Review Open Call 5
Stress-Test asvin.io

Rohit Bohara

asvin.io

5th Fed4FIRE+ Engineering Conference(FEC)

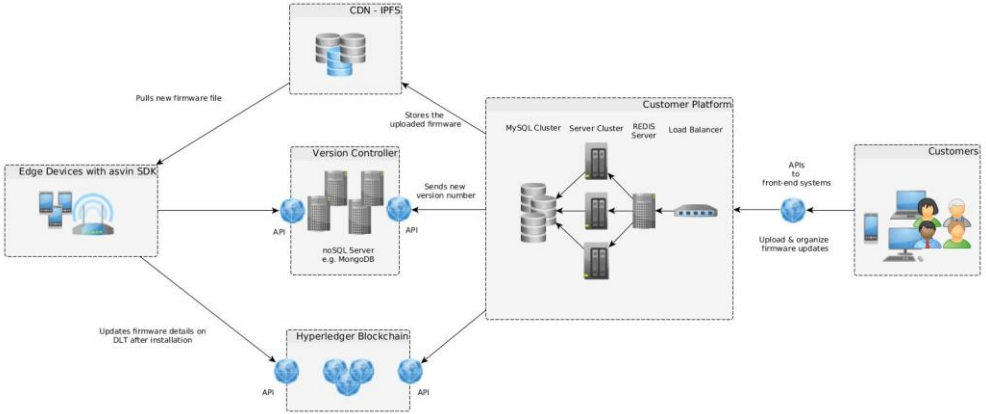
Copenhagen, 24-25th April 2019

asvin.io

HEALING THE INTERNET OF THINGS

Concept

Architecture



Front end

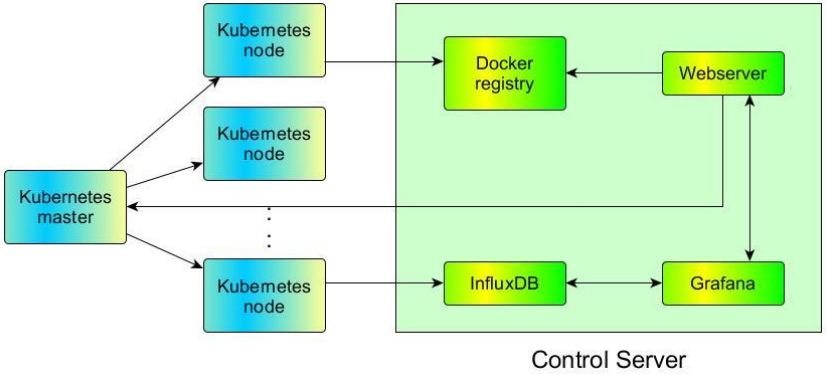
Objectives

- Controlled validation of asvin.io architecture
- Assessment of scalability and resilience
- Monitor effects of network latency and bandwidth
- Gather network logs and induce insights
- Authenticate reliability of the architecture
- Configure server parameters by iterative experiments

Background & Motivation

- Gain practical knowledge of the architecture
- Stress-test asvin.io for a market fit solution
- Proof of scalability
- Verification and validation of the architecture
- Physical experimentation not plausible
- Fed4Fire+ experiments are cost effective
- Large computing experiment are possible on Fed4FIRE+ testbeds

Experiment Setup



FEATURES

- Kubernetes cluster of 100 nodes
- Control server to build and deploy docker images
- Grafana to visualize analytics
- InfluxDB to store time series

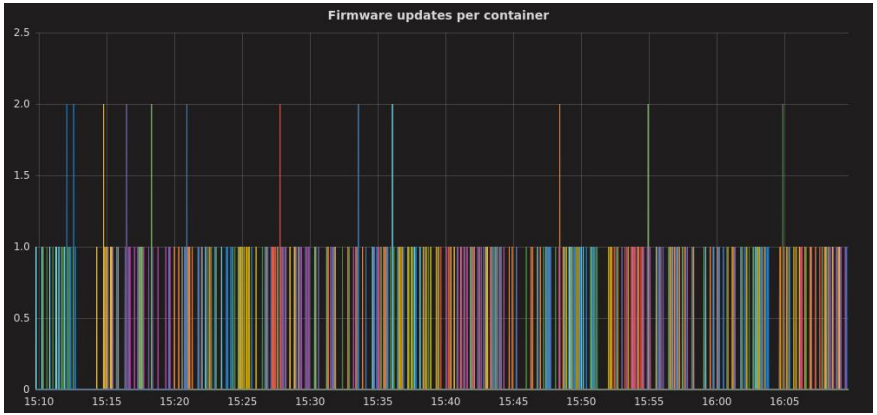
https://lib.ugent.be/catalog/rug01:002494719

