



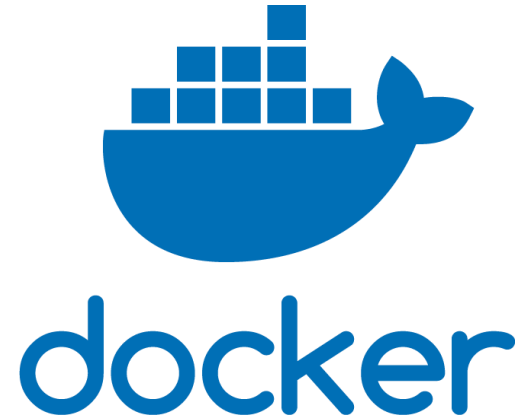
# MAINFLUX

Open Source Internet of Things Technology & Consulting Services

Technology Overview – IoT Fuse 2019  
Part 2 – Mainflux Deployment

# MAINFLUX

Local Deployment – Docker



All the scripts and slide are in the github repo

<https://github.com/janko-isidorovic/iotfuse2019>

Docker scripts are in:

<https://github.com/janko-isidorovic/iotfuse2019/tree/master/docker>

Kubernetes scripts are in:

<https://github.com/janko-isidorovic/iotfuse2019/tree/master/k8s>

- 
- The diagram illustrates the Mainflux IoT Core architecture. On the left, a green vertical bar represents **NGINX Ingress**. The core components are as follows:
- Message Reader**: Receives data from the ingress.
  - Adapters**: HTTP Adapter, Web Socket Adapter, CoAP Adapter, and MQTT Adapter. These feed into the **NATS** message bus.
  - Message DB Writer**: Receives data from the NATS bus and writes to **InfluxDB**.
  - NATS**: A central message bus (blue vertical bar) that also feeds into the **Redis - Event Log** (red vertical bar).
  - Message Normalizer**: Receives data from the NATS bus and feeds into **PostgreSQL**.
  - Redis**: Two instances are shown, one connected to the adapters and another to the **Things Devices/Apps** and **Users**.
  - Things Devices/Apps** and **Users**: Represented as dark blue blocks at the bottom.
  - InfluxDB** and **PostgreSQL**: Two database instances on the right.
- On the far right, a **Grafana** dashboard is shown, displaying various time-series and gauge charts for monitoring the system.

# MAINFLUX – Docker Compose – 1

Install Docker and Docker Compose

<https://www.docker.com/get-started>

Download Mainflux docker compose scripts

*git clone https://github.com/mainflux/mainflux.git*

*cd mainflux/docker*

Modify the ports to fit your laptop

*9080:80*

*9443:443*

```
22 services:
23   nginx:
24     image: nginx:1.14.2
25     container_name: mainflux-nginx
26     restart: on-failure
27     volumes:
28       - ./nginx/nginx-${AUTH-key}.conf:/etc/nginx/nginx.conf
29       - ./ssl/authorization.js:/etc/nginx/authorization.js
30       - ./ssl/certs/mainflux-server.crt:/etc/ssl/certs/mainflux-server.crt
31       - ./ssl/certs/ca.crt:/etc/ssl/certs/ca.crt
32       - ./ssl/certs/mainflux-server.key:/etc/ssl/private/mainflux-server.key
33       - ./ssl/dhparam.pem:/etc/ssl/certs/dhparam.pem
34     ports:
35       - 80:80
36       - 443:443
37       - 8883:8883
38     networks:
39       - mainflux-base-net
```

# MAINFLUX – Docker Compose – 2

Start docker compose script

```
docker-compose -f docker/docker-compose.yml up -d
```

For InfluxDB support start addon docker composition

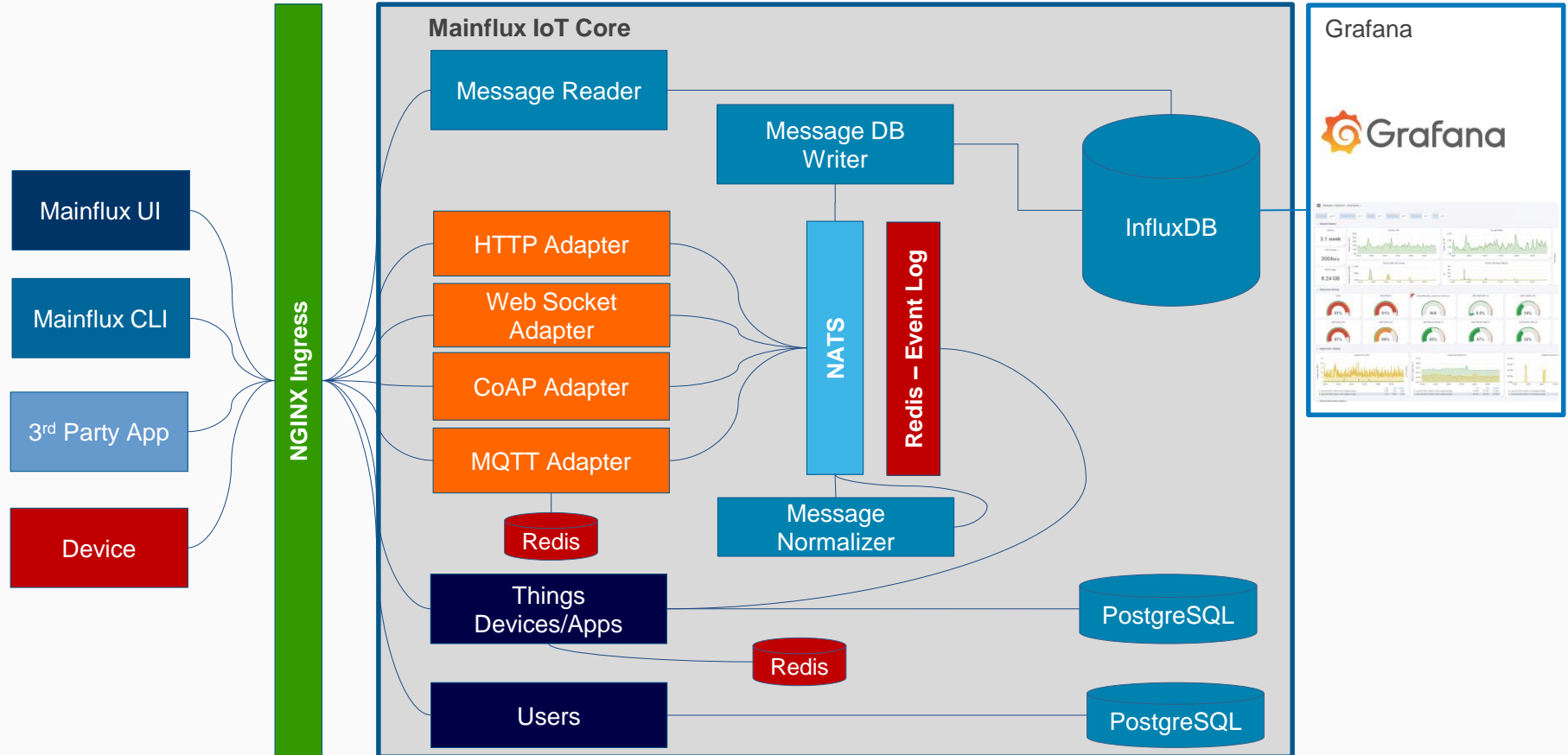
```
docker-compose -f docker/addons/influxdb-writer/docker-compose.yml up -d
```

