Leave the Clouds
You don’t need Bezos when you have Python

Alexander Nicholson • TableCheck • Tokyo Python Society

Who am I?

Alexander Nicholson

Working as a **Service Reliability Engineer** at **TableCheck** in Tokyo.

See my work at:

https://alexander.town/

What is this talk about?

- Deploying Python functions onto infrastructure in a cost effective manner.

Get ready for some *buzzwords*

Applications of this talk

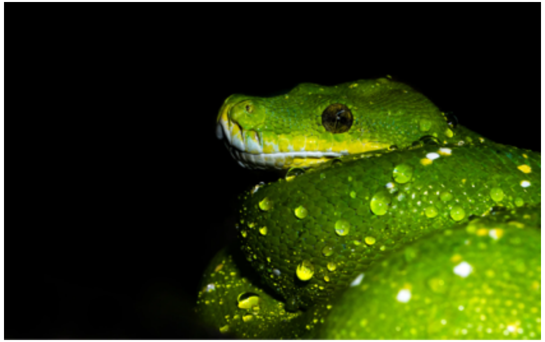
- “Private Cloud on Public Cloud”
- Machine learning model training (if you attended the previous talk!)
- Build clusters
- Run personal projects
- Web scraping
- Showing off your self-hosted website to your friends
- **Learning Python**
- Hosting turnkey apps

While I was writing this talk

- Coronavirus Disease 2019 (COVID-19) blew up.
- Many of us are quarantined around the world.
- We have a lot of time on our hands now!

- Let’s learn!

Overview



Goals

- Set up our Infrastructure as a Service
- Orchestrate it using Python
- Deploy a function
- Scale it

Warnings

- This talk contains a **LOT** of infrastructure stuff.
- I go over some parts very quick.
- This costs money if you do it, I’m not responsible for anything you do, you’re an adult, etc.

WTH is Cloud Computing?

Wikipedia’s definition

- “The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run [illegible] include operating syste[illegible]umer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).”

This cloud, specifically.

Blah blah blah

- We want servers to run containers to run our apps/ services in!

But I want it to be cheap!

Running at home

- You have to buy and own heavy and large hardware
- You have to deal with the noise of a server whirring
- **You need a public IP**
- **Cost you from $50-90/mth in power**

Running in the Cloud

- You can use spot instances to optimise cost over availability
- You can script the service to be available when needed
- **Cost from $0.01 (plus a manager server for $5)**

