目录

[一、准备基础环境配置 2](#_Toc16871)

[1. 主机名配置 2](#_Toc2258)

[2. CentOS关闭selinux 2](#_Toc5772)

[3. 关闭防火墙(可选)或者放行相应端口 2](#_Toc31195)

[4. 更新系统 4](#_Toc20140)

[5. 同步时间、时区 5](#_Toc10094)

[6. Kernel性能调优 5](#_Toc23862)

[7. 载入内核模块 5](#_Toc27298)

[8. 安并配置docker 6](#_Toc10566)

[二、安装rancher webUI 6](#_Toc7956)

[1. 运行rancher webUI容器 6](#_Toc13608)

[2. 登入rancher webUI 6](#_Toc9827)

[三、增加RKE方式部署的自定义K8S集群 7](#_Toc10878)

[四、 发布流程 10](#_Toc58)

[1. [Jenkins]pull[gitlab]代码 10](#_Toc23693)

[2. [Jenkins]执行build构建镜像 10](#_Toc25578)

[3. [Jenkins]push镜像到[harbor] 10](#_Toc6656)

[4. [jenkins]执行kubectl更新[rancher]项目pod的镜像 10](#_Toc28587)

[五、 项目部署 10](#_Toc12697)

[0. 项目结构说明 10](#_Toc25329)

[1. 添加项目 11](#_Toc27057)

[2. 进入项目页面 11](#_Toc17543)

[3. 点击部署服务 11](#_Toc26743)

[4. 更具需求部署基础服务 11](#_Toc6722)

[5. 增加服务配置，选择配置映射 11](#_Toc15170)

[6. 根据需求配置键值对 11](#_Toc18315)

[7. 再次进入工作负载，点击服务升级，进入配置页面，增加配置映射 12](#_Toc23938)

一、准备基础环境配置

1. 主机名配置

因为K8S的规定，主机名只支持包含 - 和 .(中横线和点)两种特殊符号，并且主机名不能出现重复。

hostnamectl set-hostname web.rancher

hostnamectl set-hostname node1.rancher

hostnamectl set-hostname node2.rancher

hostnamectl set-hostname node3.rancher

vim /etc/hosts

web.rancher 58.82.247.30

node1.rancher 58.82.224.107

node2.rancher 154.211.127.232

node3.rancher 154.211.138.59

1. CentOS关闭selinux

sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config

setenforce 0

1. 关闭防火墙(可选)或者放行相应端口

前期部署可以选择防火墙关闭

systemctl stop firewalld.service && systemctl disable firewalld.service

yum install iptables-services -y

systemctl enable iptables

vim /etc/sysconfig/iptables 清空规则

systemctl start iptables.service

设置防火墙

Rancher webUI

安装ipset

ipset-libs-7.1-1.el7.x86\_64

ipset-service-7.1-1.el7.noarch

ipset-7.1-1.el7.x86\_64

ipset-devel-7.1-1.el7.x86\_64

vim /etc/sysconfig/ipset-config

修改IPSET\_SAVE\_ON\_STOP="yes"

创建ip组

ipset create rancher-web hash:net

把需要放开的IP添加到该组

ipset add rancher-web xx.xx.xx.xx

重启保存

systemctl restart ipset

配置iptables 80和443端口的规则

iptables -I DOCKER-USER -p tcp -m set ! --match-set rancher-web src -m multiport --dport 80,443 -j DROP

