# Rancher Training

## Session 4

RANCHER LABS

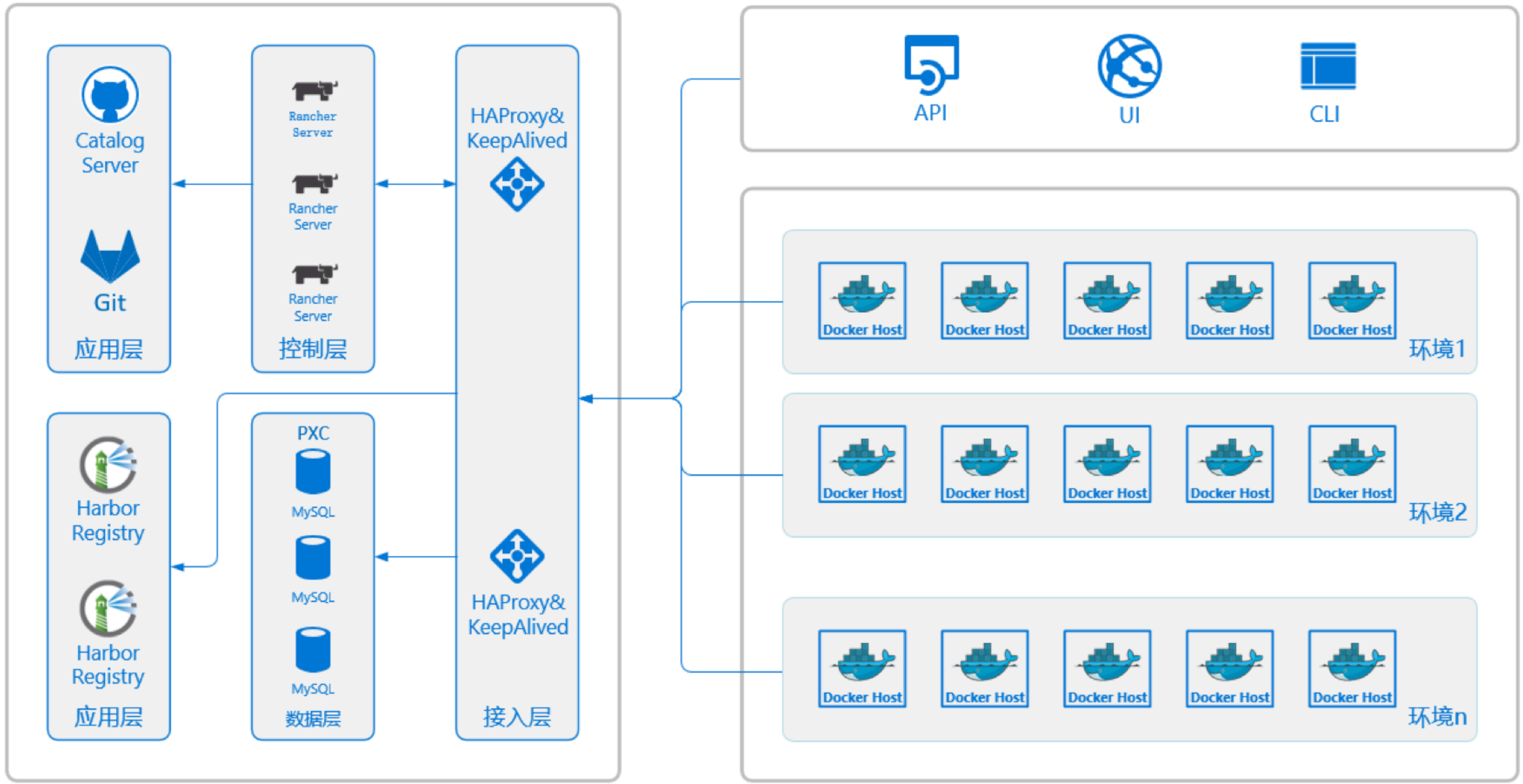
Arvin • 赵雨川

# Afternoon Agenda

- Deployment best practice
- Upgrade consideration
- Rancher Pipeline
- Monitoring & Logging
- Webhook
- Rancher 2.0
- Rancher Wiki and Roadmap
- Common issues FAQ

# Deployment best practice

- Access Layer
  Keepalived+Haproxy  2 Nodes
- Controller Layer
  Rancher Server HA(Hazelcast)  3 Nodes
- Data Layer
  Mysql/Miaradb+PXC Cluster  3Nodes
- Application Layer
  Gitlib Harbor HA


Catalog Server
Git
应用层
Rancher Server
Rancher Server
Rancher Server
控制层
HAProxy& KeepAlived
HAProxy& KeepAlived
接入层
Harbor Registry
Harbor Registry
应用层
PXC
MySQL
MySQL
MySQL
数据层
API
UI
CLI
Docker Host
环境1
环境2
环境n

# Access Layer

- Rancher Server With SSL
  http://rancher.com/docs/rancher/v1.6/en/installing-rancher/installing-server/basic-ssl-config/

- Keepalived+Haproxy
  Haproxy log level

- Add ip_vs model
  yum install ipvsadm popt-devel openssl-devel libnl* -y kernel-devel
  cat >/etc/sysconfig/modules/ip_vs.modules<<EOF
  modprobe ip_vs
  EOF
  chmod 755 /etc/sysconfig/modules/ip_vs.modules

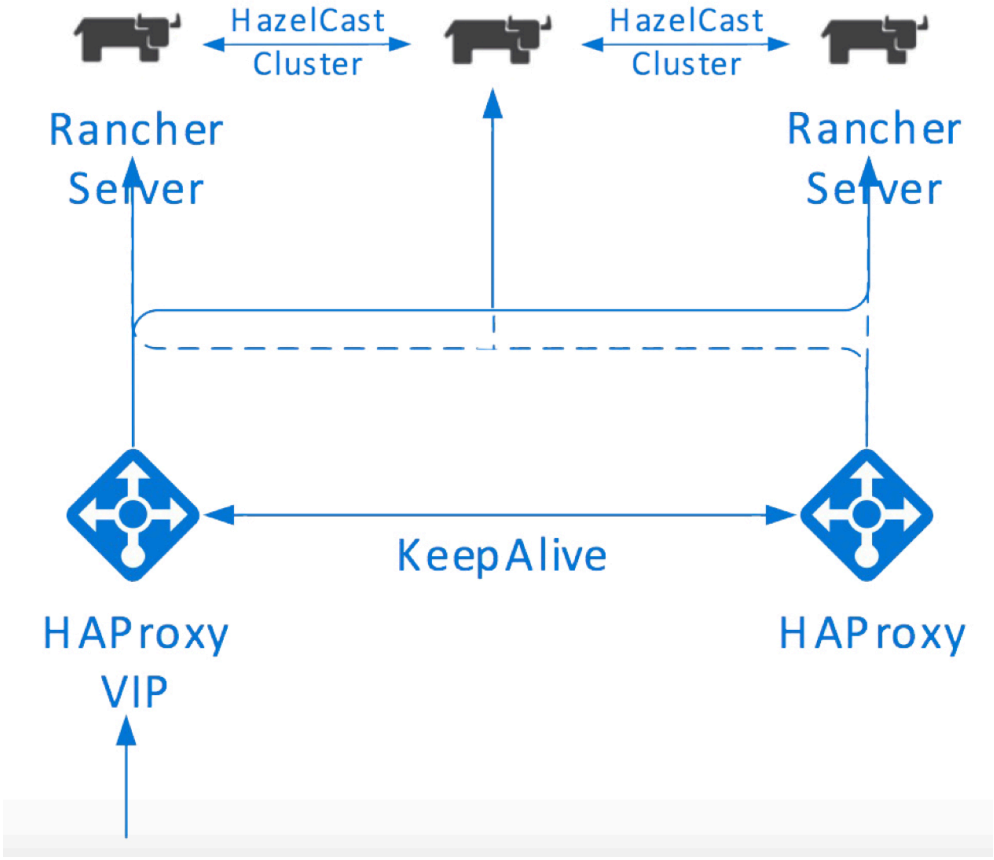
# Controller Layer

- HA Nodes:

