

SRTLA docker instructions with HTML generator has bitrate for NOALBS

20240421 fixed html generator to use the user name correctly and mismatched ports

I made a webpage you can open locally on your browser to generate the docker command and stream urls to save you time. Plug in the values then press generate.

srt docker generator.html

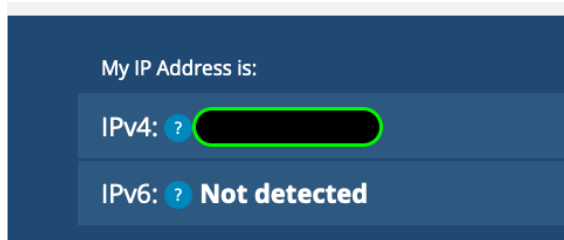
For this to work from outside your network you need to have a public IP address. Then you have to open the ports on your router. If you are running the SRT_receiver docker, OBS and NOALBS in the same network then you just have to open the video ingest port.

This guide gets the SRT docker setup. Opening ports on your router, setting up NOALBS and Other programs is up to you.

Thanks to datagutt for making this container <https://github.com/datagutt/bbox-receiver>

Find your public IP number using the webpage

<https://whatismyipaddress.com>



I blacked out my IP but lets take 87.116.10.98 for this example

Is your public IP in this range of private IP?

192.168.0.0 to 192.168.255.255 range

172.16.0.0 to 172.31.255.255 range

10.0.0.0 to 10.255.255.255 range

If yes then you can't run your own SRT server. You are behind a CGNAT.

What is CGNAT?

CGNAT (Carrier-Grade NAT) is a variant of NAT that is used by internet service providers (ISPs) to provide internet access to their customers. CGNAT works by allowing multiple customers to share a single, public IP address. This is done by assigning unique port numbers to each customer's device, similar to how PAT works.

CGNAT is used by ISPs to conserve public IP addresses, as **the pool of available IPv4 addresses has been exhausted**. IPv6 addresses, which have a much larger pool, are slowly being adopted, so many ISPs use CGNAT as a solution to the IPv4 address shortage.

You are essentially behind 2 routers. The ISP's router and your router. You can not program the ISP's router to forward the stream to your router because you do not have access to it.

I have a public IP!

1) Install docker from <https://www.docker.com/products/docker-desktop/>

2) Make a folder on your windows desktop named "srt_receiver"

3) Download the file the "config.json.example" from

<https://github.com/datagutt/bbox-receiver/blob/main/config.json.example>

4) Copy the file to your srt_receiver folder

5) Rename the file to "config.json"

This file sets your stream user and stream key.

The default values

user = belabox

key = belabox

you should change them for security

6) In windows open terminal window

drag the config.json into the terminal window. Terminal will show you the path to the file.

C:\Users\usr1\Desktop\srt_receiver\config.json

copy and keep that path for the docker run command

7) datagutt/bbox-receiver needs 4 ports to work

```
5000/udp    # video ingest
8181/tcp    # Private Statistics - leaks key
8282/udp    # video out
3000/tcp    # Public Statistics - NOLABS can can this
```

8) we want to map the ports to different numbers so they don't create conflicts on your network.

```
5077:5000
8177:8181
8277:8282
3077:3000
```

port 5077 is the external port for your docker container and internally it resolves to port 5000

9) replace "**C:\Users\usr1\Desktop\srt_receiver\config.json**" in the docker command with your path to your config.json file from step 6

Docker command to run in terminal

```
docker run -d --name belabox-receiver -p 5077:5000/udp -p 8177:8181/tcp -p 8277:8282/udp
-p 3077:3000/tcp -v C:\Users\usr1\Desktop\srt_receiver\config.json:/app/config.json
datagutt/belabox-receiver:latest
```

What is all the stuff in the docker command?

Name of the container

--name belabox-receiver

All the ports and their external ports mapped to internal ports

-p 5077:5000/udp

-p 8177:8181/tcp

-p 8277:8282/udp

-p 3077:3000/tcp

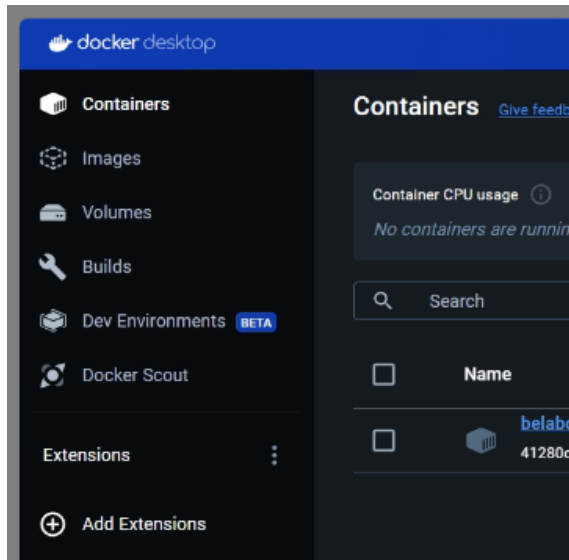
Make a volume in docker pointing the docker /app/config.json file to the one that is on your desktop

-v C:\Users\usr1\Desktop\srt_receiver\config.json:/app/config.json

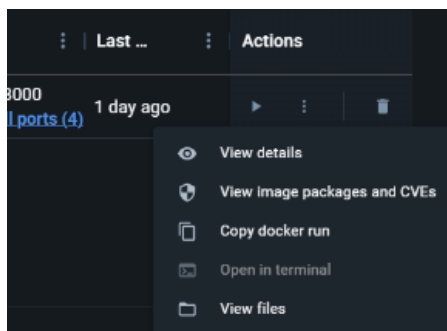
Pull the latest docker container “datagutt/belabox-receiver” from the docker repository and build and run the container.

datagutt/belabox-receiver:latest

You can now use the Docker Desktop program to start, stop and delete the container. Clicking the View detail in the



Clicking the View detail in this window will show you the log from the container. Useful when getting this working.



