

Welcome to the Discord-Linux Docs!

Welcome

Thank you for coming by and checking out our docs taking a deeper dive into what our service can offer you 100% for free.

Our team has spent an abundance of time and energy to make all of this possible, and we are very excited to meet you and see the different things people will build on our platform.

This is more than a developer hosting service, this is a community based on the love to use Linux and the drive to learn as much as we can to succeed in the art of programming.

If you are like us and would love to learn how to progress your knowledge in either programming or Linux, please follow to the next section and join our discord server.

Joining our Discord

If you have not done so yet, please join our discord server.

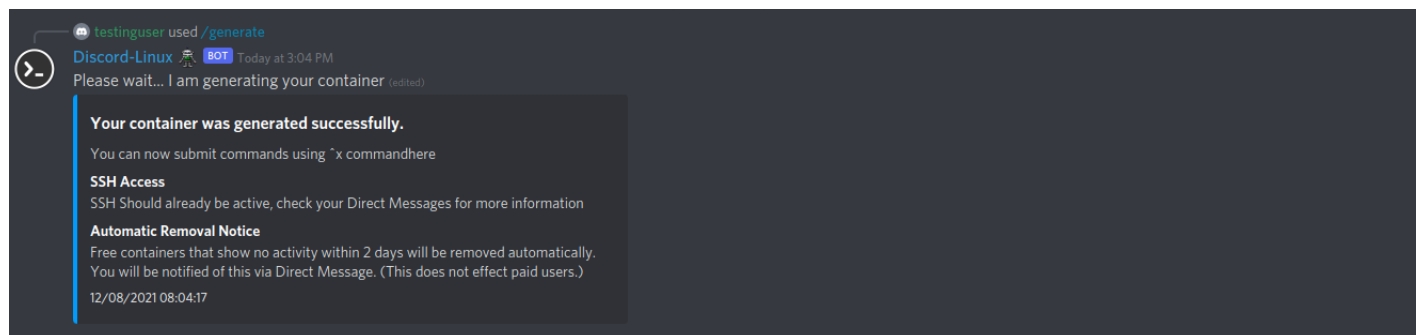


Join our discord server here: [CLICK ME](#) 

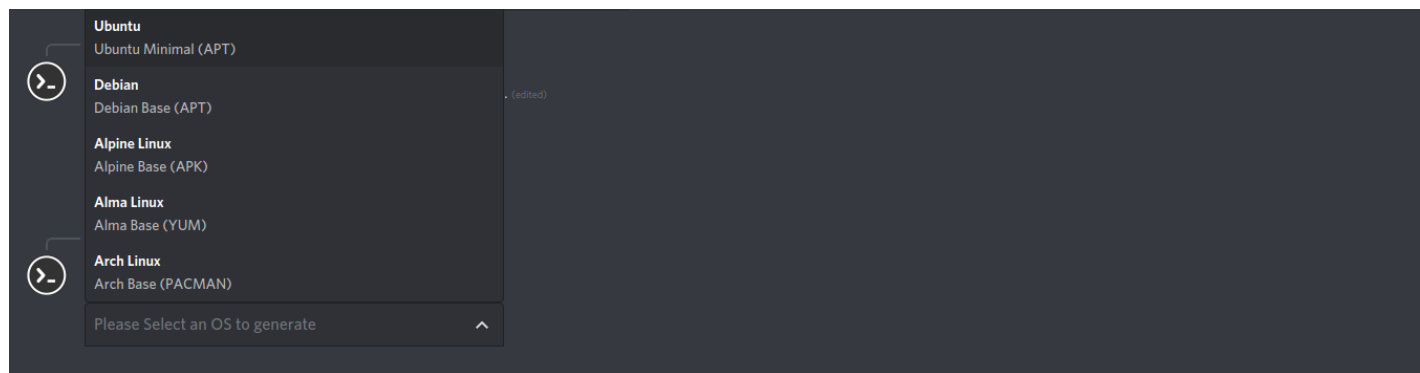
Generating a Container

Once you are a member of our discord server, you may automatically begin generating a container.

Here is an example of this using the `/generate` command which will automatically select Ubuntu.



If you would like to choose an Operating System for your container, you may use `/genos` like so:



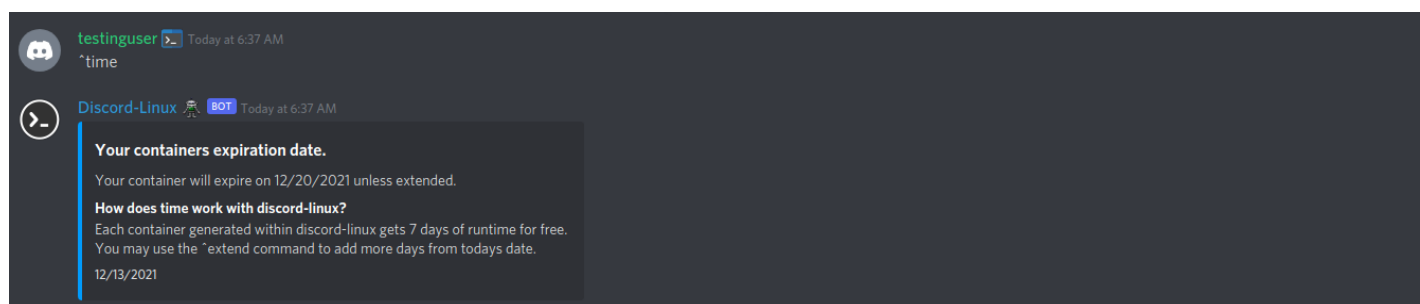
Simply use the drop down to select the OS you would like to use.

- ▶ Inactive containers which doesn't have any running processes will be automatically removed by means of automation.
- ▶ The containers root password and SSH connection information will be sent through Direct Message to you.

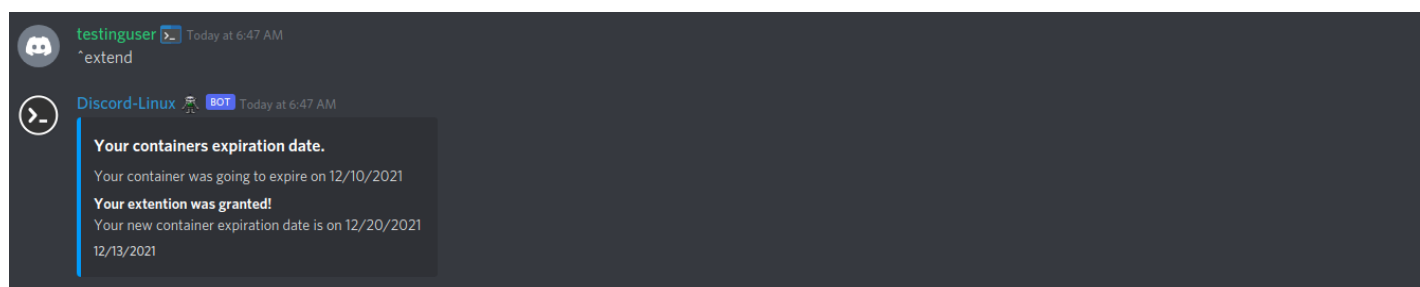
Container Expire Time

Each container generated unless on an upgraded package will automatically be removed after 7 days, If you would like a free container to run past the 7 day expire time, you may use the ^extend command to add 7 days from date its ran.

Here is an example of checking your containers time:



Here is an example of a time extension:



Securing the Container

You are provided with a random password through Direct Message once container generation has completed, however, you may wish to change this password.