其他端口如法炮制，一定明白各个服务器上服务调用关系

| **从 / 到** | **Rancher 节点** | **etcd 节点** | **Control 节点** | **Worker 节点** | **外部负载均衡** | **互联网** |
| --- | --- | --- | --- | --- | --- | --- |
| Rancher 节点 (1) |  |  |  |  |  | git.rancher.io (2): 35.160.43.145:32 35.167.242.46:32 52.33.59.17:32 |
| etcd 节点 | 443 TCP (3) | 2379 TCP |  |  | 443 TCP |  |
| 2380 TCP |  |  |  |
|  | 6443 TCP |  |  |
| 8472 UDP | | |  |
| 4789 UDP (7) | | |  |
| 9099 TCP (4) |  |  |  |  |
| Control Plane 节点 | 443 TCP (3) | 2379 TCP |  |  | 443 TCP |  |
| 2380 TCP |  |  |  |
|  | 6443 TCP |  |  |
| 8472 UDP | | |  |
| 4789 UDP (7) | | |  |
| 10250 TCP | | |  |
|  | 9099 TCP (4) |  |  |
|  | 10254 TCP (4) |  |  |  |
| Worker 节点 | 443 TCP (3) |  | 6443 TCP |  | 443 TCP |  |
| 8472 UDP | | |  |
| 4789 UDP (7) | | |  |
|  |  | 9099 TCP (4) |  |
|  |  | 10254 TCP (4) |  |  |
| 外部负载均衡 (5) | 80 TCP |  |  |  |  |  |
| 443 TCP (6) |  |  |  |  |  |
| API / UI 客户端 | 80 TCP (3) |  |  |  | 80 TCP |  |
| 443 TCP (3) |  |  |  | 443 TCP |  |
| 工作负载客户端 |  |  |  | 30000-32767 TCP / UDP (nodeport) |  |  |
|  |  | 80 TCP (Ingress) | |  |  |
|  |  | 443 TCP (Ingress) | |  |  |
| 注意：  1. 运行Rancher的单节点或者Rancher高可用的节点。 2. 需要用来拉取Rancher应用商店库。 3. 适用于不使用外部负载均衡的情况。 4. 本地流量（非跨节点流量）。 5. 负责Rancher UI / API的负载均衡或反向代理。 6. 仅适用于SSL没有在外部负载均衡终止。 7. 仅适用于 Windows 集群中使用 Overlay 网络模式。 | | | | | | |

| **协议** | **端口** | **描述** |
| --- | --- | --- |
| TCP | 22 | 使用主机驱动通过 SSH 进行节点配置 |
| TCP | 2376 | 主机驱动与 Docker 守护进程通信的 TLS 端口 |
| TCP | 2379 | etcd 客户端请求 |
| TCP | 2380 | etcd 节点通信 |
| UDP | 8472 | Canal/Flannel VXLAN overlay 网络 |
| UDP | 4789 | Windows 集群中 Flannel VXLAN overlay 网络 |
| TCP | 9099 | Canal/Flannel 健康检查 |
| TCP | 6783 | Weave 端口 |
| UDP | 6783-6784 | Weave UDP 端口 |
| TCP | 10250 | kubelet API |
| TCP | 10254 | Ingress controller 健康检查 |
| TCP/UDP | 30000-32767 | NodePort 端口范围 |

1. 更新系统

yum update -y

1. 同步时间、时区

timedatectl set-timezone Asia/Shanghai

1. Kernel性能调优

cat >> /etc/sysctl.conf<<EOF

net.ipv4.ip\_forward=1

net.ipv4.neigh.default.gc\_thresh1=4096

net.ipv4.neigh.default.gc\_thresh2=6144

net.ipv4.neigh.default.gc\_thresh3=8192

EOF

数值根据实际环境自行配置，最后执行sysctl -p保存配置。

1. 载入内核模块

mods=(br\_netfilter

ip6\_udp\_tunnel

ip\_set

ip\_set\_hash\_ip

ip\_set\_hash\_net

iptable\_filter

iptable\_nat

iptable\_mangle

iptable\_raw

nf\_conntrack\_netlink

nf\_conntrack

nf\_conntrack\_ipv4

nf\_defrag\_ipv4

nf\_nat

nf\_nat\_ipv4

nf\_nat\_masquerade\_ipv4

nfnetlink

udp\_tunnel

veth

vxlan

x\_tables

xt\_addrtype

xt\_conntrack

xt\_comment

xt\_mark

xt\_multiport

xt\_nat

xt\_recent

xt\_set

xt\_statistic

xt\_tcpudp

)