For InfluxDB Message Reader start addon docker composition

```
docker-compose -f docker/addons/influxdb-reader/docker-compose.yml up -d
```

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
abcc3454c3ca	mainflux-influxdb-reader	0.00%	2.156MiB / 1.934GiB	0.11%	1.12kB / 554B	0B / 0B	8
43dafbbec5ce	mainflux-influxdb-writer	0.00%	2.164MiB / 1.934GiB	0.11%	1.75kB / 729B	7.96MB / 0B	9
dcbbcb0463705	mainflux-grafana	0.06%	18.23MiB / 1.934GiB	0.92%	46.6kB / 5.18MB	4.13MB / 7.9MB	10
6764e5fe2514	mainflux-influxdb	0.14%	9.652MiB / 1.934GiB	0.49%	9.08kB / 2.65kB	23.9MB / 172kB	11
22c5a0cb0fab	mainflux-ws	0.00%	2.375MiB / 1.934GiB	0.12%	3.92kB / 2.37kB	7.16MB / 0B	10
a209359ebe03	mainflux-http	0.00%	2.27MiB / 1.934GiB	0.11%	4.17kB / 2.47kB	8.65MB / 0B	10
1e9136c6230d	mainflux-coap	0.00%	2.332MiB / 1.934GiB	0.12%	4.3kB / 2.65kB	3.87MB / 0B	8
1aceb3e93ba9	mainflux-mqtt	0.00%	47.02MiB / 1.934GiB	2.37%	24.5kB / 10.6kB	37.8MB / 0B	12
7bb4fefb1477	mainflux-things	0.00%	3.496MiB / 1.934GiB	0.18%	11.1kB / 8.26kB	4.47MB / 0B	9
5cc74e34525d	mainflux-users	0.00%	3.066MiB / 1.934GiB	0.15%	6.41kB / 3.4kB	2.37MB / 0B	10
850795de109c	mainflux-normalizer	0.00%	2.238MiB / 1.934GiB	0.11%	4.31kB / 2.02kB	2.68MB / 0B	8
87cb2ef2a3ff	mainflux-es-redis	0.30%	1.637MiB / 1.934GiB	0.08%	3.44kB / 4.04kB	4.1kB / 0B	4
8c9ebbd32d3d	mainflux-mqtt-redis	0.24%	1.664MiB / 1.934GiB	0.08%	10.7kB / 16.8kB	766kB / 0B	4
50c85763a303	mainflux-ui	0.00%	2.082MiB / 1.934GiB	0.11%	5.57kB / 2.92kB	4.53MB / 0B	2
1a750f15b3e0	mainflux-users-db	0.01%	6.816MiB / 1.934GiB	0.34%	3.86kB / 1.49kB	2.96MB / 664kB	8
9e3131ff5b39	mainflux-nats	0.07%	6.652MiB / 1.934GiB	0.34%	12.3kB / 9.54kB	5.71MB / 0B	9
690cfee74686	mainflux-things-redis	0.27%	1.656MiB / 1.934GiB	0.08%	2.77kB / 0B	2.96MB / 0B	4
4d402764a81e	mainflux-things-db	0.02%	6.949MiB / 1.934GiB	0.35%	5.47kB / 3.16kB	4.33MB / 672kB	9
88e8105f2a31	mainflux-nginx	0.00%	3.117MiB / 1.934GiB	0.16%	12.3kB / 11.5kB	8.07MB / 0B	3

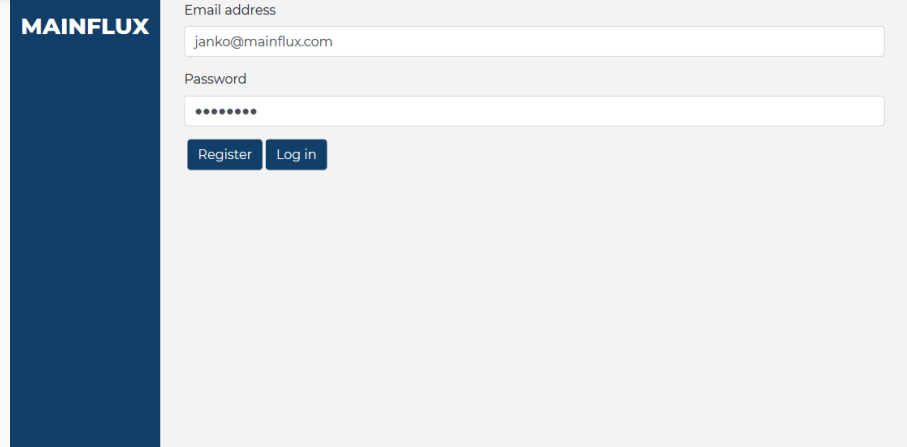
# MAINFLUX – Docker Compose – 3



# MAINFLUX – Docker Compose – 4

Login to Mainflux UI

`http://localhost:9080/`



The image shows the Mainflux UI login page. On the left is a dark blue vertical bar with the 'MAINFLUX' logo in white. The main area is light gray and contains the following elements:

- 'Email address' label above a text input field containing 'janko@mainflux.com'.
- 'Password' label above a text input field containing seven dots.
- 'Register' and 'Log in' buttons below the password field.

Login to Grafana

`http://localhost:3001/`

User: admin

Pass: admin



The image shows the Grafana login page. It features a dark blue background with a subtle grid pattern. On the left is the Grafana logo (a yellow and orange gear-like swirl) and the word 'Grafana' in white. On the right are the following elements:

- 'email or username' label above a text input field.
- 'password' label above a text input field.
- 'Log In' button below the password field.
- 'Forgot your password?' link to the right of the 'Log In' button.

At the bottom, there is a footer with links: 'Docs', 'Support Plans', 'Community', 'Grafana v5.1.3 (commit: 087143285)', and 'New version available!'.