Secure your container by running the following command:



```
echo 'root:newpasswordhere' | chpasswd
```

You can also login via SSH and use passwd like this:



```
passwd root
```



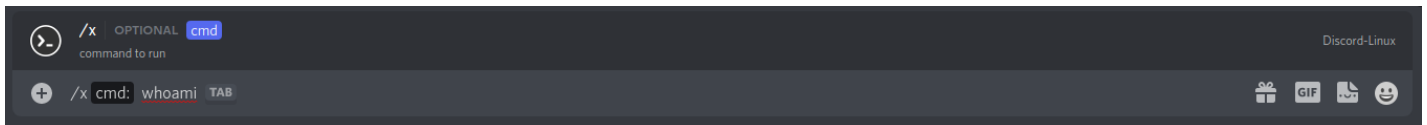
Warning: Consider not running these two commands in public as some servers does set up a discord bot that logs deleted message which logs almost every single deleted messages. Run these commands through the bot's DMs instead!

Submitting Commands

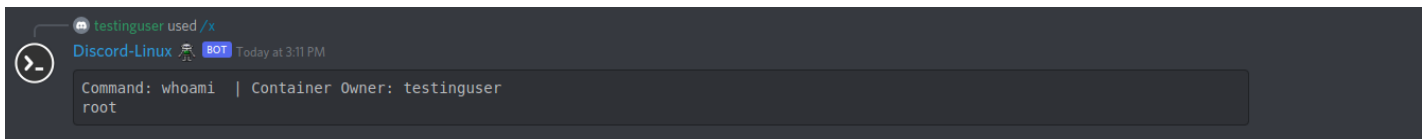
Once your container has been generated, you can submit commands to it, this can be done using the “/x command” syntax OR you may Direct Message the bot for a more “SSH” feel.

Each container is linked using Discord ID, so it's not possible for other users to submit command as you, so your container is private.

Here is an example of the “/x command” syntax



This is showing the command as it finishes running:



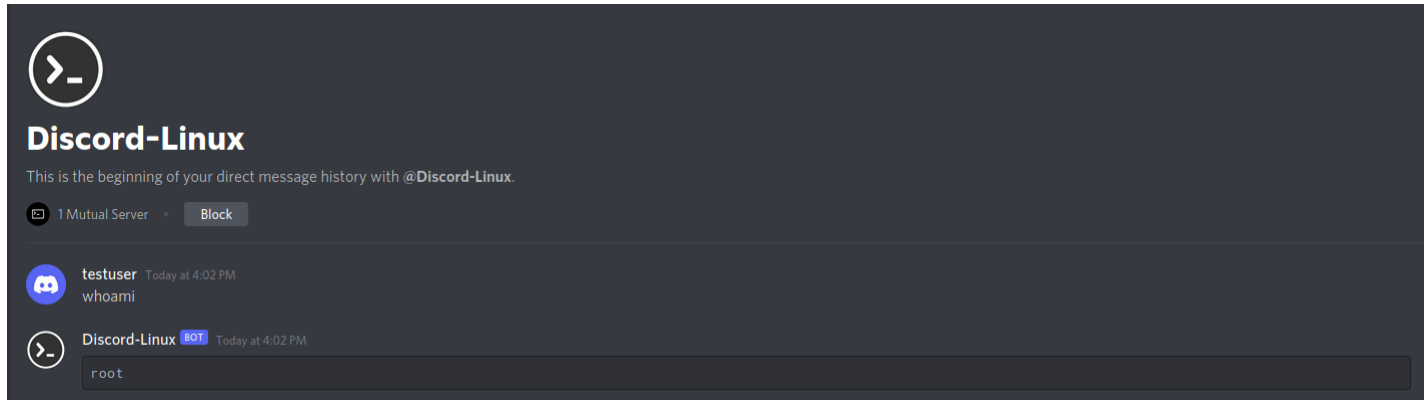
As you can see in the above, the “/x command” syntax will allow the user to submit any linux command in any public discord channel that allows the bot access.

All users interact under the “root” user.

Direct Message CMDs

You can also submit commands directly through the Direct Message for even more privacy.

Here is the same command but sent through Direct Message.



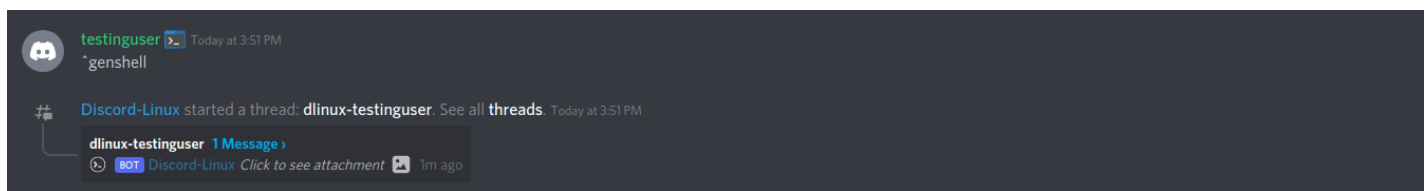
As you can see, submitting commands through Direct Message gives the commands a lot more “SSH” feel.

Create Thread Shell

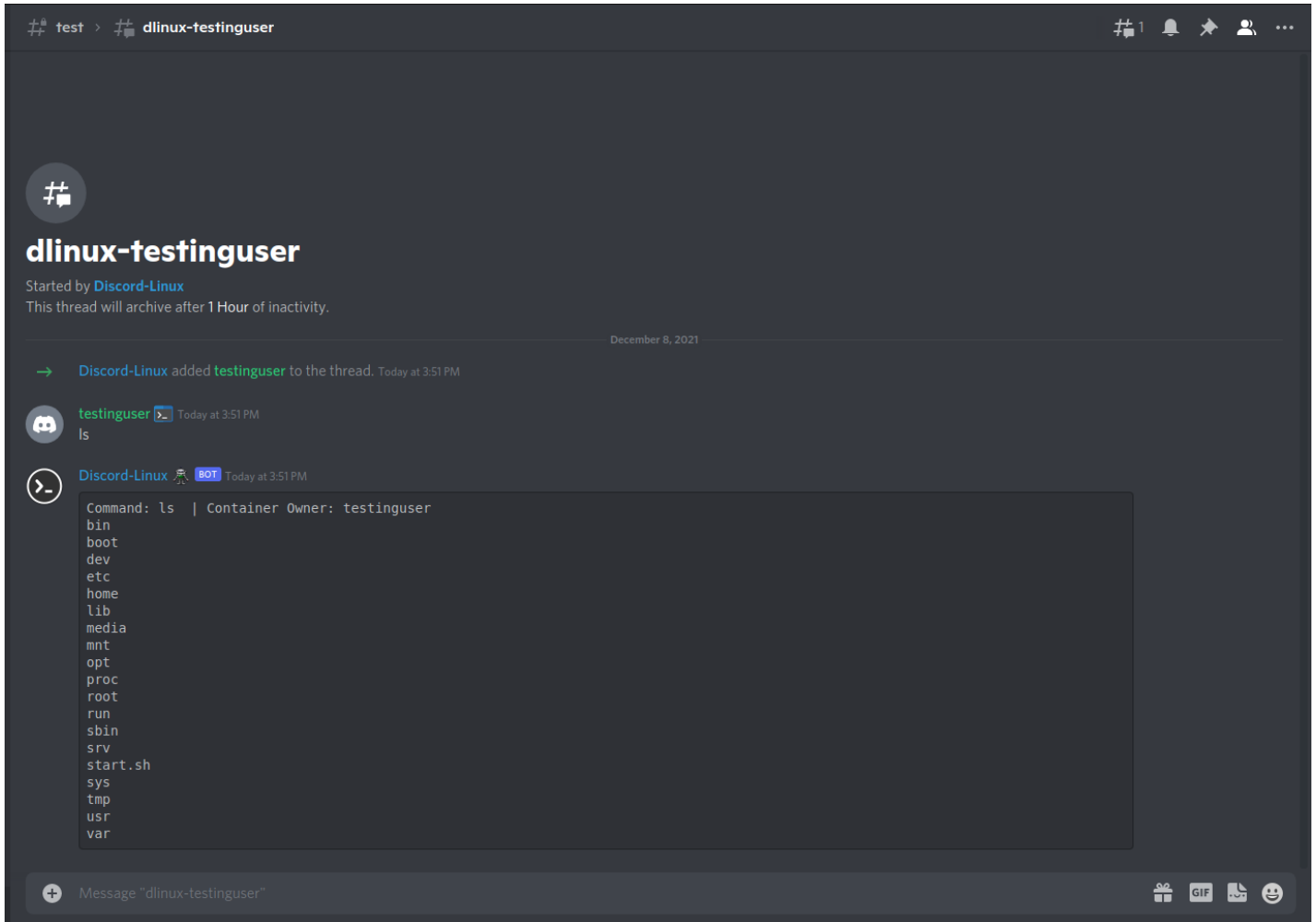
You can use the ^genshell command to generate your own shell channel.