for i in ${mods[@]}

do

modprobe $i

done

1. 安并配置docker

curl -fsSL https://get.docker.com -o get-docker.sh

sh get-docker.sh

systemctl enable docker

systemctl start docker

二、安装rancher webUI

1. 运行rancher webUI容器

docker run -d --restart=unless-stopped \

--name rancher-web \

-p 80:80 -p 443:443 \

-v /home/rancher:/var/lib/rancher/ \

-v /home/var/log/auditlog:/var/log/auditlog \

-e CATTLE\_SYSTEM\_CATALOG=bundled \

-e AUDIT\_LEVEL=3 \

-e TZ="Asia/Shanghai" \

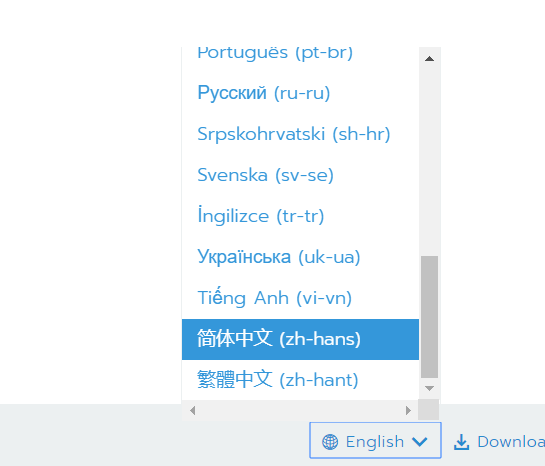
rancher/rancher:latest

1. 登入rancher webUI

浏览器打开[https://58.82.247.30/](https://58.82.247.30/update-password)

