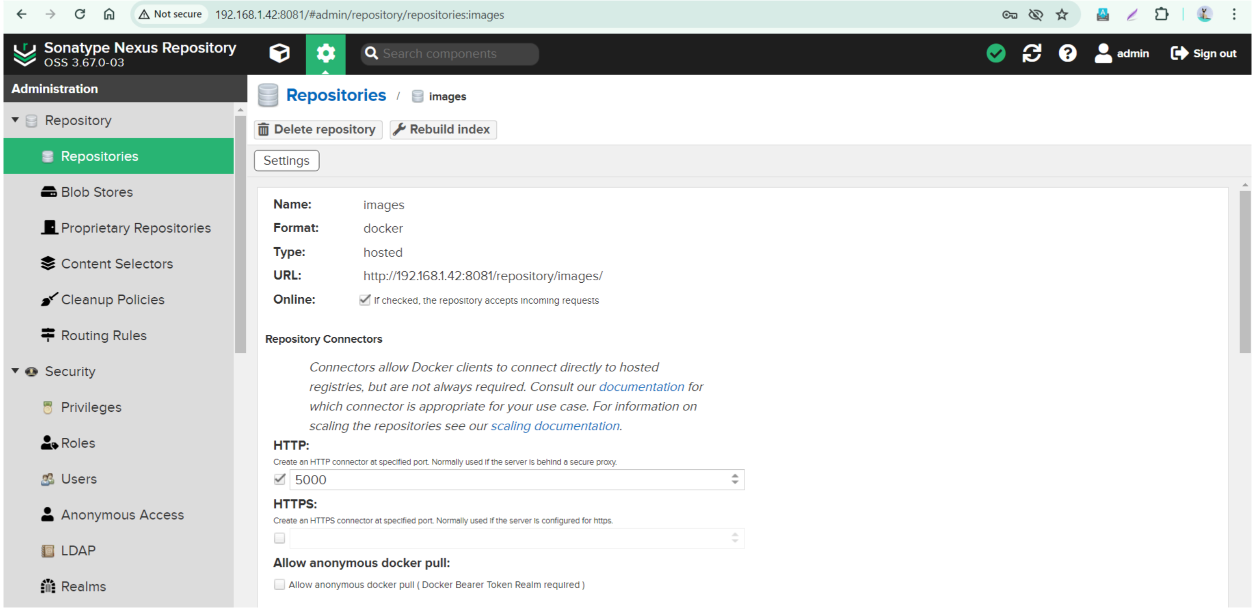
Nexus with Nginx

```
# podman run -itd --name nexus-new -p 8443:8443 -p 8081:8081 -p 5000:5000 -v
/root/nexus/nexus-data:/nexus-data sonatype/nexus3
```



Name: images

Format: docker

Type: hosted

URL: http://192.168.1.42:8081/repository/images/

Online: ☑ If checked, the repository accepts incoming requests

Repository Connectors

Connectors allow Docker clients to connect directly to hosted registries, but are not always required. Consult our documentation for which connector is appropriate for your use case. For information on scaling the repositories see our scaling documentation.

HTTP:

Create an HTTP connector at specified port. Normally used if the server is behind a secure proxy.

