

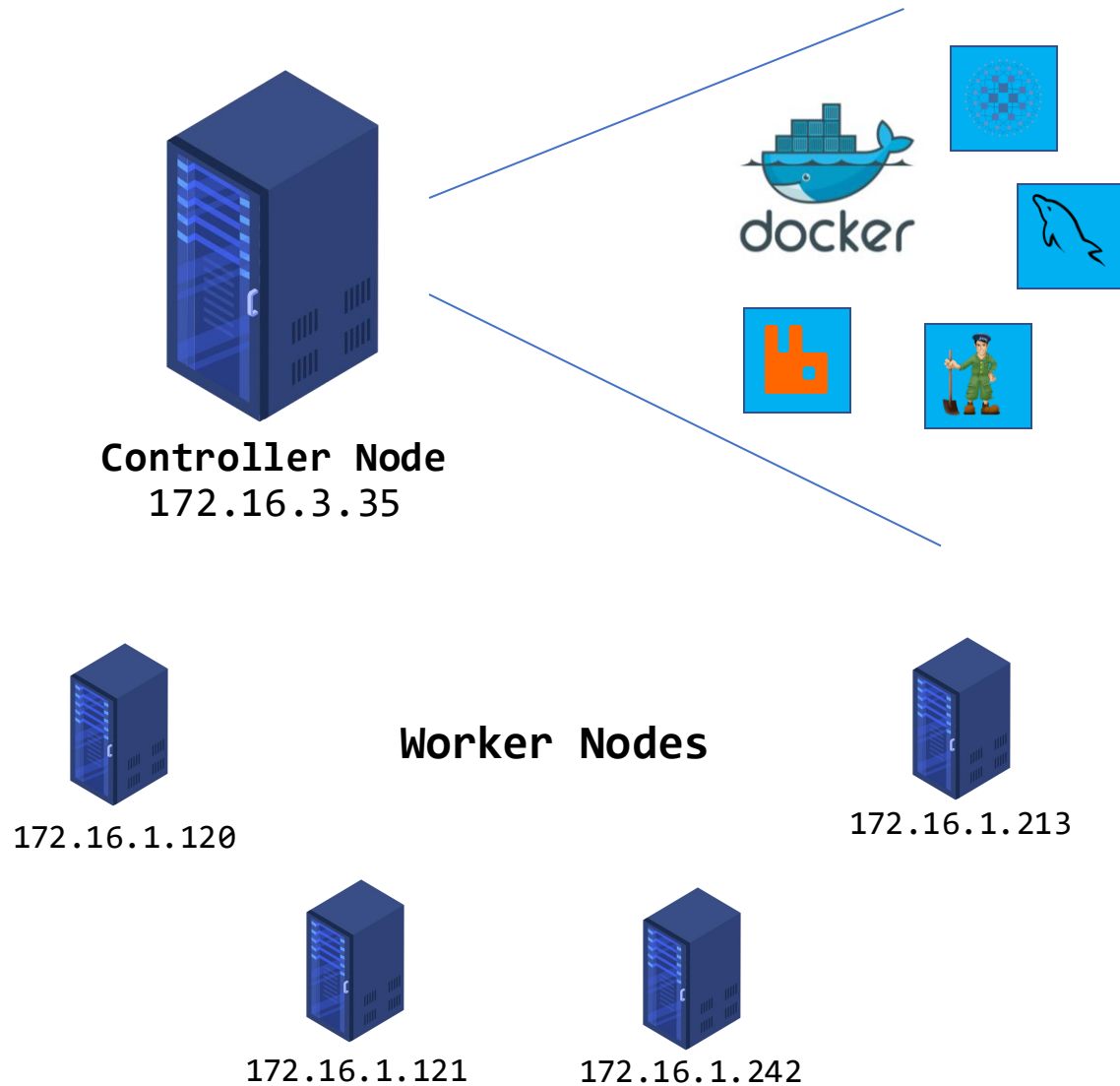


Cloud Computing Project Presentation

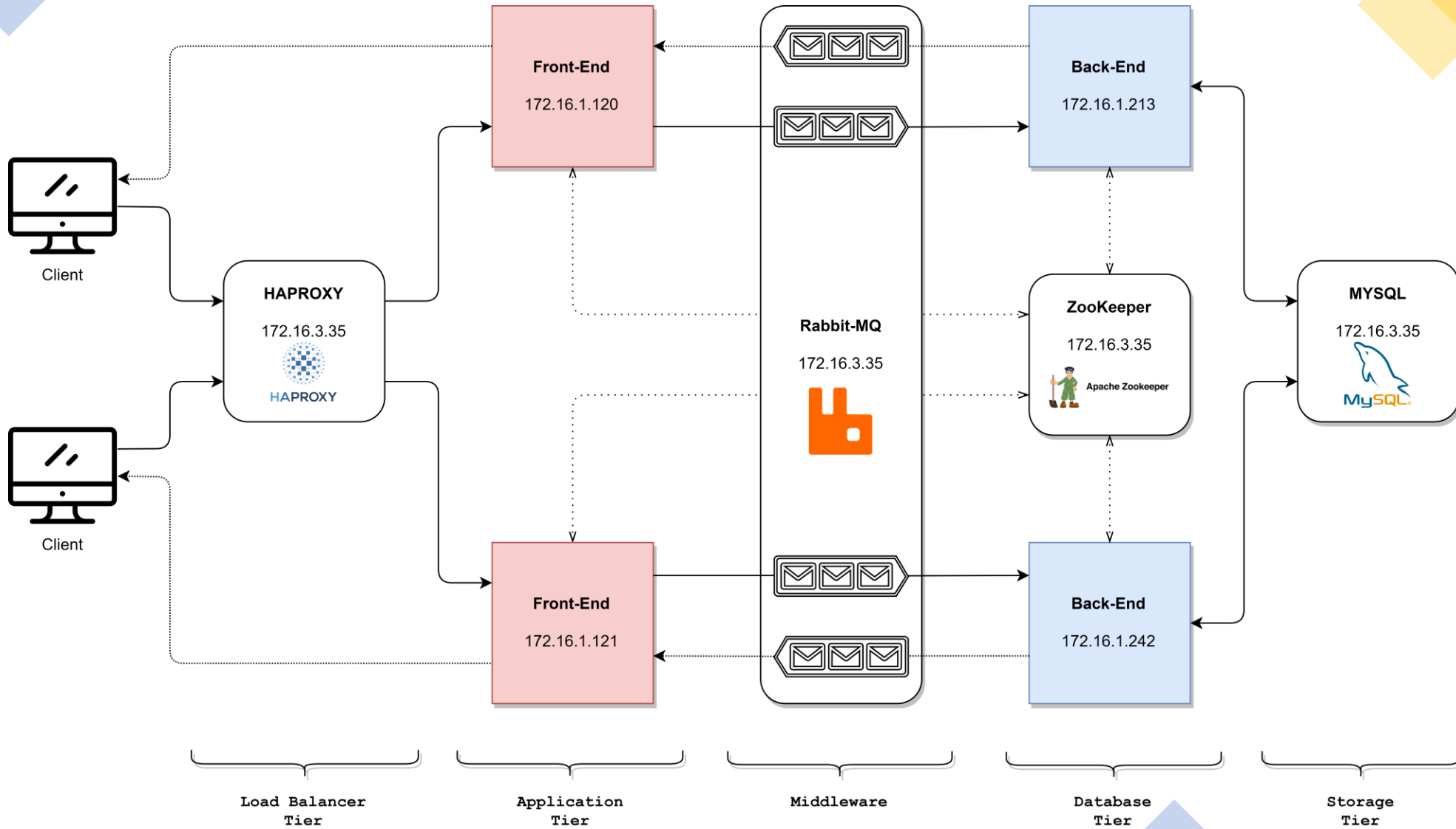
Alessandro Madonna,
Francesco Ronchieri, Andrea
Klaus Tubak, Stefano Petrocchi

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*.

