Development of a virtualization framework with LXD

Òscar Pérez Castillo

July 7, 2021

Universitat Politècnica de Catalunya

Table of contents

1. Introduction

2. Project organization

3. Concepts

4. Development

5. Future work

6. Conclusion

Outline

Introduction

Project organization

Concepts

Development

Future work

Conclusion

Introduction

Using Linux Containers, develop a framework on top of existing containerization solutions (LXC/LXD) improving:

- Containers organization
- Containers set up
- Containers distribution

Introduction

Developing a set of tools:

- lxce: command line tool
- lxce-admin: admin command line tool
- web application

Outline

Introduction

Project organization

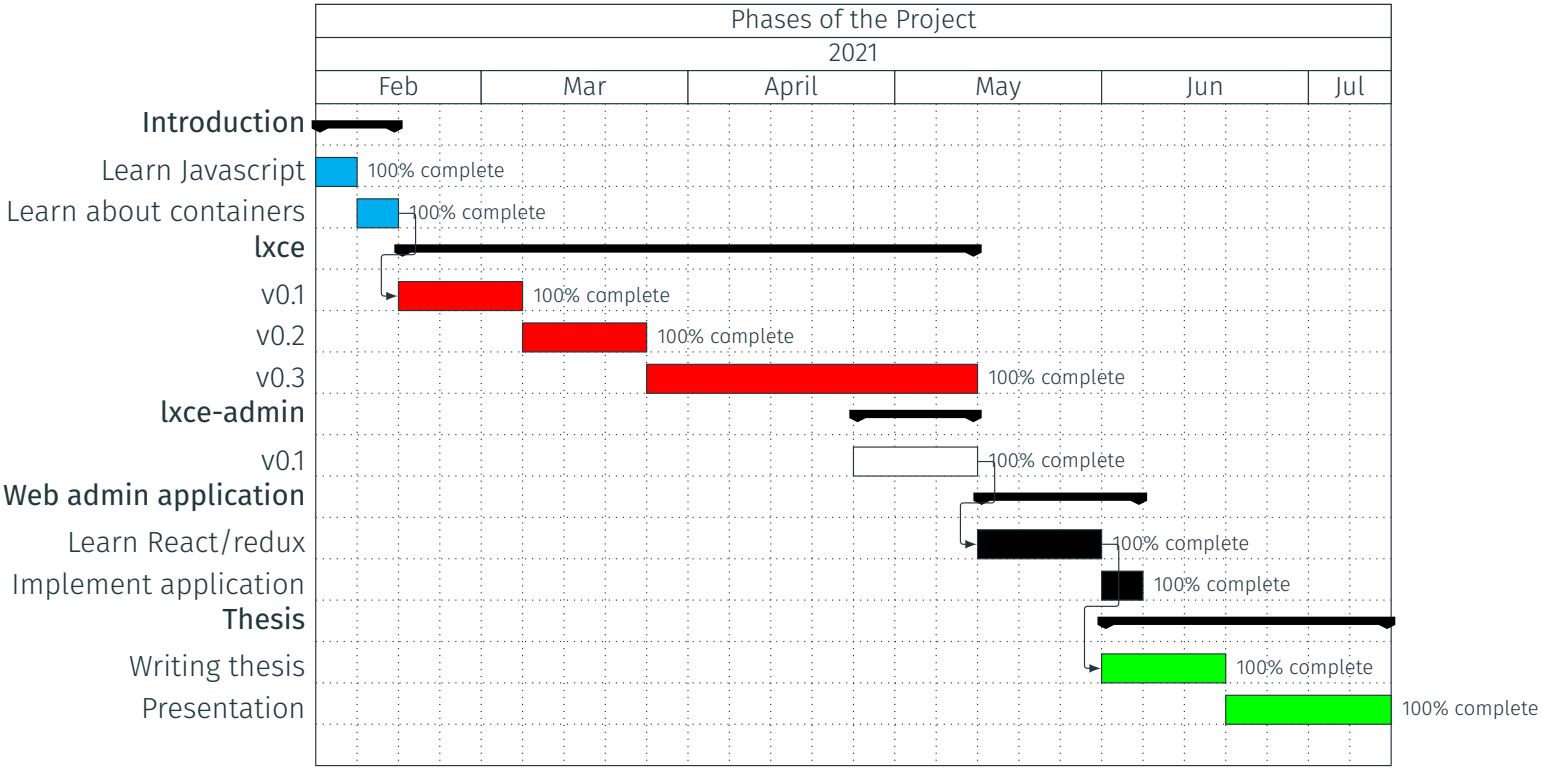
Concepts

Development

Future work

Conclusion

Project organization



Outline

Introduction

Project organization

Concepts

Development

Future work

Conclusion

Concepts

Project based on:

