

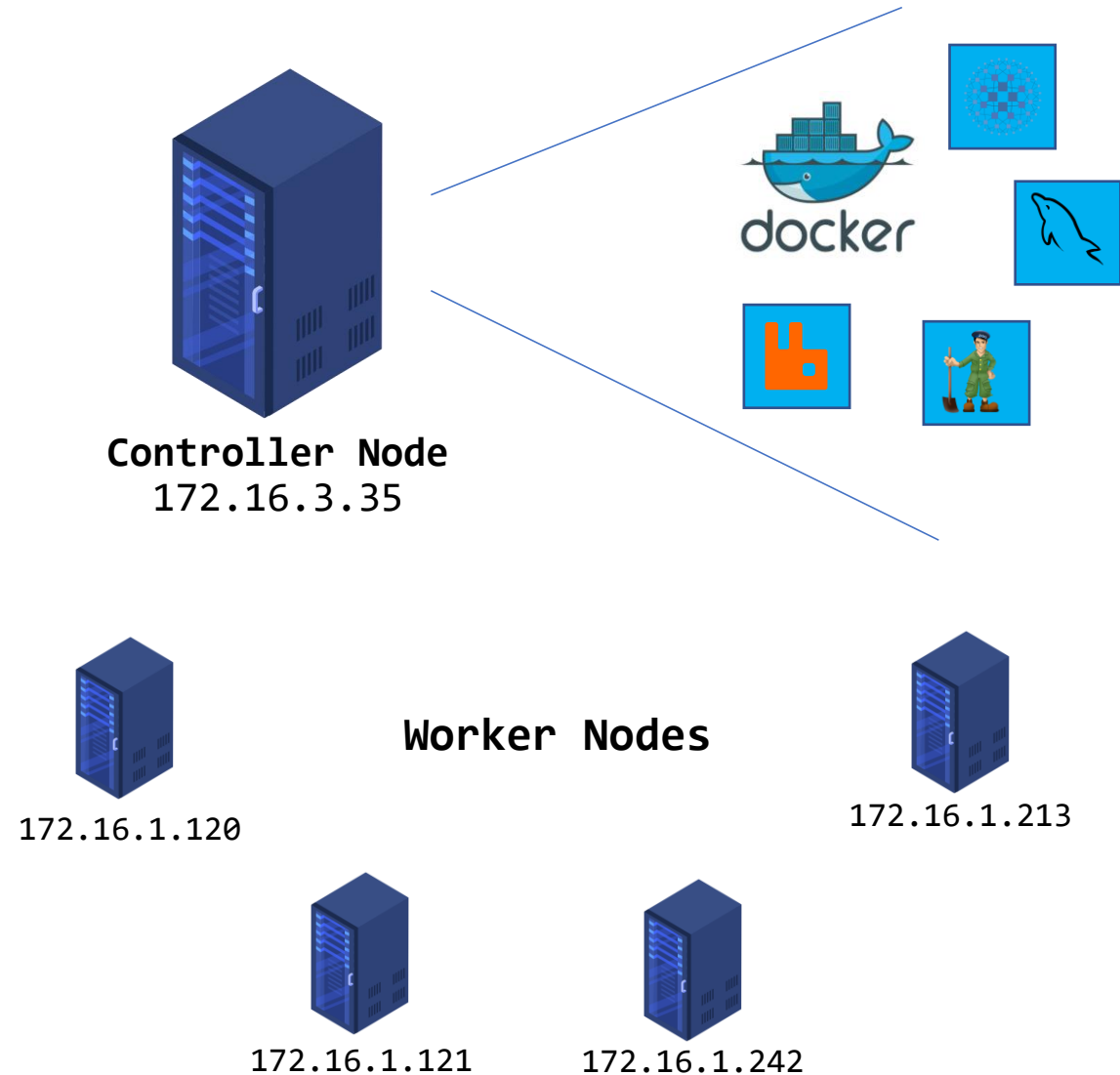


Cloud Computing Project Presentation

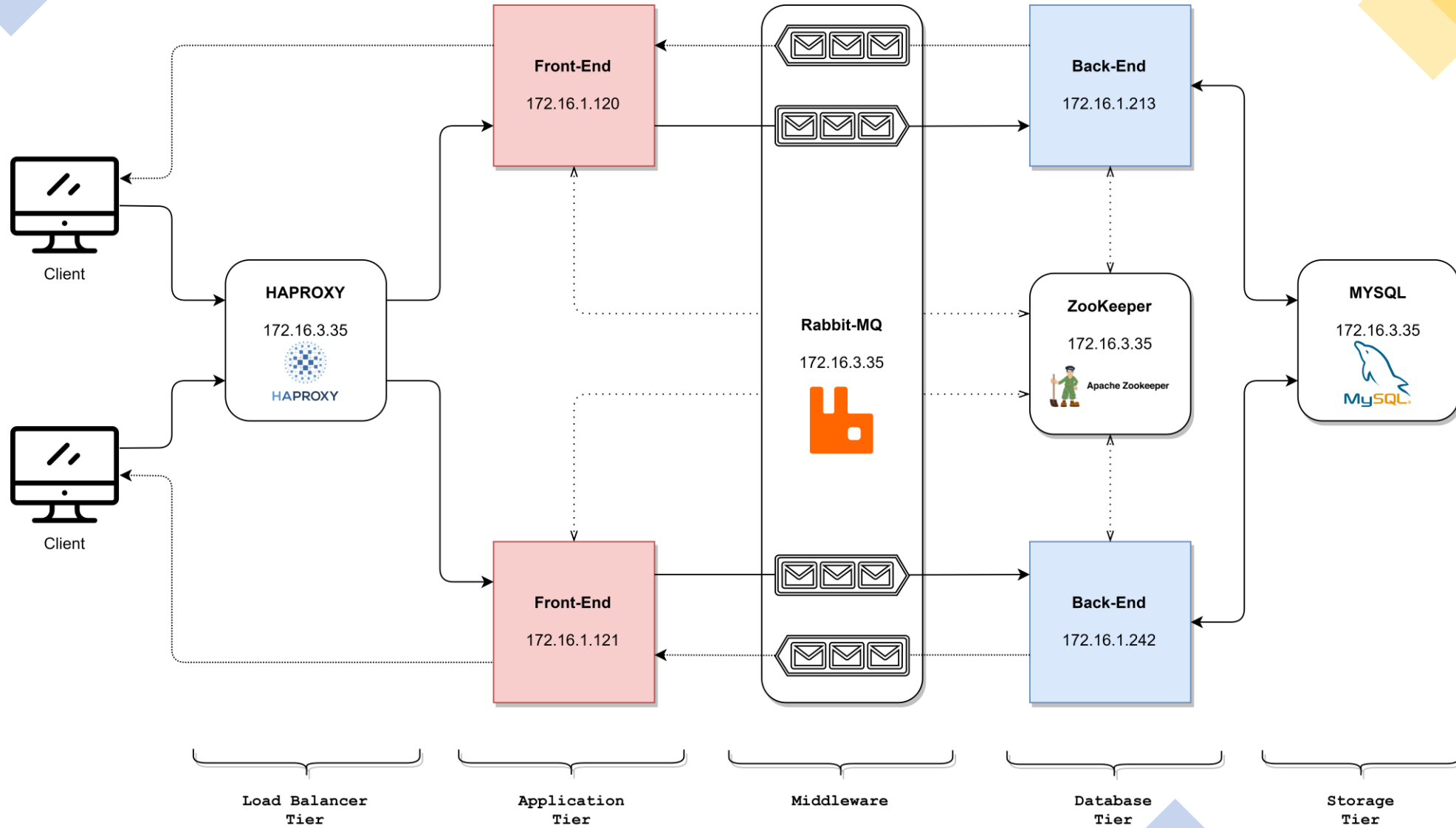
Alessandro Madonna,
Francesco Ronchieri, Andrea
Klaus Tubak, Stefano Petrocchi

OVERVIEW

- All components are deployed in **Docker containers**.
- The **Controller node** contains all the support modules needed to run the application. Can be replicated to ensure *high availability*.
- **Worker nodes** also run their software in containers and take care of the *front-end* and *back-end* tasks, they can be replicated to ensure *scalability*.

