



mysterium
network

Golang application packed into minimal Docker image

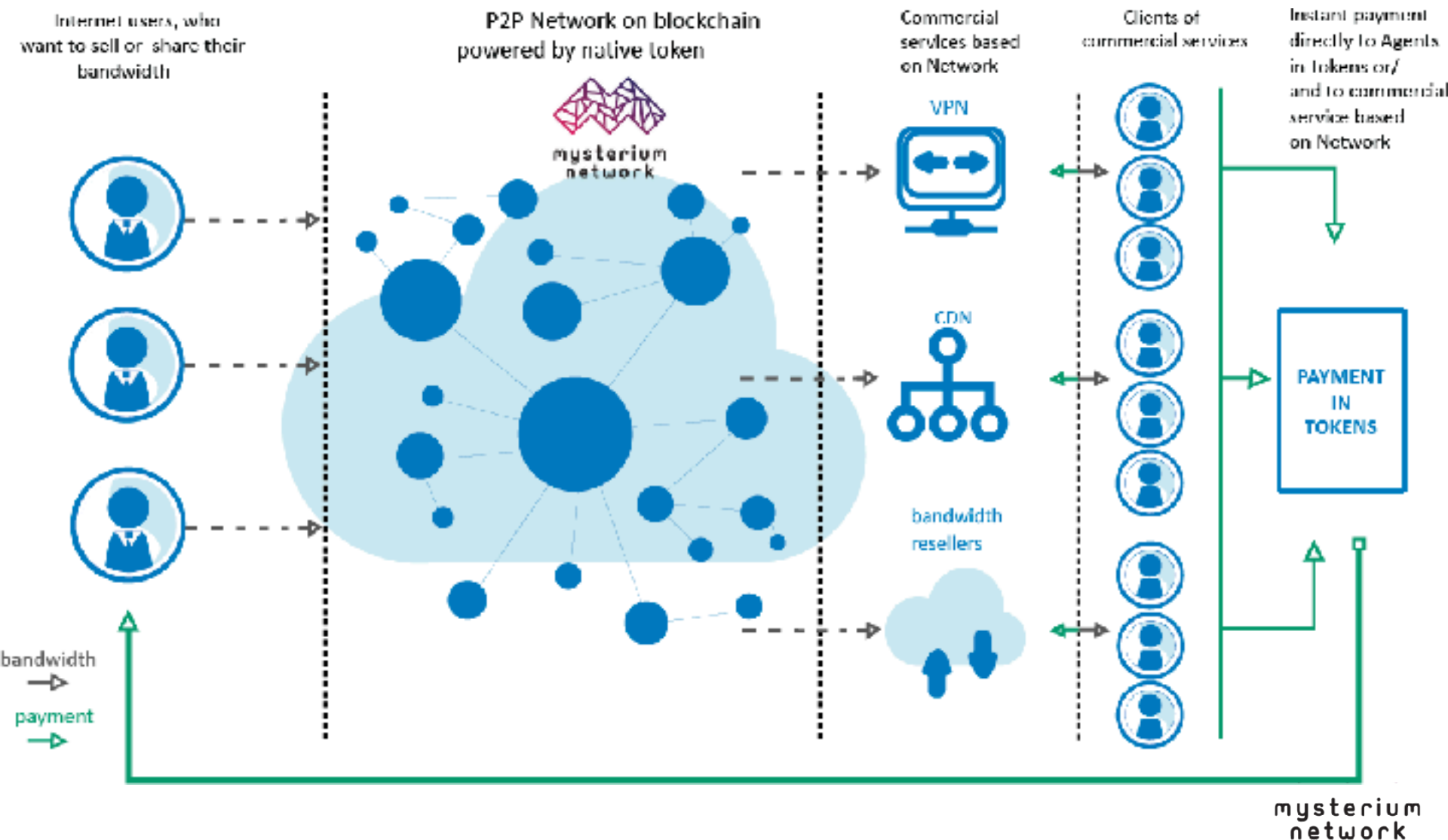
Valdas Petrulis
2018-03-06

Valdas Petrulis

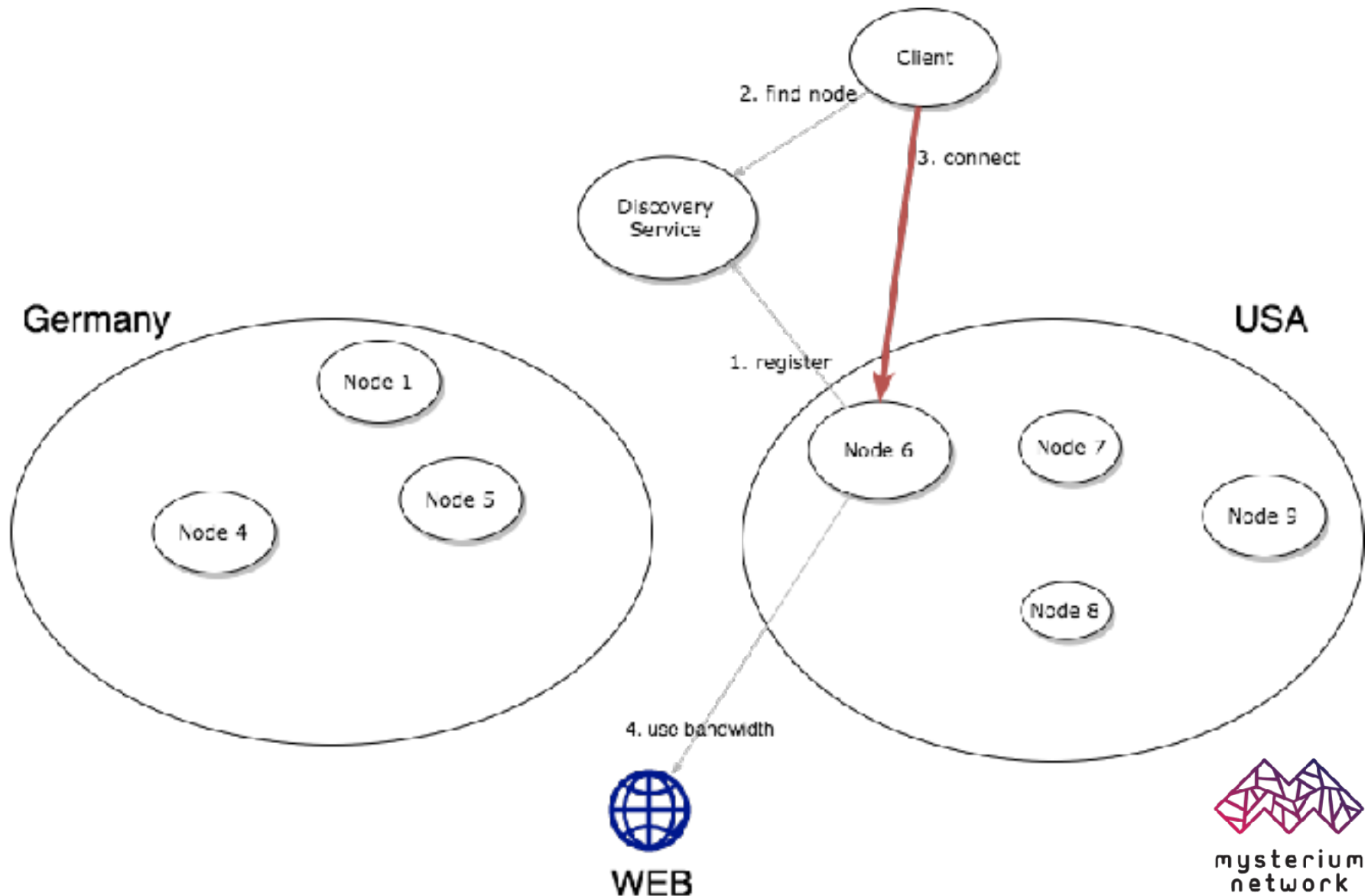
- Golang / PHP / JAVA Engineer
- Backend and Devops
- Past in Fintech, Lamoda.ru, Bigbank.eu
- Now Blockchain Developer at
www.mysterium.network



Doing Bandwidth Marketplace



Node Discovery Service



Application structure

- github.com/MysteriumNetwork/node

```
/ # ls -lah /usr/local/bin/mysterium_server
-rwxr-xr-x    1 root    root      7.8M Nov 29 08:29 /usr/local/bin/mysterium_server
```

```
/ # ls -lah /etc/mysterium-node/
total 36
drwxr-xr-x    1 root    root      4.0K Nov 29 09:57 .
drwxr-xr-x    1 root    root      4.0K Nov 29 09:49 ..
-rw-r--r--    1 root    root      1.1K Sep  4 18:46 ca.crt
-rw-r--r--    1 root    root        624 Sep  4 18:46 crl.pem
-rw-r--r--    1 root    root        423 Sep  4 18:46 dh.pem
-rw-r--r--    1 root    root      4.3K Sep  4 18:46 server.crt
-rw-r--r--    1 root    root      1.7K Sep  4 18:46 server.key
-rw-r--r--    1 root    root        635 Sep  4 18:46 ta.key
```

