

# 3. Install VNCproxy and noVNC

## ProxmoxKVM module **WHMCS**

[Order now](#) | [Download](#) | [FAQ](#)

### Preface

The module supports the ability to connect and use the console to manage a specific KVM virtual machine.

To connect to the virtual machine console, we will use third-party software.

**noVNC - the open source VNC client** - noVNC is both a VNC client JavaScript library as well as an application built on top of that library. noVNC runs well in any modern browser including mobile browsers (iOS and Android).

- Project site: <https://novnc.com>
- Project github: <https://github.com/novnc/noVNC>

As we only use external project we dont take any responsibility for data leak, hacks etc.

### We use golang to build our products.

We have used the following libraries

- <https://github.com/evangwt/go-vncproxy> (MIT License)
- <https://github.com/gin-gonic/gin> (MIT License)
- <https://pkg.go.dev/golang.org/x/net/websocket> (BSD License)

If you have any difficulties, you can use our public vncproxy server. *We strongly recommend setting up and using your own vncproxy server*

. You will retain control over server performance and security

noVNC WEB proxy server: **vncproxy.puqcloud.com**

noVNC WEB proxy key: **puqcloud**

WEB ports: **80/443**

VNC ports: **5900-5999**

With vncproxy you make a proxy between the client and your **PROXMOX** server.

vncproxy must have an unequal stable network with the proxmox server, **ports 5900-5999** are enough

Also, if you use a domain name in connecting the **PROXMOX** server to the **WHMCS** system, this domain name must be correctly resolved from the vncproxy server

Let's start with installation.

# Installation process

## Domain definition

First, define a domain name for the vncproxy server, in our case it will be **vncproxy.puqcloud.com**

Further in the example, we will use the domain name **vncproxy.puqcloud.com**, but in all your configurations you must use your own domain name.

## NGINX installation and configuration

Secondly, you need to install a server with your favorite operating system. In our case, this is the **Debian 11** operating system. You also need to set up a DNS entry on your domain so that it returns the IP address of the server.

At first, if you haven't updated the package database recently, update it:

```
sudo apt update
```

Install nginx WEB server and Certbot

```
sudo apt install certbot nginx python3-certbot-nginx zip -y
```

## Download noVNC client

```
cd /root/  
wget https://github.com/novnc/noVNC/archive/refs/tags/v1.3.0.zip  
unzip v1.3.0.zip  
cp -R noVNC-1.3.0/* /var/www/html/  
rm v1.3.0.zip  
rm -r noVNC-1.3.0/
```

Now, going to **<http://vncproxy.puqcloud.com/vnc.html>** will open the noVNC client.

## Generate SSL certificate and install it in WEB server using certbot

```
certbot --nginx -d vncproxy.puqcloud.com
```

**In order for the certificate to be updated automatically, you must add to the crontab**

```
crontab -e
```

```
0 12 * * * /usr/bin/certbot renew --quiet
```

## NGINX virtual host configuration

Make the necessary settings in your domain configuration file in the nginx server

```
nano /etc/nginx/sites-available/default
```

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    root /var/www/html;
```

```

index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
    try_files $uri $uri/ =404;
}

server {

    root /var/www/html;

    index index.html index.htm index.nginx-debian.html;
        server_name vncproxy.puqcloud.com; # managed by Certbot

    location / {
        try_files $uri $uri/ =404;
    }

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/vncproxy.puqcloud.com/fullchain.pem; # managed by
Certbot
    ssl_certificate_key /etc/letsencrypt/live/vncproxy.puqcloud.com/privkey.pem; # managed by
Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot


    location /vncproxy {
        proxy_pass http://127.0.0.1:8080/vncproxy;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}

```

```
server {
    if ($host = vncproxy.puqcloud.com) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80 ;
    listen [::]:80 ;
    server_name vncproxy.puqcloud.com;
    return 404; # managed by Certbot
}
```

```
service nginx restart
```

## Next step is to install vncproxy

```
apt-get install screen -y
cd /root/
wget http://download.puqcloud.com/WHMCS/servers/PUQ_WHMCS-Proxmox-KVM/vncproxy/vncwebproxy
chmod +x vncwebproxy
```

We will run the script in the screen.

The first script parameter is a unique key. noVNC WEB proxy key - in WHMCS module parameter.

```
screen
./vncwebproxy puqcloud
```

After a successful launch, you can watch the execution log in the console.

```
root@vncproxy: ~# ./vncwebproxy puqcloud
[./vncwebproxy puqcloud]
proxmox- test. uuq. pl59002022/09/11 19:11:08 [vncproxy][debug] ServeWS
2022/09/11 19:11:08 [vncproxy][debug] request url: /vncproxy/proxmox-
test. uuq. pl/5900/d91bac199c2ce79392d8e175076e3780
2022/09/11 19:11:13 [vncproxy][info] close peer
[GIN] 2022/09/11 - 19:11:13 | 200 | 4.740249024s | 79.184.10.217 | GET
"/vncproxy/proxmox- test. uuq. pl/5900/d91bac199c2ce79392d8e175076e3780"
```

# Security

The security setting for the server should meet your standards.

Do not forget that for the correct operation of the server, you must allow connections to 80/443 ports. And outgoing connections to the PROXMOX server.

---

Revision #20

Created 11 September 2022 15:55:42 by Ruslan

Updated 11 June 2024 09:35:11 by Yuliia Noha