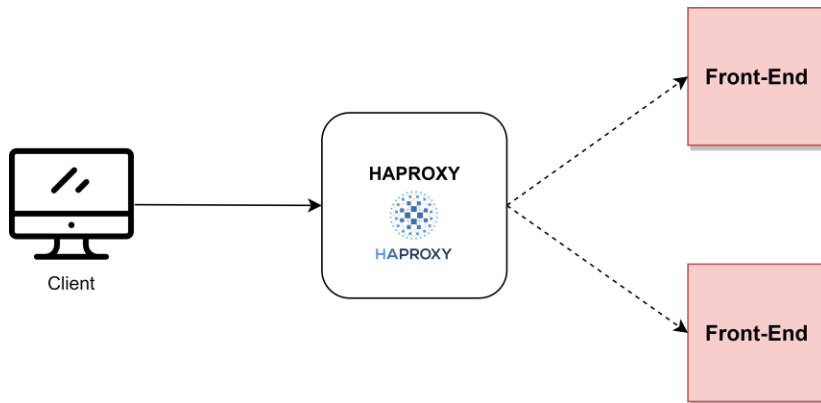


Schema



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

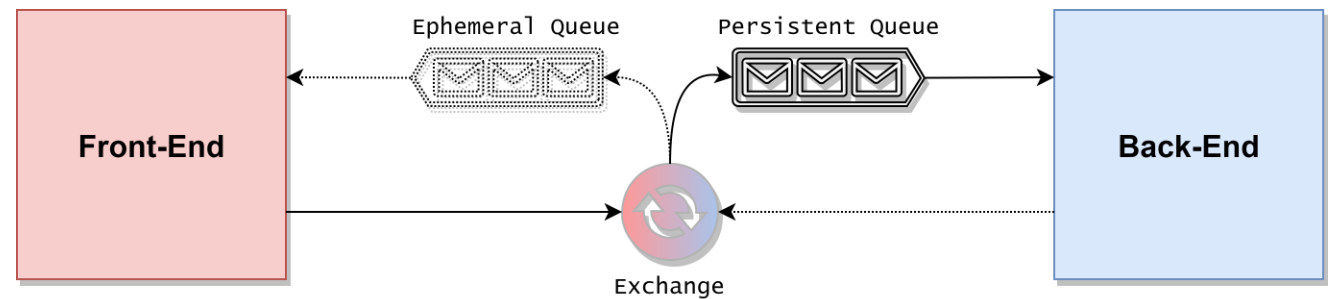
/movies/{movieId} Retrieve movie by ID

DELETE

/movies/{movieId} Deletes a movie

RABBITMQ

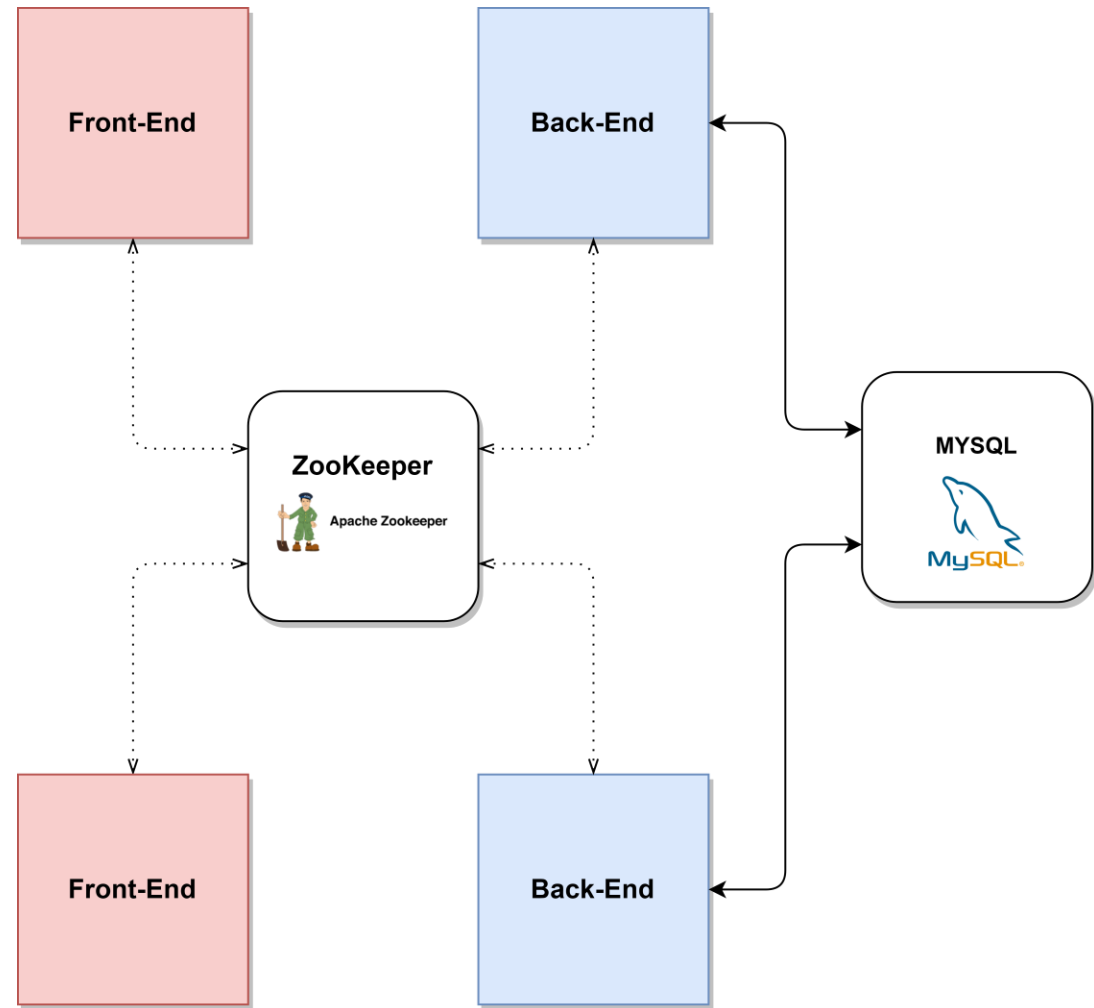
- **RabbitMQ** is deployed as single instance on the *controller*.
- **Communications** between *front-end* and *back-end* are managed by the message broker using the configuration in the figure.
- Each *front-end* instance communicates with a **single back-end** instance in order to maintain the *load-balancing*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