  ROM: 16 GB or 8GB at least
  Ports : 9345, 8080
  LB: Haproxy or F5

- DB

  PXC Cluster or Galera




```
docker run -d --restart=unless-stopped -p 8080:8080 -p 9345:9345 rancher/server \
--db-host myhost.example.com --db-port 3306 --db-user username --db-pass password --db-name cattle \
--advertise-address <IP_of_the_Node>
```

# Data Layer

- Etcd or Zookeeper

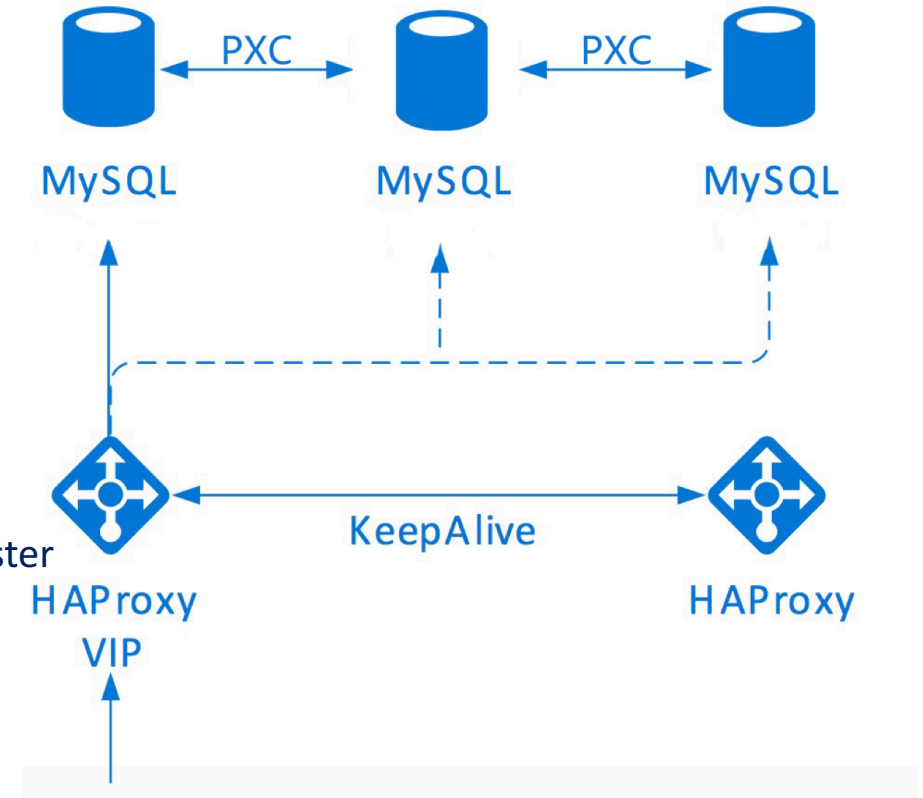
- Commod

  docker run --net=host -d -e MYSQL_ROOT_PASSWORD=123456 \

  -e CLUSTER_NAME=rancher -e XTRABACKUP_PASSWORD=123456 \

  -e DISCOVERY_SERVICE=42.62.83.5:2379 –e  percona/percona-xtradb-cluster

- SET GLOBAL innodb_file_format=DYNAMIC

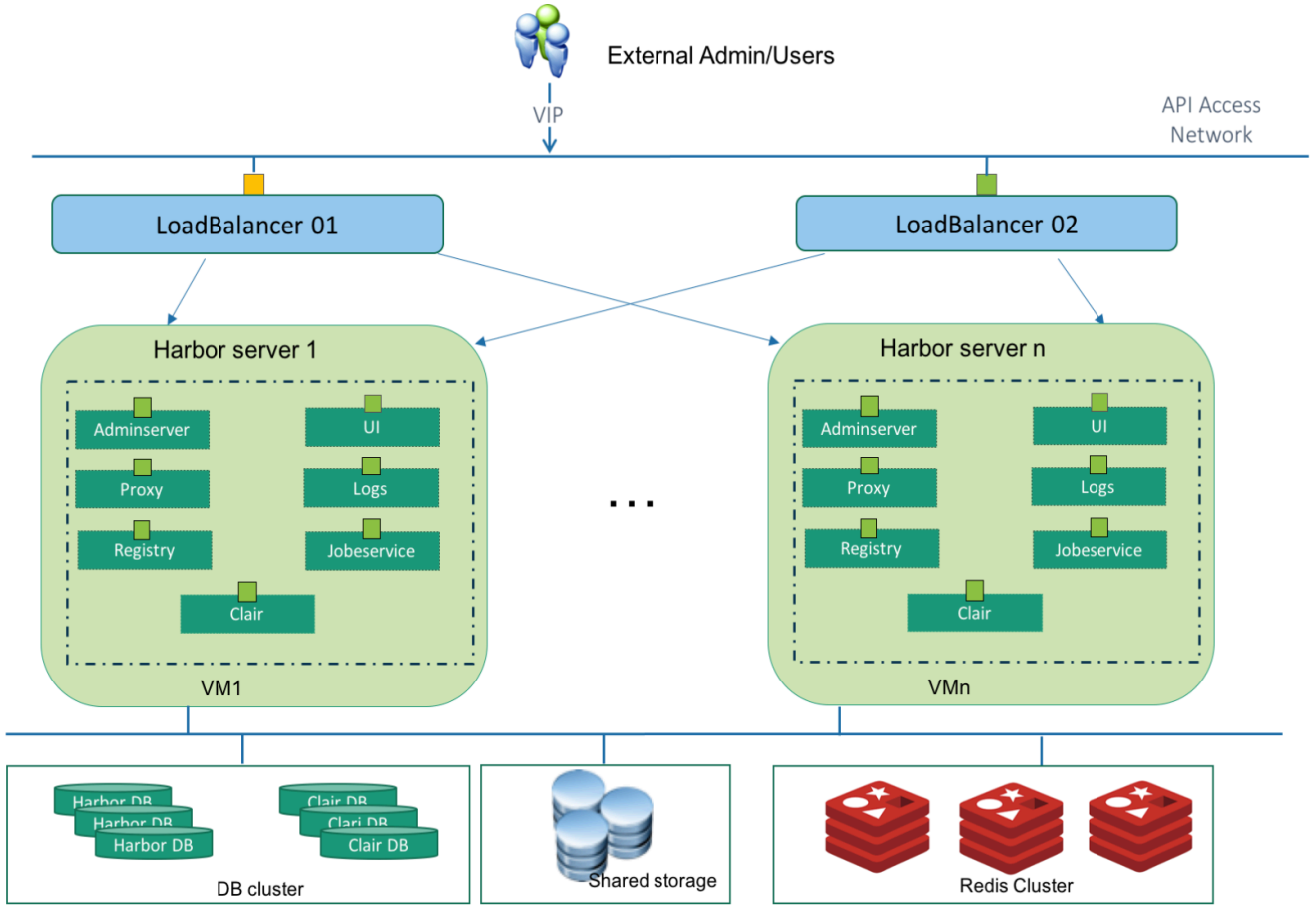


# Application Layer

Shared storage : Glusterfs Ceph

Redis Cluster

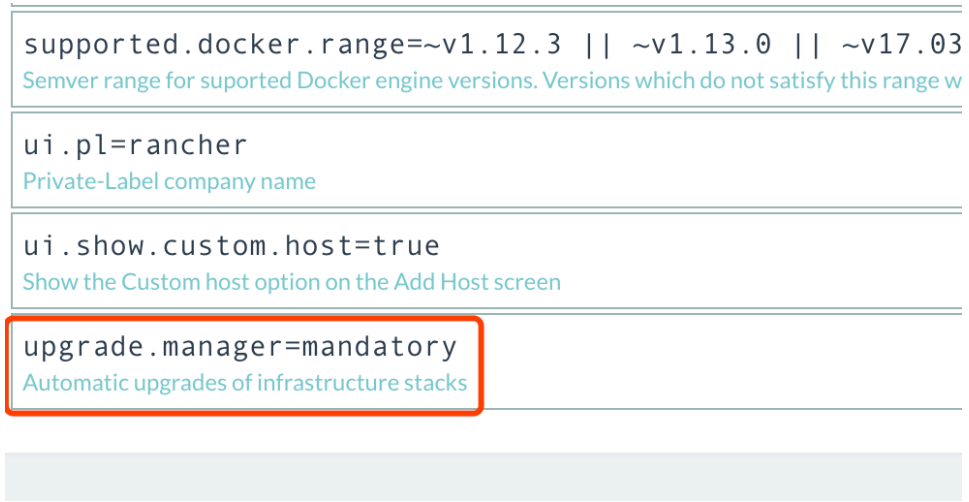
Haproxy Keepalived