# MAINFLUX – Docker Compose – 5

## Create thing using API

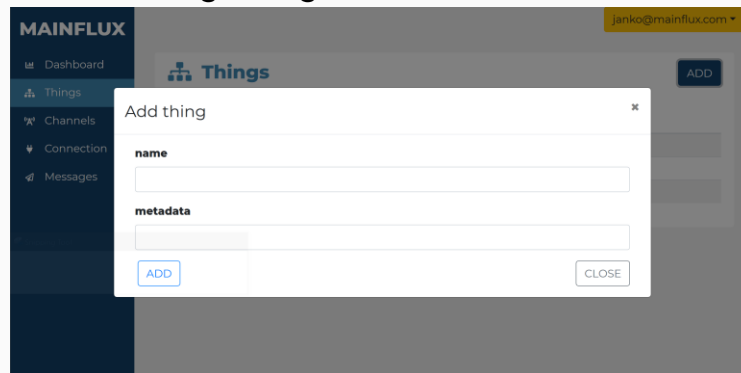
```
curl -s -S -i --cacert docker/ssl/certs/mainflux-server.crt --insecure -X POST -H "Content-Type: application/json" https://localhost/tokens -d '{"email": "john.doe@email.com", "password": "123"}'
```

```
curl -s -S -i --cacert docker/ssl/certs/mainflux-server.crt --insecure -X POST -H "Content-Type: application/json" -H "Authorization: <user_auth_token>" https://localhost/things -d '{"name": "dev02"}'
```

## Create channel using API

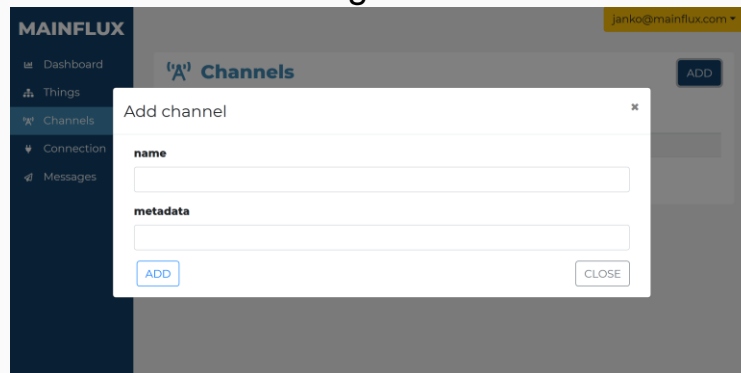
```
curl -s -S -i --cacert docker/ssl/certs/mainflux-server.crt --insecure -X POST -H "Content-Type: application/json" -H "Authorization: <user_auth_token>" https://localhost/channels -d '{"name": "mychan"}'
```

## Create thing using UI



The screenshot shows the MAINFLUX web interface. On the left is a sidebar with navigation links: Dashboard, Things, Channels, Connection, and Messages. The main area is titled 'Things' and contains an 'ADD' button. A modal dialog box titled 'Add thing' is open in the center. It has two input fields: 'name' and 'metadata'. At the bottom of the dialog are 'ADD' and 'CLOSE' buttons. The user's email 'janko@mainflux.com' is visible in the top right corner.

## Create channel using UI



The screenshot shows the MAINFLUX web interface. On the left is a sidebar with navigation links: Dashboard, Things, Channels, Connection, and Messages. The main area is titled 'Channels' and contains an 'ADD' button. A modal dialog box titled 'Add channel' is open in the center. It has two input fields: 'name' and 'metadata'. At the bottom of the dialog are 'ADD' and 'CLOSE' buttons. The user's email 'janko@mainflux.com' is visible in the top right corner.

## Connect thing to channel

MAINFLUX

Dashboard

Things

Channels

Connection

Messages

janko@mainflux.com

Things

Name	ID	
app01	23fe61c5-1267-44e7-85ae-a46f53e121cf	<input type="checkbox"/>
dev01	4d52abe0-8b52-413f-9273-7f928618e0df	<input type="checkbox"/>
dev02	cd2edc22-c138-4540-9635-0210ff83b261	<input type="checkbox"/>

Connect

Disconnect

Channels

Name	ID	
channel02	5b189d65-d0cd-4edf-b153-fd9a1b1ca589	<input type="checkbox"/>
channel01	c94b999a-f897-4ce0-8bd1-2f7a86e97526	<input type="checkbox"/>

## Configure Datasource

Name: Mainflux Default ☒

Type: InfluxDB

HTTP

URL: http://mainflux-influxdb:8086

Access: Server (Default) [Help](#)

Auth

Basic Auth ☐ With Credentials ☐

TLS Client Auth ☐ With CA Cert ☐

Skip TLS Verification (Insecure) ☐

Advanced HTTP Settings

Whitelisted Cookies [Add Name](#)

InfluxDB Details

Database: mainflux

User: mainflux Password: .....

Database Access

Setting the database for this datasource does not deny access to other databases. The InfluxDB query syntax allows switching the database in the query. For example: `SHOW MEASUREMENTS ON _internal` or `SELECT * FROM "_internal"."database" LIMIT 10`

To support data isolation and security, make sure appropriate permissions are configured in InfluxDB.

Min time interval: 10s

## Configure Graph

Graph General Metrics Axes Legend Display Alert Time range

Data Source: Mainflux

FROM autogen messages WHERE name = voltage +

SELECT field (value) +

GROUP BY +

FORMAT AS Time series

ALIAS BY Naming pattern

Add Query

## Linux

```
curl -X POST http://localhost:9080/http/channels/ChannelID/messages -H 'Authorization: DeviceKey' -H  
'Content-Type: application/senml+json' -d '{"n":"voltage", "u":"V", "v":120}'
```

## Windows

```
curl -X POST "http://localhost:9080/http/channels/ChannelID/messages" -H "Authorization: DeviceKey" -H  
"Content-Type: application/senml+json" -d "[{"n":"voltage", "u":"V", "v":123.2}]"
```

