

# Cloud VPS

- [Basic Server Setup, Caddy, Docker, JDownloader](#)
- [JDownloader](#)
- [Tunneling Basic Services \(Jellyfin, Web\) with Caddy and Tailscale](#)
- [Tunneling Minecraft Server \(tcp only\) with Nginx](#)

# Basic Server Setup, Caddy, Docker, JDownloader

## Creating the VM in oracle cloud.

1. Go to instances, new instance.
2. Select the Always Free image, ARM or x86. 1 core only, recommended 4GB RAM, should be exceed 6 GB.
3. Choose Ubuntu image.
4. Download the SSH key and name it accordingly.

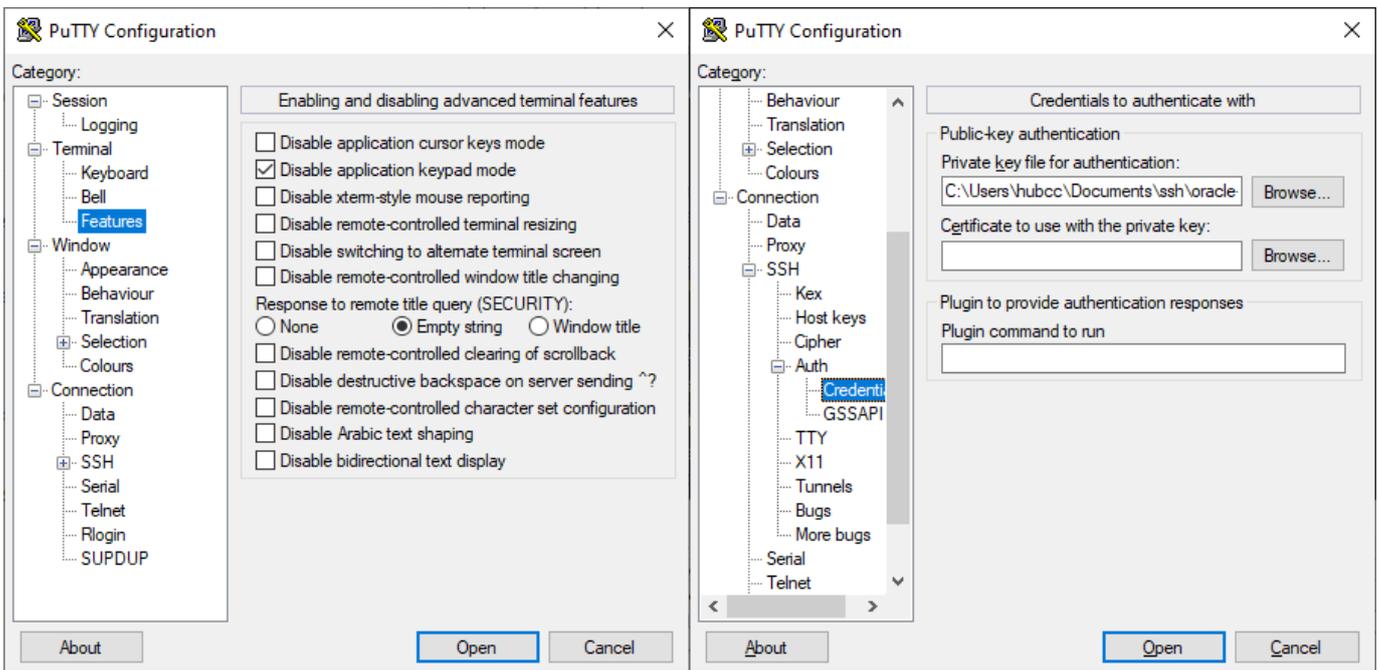
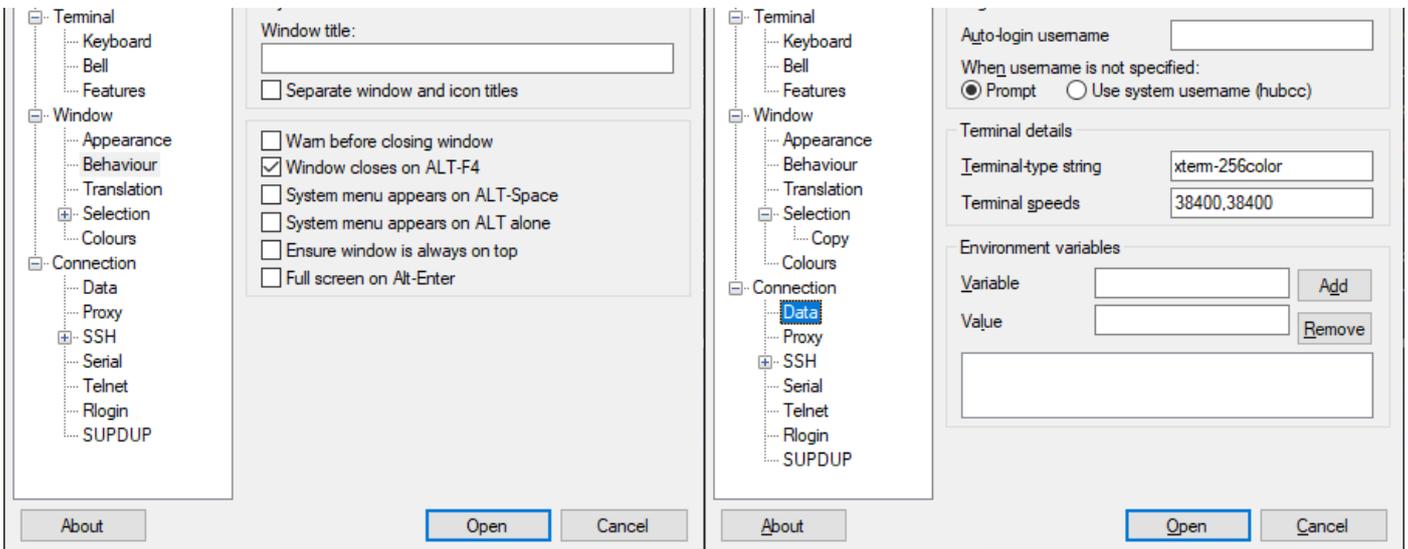
## Key Pair