- Containers technology
- LXC
- LXD

Concepts: Containers

Virtualization vs Containers

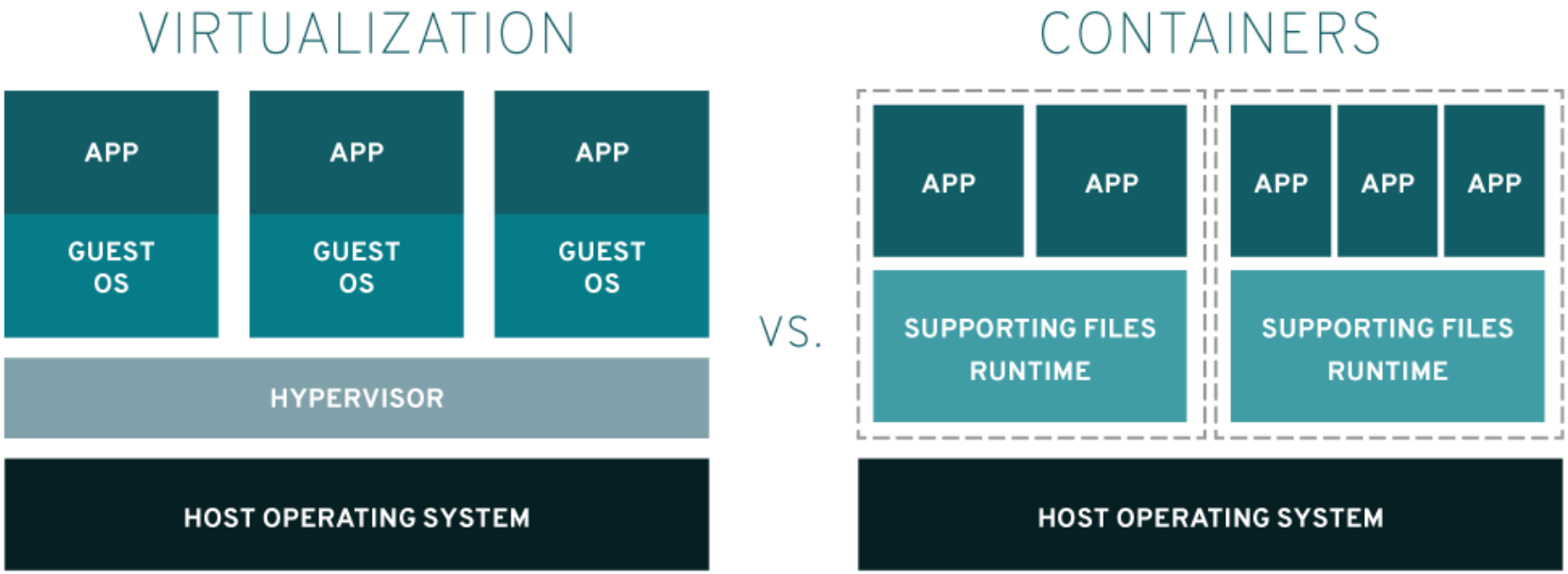


Figure 1: ref: link

Concepts: Containers

Abstracion created by the Linux Kernel:

- Namespaces
- Cgroups

which provides:

- Lightweight solution
- Isolation and control of resources

Concepts: LXC

- C library (liblxc)
- Programming language bindings
- Tools for controlling containers
- Linux Distribution templates

Concepts: LXD

- Build on top of LXC
- REST API
- New command line tool ”lxc”
- Integration with containers services and other advanced features

