

Beszel Server and Agent

- [Install and Configure Beszel](#)

Install and Configure Beszel

Beszel is an incredibly simple, but powerful, system monitoring tool that is ideal for any homelab or small business. It offers automated monitoring updates from client to a centralized server. It shows CPU, RAM, Disk IO, Temperature, and Docker stats (if using docker). It can also provide data on other disks when configured.

Beszel boasts tools for SSO using multiple options, including OIDC, and email and other notifications as well. It's open source, self hosted, and absolutely awesome!

As with all of my write-ups. The further in time we proceed from the time of the write-up, the more likely things will change with the project. I highly recommend you check [the official documentation](#) if you have any issues with my write-up.

What You'll Need

- A server (physical, VM, or container) to run Beszel Server Hub on (Binary and Docker options are available).
- (optional, but recommended) Docker-CE and Docker Compose installed on the host.
- Clients to install the Agent on (Binary and Docker agents are available).
- (Optional) an Identity Provider which uses OAuth / OIDC
- (Optional, but recommended) A VPN through which to connect remote clients for monitoring.
- About 15 minutes of your time.

Install Docker-CE and Docker Compose

You can use this handy one line command to install Docker-CE and Docker Compose on your Linux based server system. If you are installing this on a container VM such as Incus, LXC, or LXD, then you need to make sure it's got 'nesting' enabled so that it can run docker containers without issue.

SSH to your host machine, and run the following command:

```
curl -fsSL https://get.docker.com | sh
```

Enter your super user password when prompted, and allow the install to complete. If you want to use the `docker` and `docker compose` commands without typing `sudo` every time, then add your user

to the docker group with the command:

```
sudo usermod -aG docker <your username>
```

Once added, you can use these two command:

```
newgrp
```

```
newgrp docker
```

to begin using the docker command right away, or you can exit the system and re-login in order for the new group addition to take effect on your user.

Installing the Beszel Server (Using Docker and Docker Compose)

Let's create a new folder for our docker compose file.

```
mkdir -p docker/beszel
```

Move into the new folder with

```
cd docker/beszel
```

Create a new file called "compose.yaml" with

```
nano compose.yaml
```

And paste in the following code

```
services:
  beszel:
    image: henrygd/beszel
    container_name: beszel
    restart: unless-stopped
    ports:
      - 8090:8090
    volumes:
      - ./beszel_data:/beszel_data
```

The above will create a new docker service on the host by pulling down the image from 'henrygd/beszel'. The only thing you may need / want to change is the port mapping. If port 8090 is already in use on the host server where you are adding the Beszel docker container, then you'll want to change the left side port number only. So, if for instance you wanted to change it to 8220, then your port mapping will look like:

```
- 8220:8090
```

Save the file with CTRL + O, then press Enter to confirm. Next, exit the file with CTRL + X.

Now use the command

```
docker compose pull
```

to pull down the image and make sure your yaml is read properly by the system. Once it's pulled down, use the combined command:

```
docker compose up -d && docker compose logs -f
```

to start the container, and show the log output of the container.

You should see no errors, and see a few lines showing that the server is running and available at <http://0.0.0.0:8090> .

If so, well done, you can stop the logging with CTRL + C. Now open your favorite modern browser, and navigate to the IP and port for your Beszel server. In my case I go to <http://192.168.10.28:8090>. The first time the Beszel server loads, you'll be asked to create a new account. This account will be the system administrator account using the built-in Beszel Authentication.

Make sure to check out the full video for the Settings overview, as well as the ways to access various settings on the system.

Setup SSO with Open ID Connect

In Beszel, there are multiple options for setting up various forms of SSO. I recommend again, that you refer to the official documentation for any of the options you may want. My instructions, as of writing are going to be for the Open ID Connect setup using a self hosted identity provider such as Authentik, Keycloak, Authelia, Zitadel, etc. As time goes on, these instructions may become outdated, so it is always in your best interest to check the official documentation if you have issues.

In order to access the necessary settings for the OIDC setup,

1. you need to click the settings icon (gear icon) in the upper right of the Beszel Server UI. This will take you to the General settings of Beszel.
2. On the left, navigate to 'Notifications',
3. then in the right (main) area, click the link under 'Email Notifications' that says 'configure an SMTP server'.

This will take you to the backend UI for the advanced settings of Beszel. You'll likely be asked to authenticate to this area. Your initial user you created is the one you should use for this area. As far as I can tell, this is always the user you need to use to access this area.

1. Once logged in, you'll be dropped in a new UI. On the far left, click the icon for Settings (wrench crossed with a screwdriver).
2. In the 2nd pane (1 to the right) click the 'Application' option,
3. then in the main pane, disable the setting labeled 'Hide collection create and edit options'.
4. Save the change with the 'Save Changes' button.

You should now see a few new options in the menu to the left.

1. Click on the far left icon at the top (3-stacked circles) called 'Collections'.
2. In the second pane (1 to the right) click on 'Users'.
3. At the top of the screen, next to the 'Users' title, you should see a gear icon. Click it, and you'll see a panel open (currently a slide out from the right). NOTE: If you have not disabled the 'Hide collection create and edit options', you will not see this gear icon.
4. Select the 'Options' tab in the new panel.
5. Find the OAuth2 row, and expand it by clicking.
6. Enable it, if not already enabled.
7. Select 'Add Provider'.
8. Scroll down to find Open ID Connect, and click it.
9. Fill the form with your Identify Provider's Information you generate for Beszel.
10. Click the 'Save Provider Config' button.
11. Click the 'Save Changes' button to close the slide out panel.
12. Navigate back to the 'Settings' icon on the far left, and click it.
13. Select 'Applications' in the second pane.
14. Enable the 'Hide collection create and edit options' toggle.
15. Save your changes.

