오픈소스 Jitsi Meet와 함께 하는 오토스케일이 가능한 화상 회의 인프라 여정

Journey to the Auto-scalable Video Conferencing Infrastructure with Open Source - Jitsi Meet

Kanghee Kim
Ph.D, Electronics - Inha University

Jitsi

Jitsi

Jitsi is a collection of Open Source projects which provide state-of-the-art video conferencing capabilities that are secure, easy to use and easy to self-host.

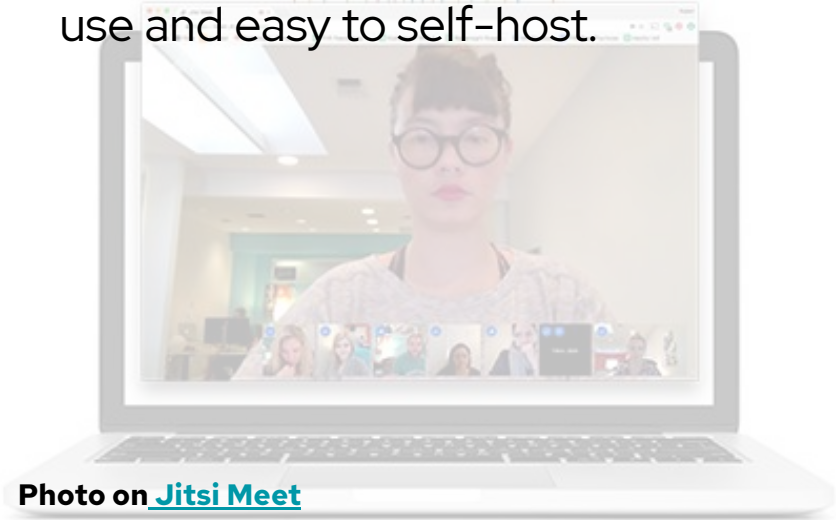
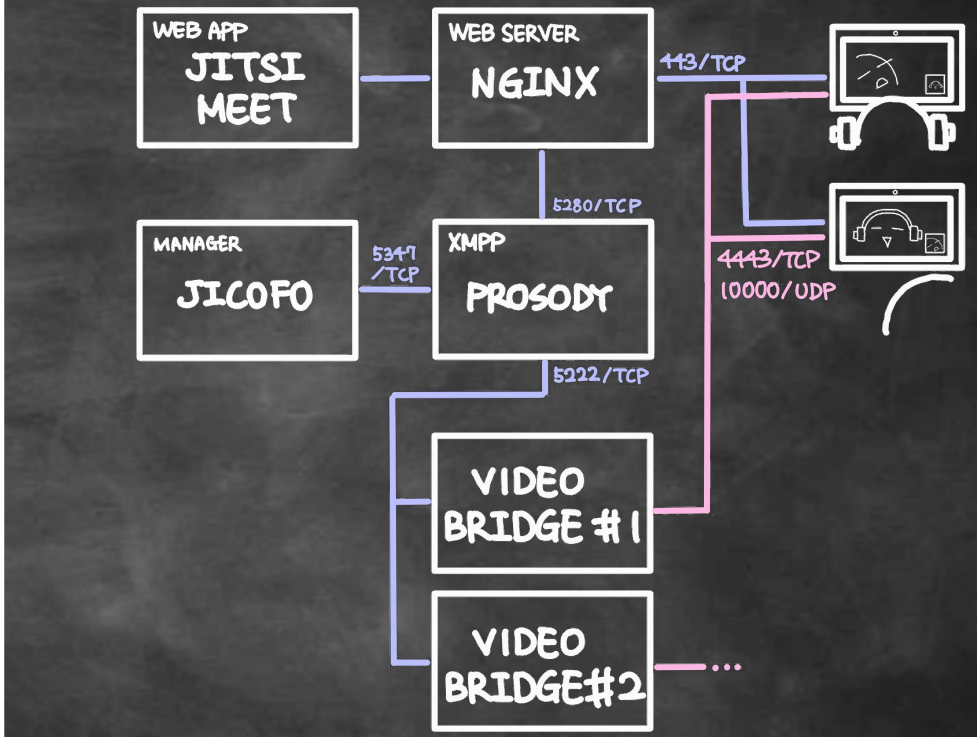


