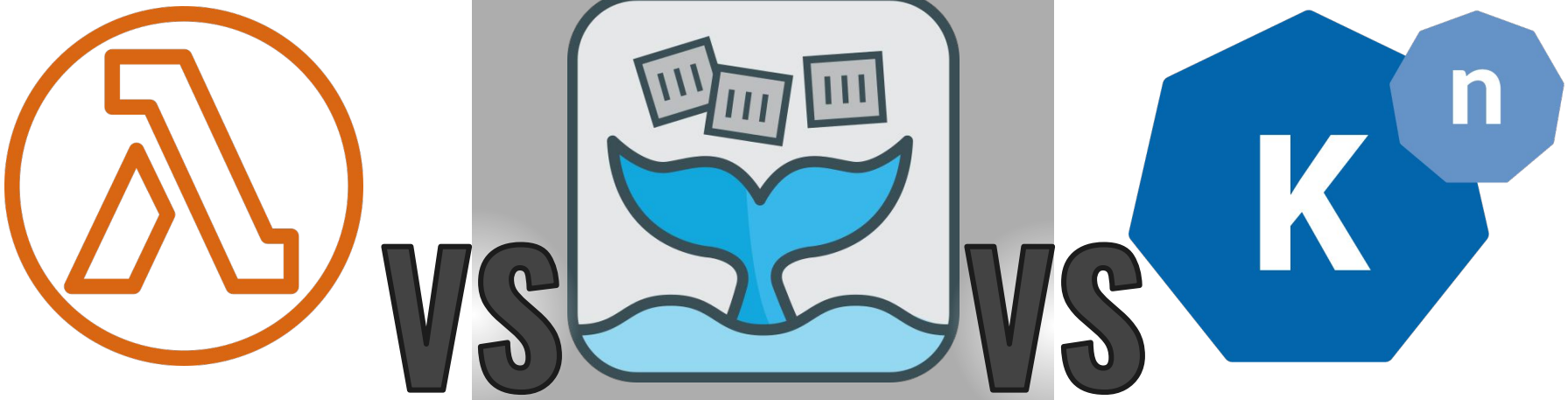
- AWS Console
- AWS API
- AWS CLI
  - `aws lambda update-code`

OpenFaaS

- Gateway GUI
- Gateway API
- FaaS CLI
  - `faas-cli deploy -f myfunc.yaml`
- Kubernetes CR (Operator Required)
  - `kubectl create -f myfunc-fn.yaml`

Knative

- Kubernetes CR
  - `kubectl create -f myfunc-fn.yaml`



Function: Invoking

AWS Lambda

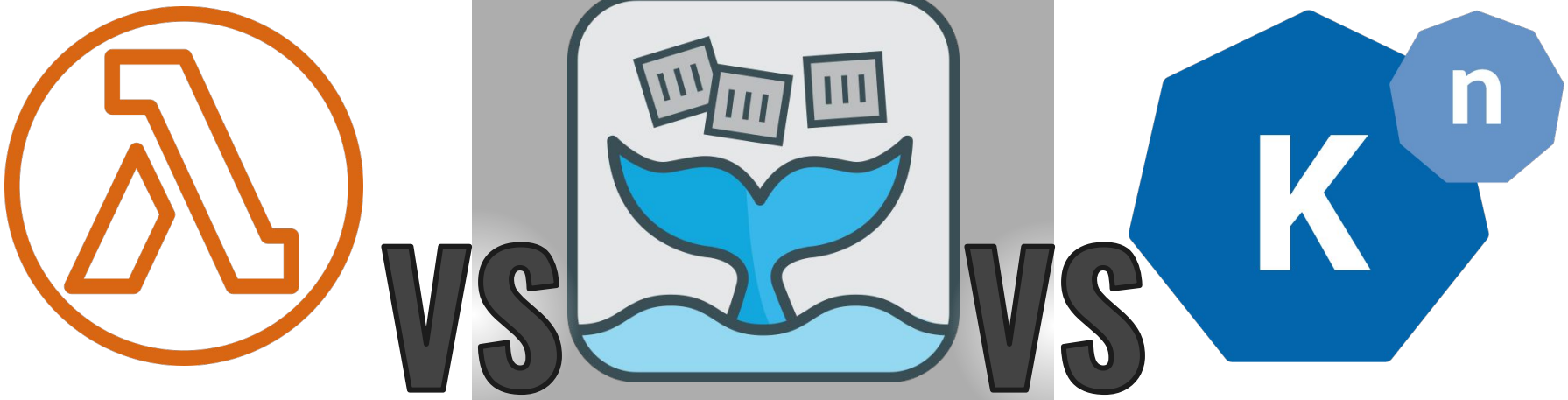
- AWS Console
- AWS API
- AWS CLI
  - `aws lambda invoke`

OpenFaaS

- Gateway GUI
- FaaS CLI
  - `faas-cli invoke ...`
- curl <ingress>/functions/get-tax

Knative

```
curl \
  -H 'Host: get-tax.example.com' \
  <ingress>
```



Summary

AWS Lambda

- No overhead
- No management
- VM runtimes
- Limits
- Lock-in

OpenFaaS

- Lightweight
- Cloud Native Components
- Faas-cli
  - Function templates
  - Full platform interaction
- Parallel invocations by default
- No canary or automated roll-back

Knative

- Full serverless platform
- Uses istio
- Istio required
- No persistent queue
- Few faas platform helpers
- No GUI

More Serverless Platforms

- OpenWhisk
  - Kafka required
  - Container per invocation
- Kubeless
  - Code Injection - no image builds
  - NATS/Kafka Optional
- Fission
  - Code injection - no image builds
  - Pooled, "warm" containers

More Serverless Platforms

- IronFunctions
  - Can import lambda functions
- Fn
  - Persistent DB required
  - Container per invocation and/or re-used "hot" containers

More of Me

github.com/carsonoid/talk-knative-vs-openfaas

@carsonoid

@carson_ops

kube-decon.carson-anderson.com

dynamic-kubernetes.carson-anderson.com

salt-decon.carson-anderson.com