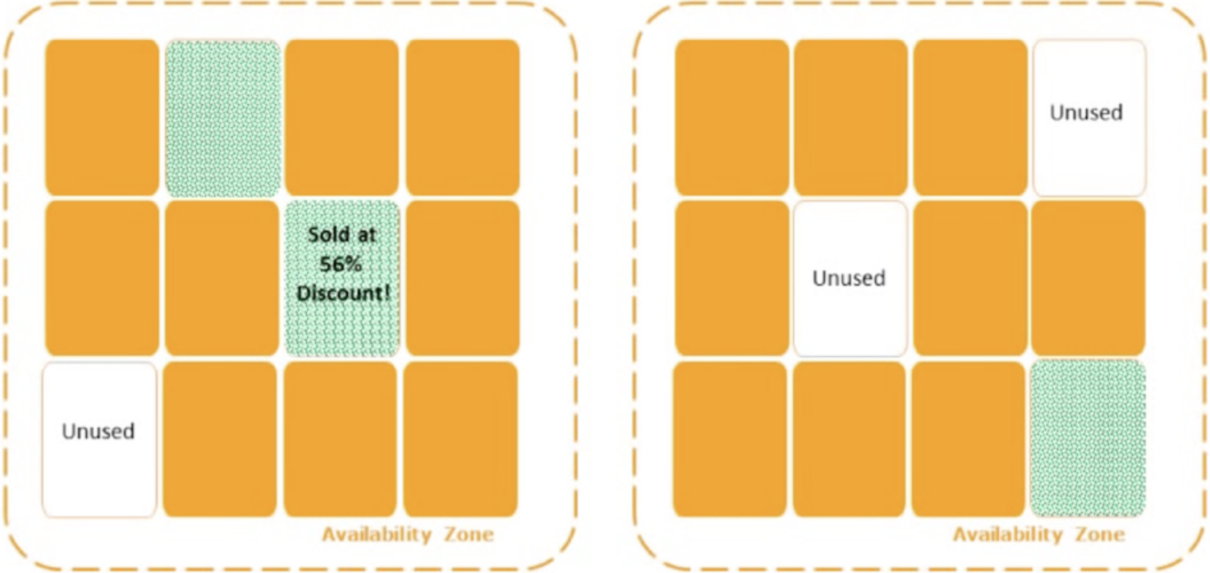
Our Goals

1. As much resources as possible
2. As private as possible (metal preferred)
3. Lifecycle as scriptable as possible

What is spot?

- Primarily, it is **dynamic resource management**.
- Amazon Web Services and other providers have a lot of spare capacity which, sitting around, makes them no money.
- They allow you to place a “bid” on these resources in order to ensure that they are being used (and thus the provider is getting paid for them).
- **We can use this to get cheaper services!**

Spot resourcing in a single picture



Our options for some spot resources

AWS

https://aws.amazon.com/ec2/instance-types/a1/

An ARM-based bare metal server which averages 15.5 JPY/hour in Tokyo.

Packet.net

https://www.packet.com/cloud/servers/t1-small/

A traditional bare-metal server which averages 3.31 → 7.72 JPY/hour in Tokyo.

Something else...

Technically, we can also use virtual services… as long as they are KVM or similar.

No Kubernetes!

It’s outside the scope of this talk.
And you don’t need it to scale anyway.

Things we want

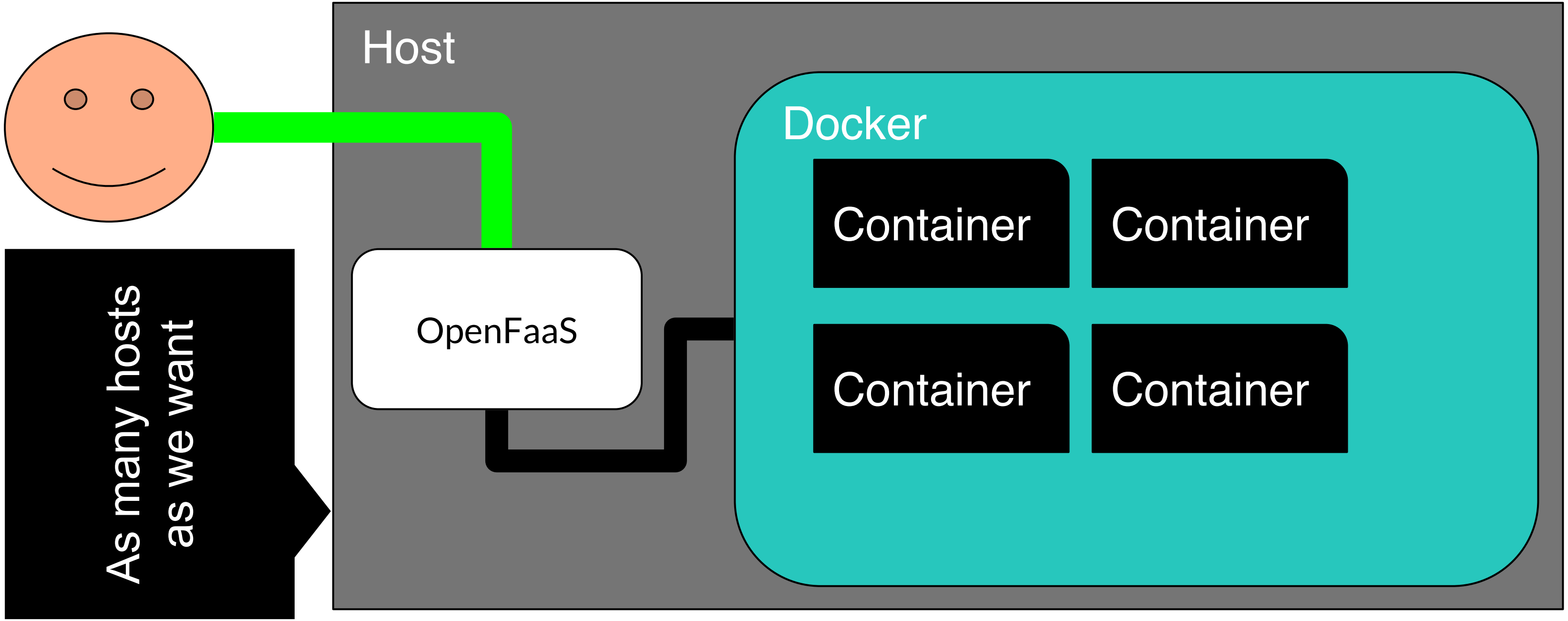
1. Clustering ← So we can have many servers
2. Isolated ← For security
3. High availability ← Mitigating downtime
4. Easy scripting ← Easy == Good
5. Low overhead ← To not waste resources
6. Scalable ← For becoming Enterprise-level