Photo on [Jitsi Meet]

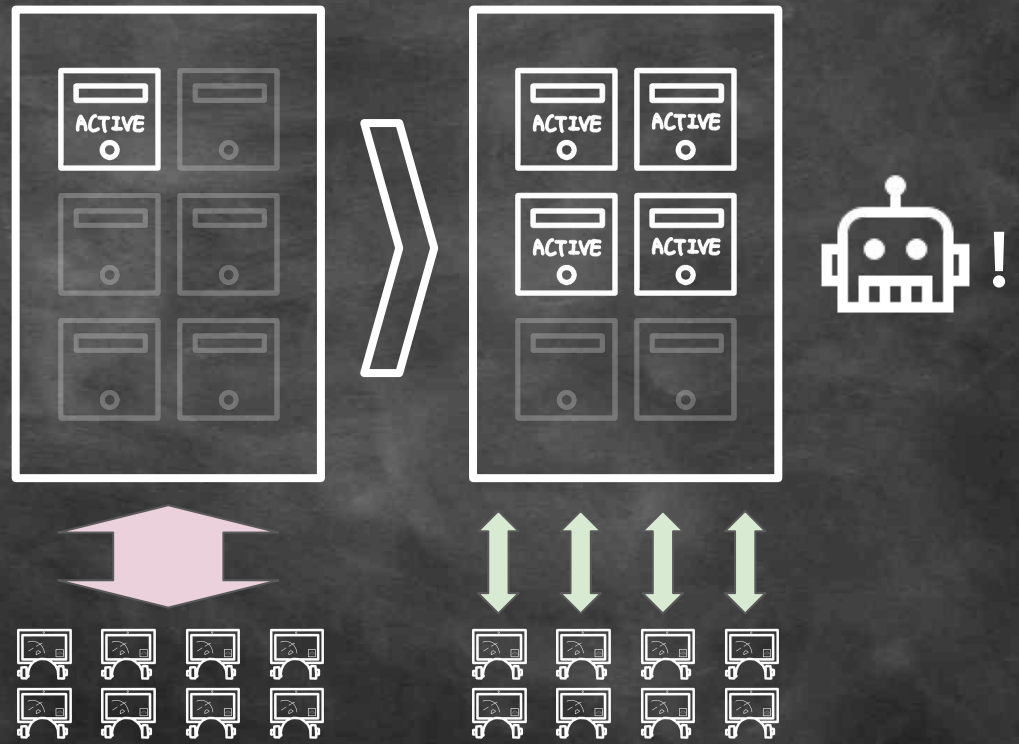


[JVB]: Jitsi Video Bridge

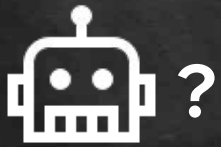
[Prosody IM: Welcome]

