1、阿里云(帐号信息,VPC 网络、SLB、SS、docker,RDS,redis,mongodb,DNAT,SNAT)
ECS 帐号： d1@123feng.com r@3;lo9Uj45a

1、阿里云翻墙服务器与 IKE vpn 服务器
i-j6c1u9ywulp7cksa4385 47.75.124.126(公) 帐号 root 密码 feng,2018HZ
10.86.32.132

2、kafak zk 服务器 --docker 部署
i-j6cdf3eqjjp127tbtm3l 47.75.144.70 帐号 root 密码 feng,2018HZ
10.86.32.130

3、conflunce 服务器
i-bp1ae0wrz90vxn5xyi3k 10.86.122.243 帐号 root 密码 cisco#123

4、无线控制器
10.0.10.3---AC
admin nipc#123

telnet huawei huawei@123

5、交换机
10.0.10.2 ---SW
admin huawei@123

6、路由器
10.0.0.1 —-路由器
admin admin

7、日志服务器

i-j6cazj9xjvk1dtkklhf5 10.86.32.128 帐号 root 密码 feng,2018HZ

8、消息服务器

i-j6c0mc4nea9j1falzdny 10.86.31.229 帐号 root 密码 xgyq1h(5S-v,iT-!

9、 帐号 root 密码 feng,2018HZ
master 服务器
i-j6cgszxf4xakzmjl1gjw 10.86.31.241
i-j6c5nvuk3jmtyzl01x8m 10.86.31.240
i-j6c0z8lpzknn10fecehs 10.86.31.239

node 服务器
i-j6cezvecfb1x2dr95orh 10.86.31.243
i-j6cezvecfb1x2dr95ori 10.86.31.242

10、邮件服务器
i-bp1eaa5819tkjp91ij2t 47.96.90.104
10.86.121.11 root cisco#123

11、数据库服务器
i-bp11vnxvfen71b1i6g5v 10.86.120.208 root cisco#123

12、RDS
rm-j6cr19qp1fg0vmh89
公网 rm-j6cr19qp1fg0vmh89po.mysql.rds.aliyuncs.com
内网 rm-j6cr19qp1fg0vmh89.mysql.rds.aliyuncs.com

12.1、帐号：root 密码： g6hfuV1kAy#u
数据库—-》
deliveryhz 读写
deliveryil 读写
deliveryryd 读写
deliverytw 读写
expresshz 读写
expressil 读写
expressryd 读写
expresstw 读写

12.2、帐号：wadmin 密码： Wad#2018Ak

数据库—》windbase

13、 monodb
dds-j6c3a25c29f2e774

内网：**dds-j6c3a25c29f2e7741.mongodb.rds.aliyuncs.com:3717**
dds-j6c3a25c29f2e7742.mongodb.rds.aliyuncs.com:3717

公网：**dds-j6c3a25c29f2e7741403-pub.mongodb.rds.aliyuncs.com:3717**
dds-j6c3a25c29f2e7742280-pub.mongodb.rds.aliyuncs.com:3717

帐号 **root** 密码 **fen_Q8SuwE**

风力帐号：**devopsdba** 风力密码 **iq7n*b2gs**

风信帐号：**windadmin 风信密码 wadmin_2019**

14、 redis

r-j6c0cd7b361b7134
内网： r-j6c0cd7b361b7134.redis.rds.aliyuncs.com
风力密码：Gomrwind123
风信密码：Gomrwind123

15、

10.0.0.210

root MIPC#123HBS

10.0.0.211

root MIPC#123HBS

一、阿里云翻墙服务器与 IKE vpn 服务器

1.1、内核优化 （翻墙&&ike2 VPN 服务器）

```
[root@izj6c1u9ywulp7cksa4385z ~]# cat /etc/sysctl.conf
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
vm.swappiness = 0
net.ipv4.neigh.default.gc_stale_time=120
net.ipv4.ip_forward = 1
# see details in https://help.aliyun.com/knowledge_detail/39428.html
net.ipv4.conf.all.rp_filter=0
net.ipv4.conf.default.rp_filter=0
net.ipv4.conf.default.arp_announce = 2
net.ipv4.conf.lo.arp_announce=2
net.ipv4.conf.all.arp_announce=2
# see details in https://help.aliyun.com/knowledge_detail/41334.html
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_keepalive_time = 1200
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_tw_reuse = 1
#net.ipv4.tcp_tw_recycle = 1
net.ipv4.ip_local_port_range = 10000 65000
net.ipv4.tcp_max_syn_backlog = 8192
net.ipv4.tcp_max_tw_buckets = 5000
```

```
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding=1
##减少超时前的探测次数
net.ipv4.tcp_keepalive_probes=5
##优化网络设备接收队列
net.core.netdev_max_backlog=3000
#net.ipv4.tcp_max_syn_backlog = 1024
net.ipv4.tcp_synack_retries = 2
kernel.sysrq = 1
net.core.default_qdisc = fq
net.ipv4.tcp_congestion_control = bbr
```

1.2、安装翻墙服务

yum install shadowsocks-libev

```
[root@izj6c1u9ywulp7cksa4385z ~]# cat /etc/shadowsocks-libev/config.json
{
    "server":"127.0.0.1",
    "server_port":4000,
    "local_port":1080,
    "password":"123456",
    "timeout":60,
    "method":"aes-256-cfb"
}
```

1.3、配置 ss.yml 与 web 管理端

```
[root@izj6c1u9ywulp7cksa4385z ~]# cat .ssmgr/ss.yml
type: s

shadowsocks:
  address: 127.0.0.1:4000
  # 这里的地址和端口需要跟上一步的 --manager-address 参数保持一致，连接上述 udp 端口
manager:
  address: 0.0.0.0:4001
  # 这个 address 参数会让程序监听一个 tcp 端口，用于接收 webgui 发送过来的控制命令
  password: '123456'
db: 'ss.sqlite'

[root@izj6c1u9ywulp7cksa4385z ~]# cat .ssmgr/webgui.yml
type: m

manager:
  address: 47.75.124.126:4001
  # 这里换成 manager 所在的服务器的 ip
  password: '123456'
  # 这部分的端口和密码需要跟上一步 manager 参数里的保持一致，以连接 type s 部分监听的 tcp 端口
plugins:
  flowSaver:
    use: true
  user:
    use: true
```

```
  account:
    use: true
  macAccount:
    use: true
  email:
    use: true
    type: 'smtp'
    username: 'myhbs520@sina.com'
    password: 'nipc#123,,'
    host: 'smtp.sina.com'
    # 这部分的邮箱和密码是用于发送注册验证邮件，重置密码邮件
  webgui:
    use: true
    host: '0.0.0.0'
    port: '80'
    site: 'http://www.myserver.com'
db: 'webgui.sqlite'

[root@izj6c1u9ywulp7cksa4385z .ssmgr]# cat default.yml
type: s
shadowsocks:
  address: 127.0.0.1:4000
manager:
  address: 0.0.0.0:4001
  password: '123456'
db: 'db.sqlite'
# db:
#   host: '1.1.1.1'
#   user: 'root'
#   password: 'abcdefg'
#   database: 'ssmgr'

[root@izj6c1u9ywulp7cksa4385z .ssmgr]# cat /etc/shadowsocks-
libev/config.json
{
    "server":"127.0.0.1",
    "server_port":4000,
    "local_port":1080,
    "password":"123456",
    "timeout":60,
    "method":"aes-256-cfb"
}


[root@izj6c1u9ywulp7cksa4385z ~]#
cat .shadowsocks/.shadowsocks_50002.conf
{
"server_port":50002,
"password":"9086656278",
"method":"aes-256-cfb"
}

[root@izj6c1u9ywulp7cksa4385z ~]#
cat .shadowsocks/.shadowsocks_50001.conf
{
```

```
"server_port":50001,
"password":"4550384915",
"method":"aes-256-cfb"
}
```

1.4、启动 ss 服务。

```
ss-server --manager-address 127.0.0.1:4000 -f
/root/.shadowsocks/.shadowsocks_50002.pid -c
/root/.shadowsocks/.shadowsocks_50002.conf

ss-manager -m aes-256-cfb -u --manager-address 127.0.0.1:4000 &
ssmgr -c ss.yml &
ssmgr -c webgui.yml &
```

2、IKE2 VPN 安装

```
vpn--IKE
```

2.1、源码安装

```
cd strongswan-5.5.2
./configure  --enable-eap-identity --enable-eap-md5 --enable-eap-
mschapv2 --enable-eap-tls --enable-eap-ttls --enable-eap-peap --enable-
eap-tnc --enable-eap-dynamic --enable-eap-radius --enable-xauth-eap --
enable-xauth-pam  --enable-dhcp  --enable-openssl  --enable-addrblock --
enable-unity --enable-certexpire --enable-radattr --enable-swanctl --
enable-openssl --disable-gmp
make && make install
```

2.2 配置

```
[root@izj6c1u9ywulp7cksa4385z ~]#  cat /usr/local/etc/ipsec.conf
config setup
     uniqueids=never
conn %default
     ikelifetime=60m
     keylife=20m
     rekeymargin=3m
     rekey=no
     keyingtries=1
     keyexchange=ike
     leftsubnet=0.0.0.0/0
     right=%any
     rightsourceip=10.28.0.0/24
     dpdaction=clear
     dpddelay=300s
     dpdtimeout=1h
conn Windows7-os+
     keyexchange=ikev2
     auto=add
     leftauth=pubkey
     leftcert=serverCert.crt
     rightauth=eap-mschapv2
     rightsendcert=never
```

```
     eap_identity=%identity
     compress=yes

conn IOS_Mac_IKEv2
    keyexchange=ikev2
    ike=aes256-sha256-modp2048,3des-sha1-modp2048!
    esp=aes256-sha256,3des-sha1!
    rekey=no
    left=%defaultroute
    leftid=gomrwind.com
    leftsendcert=always
    leftsubnet=0.0.0.0/0
    leftcert=serverCert.crt
    right=%any
    rightauth=eap-mschapv2
    rightsourceip=10.28.0.0/24
    rightsendcert=never
    eap_identity=%any
    dpdaction=clear
    fragmentation=yes
    auto=add
[root@izj6c1u9ywulp7cksa4385z ~]# cat /usr/local/etc/strongswan.conf
charon {
        i_dont_care_about_security_and_use_aggressive_mode_psk = yes
        load_modular = yes
        duplicheck.enable = no
        threads = 16
        compress = yes
        plugins {
                include strongswan.d/charon/*.conf
        }
        dns1 = 8.8.8.8
        dns2 = 114.114.114.114
        nbns1 = 8.8.8.8
        nbns2 = 8.8.4.4
}
include strongswan.d/*.conf
[root@izj6c1u9ywulp7cksa4385z ~]#
[root@izj6c1u9ywulp7cksa4385z ~]#  cat /usr/local/etc/ipsec.secrets
: RSA serverKey.key
: PSK "iloveworld"
: XAUTH "iloveworldPass"
test %any : EAP "test123"
testUserOne %any : EAP "testOnePass"

启动服务
ipsec start
上网
iptables -t nat -A POSTROUTING -s 10.28.0.0/24 -o eth0 -j MASQUERADE
```

配置 strongswan 需要用到以下三个文件：

证书：

```
[root@izj6c1u9ywulp7cksa4385z ipsec.d]# pwd
/usr/local/etc/ipsec.d
```

1. 由 symantec 签发的证书

2. symantec 的中级证书

3. 证书的私钥

证书的私钥在生成 CSR 的时候是一并生成的，而 1 和 2 则需要下载 nginx 数字证书：

打开下载下来的证书会看到两端证书内容：

```
-----BEGIN CERTIFICATE-----
...服务器证书...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...中级证书...
-----END CERTIFICATE-----
```

如果你是按照我的安装配置方法配置 strongswan（配置基于 StrongSwan 的 IKEv2），那么可以继续往下操作：

1.删除 caCert.crt：

```
[root@hk1 ~]# rm -f /usr/local/ipsec/etc/ipsec.d/cacerts/caCert.crt
```

2.修改私钥，将内容替换掉原来的内容：

```
[root@hk1 ~]# vim /usr/local/ipsec/etc/ipsec.d/private/serverKey.key

#先按两次 g，回到文件顶部
#然后按 dG，删除所有内容
#再然后按 i 并粘贴入私钥，然后按 ESC 退出编辑模式
#最后输入:wq 保存退出
```

服务器私钥

```
[root@izj6c1u9ywulp7cksa4385z private]# cat serverKey.key
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAi7giD2xjFCjghaVxkTlGZXuZerTOIe2nqvigx8IJl3KMxP3h
8C6H9soFU0V95fi9/D56wgjuk1a/kpwftvTSW0DVgcQoy+fMJefadMuWOrz5eGNN
MbzYyIPQ7x9xnGBO21ceT3l7pvVTLEdBt7ErJffcrMYo2lvpTnK/u5wXE5CxNf/E
jD7Lfun4JRaKwh3MMmjJ89JwbYuJpsSufD34JygzBorFcnTZaIUp7SODARArFr49
T3uvk5r3kEpI1IQVKrCaAKqKvUxa8n5wfVfVRahC5jHJ6i/ueEJhaYWc2txINpfI
M48budrZYweVE2GP8swSjBDOfjynu+k0YVY19QIDAQABAoIBAQCBuzNFBuroaRww
e/H7WicXS7Cy0vcjYbvxcHI5iN3HFcHHl8UWct4kNGjFbRz9T/14Hbt2Nzzu6flv
dC+ufZxkxsp4d186Fbzk5WdeUzrmx1kMg4fFywevB1YU8ilXfcU6vKWUPcSmvdvA
nVlT6tmZo4/vjP+Z4QW8paeJ1+FFdAfD/NtTswkPXr6R20PnRaOWAPHTnB6uKULC
3IdCmSq65ZXKtSqlmEsaeuSUgeiZ5py0n0jfSld27c/kGWjpWBgG7y9+VtG1Yh4O
Hrd+cEO/MnpGDeTxioqsnLsmzd/7Qxtz8LHhs4C8aLDIyJmG/wYBgzQk9tu5XGuo
Zslx6JaBAoGBAO4fSCHyc4vKdypAuH4r3wv6rpdxSaYFOeT4cx62bX0mzfZVbxwR
NOKbl8TlswyvUJbrspjYFf59tFLbQ6DtmsxtgCby5puDZXo98n9+f5pUh+qsvLp6
uztE9IpUEN1gsKvoeOO4JHDiN/t5x19ca8vbzfCb3072aI0cqwHDqrO5AoGBAJY1
iwccP8+sNCWC7V0HjnOi4kxnyXMuC3fEapeiiqxskfeybMyWsvqB/DCuOaIBP3Vc
HXk5pJC4c4er2XPPAv25pFTorybs0JmoXDHCr8Ro5YrbVtM6asVPgoMMVMN6OYQC
i/qZt7HCRpy/ftU1nrH20CKtj09NKqC3+Ct6X6odAoGAKDboO7NyiB09WslrXwOR
u7guXC0pvvwK9vxulLVCcGxpRDzRcvAE+nBEhWA1iOLivVo44087B6u7FHMrTLZT
+kpT5kAHaiQaszZbCeUXL70u5xcdGHE5xZyzn9jqsDhwHdg48lxa6yWJRtMeNKCr
I9aPRR9N60v6BDrfNoERByECgYAazf08TJc6iQFZPE79Te7mxcdaWR8C1dGoEr3u
x0u1zz4RFJ8Dt1TDDVUinsScG1rmp4R+8IFkbfX33PvWBbgD7Q/dMxSmfpYPBZNG
MfbJtupXaCLLWN5azYB9wEKUC4id/RJbqviz2PZe3HFWvNXP7jpgJdawERYtTgcU
Qyc1JQKBgB5xKeJB0F4ydN3M7T4Aa18Kz/FjHArRL4K8YweOWTTk97mNpOGkICGo
t+t+U/aL4bGkDHuHCk1RP+4frA1UqZFmr0bbzhp1wkhKPZDJnQBNA/zGp20c2djP
uqofwLV9iBRDpWhdEzg+kRMZOWEWDZ03SG0LKo+O05fY86kaGD2K
-----END RSA PRIVATE KEY-----
```

3.修改服务器证书，将内容替换掉原来的内容：

```
[root@hk1 ~]# vim
/usr/local/ipsec/etc/ipsec.d/certs/serverCert.crt

#先按两次 g，回到文件顶部
#然后按 dG，删除所有内容
#再然后按 i 并粘贴入私钥，然后按 ESC 退出编辑模式
#最后输入:wq 保存退出
```

服务器证书

```
[root@izj6c1u9ywulp7cksa4385z certs]# cat serverCert.crt
-----BEGIN CERTIFICATE-----
MIIFkzCCBHugAwIBAgIQDYo0XsbcttUVzBhSgTxYPTANBgkqhkiG9w0BAQsFADBu
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMS0wKwYDVQQDEyRFbmNyeXB0aW9uIEV2ZXJ5d2hlcmUg
RFYgVExTIENBIC0gRzEwHhcNMTgwNTE1MDAwMDAwWhcNMTkwNTE1MTIwMDAwWjAZ
MRcwFQYDVQQDDA4qLmdvbXJ3aW5kLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEP
```

```
ADCCAQoCggEBAIu4Ig9sYxQo4IWlcZE5RmV7mXq0ziHtp6r4oMfCCZdyjMT94fAu
h/bKBVNFfeX4vfw+esII7pNWv5KcH7b00ltA1YHEKMvnzCXn2nTLljq8+XhjTTG8
2MiD0O8fcZxgTttXHk95e6b1UyxHQbexKyX33KzGKNpb6U5yv7ucFxOQsTX/xIw+
y37p+CUWisIdzDJoyfPScG2LiabErnw9+CcoMwaKxXJ02WiFKe0jgwEQKxa+PU97
r5Oa95BKSNSEFSqwmgCqir1MWvJ+cH1X1UWoQuYxyeov7nhCYWmFnNrcSDaXyDOP
G7na2WMHlRNhj/LMEowQzn48p7vpNGFWNfUCAwEAAaOCAoAwggJ8MB8GA1UdIwQY
MBaAFFV0T7JyT/VgulDR1+ZRXJoBhxrXMB0GA1UdDgQWBBT/guB+dX9XeRnI8cCD
CXFaiCPt/DAnBgNVHREEIDAegg4qLmdvbXJ3aW5kLmNvbYIMZ29tcndpbmQuY29t
MA4GA1UdDwEB/wQEAwIFoDAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAwIw
TAYDVR0gBEUwQzA3BglghkgBhv1sAQIwKjAoBggrBgEFBQcCARYcaHR0cHM6Ly93
d3cuZGlnaWNlcnQuY29tL0NQUzAIBgZngQwBAgEwgYEGCCsGAQUFBwEBBHUwczAl
BggrBgEFBQcwAYYZaHR0cDovL29jc3AyLmRpZ2ljZXJ0LmNvbTBKBggrBgEFBQcw
AoY+aHR0cDovL2NhY2VydHMuZGlnaWNlcnQuY29tL0VuY3J5cHRpb25FdmVyeXdo
ZXJlRFZUTFNDQS1HMS5jcnQwCQYDVR0TBAIwADCCAQMGCisGAQQB1nkCBAIEgfQE
gfEA7wB2AKS5CZC0GFgUh7sTosxncAo8NZgE+RvfuON3zQ7IDdwQAAABY2I4UP4A
AAQDAEcwRQIgTbtwmUlfSEvsBjg0CZGzT/FojxtmJ0vskKmhRllXDn8CIQCUC3YL
sof8NyY5MPUECN5yKaPCftGIP9MyMvIRWFh/OgB1AG9Tdqwx8DEZ2JkApFEV/3cV
HBHZAsEAKQaNsgiaN9kTAAABY2I4UnIAAAQDAEYwRAIgG/P6n3Syh4sCTuiIFCP9
JwMs/H3EA7YWA+uI6MrlxWcCIALS+HPV5K09RlqmJdx2E4N1ro2EzQQ+LF2uVLUP
BORIMA0GCSqGSIb3DQEBCwUAA4IBAQBpZy+KxS/FTuUSQrM7N0rZe/dle7ahOC6c
K6oFBdI7eXec8TTlrJMkx0c7vgr48qAK1TGRIVMN/RslLCBuDuBtkzTe1yGidnaQ
fI7kFfPEALI9/8mi+1ekEqNkZQR76+w0SxzTQVCkZquEhloxJO0pfL4CoU1sDMFB
qXlmdqjTByt7mzsN1dXU6utC+oXRgGWgkDyEWMKVH2P1+rUDoiwBTdvnlUjet9K6
Ssu6yGIU/YCDZYFJywuXW9N57neG7HnrvJNGlk4pgllfsF/L+TtAxuEr+esKS6Zh
CQrjmhxPs4GkrdbFk3FVnXpdRbrNMZT+XH0Jg7Wtno7b1pywN/Mv
-----END CERTIFICATE-----
```