首次安装需要设置管理员密码

右下角可以设置为简体中文



登入后的页面

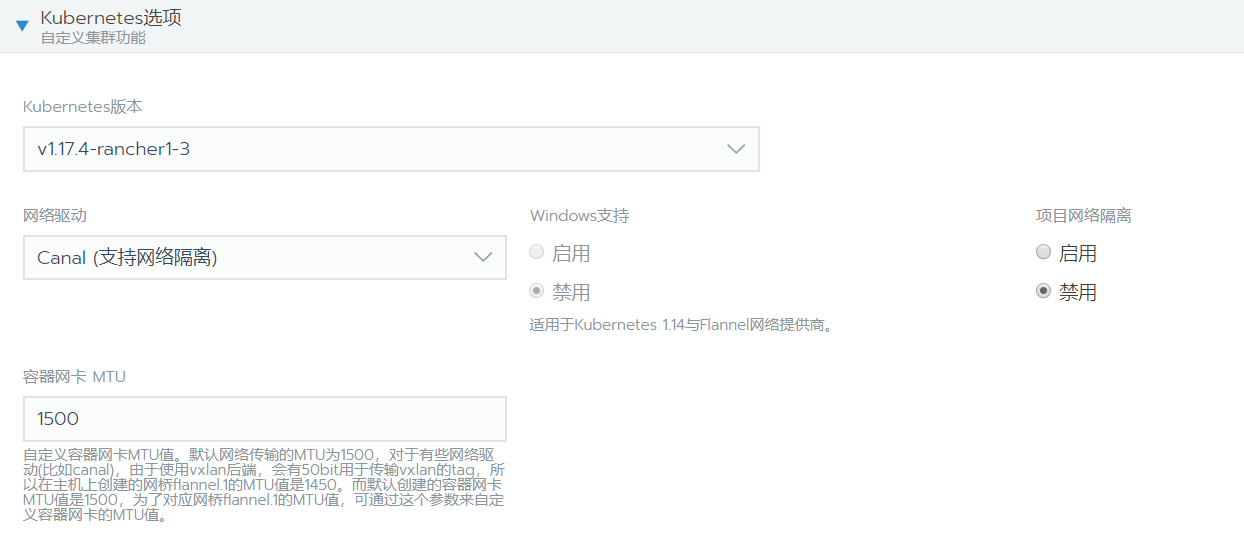


三、增加RKE方式部署的自定义K8S集群





这里配置使用v1.17.4版本的K8S，后期可以零宕机升级K8S版本。



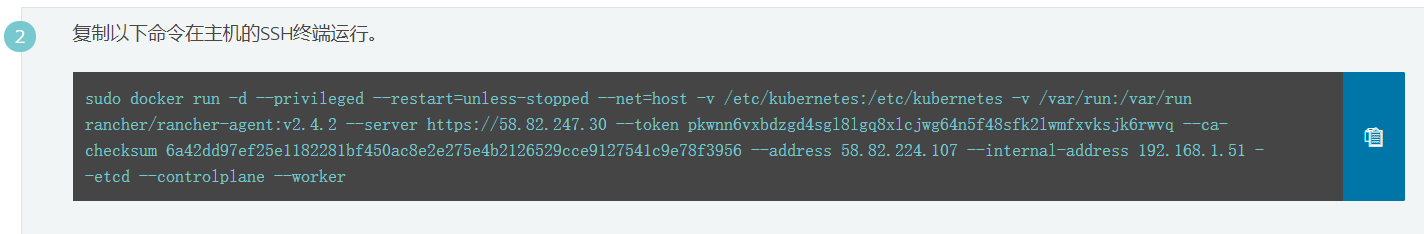
