Для выполнения задания минимум понадобится 3 машины:

1 – локальный gitlab server – ubuntu2 | 2 cpu 8gb
1 – kuber master – kmaster2 | 2 cpu 4gb
1 – kuber woker – kworker2 |2 cpu 4gb

На всех машинах установлена System Ubuntu 20.04

Увеличение места на виртуалке

В процессе установки выяснилось, что нужно увеличить место

```
Need to get 124 MB of archives.
After this operation, 454 MB of additional disk space will be used.
E: You don't have enough free space in /var/cache/apt/archives/.
```

На хосте:

```
virsh edit ubuntu2 (найти строку с диском - в данном случае он называется vm-hosting-clone-1.qcow2 и вставить в слуедующую команду)
qemu-img resize /var/lib/libvirt/images/vm-hosting-clone-1.qcow2 +10G
```

Тут в начале образ монтируется в /dev/ndb0 и партед запускается с параметром /dev/ndb0
parted надо на хосте запускать. и нажать F когда предложит Fix. тогда можно resizepart адекватно сделать

На виртуальной машине выполнить:

```
pvresize /dev/vda3
lvextend -l +100%FREE /dev/ubuntu-vg/ubuntu-lv
resize2fs /dev/mapper/ubuntu--vg-ubuntu--lv
```

Клонирование машин из заготовленного образа

```
virt-clone --original= ubuntu--name=kmaster --auto-clone
virt-clone --original= ubuntu --name=kworker --auto-clone
virsh list --al
```

После клонирование надо поменять имя хоста и machine-id на клонированных машинах

```
vi /etc/hostname
dbus-uuidgen
0e0a05341e8aa7285c48b6e963c511b9
echo 0e0a05341e8aa7285c48b6e963c511b9>/var/lib/dbus/machine-id
cat /var/lib/dbus/machine-id
echo 0e0a05341e8aa7285c48b6e963c511b9>/etc/machine-id
cat /etc/machine-id
reboot
```

Теперь у гостевых машин будет разный ip к ним можно обращаться по ssh
Однако ip пока у машин динамический и надо будет его связать с мак адресом

Узнать MAC

```
virsh  dumpxml  $VM_NAME | grep 'mac address'
```

Провреить сеть и отредактировать её. В данном случае сеть называется default

```
virsh  net-list
virsh  net-edit  default
```

```
<network>
  <name>default</name>
  <uuid>87ac664f-08b7-4c95-a0a1-6b829c7b523e</uuid>
  <forward mode='nat'/>
  <bridge name='virbr0' stp='on' delay='0'/>
  <mac address='52:54:00:90:87:f5'/>
  <ip address='192.168.122.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.122.2' end='192.168.122.254'/>
      <host mac='52:54:00:56:a2:95' name='ubuntu2' ip='192.168.122.105'/>
      <host mac='52:54:00:28:94:2c' name='kmaster' ip='192.168.122.49'/>
      <host mac='52:54:00:57:f3:f3' name='kworker' ip='192.168.122.254'/>
      <host mac='52:54:00:e9:ee:68' name='layout' ip='192.168.122.104'/>
      <host mac='52:54:00:c2:3e:c1' name='kmaster2' ip='192.168.122.123'/>
      <host mac='52:54:00:db:b9:ea' name='test' ip='192.168.122.227'/>
      <host mac='52:54:00:24:dd:37' name='ubuntu' ip='192.168.122.107'/>
      <host mac='52:54:00:6a:13:a9' name='bare_ubuntu' ip='192.168.122.108'/>
      <host mac='52:54:00:08:6d:c6' name='gitlab' ip='192.168.122.109'/>
      <host mac='52:54:00:74:31:25' name='kworker2' ip='192.168.122.230'/>
      <host mac='52:54:00:99:b3:79' name='kubertemplate' ip='192.168.122.133'/>
    </dhcp>
  </ip>
</network>
```

Чтобы изменения вступили в силу

```
virsh net-destroy network_name
virsh net-start network_name
systemctl restart libvirtd.service
```

Для удобства подключения отредактировать /etc/hosts/

```
127.0.0.1       localhost
127.0.1.1       CITADELDEVELOP

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

