

# Cloud Computing Exercises

## WiSe 20/21

### Introduction to Exercises

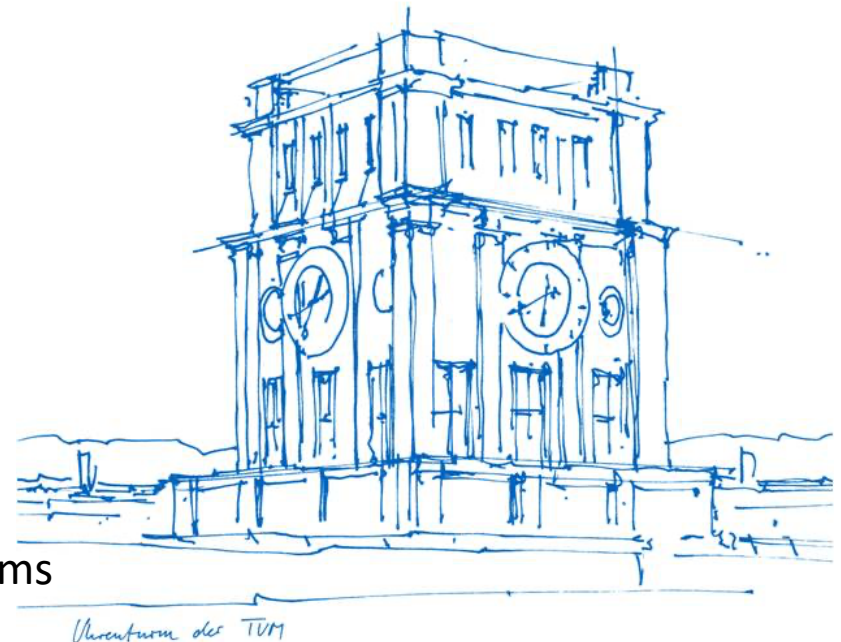
3<sup>rd</sup> November 2020

**Anshul Jindal** (M.Sc. Informatics)

[anshul.jindal@tum.de](mailto:anshul.jindal@tum.de)

Chair of Computer Architecture and Parallel Systems

*Technical University of Munich (TUM), Germany*



# Introduction about me



## Education and Work Ex:

- Bachelors in Computer Science and Engineering from NIT Hamirpur, India (2014)
- 2014 - 2016, Senior Software Engineer at Samsung Semiconductor, Bangalore, India
- 2016 - 2018, Master of Science in Informatics at TUM
- 2018 - Ongoing, PhD. at TUM

## Research Interests:

- Performance modeling of microservices.
- Anomaly detection for multivariate timeseries data (Cloud monitoring data).
- Functions scheduling on heterogeneous FaaS platforms

## Contact:

- Query or appointment via email to : [anshul.jindal@tum.de](mailto:anshul.jindal@tum.de)

# Exercises Schedule (Tentative)

Date	Exercise	On Cloud or personal laptop	Group or Individual
10 <sup>th</sup> November 2020	App. Development using Node.js	Cloud (LRZ/AWS/GCP)	Individual /Group
24 <sup>th</sup> November 2020	Cloud Access and Docker	Cloud (LRZ/AWS/GCP)	Individual /Group
8 <sup>th</sup> December 2020	Microservices architecture application	Cloud (LRZ/AWS/GCP)	Individual /Group
22 <sup>nd</sup> December 2020	OpenStack Exercise	Cloud (LRZ/AWS/GCP)	Individual /Group
12 <sup>th</sup> January 2020	App deployment using Kubernetes	Cloud (LRZ/AWS/GCP)	Individual /Group
26 <sup>th</sup> January 2020	Exercise on FaaS	Cloud (LRZ/AWS/GCP)	Individual /Group

# Exercises Bonus



- Maximum of 0.3 bonus will be provided.
- There are total of 6 exercise and each exercise contributes to 1 point.
- So total of 6 points can be achieved, however 5 points are enough to earn the 0.3 bonus.