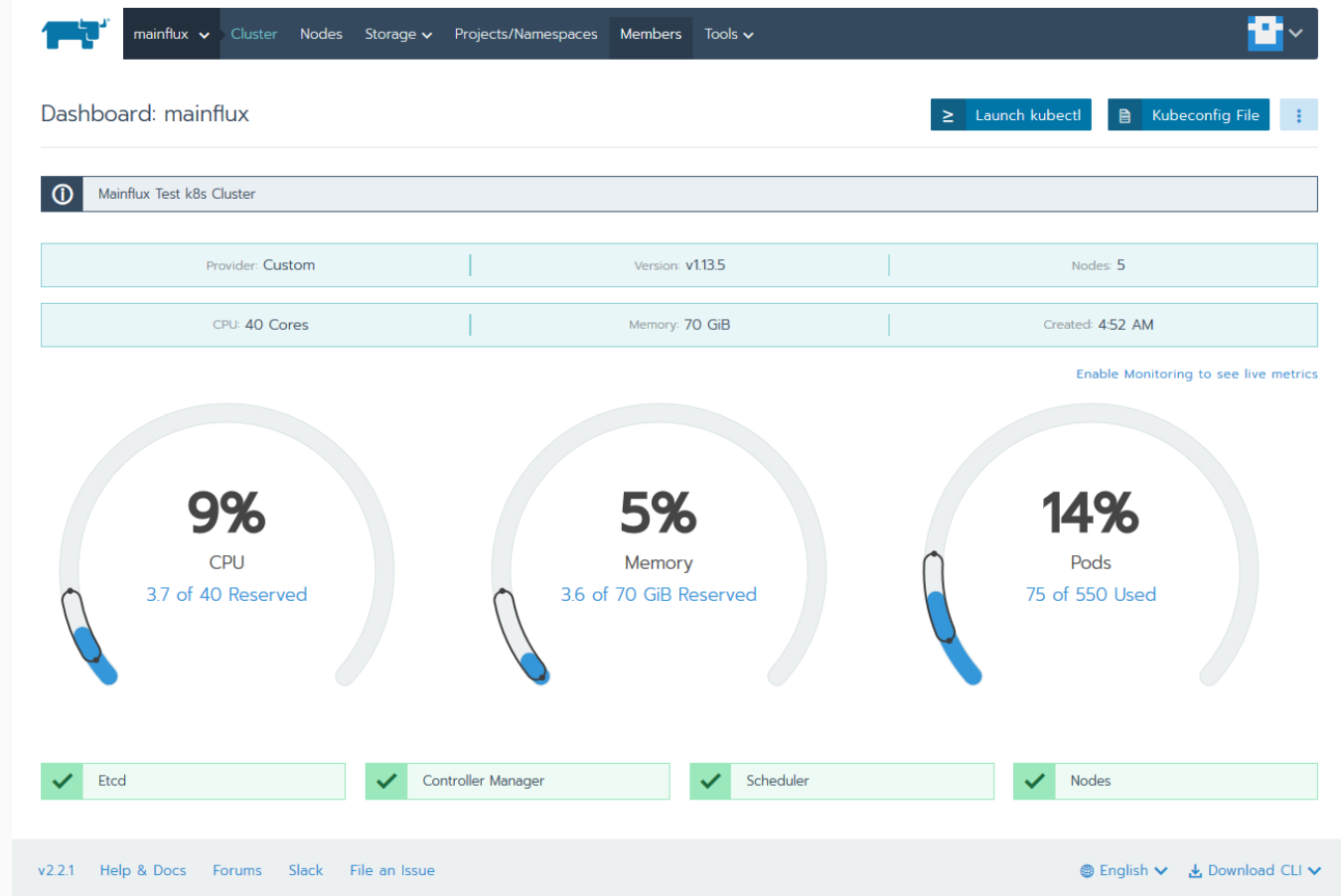
# MAINFLUX

Kubernetes Deployment



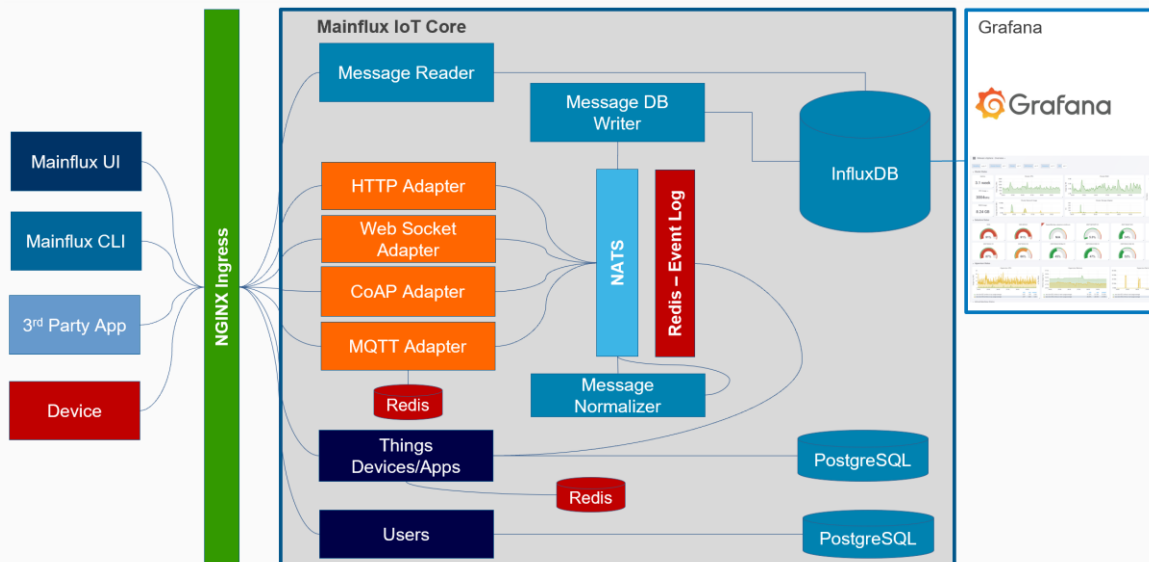
**kubernetes**

# MAINFLUX – Kubernetes Deployment Using Rancher and YAML Files



# MAINFLUX – Kubernetes Deployment Using k8s YAML Files

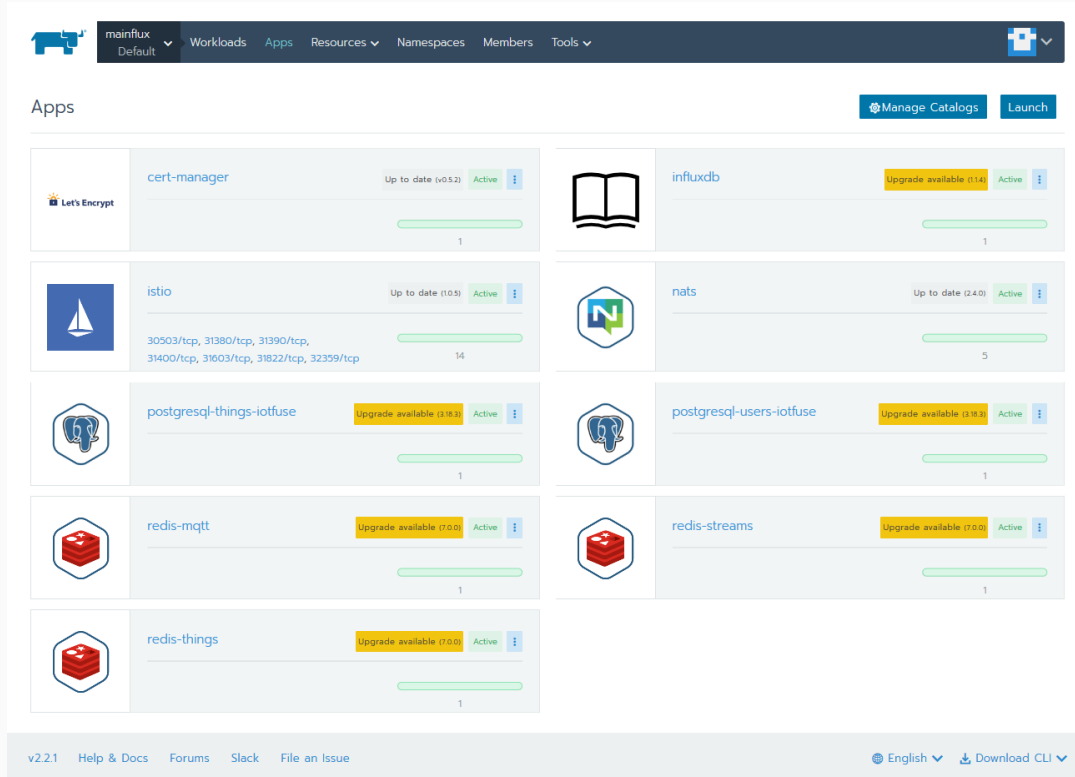
- Mainflux Components - Infrastructure
  - Postgres Databases
    - Users
    - Things
  - Redis Cache
    - MQTT Adapter
    - Things
    - Event Stream
  - NATS
  - ISTIO
- Mainflux Components – Core Services
  - Users
  - Things
  - HTTP, WS, MQTT, CoAP Adapters
  - Normalizer
  - Mainflux UI
- Mainflux Add-ons
  - InfluxDB
  - InfluxDB Writer
  - InfluxDB Reader
  - Grafana