# Upgrade consideration

- Infrastructure service upgrade order:
  1. network-policy-manager (if installed, this is an optional component in Rancher)
  2. network-services
  3. ipsec
  4. remainder of the infrastructure stacks
  5. Service upgrade API setting:
     Upgrade.manager=mandatory (Server version v1.6.1+)

- Rancher HA server node upgrade:

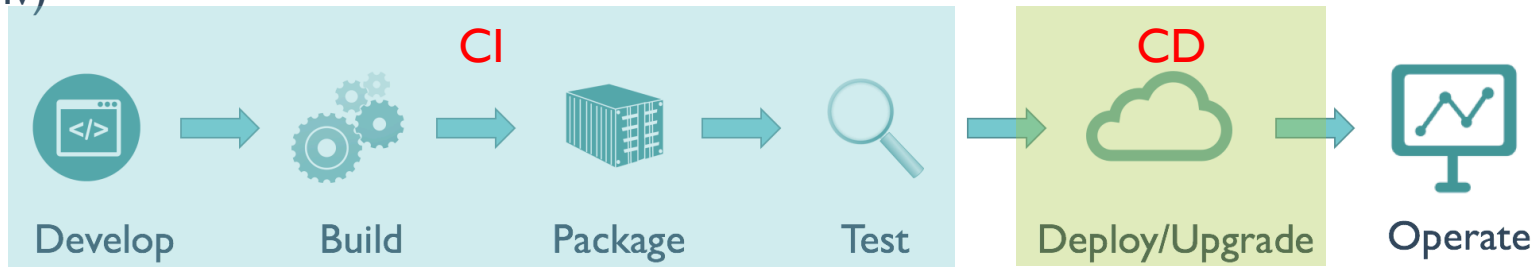


# Rancher Pipeline

- CICD Overview
- Rancher Pipeline demo
- Rancher Pipeline features

# CICD Overview

- **continuous integration** (**CI**) is the practice of merging all developer working copies to a shared mainline several times a day, includes:
  - Develop
  - Build
  - Package
  - Test
- **Continuous delivery** (**CD**) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time, which includes:
  - Deploy (to Test env or PRD env)
  - Upgrade