Concepts: LXD

Examples:

```
# Launch container
lxc launch ubuntu:20.04 box

# Launch a bash inside container
lxc exec box bash

# Add mapped proxy to the container
lxc config device add box testport80
↪  listen=tcp:0.0.0.0:80 connect=tcp:127.0.0.1:80

# Share host folder with container
lxc config device add box device www disk source=/www
↪  datapath=/var/www/html
```

Outline

Introduction

Project organization

Concepts

Development

Future work

Conclusion

Development

Objective to develop a framework:

- Improve existing ”lxc” command line tool - lxce
- Develop an admin tool - lxce-admin
- Develop a visual alternative to command line tool - web-admin

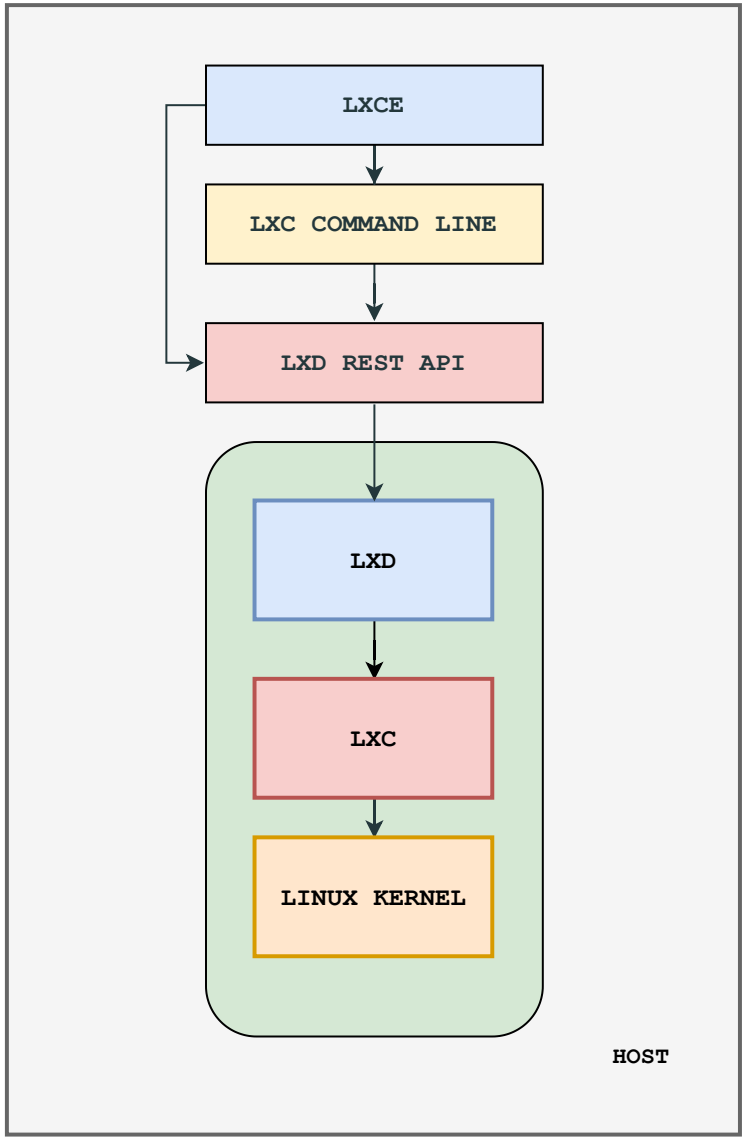
Development: lxce

lxce command line tool:

- Manage containers by configuration files.
- Organize containers by “domains”.
- Organize containers by aliases.
- Configure proxies and shared folders by a configuration file.
- Generate SSH and VNC configuration files to be distributed.

