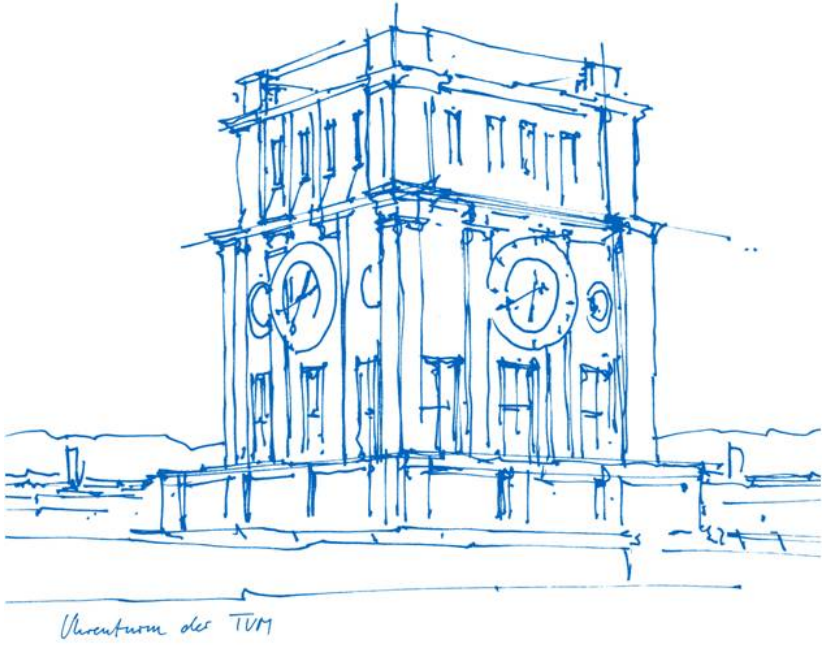
FaaS and Prediction Models

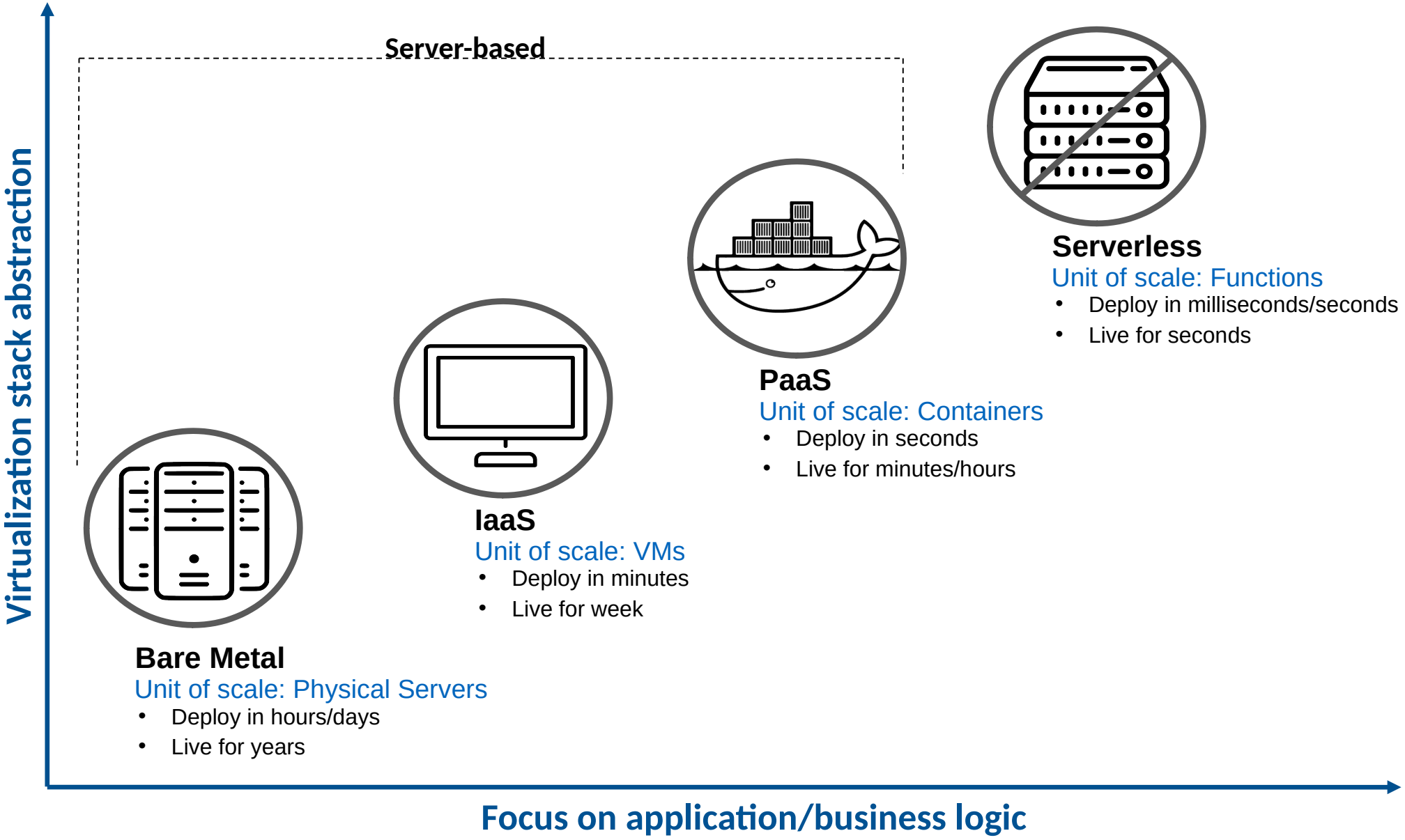
Mohak Chadha



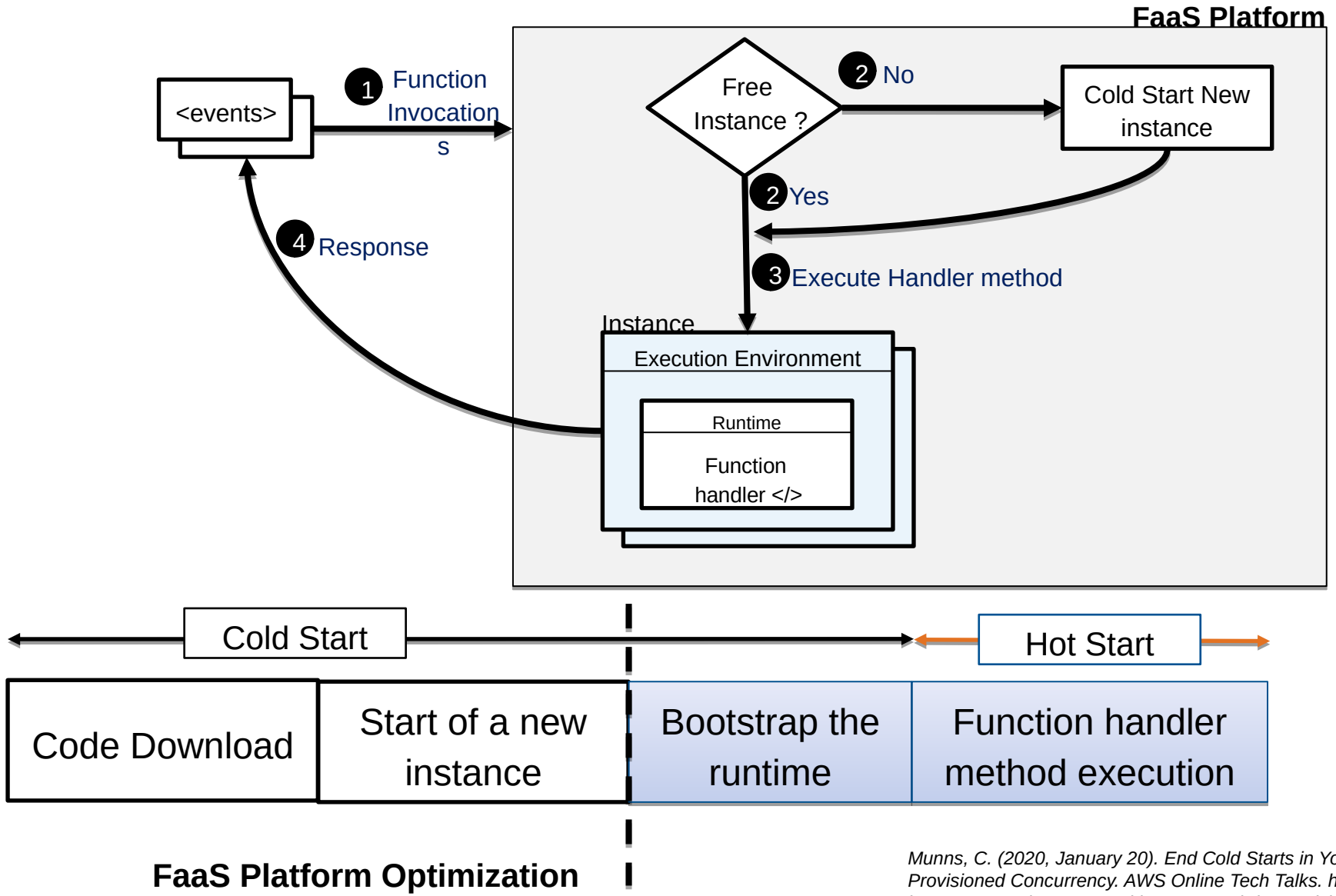
Outline

- Serverless Computing
- How FaaS Works?
- Kubernetes
- Docker
- OpenWhisk

Introduction: Serverless Computing



Introduction: How Function-as-a-Service works?