Auto scalable

Auto scale-out



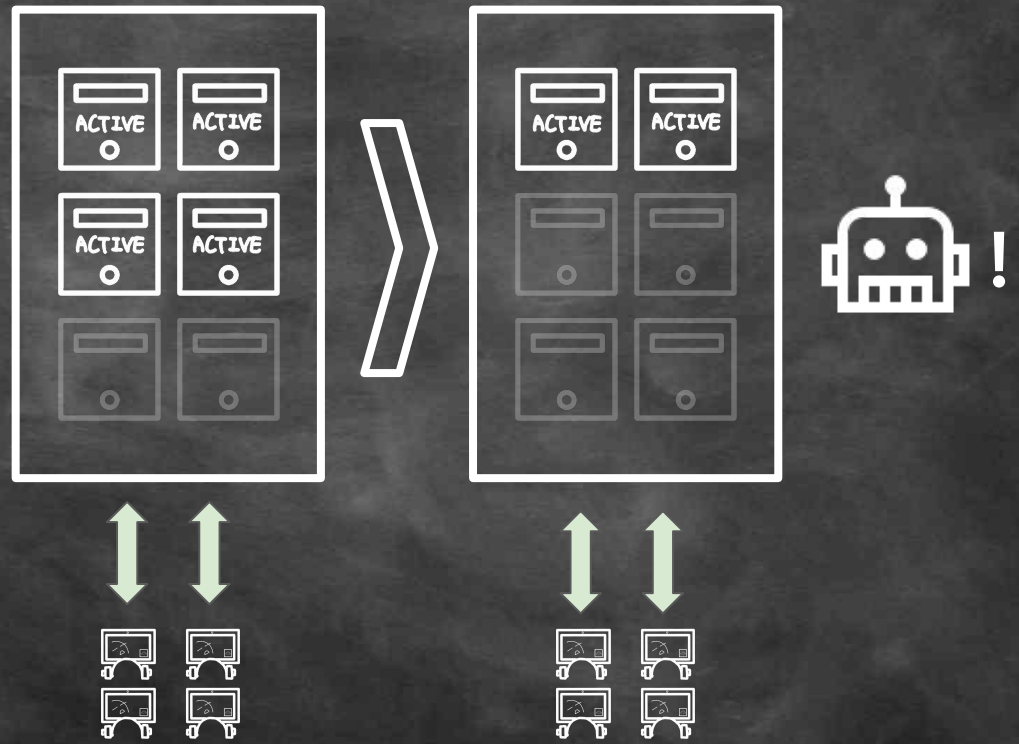
I AM NOT HUMAN



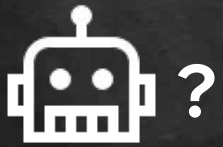
?

Bot icon by Icons8

Auto scale-in



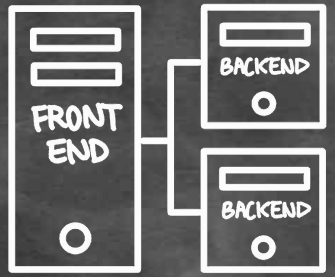
I AM NOT HUMAN



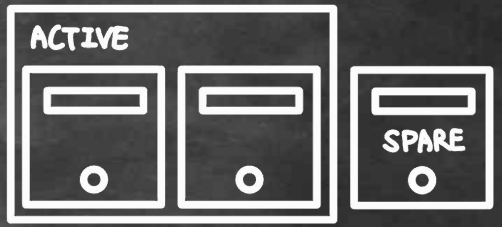
Bot icon by Icons8

Benefits

SAVING COST



LOAD BALANCE



HIGH AVAILABILITY

TL;DR

TL;DR

Requirements for flexible resource management **IF** you need to cope with temporary large-scale access. → **AUTO SCALABLE**

Highly recommended to install jitsi on **virtual machines**.

(You cannot install it on windows. 😀)

Much less consumption of infrastructure resources except **NETWORK**.

If you want to experience auto-scalable jitsi right now, please refer to the link below.

https://github.com/hpi-schul-cloud/jitsi-deployment

When Will “COVID-19” End?

The Era of COVID-19

Untact

- Neologism in Korea only, I know...
- Equal to social distancing
- “Un” + “~~Con~~tact”
- How can we have a “untact” meeting?

→ **VIDEO CONFERENCE**

Photo by Jon Tyson on Unsplash

Infrastructure for Video Conference at Home!

Photo by Chris Montgomery on Unsplash

WHY do we need auto-scalable jitsi

WHY do we need auto-scalable jitsi

With jitsi, you want:

- **PRIVATE** video conference infra
- Jitsi is the **FREE**
  - Open Source Software (Apache 2.0)

With jitsi, you do not want:

- Auto-scalable 🤔?