- **Automatization** A Python script has been developed to speed up the deployment of the front-end and back-end
- The script '*deployEviroment.py*' is in charge of execute the necessary commands to configure a machine:
 - **upload** the sources on the machine
 - **configure** a '*config.json*' file on the machine in order to associate it with its specific Exchange and Zookeeper
 - **build** the *images* and deploy the *container*
- In the local '*config.json*' are stored all information which the automated script needs to configure the machines.

```
1 import paramiko
2 import os
3 import zipfile
4 import tarfile
5 import json
6
7 config = {}
8
9 if __name__ == '__main__':
10     with open('config.json','r') as f:
11         config = json.load(f)
12         print('loading configuration:\n{config}')
13
14     tarDir('front-end','front-end')
15     for machine in config['front-end-machines']:
16         configureMachine('front-end',machineIP,machineType)
17         os.remove('front-end.tar.gz')
18
19     tarDir('back-end','back-end')
20     for machine in config['back-end-machines']:
21         configureMachine('back-end',machineIP,machineType)
22         os.remove('back-end.tar.gz')
23
24 def configureMachine(machineType,machineIP,machineType):
25     print('\n[INFO] ----- Configuring {machineType} -----')
26     configForMachine = {}
27     configForMachine["zookeeper-ip"] = config["zookeeper-ip"]
28     configForMachine["exchange"] = config["exchange"]
29     ssh = paramiko.SSHClient()
30     ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
31     ssh.connect(machineIP,22,username=sshUser,password=sshPassword)
32
33     executeSSHCommand(ssh,'rm -rf {machineType}')
34
35     sftp = ssh.open_sftp()
36     sftp.put('{machineType}.tar.gz'.format(machineType=machineType))
```

```
1 {
2     "version":"1.1",
3     "clean-front-end-container": true,
4     "clean-back-end-container": true,
5     "clean-front-end-image": true,
6     "clean-back-end-image": true,
7     "clean-base-front-end-image": false,
8     "clean-base-back-end-image": false,
9     "zookeeper-ip":"172.16.3.35",
10     "front-end-machines":[
11         {
12             "ip":"172.16.1.121",
13             "exchange":"exchange1",
14             "ssh-user":"root",
15             "ssh-password":"hal9000"
16         },
17         {
18             "ip":"172.16.1.120",
19             "exchange":"exchange2",
20             "ssh-user":"root",
21             "ssh-password":"hal9000"
22         }
23     ],
24     "back-end-machines":[
25         {
26             "ip":"172.16.1.242",
27             "exchange":"exchange1",
28             "ssh-user":"root",
29             "ssh-password":"hal9000"
30         },
31         {
32             "ip":"172.16.1.213",
33             "exchange":"exchange2",
34             "ssh-user":"root",
35             "ssh-password":"hal9000"
36         }
37     ]
38 }
```