Munns, C. (2020, January 20). End Cold Starts in Your Serverless Apps with AWS Lambda Provisioned Concurrency. AWS Online Tech Talks. https://pages.awscloud.com/End-Cold-Starts-in-Your-Serverless-Apps-with-AWS-Lambda-Provisioned-Concurrency_2020_0101-SRV_OD.html.

FaaS

- Advantages
  - No provisioning of servers.
  - Automatic Scaling.
  - Reduction of costs.
  - Underlying servers shared among different function invocations.

- Disadvantages
  - Focused on stateless functions.
  - Performance variations due to restart latencies.
  - Not suitable for heavy compute-intensive workloads, own VMs might be cheaper.
  - Limited security: shared VMs, no control over the network.

Kubernetes (k8s)

- As applications grow to span multiple containers deployed across multiple servers, operating them becomes more complex.
- Kubernetes provides an open source API that controls how and where those containers will run.
- De facto platform for deploying cloud-native applications.
- Key features:
  - Automated deployment and replication of containers.
  - Online scale-in or scale-out of container clusters.
  - Load balancing over groups of containers.
  - Rolling upgrades of application containers.
  - Resilience, with automated rescheduling of failed containers.
  - Controlled exposure of network ports to systems outside of the cluster.

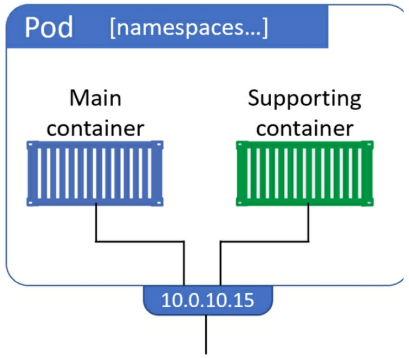
Declarative Management

- Applications are managed declaratively
  - Describe how you want the application to run in YAML files.
  - POST the descriptions to Kubernetes.
  - Follow how Kubernetes mananges the application to match the description.
- Advantages
  - Implementation entirely on K8s.
  - K8s oversees the application during runtime to always match the requirements.