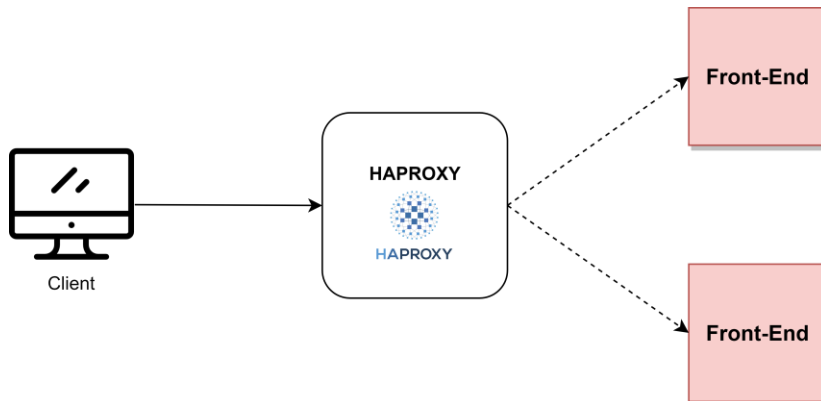


Schema



HAPROXY & FRONT-END

- **HAProxy** is deployed using a *dedicated node* model to load-balance the requests to the *front-ends*.
- The **front-ends** are obtained from a *yaml* declaration.