Using PuttyGen.

- Place the key in `./ssh/openssh_keys`
- Open PuttyGen, conversion -> import keys
- Save the key files as ppk file in root folder of `./ssh`

Putty

- Grab the IP address in the cloud console
- Give a name in saved sessions
- Go to behavior, choose these options
- Under Data, make sure Terminal-type string is `xterm-256color`
- Under Terminal -> Features, check "disable application keypad mode" to fix issues with nano
- The private key needs to be load in Connection -> SSH -> Auth -> Credentials



To get the IP address of the VPS at any time

```
curl ifconfig.me
```

## Basic Setup + Docker

1. Installing Caddy web server (simple to use reverse proxy), lightweight, easy and no need for docker. (Nginx is also a good candidate for reverse proxy as the command is easy to memorize and does not require consulting documentation sites. However, the syntax for nginx is extremely complex compared to caddy and might not be easily memorized.

<https://caddyserver.com/docs/install#debian-ubuntu-raspbian>

```
sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --dearmor -o
/usr/share/keyrings/caddy-stable-archive-keyring.gpg
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee
/etc/apt/sources.list.d/caddy-stable.list
sudo apt update
sudo apt install caddy net-tools
# net-tools is good utility, optionally can install firewall-cmd or nginx
# sudo apt install firewalld nginx
```

## 2. Install Docker

<https://docs.docker.com/engine/install/ubuntu/>

```
sudo apt-get update
sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg

echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin docker-
compose
# code modified to install docker-compose, each space in paragraph indicates a separate step
in their official blog
```

```
sudo groupadd docker \
sudo usermod -aG docker ubuntu
newgrp docker # activate docker group immediately
```

The machine needs to be rebooted from Oracle Cloud console to finish installation.