↑ Which kinda fits into point 1

The basics - 1

- Control the lifecycle of our underlying providers

How does this all fit together?



Host

Docker

Container

Container

Container

Container

OpenFaaS

As many hosts as we want

What is OpenFaaS?

- OpenFaaS (Functions as a Service) is a framework for building serverless functions which has first class support for metrics.
- Any process can be packaged as a function enabling you to consume a range of web events without repetitive boiler-plate coding.
- It has a growing community with many pre-built functions available!

How can it help me?

OPENFAAS

- If you are currently building APIs, applications, etc:
  - It gives you faas-cli which allows you to create/customise/deploy your Python functions straight from your shell.
  - It has first-class automatic scaling built in. Your Python functions will scale according to your traffic automatically!

Demo

Building our cloud
chotto.cloud

github.com
/CTRLTokyo
/scale-everything (soon)

OPENFAAS

Amazon EC2

I don’t have any money.

- Some people say you should achieve Google scale when the number of customers you have is equal to the dollars in your bank account ~~after Jeff Bezos from Amazon Incorporated has decided to take all of your money and ruin your life and your destroy your startup and this sentence is really long please ignore it and pay attention to the speaker okay cool~~
- *Please be careful when using any cloud services.*

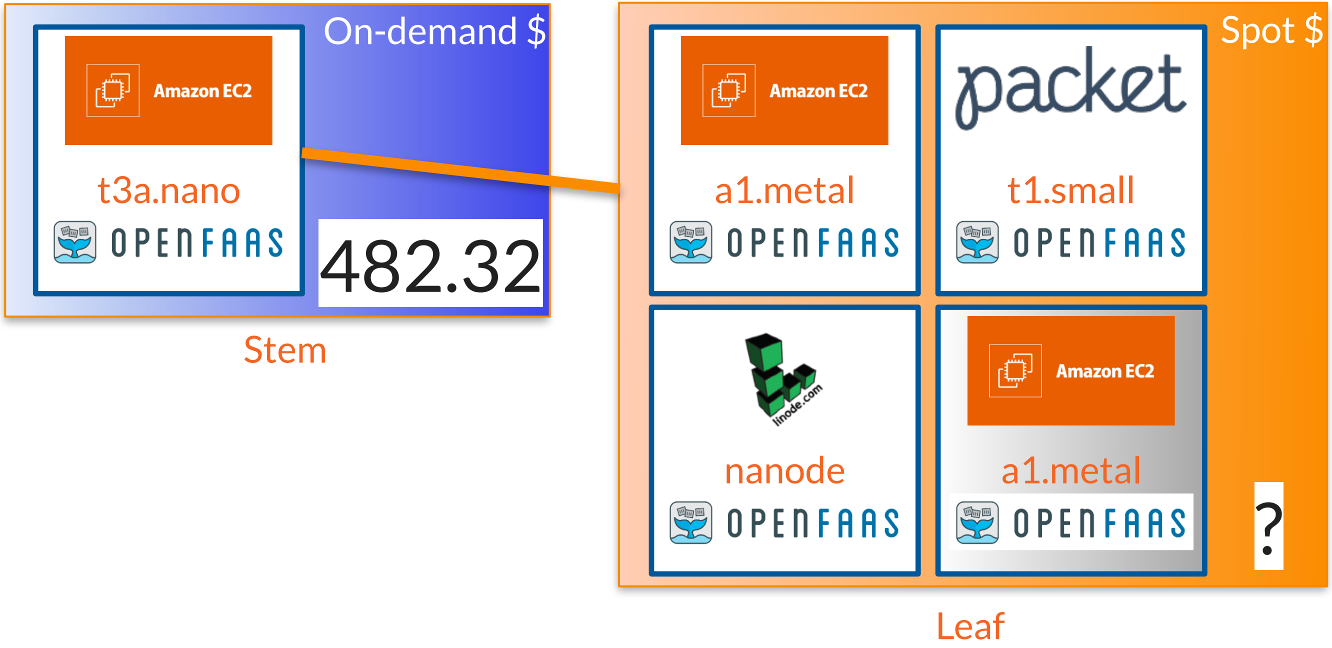
Scale From Zero

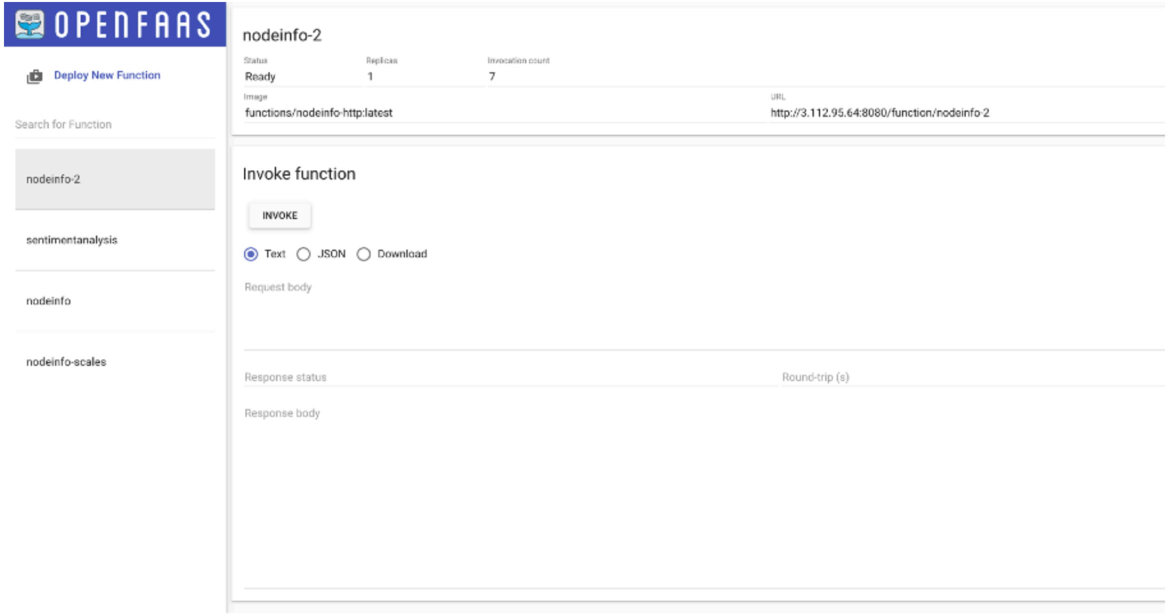
- Okay, seriously- you can run a tiny instance and scale your function as you need.
- Some functions only run once or a few times a day.
- *Scaling from zero allows you to cold-serve your app by spinning it up in the background after a request and keeping the connection active.*

Global Availability

- We need our application platform to be global.
- Not every provider has every country.
- *So, let’s use a bunch of different providers.*

