Rancher Training

Session 1

RANCHER LABS

Arvin • 赵雨川

Morning Agenda

- Rancher Overview
- Rancher UI Walkthrough
- Authentication
- Environments
- Hosts & Registries
- Cattle: Stacks & Services
- Exercise 1&2

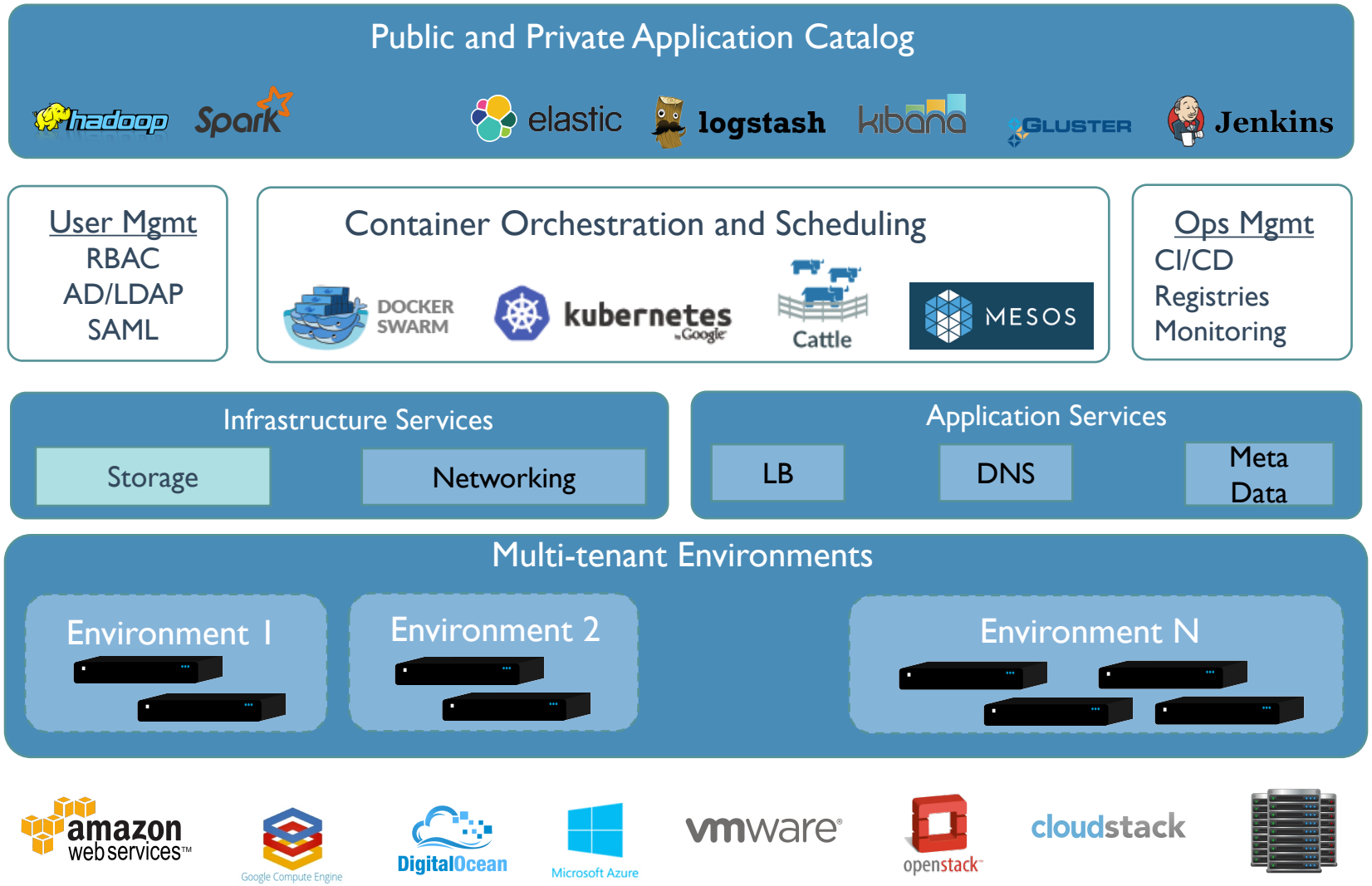
Rancher Labs

An open-source software platform for managing containers

A simplified Linux distribution built from containers, for containers