Binary resources

	Size	Time cold	Time warm	RAM
Binary	7.8 MB	3.4s	1.8	10.1 MB

Ubuntu image

```
Dockerfile x
1  >> FROM ubuntu:16.04
2
3  # e.g. PACKAGE_FILE="build/package/mysterium-node_linux_amd64.deb"
4  ARG PACKAGE_FILE
5  COPY $PACKAGE_FILE /tmp/mysterium-node.deb
6
7  # Install packages
8  RUN dpkg --install --force-depends /tmp/mysterium-node.deb \
9      && apt-get install -y --fix-broken \
10     && rm /tmp/mysterium-node.deb
11
12 COPY docker-entrypoint-node.sh /usr/bin/docker-entrypoint.sh
13 ENTRYPOINT ["/usr/bin/docker-entrypoint.sh"]
```

Ubuntu image resources

	Size	Time cold	Time warm	RAM
Binary	7.8 MB	3.4s	1.8	10.1 MB
Ubuntu + binary	145 MB 123 MB	60s	38s	13.2 MB

Why Golang 1: Lightweight

- Strong core library
- No dependencies
 - HTTP router
 - Logger
 - UID generator
 - NATS pub-sub library
- Executable size **7.8 MB**
- Eats RAM **12 MB**

Why Golang 2: Cross-platform

- Statically compiled
- Linux server (Debian, Centos)
- OSX client (same codebase)
- Windows (not tested yet)

- Arduino (community implemented)
- Router (willing to investigate)
- IoT enabler (same codebase for mobile)

Ubuntu image building time

```
docker ps -a -q | xargs docker rm  
docker rmi node_node-ubuntu ubuntu:16.04  
time docker-compose build node-ubuntu  
  
1💡7s user 1.08s system 5% cpu 58.417 total  
2.45s user 1.67s system 6% cpu 1:02.35 total  
2.49s user 1.73s system 6% cpu 1:01.15 total
```

```
docker ps -a -q | xargs docker rm  
docker rmi node_node-ubuntu  
time docker-compose build node-ubuntu  
  
2.59s user 1.82s system 10% cpu 41.750 total  
1.99s user 1.11s system 8% cpu 38.290 total  
1💡4s user 1.08s system 8% cpu 33.939 total
```

Ubuntu image size

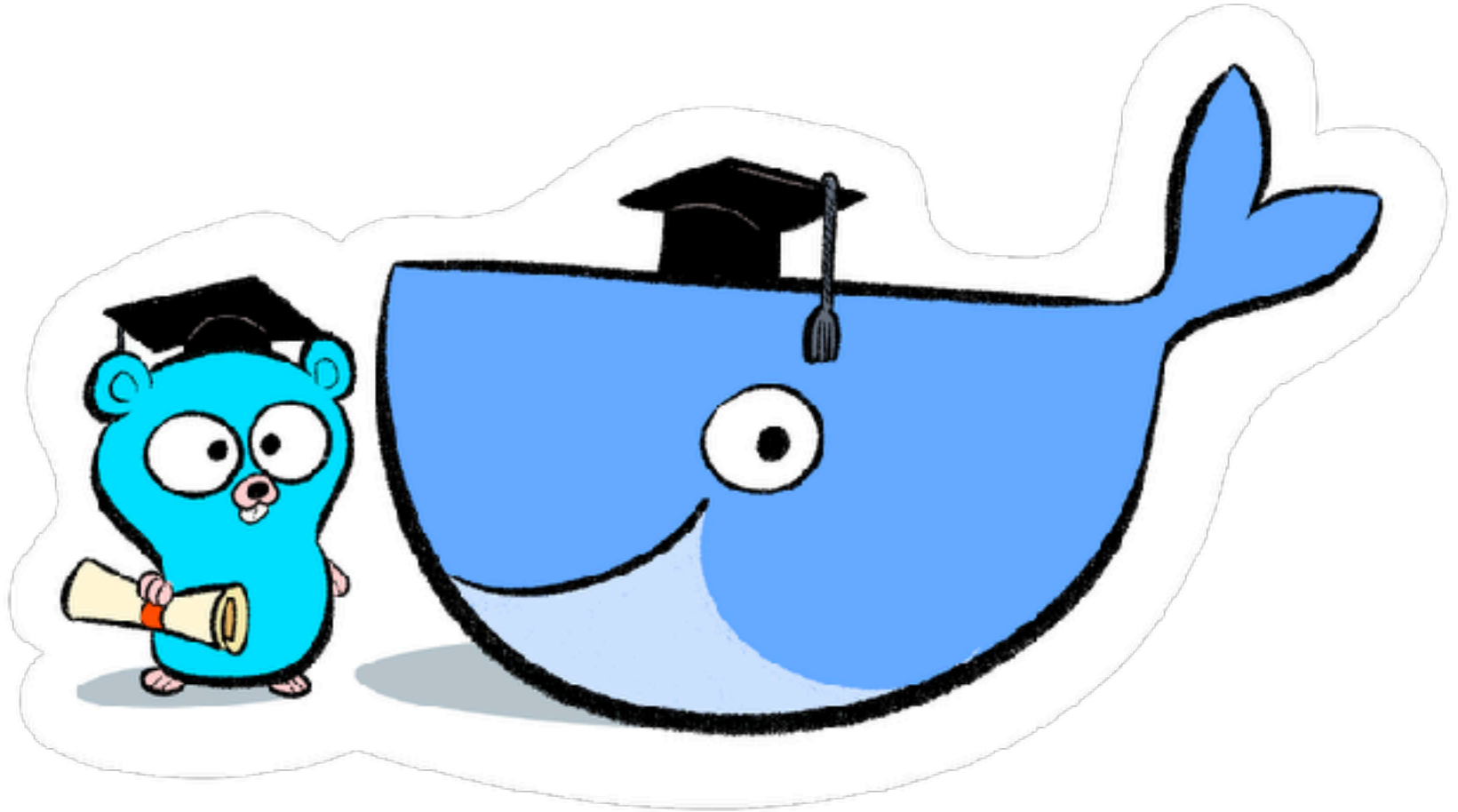
```
docker-compose up node-ubuntu
docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
node_node-ubuntu	latest	91328a3a74cf	Less than a second ago	145MB
ubuntu	16.04	20c44cd7596f	10 days ago	123MB