其他参数默认即可，点下一步。



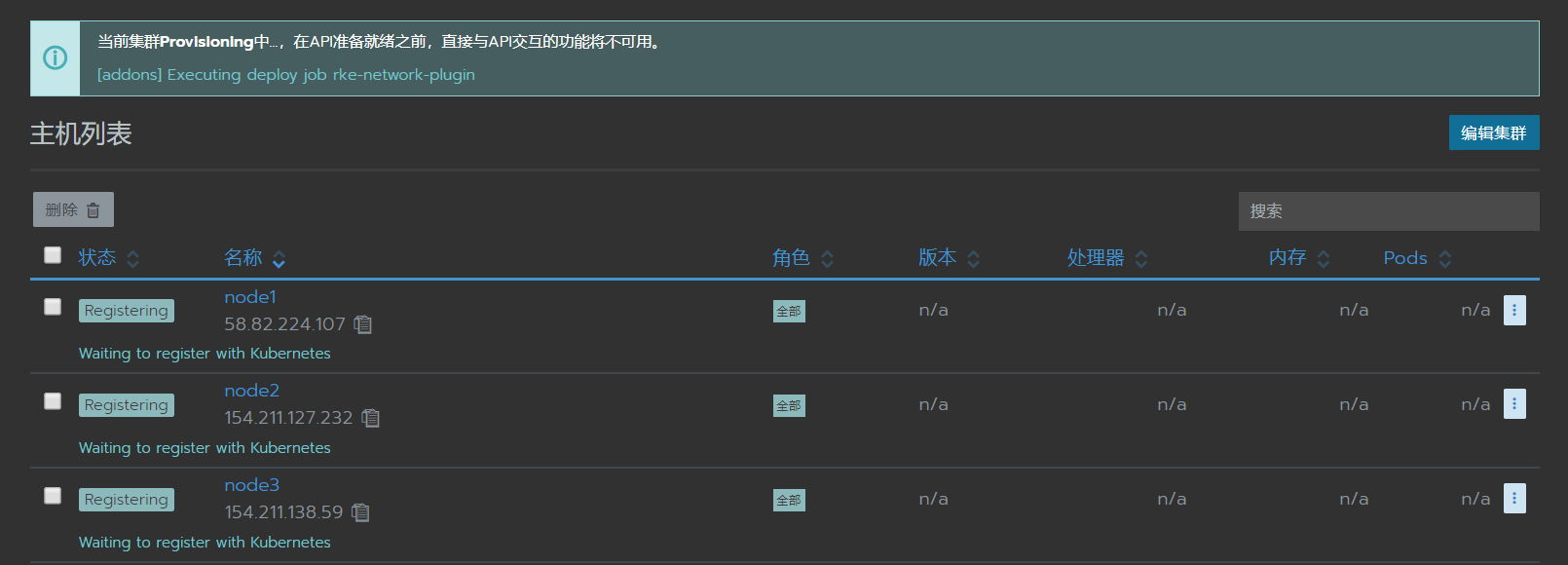
这里需要配置主机角色，我们另外3台节点服务器是复用的，因此3个角色全部勾选。后期增加节点时指用勾选Worker角色即可。公网地址和内网地址节点名称如果未填写，系统会自动识别，但如果是vps的云服务器，是无法识别外网地址的，这种情况下是必须填写的。