An open source project for microservices-based distributed block storage

Rancher 1.6

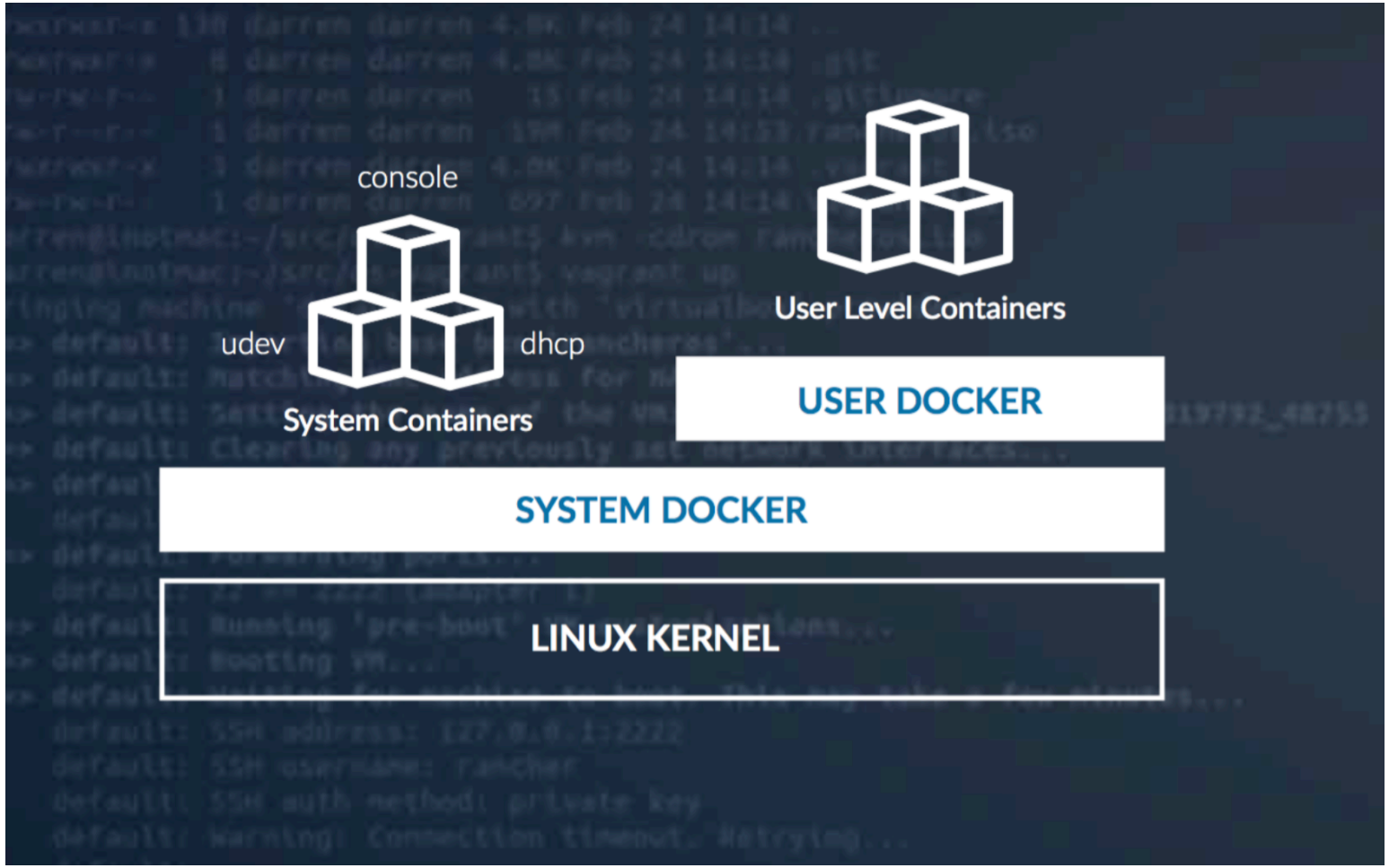

Public and Private Application Catalog
hadoop
Spark
elastic
logstash
kibana
GLUSTER
Jenkins
User Mgmt
RBAC
AD/LDAP
SAML
Container Orchestration and Scheduling
DOCKER SWARM
kubernetes
Google
Cattle
MESOS
Ops Mgmt
CI/CD
Registries
Monitoring
Infrastructure Services
Storage
Networking