Infrastructure





nodeinfo-2

Status: Ready

Replicas: 1

Invocation count: 7

Image: functions/nodeinfo-http:latest

URL: http://3.112.95.64:8080/function/nodeinfo-2

Invoke function

● Text ○ JSON ○ Download

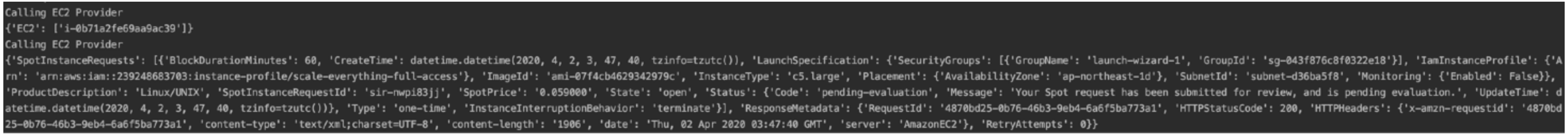
Request body

Response status

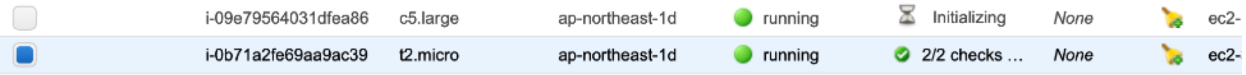
Round-trip (s)

Response body

Show the OpenFaaS interface - deploy a function etc



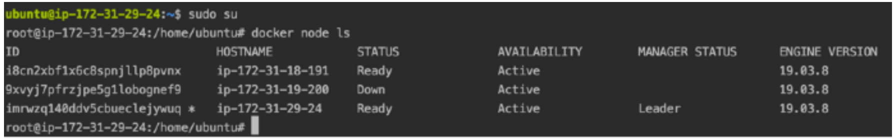
```
Calling EC2 Provider
{'EC2': ['i-0b71a2fe69aa9ac39']}
Calling EC2 Provider
{'SpotInstanceRequests': [{'BlockDurationMinutes': 60, 'CreateTime': datetime.datetime(2020, 4, 2, 3, 47, 40, tzinfo=tzutc()), 'LaunchSpecification': {'SecurityGroups': [{'GroupName': 'launch-wizard-1', 'GroupId': 'sg-043f876c8f0322e18'}], 'IamInstanceProfile': {'A
rn': 'arn:aws:iam::239248683703:instance-profile/scale-everything-full-access'}, 'ImageId': 'ami-07f4cb4629342979c', 'InstanceType': 'c5.large', 'Placement': {'AvailabilityZone': 'ap-northeast-1d'}, 'SubnetId': 'subnet-d36ba5f8', 'Monitoring': {'Enabled': False}},
'ProductDescription': 'Linux/UNIX', 'SpotInstanceRequestId': 'sir-nwpi83jj', 'SpotPrice': '0.059000', 'State': 'open', 'Status': {'Code': 'pending-evaluation', 'Message': 'Your Spot request has been submitted for review, and is pending evaluation.', 'UpdateTime': d
atetime.datetime(2020, 4, 2, 3, 47, 40, tzinfo=tzutc())}, 'Type': 'one-time', 'InstanceInterruptionBehavior': 'terminate'}], 'ResponseMetadata': {'RequestId': '4870bd25-0b76-46b3-9eb4-6a6f5ba773a1', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': '4870bd
25-0b76-46b3-9eb4-6a6f5ba773a1', 'content-type': 'text/xml;charset=UTF-8', 'content-length': '1906', 'date': 'Thu, 02 Apr 2020 03:47:40 GMT', 'server': 'AmazonEC2'}, 'RetryAttempts': 0}}
```



□	i-09e79564031dfea86	c5.large	ap-northeast-1d	● running	Initializing	*None*	ec2-
☑	i-0b71a2fe69aa9ac39	t2.micro	ap-northeast-1d	● running	✓ 2/2 checks ...	*None*	ec2-