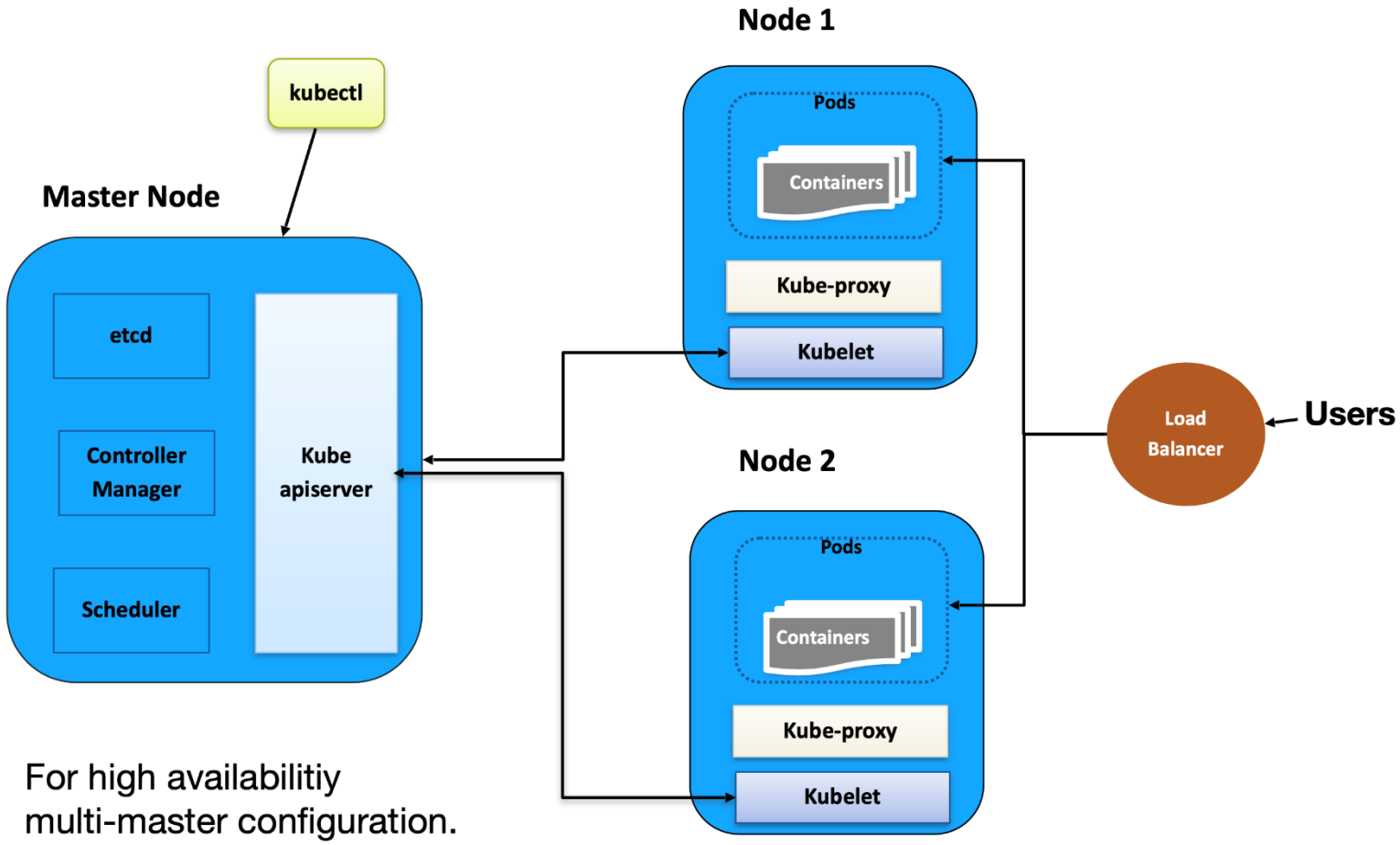
```yaml
spec:
  runtimeClassName: kata-fc
  containers:
    - image: docker.io/kkyfury/java-atax:v1
      resources:
        requests:
          memory: "512Mi"
          cpu: "1000m"
        limits:
          memory: "512Mi"
          cpu: "1000m"
```

Pods

- A pod is a group of containers.
  - A common example of multi-container pods are service meshes e.g. ISTIO.
- Pod provides the enviornment for containers.
  - Unique IP addresses, shared memory, volumes, network stack
- Pods create their own network namespace.
  - Single IP address, a single range of TCP and UDP ports, a routing table
  - Shared among all containers in a pod.
- Resource limits of pods, i.e., CPU, RAM controlled using cgroups.
- Pods are units of scheduling.
- Pods are an atomic unit.



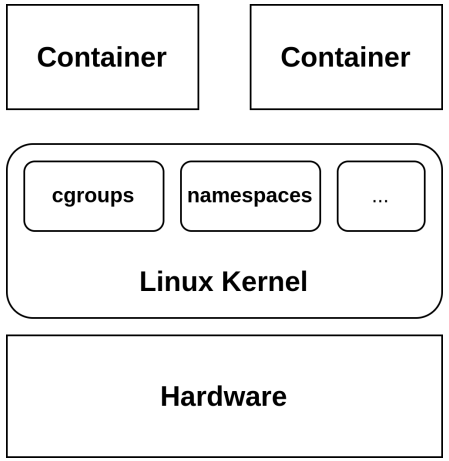
Kubernetes Architecture



Containers vs VMs

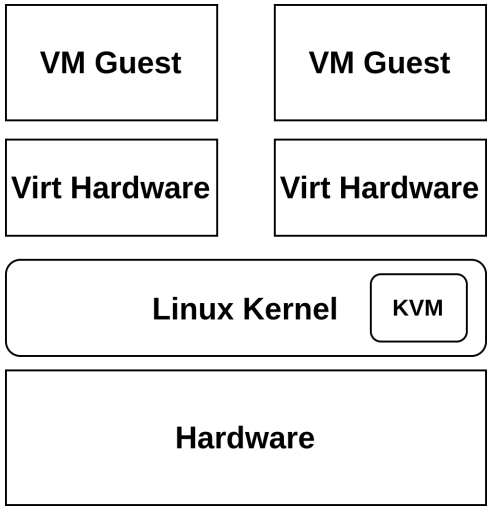
Containers

- Using Linux primitives.
- Share Linux Kernel.
- Fast Starts, minimal overheads.
- Flexible Isolation.