These channels do not have a prefix and act just like an SSH client.

Here is an example of generating a shell channel:



Now you can access the thread and submit commands, prefix free!



If the server is boosted enough at the time for a private shell so no one else can join your shell, you may attempt to use ^pvtshell

File System Navigation

Just like any Linux Environment, you will need to browse the filesystems directories and create files and directories.

In this example, we are going to change to the root directory of the container:



Once submitted, the bot will confirm its working directory:



Now, you can ls the directory to view its contents using /x cmd ls

```

Command: ls | Container Owner: testinguser
bin
boot
dev
etc
home
lib
media
mnt
opt
proc
root
run
sbin
srv
start.sh
sys
tmp
usr
var

```

Within Discord-Linux there is no use of ~ or ... So we coded a custom solution for this. In this example, I will show you how to go backwards in navigation:

We are currently in the /home/raven/test directory:

```

/home/raven/test

```

Now we use /x cmd gobk to change to /home/raven:

```

/home/raven/test /x cmd: gobk
/home/raven

```

Startup Script

Upon your first few uses of the containers, you will see `start.sh` in the root directory of the container

This script is called when the container boots up, this is where you define your systemctl service starts or any commands you would like to run during boot time.

Here I will break down each line in the stock file:

```

#!/bin/sh < - Tells the system to use SH
export CID=$(basename $(cat /proc/1/cpuset)) < - Container ID as a system VAR
service ssh start < - This starts the SSH Server on boot
sh /var/tools/getsystemctl.sh < - Replacement for systemctl which is required
tail -f /dev/null <- To keep the container running! DO NOT REMOVE!

```

- ▶ Make sure you edit this file via SSH and not the bot!

- ▶ Make sure to `chmod +x /start.sh` when you are done editing the file.

Notification Service

Now, we have a built-in Notification Service which is installed along with our AutoInstallers. This is a Linux binary which gives your container the ability to DM you about anything you wish.

Our best example of this, is within our AutoInstallers which some of them do take a lot of time to finish installing. Due to this, I wanted to provide a way for users to run the installer and then just walk away. They will be notified through DMs with a direct link to their installation when it's completed.

To install it manually, use:



```
wget -q -O - https://ssh.surf/notify | bash
```

Then, you can call it by running the binary `notif`. The bin is installed in `/usr/bin/notif`

To use, simply pass text:



```
$ notif "This is a test message"
```

I also coded it to form a message out of args like so:



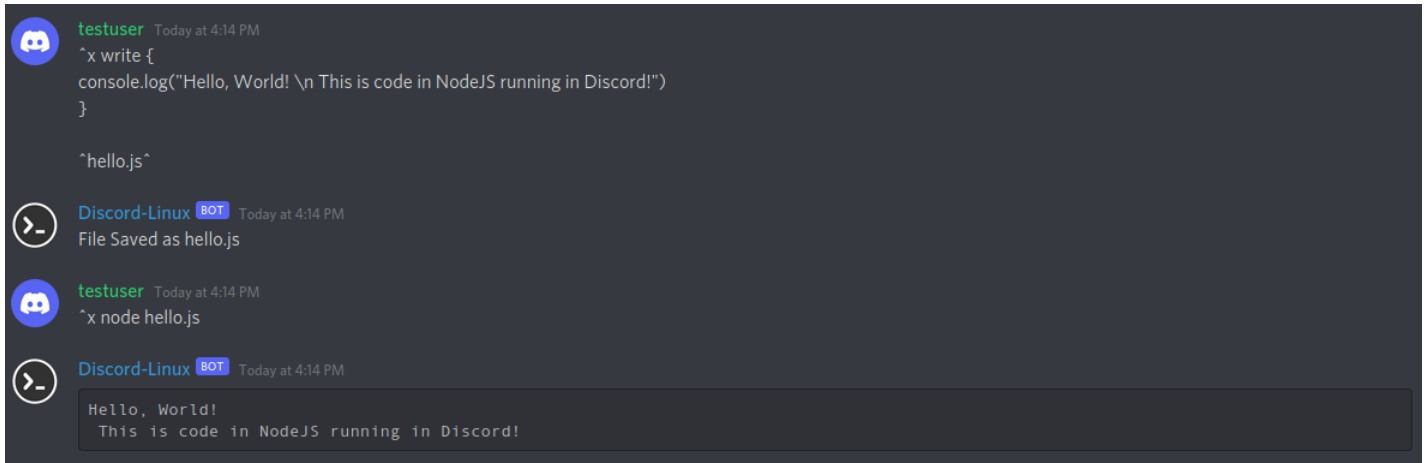
```
$ notif This is a test message
```

Both commands will output the same

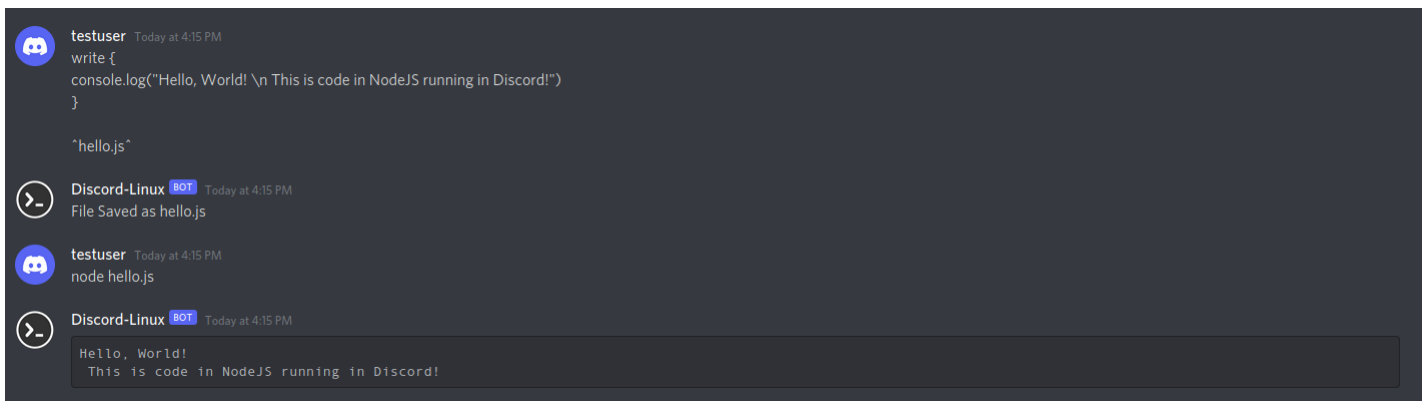
Writing Files

You can code in any language you wish using Discord-Linux, this guide will show a NodeJS example application as NodeJS and PM2 are installed by default.