Movies

Movie collection



GET

/movies Get films filtered

PUT

/movies Create or update a movie

POST

/movies Add a new movie

GET

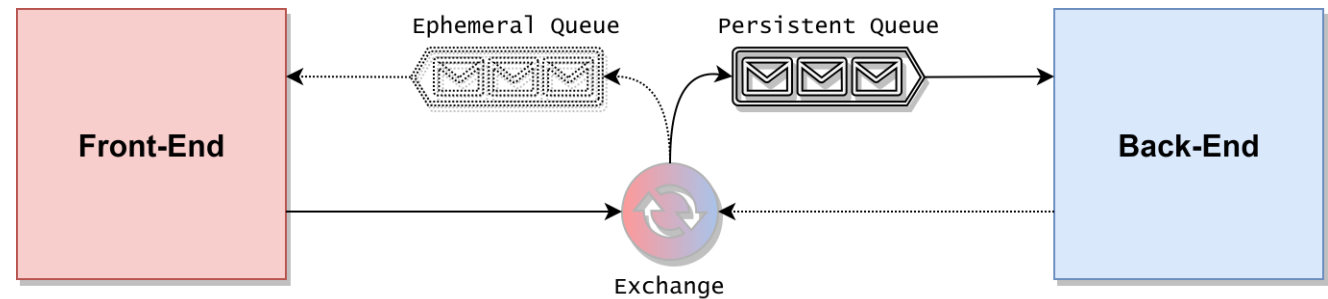
/movies/{movieId} Retrieve movie by ID

DELETE

/movies/{movieId} Deletes a movie

RABBITMQ

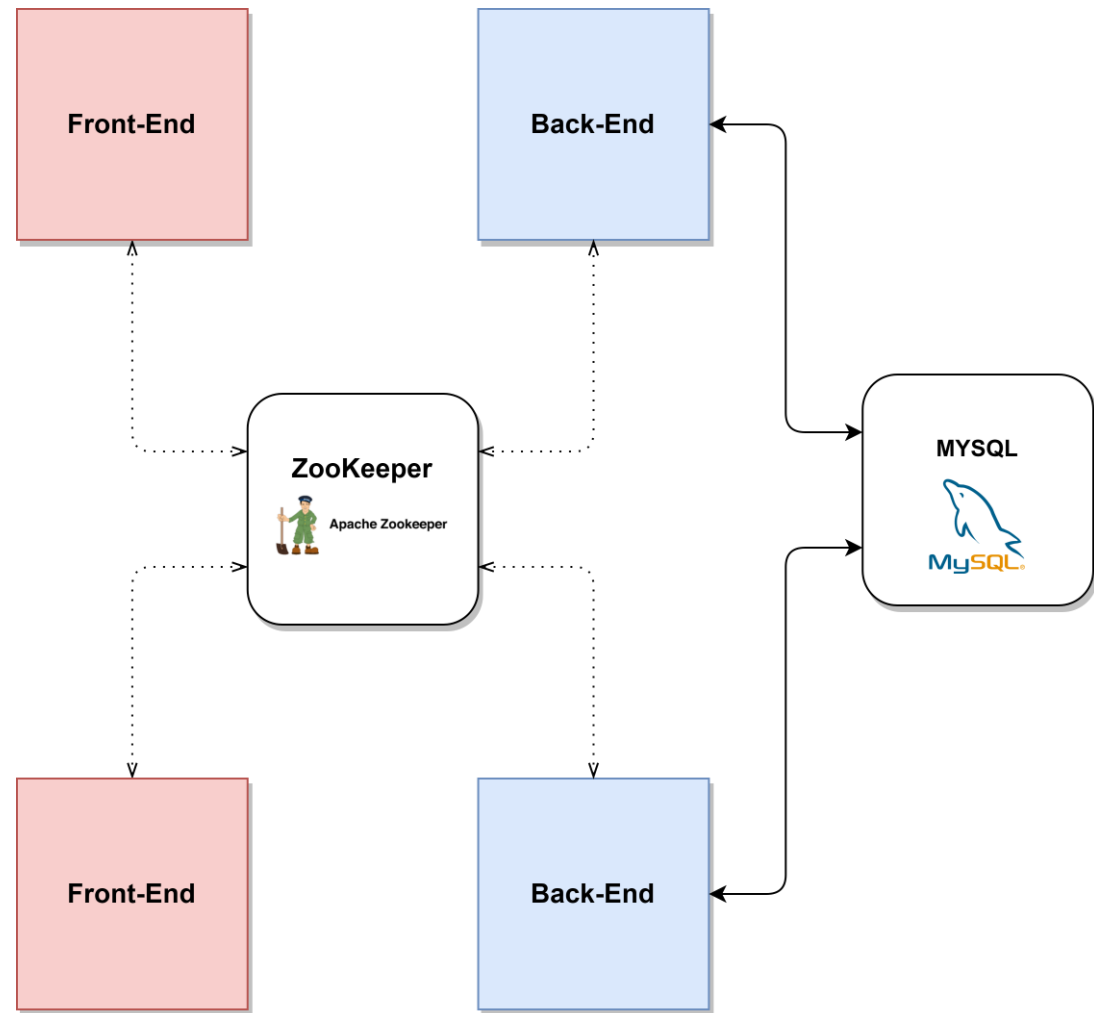
- **RabbitMQ** is deployed as single instance on the *controller*.
- **Communications** between *front-end* and *back-end* are managed by the message broker using the configuration in the figure.
- Each *front-end* instance communicates with a **single** *back-end* instance in order to maintain the *load-balancing* performed by *HAProxy*.



A single *exchange* sorts the messages to an **ephemeral queue**, created and then destroyed, for the *front-end* and to a **persistent** one for the *beck-end*.

BACK-END & MYSQL

- **Back-end** instances are connected to *MySQL* server.
- **MySQL** is deployed as single instance, in the *controller* node.
- **Zookeeper** is deployed as single instance in the *controller*, although more instances are usually preferred.
- *Zookeeper* is used to save the **configuration setting** of the *front-end* and *back-end*.



AUTO DEPLOY

- A *Python* script has been developed to **automatize** the deployment of the *front-end* and *back-end*.
- The script *deployEnviroment.py* is in charge of execute the necessary commands to **deploy** the modules:
 - **Upload** the *sources* on the machine.
 - **Configure** a *Json* file on the machines containing configuration data.
 - **Build** the *images* and deploy the *containers*.
- *config.json* contains the information which the script needs to configure the machines.