Worker

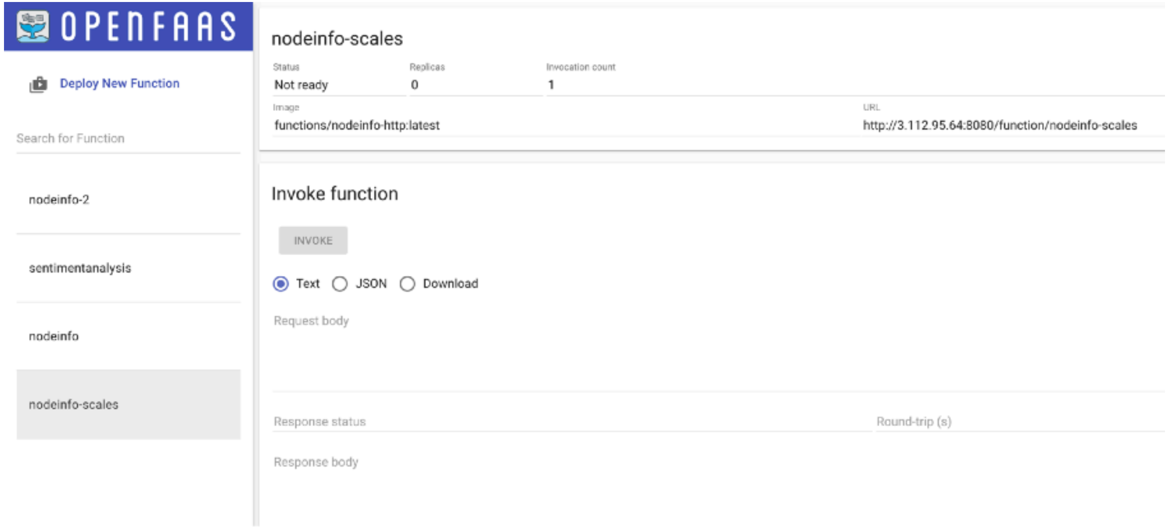
Manager



```
ubuntu@ip-172-31-29-24:~$ sudo su
root@ip-172-31-29-24:/home/ubuntu# docker node ls
ID                            HOSTNAME            STATUS              AVAILABILITY        MANAGER STATUS      ENGINE VERSION
i8cn2xbf1x6c8spnjllp8pvnx     ip-172-31-18-191    Ready               Active                                  19.03.8
9xvyj7pfrzjpe5g1lobognef9     ip-172-31-19-200    Down                Active                                  19.03.8
imrwzq140ddv5cbueclejywuq *   ip-172-31-29-24     Ready               Active              Leader              19.03.8
root@ip-172-31-29-24:/home/ubuntu#
```

We are deploying a spot instance with a 1 hour block.

Demonstrate adding a new worker via CLI



nodeinfo-scales

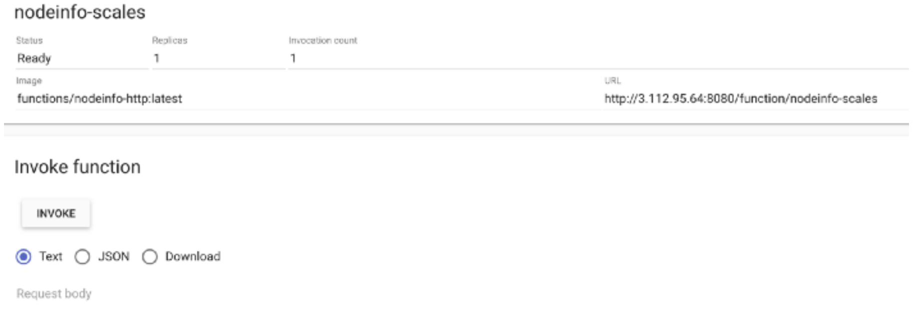
Status	Replicas	Invocation count
Not ready	0	1

Image	URL
functions/nodeinfo-http:latest	http://3.112.95.64:8080/function/nodeinfo-scales