Results

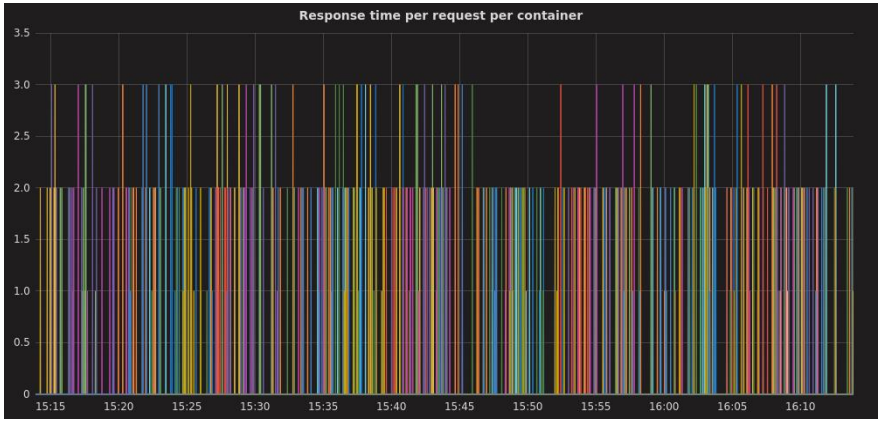
- Response time reduced by optimizing server parameters
- Authentic proof of scalability
  - Effortlessly upscale devices to 10,100, 1000x
- Architecture is robust and reliable
  - Requests from 10,000 devices handled smoothly
- Verification of life cycle of an IoT device on asvin.io
  - Registration ->Firmware updates -> Cancellation
- asvin.io is adaptable to network latency and bandwidth limitation

Analytics

FIRMWARE UPDATE



RESPONSE TIME



Lessons Learned

- asvin.io platform is robust and scalable
- Performance can be improved
- Fed4FIRE+ platform is convenient to use
- Fed4FIRE+ experiments can be tailored for specific needs
- Ample technical support and documentation available
- Fed4FIRE+ is best suitable for stress testing
- Diversified resources are available on Fed4FIRE+ testbeds

Business Impact

- Practical proof gives edge over competitors
- Improved performance
- Cost effective
- Positive response from customers
- Expanded customer base
- Seal of verifiability from EU sponsored project

How did it help?

- Enormous data is generated from experiments
- asvin.io has authentic proof of its claims
- Fed4FIRE+ experiments gave practical knowledge
- Cement confidence in architecture to fulfil growing market needs
- Fed4FIRE+ experiments reinforced trust in our solution
- Acquired new skills, e.g. Kubernetes

Value Perceived

- Increased knowledge about the architecture
- Practical experience
- Proof of scalability and resilience
- Acquired new competence
- Confidence to run experiments on Fed4FIRE+ in future
- Edge over competitors

Why Fed4FIRE+?

- Open, reliable and highly accessible
- Credibility of European Union
- Diversity of available resources
- Simple, efficient and cost effective experimental process
- Excellent technical support and documentation
- Combining infrastructures
- Cost effective

Resources and Tools

JFED

USAGE

- Provision and manage experiment on testbeds
- RSpec
  - Network and resource configuration
- ESpec
  - Bootstrap an experiment
- Testbed
  - Virtual Wall 1

Resources and Tools

GENERATE ESPEC

```
rohit$ python generate-espec/main.py --help
usage: main.py [-h] [--nodes [NODES]] [--no-control-server] [--gateway]
               [--wall {wall1,wall2}]

Generate espec for kubernetes

optional arguments:
  -h, --help            show this help message and exit
  --nodes [NODES]       amount of nodes in the generated espec, not including
                        the master node
  --no-control-server   Do not include the code to provision and setup a
                        control server with influx, grafana, private docker
                        registry and control website
  --gateway             add a gateway + apache server for delay testing
  --wall {wall1,wall2}  Target Virtual Wall, defaults to wall2
rohit$ python generate-espec/main.py --wall wall1 --nodes 100
```