GitHub

Rancher Pipeline

Rancher CI service

HARBOR™

RANCHER

amazon web services™

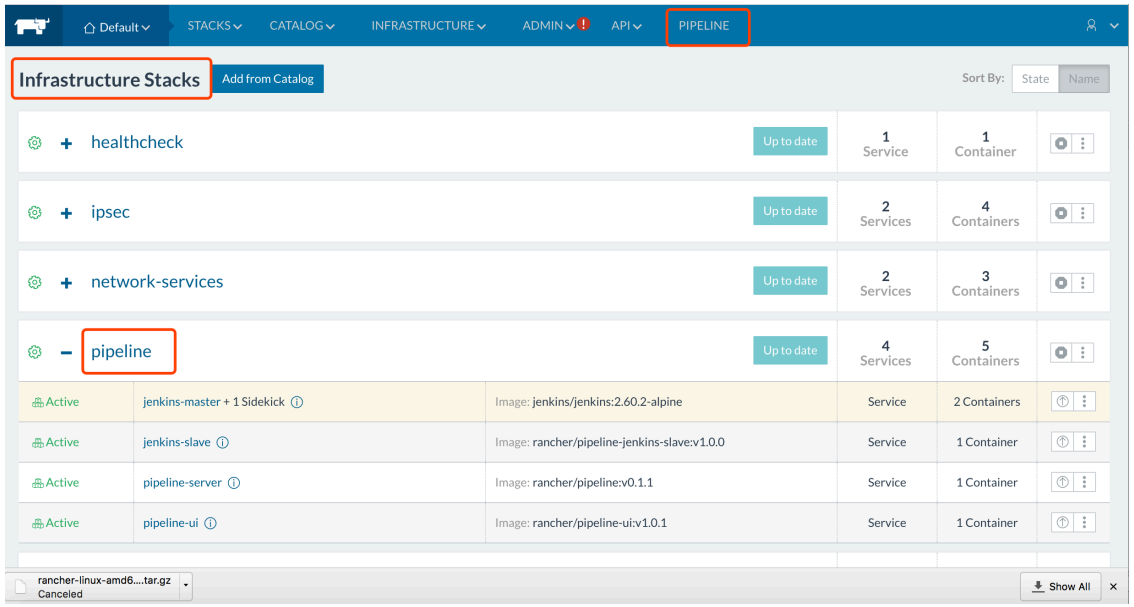
vmware®

# Pipeline Demo

- Deploy Pipeline form Catalog

  Rancher Server Version >= 1.6.13

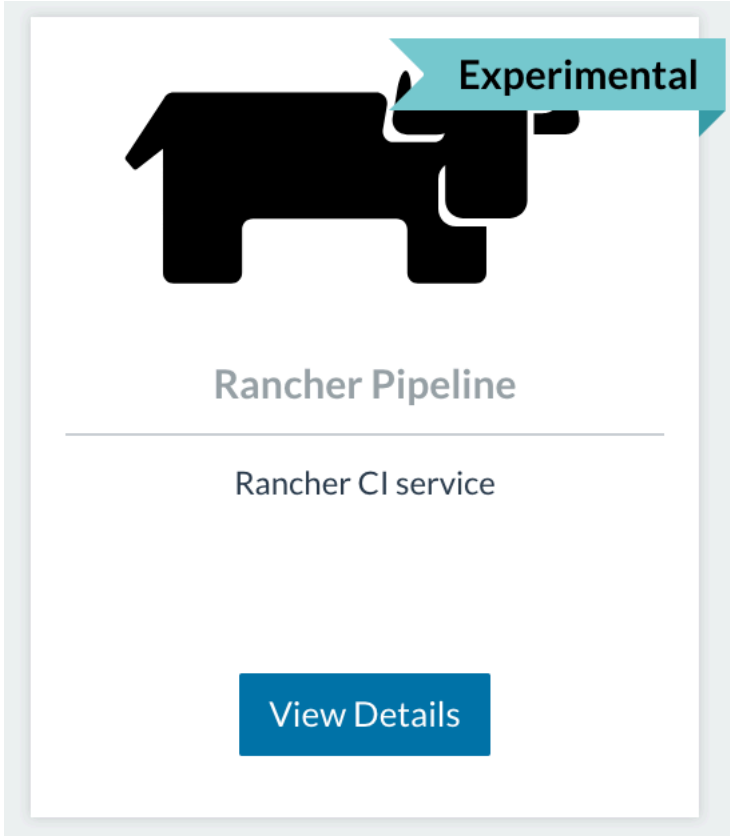
- Demo



# Rancher Pipeline

Features:

1. Support multi source code management
2. Consistent user experience
3. Support approval workflow
4. Support Scheduler
5. Compose variables
6. Integrate with Rancher (Rancher compose, registry)

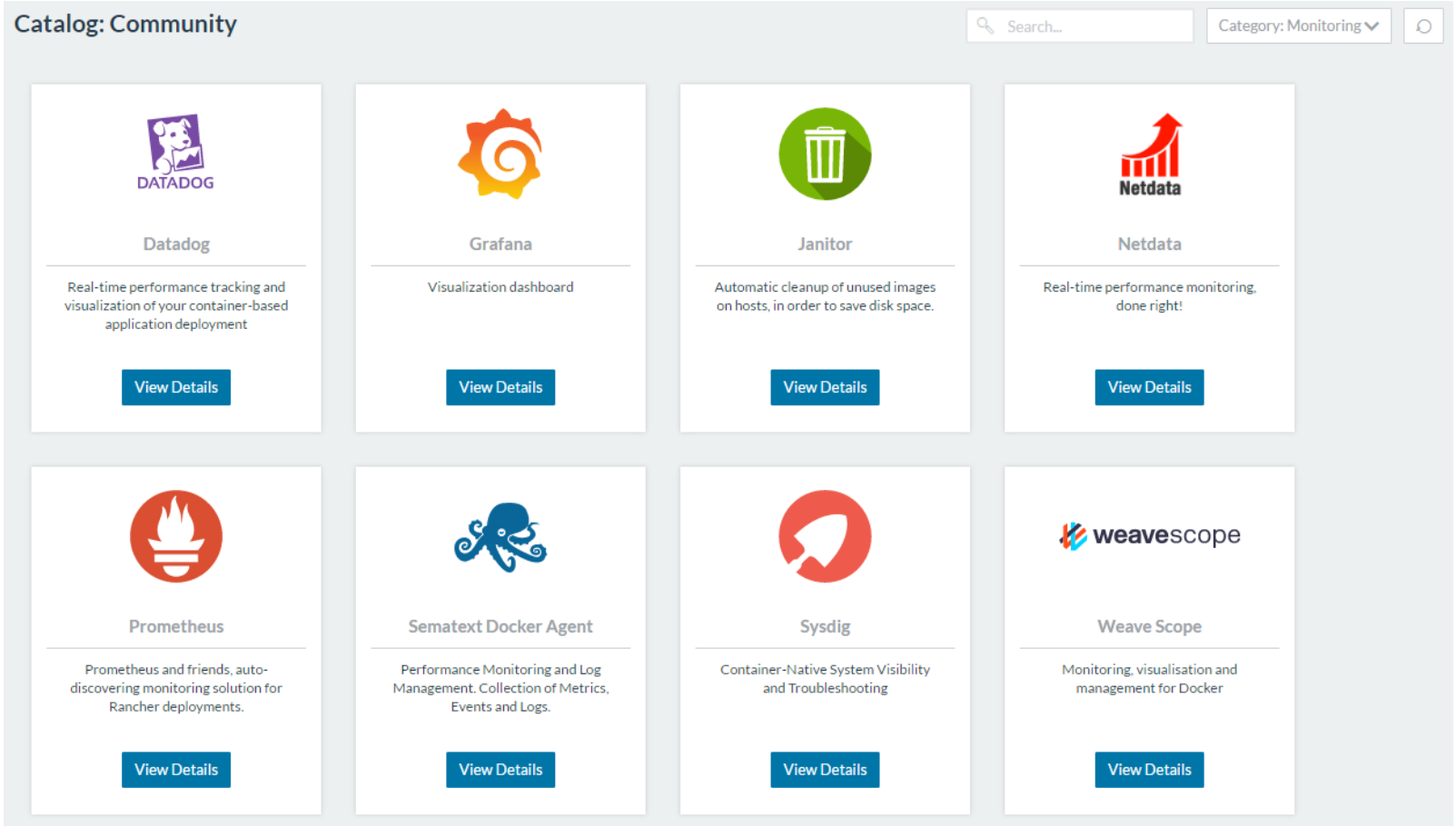


# Monitoring & Logging

- Docker Monitoring
- Monitoring in Rancher
- Webhook
- Logging

# Docker Monitoring

- Many choices
- Docker Stats
- Cadvisor
- Sysdig
- Datadog
- Sematext
- Prometheus
- ……