You can write a file using the following syntax



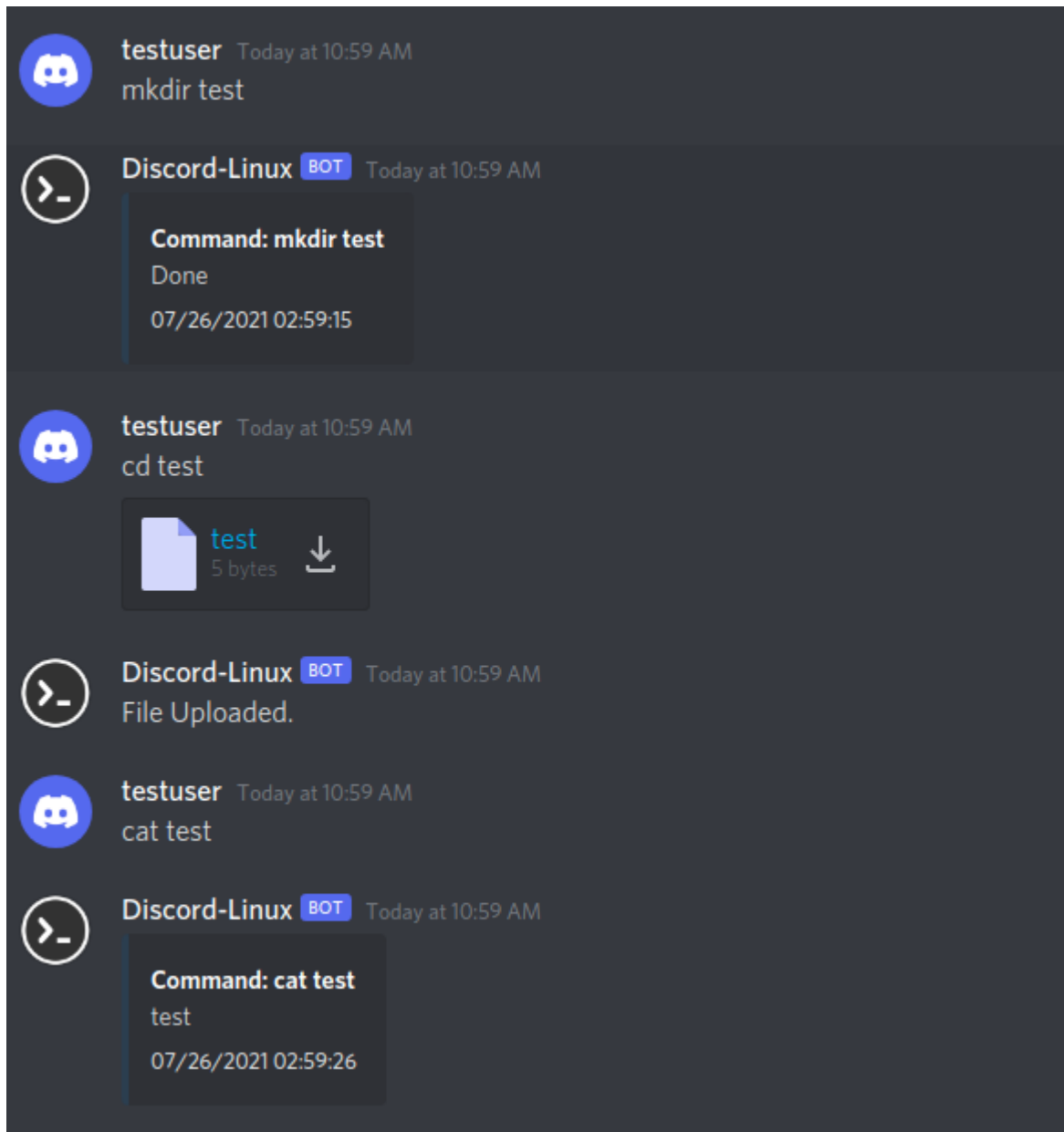
This can also be done through Direct Message like so:



Uploading files

You can upload files using default discord upload feature either with drag and drop or with the upload button.

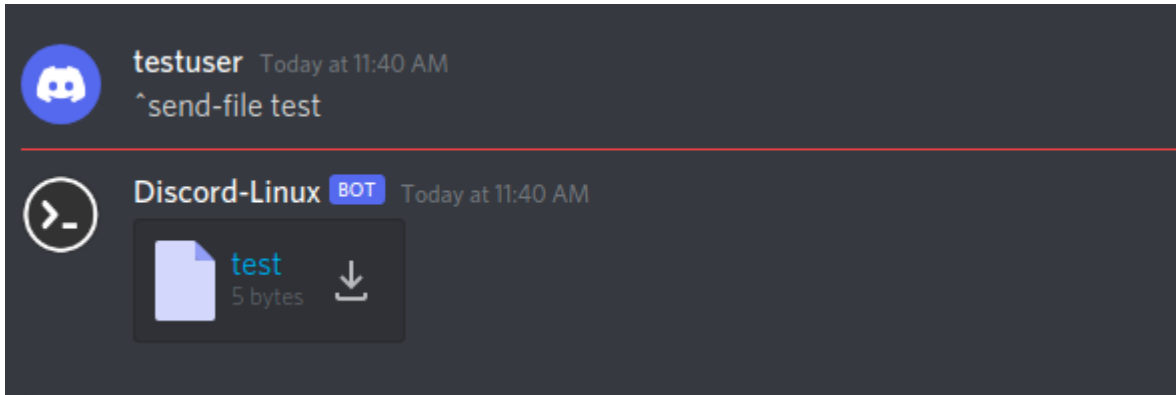
- ▶ This feature fully depends on your accounts upload limits
- ▶ To upload in a public channel include the keyword "upload" with your discord upload and the file will be added to your container.



Files --> Discord

- ▶ This feature fully depends on your accounts upload limits as well as the bots upload limits, some files may fail to send if too large.
- ▶ This feature can be used in any channel as well as Direct Message:

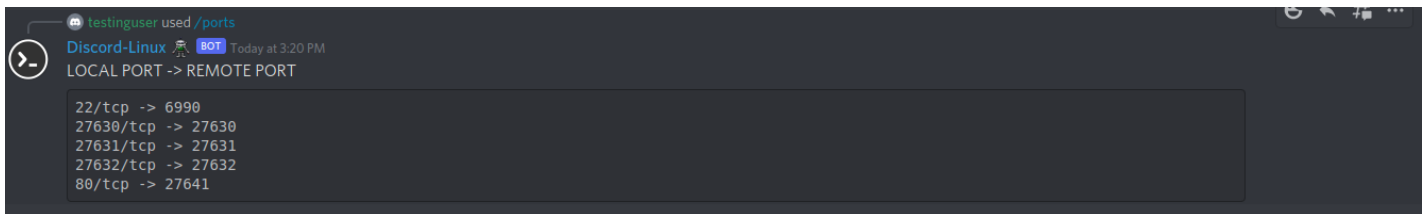
You can send your files to discord using the ^send-file (or /openfile) command like so:



Checking Open Ports

Each container gets a set of 5 ports open, one is dedicated to SSH and is binded to port 22 on your container, the other is dedicated to HTTP Server and that is binded to port 80 on your container.

You also get 10 random ports assigned to your container for other usages, here is how to check your ports with the `/ports` command:

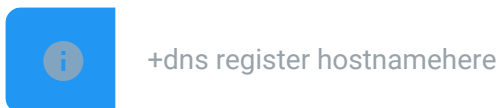


Hostname DNS & VHOST

If you would like to register a subdomain to be used as a hostname for your container, you can use our custom Hostname Generator

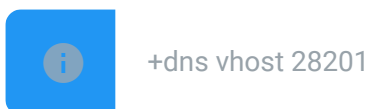
There are only two steps involved:

Step 1) Generate an A record within DNS to do so run the following command:



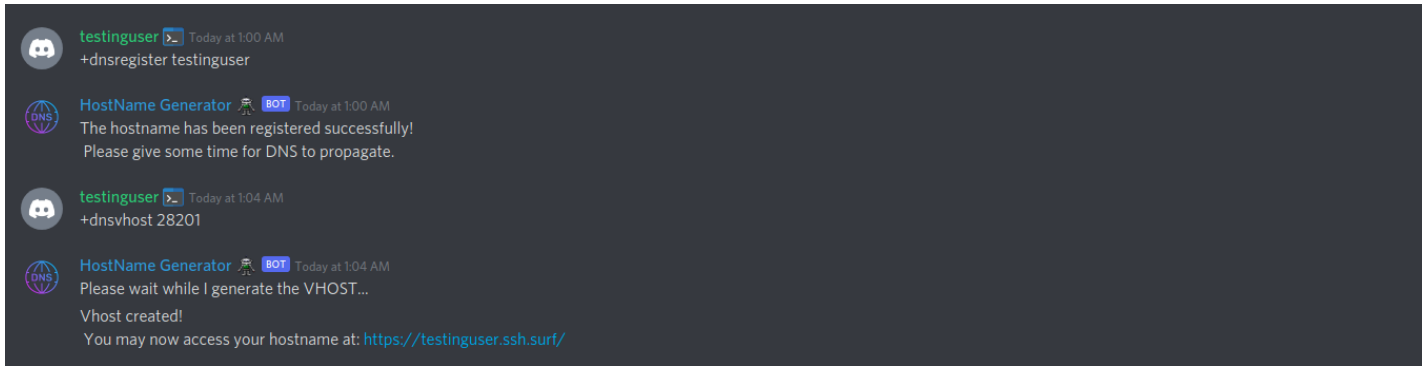
(This will register `hostnamehere.ssh.surf`)

Step 2) Generate a VHOST within Apache. To do so run the following command:

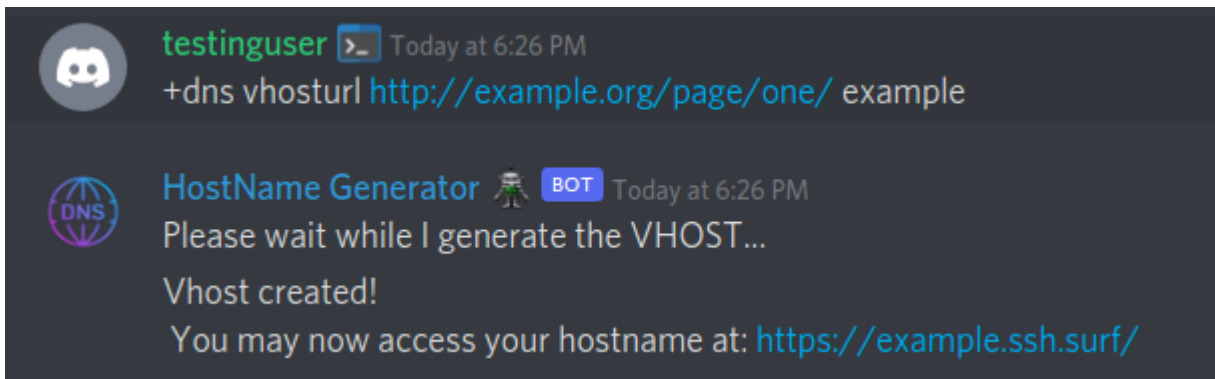


(This will bind the port 28201 to your hostname)

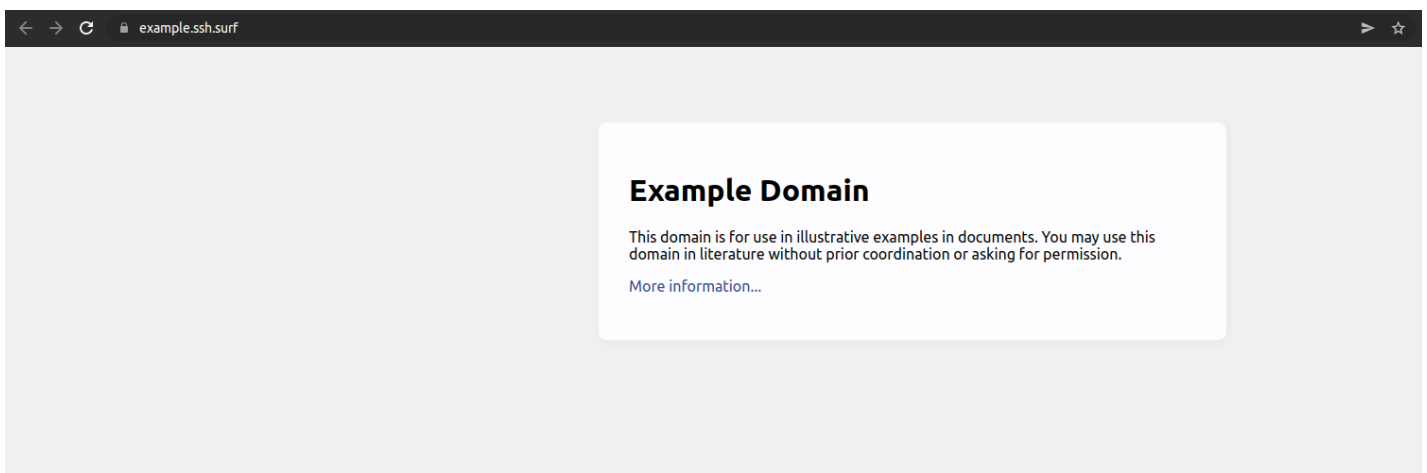
Here is an example of this working:



You can also map URLs to subdomains like so



This does infact result in the URL being accessible from the indicated subdomain



Custom DNS & VHOST



This service is in beta and may not work properly.



Need a domain? Get one free: <https://www.freenom.com/en/index.html?lang=en>

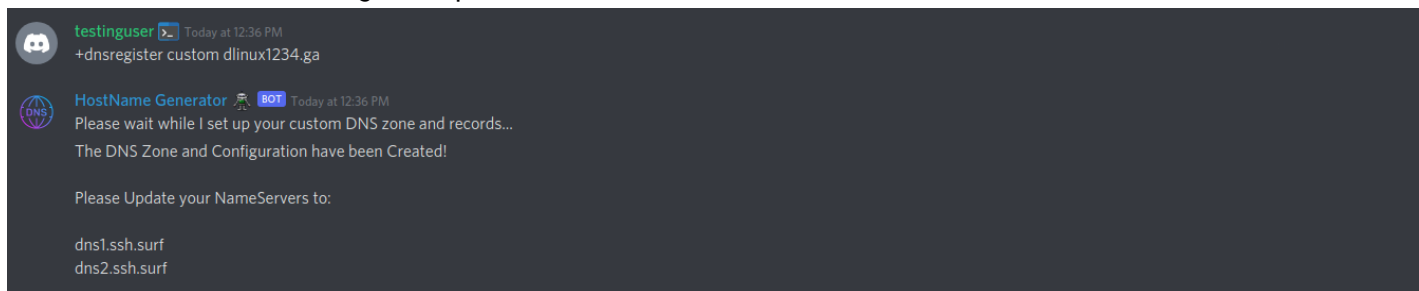
If you own your domain and would like to use it with our service, there are just few setup steps to go through:

Step 1) Generate the DNS Record, you can do so with the following command:



```
+dns register custom dlinux1234.ga 
```

Here is a screenshot showing the expected result:



Step 2) Change the domains Nameservers to the following:



```
dns1.ssh.surf  
dns2.ssh.surf
```

► Step 3) Wait for the DNS Records to propagate, you may use the site below to test:



<https://dnschecker.org/#NS/dlinux1234.ga>



Replace the domain with your own preferred domain

Once you see all locations testing with a green mark, move to the next step.

Step 4) Generate a certificate for your custom domain. This can be done with the following command:



```
+dns gencert
```

Here is a screenshot showing the expected result:



This now means that you have a valid SSL Certificate generated, now we can move to step 5 and generate the VHOST.