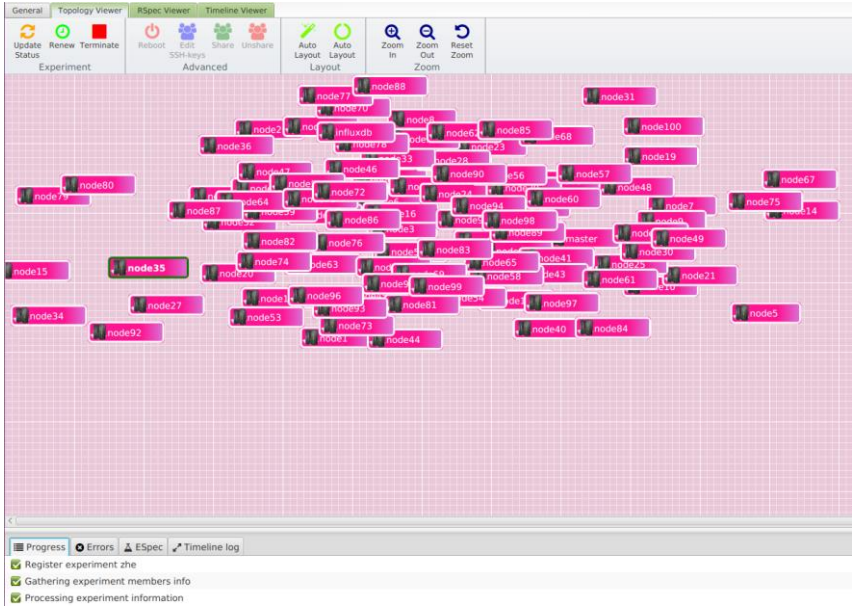
USAGE

- Tool written in python
- Generate ESpec for Virtual Wall1 and Wall2
- Easy to create and deploy Kubernetes cluster on testbeds

https://github.ugent.be/jlemaes/generate-espec

Resources and Tools

KUBERNETES

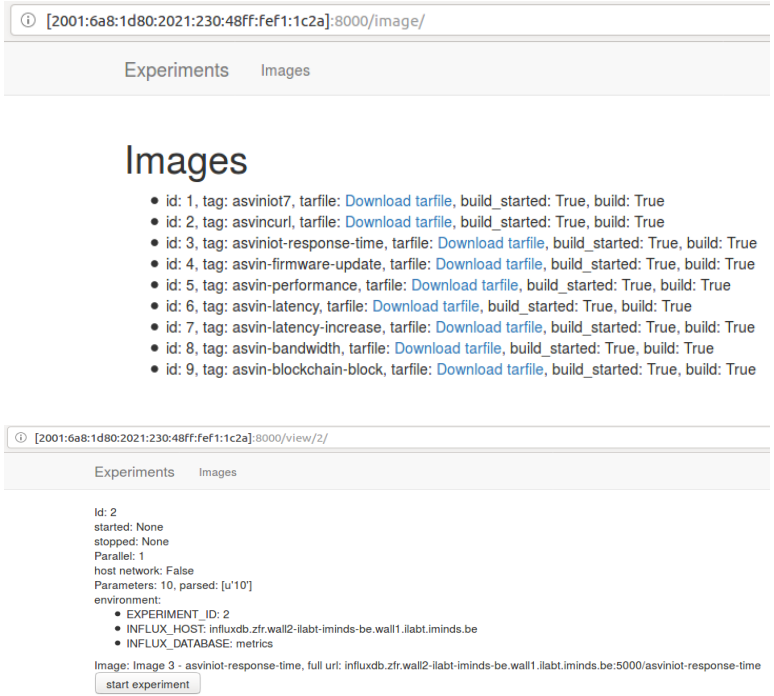


USAGE

- Container orchestration system
- Used to deploy, scale and manage container applications

Experiment Setup

CONTROL SERVER



USAGE

- Build and deploy docker images.
- Start and control an experiment on cluster
- Scale containers on the cluster
- Utilize InfluxDB and Grafana for visualization

- User friendly interface of jFed experimenter
- Around the clock technical support
- Abundant nodes on testbeds, Wall1 206 and Wall2 159
- High speed internet connectivity on testbeds
- Network impairment e.g. delay, packet loss and bandwidth limitation possible
- Multiple OS are supported

Added Values


Co-funded by the European Union


Co-funded by the Swiss Confederation

This project has received funding from the European Union's Horizon 2020 research and innovation programme, which is co-funded by the European Commission and the Swiss State Secretariat for Education, Research and Innovation, under grant agreement No 732638.

THANK YOU FOR YOUR ATTENTION

WWW.FED4FIRE.EU