Application Services
LB
DNS
Meta Data
Multi-tenant Environments
Environment 1
Environment 2
Environment N
amazon web services
Google Compute Engine
DigitalOcean
Microsoft Azure
vmware
openstack
cloudstack

Rancher 2.0

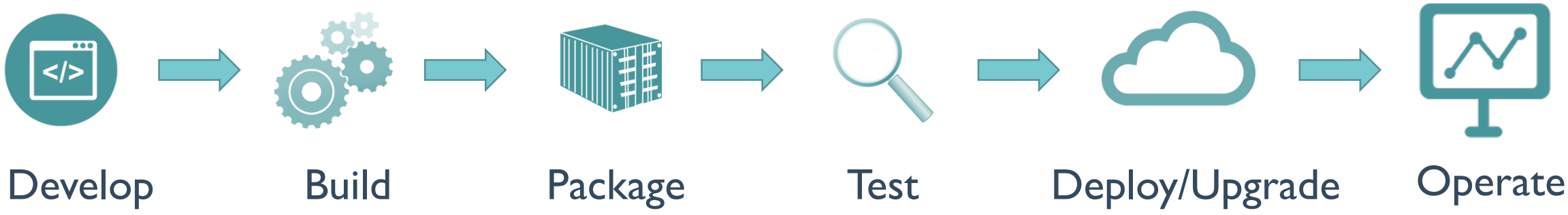
RANCHER

Complete Container Management Platform

RancherOS


console
udev
dhcp
System Containers
User Level Containers
USER DOCKER
SYSTEM DOCKER
LINUX KERNEL