Step 5) Generate the VHOST, you do this by inputting a port to bind the domain to like so:

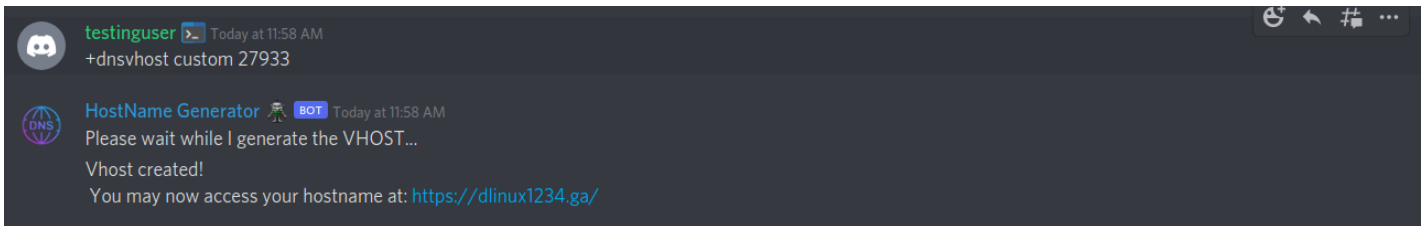


+dns vhost custom 27933



This will fail if step 4 has yet to be completed!

Here is the expected output:



If you see the above output, your domain is ready! Congrats!



This is the test user webserver

Custom Subdomains



This requires you to already have a Custom Host Name already set up within the above section.



The system will automatically attempt to renew your certificate as it expires, however, if this fails, simply request a new certificate by re-running the gencert command.

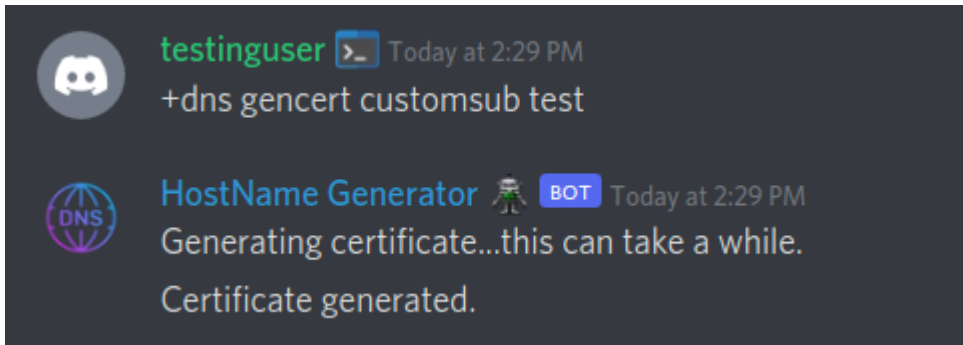
Once you have added a custom hostname, You can now generate and use an unlimited number of subdomains with your custom hostname.

This can be done following the below steps:

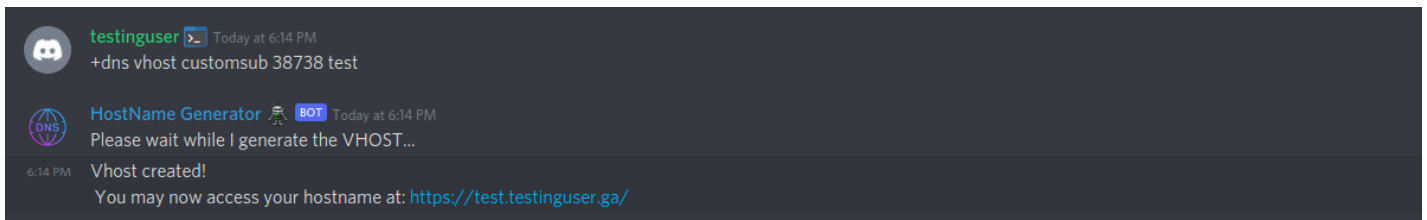
1. Register The Subdomain



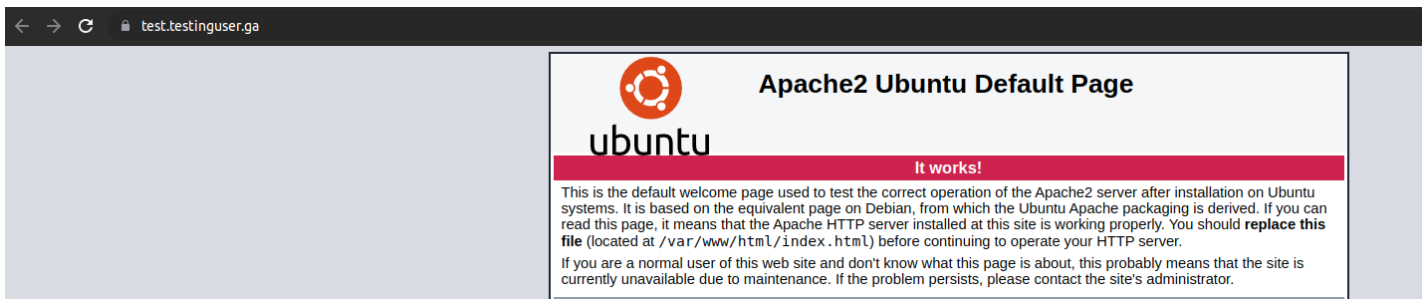
2. Generate a certificate for the subdomain



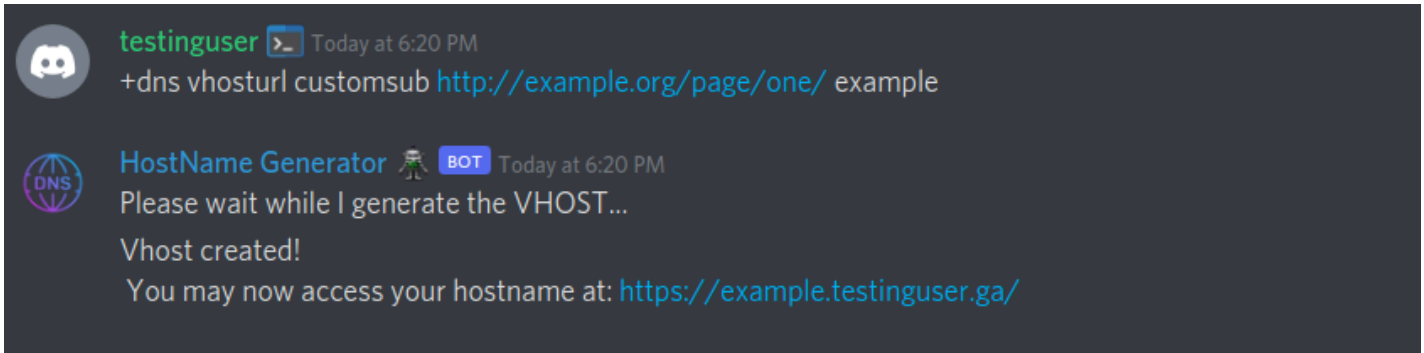
3. Generate a VHOST



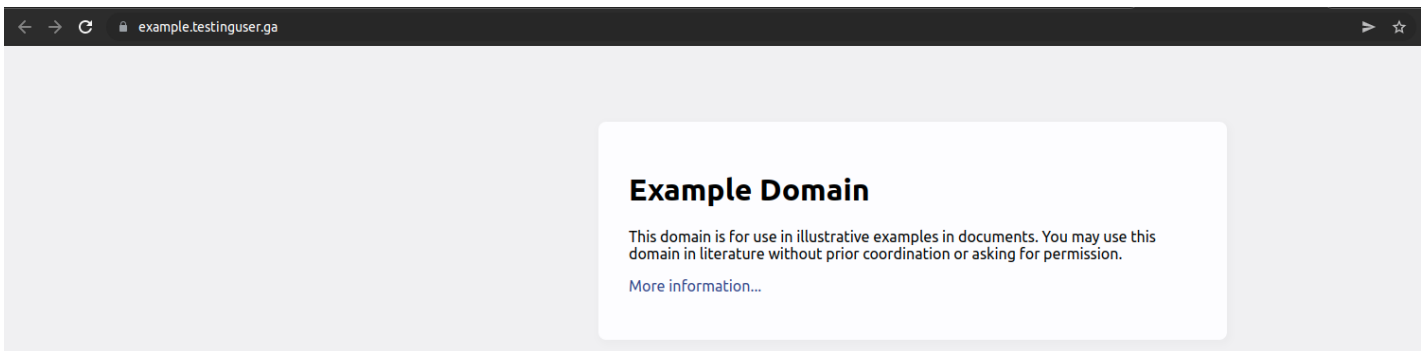
The subdomain will now be accessible



You can also map a URL to a subdomain using the vhosturl function like so



Which does result in the URL being accessible at the subdomain indicated



Resource Usage

You can check your container resource usage within Discord using `/stats` like so:



Exporting The Container

If you would like to either have a backup or download an export of your container to start up at another host, you can do so using the `/export` command, you will then see a DirectMessage containing a download link containing the entire filesystem of your container.