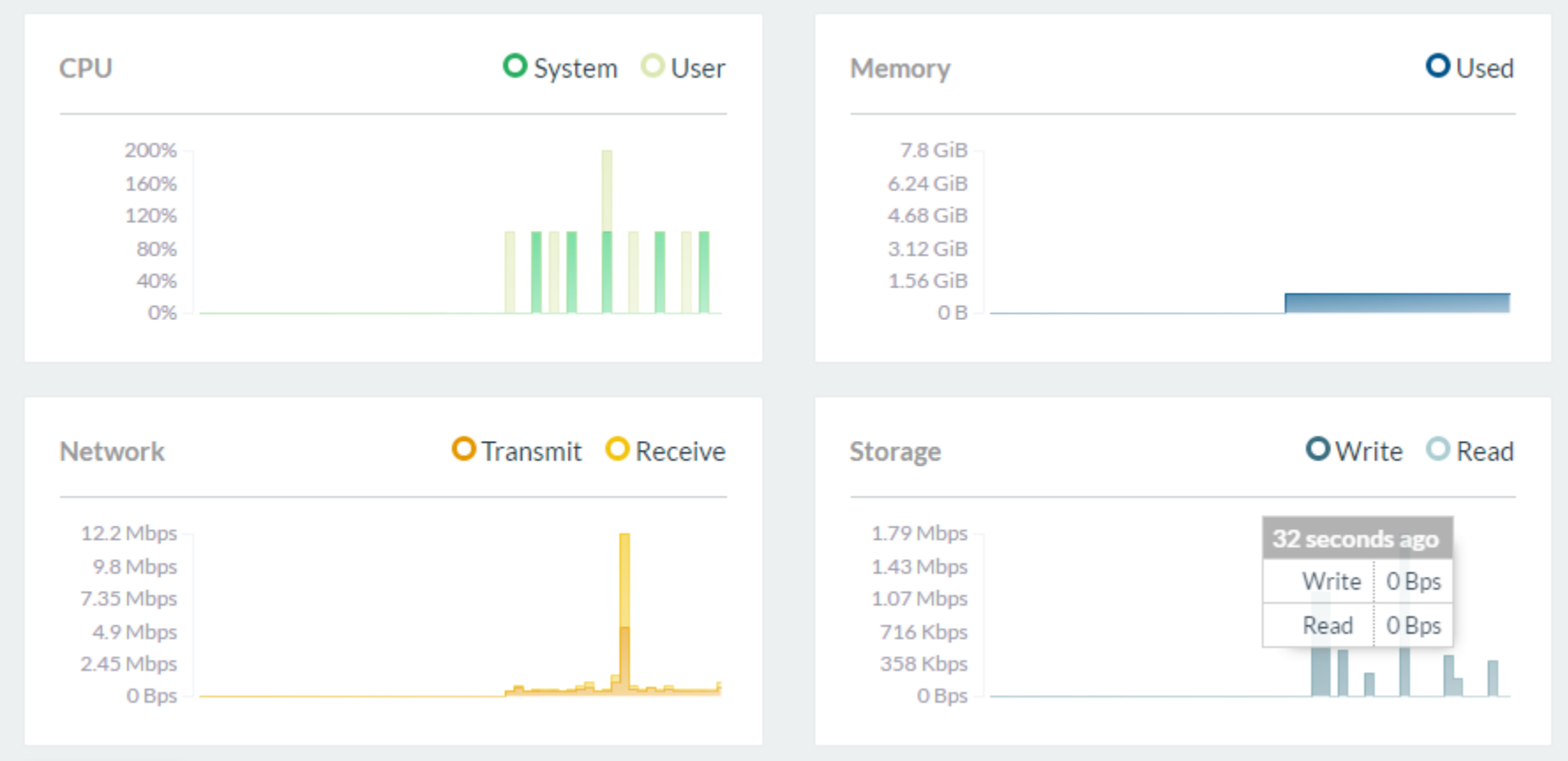



http://rancher.com/tag/container-monitoring/ ,
http://rancher.com/comparing-monitoring-options-for-docker-deployments/

# Rancher Monitoring

- Hosts/Stacks/Services/Containers
- Rancher Server

# Default Monitoring in Rancher



# Prometheus

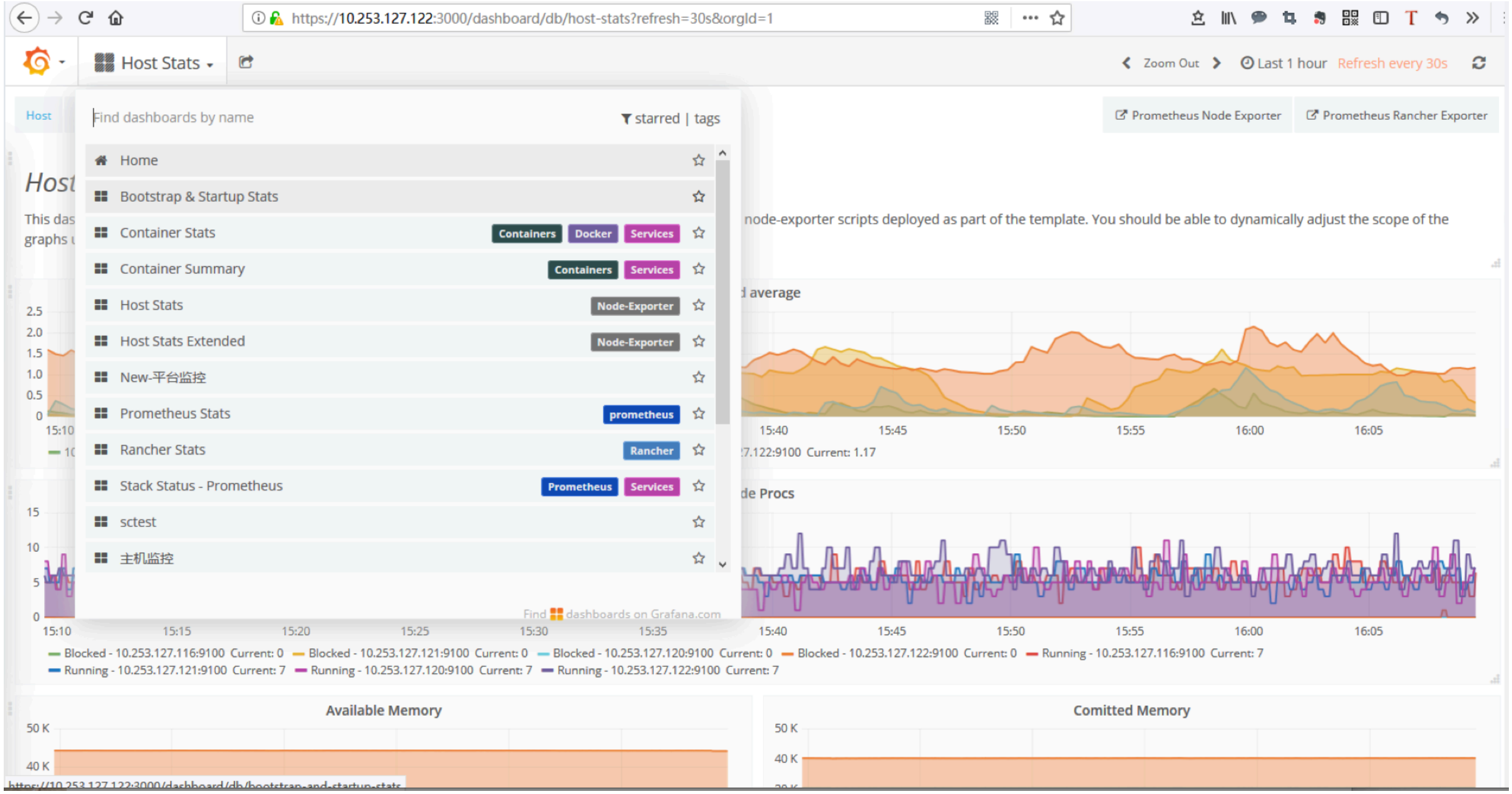
## •Enabling Prometheus metrics for the Rancher Server

- Set the environment CATTLE_PROMETHEUS_EXPORTER=true for the Rancher server container.
- Exposeport on the container as such -p 9108:9108

- docker run -d –restart=unless-stopped -e CATTLE_PROMETHEUS_EXPORTER=true -p 8080:8080 -p 9108:9108 rancher/server

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

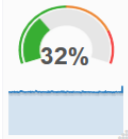
- Prometheus - Used to scrape and store metrics from our data sources.
- Prometheus Node Exporter - Gets host level metrics and exposes them to Prometheus.
- cAdvisor - Deploys and Exposes the cadvsior stats used by Rancher's agent container, to Prometheus.
- Grafana - Used to visualise the data from Prometheus and InfluxDB.
- Prometheus Rancher Exporter - Allows Prometheus to access the Rancher API and return the status of any stack or service in the rancher environment associated with the API key used.