Rancher Pipeline



GitHub

Experimenta
Rancher Pipeline
Rancher CI service

HARBOR™

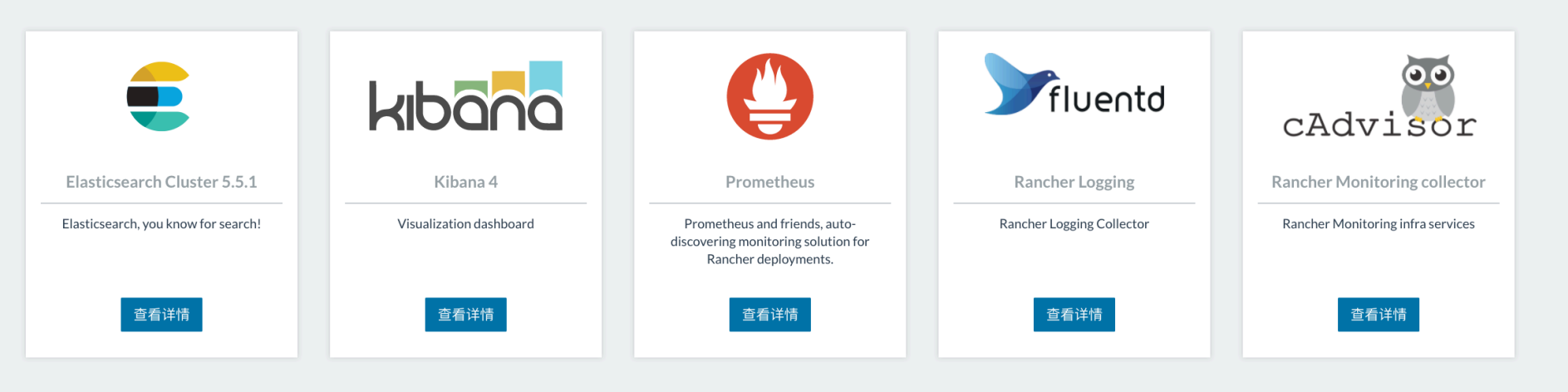
RANCHER

amazon web services™

vmware®

Monitoring & Logging

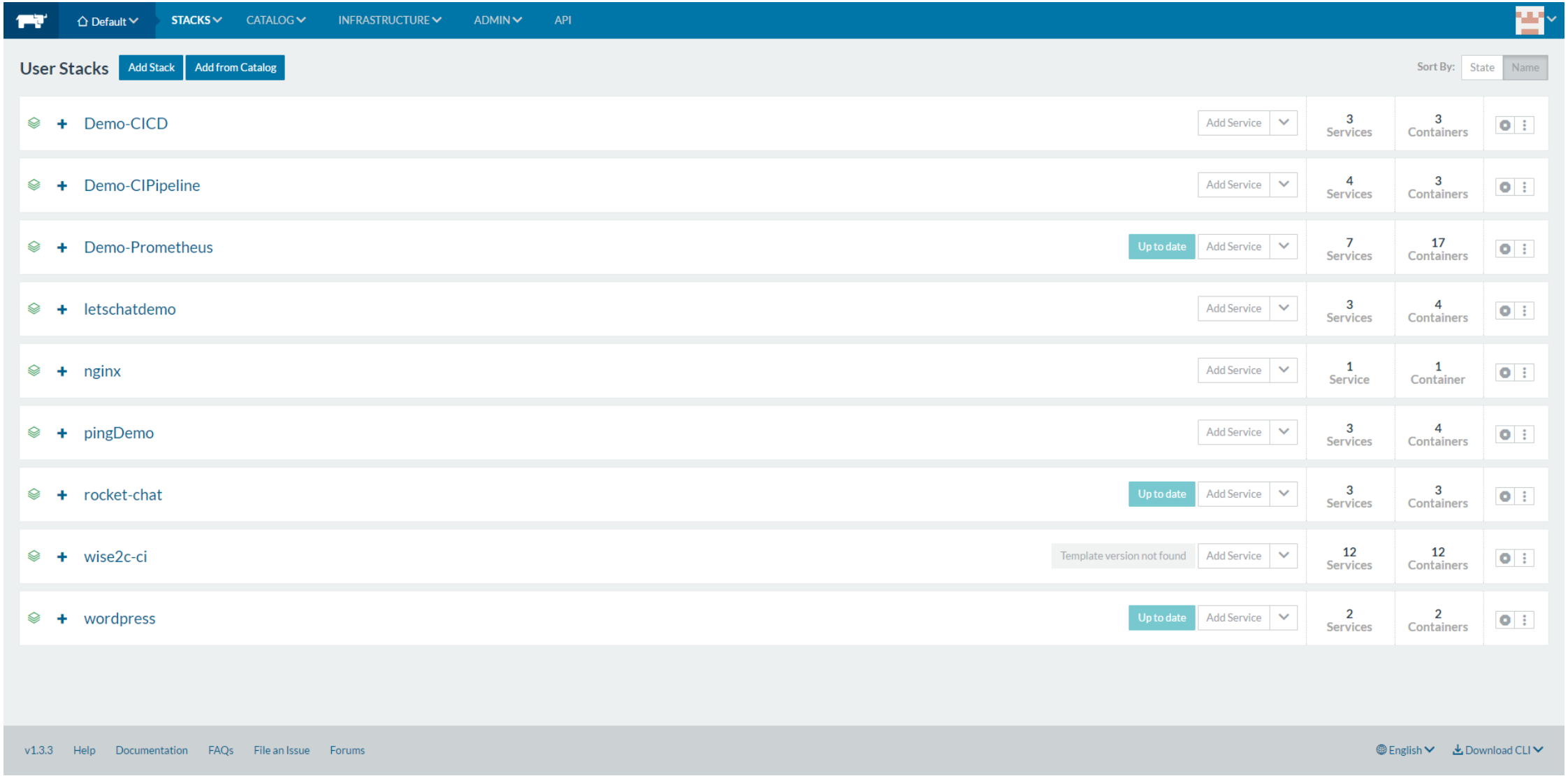
- Catalog:
- https://github.com/zionwu/monitoring-logging-catalog