4.修改中级证书，将内容替换掉原来的内容：

```
[root@hk1 ~]# vim
/usr/local/ipsec/etc/ipsec.d/cacerts/caCert_Intermediate.crt

#先按两次 g，回到文件顶部
#然后按 dG，删除所有内容
#再然后按 i 并粘贴入私钥，然后按 ESC 退出编辑模式
#最后输入:wq 保存退出
```

中间人证书（根证书）

```
[root@izj6c1u9ywulp7cksa4385z cacerts]# cat caCert_Intermediate.crt
-----BEGIN CERTIFICATE-----
MIIEqjCCA5KgAwIBAgIQAnmsRYvBskWr+YBTzSybsTANBgkqhkiG9w0BAQsFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD
QTAeFw0xNzExMjcxMjQ2MTBaFw0yNzExMjcxMjQ2MTBaMG4xCzAJBgNVBAYTAlVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j
b20xLTArBgNVBAMTJEVuY3J5cHRpb24gRXZlcnl3aGVyZSBEViBUTFMgQ0EgLSBH
MTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALPeP6wkab41dyQh6mKc
oHqt3jRIxW5MDvf9QyiOR7VfFwK656es0UFiIb74N9pRntzF1UgYzDGu3ppZVMdo
lbxhm6dWS9OK/lFehKNT0OYI9aqk6F+U7cA6jxSC+iDBPXwdF4rs3KRyp3aQn6pj
pp1yr7IB6Y4zv72Ee/PlZ/6rK6InC6WpK0nPVOYR7n9iDuPe1E4IxUMBH/T33+3h
yuH3dvfgiWUOUkjdpMbyxX+XNle5uEIiyBsi4IvbcTCh8ruifCIi5mDXkZrnMT8n
```

```
wfYCV6v6kDdXkbgGRLKsR4pucbJtbKqIkUGxuZI2t7pfewKRc5nWecvDBZf3+p1M
pA8CAwEAAaOCAU8wggFLMB0GA1UdDgQWBBRVdE+yck/1YLpQ0dfmUVyaAYca1zAf
BgNVHSMEGDAWgBQD3lA1VtFMu2bwo+IbG8OXsj3RVTAOBgNVHQ8BAf8EBAMCAYYw
HQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMBIGA1UdEwEB/wQIMAYBAf8C
AQAwNAYIKwYBBQUHAQEEKDAmMCQGCCsGAQUFBzABhhhodHRwOi8vb2NzcC5kaWdp
Y2VydC5jb20wQgYDVR0fBDswOTA3oDWgM4YxaHR0cDovL2NybDMuZGlnaWNlcnQu
Y29tL0RpZ2lDZXJ0R2xvYmFsUm9vdENBLmNybDBMBgNVHSAERTBDMDcGCWCGSAGG
/WwBAjAqMCgGCCsGAQUFBwIBFhxodHRwczovL3d3dy5kaWdpY2VydC5jb20vQ1BT
MAgGBmeBDAECATANBgkqhkiG9w0BAQsFAAOCAQEAK3Gp6/aGq7aBZsxf/oQ+TD/B
SwW3AU4ETK+GQf2kFzYZkby5SFrHdPomunx2HBzViUchGoofGgg7gHW0W3MlQAXW
M0r5LUvStcr82QDWYNPaUy4taCQmyaJ+VB+6wxHstSigOlSNF2a6vg4rgexixeiV
4YSB03Yqp2t3TeZHM9ESfkus74nQyW7pRGezj+TC44xCagCQQOzzNmzEAP2SnCrJ
sNE2DpRVMnL8J6xBRdjmOsC3N6cQuKuRXbzByVBjCqAA8t1L0I+9wXJerLPyErjy
rMKWaBFLmfK/AHNF4ZihwPGOc7w6UHczBZXH5RFzJNnww+WnKuTPI0HfnVH8lg==
-----END CERTIFICATE-----
```

二、kafak zk 服务器 --docker 部署

docker (zookeepr,kafak)

1、zookeeper. dockerfile 配置文件

```
FROM wurstmeister/base
MAINTAINER Wurstmeister
ENV ZOOKEEPER_VERSION 3.4.12
#RUN apk add --no-cache gnupg openssh-server
#Download Zookeeper
RUN wget -q http://40.79.78.1/dist/zookeeper/zookeeper-
${ZOOKEEPER_VERSION}/zookeeper-${ZOOKEEPER_VERSION}.tar.gz && \
    wget -q http://40.79.78.1/dist/zookeeper/KEYS && \
    wget -q http://40.79.78.1/dist/zookeeper/zookeeper-
${ZOOKEEPER_VERSION}/zookeeper-${ZOOKEEPER_VERSION}.tar.gz.asc && \
    wget -q http://40.79.78.1/dist/zookeeper/zookeeper-
${ZOOKEEPER_VERSION}/zookeeper-${ZOOKEEPER_VERSION}.tar.gz.md5
#Verify download
RUN md5sum -c zookeeper-${ZOOKEEPER_VERSION}.tar.gz.md5 && \
gpg --import KEYS && \
gpg --verify zookeeper-${ZOOKEEPER_VERSION}.tar.gz.asc
#Install
#RUN mkdir /opt
RUN tar -xzf zookeeper-${ZOOKEEPER_VERSION}.tar.gz -C /opt
#Configure
RUN mv /opt/zookeeper-${ZOOKEEPER_VERSION}/conf/zoo_sample.cfg
/opt/zookeeper-${ZOOKEEPER_VERSION}/conf/zoo.cfg
ENV JAVA_HOME /usr/lib/jvm/java-7-openjdk-amd64
ENV ZK_HOME /opt/zookeeper-${ZOOKEEPER_VERSION}
RUN sed  -i "s|/tmp/zookeeper|$ZK_HOME/data|g" $ZK_HOME/conf/zoo.cfg;
mkdir $ZK_HOME/data
ADD start-zk.sh /usr/bin/start-zk.sh
EXPOSE 2181 2888 3888
WORKDIR /opt/zookeeper-${ZOOKEEPER_VERSION}
VOLUME ["/opt/zookeeper-${ZOOKEEPER_VERSION}/conf", "/opt/zookeeper-
${ZOOKEEPER_VERSION}/data"]
```

```
CMD /usr/sbin/sshd && bash /usr/bin/start-zk.sh
```

2、zookeeper. 镜像

```
docker build -t zookeeper:3.4.12 .
```

3、kafak dockerfile

```
FROM openjdk:8u171-jre-alpine
ARG kafka_version=2.0.0
ARG scala_version=2.12
ARG glibc_version=2.27-r0
MAINTAINER wurstmeister
ENV KAFKA_VERSION=$kafka_version \
    SCALA_VERSION=$scala_version \
    KAFKA_HOME=/opt/kafka \
    GLIBC_VERSION=$glibc_version
ENV PATH=${PATH}:${KAFKA_HOME}/bin
COPY download-kafka.sh start-kafka.sh broker-list.sh create-topics.sh
versions.sh /tmp/
RUN apk add --no-cache bash curl jq docker \
 && mkdir /opt \
 && chmod a+x /tmp/*.sh \
 && mv /tmp/start-kafka.sh /tmp/broker-list.sh /tmp/create-topics.sh
/tmp/versions.sh /usr/bin \
 && sync && /tmp/download-kafka.sh \
 && tar xfz /tmp/kafka_${SCALA_VERSION}-${KAFKA_VERSION}.tgz -C /opt \
 && rm /tmp/kafka_${SCALA_VERSION}-${KAFKA_VERSION}.tgz \
 && ln -s /opt/kafka_${SCALA_VERSION}-${KAFKA_VERSION} /opt/kafka \
 && rm /tmp/* \
 && wget https://github.com/sgerrand/alpine-pkg-
glibc/releases/download/${GLIBC_VERSION}/glibc-${GLIBC_VERSION}.apk \
 && apk add --no-cache --allow-untrusted glibc-${GLIBC_VERSION}.apk \
 && rm glibc-${GLIBC_VERSION}.apk
COPY overrides /opt/overrides
VOLUME ["/kafka"]
# Use "exec" form so that it runs as PID 1 (useful for graceful
shutdown)
CMD ["start-kafka.sh"]
```

4、kafak 镜像

```
docker build -t kafka:2.0 .
```

5、 启动 zk 与 kafak

启动 zk :

```
docker run -it -d --restart unless-stopped --network mynetwork --ip
172.18.2.115 --dns=172.18.3.251 --name zookeeper zookeeper:3.4.2
```

启动 kafak

```
docker run -it -d --restart unless-stopped --network mynetwork --ip
172.18.2.116 --dns=172.18.3.251 -e
```

KAFKA_BROKER_ID=0 -e KAFKA_ZOOKEEPER_CONNECT=172.18.2.115:2181 -e KAFKA_ADVERTISED_LISTENERS=PLAINTEXT://172.18.2.115:9092 -e KAFA_LISTENERS=PLAINTEXT://0.0.0.0:9092 --name kafak kafka:2.0

RDS:
rm-j6cr19qp1fg0vmh89.mysql.rds.aliyuncs.com
wadmin Wad#2018Ak
root g6hfuV1kAy#u

NAT:
NAT 网关，需要申请 EIP 地址
EIP:

DNAT

SNAT： 源地址转换

SNAT条目列表

创建SNAT条目

SNAT条目ID	源网段	交换机ID	公网IP地址	状态	操作
snat-bp1rr6qum88c1erq0qc62	10.86.122.0/24	vsw-bp1e3ga00ub5e4k21i98y	47.96.141.14	可用	编辑 移除
snat-bp179iwekrcoo9486131x	10.86.123.0/24	vsw-bp1w6nh54ib9v7csy3c9e	47.96.141.14	可用	编辑 移除
snat-bp1jvr3da9qb5z6h2nko2	10.86.121.0/24	vsw-bp1uj9bm3dfytg4y4lgqf	47.96.141.14	可用	编辑 移除

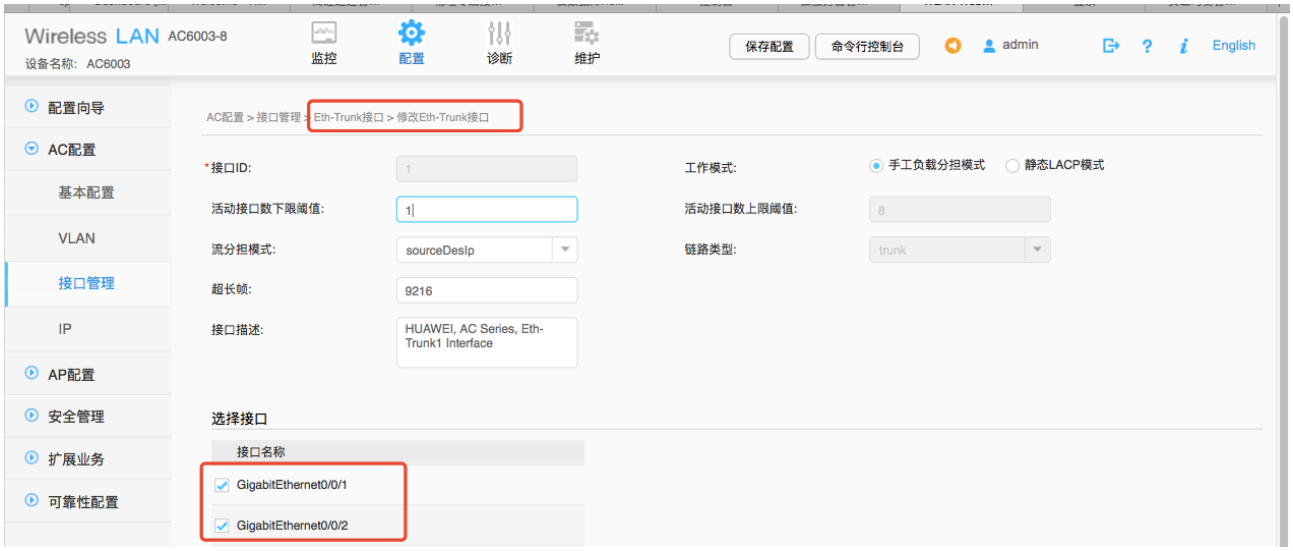
咨询·建议

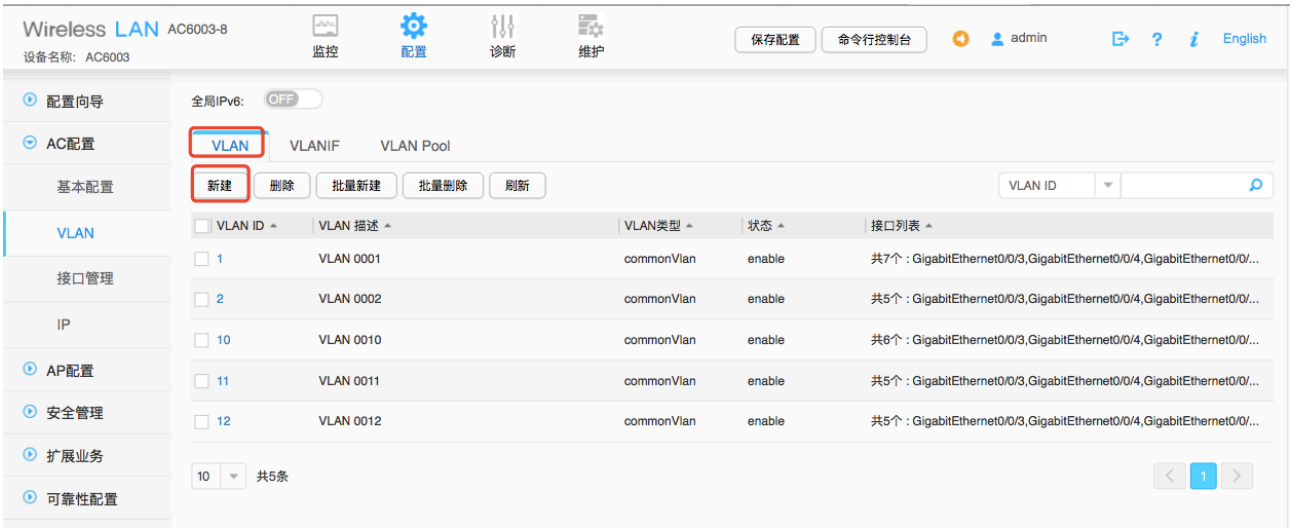
2、网络设备帐号密码：

一、无线控制器：

1、

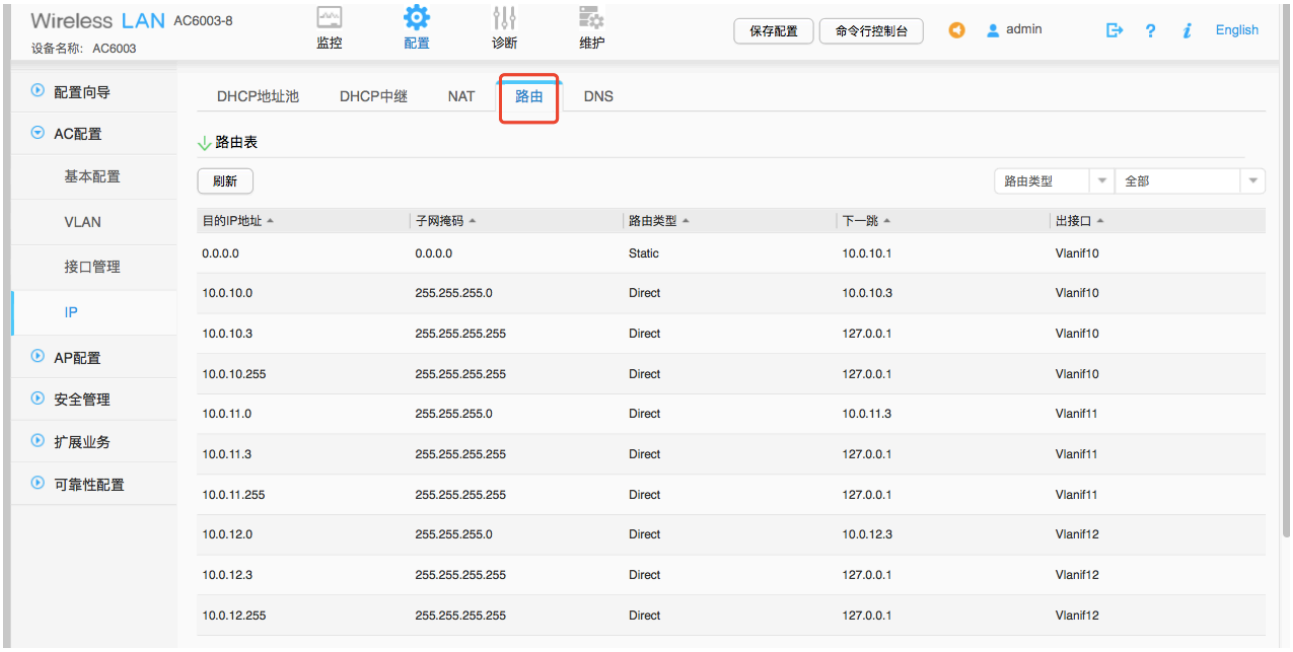
新建 VLAN：





接口配置： Trunk

路由：

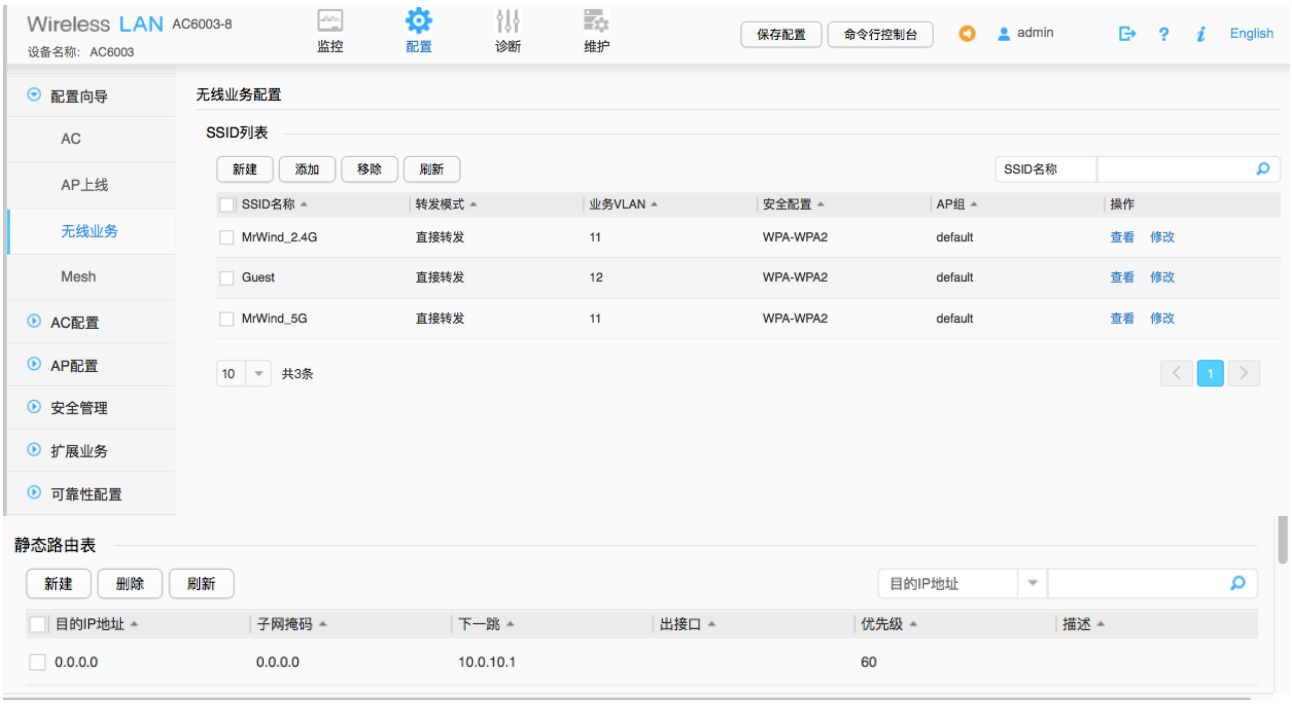

Wireless LAN AC6003-8
设备名称: AC6003
监控
配置
诊断
维护
保存配置
命令行控制台
admin
English
配置向导
AC配置
基本配置
VLAN
接口管理
IP
AP配置
安全管理
扩展业务
可靠性配置
DHCP地址池
DHCP中继
NAT
路由
DNS
路由表
刷新
路由类型
全部
目的IP地址
子网掩码
路由类型
下一跳
出接口
0.0.0.0 0.0.0.0 Static 10.0.10.1 Vlanif10
10.0.10.0 255.255.255.0 Direct 10.0.10.3 Vlanif10
10.0.10.3 255.255.255.255 Direct 127.0.0.1 Vlanif10
10.0.10.255 255.255.255.255 Direct 127.0.0.1 Vlanif10
10.0.11.0 255.255.255.0 Direct 10.0.11.3 Vlanif11
10.0.11.3 255.255.255.255 Direct 127.0.0.1 Vlanif11
10.0.11.255 255.255.255.255 Direct 127.0.0.1 Vlanif11
10.0.12.0 255.255.255.0 Direct 10.0.12.3 Vlanif12
10.0.12.3 255.255.255.255 Direct 127.0.0.1 Vlanif12
10.0.12.255 255.255.255.255 Direct 127.0.0.1 Vlanif12