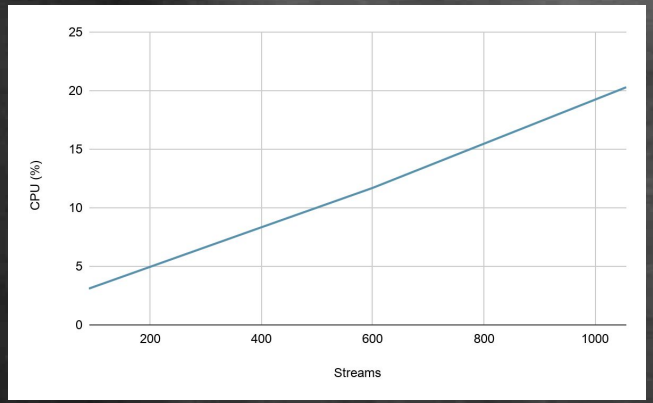
1000 video streams
550 Megabits
20% CPU

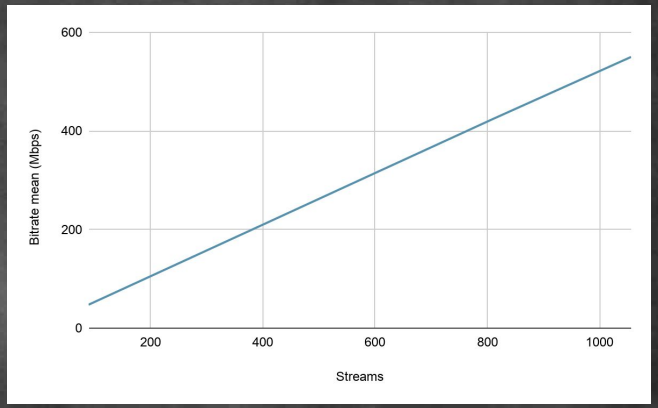
HEEEY! THAT'S PRETTY GOOD. 👍

https://jitsi.org/jitsi-videobridge-performance-evaluation/

WHY do we need auto-scalable jitsi

Streams from 90 to 1056





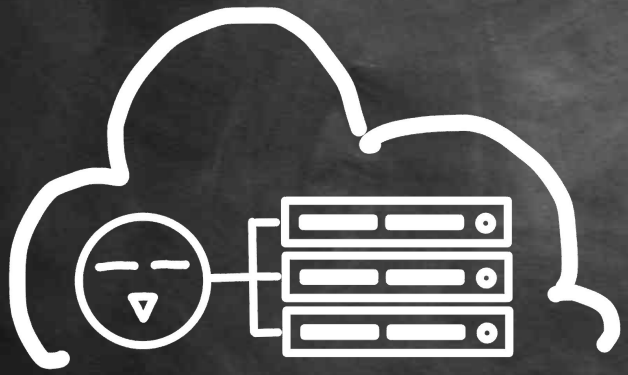
CPU 6.55 times ↑

NETWORK 11.56 times ↑

https://jitsi.org/jitsi-videobridge-performance-evaluation/