# Rancher – Setup Infrastructure Components

## Mainflux Components - Infrastructure

- Postgres Databases
  - Users
  - Things
- Redis Cache
  - MQTT Adapter
  - Things
  - Event Stream
- NATS
- Istio – for gRPC load balancing
- Cert manager – Lets Encrypt



The screenshot displays the Rancher management interface for infrastructure components. The top navigation bar includes the Rancher logo, a dropdown menu for 'mainflux Default', and links for 'Workloads', 'Apps', 'Resources', 'Namespaces', 'Members', and 'Tools'. Below the navigation bar, the 'Apps' section is active, showing a grid of installed applications. Each application card includes an icon, the app name, version information, status, and a progress bar.

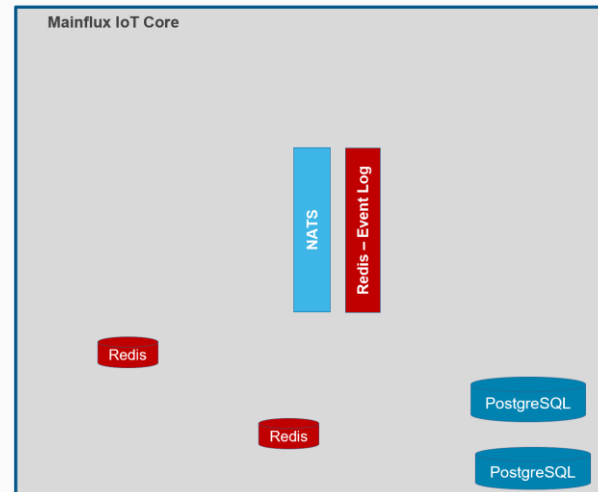
App Name	Icon	Status	Version	Upgrade Available	Count
cert-manager	Let's Encrypt	Active	Up to date (v0.5.2)	No	1
istio	Sailboat	Active	Up to date (1.0.5)	No	14
postgresql-things-iotfuse	Elephant	Active	Upgrade available (1.10.0)	Yes	1
redis-mqtt	Redis	Active	Upgrade available (7.0.0)	Yes	1
redis-things	Redis	Active	Upgrade available (7.0.0)	Yes	1
influxdb	Open Book	Active	Upgrade available (0.14)	Yes	1
nats	NATS	Active	Up to date (2.4.0)	No	5
postgresql-users-iotfuse	Elephant	Active	Upgrade available (1.10.0)	Yes	1
redis-streams	Redis	Active	Upgrade available (7.0.0)	Yes	1

The bottom of the interface shows the version 'v2.2.1' and links for 'Help & Docs', 'Forums', 'Slack', and 'File an Issue'. On the right, there are links for 'English' and 'Download CLI'.



# Rancher – Setup Infrastructure Components

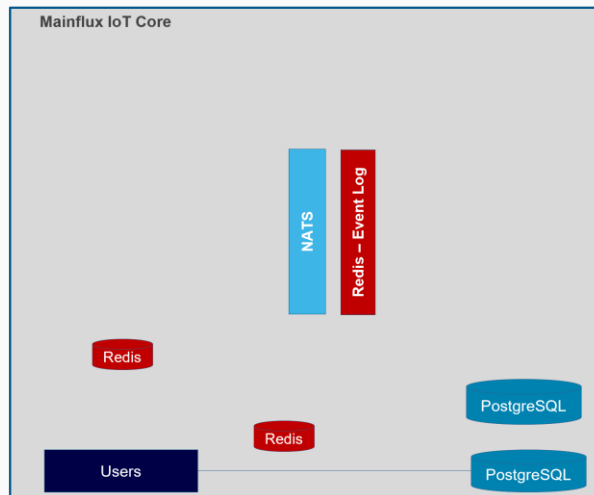
- Setup *mainflux* namespaces:
- *mainflux* - Enable Istio side cart injection
- *nats*
- *redis*
- *postgres*
- *Influxdb*
- *istio*



## Mainflux Users

```
25 spec:
26   containers:
27   - env:
28     - name: MF_USERS_DB
29       value: users
30     - name: MF_USERS_DB_HOST
31       value: postgresql-users-iotfuse-postgresql.mainflux-postgresql
32     - name: MF_USERS_DB_PASS
33       value: mainflux
34     - name: MF_USERS_DB_PORT
35       value: '5432'
36     - name: MF_USERS_DB_USER
37       value: mainflux
38     - name: MF_USERS_GRPC_PORT
39       value: '8181'
40     - name: MF_USERS_HTTP_PORT
41       value: '8180'
42     - name: MF_USERS_LOG_LEVEL
43       value: debug
44     - name: MF_USERS_SECRET
45       value: secret
46     image: mainflux/users:latest
47     imagePullPolicy: Always
48     name: users
```

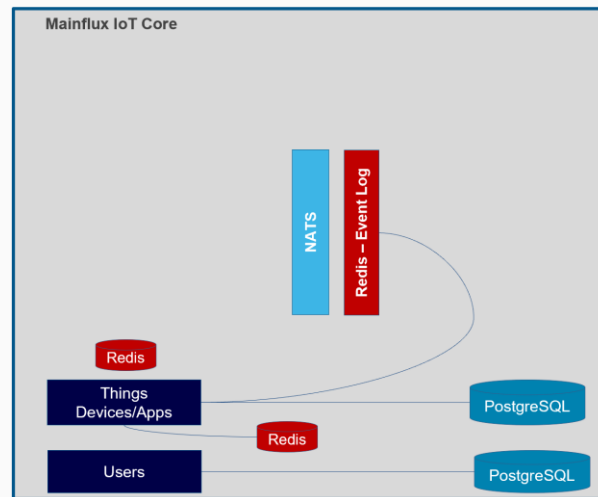
*kubectl create -f 1-users.yml*



## Mainflux Things

*kubectl create -f 2-things.yml*

```
27 - env:
28   - name: MF_THINGS_CACHE_URL
29     value: redis-things-master.mainflux-redis:6379
30   - name: MF_THINGS_DB
31     value: things
32   - name: MF_THINGS_DB_HOST
33     value: postgresql-things-iotfuse-postgresql.mainflux-postgresql
34   - name: MF_THINGS_DB_PASS
35     value: mainflux
36   - name: MF_THINGS_DB_PORT
37     value: '5432'
38   - name: MF_THINGS_DB_USER
39     value: mainflux
40   - name: MF_THINGS_ES_URL
41     value: redis-streams-master.mainflux-redis:6379
```