Development: lxce

Architecture:



Development: lxce

Commands:

- lxce init
- lxce alias
- lxce delete
- lxce launch
- lxce list
- lxce pass
- lxce proxy
- lxce rebase
- lxce show
- lxce start
- lxce stop
- lxce uninstall

Development: lxce

Configuration files:

```
/etc/lxce
|--- container.conf.d
|   |--- default
|   |   '--- voiceless-blue
|   '--- derecho
|       '--- relieved-beige
|--- container_default.conf
|--- lxce.conf
|--- remmina
|   |--- default
|   |   '--- oscar-vm.default.voiceless-blue.remmina
|   '--- derecho
|       '--- oscar-vm.derecho.relieved-beige.remmina
'--- ssh
    |--- default
    |   '--- voiceless-blue.conf
    '--- derecho
        '--- relieved-beige.conf
```

Development: lxce

Default configuration file:

```
{
  "name": "",
  "alias": "",
  "user": "",
  "id_domain": 0,
  "id_container": 0,

  "domain": "default",
  "base": "ubuntu:20.04",
  "userData": "/datasdd",

  "proxies": [
    {
      "name": "ssh",
      "type": "tcp",
      "listen": "0.0.0.0",
      "port": 22
    },
    {
      "name": "test",
      "type": "tcp",
      "listen": "0.0.0.0",
      "port": 3000
    }
  ],
}
```

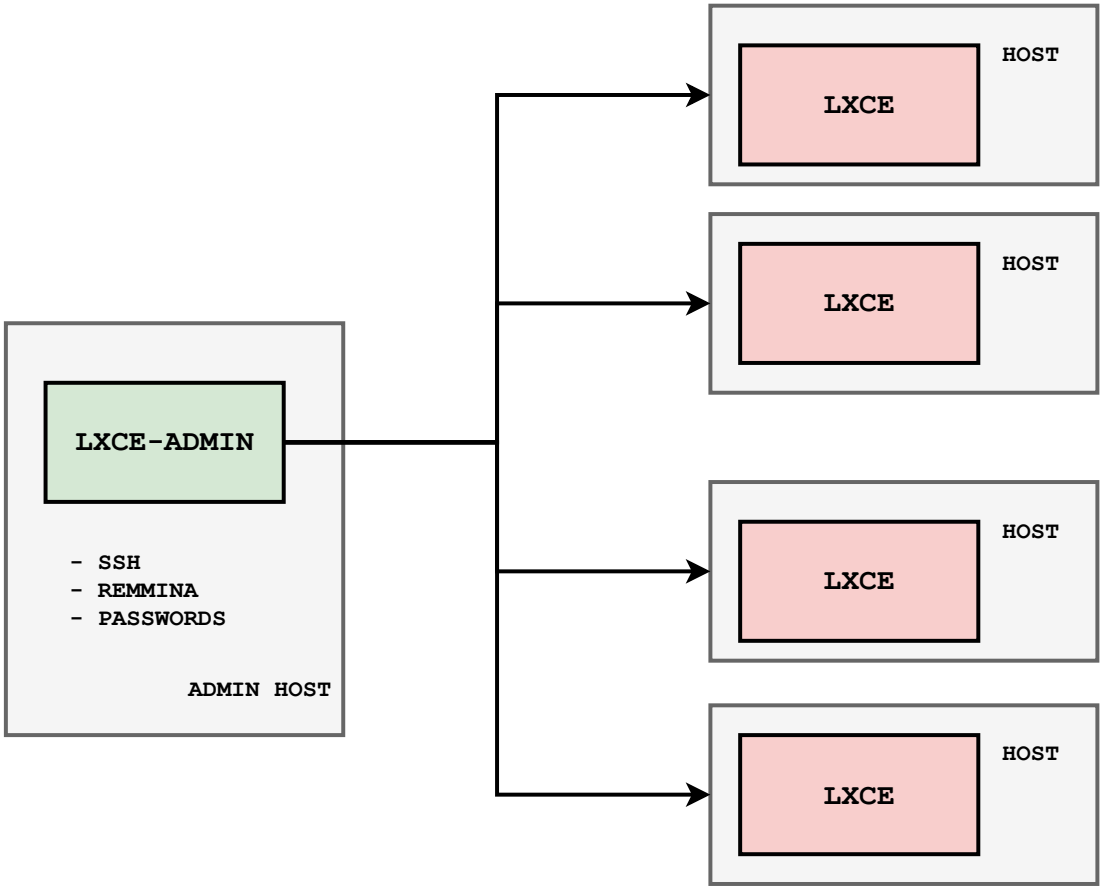
Development: lxce-admin

lxce-admin command line tool:

- Complete view of all container across different hosts
- Access to SSH and VNC configuration files
- VNC clients integrated
- Compute passwords for each container

Development: lxce-admin

Architecture:



Development: lxce-admin

Commands:

- lxce-admin config add
- lxce-admin config list
- lxce-admin config remove
- lxce-admin config update
- lxce-admin pass
- lxce-admin remmina
- lxce-admin vnc

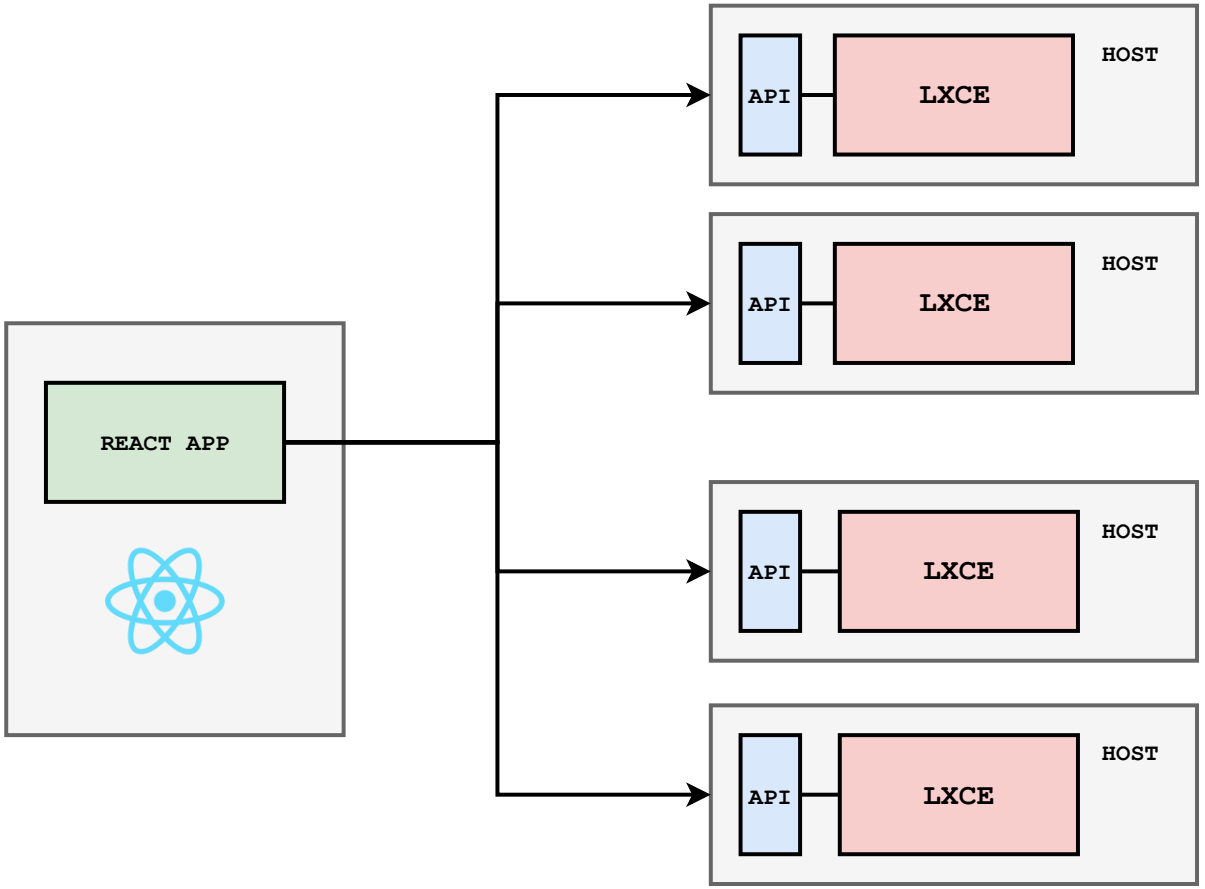
Development: web-admin

web-admin React web application:

- Visualize all current containers from each host
- Served along an API for each host
- Possibility to be extended