HOW can we provide auto-scalable jitsi

Virtual machines



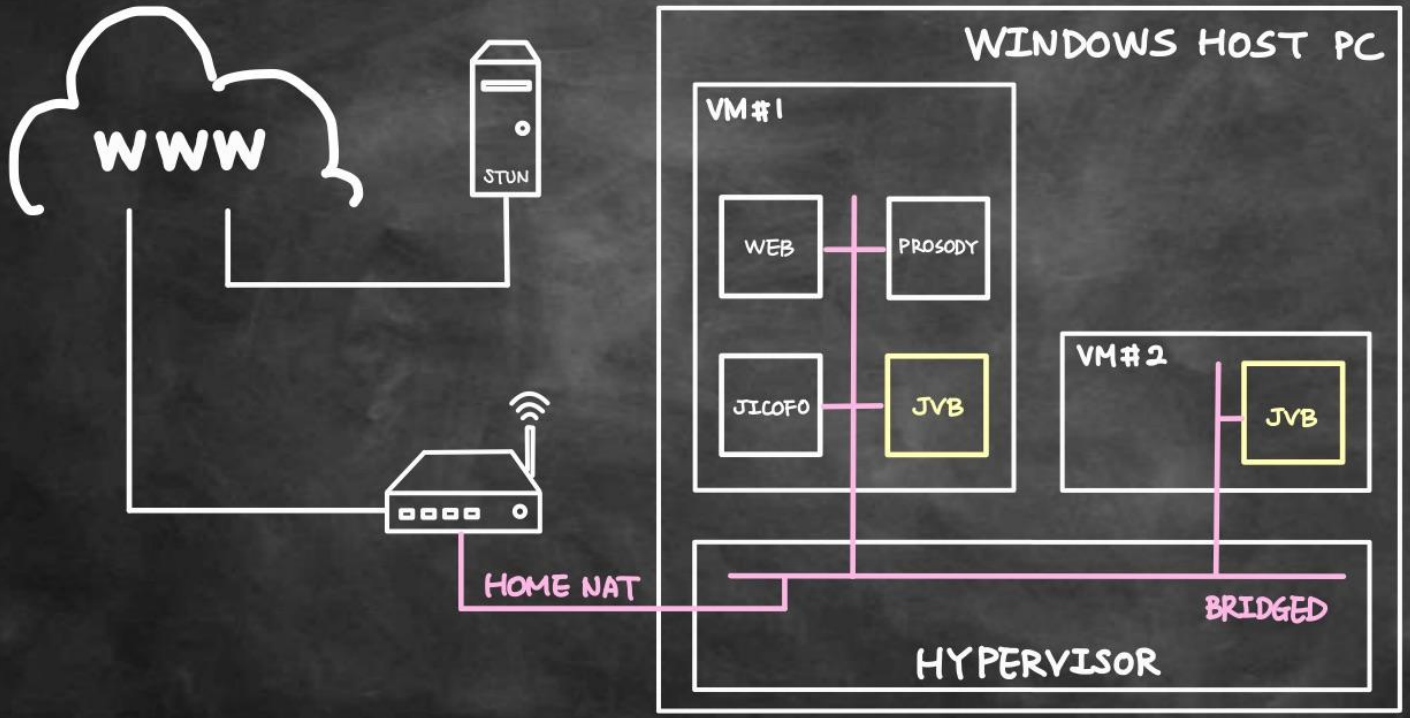
EASY INSTALLATION 😅

Ubuntu OS = apt packages

Horizontal scale-out = Increase # of VMs

https://jitsi.github.io/handbook/docs/devops-guide/devops-guide-quickstart

VM Only



https://jitsi.github.io/handbook/docs/devops-guide/devops-guide-quickstart

Docker

EASY INSTALLATION (docker-compose)