Invoke function

● Text ○ JSON ○ Download

Show the OpenFaaS Scaling to Zero feature



nodeinfo-scales

Status: Ready

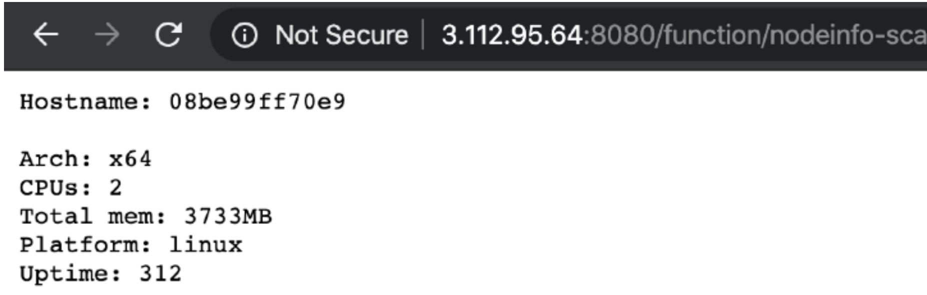
Replicas: 1

Invocation count: 1

Image: functions/nodeinfo-http:latest

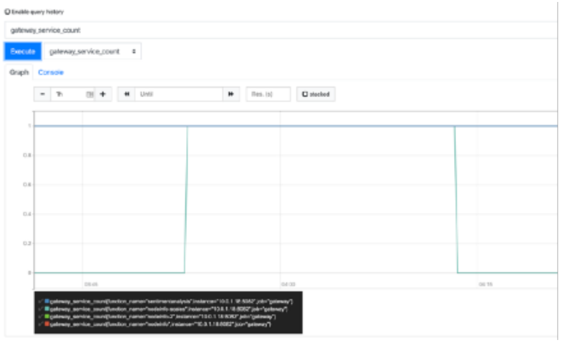
URL: http://3.112.95.64:8080/function/nodeinfo-scales

Invoke function



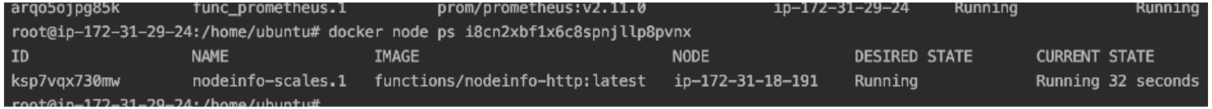
```
Hostname: 08be99ff70e9

Arch: x64
CPUs: 2
Total mem: 3733MB
Platform: linux
Uptime: 312
```



gateway_service_count

gateway_service_count{function_name="sentimentanalysis",instance="10.0.1.18:8082",job="gateway"}
gateway_service_count{function_name="nodeinfo-scales",instance="10.0.1.18:8082",job="gateway"}
gateway_service_count{function_name="nodeinfo-2",instance="10.0.1.18:8082",job="gateway"}
gateway_service_count{function_name="nodeinfo",instance="10.0.1.18:8082",job="gateway"}



```
arqo5ojpg85k        func_prometheus.1        prom/prometheus:v2.11.0            ip-172-31-29-24     Running             Running
root@ip-172-31-29-24:/home/ubuntu# docker node ps i8cn2xbf1x6c8spnjllp8pvnx
ID                  NAME                IMAGE                               NODE                DESIRED STATE       CURRENT STATE
ksp7vqx730mw        nodeinfo-scales.1   functions/nodeinfo-http:latest      ip-172-31-18-191    Running             Running 32 seconds
root@ip-172-31-29-24:/home/ubuntu#
```

Demonstrate scaling a function via HTTP calling

1

Let’s deploy something!

- faas-cli new --lang python hello-python
- nano hello-python/handler.py

  return "Wow, I never expected " + req

- We need to push it to a Docker Registry.
  - GitHub Packages can help!
  - https://github.com/orgs/ctrltokyo/packages?package_type=Docker

2

Let’s deploy something!