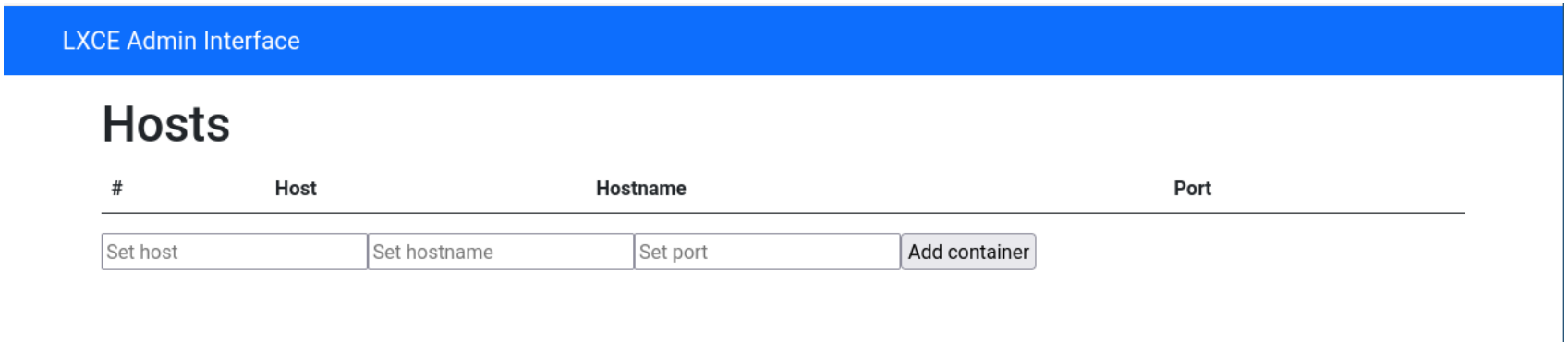
Development: web-admin

Architecture:



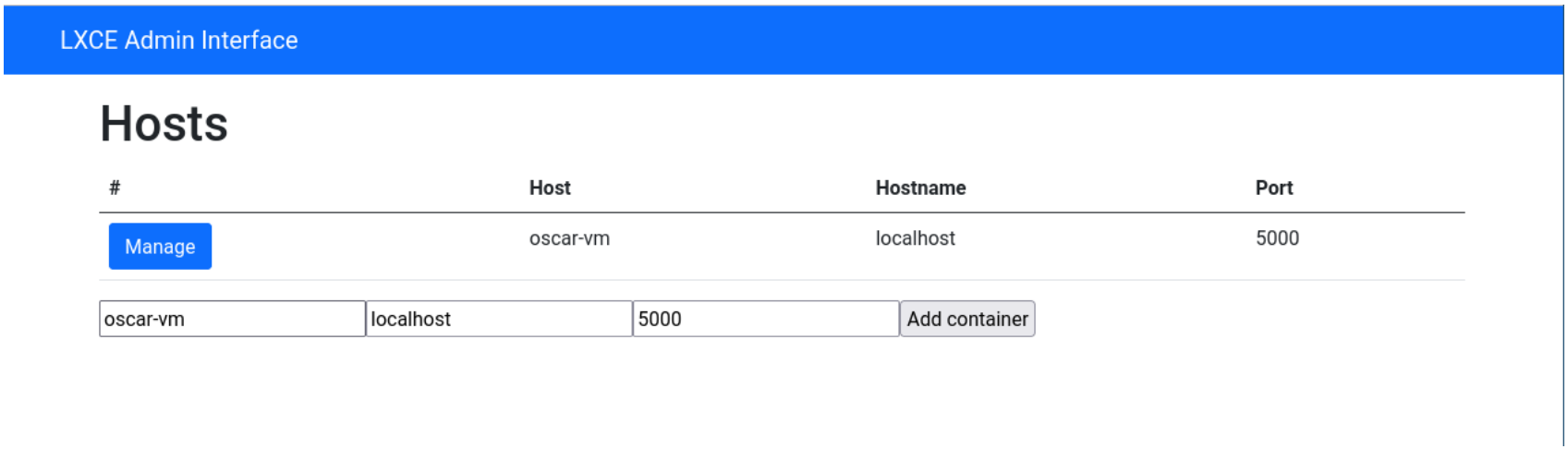
Development: web-admin

Application:



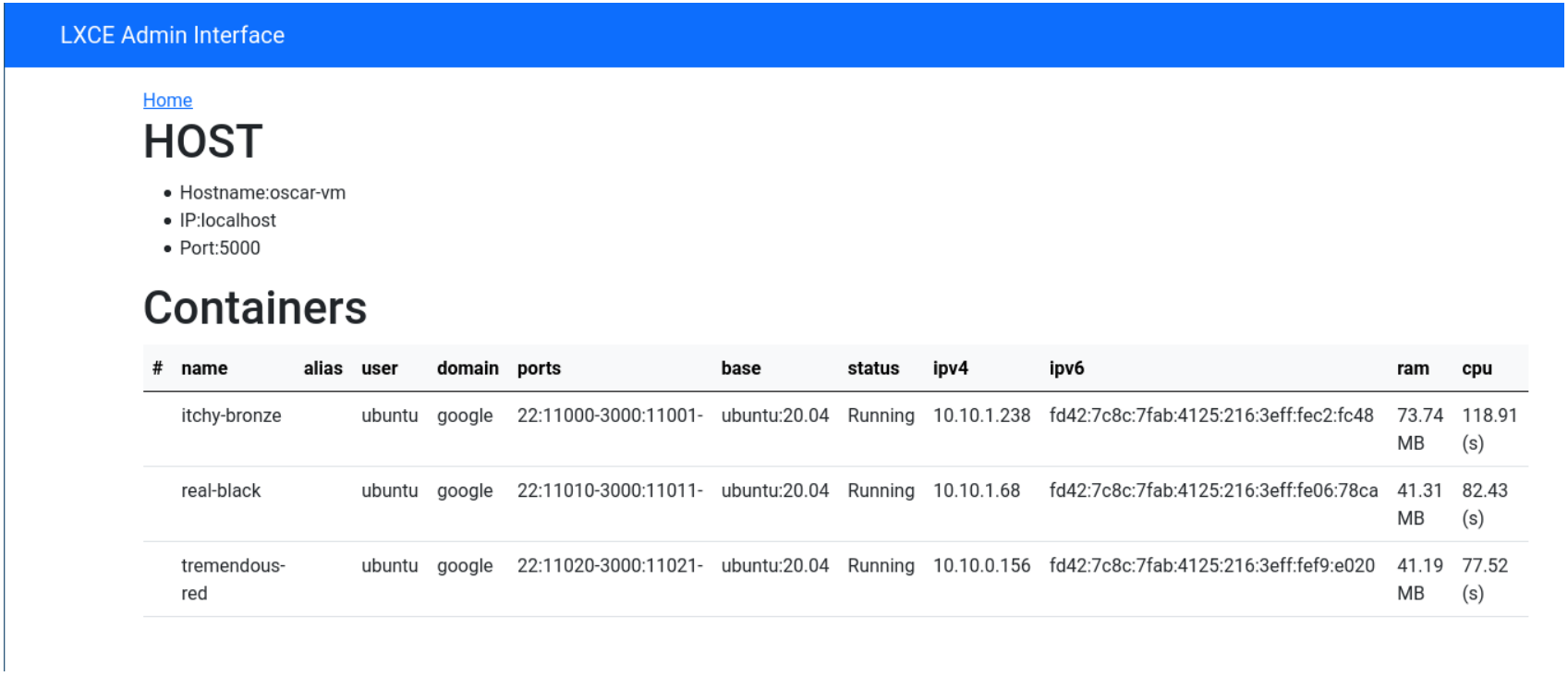
Development: web-admin

Application:



Development: web-admin

Application:



Outline

Introduction

Project organization

Concepts

Development

Future work

Conclusion

Future work

Possible improvements:

- Add nginx and certificates functionality
- Extend API to provide more functionalities

Outline

Introduction

Project organization

Concepts

Development

Future work

Conclusion

Conclusion

Learn:

- Javascript/Typescript
- Containers
- Web development
- Systems administration

Thanks!