虚拟接口：


静态路由配置表
新建
删除
刷新
目的IP地址
子网掩码
下一跳
出接口
优先级
描述
0.0.0.0 0.0.0.0 10.0.10.1 60
10
共1条
1

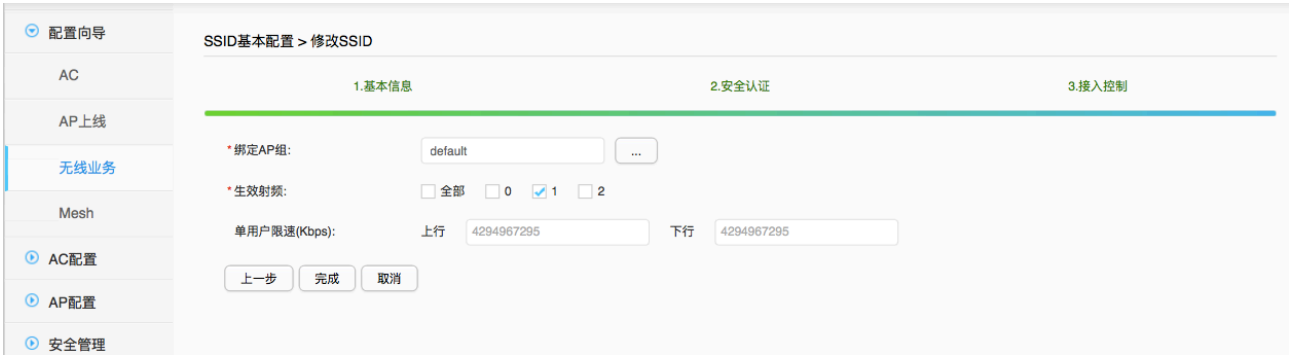

配置向导
AC
AP上线
无线业务
Mesh
AC配置
AP配置
安全管理
扩展业务
可靠性配置
1.AC基本配置
2.端口配置
3.网络互联配置
4.AC备份配置
5.AC源地址
6.配置确认
IPv6:
OFF
接口配置
新建
删除
刷新
虚拟接口
虚拟接口
描述
IPv4地址/掩码
DHCP
成员接口
Vlanif1
HUAWEI, AC Series, Vlanif1 Interfa...
169.254.1.1/255.255.0.0
查看详情
Vlanif10
HUAWEI, AC Series, Vlanif10 Interf...
10.0.10.3/255.255.255.0
查看详情
Vlanif11
HUAWEI, AC Series, Vlanif11 Interf...
10.0.11.3/255.255.255.0
查看详情
Vlanif12
HUAWEI, AC Series, Vlanif12 Interf...
10.0.12.3/255.255.255.0
查看详情
5
共4条
1
全局地址池
新建
删除
刷新
地址池名称
地址池名称
IP地址池子网地址
子网掩码
暂无数据

无线业务：


Wireless LAN
AC6003-8
设备名称: AC6003
监控
配置
诊断
维护
保存配置
命令行控制台
admin
English
配置向导
AC
AP上线
无线业务
Mesh
AC配置
AP配置
安全管理
扩展业务
可靠性配置
无线业务配置
SSID列表
新建
添加
移除
刷新
SSID名称
SSID名称
转发模式
业务VLAN
安全配置
AP组
操作
MrWind_2.4G
直接转发
11
WPA-WPA2
default
查看 修改
Guest
直接转发
12
WPA-WPA2
default
查看 修改
MrWind_5G
直接转发
11
WPA-WPA2
default
查看 修改
10
共3条
1
静态路由表
新建
删除
刷新
目的IP地址
目的IP地址
子网掩码
下一跳
出接口
优先级
描述
0.0.0.0
0.0.0.0
10.0.10.1
60

2.4G 网络：

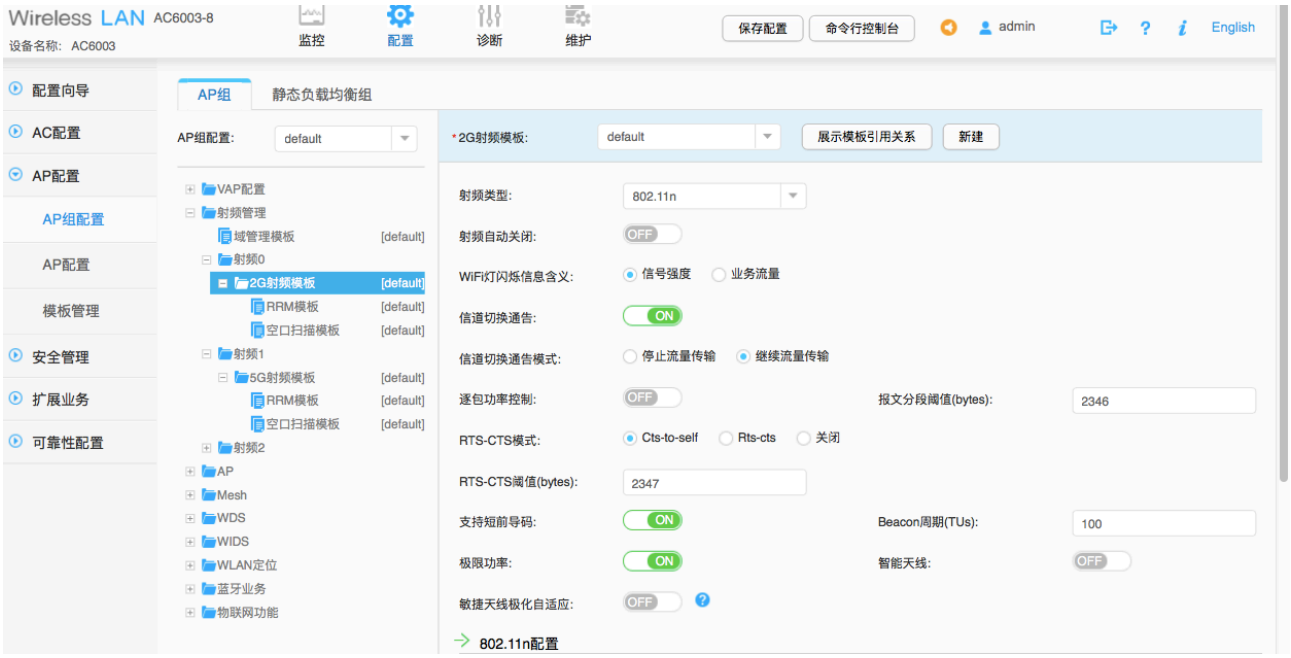
5G 网络：



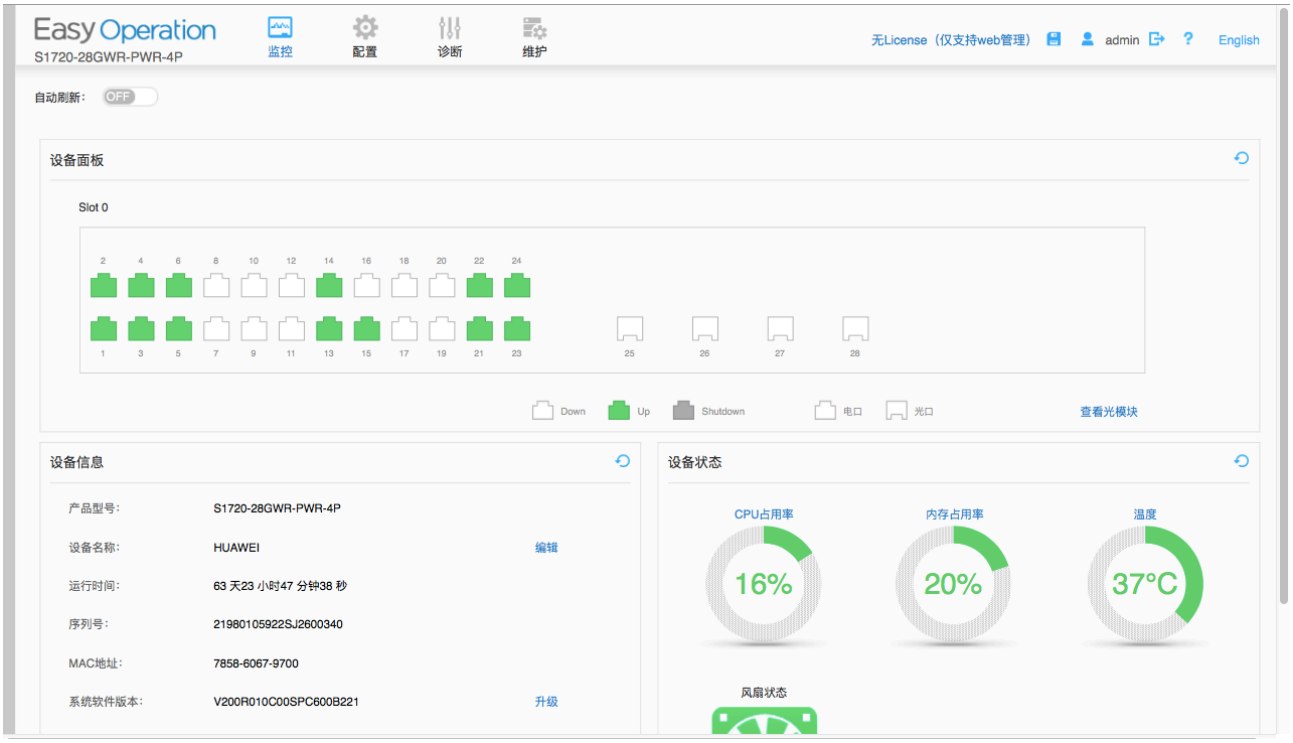
Guest 网络：

AP 组配置：

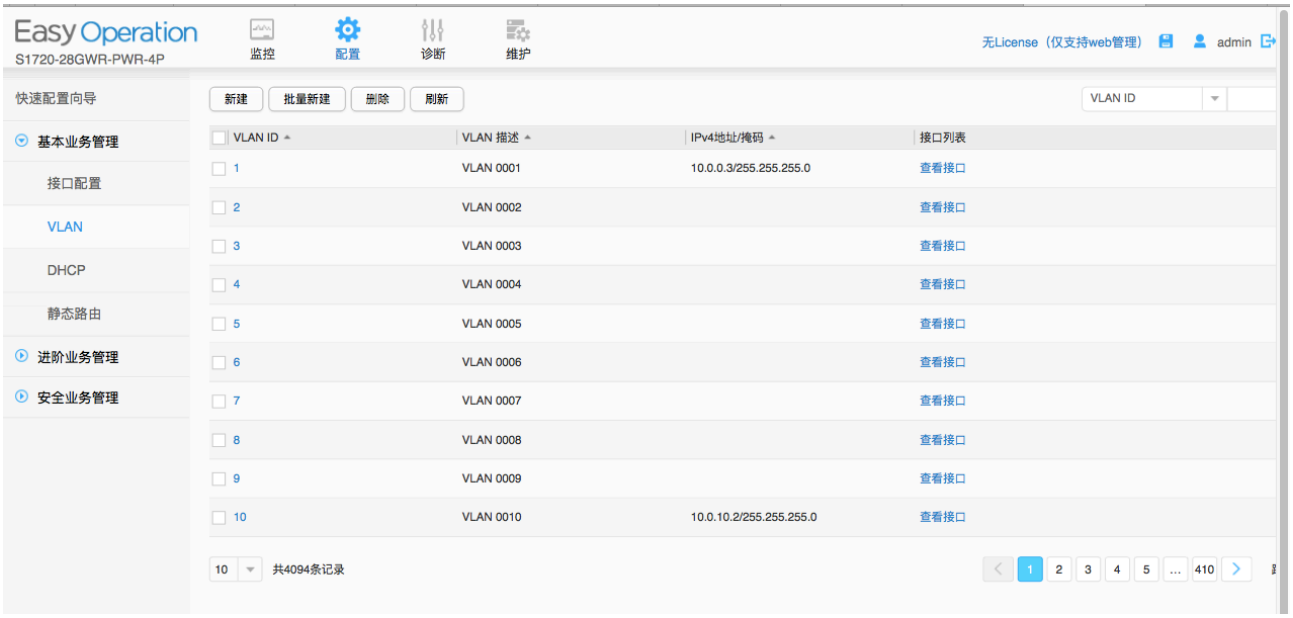
射配信息 2G 与 5G 网络 AP 组默认模板。



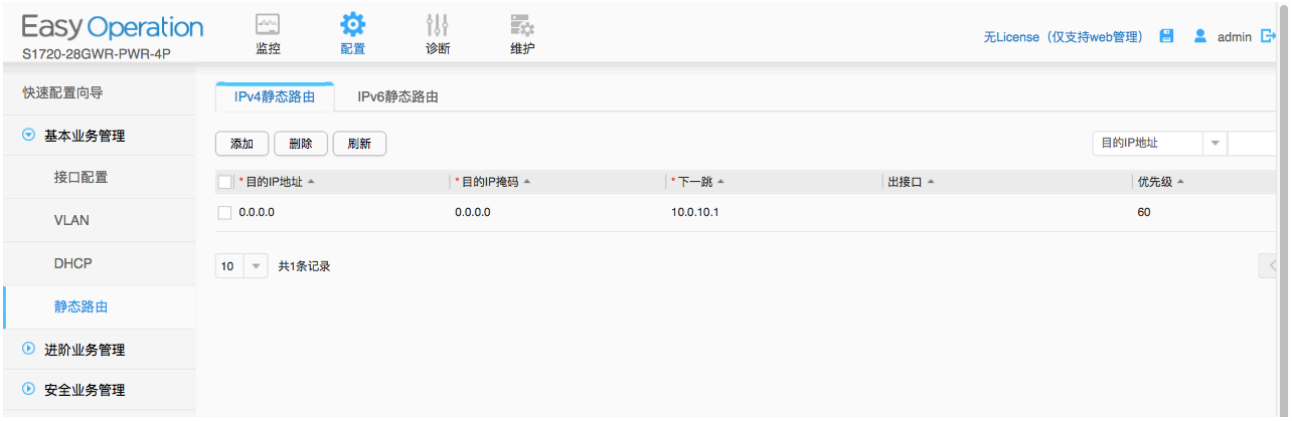
二、交换机：



1、VLAN 信息



2、静态路由：



3、接口信息： 连接 AP

4、接口配置 连接服务器

5、连接交换机与连接路由器 的以太困绑

三、路由器 配置文件

```
localhost:newcode zhuxinghong$ telnet 10.0.0.1
Trying 10.0.0.1...
Connected to 10.0.0.1.
Escape character is '^]'.
```

```
******************************************************************************
******
* Copyright (c) 2004-2015 Hangzhou H3C Tech. Co., Ltd. All rights
reserved.  *
* Without the owner's prior written consent,
*
* no decompiling or reverse-engineering shall be allowed.
*
******************************************************************************
******


Login authentication


Username:admin
Password:
<H3C>sys
System View: return to User View with Ctrl+Z.
[H3C]dis cur
#
 version 5.20, Release 2514P14
#
 sysname H3C
#
 l2tp enable
#
 firewall enable
#
 domain default enable system
#
 dns proxy enable
#
 telnet server enable
#
 dar p2p signature-file flash:/p2p_default.mtd
#
 ndp enable
#
 ntdp enable
#
 qos carl 1 destination-ip-address range 10.0.12.2 to 10.0.12.254 per-
address shared-bandwidth
#
 cluster enable
#
 port-security enable
#
 password-recovery enable
#
acl number 3000
 rule 5 permit ip source 10.0.12.0 0.0.0.255 destination 10.0.12.0
0.0.0.255
 rule 10 deny ip source 10.0.12.0 0.0.0.255 destination 10.0.0.0
0.255.255.255
 rule 20 permit ip
```