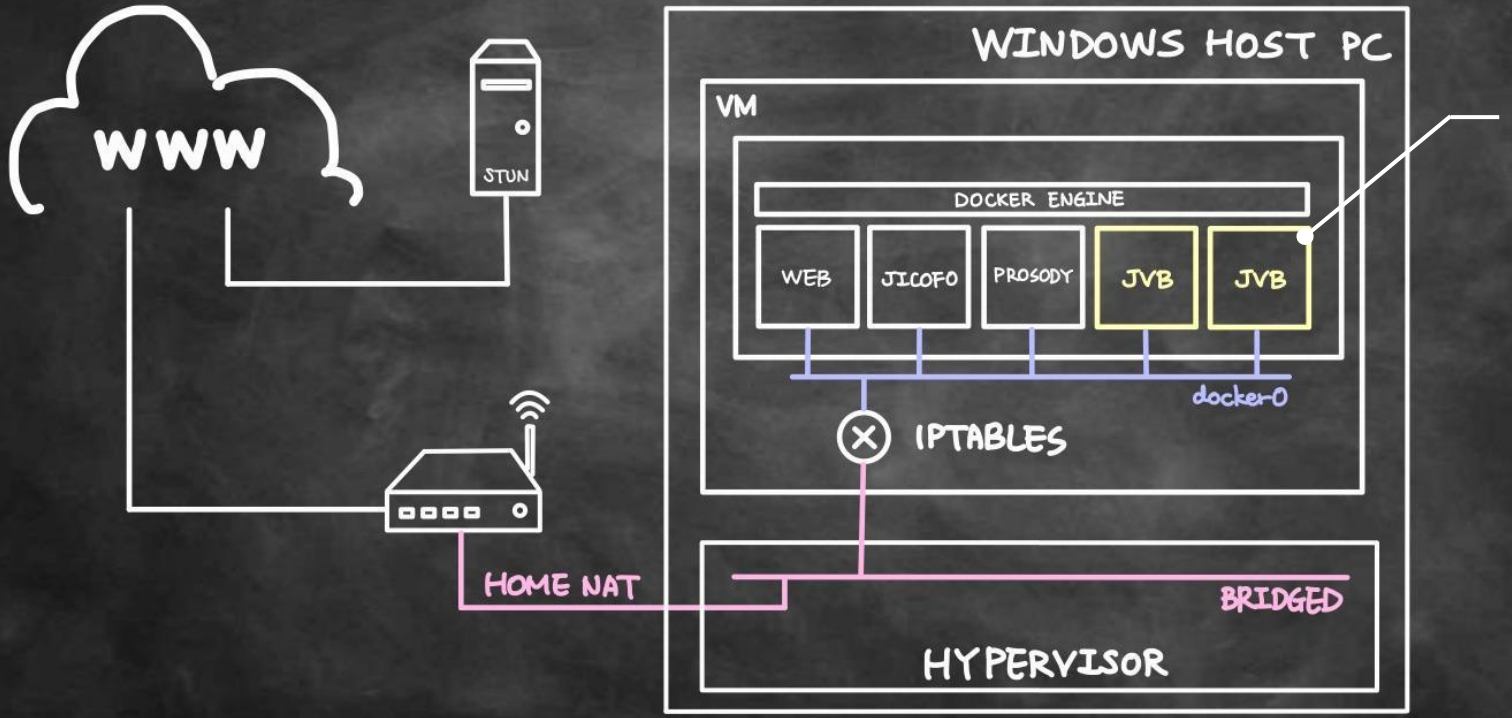
→ if you are familiar with docker 🙂

No OS dependency

→ if docker is installed

https://jitsi.github.io/handbook/docs/devops-guide/devops-guide-docker

Docker



JVBs on same machine? 🤔 Sorry, I am not familiar with docker overlay network...

