

Executable Tutorial

renstr@kth.se alekru@kth.se

April 2021

Contents

1	Linode	2
2	Abstruse	2
3	GitHub Action vs Linode & Abstruse	2
4	Relation to DevOps	3
5	Usage of markers	3
6	Installation	3
6.1	Creating a Linode	3
6.2	What is VNC and how to get a graphical view	5
6.3	Connecting to Linode's Graphical display	7
6.4	Abstruse	8
7	Automatic monitoring	12
7.1	Notification Thresholds	12
7.2	Longview	12
8	Example usage	13
8.1	Testing mypy	13
9	Where to go now?	14

1 Linode

Linode was founded in 2003 by Christopher Aker, who graduated from Full Sail University located in Florida. Today the company offers cloud computing, block/object storage, network load balancers and DDOS protection. The goal in 2003 was and has continued to be to offer better support than it's competitors at a more straightforward pricing structure. The result is a more predictable pricing strategy where the customer is never meant to be surprised by their bill at the end of the month[5].

When Amazon Web Services (AWS) became one of their primary competitors, another challenge became apparent: to be more affordable than AWS. Today Linode is the largest independent open cloud providers in the world. Independent means that their infrastructure is not part of another company's cloud hosting services, such as AWS with amazon or Azure with Microsoft[5][4].

An important result of this independence is the knowledge for it's customers that no potential competitor such as Amazon or Microsoft could ever threaten their core cloud infrastructure. A recent case is the one of Parler getting kicked off AWS in a potentially coordinated attack. While there in this case might not be any doubt for the reason behind AWS's choice, next time it might not be as well received. This can be especially worrying considering that AWS can shut down an account for any reason without prior notice. Hosting through an independent provider such as Linode means never having to worry about being subjected to any major cloud providers whims.[2][1]

Another benefit of using Linode is the standardized price across it's data-center. Having the option of placing an edge server without having to consider any potential cost increase is another advantage when using Linode.[5]

2 Abstruse

Abstruse is a tool for Continuous Integration (CI) based on Go that uses docker to run a web GUI to administer the integrating with GitHub. Where many other CI tools potential uses is obfuscated by marketing speak, Abstruse has a clean and simple design with straightforward options. If it runs on any available docker image, Abstruse can probably test it[3]. Abstruse is lightweight enough that it can run on one of the most affordable plans from Linode, or almost any laptop or desktop available. The software can also use as many nodes or workers as a project requires. Because Abstruse is based on the MIT license and available as open source, making changes to the base code is easy.

3 GitHub Action vs Linode & Abstruse

The case can be made for using GitHub Action instead managing a dedicated server such as Linode with a CI tool such as Abstruse. The advantages of using the Linode and Abstruse combination are many but for example: no spending limit in regards to CPU time. Linode also offers much easier storage compared to Github.[6]

4 Relation to DevOps

The relation between Linode, Abstruse, and developer operations comes from how a CI tool can assist in avoiding the phenomenon known as merge hell. This is a situation where multiple developers have checked out a branch that has since been outdated due to new commits to the branch and then later try to merge the outdated versions with new features added or bugs patched. By constantly integrating the work into the branch and having a tool such as Abstruse testing those integration's, merge hell can be avoided. In this way, CI tools are invaluable to DevOps.