# kubectl – Setup Protocol Adapters

- Mainflux HTTP

```
29 | | | | - env:
30 | | | | - name: MF_HTTP_ADAPTER_LOG_LEVEL
31 | | | |   value: debug
32 | | | | - name: MF_HTTP_ADAPTER_PORT
33 | | | |   value: '8185'
34 | | | | - name: MF_NATS_URL
35 | | | |   value: nats://nats-nats-client.mainflux-nats:4222
36 | | | | - name: MF_THINGS_URL
37 | | | |   value: things:8183
38 | | | | image: mainflux/http:latest
```

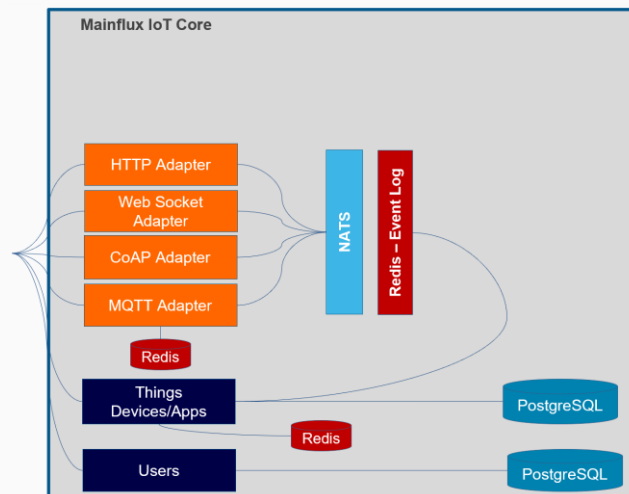
- Mainflux MQTT

```
88 | | | | - env:
89 | | | | - name: MF_MQTT_ADAPTER_LOG_LEVEL
90 | | | |   value: debug
91 | | | | - name: MF_MQTT_ADAPTER_PORT
92 | | | |   value: '1883'
93 | | | | - name: MF_MQTT_ADAPTER_REDIS_HOST
94 | | | |   value: redis-mqtt-master.mainflux-redis
95 | | | | - name: MF_MQTT_ADAPTER_WS_PORT
96 | | | |   value: '8880'
97 | | | | - name: MF_MQTT_INSTANCE_ID
98 | | | |   value: mqtt-adapter-1
99 | | | | - name: MF_NATS_URL
100 | | | |   value: nats://nats-nats-client.mainflux-nats:4222
101 | | | | - name: MF_THINGS_URL
102 | | | |   value: things:8183
```

- Mainflux WebSocket

- Mainflux CoAP

*kubectl create -f 3-protocol-adapters.yml*



# kubectl – Setup Normalizer

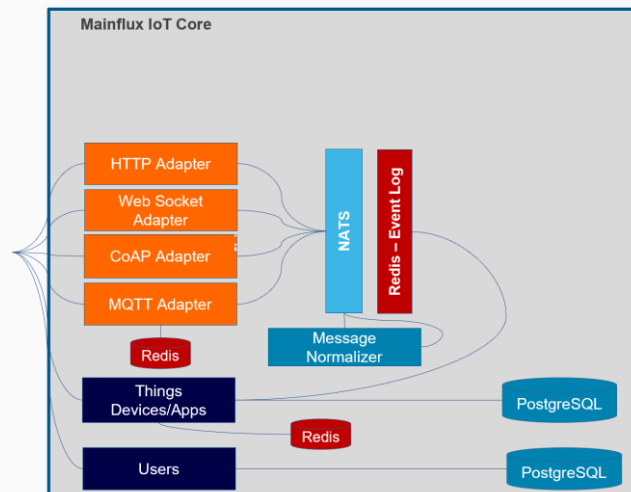
## Mainflux Normalizer

### Disable istio side car injection

```
22 | | | annotations:  
23 | | |   sidecar.istio.io/inject: 'false'
```

```
28 | | | - env:  
29 | | |   - name: MF_NATS_URL  
30 | | |     value: nats://nats-nats-client.mainflux-nats:4222  
31 | | |   - name: MF_NORMALIZER_LOG_LEVEL  
32 | | |     value: debug  
33 | | |   - name: MF_NORMALIZER_PORT  
34 | | |     value: '8184'
```

*kubectl create -f 4-normalizer.yml*



# kubectl – Setup Ingress Controller

## Two Ingress Controllers

We need different rewrite targets for users, things and Protocol adapters

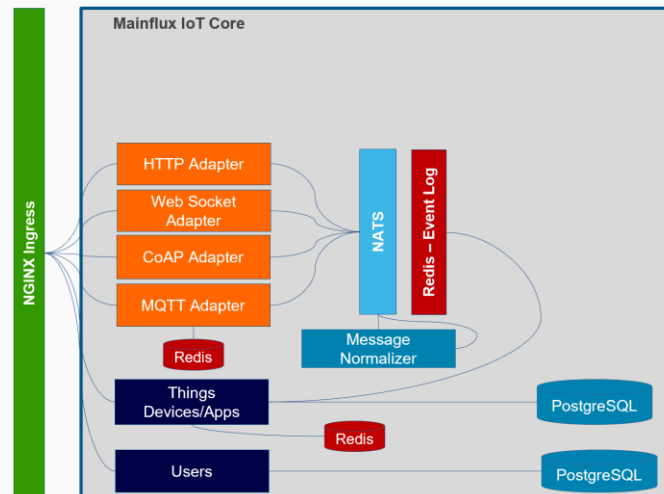
`http://localhost/users` -> `users:8180`

`http://localhost/http` -> `http:8185/http`

*`kubectl create -f 5-ingress.yaml`*

*`kubectl create -f 5-ingress-http-ws.yaml`*

```
4 annotations:  
5   certmanager.k8s.io/cluster-issuer: letsencrypt-prod  
6   kubernetes.io/tls-acme: "true"  
7   nginx.ingress.kubernetes.io/rewrite-target: /  
8   nginx.ingress.kubernetes.io/secure-backends: "true"
```

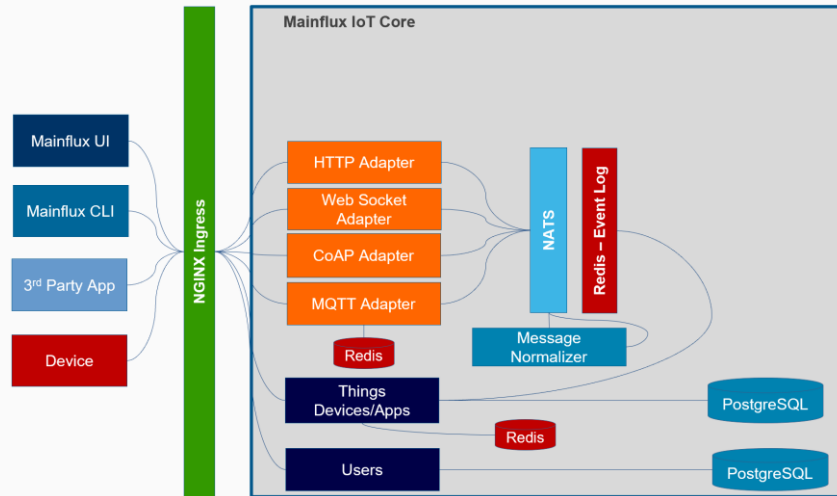


# kubectl – Setup UI

- Mainflux UI

annotations:  
sidecar.istio.io/inject: 'false'