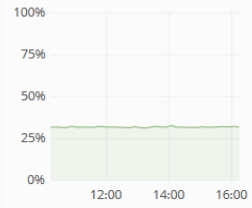

https://10.253.127.122:3000/dashboard/db/host-stats?refresh=30s&orgId=1
Host Stats
Zoom Out
Last 1 hour
Refresh every 30s
Prometheus Node Exporter
Prometheus Rancher Exporter
Find dashboards by name
starred | tags
Home
Bootstrap & Startup Stats
Container Stats
Containers
Docker
Services
Container Summary
Containers
Services
Host Stats
Node-Exporter
Host Stats Extended
Node-Exporter
New-平台监控
Prometheus Stats
prometheus
Rancher Stats
Rancher
Stack Status - Prometheus
Prometheus
Services
sctest
主机监控
Find dashboards on Grafana.com
node-exporter scripts deployed as part of the template. You should be able to dynamically adjust the scope of the
average
de Procs
Blocked - 10.253.127.116:9100 Current: 0
Blocked - 10.253.127.121:9100 Current: 0
Blocked - 10.253.127.120:9100 Current: 0
Blocked - 10.253.127.122:9100 Current: 0
Running - 10.253.127.116:9100 Current: 7
Running - 10.253.127.121:9100 Current: 7
Running - 10.253.127.120:9100 Current: 7
Running - 10.253.127.122:9100 Current: 7
Available Memory
Comitted Memory

主机监控 | Zoom Out | Last 6 hours | Refresh every 30s

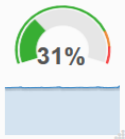
环境： Management | 主机： 10.253.127.122 | interval 10m

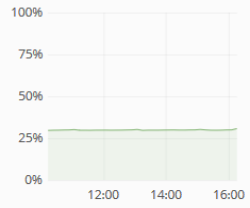
## 内核数
4Cpu

## CPU使用率


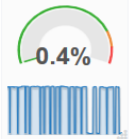

## CPU使用率（%）


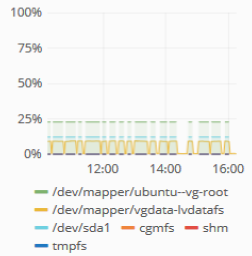

## 内存容量
16G

## 内存使用率



## 内存使用率（%）



## 硬盘容量
196G

## 硬盘使用率



## 硬盘使用率（%）



## CPU使用率(%)

| Metric | Current |
|---|---|
| 服务：logging-services/logging-agent 容器：r-logging-services-logging-agent-1-677037ec | 25.01 |
| 服务：monitoring-services/cadvisor 容器：r-monitoring-services-cadvisor-1-d705fb03 | 2.19 |
| 服务：container_label_io_rancher_stack_service_name 容器：name | 1.73 |
| 服务：kibana-test/kibana-test 容器：r-kibana-test-kibana-test-1-c7e0bbf4 | 0.07 |
| 服务：network-services/network-manager 容器：r-network-services-network-manager-1-ad40d56f | 0.05 |

1 2 3 4

## 内存使用率（%）

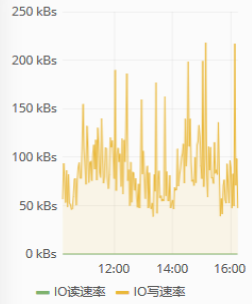
| Metric | Current |
|---|---|
| 服务：logging-services/logging-agent 容器：r-logging-services-logging-agent-1-677037ec | 10.08 |
| 服务：container_label_io_rancher_stack_service_name 容器：name | 2.15 |
| 服务：monitoring-services/cadvisor 容器：r-monitoring-services-cadvisor-1-d705fb03 | 0.75 |
| 服务：kibana-test/kibana-test 容器：r-kibana-test-kibana-test-1-c7e0bbf4 | 0.72 |
| 服务：network-services/network-manager 容器：r-network-services-network-manager-1-ad40d56f | 0.64 |

1 2 3 4

## 硬盘使用率（%）

| Metric | Current |
|---|---|
| 容器：name;device:/dev/mapper/ubuntu--vg-root | 22.72 |
| 容器：name;device:/dev/sda1 | 12.11 |
| 容器：name;device:/dev/mapper/vgdata-lvdatafs | 8.70 |
| 容器：r-monitoring-services-rancher-health-exporter-1-13d6cad8;device:/dev/mapper/vgdata-lvdatafs | 0.11 |
| 容器：r-Prometheus-monitoring-manager-1-19aea498;device:/dev/mapper/vgdata-lvdatafs | 0.09 |

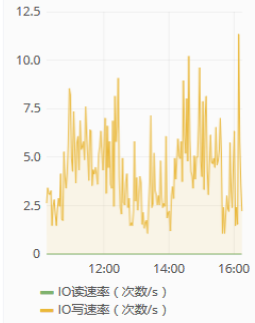
1 2 3 4 5 6 7

## IO操作总速率



## disk-IO操作速率

| Metric | Current |
|---|---|
| 写速率-容器：name | 11.74 kBs |
| 写速率-容器：r-Prometheus-alertmanager-1-f103e889 | 0.17 kBs |
| 写速率-容器：r-network-services-network-manager-1-ad40d56f | 0 kBs |
| 写速率-容器：r-network-services-metadata-dns-1-c4f15230 | 0 kBs |
| 写速率-容器：r-monitoring-services-node-exporter-1-d086359c | 0 kBs |

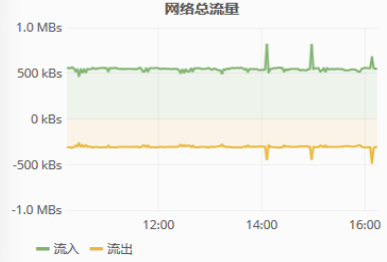
1 2 3 4 5

## disk-IO操作总速率（次数/s）



## disk-IO操作速率（次数/s）

| Metric | Current |
|---|---|
| 写速率-容器：name | 1.87 |
| 写速率-容器：r-Prometheus-alertmanager-1-f103e889 | 0.02 |
| 写速率-容器：r-rancherintercptor-rancherinterceptor-1-89745e41 | 0 |
| 写速率-容器：r-network-services-network-manager-1-ad40d56f | 0 |
| 写速率-容器：r-network-services-metadata-dns-1-c4f15230 | 0 |