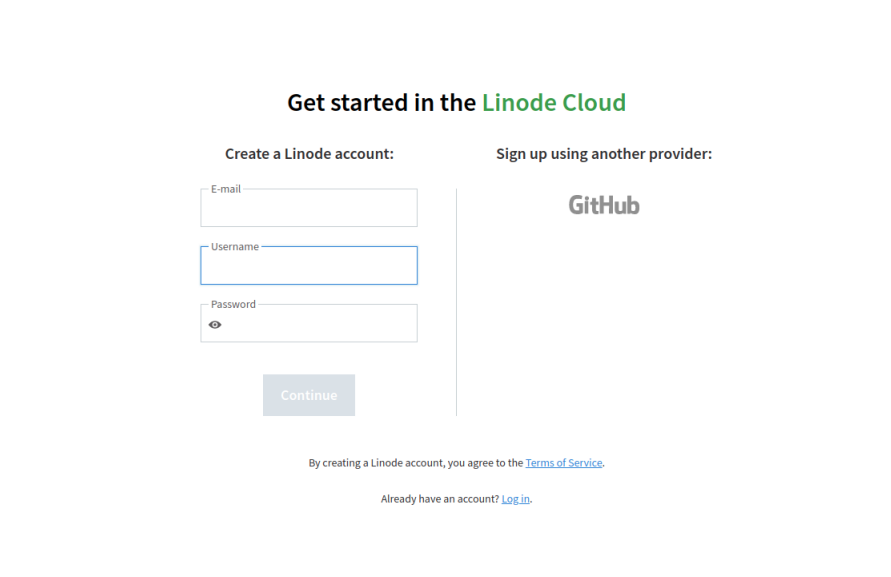
5 Usage of markers

IMPORTANT During this tutorial LW stands for Linode Website, LT for Linode terminal ergo a terminal connected to Linode's server using ssh, HT stands for home terminal ergo the one NOT connected to Linode, L-VNC stands for Linode's VNC connection by Remmina, and H-Browser stands for Browser on home computer.

IMPORTANT Every single problem early testers reported stemmed from the simple statement: "Oh, I didn't feel like reading all that"

6 Installation

6.1 Creating a Linode

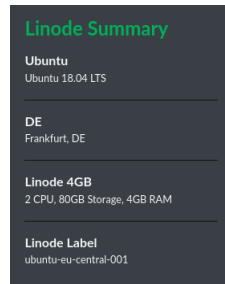


The first step in creating a Linode server and installing Abstruse is to setup an account on [linode.com](https://www.linode.com)¹. To register, a credit card is required but since Linode offers 100\$ to start with, you should be able to get Abstruse running without fronting any money. Feel free to google Linode instead of simply using the link. Entering credit card details based on a link may feel like a scam, therefore take a few minutes to check out Linode so that the risk may feel smaller. There is also

¹Linode website: <https://www.linode.com>

the option of looking at Linode's ssl certificate to verify that the right website is loaded.

Begin by creating a "Linode". An Abstruse instance is more than satisfied running on the 4GB shared CPU plan that comes with 80GB of storage. If the plan is to use the graphical view often or run something else besides a CI tool such as Abstruse, there are multiple higher tiers available. Note that automatic backup is not included, but can often be added for a small fee. Multiple OS's are available; in this tutorial Ubuntu 18.04 will be used.