```
192.168.122.105 ubuntu2
192.168.122.123 kmaster2
192.168.122.227 test
192.168.122.107 ubuntu
192.168.122.108 bare_ubuntu
192.168.122.109 gitlab
192.168.122.230 kworker2
192.168.122.133 kubertemplate
```

установка гитлаба

Полезная ссылка https://about.gitlab.com/install/#ubuntu
В качестве gitlab server выбрана машина ubuntu2

```
sudo apt-get update
sudo apt-get install -y curl openssh-server ca-certificates tzdata perl

curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ee/script.deb.sh | sudo
bash
```

Проблема в блокировке гитлаба РФ

```
root@test:/home/user/empty# curl
https://packages.gitlab.com/install/repositories/gitlab/gitlab-ee/script.deb.sh |
sudo bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  6865  100  6865    0     0  14754      0 --:--:-- --:--:-- --:--:-- 14763
Detected operating system as Ubuntu/jammy.
Checking for curl...
Detected curl...
Checking for gpg...
Detected gpg...
Running apt-get update... done.
Installing apt-transport-https... done.
Installing /etc/apt/sources.list.d/gitlab_gitlab-ee.list...done.
Importing packagecloud gpg key... curl: (22) The requested URL returned error: 403
gpg: no valid OpenPGP data found.
done.
Running apt-get update... done.

The repository is setup! You can now install packages.

****
```

Нужно установить vpn https://freevpn.me/accounts/ - скорость хорошая или любой другой.
Качается zip, распаковывается, закидывается по scp на ноду
конфиг лежит в виртуалке в /home/user/Server1-TCP80.ovpn
Затем продолжается установка и проверяется статус.

```
sudo EXTERNAL_URL="https://gitlab.example.com" apt-get install gitlab-ee
systemctl status gitlab-runsvdir.service
```

В инструкции указано сохранить пароль

Unless you provided a custom password during installation, a password will be randomly generated and stored for 24 hours in /etc/gitlab/initial_root_password. Use this password with username root to login.

gitlab 443 слушает

```
root@test:~# curl https://127.0.0.1:443
curl: (60) SSL certificate problem: self-signed certificate
```

Проброс порта на хосте в kvm виртуалке
Файл с хуком должен находиться в директории /etc/libvirt/hooks.
Создаём енеобходимое

```
# mkdir /etc/libvirt/hooks
# chmod 700 /etc/libvirt/hooks
```

Создадим файл /etc/libvirt/hooks/qemu, в котором укажем, что входящие на `'40443' '40450' '45000' '1818' '40022'` порты должны быть перенаправлены на `'443' '450' '5000' '1818' '22'` порты гостевой `ubuntu2`. В моем примере `10.8.0.1`- внешний ip-адрес хоста KVM, а `192.168.122.105`- ip-адрес виртуальной машины `ubuntu2` на этом хосте.

```
#!/bin/bash

# https://bozza.ru/art-268.html
# used some from advanced script to have multiple ports: use an equal number of guest and
host ports

# Update the following variables to fit your setup
Guest_name=ubuntu2
Guest_ipaddr=192.168.122.105
Host_ipaddr=10.8.0.1
Host_port=(  '40443' '40450' '45000' '1818' '40022')
Guest_port=( '443' '450' '5000' '1818' '22')

length=$(( ${#Host_port[@]} - 1 ))
if [ "${1}" = "${Guest_name}" ]; then
   if [ "${2}" = "stopped" ] || [ "${2}" = "reconnect" ]; then
     for i in `seq 0 $length`; do
        iptables -t nat -D PREROUTING -d ${Host_ipaddr} -p tcp --dport ${Host_port[$i]} -j
DNAT --to ${Guest_ipaddr}:${Guest_port[$i]}
        iptables -D FORWARD -d ${Guest_ipaddr}/32 -p tcp -m state --state NEW -m tcp --dport
${Guest_port[$i]} -j ACCEPT
     done
   fi
   if [ "${2}" = "start" ] || [ "${2}" = "reconnect" ]; then
     for i in `seq 0 $length`; do
```