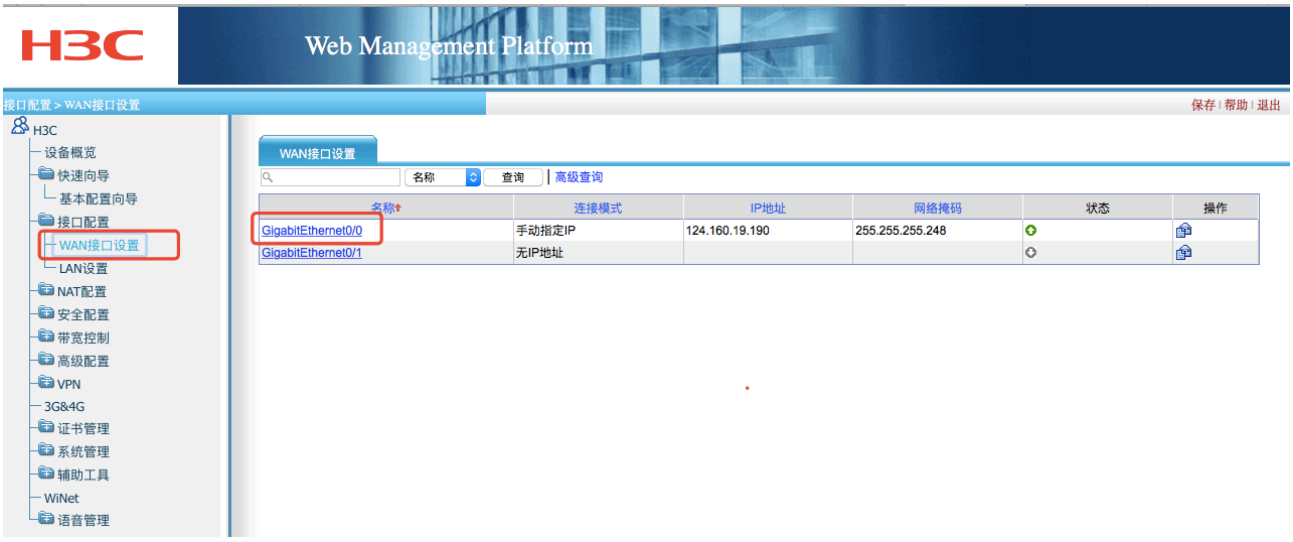
```
#
vlan 1
#
vlan 2
#
vlan 10 to 12
#
domain ststem
 authentication ppp local
 access-limit disable
 state active
 idle-cut disable
 self-service-url disable
 ip pool 2 192.168.0.100 192.168.0.200
domain system
 authentication ppp local
 access-limit disable
 state active
 idle-cut disable
 self-service-url disable
 ip pool 1 100.18.0.2 100.18.0.20
#
dhcp server ip-pool vlan1 extended
 network ip range 10.0.0.2 10.0.0.200
 network mask 255.255.255.0
 forbidden-ip 10.0.0.101 10.0.0.102 10.0.0.103 10.0.0.104 10.0.0.105
10.0.0.106 10.0.0.107 10.0.0.108
 gateway-list 10.0.0.1
 dns-list 10.0.0.1
#
dhcp server ip-pool vlan10 extended
 network ip range 10.0.10.15 10.0.10.254
 network mask 255.255.255.0
 gateway-list 10.0.10.1
#
dhcp server ip-pool vlan11 extended
 network ip range 10.0.11.2 10.0.11.254
 network mask 255.255.255.0
 gateway-list 10.0.11.1
 dns-list 221.12.33.227
#
dhcp server ip-pool vlan12 extended
 network ip range 10.0.12.2 10.0.12.254
 network mask 255.255.255.0
 gateway-list 10.0.12.1
 dns-list 10.0.12.1
#
dhcp server ip-pool vlan2 extended
 network ip range 10.0.1.15 10.0.1.254
 network mask 255.255.255.0
 gateway-list 10.0.1.1
 dns-list 10.0.1.1
#
user-group system
 group-attribute allow-guest
#
```

```
local-user admin
 password cipher $c$3$40gC1cxf/wIJNa1ufFPJsjKAof+QP5aV
 authorization-attribute level 3
 service-type telnet
 service-type web
local-user baosong
 password cipher $c$3$Qz523NBIgOgqBnNAlrrsSPyfq9ss1JfCEaxsJsk=
 authorization-attribute level 2
 service-type ppp
local-user h3c
 password cipher $c$3$+AjSdMA7OjL8ORLapGGUwRdq6GYLkS2HUxw=
 service-type ppp
local-user system
 password cipher $c$3$skPgsnm8LEXuWCsce+lqA/NY14JGEYhNeA==
 service-type ppp
local-user vpnuser
 password cipher $c$3$KaKn4obcOvwupqeMc8526GLpqr5CMG/wzjZMT/I=
 authorization-attribute level 2
 service-type ppp
#
cwmp
 undo cwmp enable
#
l2tp-group 1
 undo tunnel authentication
 allow l2tp virtual-template 1
 tunnel name H3C
#
interface Bridge-Aggregation1
 port link-type trunk
 port trunk permit vlan 1 to 2 10 to 12
#
interface Aux0
 async mode flow
 link-protocol ppp
#
interface Cellular0/0
 async mode protocol
 link-protocol ppp
 tcp mss 1024
#
interface Virtual-Template0
#
interface Virtual-Template1
 ppp authentication-mode chap domain system
 remote address pool 1
 ip address 100.18.0.1 255.255.255.0
#
interface Virtual-Template2
 ppp authentication-mode chap domain system
 ppp ipcp dns 114.114.114.114
 remote address pool 1
 ip address 192.168.10.1 255.255.255.0
#
interface NULL0
#
```

```
interface Vlan-interface1
 ip address 10.0.0.1 255.255.255.0
 tcp mss 1024
 dhcp server apply ip-pool vlan1
 ip flow-ordering internal
#
interface Vlan-interface2
 ip address 10.0.1.1 255.255.255.0
 dhcp server apply ip-pool vlan2
 ip flow-ordering internal
#
interface Vlan-interface10
 ip address 10.0.10.1 255.255.255.0
 dhcp server apply ip-pool vlan10
#
interface Vlan-interface11
 ip address 10.0.11.1 255.255.255.0
 dhcp server apply ip-pool vlan11
 ip flow-ordering internal
#
interface Vlan-interface12
 ip address 10.0.12.1 255.255.255.0
 dhcp server apply ip-pool vlan12
 firewall packet-filter 3000 inbound
 ip flow-ordering internal
#
interface GigabitEthernet0/0
 port link-mode route
 nat outbound
 nat server protocol tcp global 124.160.19.190 5001 inside 172.18.3.106
5001
 nat server protocol tcp global 124.160.19.190 8088 inside 172.18.2.200
8088
 nat server protocol tcp global 124.160.19.190 8111 inside 172.18.2.149
8111
 nat server protocol tcp global 124.160.19.190 5000 inside 172.18.2.102
8035
 nat server protocol tcp global 124.160.19.190 8034 inside 172.18.2.100
8034
 nat server protocol tcp global 124.160.19.190 8053 inside 172.18.2.101
8053
 nat server protocol tcp global 124.160.19.190 2323 inside 172.18.2.213
2323
 nat server protocol tcp global 124.160.19.190 8035 inside 172.18.2.102
8035
 nat server protocol tcp global 124.160.19.190 8030 inside 172.18.3.106
8080
 nat server protocol tcp global 124.160.19.190 8082 inside 172.18.2.222
www
 ip address 124.160.19.190 255.255.255.248
 tcp mss 1024
 qos car inbound carl 1 cir 10000 cbs 62500 ebs 0 green pass red discard
 dns server 221.12.1.227
 dns server 221.12.33.227
 ip flow-ordering external
#
```

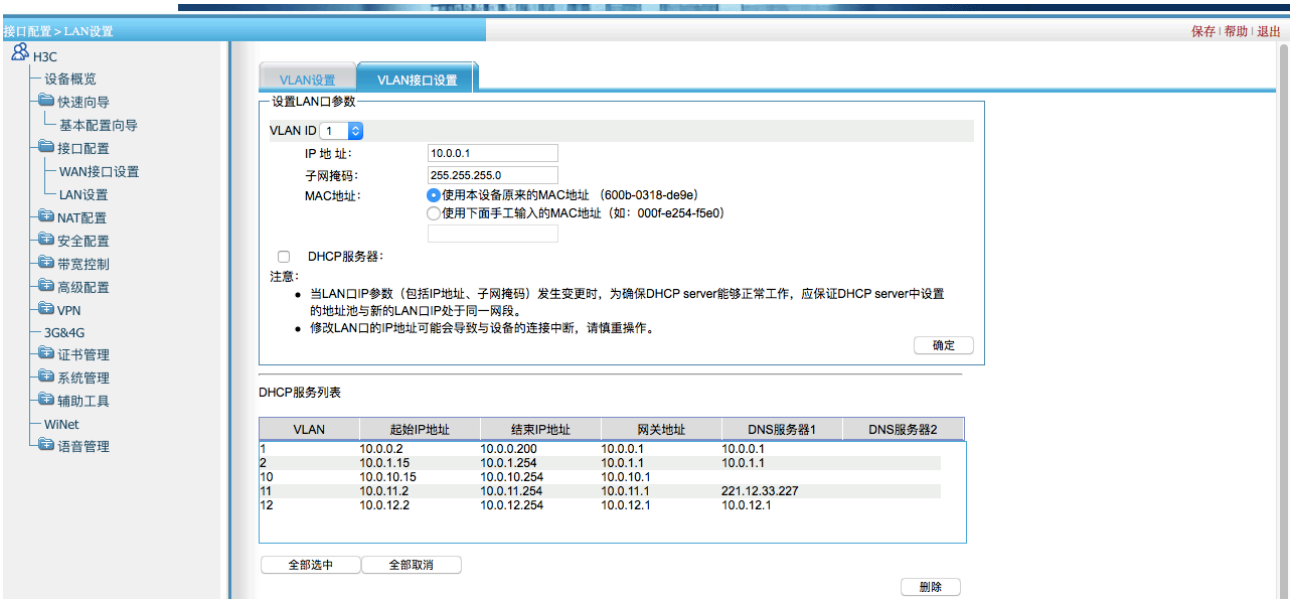
```
interface GigabitEthernet0/1
 port link-mode route
 qos max-bandwidth 1000000
 tcp mss 1024
 qos reserved-bandwidth pct 100
 qos lr outbound cir 1000000 cbs 19375000 ebs 0
 ip flow-ordering external
#
interface GigabitEthernet0/2
 port link-mode bridge
#
interface GigabitEthernet0/3
 port link-mode bridge
#
interface GigabitEthernet0/4
 port link-mode bridge
#
interface GigabitEthernet0/5
 port link-mode bridge
#
interface GigabitEthernet0/6
 port link-mode bridge
#
interface GigabitEthernet0/7
 port link-mode bridge
 port link-type trunk
 port trunk permit vlan 1 to 2 10 to 12
 port link-aggregation group 1
#
interface GigabitEthernet0/8
 port link-mode bridge
 port link-type trunk
 port trunk permit vlan 1 to 2 10 to 12
 port link-aggregation group 1
#
interface GigabitEthernet0/9
 port link-mode bridge
 port access vlan 11
#
#
voice-setup
 #
 sip
 #
 sip-server
  #
  call-rule-set
  #
  call-route
 #
 dial-program
 #
 aaa-client
 #
 gk-client
#
```

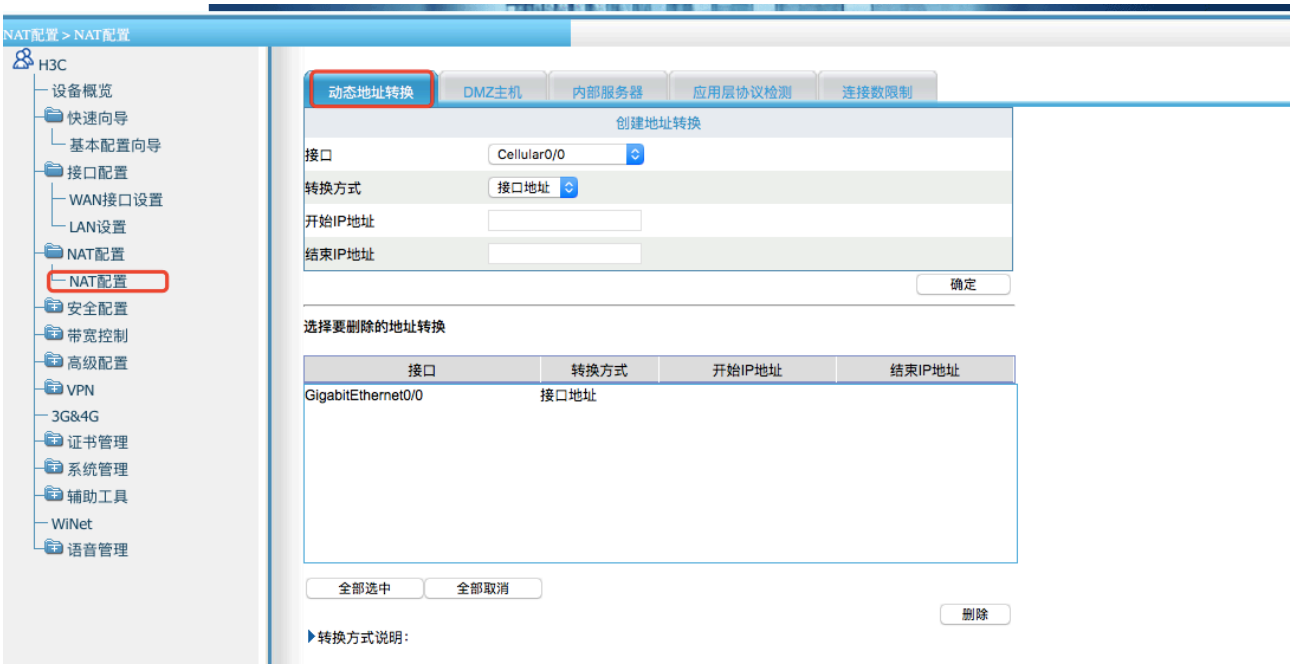
```
 ip route-static 0.0.0.0 0.0.0.0 GigabitEthernet0/0 124.160.19.185
 ip route-static 172.18.0.0 255.255.255.0 10.0.0.211
 ip route-static 172.18.1.0 255.255.255.0 10.0.0.210
 ip route-static 172.18.2.0 255.255.255.0 10.0.0.211
 ip route-static 172.18.3.0 255.255.255.0 10.0.0.210
 ip route-static 172.18.20.0 255.255.255.0 10.0.0.210
#
 dhcp server forbidden-ip 10.0.0.100 10.0.0.120
#
 dhcp enable
#
 ssh server enable
#
 nms primary monitor-interface GigabitEthernet0/0
#
 ip flow-ordering stat-interval 10
#
 load xml-configuration
#
 load tr069-configuration
#
user-interface tty 12
user-interface aux 0
user-interface vty 0 4
 authentication-mode scheme
#
return
[H3C]
```



2、公网地址配置

3、内网地址配置


接口配置 > LAN设置
保存 | 帮助 | 退出
H3C
设备概览
快速向导
基本配置向导
接口配置
WAN接口设置
LAN设置
NAT配置
安全配置
带宽控制
高级配置
VPN
3G&4G
证书管理
系统管理
辅助工具
WiNet
语音管理
VLAN设置
VLAN接口设置
设置LAN口参数
VLAN ID 1
IP 地 址: 10.0.0.1
子网掩码: 255.255.255.0
MAC地址:
使用本设备原来的MAC地址 (600b-0318-de9e)
使用下面手工输入的MAC地址 (如: 000f-e254-f5e0)
DHCP服务器:
注意:
当LAN口IP参数 (包括IP地址、子网掩码) 发生变更时, 为确保DHCP server能够正常工作, 应保证DHCP server中设置的地址池与新的LAN口IP处于同一网段。
修改LAN口的IP地址可能会导致与设备的连接中断, 请慎重操作。
确定
DHCP服务列表
VLAN 起始IP地址 结束IP地址 网关地址 DNS服务器1 DNS服务器2
1 10.0.0.2 10.0.0.200 10.0.0.1 10.0.0.1
2 10.0.1.15 10.0.1.254 10.0.1.1 10.0.1.1
10 10.0.10.15 10.0.10.254 10.0.10.1
11 10.0.11.2 10.0.11.254 10.0.11.1 221.12.33.227
12 10.0.12.2 10.0.12.254 10.0.12.1 10.0.12.1
全部选中
全部取消
删除


NAT配置 > NAT配置
H3C
设备概览
快速向导
基本配置向导
接口配置
WAN接口设置
LAN设置
NAT配置
NAT配置
安全配置
带宽控制
高级配置
VPN
3G&4G
证书管理
系统管理
辅助工具
WiNet
语音管理
动态地址转换
DMZ主机
内部服务器
应用层协议检测
连接数限制
创建地址转换
接口 Cellular0/0
转换方式 接口地址
开始IP地址
结束IP地址
确定
选择要删除的地址转换
接口 转换方式 开始IP地址 结束IP地址
GigabitEthernet0/0 接口地址
全部选中
全部取消
删除
转换方式说明:

3、网络地址映射


H3C
Web Management Platform
AT配置 > NAT配置
H3C
设备概览
快速向导
基本配置向导
接口配置
WAN接口设置
LAN设置
NAT配置
NAT配置
安全配置
带宽控制
高级配置
VPN
3G&4G
证书管理
系统管理
辅助工具
WiNet
语音管理
动态地址转换
DMZ主机
内部服务器
应用层协议检测
连接数限制
创建虚拟服务器
接口
Cellular0/0
协议类型
TCP
UDP
外部IP地址
当前接口IP地址
外部端口
Other
(0－65535，0表示任意端口)
内部IP地址
内部端口
Other
(0－65535，0表示任意端口)
确定
选择要删除的内部服务器
接口 外部IP地址 外部端口 内部IP地址 内部端口 协议类型
GigabitEthernet0/0 124.160.19.190 5001 172.18.3.106 5001 TCP
GigabitEthernet0/0 124.160.19.190 8088 172.18.2.200 8088 TCP
GigabitEthernet0/0 124.160.19.190 8111 172.18.2.149 8111 TCP
GigabitEthernet0/0 124.160.19.190 5000 172.18.2.102 8035 TCP
GigabitEthernet0/0 124.160.19.190 8034 172.18.2.100 8034 TCP
GigabitEthernet0/0 124.160.19.190 8053 172.18.2.101 8053 TCP
GigabitEthernet0/0 124.160.19.190 2323 172.18.2.213 2323 TCP
全部选中
全部取消
删除

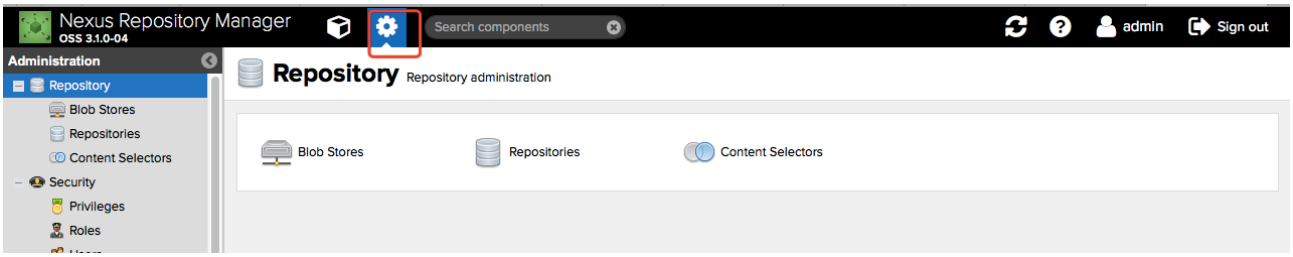
4、备份

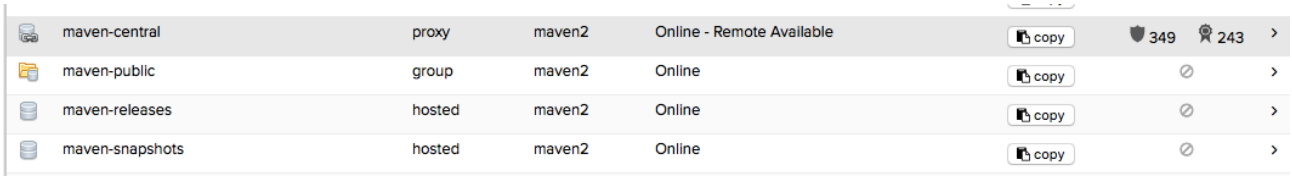

H3C
Web Management Platform
系统管理 > 配置管理
保存 | 帮助 | 退出
带宽控制
高级配置
VPN
IPsec VPN
L2TP
L2TP配置
隧道信息
GRE
SSL VPN
3G&4G
证书管理
PKI实体
PKI域
证书
CRL
系统管理
基本配置
配置管理
设备重启
软件升级
服务管理
用户管理
时间设置
TR-069
辅助工具
WiNet
保存配置
初始化
配置备份
配置恢复
U盘备份和恢复
保存当前配置
注意：保存当前配置信息。
保存出厂配置
注意：将当前配置信息保存为出厂配置信息。

四、MAVA 仓库

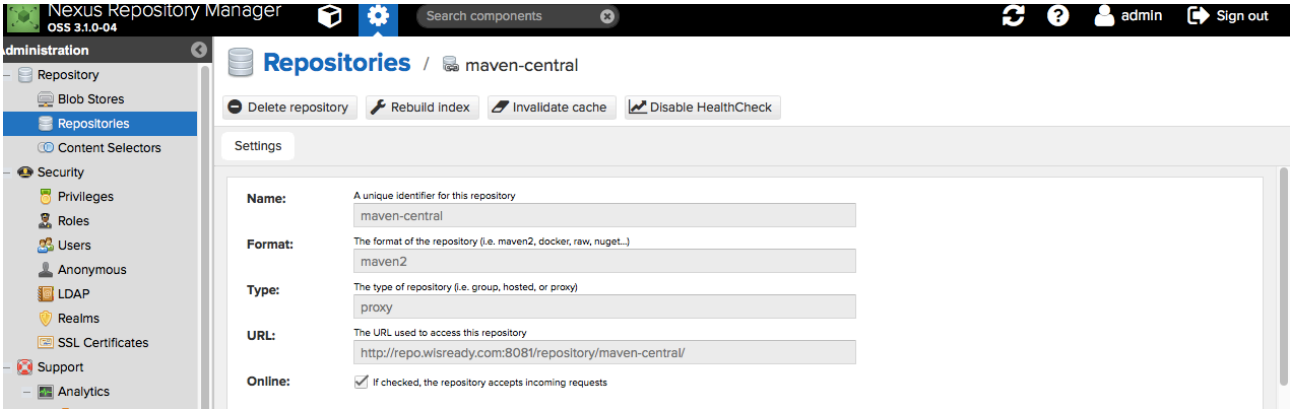
http://repo.wisready.com:8081

帐号 admin 密码：nipc#123





1、代理



Maven 2

Version policy:

What type of artifacts does this repository store?