填写完一台的信息后这里会自动生产命令，去对应的服务器执行即可，服务器会自动安装RKE集群。其他主机如法炮制。

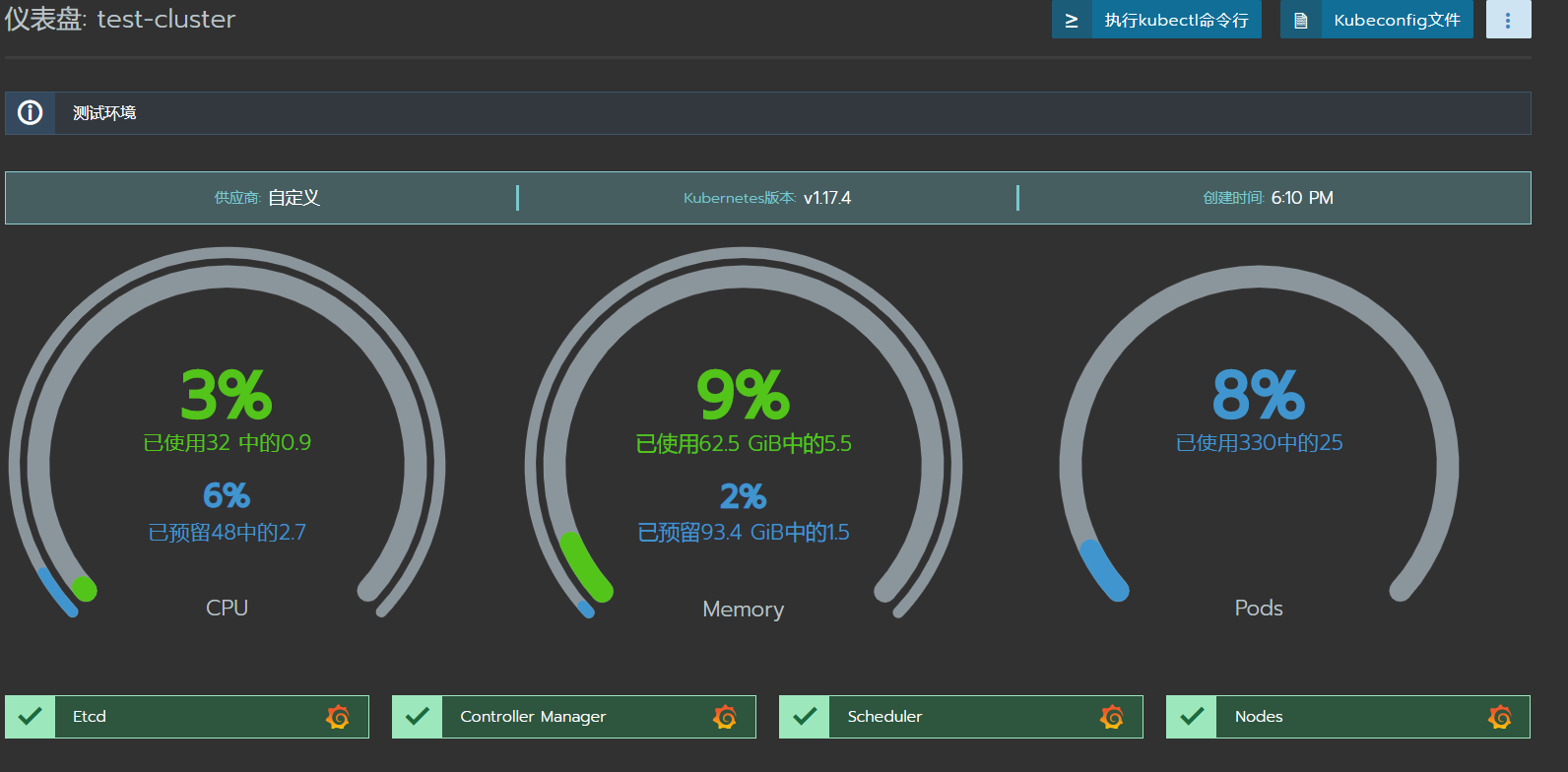


全部添加后，点击完成，等待服务部署完成即可。



完成后的状态



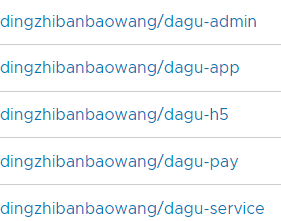
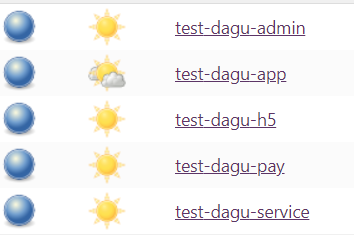


后期添加节点或升级集群都可以点击右上角的编辑集群进入配置页面。

1. 发布流程
2. [Jenkins]pull[gitlab]代码
3. [Jenkins]执行build构建镜像
4. [Jenkins]push镜像到[harbor]
5. [jenkins]执行kubectl更新[rancher]项目pod的镜像

图例：

gitlab代码库 Jenkins jobs harbor 镜像 rancher pod



1. 项目部署
2. 项目结构说明

集群：对应K8S集群，rancher web ui 可以管理多个K8Sjiqun，集群之间网络隔离

主机：对应K8S节点，集群中包含服务器，主机分两个角色

worker[运行业务容器的节点]

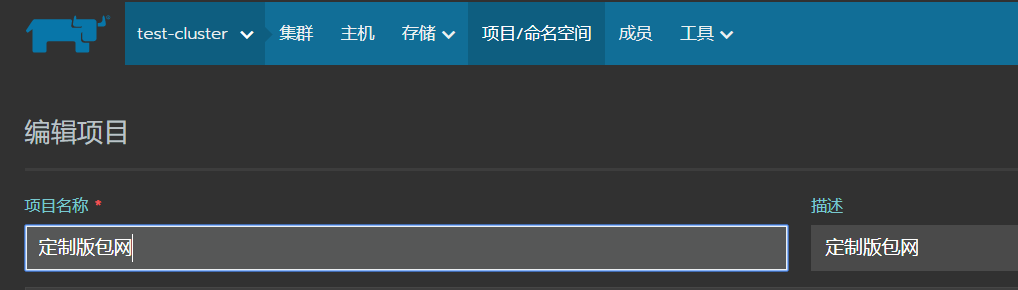
master[运行平台核心容器的节点]