Listing 1: Linode settings

LW: image = Ubuntu 18.04 LTS

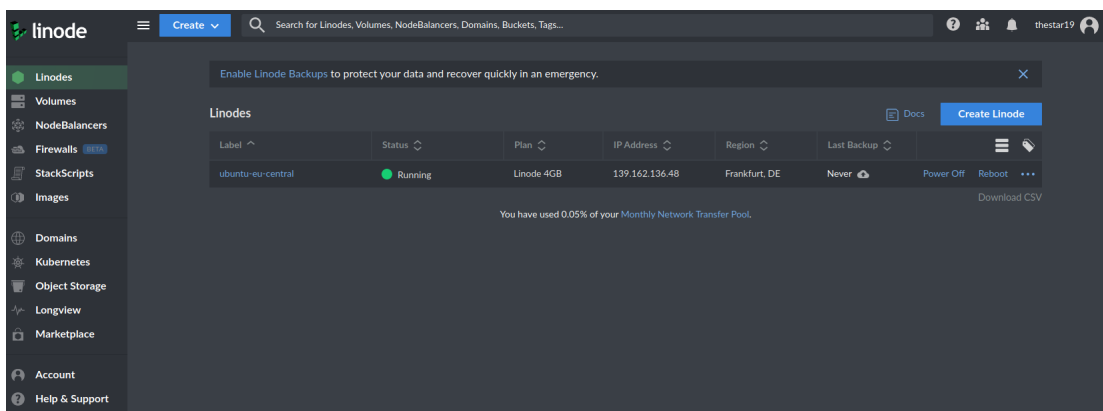
LW: Region = Frankfurt , DE

LW: Plan = Shared CPU, Linode 4GB

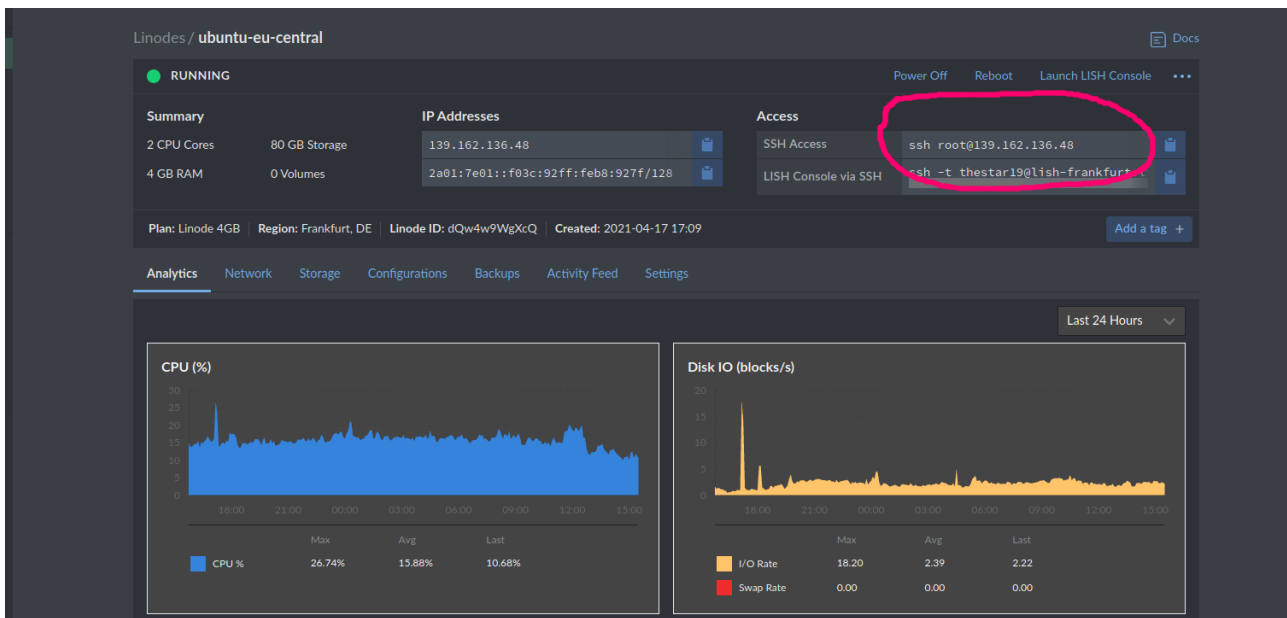
LW: SSH Keys = SSH key , usually from `~/.ssh/id_ed25519.pub`

If you have used Github ssh integration before , try:

```
xclip -selection clipboard < ~/.ssh/id_ed25519.pub
```



Remember to add any ssh keys in this step to make it easier later. If you already use GitHub on the regular and are on a Linux computer, you might just be able to run xclip. Otherwise, there are plenty of tutorials available to generate and copy a ssh key. After having ordered a Linode, the dashboard should appear where all the Linodes linked to a account are listed. By selecting a Linode, a more detailed overview appears. The server might take a few minutes to launch, but after that a ssh command should appear in the top right corner. For us this was "ssh root@139.162.136.52". The server might ask you to verify the fingerprint and log in using the password from when the node was created.



Begin by updating apt and apt-get. Sometimes, to install an application the program will ask if the installation shall continue; generally, simply type y or yes. Ubuntu may ask you more than once for every command. If the user does not enter y, the program will halt until permission is given. After having issued a command, the program is finished when the classical "root@localhost: " appears. Copy paste is unreliable due to VNC, ssh or this being a pdf file; as a result, sometimes the exact same command might not run if it has been copied and pasted. Therefore, we advise always writing each command line by line. Also, sometimes Linode does not permit using sudo, simply write all commands without it in such case.