Release

Layout policy:

Validate that all paths are maven artifact or metadata paths

Permissive

Proxy

Remote storage:

Location of the remote repository being proxied

https://repo1.maven.org/maven2/

Use the Nexus truststore:

☐ Use certificates stored in the Nexus truststore to connect to external systems View certificate

Blocked:

☐ Block outbound connections on the repository

Auto blocking enabled:

☑ Auto-block outbound connections on the repository if remote peer is detected as unreachable/unresponsive

Maximum component age:

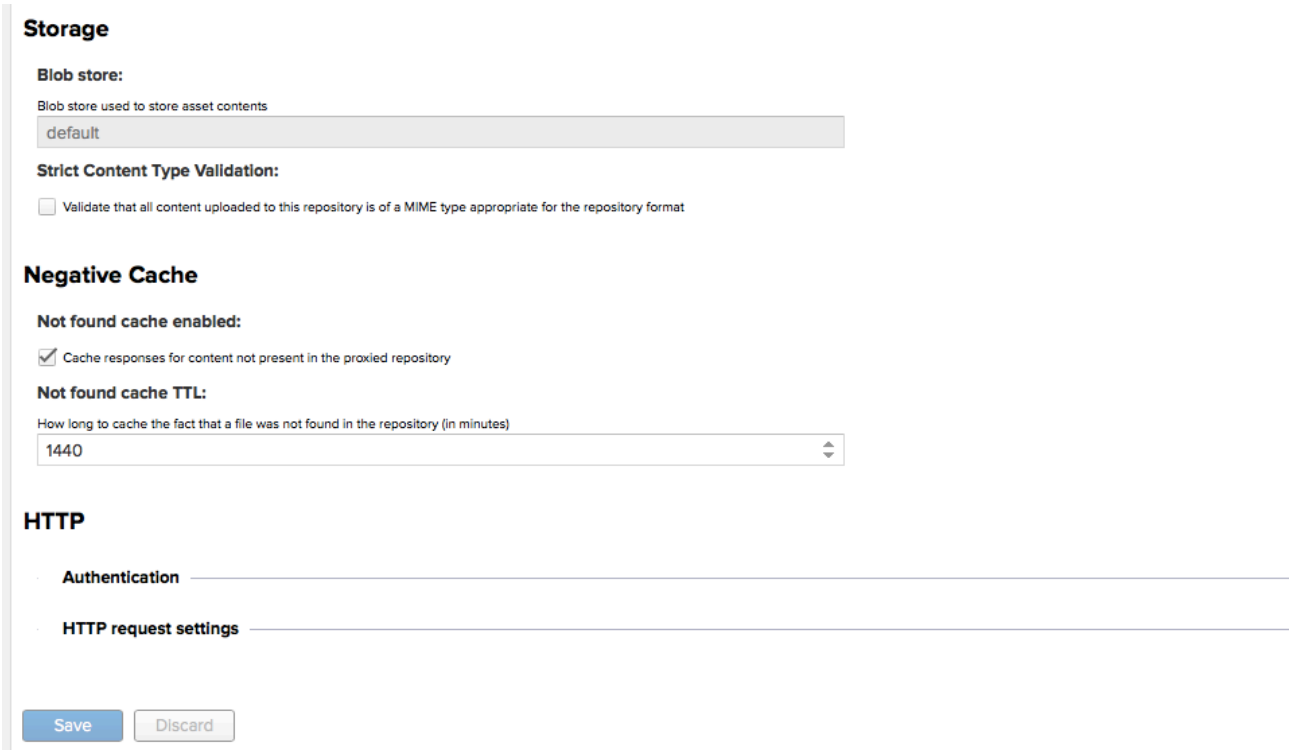
How long (in minutes) to cache artifacts before rechecking the remote repository. Release repositories should use -1.

-1

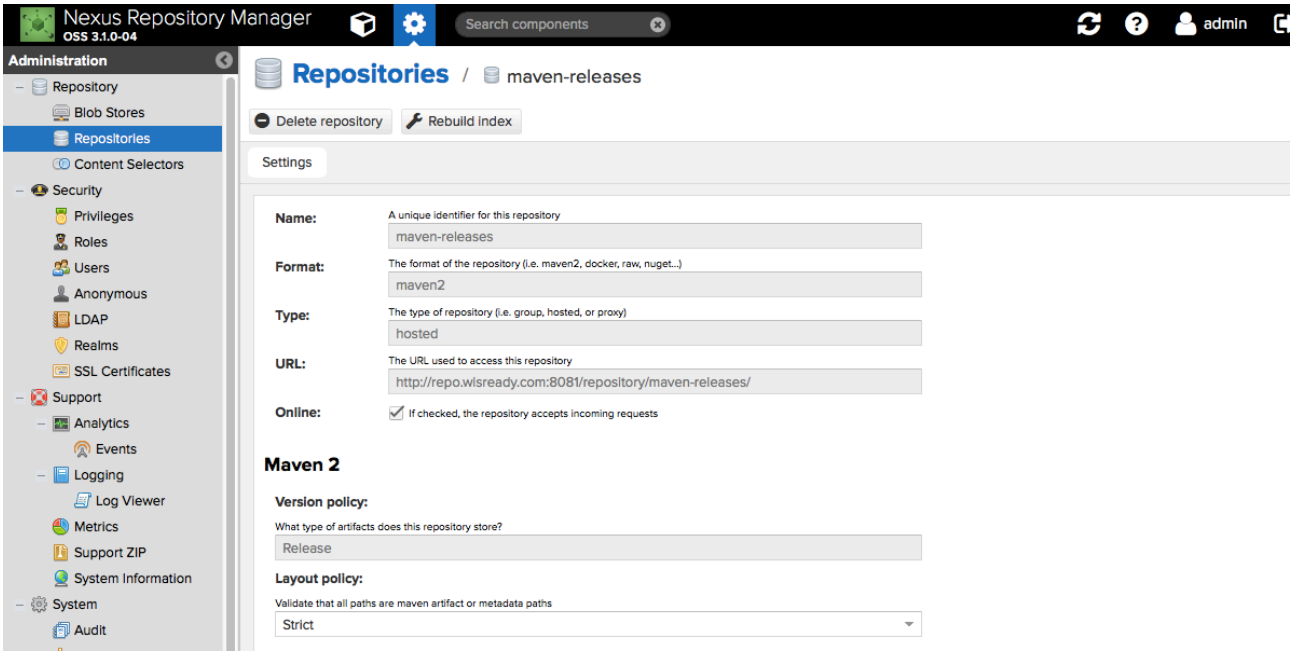
Maximum metadata age:

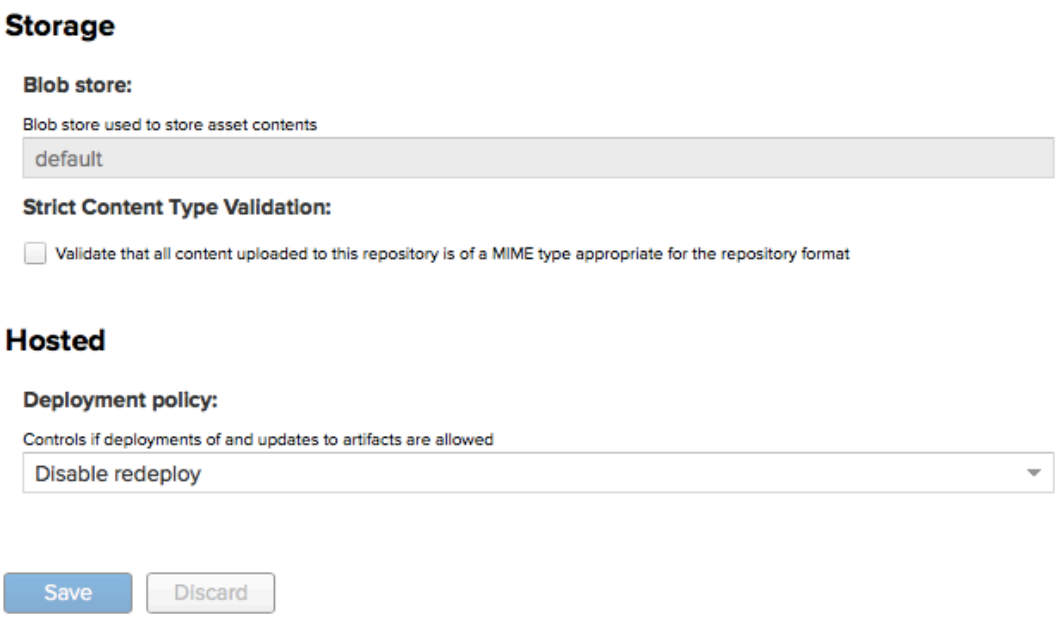
How long (in minutes) to cache metadata before rechecking the remote repository.

1440

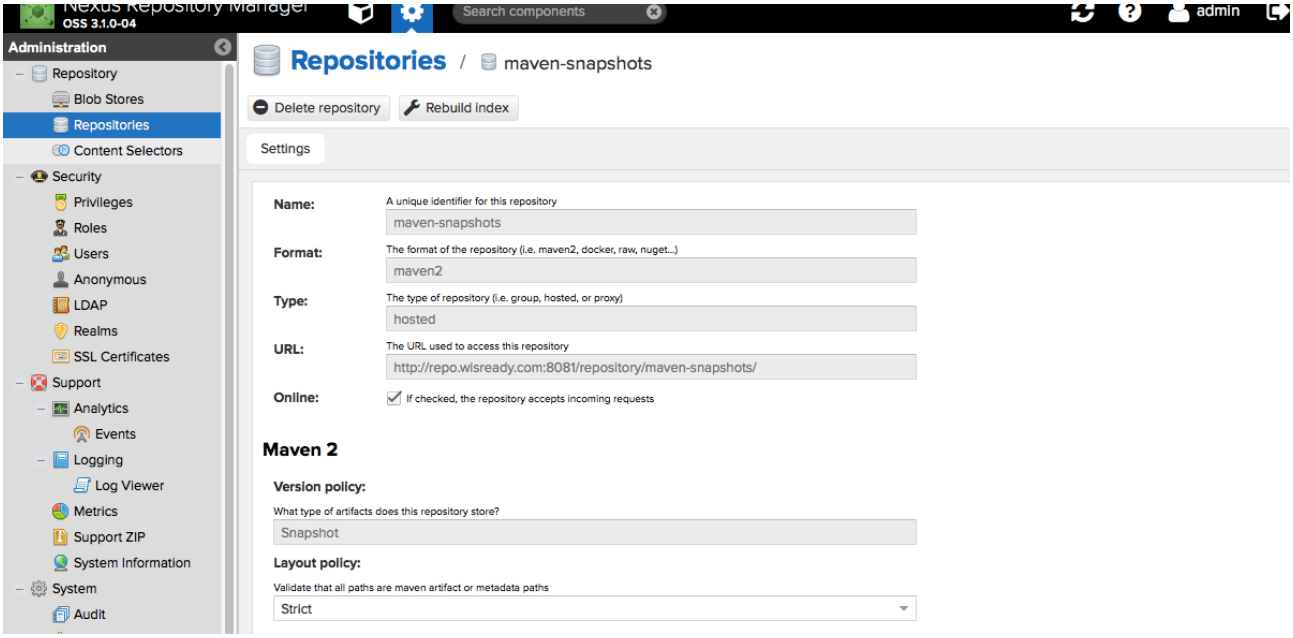

Storage
Blob store:
Blob store used to store asset contents
default
Strict Content Type Validation:
Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format
Negative Cache
Not found cache enabled:
Cache responses for content not present in the proxied repository
Not found cache TTL:
How long to cache the fact that a file was not found in the repository (in minutes)
1440
HTTP
Authentication
HTTP request settings
Save
Discard

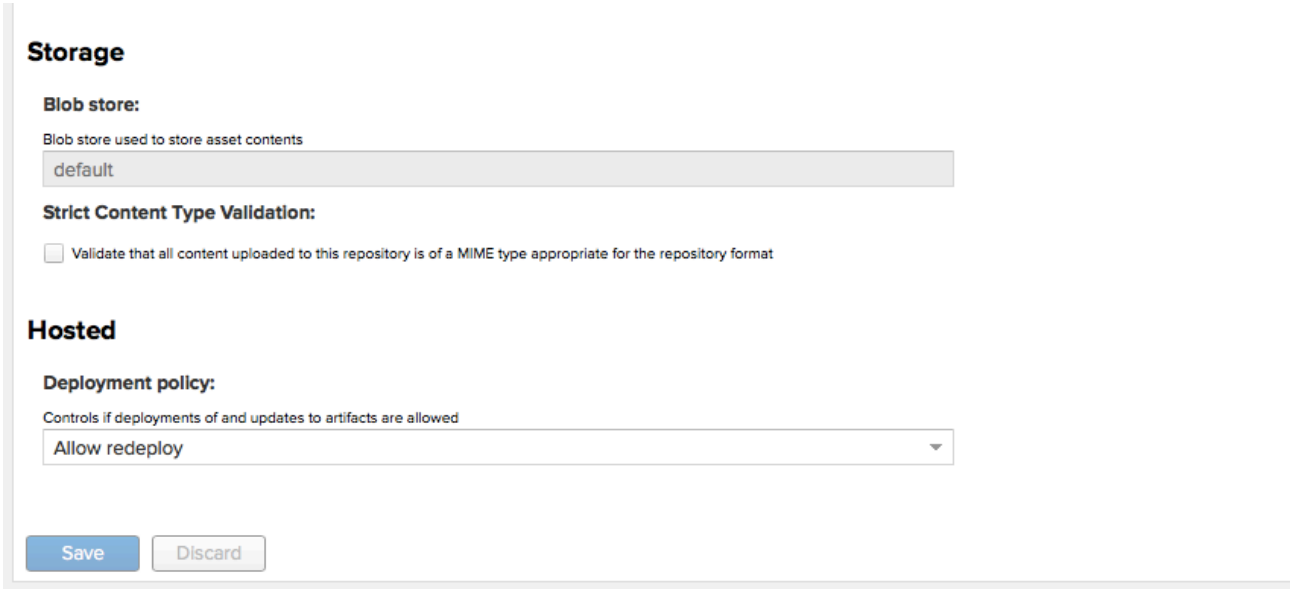
2、release


Nexus Repository Manager
OSS 3.1.0-04
Search components
admin
Administration
Repository
Blob Stores
Repositories
Content Selectors
Security
Privileges
Roles
Users
Anonymous
LDAP
Realms
SSL Certificates
Support
Analytics
Events
Logging
Log Viewer
Metrics
Support ZIP
System Information
System
Audit
Repositories / maven-releases
Delete repository
Rebuild index
Settings
Name:
A unique identifier for this repository
maven-releases
Format:
The format of the repository (i.e. maven2, docker, raw, nuget...)
maven2
Type:
The type of repository (i.e. group, hosted, or proxy)
hosted
URL:
The URL used to access this repository
http://repo.wisready.com:8081/repository/maven-releases/
Online:
If checked, the repository accepts incoming requests
Maven 2
Version policy:
What type of artifacts does this repository store?
Release
Layout policy:
Validate that all paths are maven artifact or metadata paths
Strict

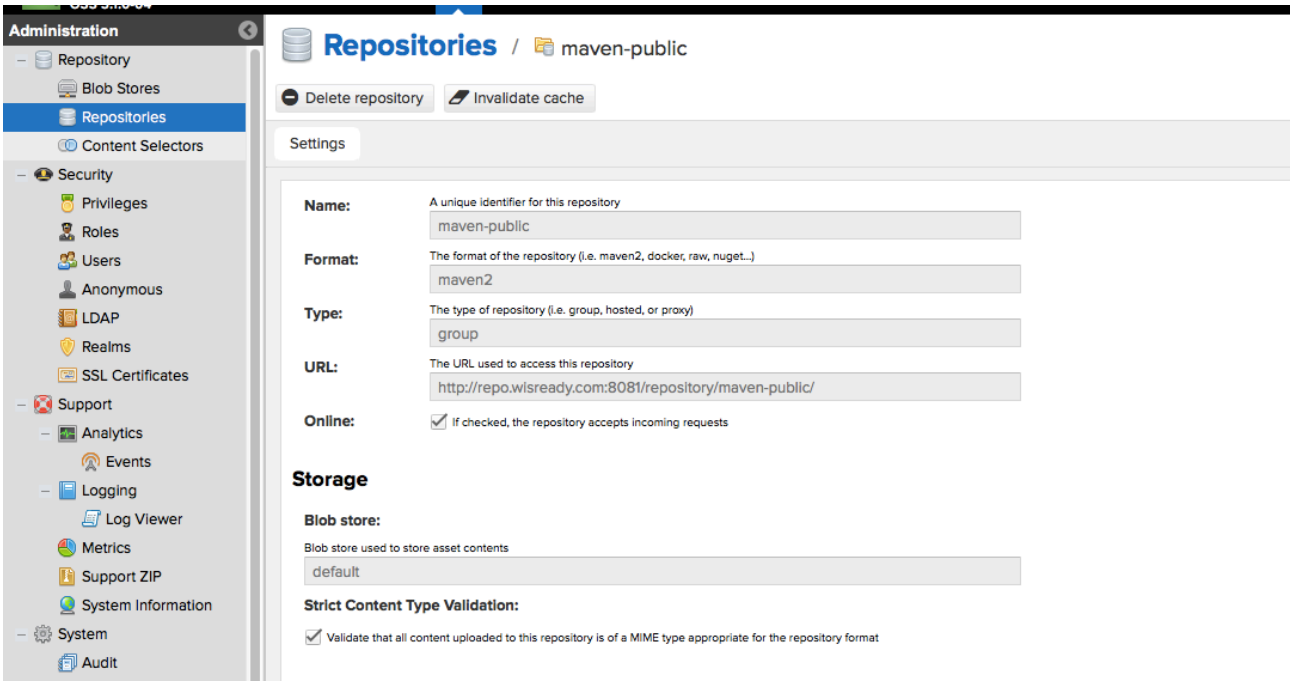

Storage
Blob store:
Blob store used to store asset contents
default
Strict Content Type Validation:
Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format
Hosted
Deployment policy:
Controls if deployments of and updates to artifacts are allowed
Disable redeploy
Save
Discard

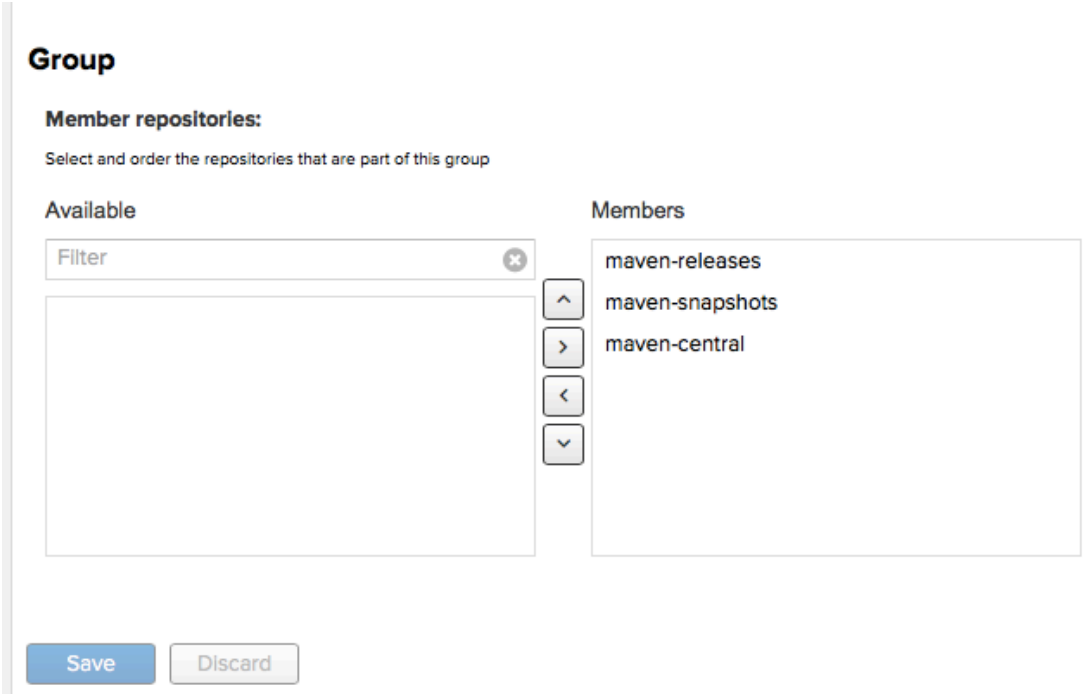
3\snapshots


Nexus Repository Manager
OSS 3.1.0-04
Search components
admin
Administration
Repository
Blob Stores
Repositories
Content Selectors
Security
Privileges
Roles
Users
Anonymous
LDAP
Realms
SSL Certificates
Support
Analytics
Events
Logging
Log Viewer
Metrics
Support ZIP
System Information
System
Audit
Repositories / maven-snapshots
Delete repository
Rebuild index
Settings
Name:
A unique identifier for this repository
maven-snapshots
Format:
The format of the repository (i.e. maven2, docker, raw, nuget...)
maven2
Type:
The type of repository (i.e. group, hosted, or proxy)
hosted
URL:
The URL used to access this repository
http://repo.wisready.com:8081/repository/maven-snapshots/
Online:
If checked, the repository accepts incoming requests
Maven 2
Version policy:
What type of artifacts does this repository store?
Snapshot
Layout policy:
Validate that all paths are maven artifact or metadata paths
Strict


Storage
Blob store:
Blob store used to store asset contents
default
Strict Content Type Validation:
Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format
Hosted
Deployment policy:
Controls if deployments of and updates to artifacts are allowed
Allow redeploy
Save
Discard