1 2 3 4 5 6 7

## 网络总流量


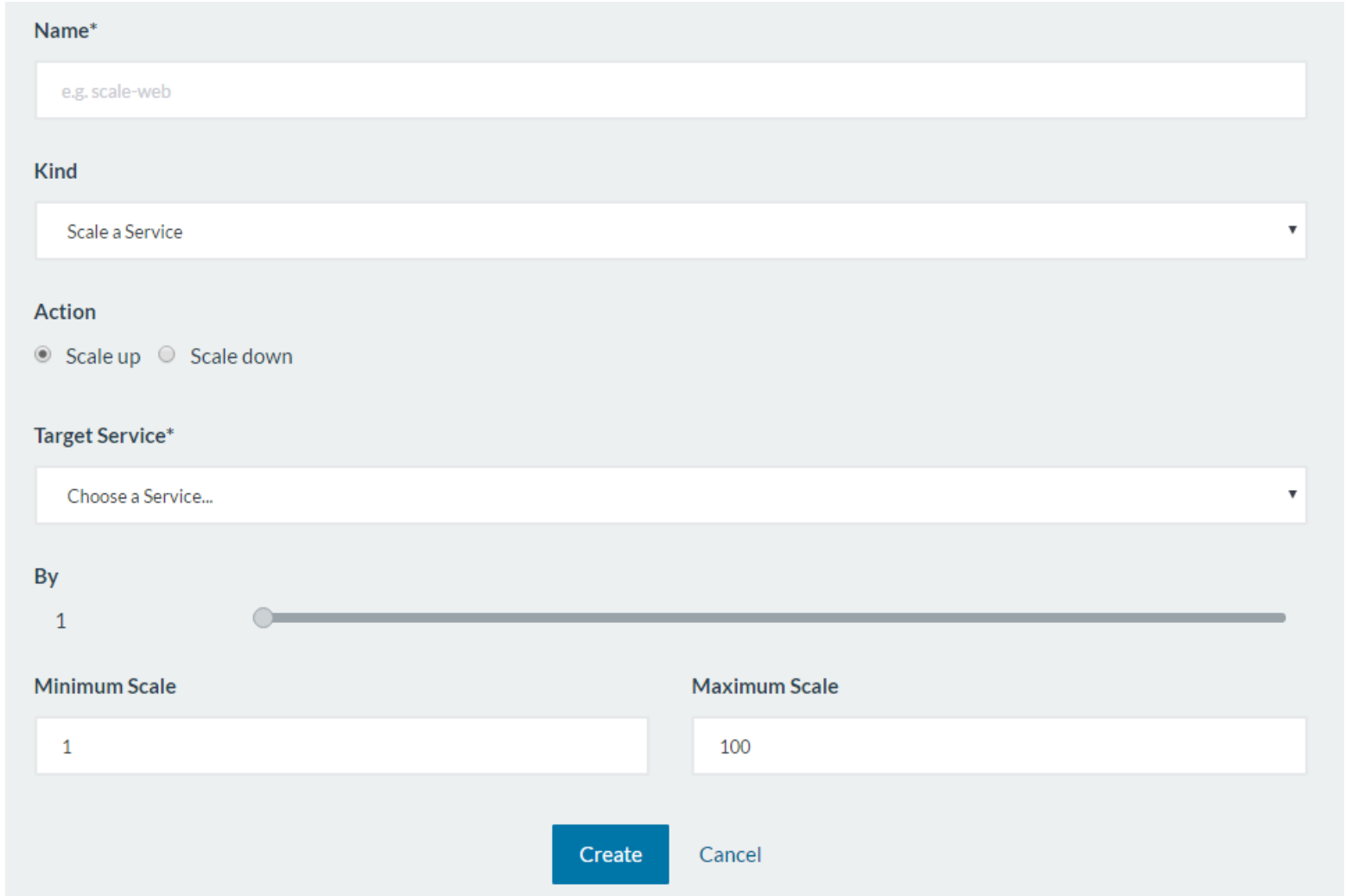

## 网络流量

| Metric | Current |
|---|---|
| 流出-容器：r-monitoring-services-node-exporter-1-d086359c | 130.67 kBs |
| 流出-容器：r-ipsec-ipsec-1-94610607 | 129.94 kBs |
| 流出-容器：r-Prometheus-prometheus-1-14672836 | 128.60 kBs |
| 流入-容器：r-ipsec-ipsec-1-94610607 | 123.97 kBs |
| 流出-容器：r-monitoring-services-cadvisor-1-d705fb03 | 123.93 kBs |

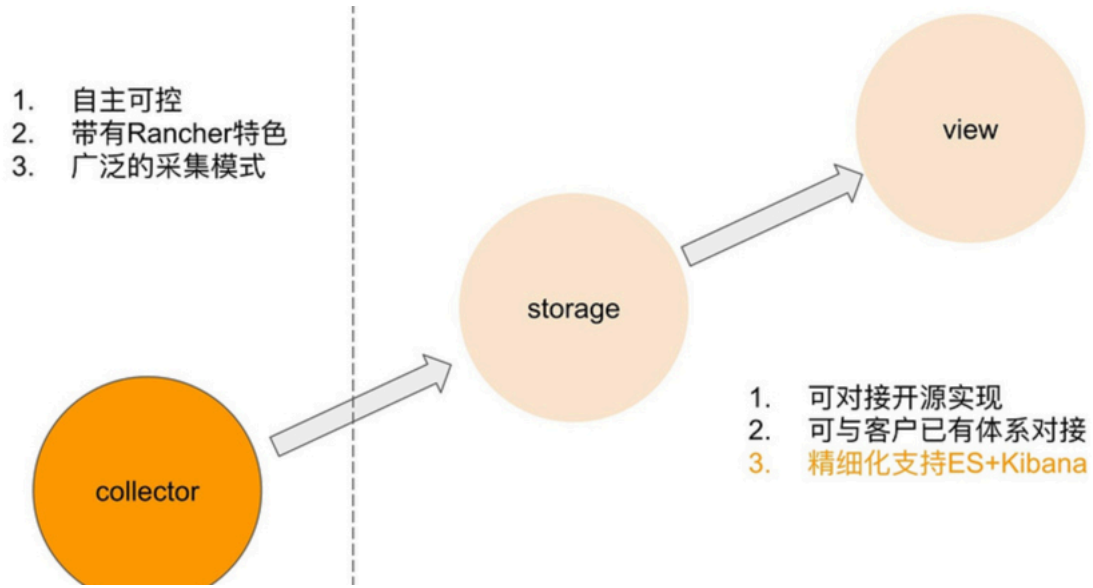
1 2 3 4 5 6 7 8 9

## 网络异常情况（包数/s）


# Webhook


Name*
e.g. scale-web
Kind
Scale a Service
Action
Scale up
Scale down
Target Service*
Choose a Service...
By
1
Minimum Scale
1
Maximum Scale
100
Create
Cancel

# Logging




kibana

Kibana 4

Visualization dashboard

View Details

Elasticsearch Cluster 5.5.1

Elasticsearch, you know for search!