Rancher UI

- ADMIN
- INFRASTRUCTURE
- STACKS
- CATALOG
- API
- ENVIRONMENT

Rancher UI Walkthrough



Authentication & Access Control

- Access Control Providers
- Rancher Site Access
- Accounts

Access Control Providers

- Active Directory
- Azure AD
- Github
- Local Authentication
- OPENLDAP
- Shibboleth

Accounts

- Account Type
- Determines whether or not an account will have access to the admin tab
- Admin (ability to view all the environments by default)
- Users (can only view the environment they are members of)

- Membership Roles

Environments

- Environments
- Managing Environments
- Environment Templates
- Roles in an Environments

Environments

- Administrative Boundary
- Hosts only associated with a single environment
- Per Environment scheduling tool

Managing Environments

- Adding Environments
- Deactivating and deleting environments
- Editing members

- Membership roles
  - Provide different level of access for a specific environment
  - Owners
  - Members
  - Restricted
  - Read only

Environment Template

- An environment template allows users to define a different combination of infrastructure services to be deployed.

Hosts & Registries

- Hosts
- Registries

Hosts

- Any modern Linux distribution with supported version of Docker
- Check release note

Supported Docker Versions

- Docker 1.12.3-1.12.6
- Docker 1.13.1
- Docker 17.03-ce/ee
- Docker 17.06-ce/ee
- Docker 17.09-ce/ee
- Docker 17.12-ce/ee

> Note: Kubernetes 1.8 supports Docker 1.12.6, 1.13.1 and 17.03.2. Kubernetes 1.7 supports up to Docker 1.12.6

Hosts

- Add hosts
- Custom hosts
- Docker Machine Drivers
- Registration token
- Agent account & API key pair
- Edit Host
- Evacuate
- Clone Host
- Deactivate/Activate Host
- Remove Host

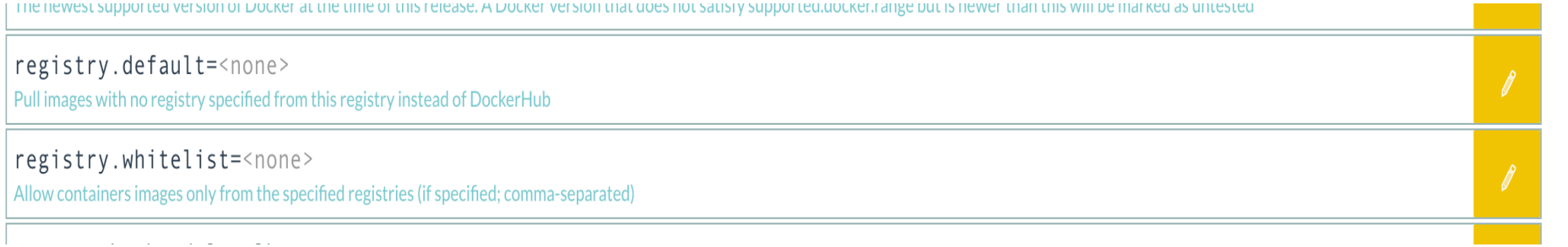
Registries

- Add credentials to access private registries
- Only one credential per registry address
- Insecure registries must be configured manually (for custom host)

Registries

- Change default registry
- Registry whitelist

Admin--Settings--Advanced Settings



Cattle: Stacks & Services

- Stacks
- Services
- Load Balancers
- Health Checks
- Scheduling

Stacks

- A group of services that together implement an application
- Add Stack
  - Manually
  - Compose file
- Stack Configuration
- Graph View