```
        iptables -t nat -A PREROUTING -d ${Host_ipaddr} -p tcp --dport ${Host_port[$i]} -j
DNAT --to ${Guest_ipaddr}:${Guest_port[$i]}
        iptables -I FORWARD -d ${Guest_ipaddr}/32 -p tcp -m state --state NEW -m tcp --dport
${Guest_port[$i]} -j ACCEPT
     done
   fi
fi
```

Файл будет исполняемым

```
chmod 700 /etc/libvirt/hooks/qemu
```

Машина и сеть после рестарта сами не поднимаются - после ребута надо сделать

```
virsh net-start default && virsh start ubuntu2
```

Установка kibernetes на master

sudo apt-get update

curl -fsSL https://packages.cloud.google.com/apt/doc/apt-key.gpg|sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/k8s.gpg

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt update

sudo apt install wget curl vim git kubelet kubeadm kubectl -y

sudo apt-mark hold kubelet kubeadm kubectl

kubectl version --client && kubeadm version

cat /etc/fstab

free -h

vim /etc/fstab

sudo vim /etc/fstab

sudo swapoff -a

free -h

sudo modprobe overlay

sudo modprobe br_netfilter

sudo tee /etc/sysctl.d/kubernetes.conf<<EOF

net.bridge.bridge-nf-call-ip6tables = 1

```
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF
cat /etc/sysctl.d/kubernetes.conf
sudo sysctl --system
sudo apt update
sudo apt install -y curl gnupg2 software-properties-common apt-transport-https ca-certificates
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt update
sudo apt install -y containerd.io docker-ce docker-ce-cli
sudo mkdir -p /etc/systemd/system/docker.service.d
sudo tee /etc/docker/daemon.json <<EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
   "max-size": "100m"
  },
    "storage-driver": "overlay2"
  }
  EOF
  sudo systemctl daemon-reload
  sudo systemctl restart docker
  sudo systemctl enable docker
  sudo systemctl status docker
  sudo tee /etc/modules-load.d/k8s.conf <<EOF
  overlay
  br_netfilter
  EOF
```

```
sudo modprobe overlay

sudo modprobe br_netfilter

cat /etc/sysctl.d/kubernetes.conf

sudo apt install git wget curl

VER=$(curl -s https://api.github.com/repos/Mirantis/cri-dockerd/releases/latest|grep tag_name | cut -d '"' -f 4|sed 's/v//g')

echo $VER

wget https://github.com/Mirantis/cri-dockerd/releases/download/v${VER}/cri-dockerd-${VER}.amd64.tgz

tar xvf cri-dockerd-${VER}.amd64.tgz

sudo mv cri-dockerd/cri-dockerd /usr/local/bin/

cri-dockerd --version

sudo cri-dockerd --version

cri-dockerd --version

ls /usr/local/bin/ -hal | grep cri-do

chmod +x /usr/local/bin/cri-dockerd

chown root:root /usr/local/bin/cri-dockerd

sudo chown root:root /usr/local/bin/cri-dockerd

cri-dockerd --version

wget https://raw.githubusercontent.com/Mirantis/cri-dockerd/master/packaging/systemd/cri-docker.service

wget https://raw.githubusercontent.com/Mirantis/cri-dockerd/master/packaging/systemd/cri-docker.socket

sudo mv cri-docker.socket cri-docker.service /etc/systemd/system/

sudo sed -i -e 's,/usr/bin/cri-dockerd,/usr/local/bin/cri-dockerd,' /etc/systemd/system/cri-docker.service

ls -al /etc/systemd/system | grep cri-do

systemctl status cri-docker.socket

sudo systemctl daemon-reload

sudo systemctl enable cri-docker.service

sudo systemctl enable --now cri-docker.socket

systemctl status cri-docker.socket

systemctl status docker

lsmod | grep br_netfilter
```

```
sudo systemctl enable kubelet
sudo kubeadm config images pull --cri-socket /run/cri-dockerd.sock
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --cri-socket /run/cri-dockerd.sock
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
kubectl cluster-info
kubectl get nodes
wget https://raw.githubusercontent.com/flannel-io/flannel/master/Documentation/kube-flannel.yml
kubectl apply -f kube-flannel.yml
```

Установка kibernetes на worker