Destroy The Container

Say that if you want to start all over brand new, that is easy for our system to handle, simply use the `/destroy` command to destroy and remove your existing container and then, you can use `/generate` again to start fresh!



Note: Running this command will remove all of your existing data in the container, please think twice if you really want to run this command.



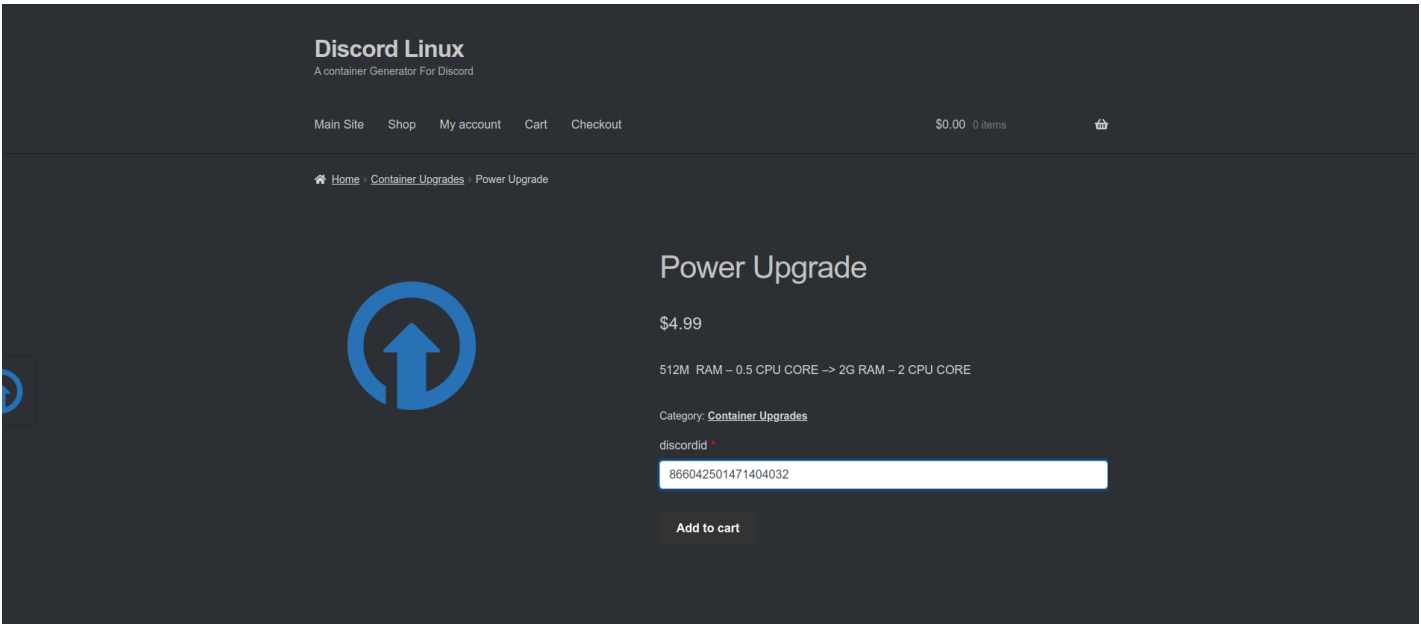
Container Upgrades

We offer container upgrades at our upgrade store <https://store.discord-linux.com/>

Before purchasing an upgrade at our store, please ensure you have a container running within Discord-Linux.

Within this example, I will show us upgrading our test users container to the "Power" plan.

At the upgrades page, add your DiscordID as shown below



Submit your order and complete your checkout using Paypal:

Billing details

First name *

Last name *

Company name (optional)

Country / Region *

Street address *

Apartment, suite, unit, etc. (optional)

Town / City *

State *

ZIP *

Phone *

Email address *

Your order

Product	Subtotal
Power Upgrade × 1	\$4.99
discordid: 866042501471404032	
Subtotal	\$4.99
Total	\$4.99

PayPal [What is PayPal?](#)

Pay via PayPal; you can pay with your credit card if you don't have a PayPal account.

Your personal data will be used to process your order, support your experience throughout this website, and for other purposes described in our [privacy policy](#).

Proceed to PayPal

Once you see the "Order Confirmation" screen, the upgrade has already completed and you will notice a license key:

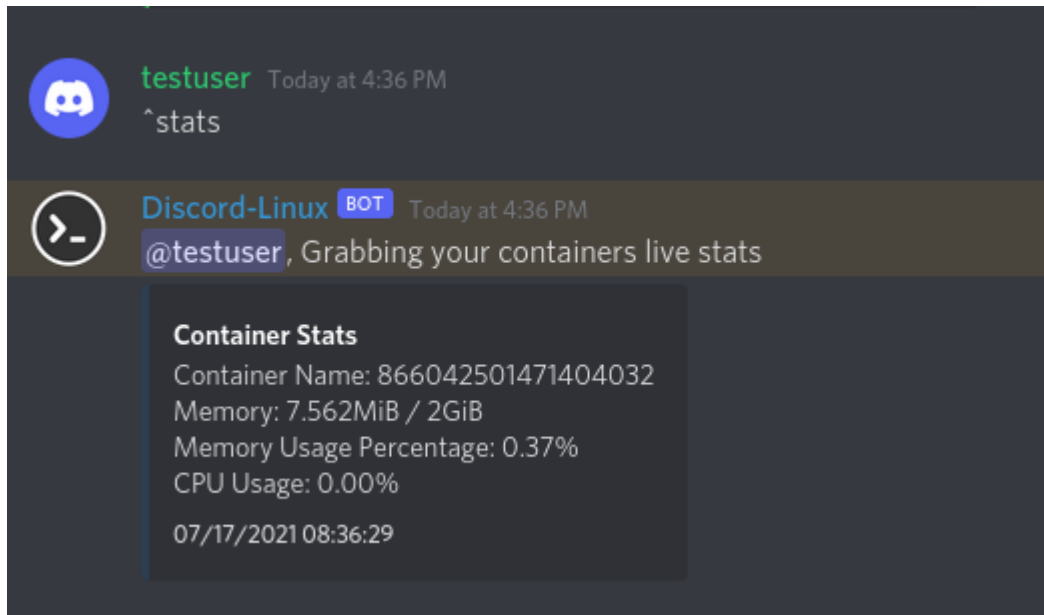
Order details

Product	Total
Power Upgrade × 1 discordid:866042501471404032	\$4.99
Subtotal:	\$4.99
Discount:	-\$4.99
Total:	\$0.00