View Details

fluentd

Rancher Logging

Rancher Logging Collector

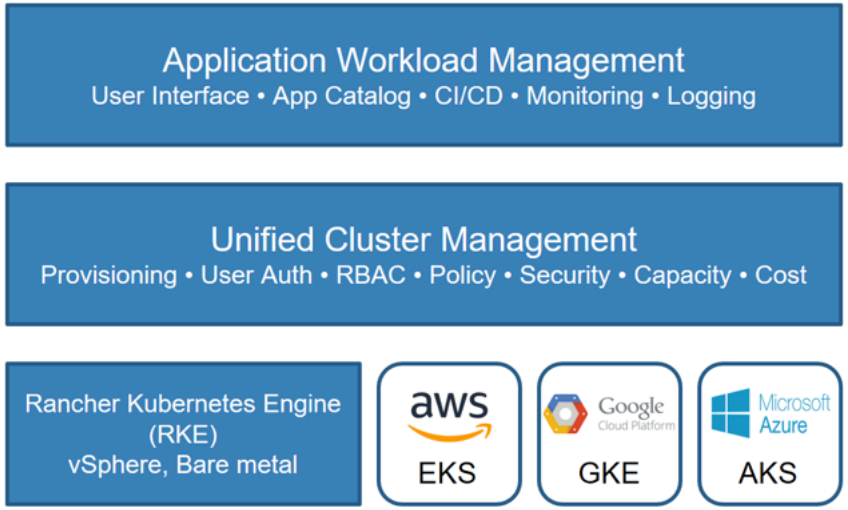
Already Deployed

# Rancher 2.0

- Simplified K8s deployment for running on-prem
- Simplified UI, real-time data stream (no need to refresh like Kube dashboard)
- Management of multiple K8s clusters, even those not created by Rancher (GKE, ACS, on-prem)
- Catalog in Rancher 2.0 has been expanded to support both compose templates and Helm templates
- Rancher 2.0 will additionally include a managed CI/CD service
- Rancher 2.0 include a managed Prometheus service for out-of-the-box monitoring

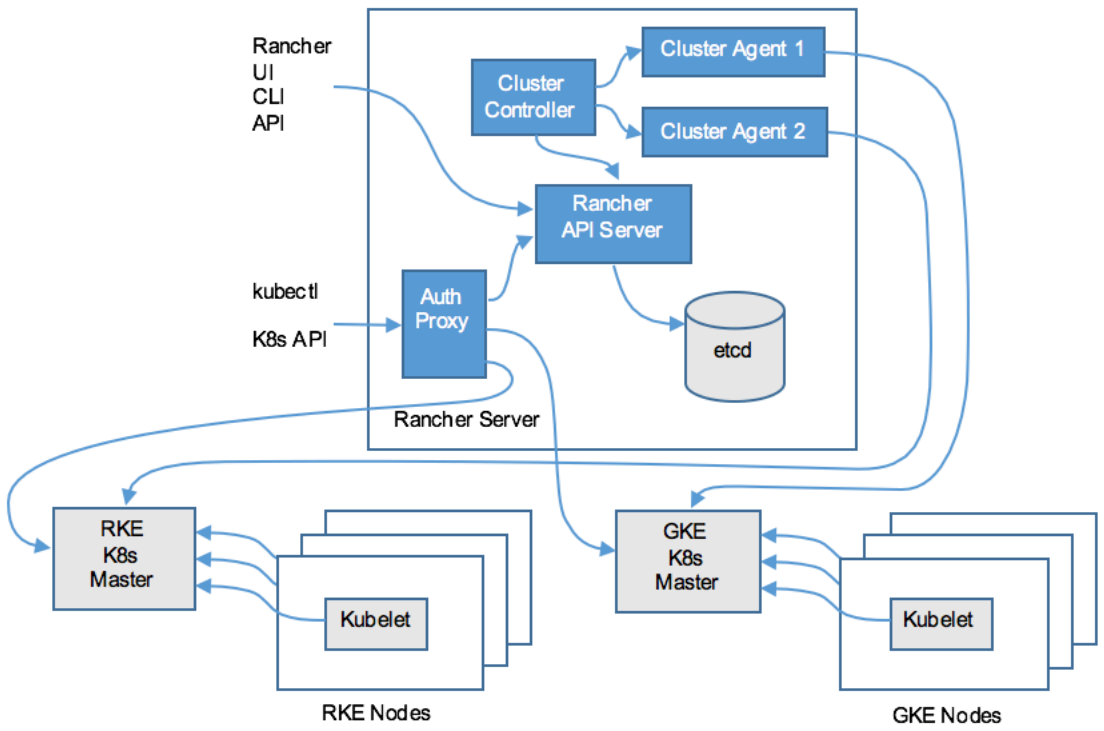
RANCHER 2.0

COMPLETE CONTAINER MANAGEMENT PLATFORM



# Rancher 2.0 Architecture

- Rancher API Server
- Cluster Controller and Agents
- Auth Proxy
- High Availability



# Rancher Wiki and Project Plan

- https://github.com/rancher/rancher/wiki

# Common issues FAQ

https://www.cnrancher.com/common-troubleshooting-and-repair-methods/

http://rancher.com/docs/rancher/v1.6/zh/faqs/troubleshooting/

# Exercise 6

- Deploy Prometheus
- Autoscale with webhook

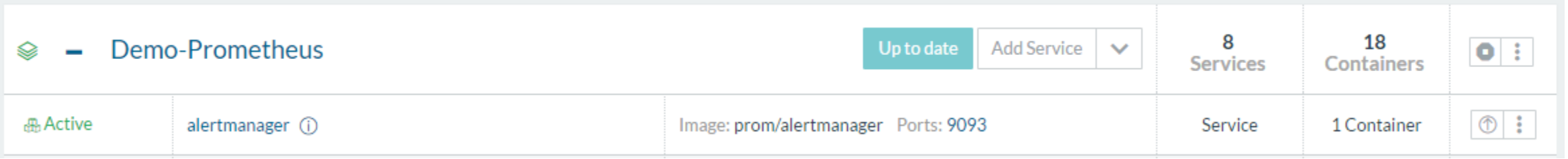
# Rancher 2.0

- Deploy Prometheus
- Autoscale with webhook

# Exercise 6

1. Deploy Prometheus from Catalog
2. Deploy cAdvisor from Catalog
3. Access Prometheus and Grafana
4. Check Rancher Server Monitoring

# 5. Add alertmanager in Prometheus stack




# 6. Create webhook receiver for a test service, set action to scale up.

http://docs.rancher.com/rancher/v1.6/en/cattle/webhook-service/

# 7. Create alert rule in Prometheus

```
ALERT CpuUsageSpike
IF rate(container_cpu_user_seconds_total{container_label_io_rancher_container_name="Your_container_name"}[30s]) * 100 > 20
LABELS {
  severity="critical",
  action="up"
}
ANNOTATIONS {
  summary = "ADDITIONAL CONTAINERS NEEDED",
  description = "CPU usage is above 70%"
}
```

# 8. Create actions in alertmanager

```
# Autoscale test route
routes:
- match:
    action: up
  receiver: "webhook-receiver-up"
```

```
- name: "webhook-receiver-up"
  webhook_configs:
  - url: http://rancherapps.com:8080/v1-webhooks/endpoint?token=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb2
    send_resolved: true
```

# 9. Simulate high in container and check if autoscale success

**dd if=/dev/zero of=/dev/null**

# Thanks