☑ 5000

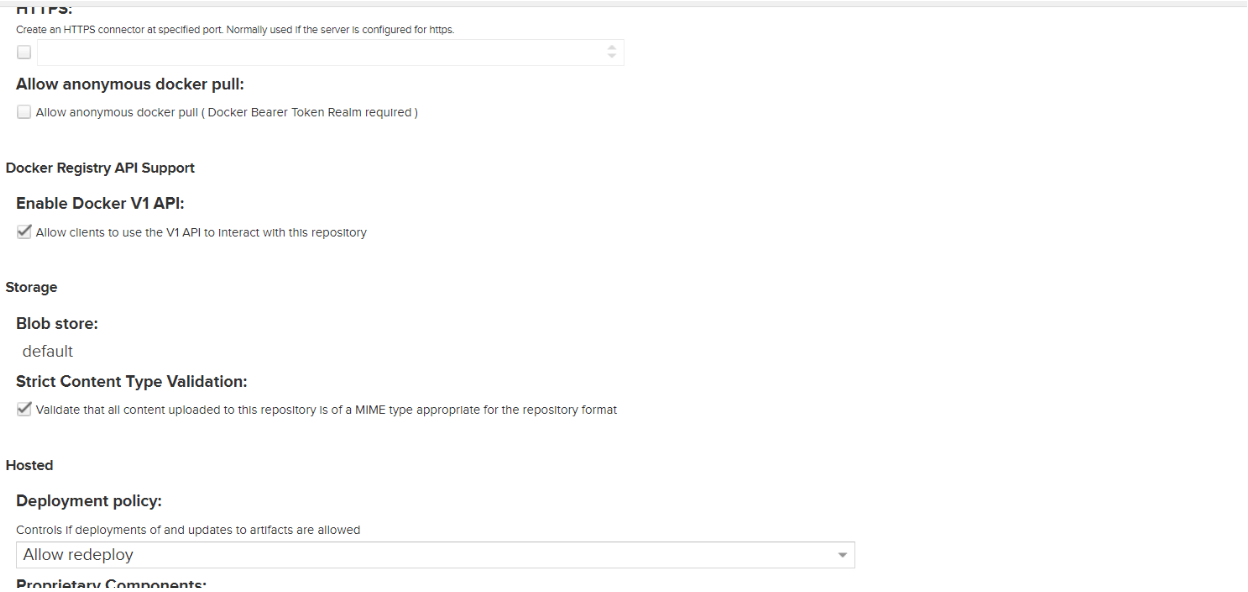
HTTPS:

Create an HTTPS connector at specified port. Normally used if the server is configured for https.

☐

Allow anonymous docker pull:

☐ Allow anonymous docker pull (Docker Bearer Token Realm required)



HTTPS:

Create an HTTPS connector at specified port. Normally used if the server is configured for https.

☐

Allow anonymous docker pull:

☐ Allow anonymous docker pull (Docker Bearer Token Realm required)

Docker Registry API Support

Enable Docker V1 API:

☑ Allow clients to use the V1 API to interact with this repository

Storage

Blob store:

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:

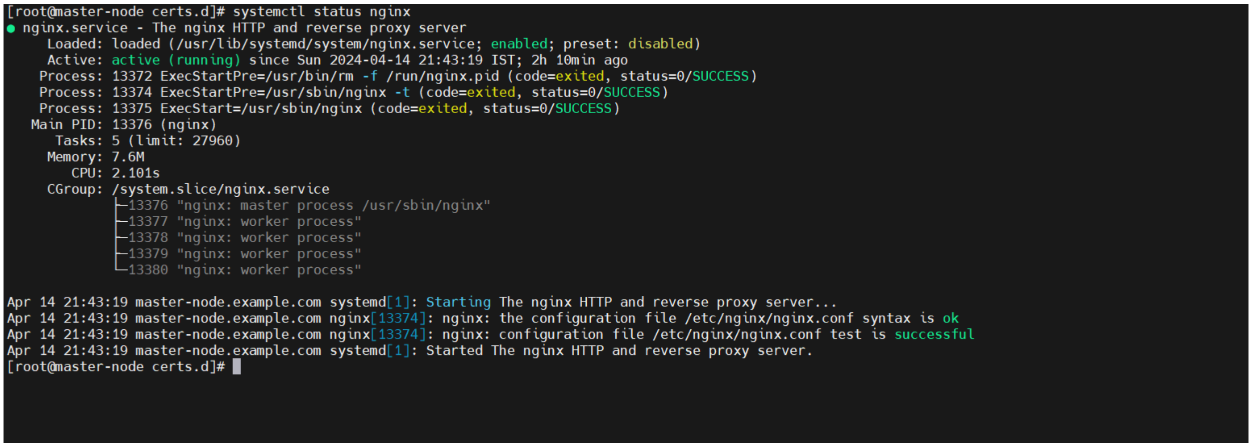
Controls if deployments of and updates to artifacts are allowed