## JDownloader

<https://hub.docker.com/r/jlesage/jdownloader-2>

```
docker run -d \  
  --name=jdownloader-2 \  
  -p 5800:5800 \  
  -v $HOME/appdata/jdownloader-2:/config:rw \  
  -v $HOME/Downloads:/output:rw \  
  --restart unless-stopped \  
  jlesage/jdownloader-2
```

If port forwarding configured properly, entering ipaddress:5800 should work. If not open ports manually.

```
sudo iptables -I INPUT 6 -m state --state NEW -p tcp --dport 5800 -j ACCEPT  
sudo netfilter-persistent save
```

## Other Useful Ports

```
sudo iptables -I INPUT 6 -m state --state NEW -p tcp --dport 443 -j ACCEPT  
sudo iptables -I INPUT 6 -m state --state NEW -p tcp --dport 80 -j ACCEPT  
sudo iptables -I INPUT 6 -m state --state NEW -p tcp --dport 25565 -j ACCEPT  
sudo iptables -I INPUT 6 -m state --state NEW -p tcp --dport 19132 -j ACCEPT  
sudo iptables -I INPUT 6 -m state --state NEW -p udp --dport 25565 -j ACCEPT  
sudo iptables -I INPUT 6 -m state --state NEW -p udp --dport 19132 -j ACCEPT  
sudo iptables -I INPUT 6 -m state --state NEW -p udp --dport 51820 -j ACCEPT  
sudo netfilter-persistent save
```

## Alternative (firewall-cmd)

```
sudo apt install firewalld
```

Firewalld is a CentOS package, it may be unstable and crash, but command easy to memorize.

```
sudo firewall-cmd --zone=public --add-port 19132/tcp --permanent  
sudo firewall-cmd --zone=public --add-port 19132/udp --permanent  
sudo firewall-cmd --zone=public --add-port 25565/tcp --permanent  
sudo firewall-cmd --zone=public --add-port 25565/udp --permanent  
sudo firewall-cmd --zone=public --add-port 80/tcp --permanent
```

```
sudo firewall-cmd --zone=public --add-port 443/tcp --permanent
sudo firewall-cmd --zone=public --add-port 5800/tcp --permanent
sudo firewall-cmd --reload
```

## Troubleshooting network

For firewall-cmd, use this command to check all open ports.

```
sudo firewall-cmd --list-all
```

Using netstat, or pipe it to grep

```
netstat -tln
# | grep 8080 etc...
```

## Configuring JDownloader

- Go to the JDownloader WebUI
- Go to Settings
- Under general, change the max number of downloads (2) and DL per hoster (1) to minimize issues

**Download Management**  
Connection limits, Download order, Priorities, ... set up the Download controller details.

Max. simultaneous Downloads

Maximum of simultaneous downloads per host

- Go to MyJDownloader and configure MyJDownloader account

**My Account**  
Enter your MyJDownloader Logins below. If you do not have an account yet, click the Link above and register for an account. It's free!

Email

Password

Enter the device name of this JDownloader instance. You can manage different JDownloader instances in one MyJDownloader Account.

Device Name

Connection established. Great!  
Current connections: 0

- Go to extension modules, install and enable "folderwatch"

Enabled	Name	Installation	Description
<input checked="" type="checkbox"/>	Folder Watch	<input type="button" value="Remove now"/>	Add Links to JDownloader just by putting Linklist files (*.crawljob) in a special folder on your harddisk.
<input type="checkbox"/>	Event Scripiter	<input type="button" value="Install now"/>	Listens to internal JDownloader event, and allows to execute relevant scripts (Javascript). This makes JDownloader even more customizable.
<input type="checkbox"/>	Scheduler	<input type="button" value="Install now"/>	Define time schedules to execute actions, start downloads,...
<input type="checkbox"/>	JD Shutdown	<input type="button" value="Install now"/>	Automatically shut down, suspend, hibernate your PC or close JDownloader when downloads are finished.
<input type="checkbox"/>	Info Bar	<input type="button" value="Install now"/>	Show a tiny transparent overlay window that can be used to check the Download status or to drop links.
<input type="checkbox"/>	Support Chat	<input type="button" value="Install now"/>	Contact JDownloader Developers and Supporters
<input type="checkbox"/>	Translator	<input type="button" value="Install now"/>	This Extension can be used to edit JDownloader translations. You need a developer account to use this extension. If you do not want to translate, but just change the current language, pl...