Now we need to modify our docker compose file for the server to allow us to use our SSO to login.

Go back to the terminal on your server, and move to the 'docker/beszel' folder.

```
cd docker/beszel
```

Open the "compose.yaml" file to edit it.

```
nano compose.yaml
```

add the lines at the end of the file.

```
environment:  
  - USER_CREATION=true
```

Once added the full file should look something like this:

```
services:  
  beszel:  
    image: henrygd/beszel  
    container_name: beszel  
    restart: unless-stopped  
    ports:  
      - 8090:8090  
    volumes:  
      - ./beszel_data:/beszel_data  
    environment:  
      - USER_CREATION=true
```

If you don't add the `USER_CREATION=true` portion, your SSO provider won't be able to create new users in Beszel when your SSO users first try to access it.

Save the file with CTRL + O, then press Enter to confirm, and exit the nano editor with CTRL + X.

Now, we'll restart our docker server with the command:

```
docker compose up -d --force-recreate
```

This will stop and restart the container, and re-create it using our new settings.

Now you can log out of the Web UI, and re-login. Try using the new Button for your SSO provider.

Installing the Beszel Agent (Client)

Let's log into one of our client machines. While it's a bit manual at this point, I've no doubt that this could be automated with bash scripting, or Ansible, etc. It's definitely worth going through the process of adding a client manually a couple of times, so we understand the process though.

1. In your Beszel server Web User Interface. click the button in the upper right corner of the main overview page labeled 'Add System'.
2. This will bring up a form. First, choose the Docker tab for a docker-based agent to deploy, or 'Binary' for a binary agent to deploy. (When possible, I recommend the docker agent,

as it provides docker info as well).

3. Next, fill in a name for the agent. Keep it simple, yet easy to identify which machine it is in a list of machines you are monitoring.
4. Fill in the IP address of the client. If you are using your LAN address, fill in that IP (again for the client machine, not the server), and if you are using a VPN, then enter the VPN IP address. Make sure the IP address you use is routable by the server (can be reached by the server for SSH communication).
5. Leave the Port and Public Key alone, and click the 'Copy Docker Compose', or 'Copy Linux Command' button at the bottom of the pop-up window. (**DO NOT** click the 'Add System' button yet! You must install the agent before clicking this button.)
6. Copy the compose file, or command, and move to your client system.
7. If you used compose, then create a docker-compose.yml or compose.yaml file on the client and paste in the information provided. Save the file and run it with `docker compose up -d`. If you chose the binary, then open a terminal on the remote system, if not SSH'd in, and paste the command in and run it.
8. Once the command has run, or the compose file is up and running, go back to your server, and complete the process by now clicking the 'Add System' button on the Web UI. **DO NOT** click the Add System button before you have finished installing the agent on the client machine.

You should now see the new system appear on your overview page. If you do not, ensure you have provided the correct IP address, and that the IP address is accessible by the server host.

Monitor More Disks

In many cases, you'll have systems with multiple disks or pools of drives you want to monitor. When you do, I again highly recommend the docker setup, but this can be accomplished with the binary as well. I'll point you to the [official documentation](#) for the binary, but we can go through adding drives in the Docker system here.

On the docker-compose.yml or compose.yaml that you created for your agent (client machine) you'll want to add a couple of lines in order to monitor any disks / volumes.

In my case, I wanted to monitor an nvme drive I had setup with btrfs in my incus server. To find this, I did a little internet search, and found the path to be `/var/lib/incus/storage-pools`.

I then did `sudo su` to become the root user on the client machine. Next, I just listed out the files in that directory.

```
ls /var/lib/incus/storage-pools
```

I tend to keep things simple, and I named it simply 'btrfs-pool'. I added that to the path above, then appended '.beszel' to the end of it. I mapped that to the location in the docker container '/extra-filesystems/' and gave it a name, in my case 'nvme0n1p2', and set it to read-only with ':ro'. So the full mapping looked something like this.

```
- /var/lib/incus/storage-pools/btrfs-pool/.beszel:/extra-filesystems/nvme0n1p2:ro
```

Set this in the volumes section in your compose file, save, and restart the docker container with

```
docker compose up -d --force-recreate
```

Then go back to your server. Give it a little time to start collecting data, and the new drive should show up. You'd repeat this for as many drives as you want to add.

But can I run the agent on Windows?

Yes, though the Beszel development team doesn't officially support it yet. There are two ways currently that this can be done.

1. The Beszel team suggests running the Linux binary agent in WSL2 (Windows Subsystem for Linux 2). This will work, but is limited to reporting data for the VM of WSL in windows, and not specifically for Windows itself.
2. You can build the agent for Windows. I'm going to link to [an article about how to do this here](#). If you are finding this tutorial a few months or years down the road, definitely check the official project to see if they support Windows natively at this point.

Support My Channel and Content

Support my Channel and ongoing efforts through Patreon:

<https://www.patreon.com/awesomeopensource>

Buy me a Beer / Coffee:

<https://paypal.me/BrianMcGonagill>