```
sudo apt update
sudo apt -y full-upgrade
sudo reboot
sudo su -
sudo apt install curl apt-transport-https -y
curl -fsSL  https://packages.cloud.google.com/apt/doc/apt-key.gpg|sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/k8s.gpg
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
sudo apt update
sudo apt install wget curl vim git kubelet kubeadm kubectl -y
sudo apt-mark hold kubelet kubeadm kubectl
kubectl version --client && kubeadm version
swapoff -a
sudo swapoff -a
free -h
sudo vim /etc/fstab
sudo mount -a
```

```
  mount
  free -h
  sudo modprobe overlay
  sudo modprobe br_netfilter
  sudo tee /etc/sysctl.d/kubernetes.conf<<EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF

  sudo sysctl --system
  sudo apt install -y curl gnupg2 software-properties-common apt-transport-https ca-certificates
  curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
  sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
  sudo apt update
  sudo apt install -y containerd.io docker-ce docker-ce-cli
  sudo mkdir -p /etc/systemd/system/docker.service.d
  sudo tee /etc/docker/daemon.json <<EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF

  sudo systemctl daemon-reload
  sudo systemctl restart docker
```

```
  sudo systemctl enable docker
  sudo tee /etc/modules-load.d/k8s.conf <<EOF
overlay
br_netfilter
EOF

  sudo modprobe overlay
  sudo modprobe br_netfilter
  sudo tee /etc/sysctl.d/kubernetes.conf<<EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF

  systemctl status docker
  sudo apt install git wget curl
  VER=$(curl -s https://api.github.com/repos/Mirantis/cri-dockerd/releases/latest|grep tag_name | cut -d '"' -f 4|sed 's/v//g')
  echo $VER
  wget https://github.com/Mirantis/cri-dockerd/releases/download/v${VER}/cri-dockerd-${VER}.amd64.tgz
  tar xvf cri-dockerd-${VER}.amd64.tgz
  sudo mv cri-dockerd/cri-dockerd /usr/local/bin/
  cri-dockerd --version
  chmod +x /usr/local/bin/cri-dockerd
  sudo chown root:root /usr/local/bin/cri-dockerd
  cri-dockerd --version
  wget https://raw.githubusercontent.com/Mirantis/cri-dockerd/master/packaging/systemd/cri-docker.service
  wget https://raw.githubusercontent.com/Mirantis/cri-dockerd/master/packaging/systemd/cri-docker.socket
  sudo mv cri-docker.socket cri-docker.service /etc/systemd/system/
  sudo sed -i -e 's,/usr/bin/cri-dockerd,/usr/local/bin/cri-dockerd,' /etc/systemd/system/cri-docker.service
```

sudo systemctl daemon-reload

sudo systemctl enable cri-docker.service

sudo systemctl enable --now cri-docker.socket

systemctl status cri-docker.socket

sudo kubeadm config images pull --cri-socket /run/cri-dockerd.sock

sudo kubeadm join 192.168.122.123:6443 --token cvvwfq.5byl356blbqwn4t7 --discovery-token-ca-cert-hash sha256:a8b44e8ed955d51aV2PLNRrdGGH9b1Gv9m1h49MJEGt9h8ef8622e00c41c0d4f --cri-socket /run/cri-dockerd.sock

Установка runner

Раннер по гитлабовской ссылке, по официальной документации

Раннер не получается зарегать сразу, из-за того, что нет сертификата

x509: certificate relies on legacy Common Name field, use SANs instead

Тут генерирует сертификат с 'legacy common name field', а ранеру надо чтобы некий 'sans' был
Решение:

в etc/hosts добавил 127.0.0.1 gitlabubuntu2.com
в /etc/gitlab/gitlab.rb сменил хостнейм на gitlabubuntu2.com

выполнить

```
gitlab-ctl restart
```

потом выполнил всё отсюда https://gitlab.com/gitlab-org/gitlab-runner/-/issues/28841#note_1004765355
заменив example.com на gitlabubuntu2.com