4\public





完成：

程序 setting.xml

```
 <servers>
    <server>
      <id>nexus</id>
      <username>admin</username>
      <password>nipc#123</password>
```

```
    </server>

    <server>
      <id>nexus release</id>
      <username>admin</username>
      <password>nipc#123</password>
    </server>

    <server>
      <id>nexus snapshot</id>
      <username>admin</username>
      <password>nipc#123</password>
    </server>

  <mirror>
      <id>nexus</id>
      <mirrorOf>central</mirrorOf>
      <name>nexus</name>
      <url>http://repo.wisready.com:8081/repository/maven-public/</url>
```

完

五 gitlab

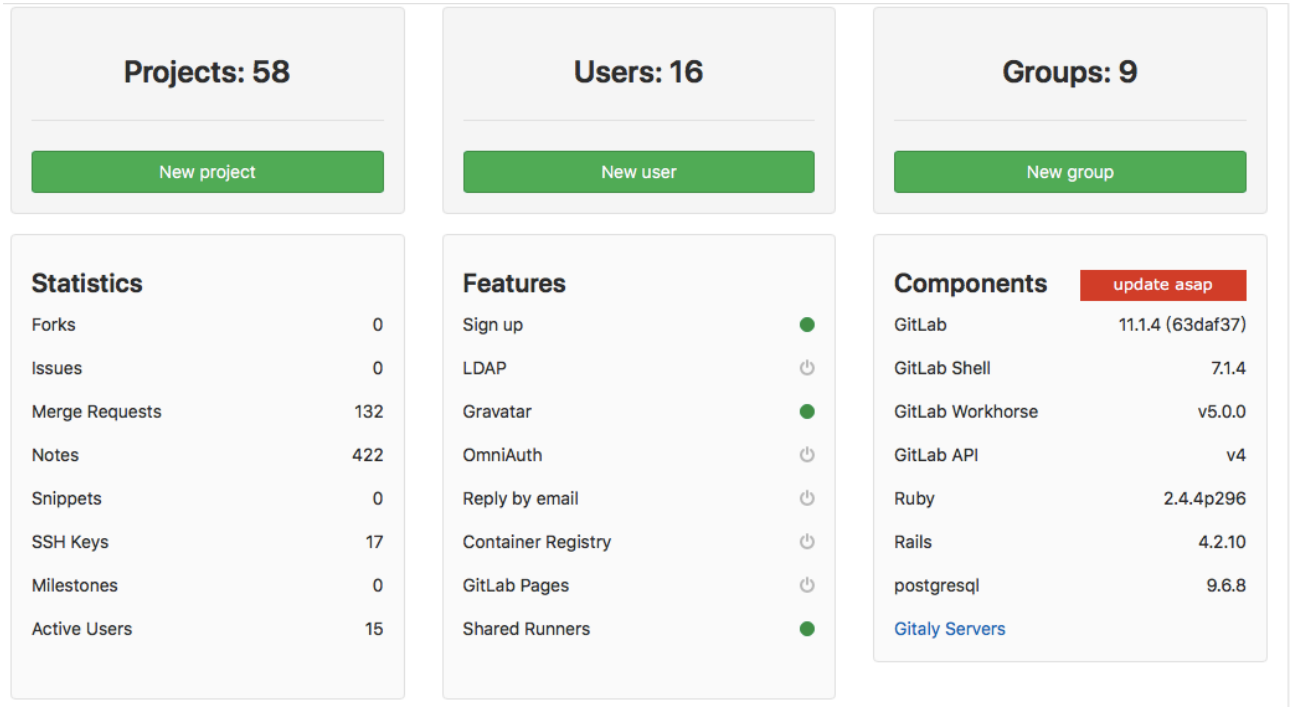
http://code.wisready.com
帐号 root 密码 nipc#123

备份脚本：

```
dir=/newcode/
dirlist=`ls $dir`
for element in $dirlist
do
cd $dir
        if test -d $element; then
        dir2=$dir/$element
        dir2list=`ls $dir2`
        for i in $dir2list
          do
            cd $dir2/$i
            git branch -r | grep -v '\->' | while read remote; do git branch --track
"${remote#origin/}" "$remote"; done
            git fetch --all
            git pull --all
          done
        fi
```

获取 git 仓库

```
#!/bin/bash
#
#
dir=/var/opt/gitlab/git-data/repositories
dirlist=`ls $dir`
for element in $dirlist
do
cd $dir
        if test -d $element; then

            dir2=$dir/$element
            dir2list=`ls $dir2`
            for i in $dir2list
            do
                echo git@code.gomrwind.com:$element/$i
            done
        fi
done

############

done
```



gitlab 管理
gitlab 创建项目之前进行分组，分组后，所有的项目都应该已组为单位创建。否者时间长了会乱。

6、docker (gitlab,jenkins,rabbitmq,dns,nginx,mysql,zookeeper,kafka,mongdb)
dockerfile
docker 网络
docker 日志监控
dns 解析
网络配置信息

```
1 apiVersion: v1        //描述 RC 对象的版本是 v1
2 kind: ReplicationController     //我现在在声明 RC 对象
3 metadata:      //metadata 中的是对此 RC 对象描述信息
4    name: myweb       //此 RC 对象在 default 命名空间中名为 myweb，同一个命名空间中的命名一定是不同的
5 spec:      //spec 中是对 RC 对象的具体描述
6    replicas: 5     //我要创建 5 个副本，单位当然是 pod
```

```
  selector:        //选择器，用来选择对象的
    app: myweb     //我选择了标签为 app: myweb 的 pod
  template:        //模版，以下用来描述创建的 pod 的模版
    metadata:      //对 pod 模版描述的元数据
      labels:      //给以下的东西打上标签，以让 selector 来选择
        app: myweb   //给 pod 模版打上 app: myweb 这样的标签
    spec:              //对 pod 模版的具体描述
      containers:      //以下就是要放入 pod 模版中的容器了
      - image: kubeguide/tomcat-app:v1    //选择镜像
        name: myweb          //容器名
        resources:           //给该容器分配的资源大小
          limits:
            cpu: "2"
            memory: 4Gi
        ports:         //容器端口号
        - containerPort: 8080
        env:           //给该容器设置环境变量，这里就可以将 mysql 与我们的 tomcat 连接
        - name: MYSQL_SERVICE_HOST
          value: 'mysql'
        - name: MYSQL_SERVICE_PORT
          value: '3306'
```

```
apiVersion: v1
kind: Service   //对象是 Service 了哦
metadata:
  name: myweb
spec:
  ports:
```

```
  7    - name: myweb-svc          //端口名称，Service 是必须指定端口名称的
  8      port: 8080               //Service 的端口号
  9      targetPort: 8080         //容器暴露的端口号
 10      nodePort: 31111          //node 的真实端口号
 11    selector:
 12      app: myweb               //Service 选择了标签为 app: myweb 的 pod
 13    type: NodePort
```

```
localhost:~ zhuxinghong$ ssh root@elk.wisready.com
root@elk.wisready.com's password:
Last login: Tue Dec  4 09:40:38 2018 from 124.160.19.190
Welcome to Alibaba Cloud Elastic Compute Service !
[root@izj6cazj9xjvk1dtkklhf5z ~]# ls
backup     docker_file               elasticsearch-head-
master  gogit.tar                 kibana-5.6.5-linux-
x86_64.tar.gz  nginx.tar    package-lock.json    rydbackup
bakup        dump.rdb                  filebeat-5.6.5-x86_64.rpm  jdk-
8u65-linux-x64.tar.gz  logstash-
5.6.5.tar.gz           node_modules  postgres.sql        smb.tar
dead.letter  elasticsearch-
5.6.5.tar.gz  file.jar                kafka.tar                master.
zip                         nohup.out     redis-4.0.10.tar.gz
[root@izj6cazj9xjvk1dtkklhf5z ~]# cd /windfoceprod
[root@izj6cazj9xjvk1dtkklhf5z windfoceprod]# ls
attendance  deliverylyd  expresslyd  go  hzdelivery  hzdispatch  hzexpress
hzopenapi  iceland  openapilyd  spring_cloud  static  tw  windbase
[root@izj6cazj9xjvk1dtkklhf5z windfoceprod]#
[root@izj6cazj9xjvk1dtkklhf5z windfoceprod]#
```

```
[root@izj6cazj9xjvk1dtkklhf5z twdispatch]# kubectl get deployment
NAME             DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
attendance       1         1         1            1           77d
deliverylyd      1         1         1            1           133d
eurekav1         1         1         1            1           87d
eurekav2         1         1         1            1           86d
eurekav3         1         1         1            1           86d
expresslyd       1         1         1            1           133d
exptracklyd      1         1         1            1           132d
gateway          2         2         2            2           86d
hzdelivery       1         1         1            1           126d
```

```
hzdispatch       1         1         1            1           10d
hzexpress        1         1         1            1           126d
hzopenapi        1         1         1            1           126d
icedelivery      1         1         1            1           78d
iceexpress       1         1         1            1           78d
iceopenapi       1         1         1            1           78d
newdeliveryweb   1         1         1            1           72d
openapilyd       1         1         1            1           133d
paysystem        1         1         1            1           72d
prodwindchat     1         1         1            1           45d
twdelivery       1         1         1            1           31d
twdispatch       1         1         1            1           7d
twexpress        1         1         1            1           31d
twopenapi        1         1         1            1           31d
versioninfo      1         1         1            1           104d
windbase         1         1         1            1           80d
windh5web        1         1         1            1           51d
windworkpage     1         1         1            1           105d
[root@izj6cazj9xjvk1dtkklhf5z twdispatch]# kubectl get pod
NAME                               READY     STATUS    RESTARTS   AGE
attendance-6db97469c-khp7g         2/2       Running   0          21h
deliverylyd-744c87c9bd-k6mn8       2/2       Running   0          16d
eurekav1-6c8f6bc7dd-cb8dc          2/2       Running   0          48d
eurekav2-67dc5c85f6-l7qrm          2/2       Running   0          48d
eurekav3-548cb9d9b4-6n42t          2/2       Running   0          48d
expresslyd-854fff7d55-7r65w        2/2       Running   0          17d
exptracklyd-744c8f7456-dvbjg       2/2       Running   0          64d
gateway-f96cbc8f7-hcr98            2/2       Running   0          50d
gateway-f96cbc8f7-hn89t            2/2       Running   0          50d
hzdelivery-7446966678-tcq2q        2/2       Running   0          3d
hzdispatch-5789b9dc7d-6pwlz        2/2       Running   0          16h
hzexpress-7ddf7cfd-2nz2n           2/2       Running   0          18h
hzopenapi-79c859969b-k7k89         2/2       Running   0          2d
icedelivery-688c677fbb-rmsfc       2/2       Running   0          5d
iceexpress-57b87f8875-8cm72        2/2       Running   0          1d
iceopenapi-57d89f58d4-sdbvt        2/2       Running   0          5d
newdeliveryweb-59c4699bdc-54zp8    2/2       Running   0          18h
openapilyd-8597fc956c-2h7bj        2/2       Running   0          17d
paysystem-65677954b5-fxr6w         2/2       Running   0          20d
prodwindchat-69f55bc4f8-f4xcs      2/2       Running   0          16h
twdelivery-5bf77db655-4jb5l        2/2       Running   0          3d
twdispatch-6544b97f69-lp7g2        2/2       Running   0          17h
twexpress-7f84c88dcf-rzl6b         2/2       Running   0          22h
twopenapi-675d74cd9-t474l          2/2       Running   0          3d
versioninfo-7c96bfb965-7zfq6       2/2       Running   0          3h
windbase-668fcb4d45-r2qmc          2/2       Running   0          18h
windh5web-6dd7c8d7b9-97b7n         2/2       Running   0          50d
windworkpage-8486cb7f9-4h5hr       2/2       Running   0          20h
[root@izj6cazj9xjvk1dtkklhf5z twdispatch]# kubectl get ingress
NAME                 HOSTS
                                  ADDRESS       PORTS     AGE
nginxstatic          expresstracking.gomrwind.com,express.wisready.com,wi
ndforce.wisready.com + 6 more...   47.91.161.5   80        132d
windbase-
attatench   hk.wisready.com
                          47.91.161.5   80        72d
```

```
[root@izj6cazj9xjvk1dtkklhf5z twdispatch]# kubectl  get svc
NAME                       TYPE           CLUSTER-IP      EXTERNAL-
IP     PORT(S)                         AGE
attendance-
nodeport       NodePort       172.23.4.36     <none>          80:30897/T
CP                  72d
eurekav1                   LoadBalancer   172.23.1.108    47.52.84.40
8111:31317/TCP,80:30896/TCP   86d
eurekav2                   NodePort       172.23.4.143    <none>
8112:30401/TCP                86d
eurekav3                   NodePort       172.23.14.163   <none>
8113:30296/TCP                86d
exptralyd-
nodeport        NodePort       172.23.12.229   <none>          80:30083/
TCP                 132d
gateway                    LoadBalancer   172.23.9.230    47.52.125.57
80:32612/TCP                  72d
hz-delivery-
internet      LoadBalancer   172.23.15.150   47.91.174.51    80:31904/TC
P                 111d
hz-dispatch-
internet      LoadBalancer   172.23.1.227    47.52.49.162    80:31644/TC
P                 10d
hz-express-
internet       LoadBalancer   172.23.14.51    47.91.161.164   80:31596/T
CP                 111d
hz-openapi-
internet       LoadBalancer   172.23.15.148   47.89.57.21     80:30423/T
CP                 111d
ice-delivery-
internet     LoadBalancer   172.23.8.96     47.52.83.101    80:32406/TCP
                78d
ice-express-
internet      LoadBalancer   172.23.6.178    47.52.83.102    80:31511/TC
P                 78d
ice-openapi-
internet      LoadBalancer   172.23.6.4      47.52.83.106    80:32469/TC
P,5000:32065/TCP   78d
kubernetes                 ClusterIP      172.23.0.1      <none>
443/TCP                       134d
lyd-delivery-
internet     LoadBalancer   172.23.7.27     47.52.55.50     80:32547/TCP
                111d
lyd-expresslyd-
internet   LoadBalancer   172.23.13.203   47.52.55.152    80:31701/TCP
              111d
lyd-openapi-
internet      LoadBalancer   172.23.0.160    47.52.55.140    80:32060/TC
P,5000:31795/TCP   111d
newdeliveryweb-
nodeport   NodePort       172.23.7.137    <none>          80:32488/TCP
              72d
paysystem                  LoadBalancer   172.23.3.194    47.52.80.28
80:32482/TCP                  73d
```

```
tw-delivery-
internet      LoadBalancer   172.23.8.111    47.52.183.140   80:30621/TC
P                 31d
tw-dispatch-
internet      LoadBalancer   172.23.9.227    47.52.125.125   80:32264/TC
P                 7d
tw-express-
internet       LoadBalancer   172.23.14.78    47.91.174.28    80:32395/T
CP                30d
tw-openapi-
internet       LoadBalancer   172.23.5.132    47.91.160.186   80:31779/T
CP                31d
versioninfoweb-
nodeport   NodePort       172.23.13.81    <none>          80:32389/TCP
               104d
windbase-
nodeport         NodePort       172.23.11.43    <none>          80:32059
/TCP                72d
windchat-
nodeport         NodePort       172.23.3.56     <none>          80:30255
/TCP                70d
windh5web-
nodeport        NodePort       172.23.10.84    <none>          80:30039/
TCP                51d
windpush-
nodeport         NodePort       172.23.6.206    <none>          80:31765
/TCP                70d
windworkpage-
internet     LoadBalancer   172.23.8.75     47.52.124.224   80:30031/TCP
                105d
[root@izj6cazj9xjvk1dtkklhf5z twdispatch]# kubectl  get secret
NAME                  TYPE                                  DATA      AG
E
default-token-x95xh   kubernetes.io/service-account-
token   3         134d
regcredv1             kubernetes.io/dockerconfigjson        1         111
d
regcredv2             kubernetes.io/dockerconfigjson        1         133
d


[root@izj6cazj9xjvk1dtkklhf5z twdispatch]# kubectl  exec -it hzexpress-
7ddf7cfd-2nz2n  -c hzexpress bash
root@hzexpress-7ddf7cfd-2nz2n:~#
root@hzexpress-7ddf7cfd-2nz2n:~#
root@hzexpress-7ddf7cfd-2nz2n:~#
root@hzexpress-7ddf7cfd-2nz2n:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: eth0@if553: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP group default
    link/ether 0a:58:ac:16:02:60 brd ff:ff:ff:ff:ff:ff
```

```
    inet 172.22.2.96/25 scope global eth0
       valid_lft forever preferred_lft forever
root@hzexpress-7ddf7cfd-2nz2n:~# exit
exit
[root@izj6cazj9xjvk1dtkklhf5z twdispatch]# kubectl  exec -it hzexpress-
7ddf7cfd-2nz2n  -c filebeat bash
root@hzexpress-7ddf7cfd-2nz2n:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: eth0@if553: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP group default
    link/ether 0a:58:ac:16:02:60 brd ff:ff:ff:ff:ff:ff
    inet 172.22.2.96/25 scope global eth0
       valid_lft forever preferred_lft forever
```

8、FELK+redis

ElasticSearch

```
[root@izj6cazj9xjvk1dtkklhf5z elasticsearch]# cat
config/elasticsearch.yml |grep -v ^#
cluster.name: fengapp
node.name: node-1
path.data: /elkdata/data
path.logs: /elkdata/logs
bootstrap.memory_lock: false
bootstrap.system_call_filter: false
network.host: 10.86.32.128
http.port: 9200
discovery.zen.ping.unicast.hosts: ["10.86.32.128"]

http.cors.enabled: true
http.cors.allow-origin: "*"
```

Logstash

```
[root@izj6cazj9xjvk1dtkklhf5z elasticsearch]#
cat ../logstash/config/redis.yml
input {
    redis {
        host => "10.86.32.128"
        port => 6379
        key => "filebeat"
#        type => "nginx"
        data_type => "list"
    }
}
```

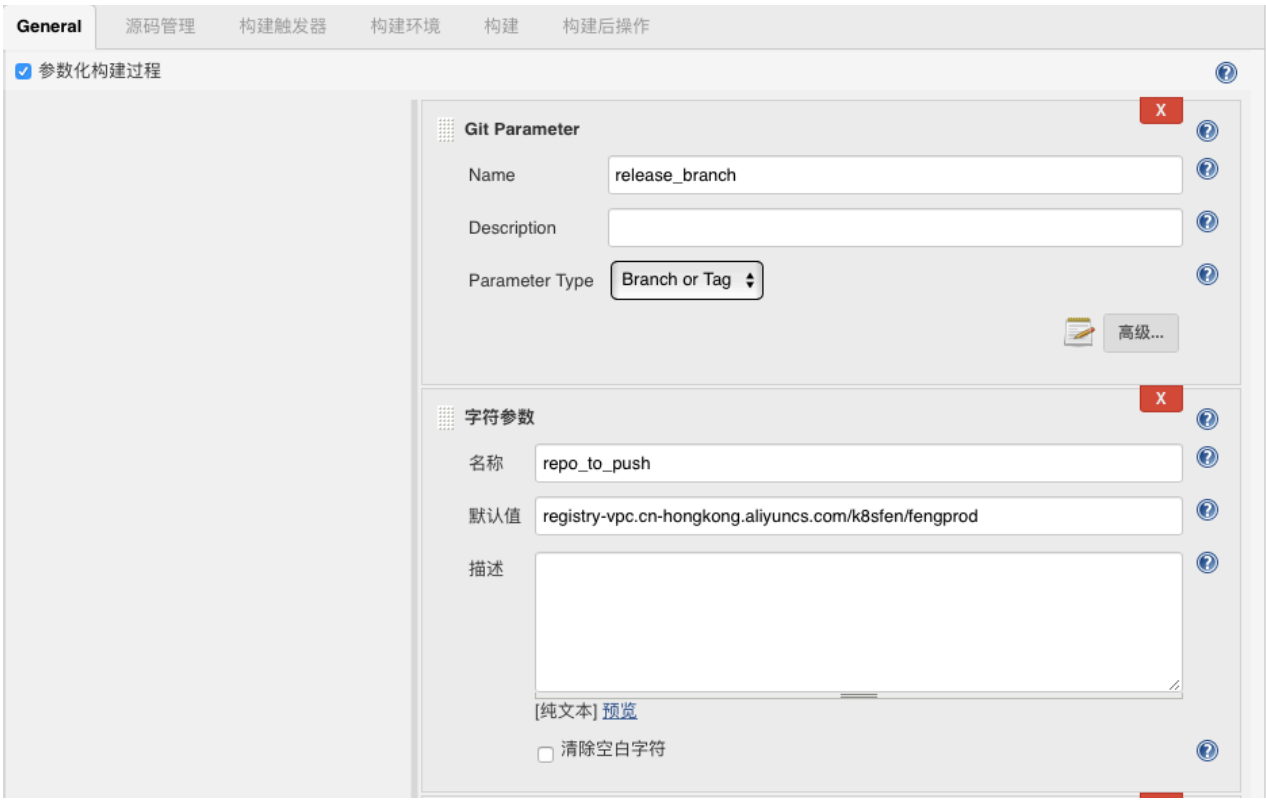
```
output {
    if[type] =="linux.system"{
    elasticsearch {
        hosts => [ "10.86.32.128:9200" ]
    index => "message-%{+YYYY.MM.dd}"
        document_type => "%{[@metadata][type]}"
        flush_size => 20000
        idle_flush_time => 10
        template_overwrite => true
    }
    }else if[type] =="k8s-logs"{
        elasticsearch {
                    hosts => [ "10.86.32.128:9200" ]
                index => "docker-%{+YYYY.MM.dd}"
                    document_type => "%{[@metadata][type]}"
                    flush_size => 20000
                    idle_flush_time => 10
                    template_overwrite => true
    }
    }



      stdout { codec => rubydebug }

 }
```