You need to complete at least 5 exercises successfully to get a full bonus of 0.3

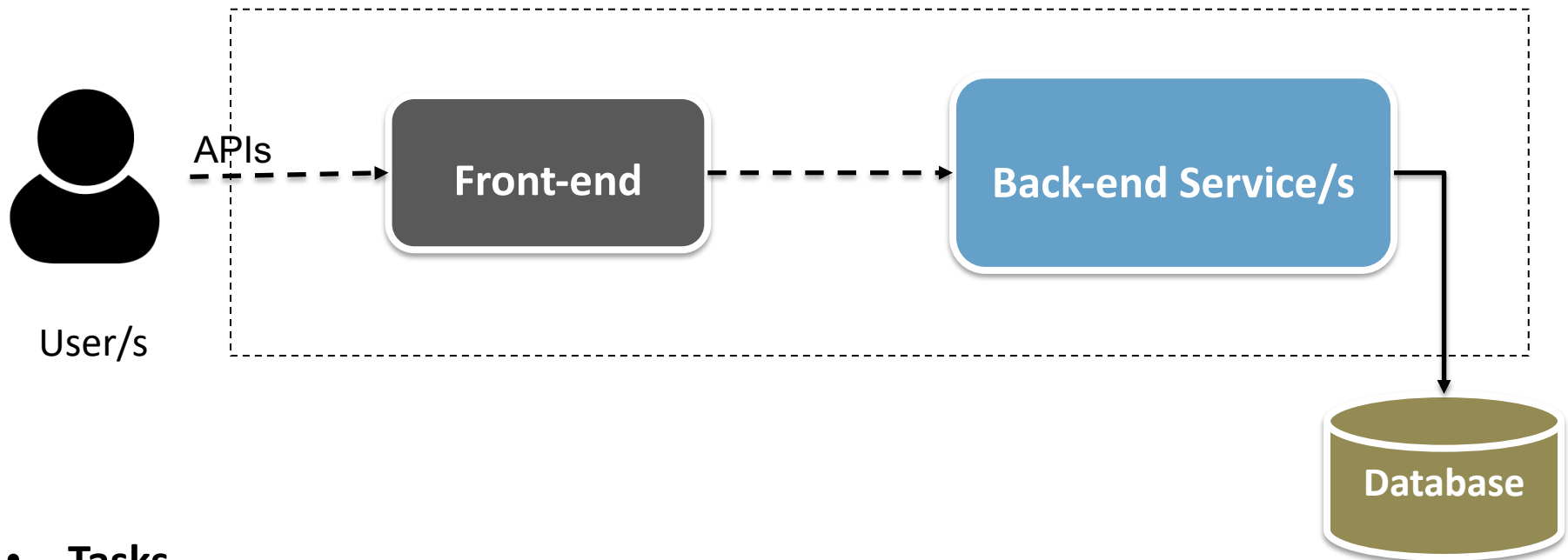
- There are no points for submission of a partial solution to an exercise.

Bonus does not apply to the grades 1.0, 4.3, 4.7 and 5.0.

# Introduction to Exercises

# Ex.1: Application Development using Node.js

- Introduction to Node.js.
- Introduction to REST (**RE**presentational **State Transfer** ) API.
- Introduction to MongoDB.
- Installation of Node.js and given code explanation.

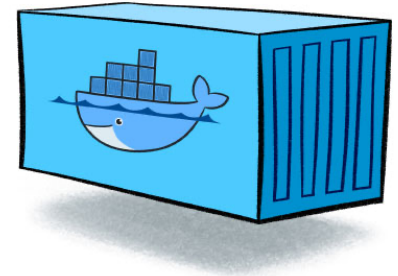


- **Tasks**

- Study the given code structure, about Node.js and queries to MongoDB.
- Write and complete the REST based APIs.