```
openssl genrsa -out ca.key 2048
openssl req -new -x509 -days 365 -key ca.key -subj "/C=CN/ST=GD/L=SZ/O=Acme, Inc./CN=Acme Root CA" -out ca.crt
openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/C=CN/ST=GD/L=SZ/O=Acme, Inc./CN=*.gitlabubuntu2.com" -out server.csr
openssl x509 -req -extfile <(printf "subjectAltName=DNS:example.com,DNS:gitlabubuntu2.com") -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt
```

В конце в ранер скормить сертификат

```
sudo gitlab-runner register --tls-ca-file=/etc/gitlab/ssl/ca.crt
```

В /etc/hosts это имя добавлено только на ubuntu2

https://10.8.0.1:40443/admin/runners

гитлаб его видит

https://gitlab.com/gitlab-org/gitlab-runner/-/issues/28841

Добавлении кластера кубера в гитлаб

Это вот сам гитлаб выдал при добавлении кластера кубера в него, и эти команды вводятся на кубер мастере

Это то что ставится внутрь кубера через хельм, для связи с гитлабом

```
helm repo add gitlab https://charts.gitlab.io
helm repo update
helm upgrade --install agents-one gitlab/gitlab-agent \
    --namespace gitlab-agent-agents-one \
    --create-namespace \
    --set image.tag=v15.7.0 \
    --set config.token=-VFD7L8goQFXYu22uW-hC-Gem6rAYadmmzqQ8j3XKxxrezPczw \
    --set config.kasAddress=wss://gitlabubuntu2.com/-/kubernetes-agent/
```

kmaster2 тоже нужно поставить впн - ещё что-то залочено gitlab agent

На kworker2 и на kmaster2 добавил в /etc/hosts 192.168.122.105 gitlabubuntu2.com
В противном случае агент не может достучаться

Ещё на сертификат жалуется
он достучаться до gitlabubuntu2.com не может

Ему нужно скормить hostAliases, теперь проблема с сертификатом. но тут можно его указать

```
user@kmaster2:~$ cat values.yaml
image:
  tag: "v15.7.0"

config:
  token: "-VFD7L8goQFXYu22uW-hC-Gem6rAYadmmzqQ8j3XKxxrezPczw"
  kasAddress: "wss://gitlabubuntu2.com/-/kubernetes-agent/"

hostAliases:
  - ip: "192.168.122.105"
    hostnames:
      - "gitlabubuntu2.com"

user@kmaster2:~$ helm upgrade --install agents-one gitlab/gitlab-agent   --namespace gitlab-
agent-agents-one     -f values.yaml
```

https://gitlab.com/gitlab-org/charts/gitlab-agent/-/blob/main/values.yaml
тут можно сертификат скормить

```
# caCert: "PEM certificate file to use to verify config.kasAddress. Useful if
config.kasAddress is self-signed."
```

ща перенесу его с машины с гитлабом сюда и переустановлю агент

```
helm uninstall agents-one  --namespace gitlab-agent-agents-one
```

это удаление агента

это просто сам создаешь с любым названием чтобы при helm install gitlab-agent его скормить

```
helm upgrade --install agents-one gitlab/gitlab-agent   --namespace gitlab-agent-agents-one
-f values.yaml --set config.caCert="$(cat ca_cert_gitlab.crt)"
```

и 2 файла - values.yaml и ca_cert_gitlab.crt с машиты с гитлабом из /etc/gitlab/ssl/ca.crt

ошибок в логе нет, но гитлаб почему-то не видит чтобы агент ожил

```
user@kmaster2:~$ kubectl logs agents-one-gitlab-agent-7f4f466c8d-85sn4 --namespace gitlab-
agent-agents-one -f
{"level":"info","time":"2023-01-18T07:19:55.659Z","msg":"Observability endpoint is
up","mod_name":"observability","net_network":"tcp","net_address":"[::]:8080"}
```

открыл 8080

гитлаб пока без изменений

https://10.8.0.1:40443/gitlab-instance-32af17e0/test-py/-/jobs/4