Looking at the log from view details if you see the “srtla_rec is now running” then the container is working

```
2024-04-19 07:53:53  
2024-04-19 07:53:53 Node.js v21.7.3  
2024-04-19 07:53:55 Trying to connect to SRT at 127.0.0.1:8282... success  
2024-04-19 07:53:55 srtla_rec is now running  
2024-04-19 07:53:56
```

Opening ports on your router.

Using my default ports from above you want to open and point the ports to your OBS computer.

In this example lets say my OBS computer ip is
192.168.1.200

These are the ports from above

5077:5000/udp	# video ingest
8177:8181/tcp	# Private Statistics - leaks key
8277:8282/udp	# video out
3077:3000/tcp	# Public Statistics - NOLABS can can this

If your your OBS and SRTLA container is running on the same network the only port you need to open is

5077 to 5077 to 192.168.1.200

If you are running the SRTLA container for a friend on your computer and their computer running OBS is somewhere else then these are the ports you want to open

5077 to 5077 udp to 192.168.1.200
8277 to 8277 udp to 192.168.1.200
3077 to 3077 tcp to 192.168.1.200

* A user with a eero router reported that the port forwarding was restricted and had to use different port numbers then my default ones.

NOALBS

It is best to watch a YouTube video for all the different settings you need to change in the NOALBS config.json file.

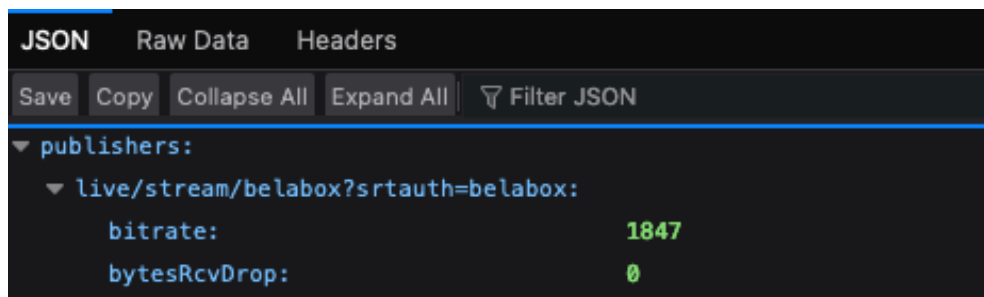
When you get to the server part of the video follow the directions below to get it working with the container.

In this example my public ip is

47.87.21.65

1) Pull up your public stats

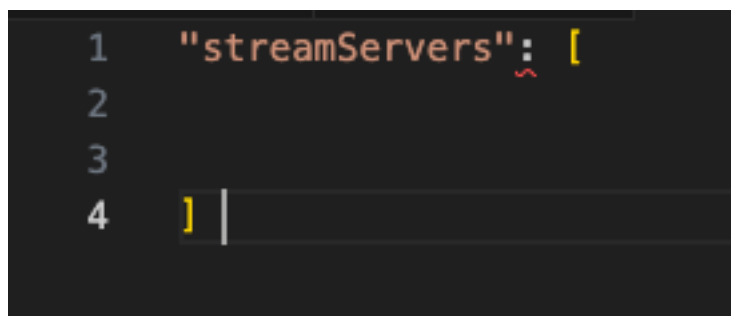
47.87.21.65:3077/stats?streamer=belabox&key=belabox



You need to copy the publisher from your stats page
"live/stream/belabox?srtauth=belabox"

2) In the NOALBS config.json delete everything between the 2 brackets in the file

```
[
]
```



3) replace with this

```
{
  "streamServer": {
    "type": "SrtLiveServer",
    "statsUrl": "http://47.87.21.65:3077/stats?streamer=belabox&key=belabox",
    "publisher": "live/stream/belabox?srtauth=belabox"
  },
  "name": "SLS2",
  "priority": 0,
  "overrideScenes": null,
  "dependsOn": null,
  "enabled": true
}
```

So it looks like the below

```
1  "streamServers": [
2    {
3      "streamServer": {
4        "type": "SrtLiveServer",
5        "statsUrl": "http://47.87.21.65:3077/stats?streamer=belabox&key=belabox",
6        "publisher": "live/stream/belabox?srtauth=belabox"
7      },
8      "name": "SLS2",
9      "priority": 0,
10     "overrideScenes": null,
11     "dependsOn": null,
12     "enabled": true
13   }
14 ]
```

4) Replace the statsUrl with your public stats url

5) Replace the publisher with what you copied from above in your stats page.