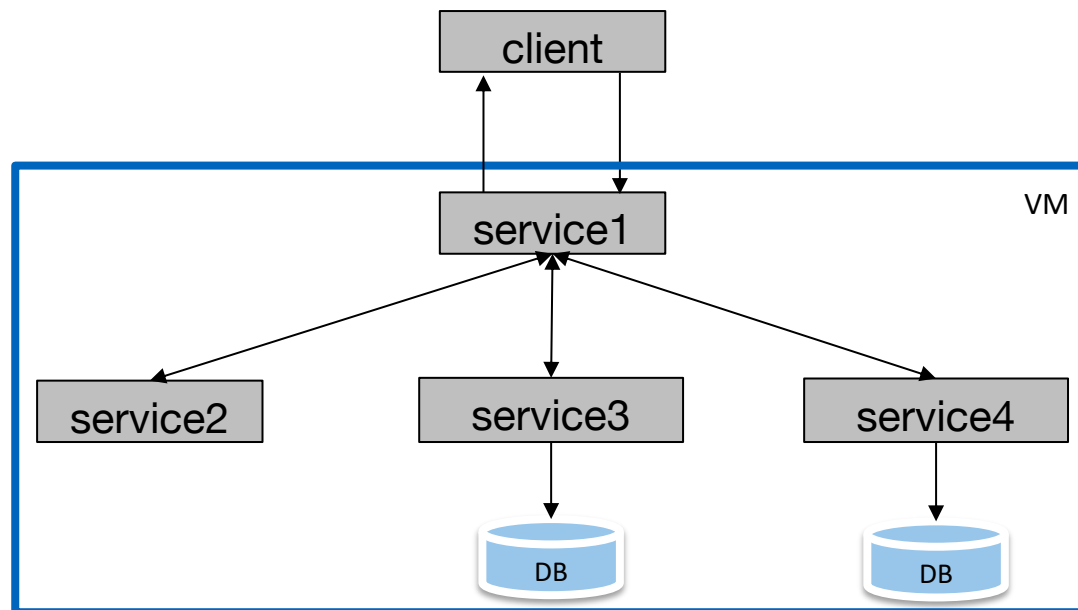
# Ex.2: Cloud Access & App. Deployment using Docker

- Introduction to creating/starting/stopping of VMs on the Cloud.
  - Introduction to application containerization using Docker.
  - Writing a Dockerfile.
  - Creating Docker images and containers.
  - Introduction to Docker registry : Docker Hub
  - Introduction to docker-compose for running multi-containers application.
- 
- **Tasks**
    - Writing Dockerfile for creating Docker images for the application in Ex.1
    - Creating Docker image of the application.
    - Deployment of the docker-based application on the VM.



## Ex.3: Building Microservices Application

- Introduction to Microservices application architecture.
- Introduction to Microservices architecture terms (API Gateway, Service registry, Service Discovery etc.).
- Explanation of the given application application structure and code.

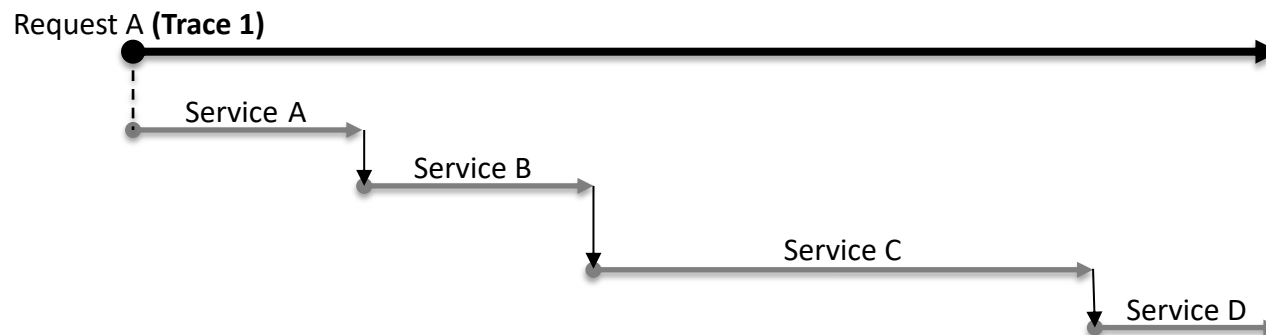


- **Tasks**
  - Writing and completion of microservice's code.
  - Deployment of the application using docker-compose on the VM.



## Ex.4: OpenStack Exercise

- Introduction to OpenStack and its components.
- Creation of a server/VM inside OpenStack using CLI (Command Line Interface).
- Attaching volume, network etc. to the created server/VM.
- Introduction to OpenStack distributed tracing library : OSProfiler.
- Understanding of an OpenStack command (like creation of a VM) trace.