```
fatal: unable to access 'https://gitlabubuntu2.com/gitlab-instance-32af17e0/test-py.git/':
Could not resolve host: gitlabubuntu2.com
```

в .gitlab-ci.yml добавил extra_hosts = ["gitlabubuntu2.com:192.168.122.105"] в секцию runners.docker в "/etc/gitlab-runner/config.toml"

Ранер не подхватывает доменное имя

добавил extra_hosts = ["gitlabubuntu2.com:192.168.122.105"] в секцию runners.docker в "/etc/gitlab-runner/config.toml"

pipeline заработал

https://10.8.0.1:40443/gitlab-instance-32af17e0/test-py/-/jobs/6

теперь другая ошибка

```
$ kubectl config use-context path/to/agent/repository:agent-name
error: no context exists with the name: "path/to/agent/repository:agent-name"
```

это, написано path to agent repository, но контекст в кубере это несколько другое

error: no context exists with the name: "gitlab-instance-32af17e0/test-py:agent-one"

сохранить пространство имен для всех последующих команд kubectl в этом контексте.

```
$ kubectl config get-contexts
CURRENT   NAME                                          CLUSTER   AUTHINFO   NAMESPACE
          gitlab-instance-32af17e0/test-py:agents-one   gitlab    agent:1
$ kubectl config use-context gitlab-instance-32af17e0/test-py:agent-one
```

смотрим
опять сертификат

Unable to connect to the server: x509: certificate signed by unknown authority
Cleaning up project directory and file based variables
00:04
ERROR: Job failed: exit code 1

kubectl get pods в раннере идёт сюда https://gitlabubuntu2.com/-/kubernetes-agent/k8s-proxy
т.е. гитлаб типа проксирует запросы в кластер

серт сюда надо добавить
extra_hosts = ["gitlabubuntu2.com:192.168.122.105"] в секцию runners.docker в "/etc/gitlab-runner/config.toml"

агент не может связаться с гитлабом

https://10.8.0.1:40443/gitlab-instance-32af17e0/test-py/-/cluster_agents/agents-one?tab=tokens

посколько здесь last contact never
вот видно что агент ходит в гитлаб

```
root@ubuntu2:~# cat /var/log/gitlab/nginx/gitlab_access.log | grep kubernetes-agent
192.168.122.230 - - [18/Jan/2023:10:00:03 +0000] "GET /-/kubernetes-agent/ HTTP/1.1" 101 833
"" "gitlab-agent/v15.7.0/77d9ff24" -
```

ещё ошибка с сертификатом

```
root@ubuntu2:~# tail /var/log/gitlab/gitlab-kas/current -f
2023-01-18_10:00:17.85950 {"level":"error","time":"2023-01-
18T10:00:17.859Z","msg":"AgentInfo()","grpc_service":"gitlab.agent.reverse_tunnel.rpc.Revers
eTunnel","grpc_method":"Connect","error":"Get
\"https://gitlabubuntu2.com/api/v4/internal/kubernetes/agent_info\": x509: certificate
signed by unknown authority"}
```

Для решения в основной конфиг гитлаба /etc/gitlab/gitlab.rb вставляется это:

```
gitlab_kas['env'] = {
    'SSL_CERT_DIR' => '/etc/gitlab/ssl'
}
```

https://10.8.0.1:40443/gitlab-instance-32af17e0/test-py/-/cluster_agents/agents-one?tab=tokens

логи kubernetes agent service встроенного в гитлаб,

```
tail /var/log/gitlab/gitlab-kas/current -f
```

логи самого агента в кластере на мастере

```
kubectl logs -f -l=app=gitlab-agent -n gitlab-agent-agents-one
```

Включение регистри

Добавить в /etc/gitlab/gitlab.rb

```
registry_external_url 'https://gitlabubuntu2.com:5050'
```

Рестартовать гитлаб

Создать personal access token здесь https://10.8.0.1:40443/-/profile/personal_access_tokens
с токеном заходит так:

```
docker login gitlabubuntu2.com:5050 -u root -p glpat-_za8T45arqT8wWKk5aMz
```

Открыть порт 5050

registry работает