*kubectl create -f 6-ui.yml*



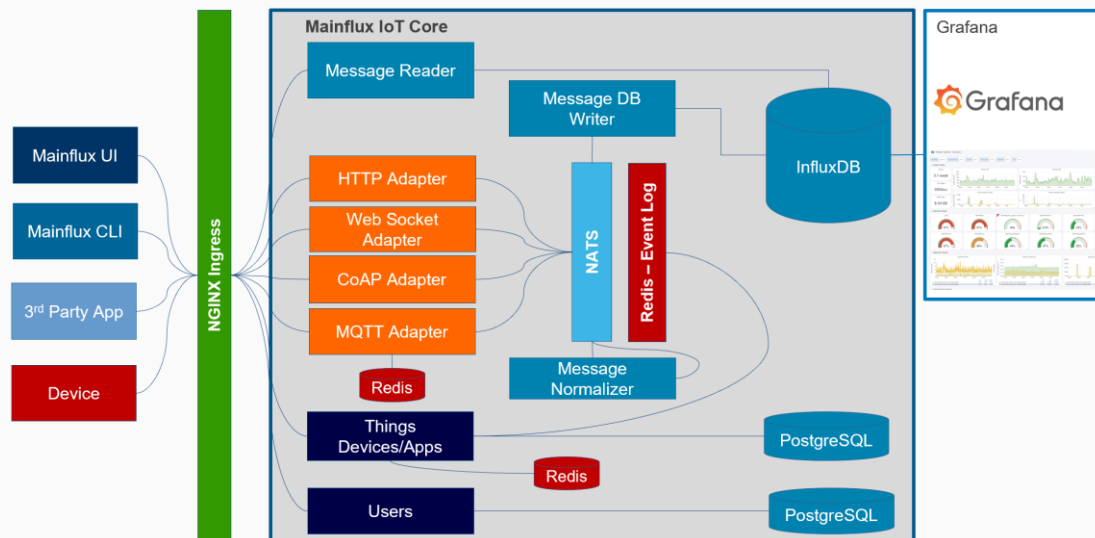
# kubectl – Setup Add-ons – DB Writer, DB Reader, Grafana

## Add DB Writer, DB Reader, Grafana

```
92 | annotations:  
93 |   sidecar.istio.io/inject: 'false'
```

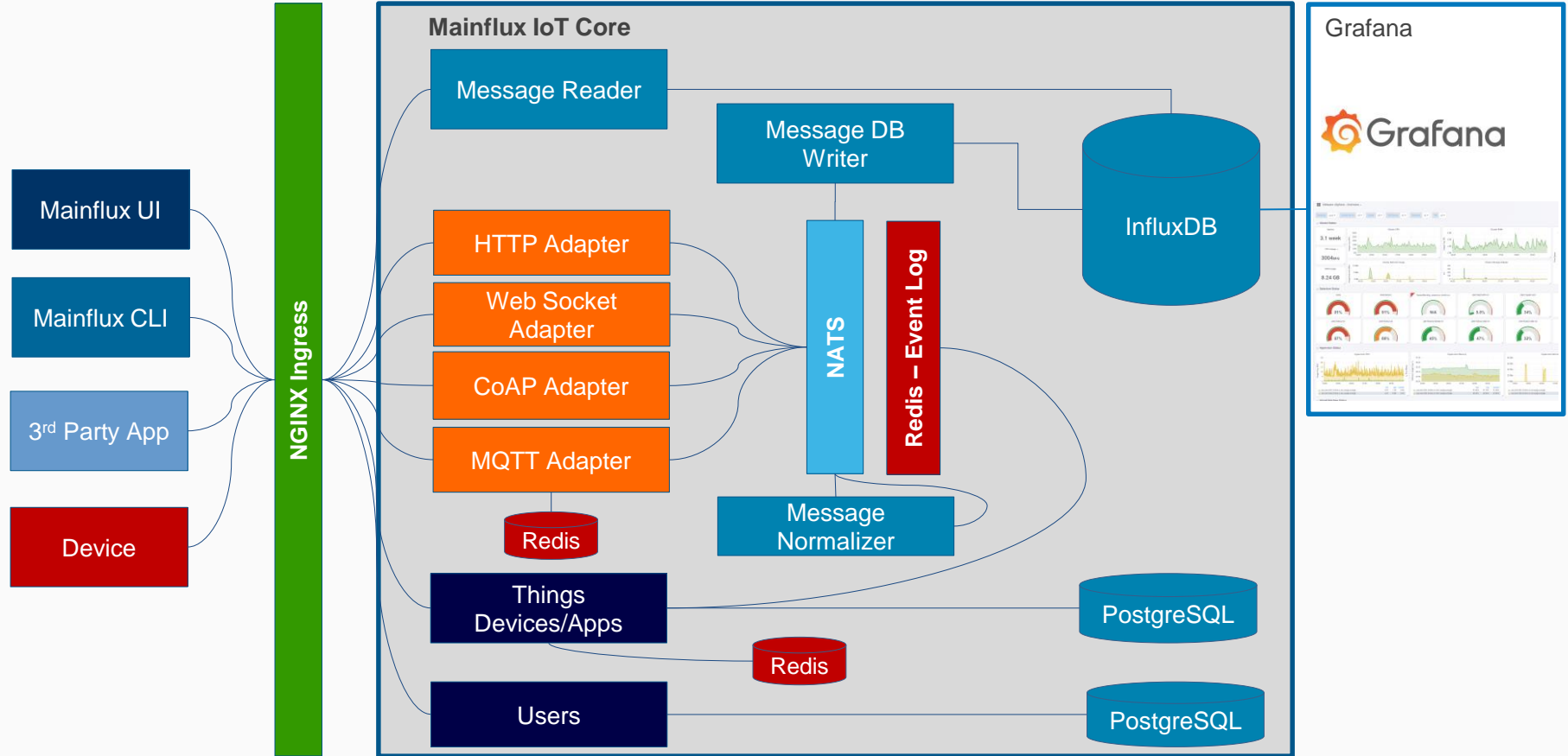
```
98 | - env:  
99 |   - name: MF_INFLUX_WRITER_BATCH_SIZE  
100 |     value: '5000'  
101 |   - name: MF_INFLUX_WRITER_BATCH_TIMEOUT  
102 |     value: '5'  
103 |   - name: MF_INFLUX_WRITER_DB_HOST  
104 |     value: influxdb.mainflux-influxdb  
105 |   - name: MF_INFLUX_WRITER_DB_NAME  
106 |     value: mainflux  
107 |   - name: MF_INFLUX_WRITER_DB_PASS  
108 |     value: mainflux  
109 |   - name: MF_INFLUX_WRITER_DB_PORT  
110 |     value: '8086'  
111 |   - name: MF_INFLUX_WRITER_DB_USER  
112 |     value: mainflux  
113 |   - name: MF_INFLUX_WRITER_LOG_LEVEL  
114 |     value: debug  
115 |   - name: MF_INFLUX_WRITER_PORT  
116 |     value: '8900'  
117 |   - name: MF_NATS_URL  
118 |     value: nats://nats-nats-client.mainflux-nats:4222
```