Virtual Machines

- Virtualisation or emulate hardware components.
- Completely separate kernels.
- Slower starts, must boot kernel and set-up hardware.



Docker

- Docker provides unified access to
  - Linux container technology (cgroups, namespaces)
  - Various container implementations (lxc, libvirt, libcontainer, etc.)
  - ‘libcontainer’ is Docker’s implementation of container technology
- Client-server architecture.
- Provides functionality to building, running, and distributing OCI-compliant container images.
- Main advantages: Isolation and portability

OpenWhisk

- Open-source FaaS platform.
- Users implement actions that are executed on HTTP requests.
- Actions are executed in docker containers.
- Support for implementing actions in NodeJs, Python, Go, Java, Ruby, Swift, Go, and PhP.
- Deployed on top of Kubernetes.
- Used in IBM Cloud Functions.

Writing OpenWhisk Actions: Python

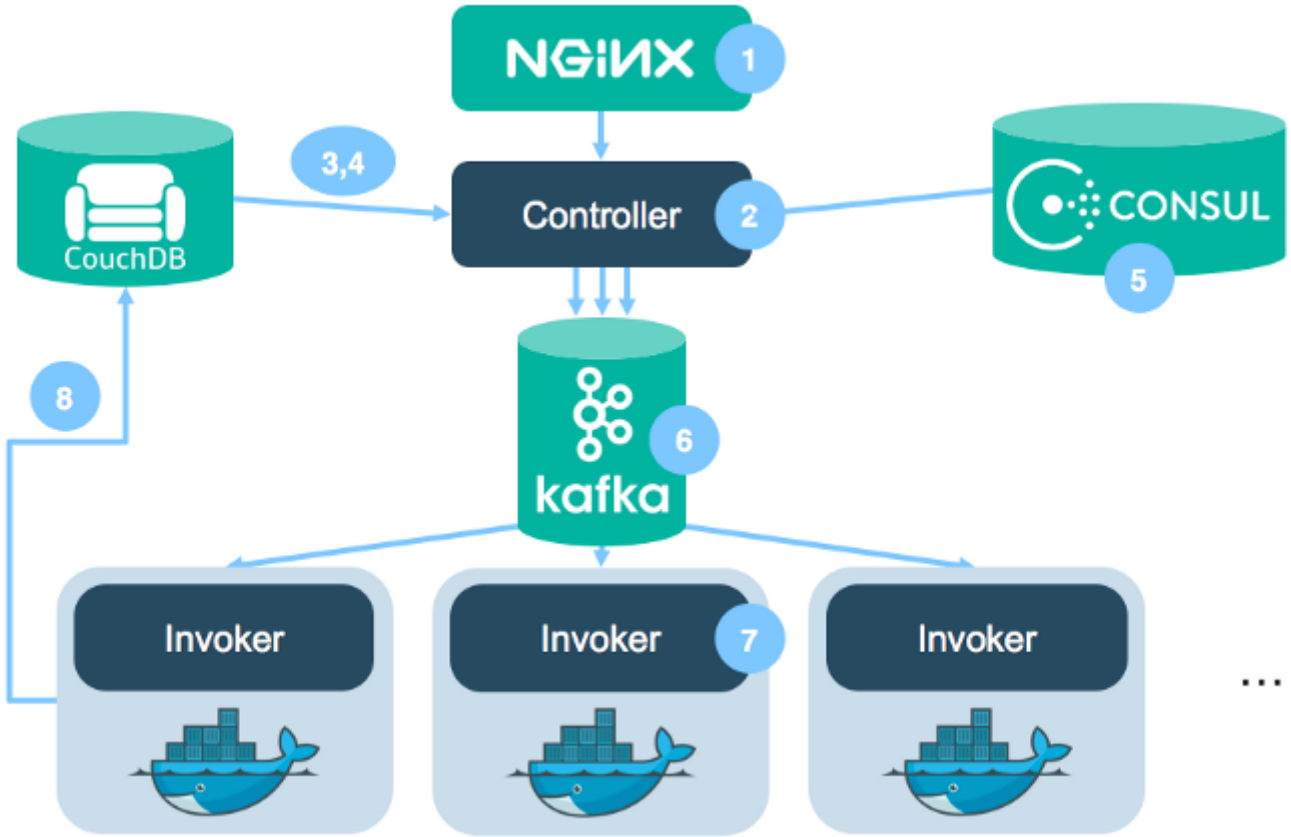
```python
import os
import subprocess
import requests
import json

def main(request):

    name = request["name"]
    greeting = "Hello " + name + "!"
    return {"result": greeting}

```

OpenWhisk Archictecture



Demo