项目：rancher平台的抽象空间，实现项目隔离

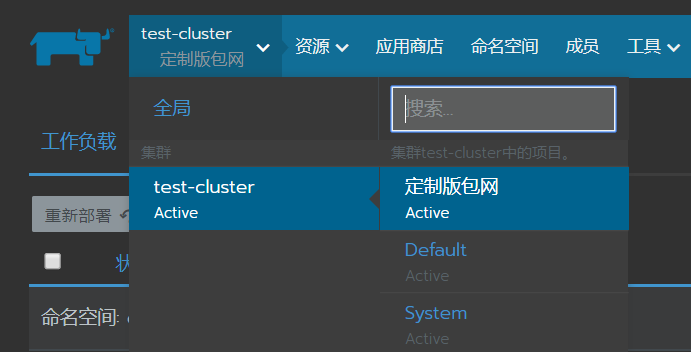
命名空间：对应K8S命名空间，包含不同的业务POD

POD：对应K8S的POD，运行业务容器的集合，多个容器共享网络

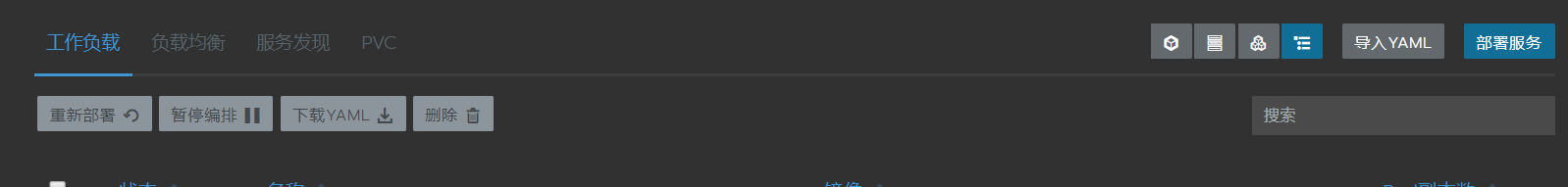
1. 添加项目



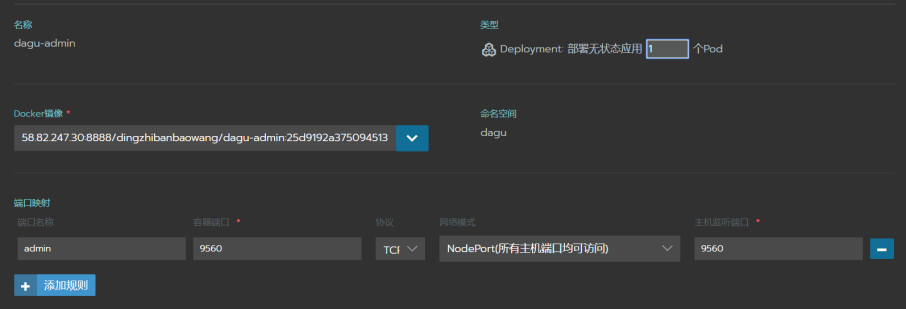
1. 进入项目页面



1. 点击部署服务



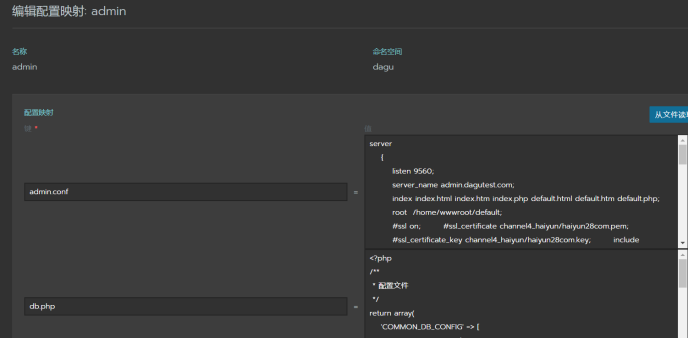
1. 更具需求部署基础服务



1. 增加服务配置，选择配置映射

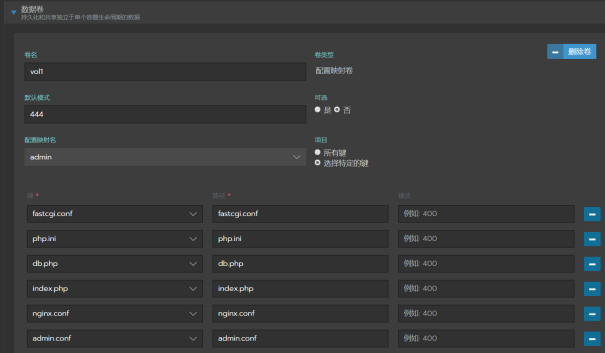


1. 根据需求配置键值对



1. 再次进入工作负载，点击服务升级，进入配置页面，增加配置映射







至此项目应用部署完成，后期代码更新，使用Jenkins来构建镜像，自动更新服务镜像。