https://jitsi.github.io/handbook/docs/devops-guide/devops-guide-docker

OpenShift 3.11

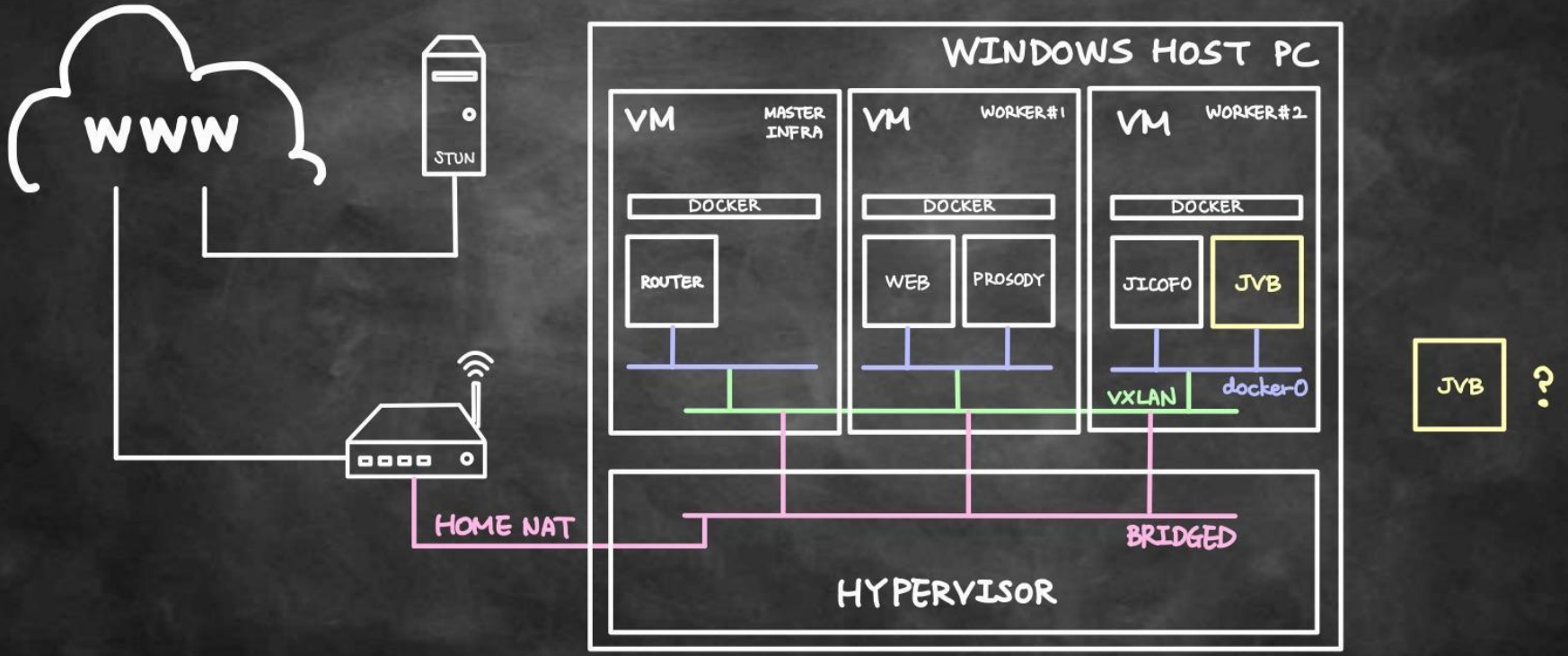
> Kubernetes

Required relatively high resources 😟

Flexible management for VIRTUALIZED RESOURCES (= CONTAINER)