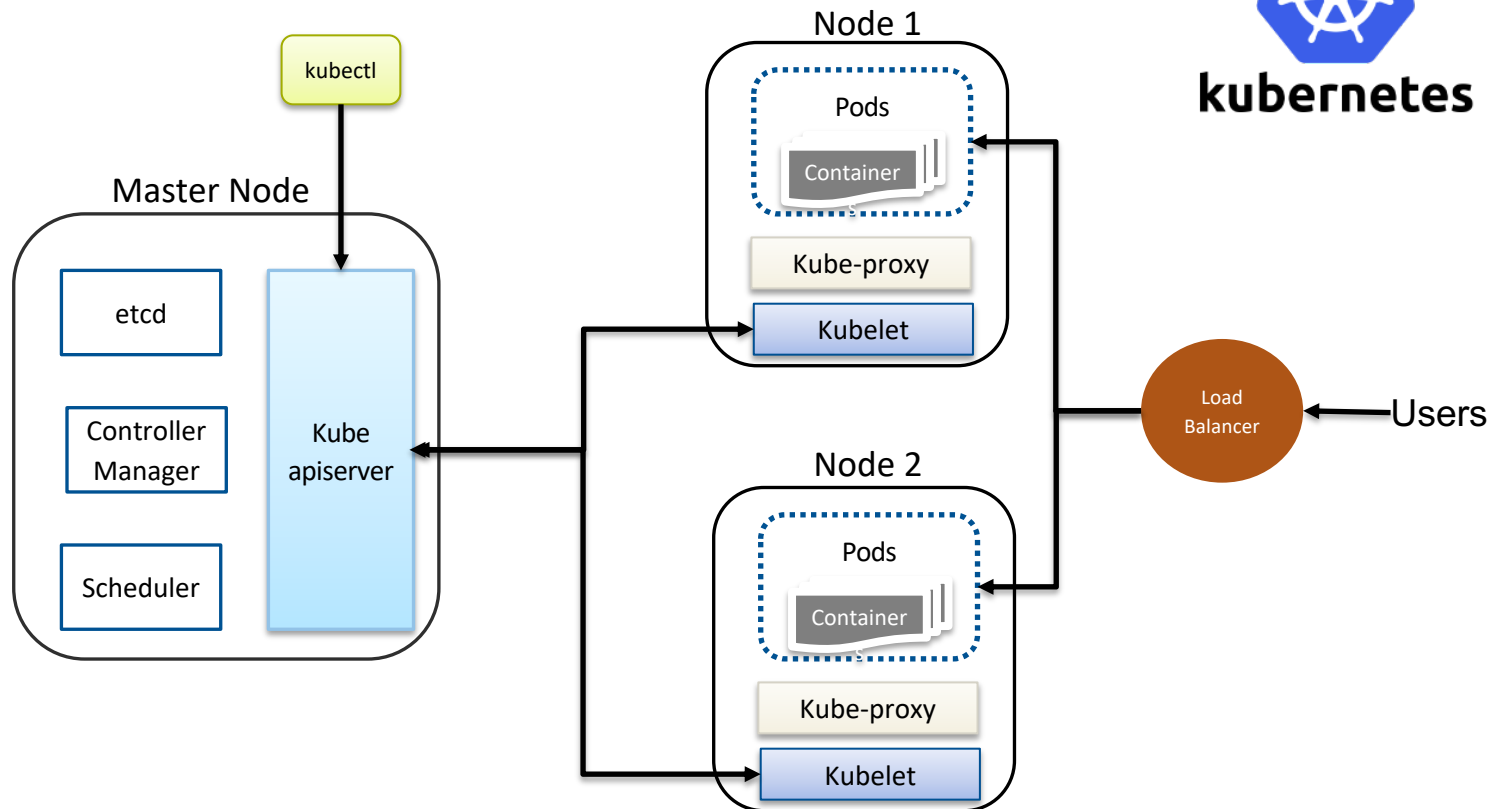


- **Tasks**

- Installation of OpenStack on a VM.
- Creation of at least one server inside OpenStack.
- Trace generation for the given command.
- Understanding of the service calls and their dependencies using the generated trace.