License details

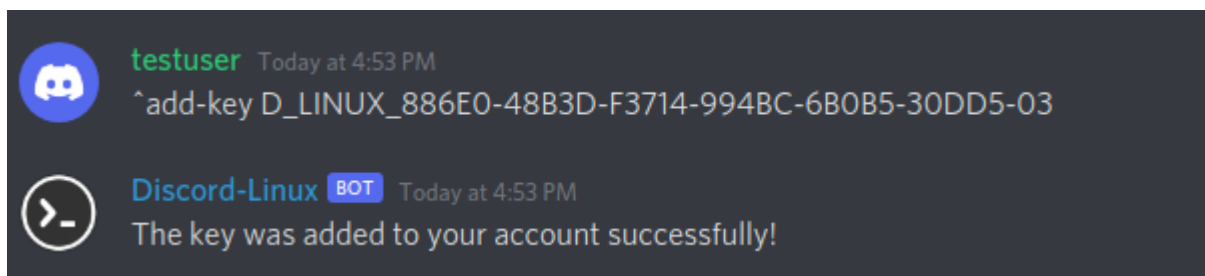
License key	Type
D_LINUX_886E0-48B3D-F3714-994BC-6B0B5-30DD5-03 - view my licenses	subscription

Here is showing the upgraded resources:

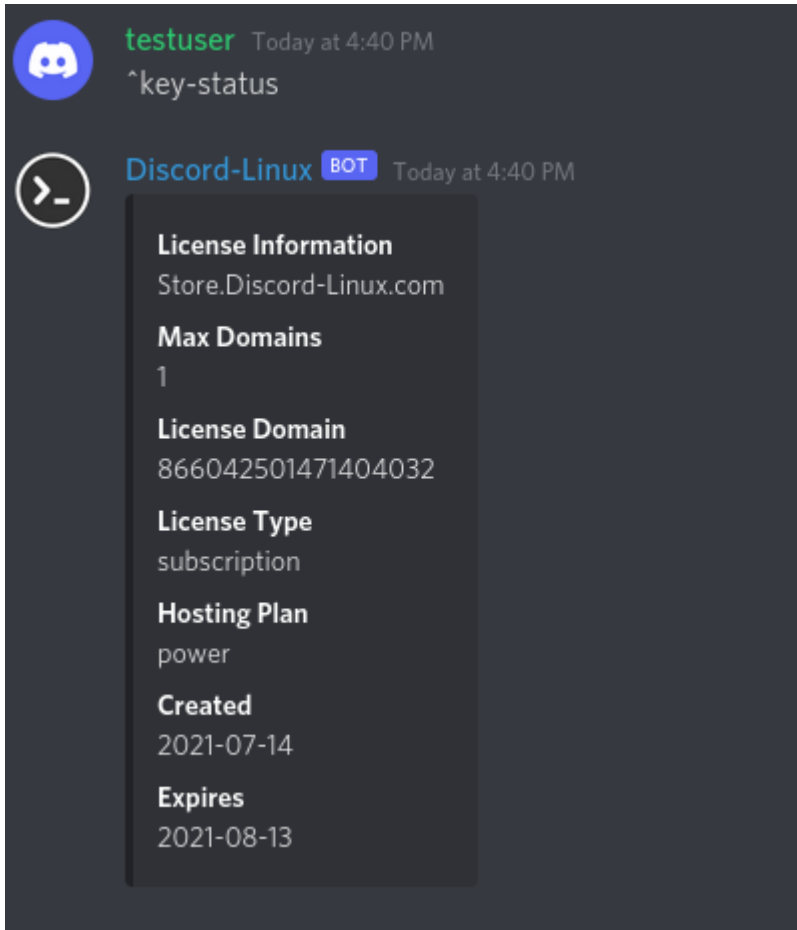


Now that you have a key, this can be used on your account to allow the upgrade for your account to persist during ^destroy and ^generate events.

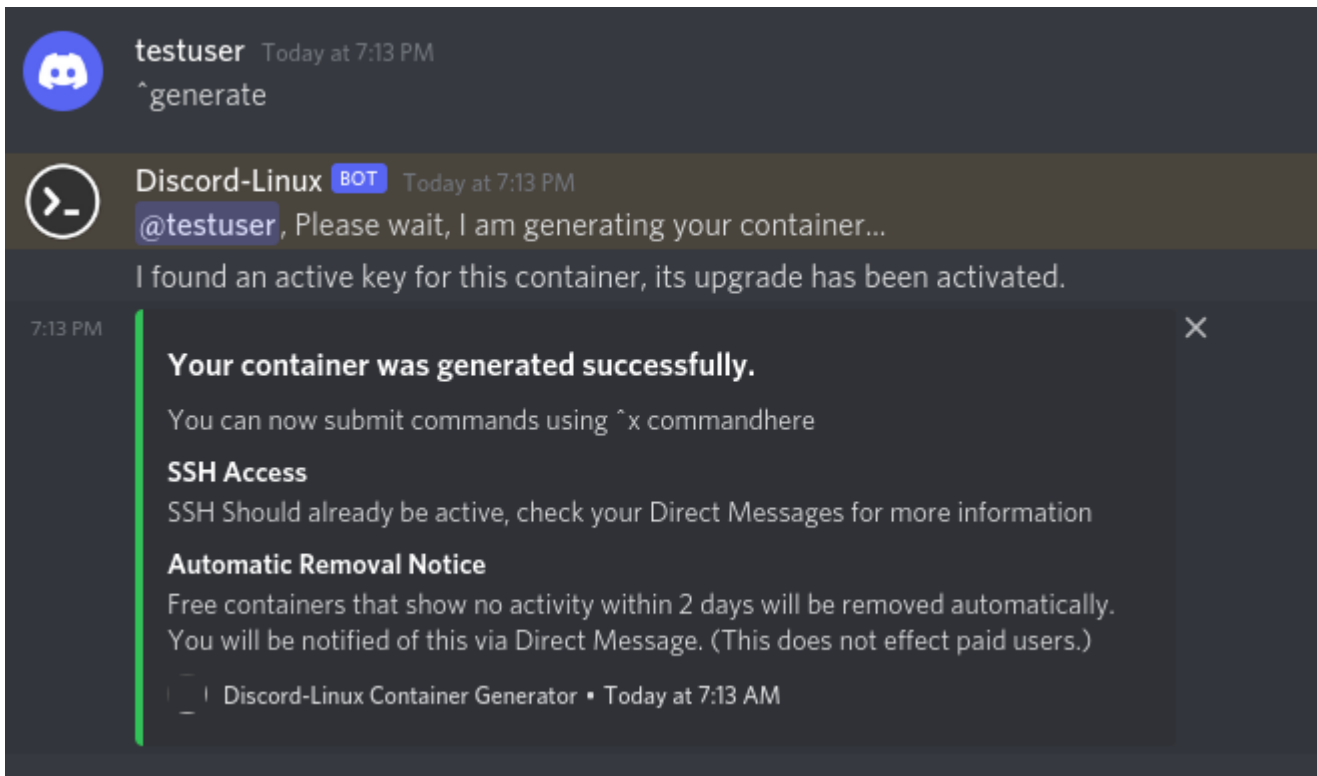
To apply the key use the ^add-key command like this:



Then, you can now check the details about this license key using ^key-status like so:



As you can see below, the key was detected and the container was upgraded automatically:



These keys need to be renewed each month, this can be done at:

<https://store.discord-linux.com/product/renew-resources/> 

Simply, input your DiscordID and select your plan, completing payment will add 30 days from the date of purchase.

Any key that expires gets automatically removed from the system and the container will be reset to default limits. Upgrades can be purchased at any time.

Donations

If you would like to donate to help keep our services online and provide much more powerful machines, please consider purchasing a container upgrade, or donate to us through the methods below:

Donate to Discord-Linux via Paypal



payments@discord-linux.com

Donate to Discord-Linux via Bitcoin



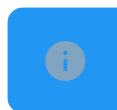
bc1qqf6g4s77dtf4v3ucpk0a4r90684x4c7jq5xcyw

Donate to Discord-Linux via LiteCoin



ltc1q69r39hy7ggu7cjmud9memkqa3hdhh4v775sm9

Donate to Discord-Linux via Ethereum



0x405132617843D0c86dC7fCc8766871d2A884cf70

Any donations are very much appreciated.

Content is available under the Public Domain, by Discord-Linux 2022. | Powered by [Wiki.js](#)