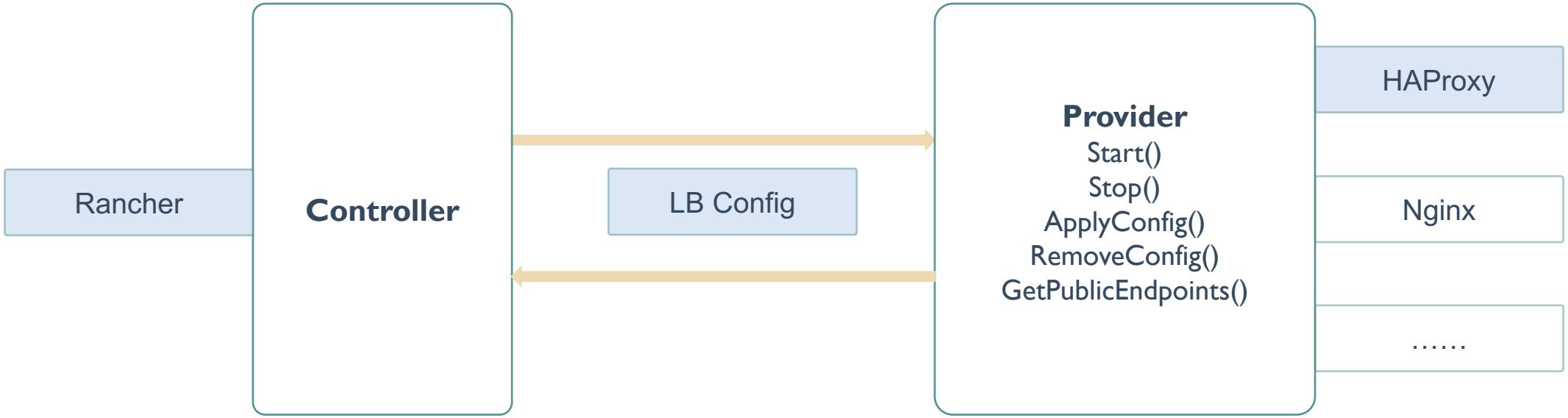
Services

- One or more containers created from the same Docker image
- Service options
- Docker compose version supported
- Sidekick service

Load Balancer

- Internal – not exposed on host port
- External – used to expose internal service to outside world
- Advanced routing:
- SNI
- Host based routing
- Path based routing

Load Balancer



```
bind *:38088
mode http
acl 38088_api_path path_beg -i /api
use_backend 38088_api if 38088_api_path
default_backend 38088_

backend 38088_api
acl forwarded_proto hdr_cnt(X-Forwarded-Proto) eq 0
acl forwarded_port hdr_cnt(X-Forwarded-Port) eq 0
    http-request add-header X-Forwarded-Port %[dst_port] if forwarded_port
    http-request add-header X-Forwarded-Proto https if { ssl_fc } forwarded_prot
o
mode http
server 10.42.85.93 10.42.85.93:8080

backend 38088_
acl forwarded_proto hdr_cnt(X-Forwarded-Proto) eq 0
acl forwarded_port hdr_cnt(X-Forwarded-Port) eq 0
    http-request add-header X-Forwarded-Port %[dst_port] if forwarded_port
    http-request add-header X-Forwarded-Proto https if { ssl_fc } forwarded_prot
o
mode http
server 10.42.183.159 10.42.183.159:80
```

Healthcheck

- Performed by healthcheck infrastructure service (one per host)
- Utilize HAProxy (port 42)
- Support TCP and HTTP (no exec support)
- Each container is monitored by up to 3 healthcheck containers running on separate hosts
- In healthy state only when at least one HAProxy instance reports “passed”