# Ex.5: App. deployment using Kubernetes

- Introduction to container orchestration.
- Introduction to Kubernetes and its architecture.

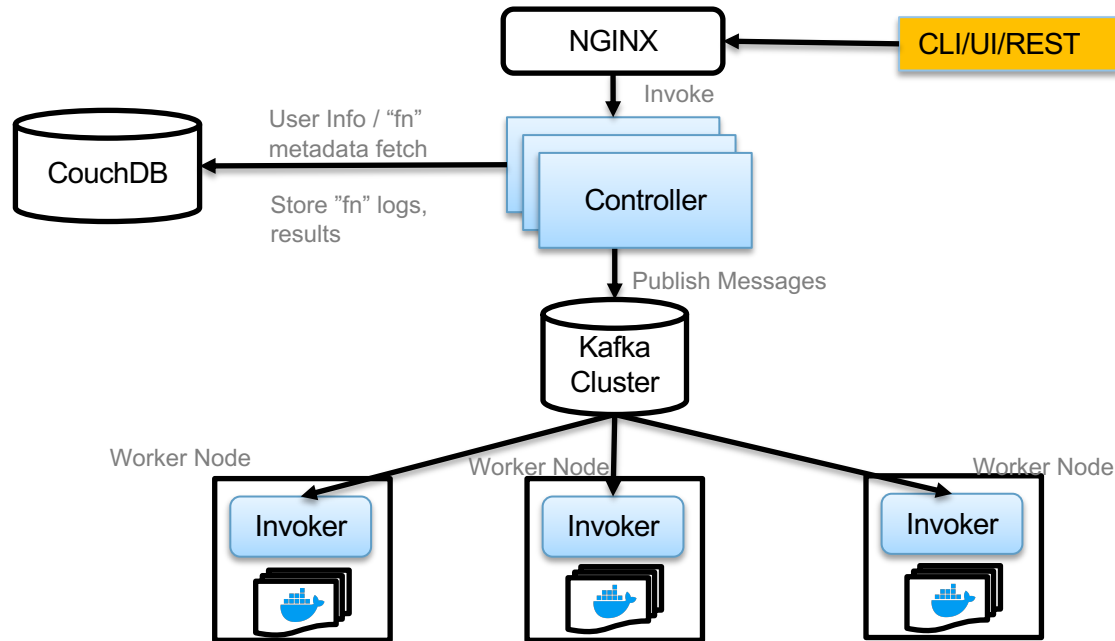


## • Tasks

- Kubernetes installation.
- Application deployment using Kubernetes.
- Scaling of service(s).

# Ex.6: Exercise on FaaS

- Introduction to Function-as-a-Service.
- Introduction to FaaS platform: OpenWhisk



## • Tasks

- OpenWhisk installation.
- Microservices application conversion to FaaS based.
- Deployment of application on OpenWhisk

# Information about group based exercises



- 2<sup>nd</sup> and onwards exercises can be submitted as part of a **group** as well.
- A **discussion forum** and **Group choice formation activity** to form groups will be created on the Moodle.
- A group can have a maximum of 2 participants.
- Due to limited accounts on LRZ we recommend to create student account on **AWS**. Use your official **TUM email id** for account creation and University name as “**Technische Universitaet Muenchen**”  
<https://www.awseducate.com/registration#INFO-Student>
- Or Create on **Google Compute Platform**:
  - Free Student credits here : [https://edu.google.com/programs/students/?modal\\_active=sign-up](https://edu.google.com/programs/students/?modal_active=sign-up)
  - \$350 free credits but would require bank or credit card details: <https://cloud.google.com/free/>
- Or we could provide Google Cloud credits (not sure yet, waiting for the reply from Google)

# Submission Instructions



- All exercise submissions are checked automatically.
- Submissions are done by submitting to a server which will be online by next week.
- Registered participants will get their credentials for logging to the server via e-mail.
- More about submissions will be explained with the first exercise.

Thank you for your attention!