The configuration for JDownloader is complete and should appear and be functional in WebUI.

[Advanced JDownloader](#) documentation will be covered in detailed in another section. It is recommended to close port 5800 after configuring to prevent others accessing.

## Basic Caddy Syntax (if applicable)

If the server that is being setup or restored needs functional service like bookstack or uptime-kuma, reverse proxy is needed.

```
sudo nano /etc/caddy/Caddyfile
```

```
{
    email weebly2x10@gmail.com
}

your-uptime-kuma.yoursubdomain.duckdns.org {
    reverse_proxy http://127.0.0.1:3001
}

wiki.yoursubdomain.duckdns.org {
    reverse_proxy http://127.0.0.1:6975
}
```

## Advanced

[Tunneling Jellyfin and other web services with tailscale and caddy](#)

[Minecraft Server tunneling via Nginx \(tcp only\)](#)

# JDownloader

After setting up JDownloader and it appears well in WebUI.

The section is useless now as UHDMV has shutdown and it's pointless to setup multiple automated JDownloader server on VPS.

# Tunneling Basic Services (Jellyfin, Web) with Caddy and Tailscale

This procedure is not reproducible yet. Rigorous testing is still required before being documented. Here are the known procedures.

The purpose is to tunnel normal web or network intensive traffic such as Jellyfin when faced with CG-NAT or similar situations (in this case locked down dorm internet), also configure hardware transcoding (in this case NVENC, but Intel QSV for future) to mitigate limitations with Canadian ISP(s).

## Jellyfin Install

<https://jellyfin.org/downloads/server>

Download and run the server installer.

## Jellyfin Server Configuration

### Tailscale (Windows Client)

<https://tailscale.com/download/windows>

Download, install and login.

### Tailscale (Linux Server)

```
curl -fsSL https://tailscale.com/install.sh | sh
```

```
sudo tailscale up
```

All the tailscale management is done in the WebUI.

MACHINE	IP	OS	LAST SEEN	
[REDACTED] Win Client	100.79.28.31	Windows 1.32.3	Connected	...
[REDACTED]	100.84.117.44	Android 1.32.3	Dec 6, 2022	...
[REDACTED] Server	100.64.145.75	Linux 1.32.3	Connected	...

The Windows client is given a tailscale network IP address in 100 range. Check if Windows client is pingable on server.

```
ping 100. x. y. z
```

Check if Jellyfin is running and tunneled properly on Oracle cloud. It should get a webpage html rather than unable to resolve host etc.

```
curl http://100. x. y. z: 8096
```

## Reverse Proxy

<https://wiki.calgaryab.duckdns.org/books/cloud-vps/page/basic-server-setup-caddy-docker-jdownloader>

Caddy installation and syntax is can be found on this page. Replace 127.0.0.1 with the tailscale IP address.

```
{
  email weebly2x10@gmail.com
}

movies.yoursubdomain.duckdns.org {
  reverse_proxy http://100. x. y. z
}
```

It is possible to set use the root domain (yoursub.duckdns.org) or a subfolder domain (movies.yoursub.duckdns.org) for Jellyfin. After configuring the Caddyfile.

```
sudo systemctl reload caddy
```

Use netstat to check port 80, 443 is being listened. Make sure to [port forward](#) Oracle VPS.

## Other Services

Follow the same syntax as the caddy file provided, if the root domain is used, then a subdomain must be used for other services.

## **Results**

Inconclusive yet, more testing required.

# Tunneling Minecraft Server (tcp only) with Nginx

Procedure not reproducible yet, will be documented later.