Allow redeploy

Proprietary Components:

```
# yum install nginx -y
```

# systemctl start nginx
# systemctl enable nginx

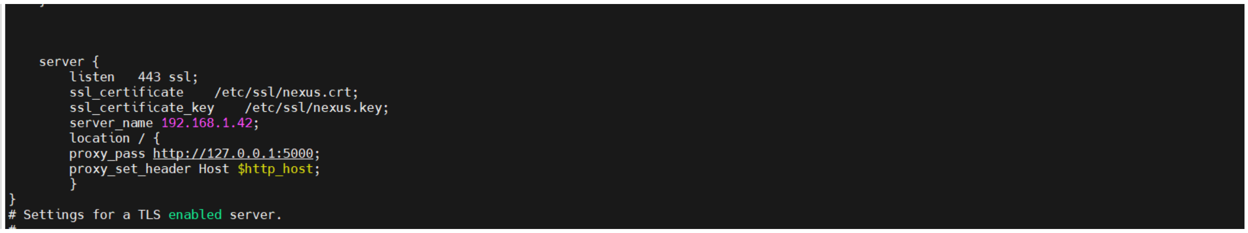


```
[root@master-node certs.d]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
     Active: active (running) since Sun 2024-04-14 21:43:19 IST; 2h 10min ago
    Process: 13372 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 13374 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 13375 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 13376 (nginx)
      Tasks: 5 (limit: 27960)
     Memory: 7.6M
        CPU: 2.101s
     CGroup: /system.slice/nginx.service
             ├─13376 "nginx: master process /usr/sbin/nginx"
             ├─13377 "nginx: worker process"
             ├─13378 "nginx: worker process"
             ├─13379 "nginx: worker process"
             └─13380 "nginx: worker process"

Apr 14 21:43:19 master-node.example.com systemd[1]: Starting The nginx HTTP and reverse proxy server...
Apr 14 21:43:19 master-node.example.com nginx[13374]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Apr 14 21:43:19 master-node.example.com nginx[13374]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Apr 14 21:43:19 master-node.example.com systemd[1]: Started The nginx HTTP and reverse proxy server.
[root@master-node certs.d]# █
```

[root@master-node certs.d]# cat /etc/nginx/nginx.conf

Add below lines.

```
    server {
        listen    443 ssl;
        ssl_certificate     /etc/ssl/nexus.crt;
        ssl_certificate_key     /etc/ssl/nexus.key;
        server_name 192.168.1.42;
        location / {
        proxy_pass http://127.0.0.1:5000;
        proxy_set_header Host $http_host;
        }
```



```
    server {
        listen   443 ssl;
        ssl_certificate    /etc/ssl/nexus.crt;
        ssl_certificate_key    /etc/ssl/nexus.key;
        server_name 192.168.1.42;
        location / {
        proxy_pass http://127.0.0.1:5000;
        proxy_set_header Host $http_host;
        }
}
# Settings for a TLS enabled server.
```

SSL Certficate

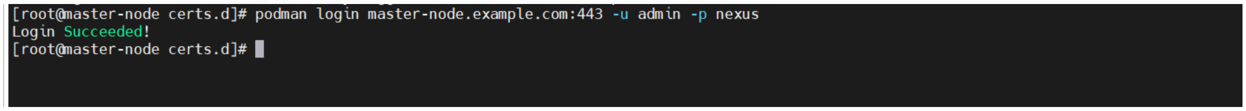
[root@master-node ssl]# openssl genpkey -algorithm RSA:2048 -out nexus.key

[root@master-node ssl]# cat ca.cnf

```
[req]
default_bits = 2048
prompt = no
default_md = sha256
distinguished_name = dn
req_extensions = SAN   # Specify the SAN extension here

[dn]
C = in
ST = maharastra
L = navimumbai
O = kr
OU = it
emailAddress = admin@example.com
CN = example.com

[SAN]
subjectAltName = @alt_names

[alt_names]
DNS.1 = master-node
DNS.2 = master-node.example.com
```