```
docker stats
```

CONTAINER	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
1a5a03040a7a	0.00%	13.2MiB / 1.952GiB	0.65%	6.62kB / 3.01kB	0B / 0B	10

Hey Docker, we're Effective Go :(



Golang official image

```
Dockerfile x
1  FROM golang:1.9.2-alpine
2
3  ADD . /go/src/github.com/mysterium/node
4  WORKDIR /go/src/github.com/mysterium/node
5
6  # Install packages
7  RUN apk update \
8      && apk add --no-cache iptables ca-certificates openvpn \
9      && rm -rf /var/cache/apk/*
10
11 # Compile application
12 RUN source .env \
13     && GOBIN=`pwd`/build/server CGO_ENABLED=0 go install cmd/mysterium_server/mysterium_server.go
14
15 COPY build/server/mysterium_server /usr/local/bin/mysterium_server
16 COPY docker-entrypoint-node-golang.sh /usr/local/bin/docker-entrypoint.sh
17 COPY bin/server_package/config /etc/mysterium-node
18
19 ENTRYPOINT ["/usr/local/bin/docker-entrypoint.sh"]
```

Golang images resources

	Size	Time cold	Time warm	RAM
Binary	7.8 MB	3.4s	1.8	10.1 MB
Ubuntu + binary	145 MB 123 MB	60s	38s	13.2 MB
Alpine + Golang	535 MB 269 MB	44s	43s	11.75 MB

Alpine minimal image

Dockerfile x

```
1  ➤ FROM alpine:3.6
2
3  # Install packages
4  RUN apk update \
5      && apk add --no-cache iptables ca-certificates openvpn \
6      && rm -rf /var/cache/apk/*
7
8  # Install application
9  COPY build/server/mysterium_server /usr/local/bin/mysterium_server
10 COPY docker-entrypoint-node-alpine.sh /usr/local/bin/docker-entrypoint.sh
11 COPY bin/server_package/config /etc/mysterium-node
12
13 ENTRYPOINT ["/usr/local/bin/docker-entrypoint.sh"]
14 EXPOSE 1194
```


Alpine builder image

```
Dockerfile x
1  FROM golang:1.9.2-alpine
2
3  WORKDIR /go/src/github.com/mysterium/node
4  ENTRYPOINT ["/go/src/github.com/mysterium/node/docker-entrypoint-builder.sh"]