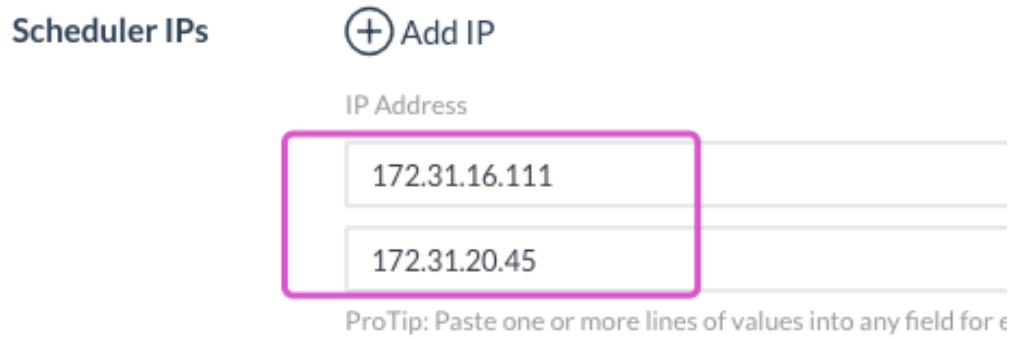
Scheduling

- port conflicts
- host tagging
- strict and soft affinity/anti-affinity rules

Label	Value
io.rancher.stack.name	`$${stack_name}`
io.rancher.stack_service.name	`$${stack_name}/$${service_name}`

```
io.rancher.scheduler.affinity:container_label_soft_ne:
io.rancher.stack_service.name=$${stack_name}/$${service_name}
```

Scheduling against multiple IPs of a host




```
Chain CATTLE_PREROUTING (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 MARK       udp  --  !docker0 *       0.0.0.0/0            0.0.0.0/0           udp dpt:500 MARK set 0x1068
    0     0 DNAT       udp  --  !docker0 *       0.0.0.0/0            0.0.0.0/0           udp dpt:500 to:10.42.187.188:500
    0     0 DNAT       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp dpt:500 ADDRTYPE match dst-type LOCAL t
    0     0 MARK       udp  --  !docker0 *       0.0.0.0/0            0.0.0.0/0           udp dpt:4500 MARK set 0x1068
    0     0 DNAT       udp  --  !docker0 *       0.0.0.0/0            0.0.0.0/0           udp dpt:4500 to:10.42.187.188:4500
    0     0 DNAT       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp dpt:4500 ADDRTYPE match dst-type LOCAL
    0     0 MARK       tcp  --  !docker0 *       0.0.0.0/0            172.31.16.111       tcp dpt:18088 MARK set 0x1068
    0     0 DNAT       tcp  --  !docker0 *       0.0.0.0/0            172.31.16.111       tcp dpt:18088 to:10.42.228.5:18088
    0     0 DNAT       tcp  --  *      *       0.0.0.0/0            172.31.16.111       tcp dpt:18088 to:10.42.228.5:18088
   23  1436 MARK       tcp  --  !docker0 *       0.0.0.0/0            172.31.20.45        tcp dpt:18088 MARK set 0x1068
   23  1436 DNAT       tcp  --  !docker0 *       0.0.0.0/0            172.31.20.45        tcp dpt:18088 to:10.42.75.49:18088
    0     0 DNAT       tcp  --  *      *       0.0.0.0/0            172.31.20.45        tcp dpt:18088 to:10.42.75.49:18088
```

Exercise 1

- Running Rancher Server

Exercise 1

- Prepare 3 hosts: 2vCPU, 4GB memory, 50GB disk, Ubuntu 16.04 or Centos7+ preferred.
- Install docker engine in all 3VMs

  curl -sSL https://releases.rancher.com/install-docker/1.12.sh | sh
  systemctl enable docker
  systemctl start docker

- Run rancher/server in VM1

  mkdir -p /rancher/db
  docker run -d --restart=unless-stopped -v /rancher/db:/var/lib/mysql -p 8080:8080 rancher/server:v1.6.14

- Check rancher/server container startup logs

  docker logs -f *id_of_your_container*
- Check processes running inside rancher/server container

  docker exec -it *id_of_your_container* bash

  ps -auxww
- Open access to VM1 port 8080 if there is firewall.
- Access http://*ip_of_VM1*:8080 to open Rancher UI

Exercise 2

- Add Hosts
- Deploy Services

Exercise 2

- Add (1-2) hosts
- Create a stack
- Create a service
- Scale it
- Delete a container
- Create a service with 2 containers and distribute it to 2 hosts (scheduling rule)
- Enable access control, create account with different membership roles, login to check the difference.