*kubectl create -f 7-add-ons-influxdb.yml*





# MAINFLUX + InfluxDB + Grafana



- Create Kubernetes Horizontal POD Auto Scaler

```
kubectl autoscale deployment mqtt-adapter --cpu-percent=50 --min=1 --max=10
```

- Manual Scaling using Rancher UI

Redeploy

Pause Orchestration

Download YAML

Delete

1 Workload

Search

<input type="checkbox"/>	State	Name	Image	Scale
Namespace: mainflux				
<input type="checkbox"/>	Active	coap-adapter	mainflux/coap:latest 1 Pod / Created 8 days ago	1
<input type="checkbox"/>	Active	http-adapter	mainflux/http:latest 1 Pod / Created 8 days ago	1
<input type="checkbox"/>	Active	influxdb-reader	mainflux/influxdb-reader:latest 1 Pod / Created 8 days ago	1
<input type="checkbox"/>	Active	influxdb-writer	mainflux/influxdb-writer:latest 1 Pod / Created 8 days ago	1
<input checked="" type="checkbox"/>	Active	mqtt-adapter	mainflux/mqtt:latest 5 Pods / Created 8 days ago	5
				<div><div>-</div><div>+</div></div>
<input type="checkbox"/>	Active	mqtt-ssl	nginx 1 Pod / Created 2 days ago	1
<input type="checkbox"/>	Active	normalizer	mainflux/normalizer:latest 1 Pod / Created 8 days ago	1
<input type="checkbox"/>	Active	things	mainflux/things:latest 1 Pod / Created 8 days ago	1
<input type="checkbox"/>	Active	ui	mainflux/ui:latest 1 Pod / Created 4 days ago	1
<input type="checkbox"/>	Active	users	mainflux/users:latest 1 Pod / Created 8 days ago	1
<input type="checkbox"/>	Active	ws-adapter	mainflux/ws:latest 1 Pod / Created 8 days ago	1

## LINUX

```
curl -X POST https://k8s-dev.mainflux.com/http/channels/7830ec4d-d506-45a0-945d-4a3249d2c417/messages -H 'Authorization: 80cca27f-5572-4f98-9102-62b5dd01ce45' -H 'Content-Type: application/senml+json' -d '{"n":"voltage", "u":"V", "v":120}'
```

## WINDOWS

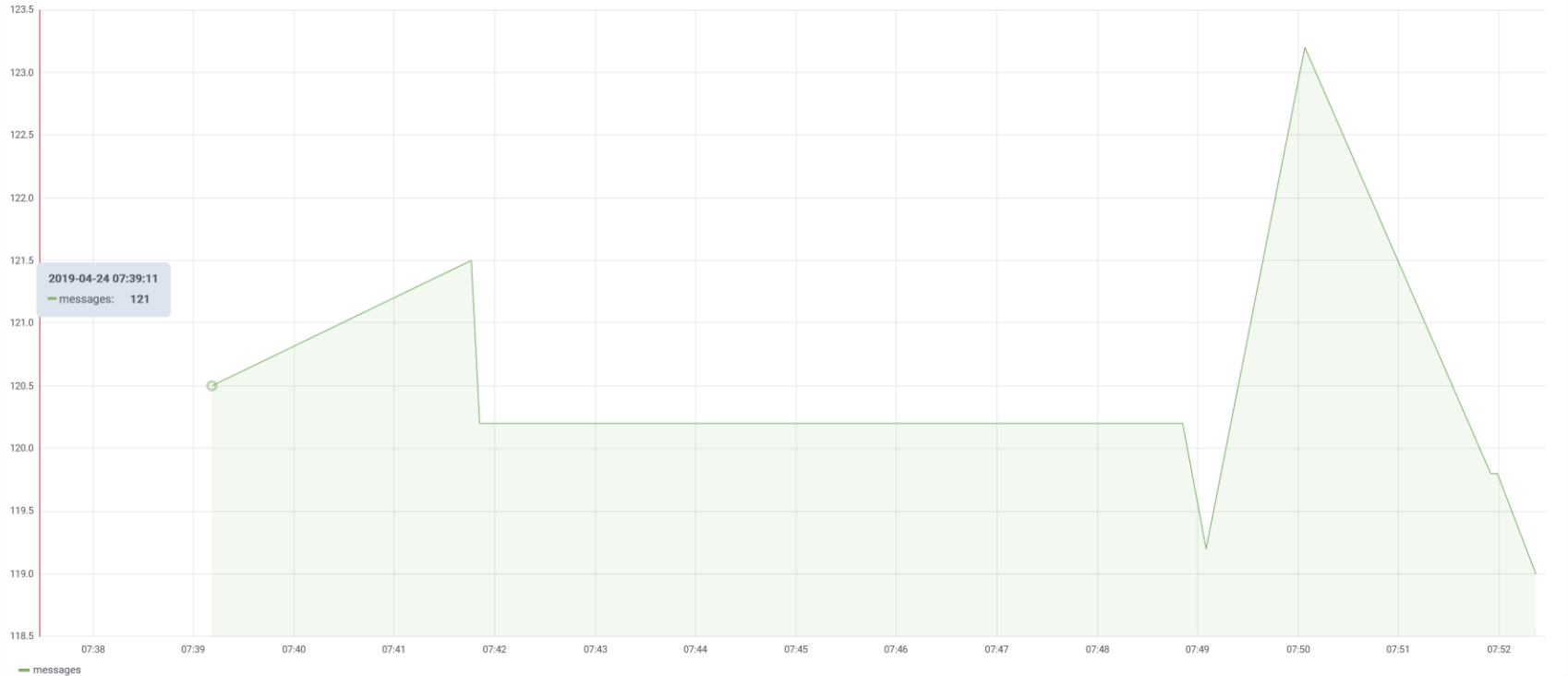
```
curl -X POST https://k8s-dev.mainflux.com/http/channels/7830ec4d-d506-45a0-945d-4a3249d2c417/messages -H "Authorization: 80cca27f-5572-4f98-9102-62b5dd01ce45" -H "Content-Type: application/senml+json" -d "[{"n":"voltage", "u":"V", "v":123.2}]"
```

# MAINFLUX – Send Message to Mainflux on Kubernetes

IoTfuse 2019 ▾

📊 ☆ 🔄 📄 ⚙️ ⏪ 🔍 ⏩ ⌚ Last 15 minutes ↺

IoTfuse 2019



# THANK YOU!

[www.mainflux.com](http://www.mainflux.com)  
[info@mainflux.com](mailto:info@mainflux.com)