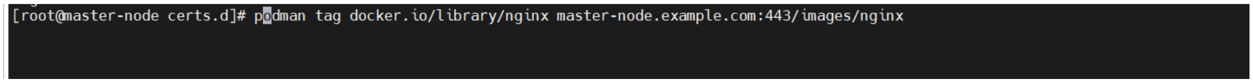
```
[root@master-node ssl]# openssl req -new -key nexus.key -out nexus.csr -config ca.cnf

[root@master-node ssl]# openssl x509 -req -in nexus.csr -signkey nexus.key -out nexus.crt
-days 365 -extensions SAN -extfile ca.cnf

####################### copy nexus.crt and nexux.key to /etc/ssl/
###################################
[root@master-node ssl]# cp nexus.crt /etc/ssl/
[root@master-node ssl]# cp nexus.key /etc/ssl/

####################### copy nexus.crt to
/etc/pki/ca-trust/source/anchors/#######################################
[root@master-node ssl]# cp nexus.crt /etc/pki/ca-trust/source/anchors/

######################## copy nexus.crt to
/etc/containers/certs.d/master-node.example.com\:443/ ######################
[root@master-node ssl]# mkdir /etc/containers/certs.d/master-node.example.com:443   # the file
will have to create named <hostname>:443
[root@master-node ssl]# cp nexus.crt /etc/containers/certs.d/master-node.example.com\:443/
```

Note: the host entry will add in master as well as all workers and nexus.crt need to put in /etc/containers/certs.d/master-node.example.com:443 on all workers



```
[root@master-node certs.d]# podman login master-node.example.com:443 -u admin -p nexus
Login Succeeded!
[root@master-node certs.d]#
```

Tag image
# podman tag docker.io/library/nginx master-node.example.com:443/images/nginx

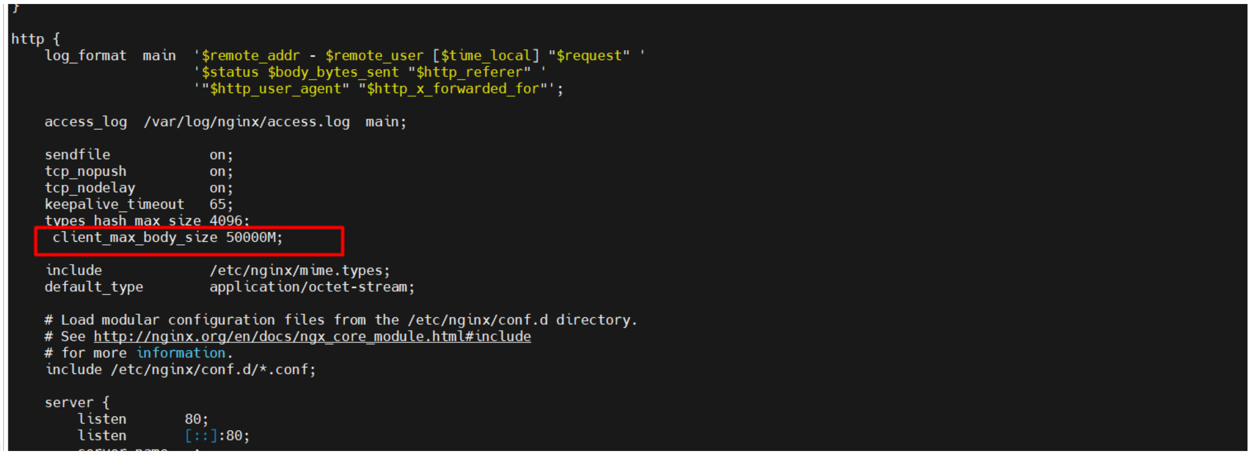


```
[root@master-node certs.d]# podman tag docker.io/library/nginx master-node.example.com:443/images/nginx
```

Push image
# podman push master-node.example.com:443/images/nginx

Note: If you face error to push image due to size of image”too large” , will be added in nginx.conf

client_max_body_size 50000M;



```
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 4096;
     client_max_body_size 50000M;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
    # for more information.
    include /etc/nginx/conf.d/*.conf;

    server {
        listen       80;
        listen       [::]:80;
```

Pull image

# podman pull master-node.example.com:443/images/nginx