docker-entrypoint-builder.sh x
1  #!/bin/sh
2  set -e
3
4  source .env
5  GOBIN=`pwd`/build/server CGO_ENABLED=0 go install cmd/mysterium_server/mysterium_server.go
```

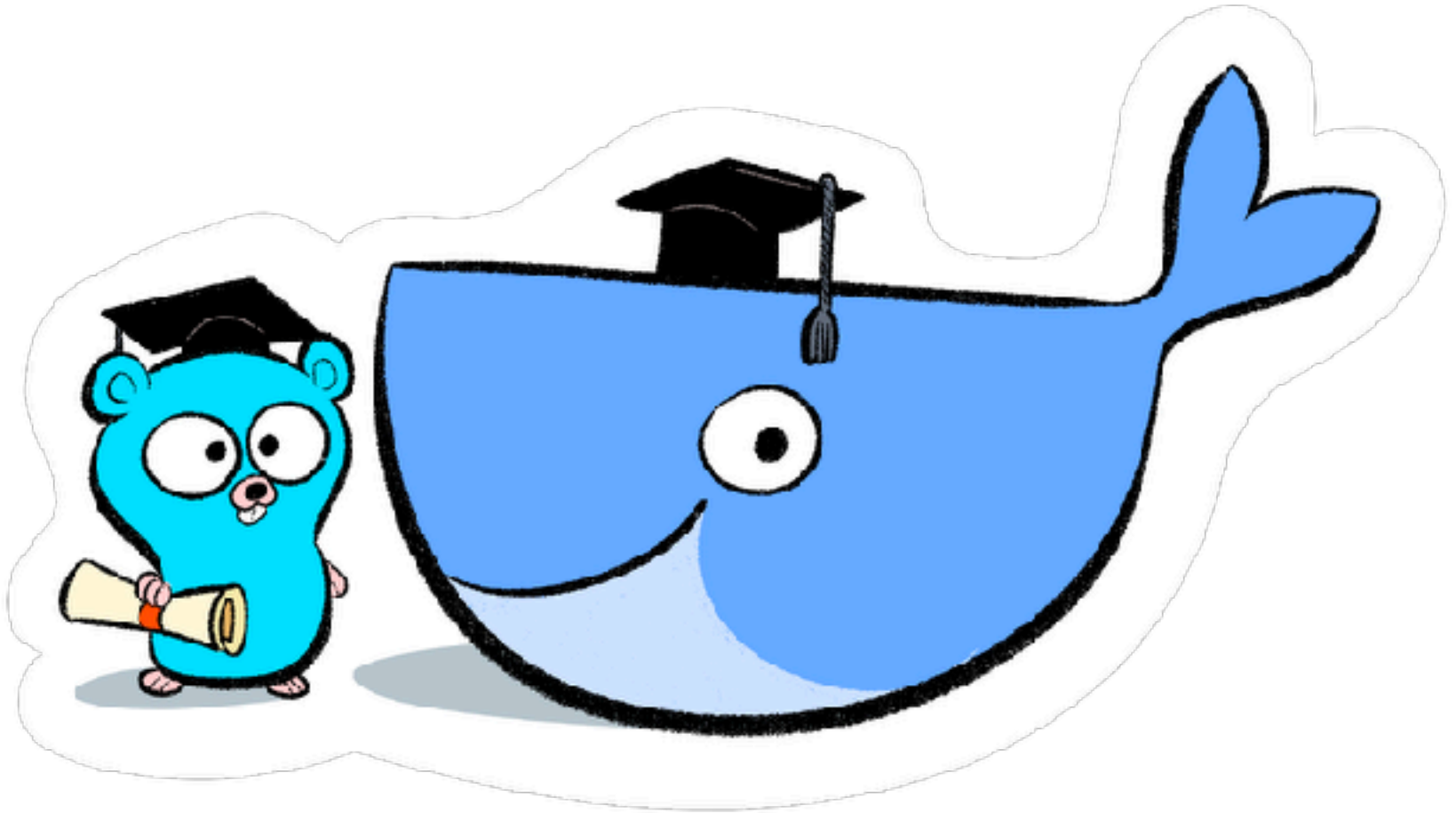
Shipping resources comparison

	Size	Time cold	Time warm	RAM
Binary	7.8 MB	3.4s	1.8	10.1 MB
Ubuntu + binary	145 MB 123 MB	60s	38s	13.2 MB
Alpine + Golang	535 MB 269 MB	44s	43s	11.75 MB
Alpine + binary	16.5 MB 3.97 MB	30s	27s	12.93MB

Shipping resources comparison

	Size	Time cold	Time warm	RAM
Binary	7.8 MB	3.4s	1.8	10.1 MB
Ubuntu + binary	145 MB 123 MB	60s	38s	13.2 MB
Alpine + Golang	535 MB 269 MB	44s	43s	11.75 MB
Alpine + binary	16.5 MB 3.97 MB	30s	27s	12.93MB
AlpineGlibc + binary	35.5MB 11.2MB	-	22s	17MB

Lets make Go Effective again



Effective Alpine



alpine
Linux



docker

Questions?

- <https://github.com/MysteriumNetwork/node>
- valdas@mysterium.network