- faas-cli build -f ./hello-python.yml
- faas-cli push -f ./hello-python.yml
- faas-cli deploy -f ./hello-python.yml
- curl http://app.chotto.cloud:8080/function/hello-python -d "this magical event”

3

Let’s all run this

- curl http://app.chotto.cloud:8080/function/hello-python -d "a message”

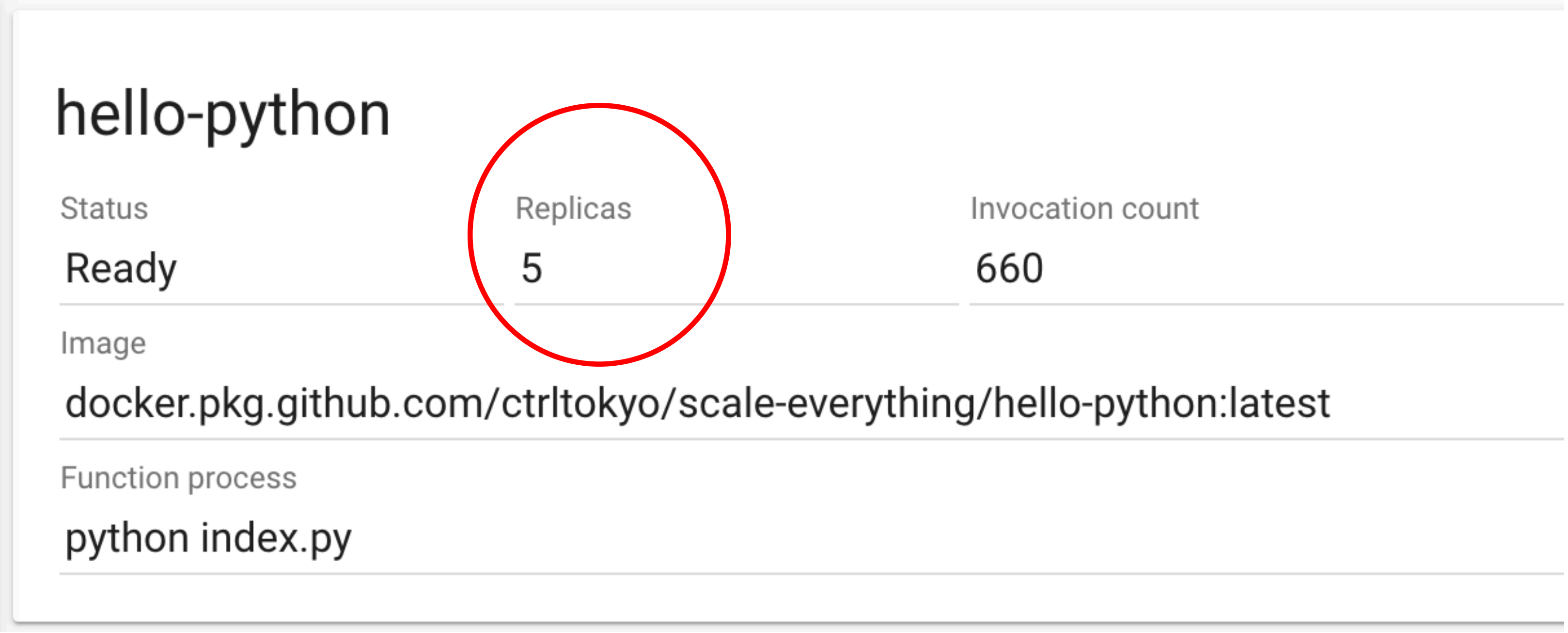
- (I’ll do it myself by looping infinitely…)

  ```
  while true
     curl http://app.chotto.cloud:8080/function/hello-python -d "this magical event"
  end
  ```

- Now let’s take a look at the replicas…

4

Whoah!



hello-python

Status
Ready

Replicas
5

Invocation count
660

Image
docker.pkg.github.com/ctrltokyo/scale-everything/hello-python:latest

Function process
python index.py

5

What did we learn?

- We can build our own cloud services using spot instances very cheaply!
- Orchestrating the cloud provider’s API is very easy.
- We can use this at our companies, in our projects etc in order to easily deploy our Python functions.

I want my own Cloud

Alexander Nicholson

Please go to
https://chotto.cloud/
and fill out the form!

See my other work at:

https://alexander.town/

That’s all!

Leave the Clouds

You don’t need Bezos when you have Python

Thank you for your time.

Any questions?

Alexander Nicholson • TableCheck • Tokyo Python Society