Kibana

```
[root@izj6cazj9xjvk1dtkklhf5z elasticsearch]#
cat ../kibana/config/kibana.yml |grep -v ^#
server.port: 5601
server.host: "10.86.32.128"
elasticsearch.url: "http://10.86.32.128:9200"
kibana.index: ".kibana"
sentinl:
  settings:
    email:
       active: true
       user: gitlab@gomrwind.com
       password: gitlab,2018
       host: smtp.gomrwind.com
       ssl: true
    report:
       active: true
       tmp_path: /tmp/
```

jenkins 项目配置：

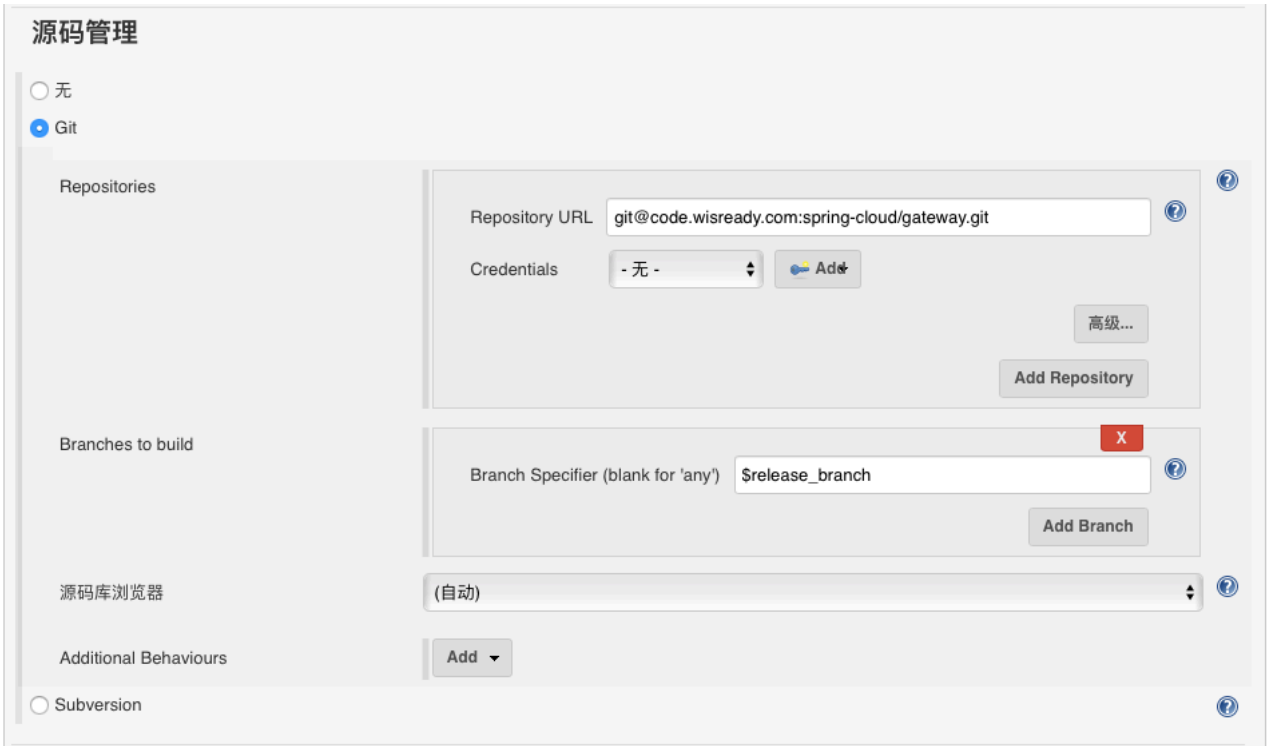
1、创建项目
第一步：添加参数化构建


General
源码管理
构建触发器
构建环境
构建
构建后操作
参数化构建过程
Git Parameter
Name
release_branch
Description
Parameter Type
Branch or Tag
高级...
字符参数
名称
repo_to_push
默认值
registry-vpc.cn-hongkong.aliyuncs.com/k8sfen/fengprod
描述
[纯文本] 预览
清除空白字符

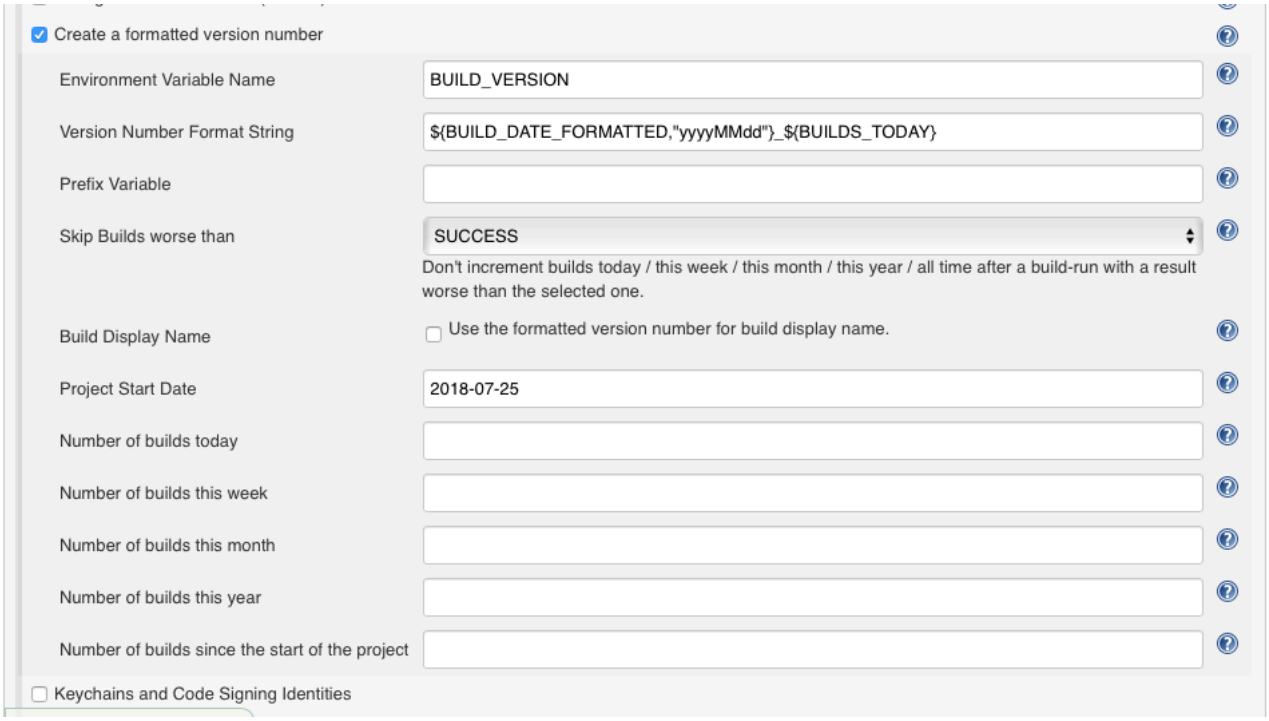

选项参数
名称
image
选项
gateway
描述
网关中心
[纯文本] 预览
选项参数
名称
status
选项
Deploy
RollBACK
描述
Deploy: 发布
RollBACK: 回滚


字符参数
名称
oldversion
默认值
描述
填写回滚的版本号
[纯文本] 预览
清除空白字符
添加参数

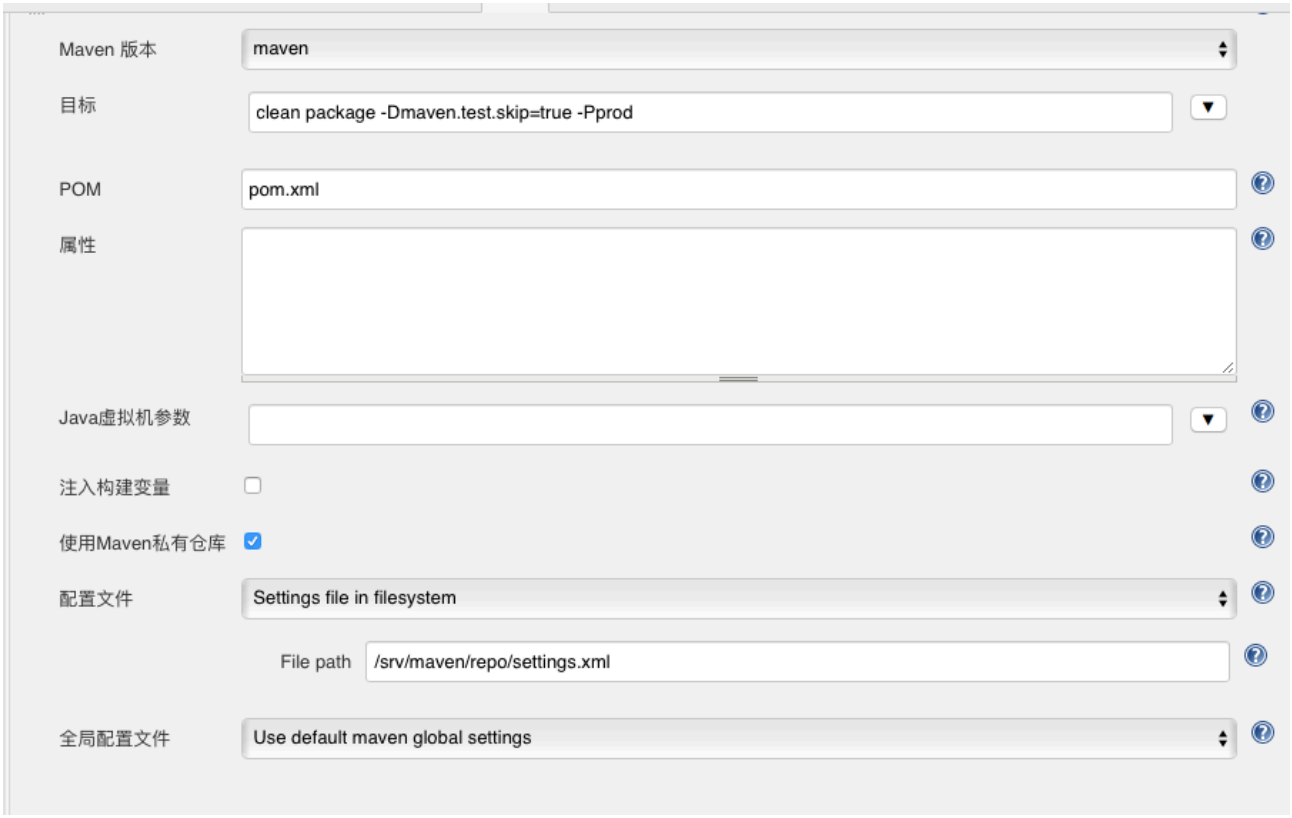
第二步：添加仓库


源码管理
无
Git
Repositories
Repository URL
git@code.wisready.com:spring-cloud/gateway.git
Credentials
- 无 -
Add
高级...
Add Repository
Branches to build
Branch Specifier (blank for 'any')
$release_branch
Add Branch
源码库浏览器
(自动)
Additional Behaviours
Add
Subversion

第三步 创建版本控制

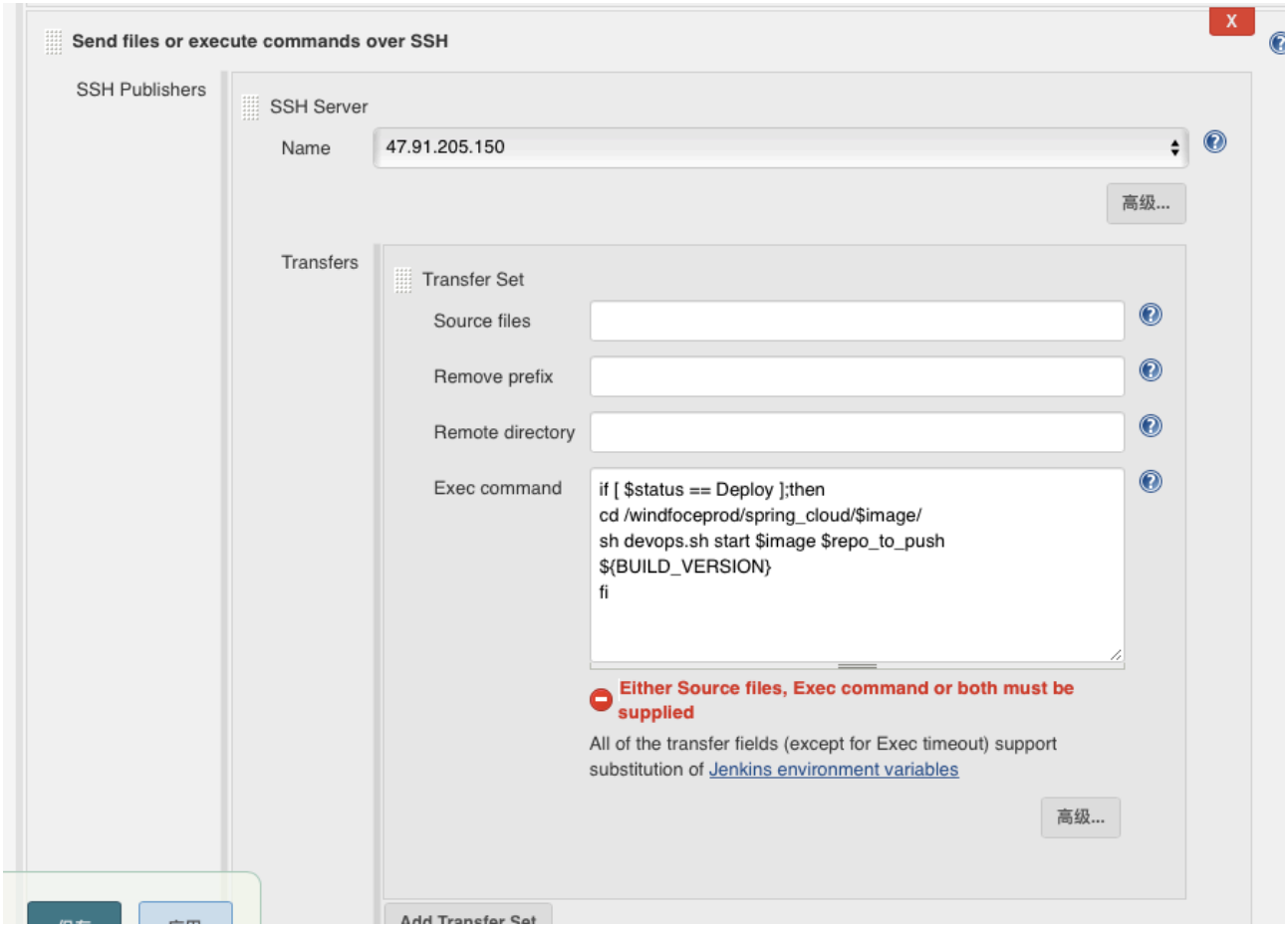


第四步：mvn 构建



第五步：执行 shell 脚本

第六步：部署到 k8s




```
zhuxinghongdeMacBook-Air:newcode zhuxinghong$ ssh root@elk.wisready.com
root@elk.wisready.com's password:
Last login: Tue Dec 11 19:21:51 2018 from 124.160.19.190

Welcome to Alibaba Cloud Elastic Compute Service !
```

登入服务器：查看脚本

```
[root@izj6cazj9xjvk1dtkklhf5z gateway]# ls
cmsdev-pod-v1.yaml  delivery-svc-v3.yaml  devops.sh  Dockerfile
file.jar
[root@izj6cazj9xjvk1dtkklhf5z gateway]#
```

看部署脚本

```
[root@izj6cazj9xjvk1dtkklhf5z gateway]# cat devops.sh
#!/bin/bash
#
```

```
image="$2"
repo_to_push="$3"
version="$4"

#DIP='registry.cn-hongkong.aliyuncs.com'
#DIMAGE_USER='d1@123feng.com'
#DUSER_PASSWD='feng2018hz'
#docker login -u ${DIMAGE_USER} -p ${DUSER_PASSWD}  ${DIP}

for i in `docker images|grep none|awk -F ' ' '{print $3}'`; do docker
rmi $i; done

rallback(){
kubectl set image Deployment/${image} ${image}=${repo_to_push}:${image}-
${version}
}

BuildImage () {

############################定义目录###################################
BaseDir='/windfoceprod'
DockerFileDir=`find $BaseDir -name $image`

if [ ! -d ${DockerFileDir} ]; then
        mkdir -p ${DOCKER_FILE_DIR}
fi

DockerFILE="$DockerFileDir/Dockerfile"
########################################################################

# check docker images
DOCKER_IMAGE=`docker images | grep ${image} | awk '{print $3}'`
if [ -n "${DOCKER_IMAGE}" ]; then
        # check docker container
        for dc in `docker ps -a | grep ${image} | awk '{print $1}'`
        do
                echo "Stop container: ${dc}"
                docker stop ${dc}
                # delete while docker container was exists
                echo "##Delete exists Container_Id: "${dc}
                docker rm ${dc}
        done

        # delete while docker image was exists
        echo "#Delete exists Image: "${image}
        docker rmi --force ${DOCKER_IMAGE}
fi


#Build for linux amd64
cd $DockerFileDir

#echo "FROM jdk:1.8" > ${DockerFILE}
#echo "ADD  file.jar /app.jar" >>  ${DockerFILE}
#echo "RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime &&
echo 'Asia/Shanghai' >/etc/timezone" >> ${DockerFILE}
```

```
#echo "EXPOSE 8034" >> ${DockerFILE}
#echo "ENTRYPOINT [\"java\",\"-server\",\"-Xms1024m\",\"-Xmx1024m\",\"-
Xmn384m\",\"-XX:+UseCompressedOops\",\"-XX:SurvivorRatio=5\",\"-
XX:+UseConcMarkSweepGC\",\"-XX:+UseCMSCompactAtFullCollection\",\"-
XX:CMSMaxAbortablePrecleanTime=5000\",\"-
XX:+CMSClassUnloadingEnabled\",\"-
XX:CMSInitiatingOccupancyFraction=80\",\"-
XX:+UseCMSInitiatingOccupancyOnly\",\"-
XX:+CMSParallelRemarkEnabled\",\"-XX:+CMSScavengeBeforeRemark\",\"-
Duser.timezone=Asia/Riyadh\",\"-
Djava.security.egd=file:/dev/./urandom\",\"-jar\",\"/app.jar\"]" >>
${DockerFILE}

docker build -t ${repo_to_push}:${image}-${version} .

#push to aliyun

docker push ${repo_to_push}:${image}-${version}

kubectl set image Deployment/${image} ${image}=${repo_to_push}:${image}-
${version}

}


case "$1" in
  start)
    BuildImage
    ;;
  renew)
    rallback
    ;;
  *)
    echo $"Usage: $0 {start|renew}"
    exit 1
esac


安装 pod
[root@izj6cazj9xjvk1dtkklhf5z gateway]# cat cmsdev-pod-v1.yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: gateway
spec:
  replicas: 2
  selector:
    matchLabels:
      app: gateway
  template:
    metadata:
      labels:
        app: gateway
    spec:
      imagePullSecrets:
        - name: regcredv1
```

```
      containers:
      - name: filebeat
        image: registry-vpc.cn-
hongkong.aliyuncs.com/k8sfen/fengprod:filebeatv1.0
        imagePullPolicy: Always
        env:
          - name: POD_NAME
            value: gateway
          - name: MY_POD_NAME
            valueFrom:
                fieldRef:
                  fieldPath: metadata.name
          - name: CONTAINER_NAME
            value: gateway
          - name: MY_POD_IP
            valueFrom:
                fieldRef:
                  fieldPath: status.podIP
        volumeMounts:
        - name: varlog
          mountPath: /var/log/containers
        - name: varlogpods
          mountPath: /var/log/pods
          readOnly: true
        - name: varlibdockercontainers
          mountPath: /var/lib/docker/containers
          readOnly: true

      - name: gateway
        image:  registry-vpc.cn-
hongkong.aliyuncs.com/k8sfen/fengprod:gateway-20180905_1
        imagePullPolicy: Always
        ports:
        - containerPort: 8080
        livenessProbe:
          httpGet:
            port: 8080
            path: /actuator/health
          initialDelaySeconds: 30
          timeoutSeconds: 5
          periodSeconds: 15

      terminationGracePeriodSeconds: 30
      volumes:
      - name: varlog
        hostPath:
          path: /var/log/containers
      - name: varlogpods
        hostPath:
          path: /var/log/pods
      - name: varlibdockercontainers
        hostPath:
          path: /var/lib/docker/containers
#######
```

```
[root@izj6cazj9xjvk1dtkklhf5z gateway]# kubectl create -f  cmsdev-pod-
v1.yaml
########

安装 service

[root@izj6cazj9xjvk1dtkklhf5z gateway]# cat delivery-svc-v3.yaml
apiVersion: v1
kind: Service
metadata:
  labels:
    app: gateway
  name: gateway
spec:
  ports:
  - port: 80
    protocol: TCP
    targetPort: 8080
  selector:
    app: gateway
  type: LoadBalancer


####
[root@izj6cazj9xjvk1dtkklhf5z gateway]# kubectl create -f  delivery-svc-
v3.yaml



####### 生产所有的信息都在这个目录下################

[root@izj6cazj9xjvk1dtkklhf5z windfoceprod]# ls
attendance  deliverylyd  expresslyd  go  hzdelivery  hzdispatch
hzexpress  hzopenapi  iceland  openapilyd  spring_cloud  static  tw
windbase
[root@izj6cazj9xjvk1dtkklhf5z windfoceprod]# pwd
/windfoceprod


####################################
服务器 10.0.0.211

####################################

UAT DEV 所有的环境都在这个目录下

[root@localhost ~]# cd /windfoce/uat/
[root@localhost uat]# ls
go  hrattendance  hrdevattendance  hrdevwindbase  hrwindbase  hz
hzuathub  node  ryd  spring_cloud  spring_cloud_dev  spring_cloud_test
static  staticuatexptrack



####3
```