Auto-scalable! 😀

OpenShift 3.11

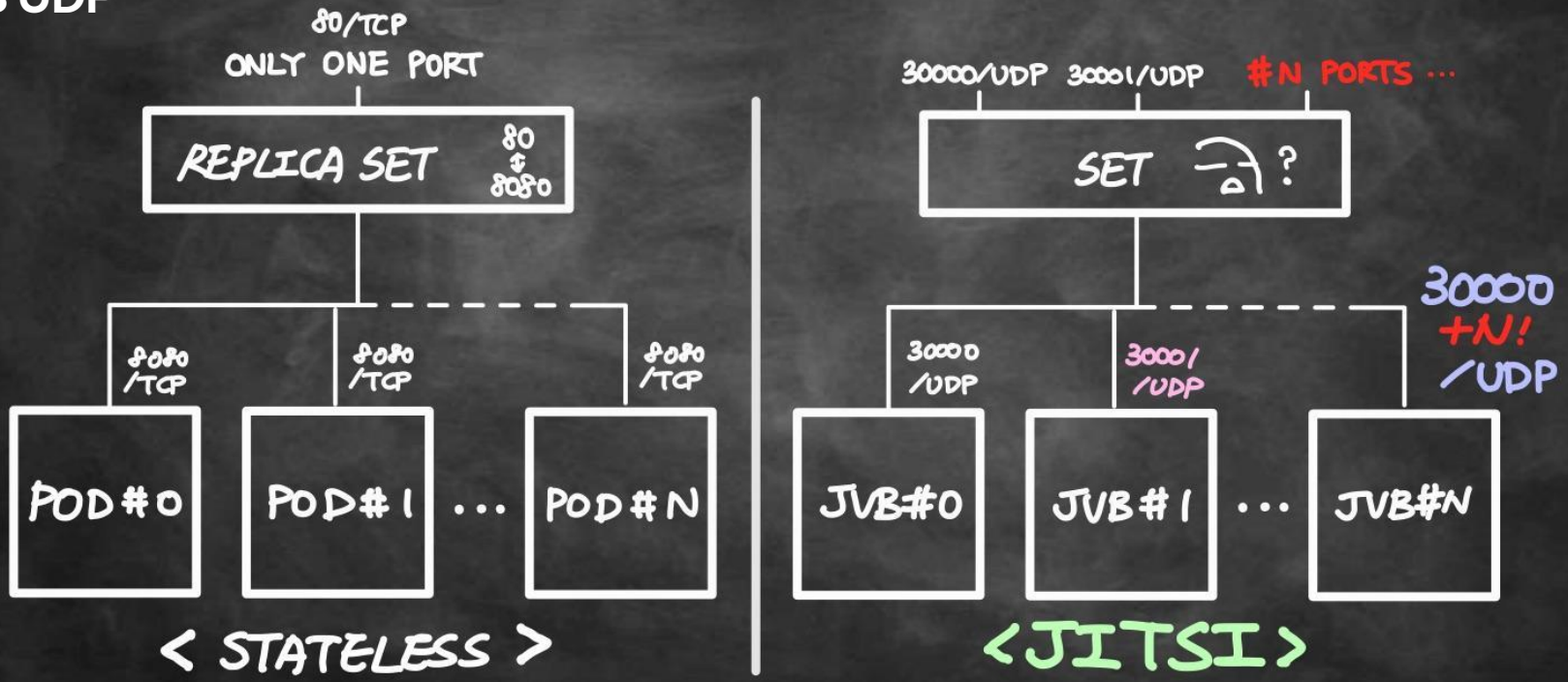


Challenges (1/4)

Let’s encrypt (HTTPS)

Challenges (2/4)

JVB UDP



Challenges (3/4)

Scalable StatefulSet and Service

- How to generate different services for each pod
- Do I have to **MANUALLY** create them? → **NO** 😀
- Thanks to the contributors of the **METACONTROLLER**.
- Service-Per-Pod Decorator

hooks/sync-service-per-pod.jsonnet

```
...
type: "NodePort",
        ports: [
          {
            local basePort = 30000,
            port: basePort + index,
            protocol: "UDP",
            targetPort: basePort + index,
            nodePort: basePort + index,
            name: "jvb"
          }
...
for index in std.range(0,
statefulset.spec.replicas - 1)
...
```

Challenges (4/4)

Custom metric

- Scale pods based on custom application metrics, in addition to the default CPU and memory usage metrics.
- Thanks to the contributors of the **PROMETHEUS ADAPTER** and **JITSI MEET EXPORTER**.
- Prometheus Adapter for Kubernetes Metrics APIs
- Jitsi Meet Metrics Exporter

```
apiVersion: v1
kind: ConfigMap
...
data:
  config.yaml: |
    rules:
      - seriesQuery:
'{__name__="jitsi_total_data_channel_messages_received",namespace!="",pod!=""}'
        resources:
              ...
        metricsQuery:
'sum(rate(<<.Series>>{<<.LabelMatchers>>}[1m])) by (<<.GroupBy>>)'
```

DEMO

References

jitsi github k8s Link

auto scalable jitsi k8s Link

custom metric on OpenShift 3.11 Link

podium/jitsi Link

metacontroller Link

jitsi prometheus exporter Link

docker for jitsi torture Link