[root@localhost hzdevdelivery]# cat devopsv2.sh   ### 作用是部署 dockerfile 文件

```
#!/bin/bash
#

mem1=$1
mem2=$2

#Build for linux amd64
BASENAME=/windfoce/uat/hz/hzdevdelivery

DockerFILE=${BASENAME}/Dockerfile
cd ${BASENAME}
echo "FROM jdk:1.8" > ${DockerFILE}
echo "ADD file.jar /app.jar" >>  ${DockerFILE}
echo "RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime &&
echo 'Asia/Shanghai' >/etc/timezone" >>  ${DockerFILE}
echo "EXPOSE 8034" >>  ${DockerFILE}
#echo "EXPOSE 443" >>  ${DockerFILE}
echo "ENTRYPOINT [\"java\",\"-server\",\"-Xms${mem1}m\",\"-
Xmx${mem1}m\",\"-Xmn${mem2}m\",\"-XX:+UseCompressedOops\",\"-
XX:SurvivorRatio=5\",\"-XX:+UseConcMarkSweepGC\",\"-
XX:+UseCMSCompactAtFullCollection\",\"-
XX:CMSMaxAbortablePrecleanTime=5000\",\"-
XX:+CMSClassUnloadingEnabled\",\"-
XX:CMSInitiatingOccupancyFraction=80\",\"-
XX:+UseCMSInitiatingOccupancyOnly\",\"-
XX:+CMSParallelRemarkEnabled\",\"-XX:+CMSScavengeBeforeRemark\",\"-
Djava.security.egd=file:/dev/./urandom\",\"-jar\",\"/app.jar\"]">>
${DockerFILE}
[root@localhost hzdevdelivery]# ls
devops.sh  devopsv2.sh  devopsv3.sh  devopsv3.sh.bak  Dockerfile
file.jar  iptables.conf
```

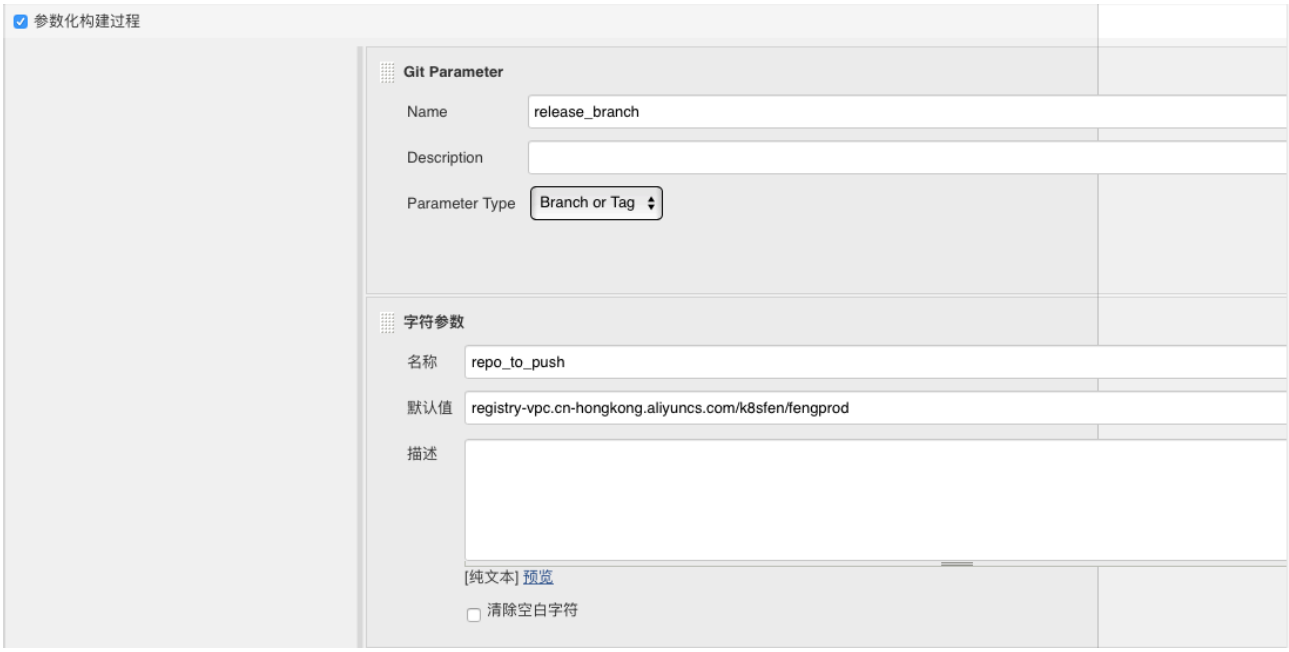
```
[root@localhost hzdevdelivery]# cat devopsv3.sh#作用是部署 docker 容器
#!/bin/bash
# this script function is :
# deploy new docker container
#
parasnum=2
# function
help_msg()
{
cat << help
+ Error Cause:
+ you enter $# parameters
+ the total paramenter number must be $parasnum
```

```
+ 1st :DOCKER_NAME
+ 2nd :DOCKER_VERSION
help
}
# Check parameter number
if [ $# -ne ${parasnum} ]
then
        help_msg
        exit
fi







# Initialize the parameter.
DOCKER_BASE=/windfoce/uat/hz/
DOCKER_NAME=$1
DOCKER_VERSION=$2
DOCKER_FILE="${DOCKER_BASE}${DOCKER_NAME}/Dockerfile"
DOCKER_FILE_DIR=${DOCKER_BASE}${DOCKER_NAME}
if [ ! -d ${DOCKER_FILE_DIR} ]; then
        mkdir -p ${DOCKER_FILE_DIR}
fi


# check docker images
DOCKER_IMAGE=`docker images | grep ${DOCKER_NAME} | awk -F ' ' '{print
$3}'`
if [ -n "${DOCKER_IMAGE}" ]; then
        # check docker container
        for dc in `docker ps -a | grep ${DOCKER_NAME} | awk -F " "
'{print $1}'`
        do
                echo "Stop container: ${dc}"
                docker stop ${dc}
                # delete while docker container was exists
                echo "##Delete exists Container_Id: "${dc}
                docker rm -f ${dc}
        done

        # delete while docker image was exists
        echo "#Delete exists Image: "${DOCKER_IMAGE}
        docker rmi ${DOCKER_IMAGE}
fi



cd ${DOCKER_FILE_DIR}
echo "##Build dockerfile for "${DOCKER_NAME}
docker build -t ${DOCKER_NAME}:${DOCKER_VERSION} .

for i in `docker images|grep none|awk -F ' ' '{print $3}'`; do docker
rmi $i; done
```

```
# Run docker container
echo ""
echo "##Running docker container: "${DOCKER_NAME}

docker run -it -d --restart unless-stopped --network mynetwork --ip
172.18.2.120 --dns=172.18.3.251 --name ${DOCKER_NAME}
${DOCKER_NAME}:${DOCKER_VERSION}
```

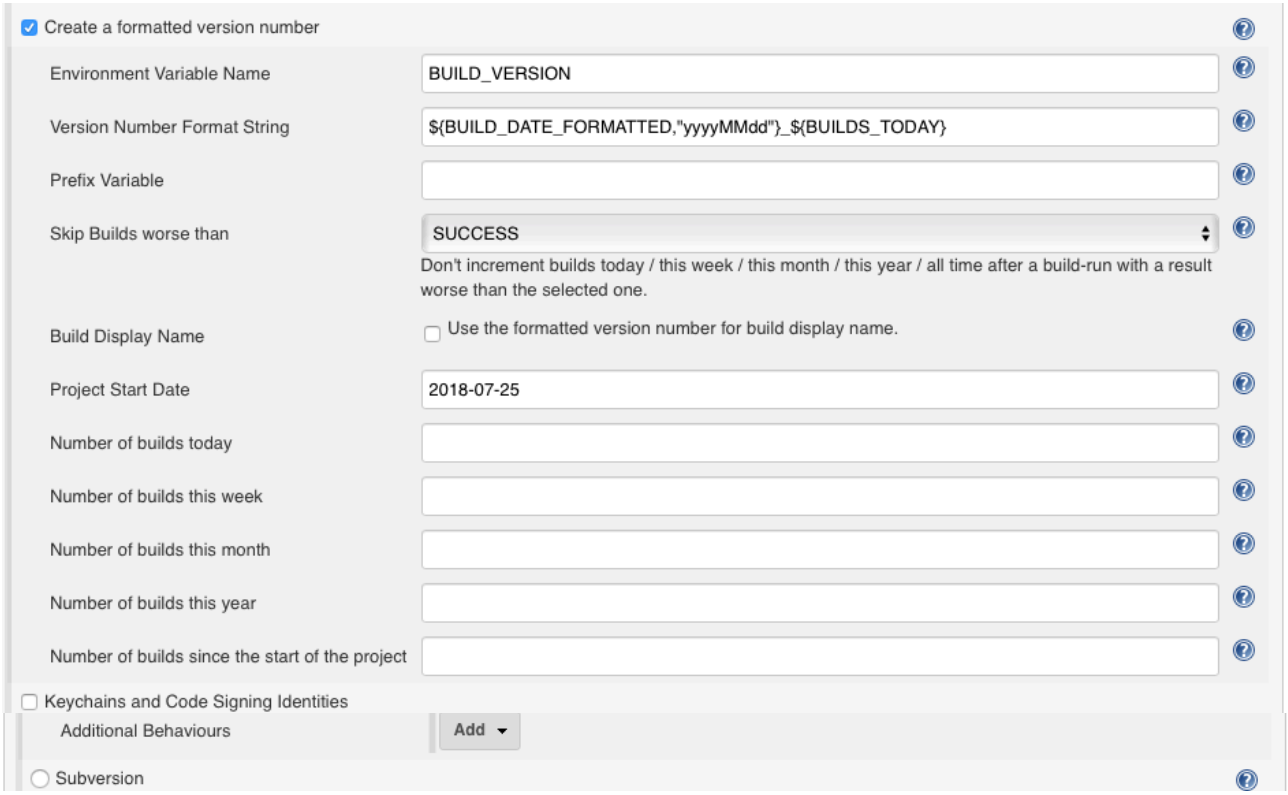
#######

整合 jenkins 举例一个环境

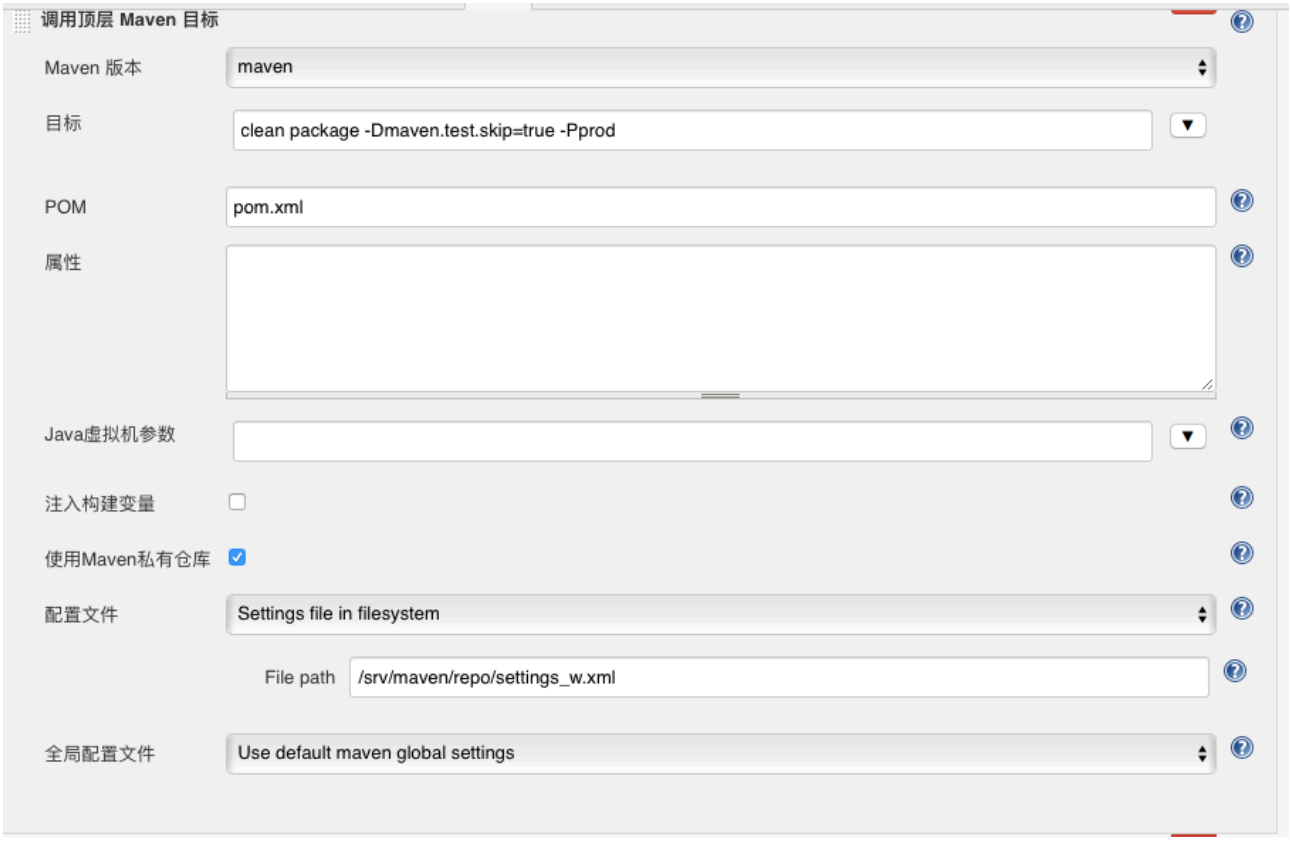


第一步：参数化构建

第二步：源码仓库

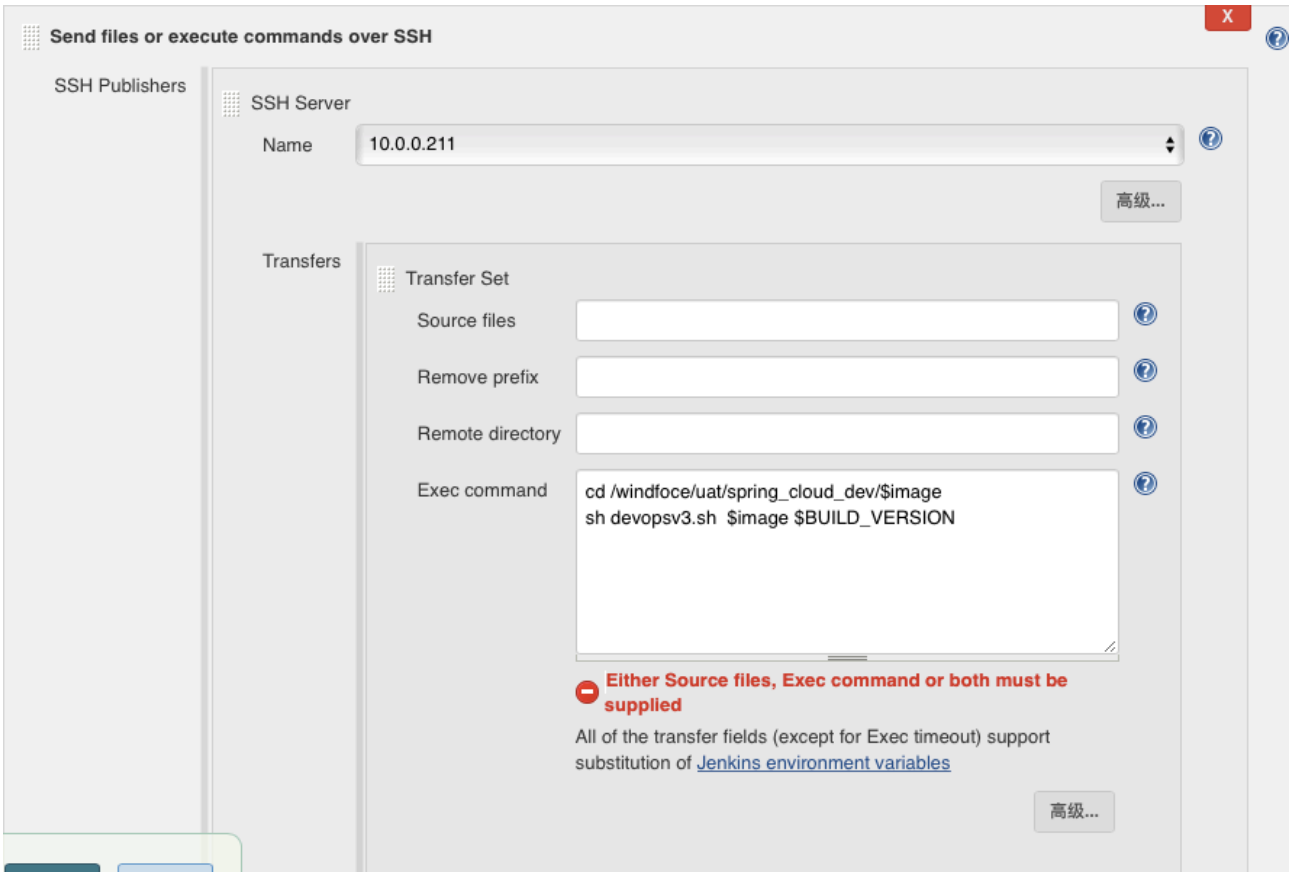


第三步：版本控制
第四步：构建



第五步：shell 脚本 作用：生产 dockerfile

第六步：生产 docker 容器


Send files or execute commands over SSH
X
SSH Publishers
SSH Server
Name
10.0.0.211
高级...
Transfers
Transfer Set
Source files
Remove prefix
Remote directory
Exec command
cd /windfoce/uat/spring_cloud_dev/$image
sh devopsv3.sh $image $BUILD_VERSION
Either Source files, Exec command or both must be supplied
All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables
高级...