Listing 2: Updating system

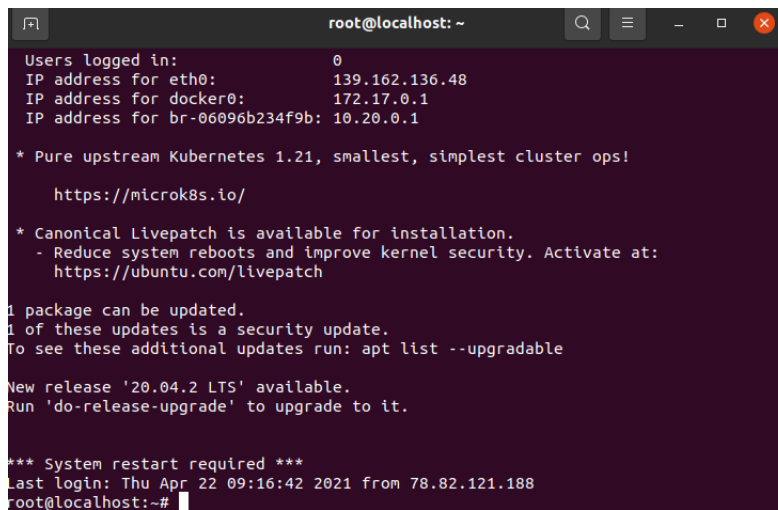
```
LT: $ sudo apt update && sudo apt upgrade
LT: $ sudo apt-get update && sudo apt-get upgrade
```

6.2 What is VNC and how to get a graphical view

The second step is to setup a VNC. A Virtual network computing (VNC), is a graphical desktop software system using the RFB-protocol to remotely control another computer over the internet. A VNC supports cross-platform usage, meaning if a VNC client uses one operative system and the server uses a different one. This makes it possible for several different clients to connect to the same server.

Once connected, it's time to get a graphical interface up and running if the goal is to use Abstruse. Linode has written a great guide that mostly works and is available as a source for more information[7].

First, the Linode needs a graphical environment to display. The server does not come with a desktop environment, therefore begin by adding one. This may take some time. Like, wondering if the server crashed long. Once done, install a VNC server such as vnc4server. To launch the server, simply type "vncserver :1".

A terminal window titled 'root@localhost: ~' with a dark background. It displays system information: 'Users logged in: 0', 'IP address for eth0: 139.162.136.48', 'IP address for docker0: 172.17.0.1', and 'IP address for br-06096b234f9b: 10.20.0.1'. It also shows messages about Kubernetes 1.21, Canonical Livepatch, and system updates, including a prompt to restart the system. The prompt at the bottom is 'root@localhost:~#'.

```
root@localhost: ~
Users logged in:      0
IP address for eth0:  139.162.136.48
IP address for docker0: 172.17.0.1
IP address for br-06096b234f9b: 10.20.0.1

* Pure upstream Kubernetes 1.21, smallest, simplest cluster ops!

https://microk8s.io/

* Canonical Livepatch is available for installation.
  - Reduce system reboots and improve kernel security. Activate at:
    https://ubuntu.com/livepatch

1 package can be updated.
1 of these updates is a security update.
To see these additional updates run: apt list --upgradable

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Thu Apr 22 09:16:42 2021 from 78.82.121.188
root@localhost:~#
```

Listing 3: Installing graphical environment and VNC-server

```
LT: $ sudo apt-get install ubuntu-desktop gnome-panel
LT: $ sudo apt-get install gnome-settings-daemon
LT: $ sudo apt-get install metacity nautilus gnome-terminal
LT: $ sudo apt-get install vnc4server
LT: $ vncserver :1
LT: $ vncserver -kill :1
```

Alternatively if `apt-get install vnc4server` does not work:

```
LT: $ snap install vnc4server
```

To get the desktop environment setup correctly, edit the file `xstartup` at “`~/vnc/xstartup`” to include the new environment. Navigate to the file using “`cd`” and “`ls`”. Once there, use “`nano`” to edit the file (`nano xstartup`), CTRL + X to exit and write to file. In this case, we add from “`gnome-panel`” to “`nautilus`”. Once this is done, it’s time to secure the VNC connection by connecting to the IP and Linodes.

Listing 4: Editing xstartup at ~/.vnc/xstartup

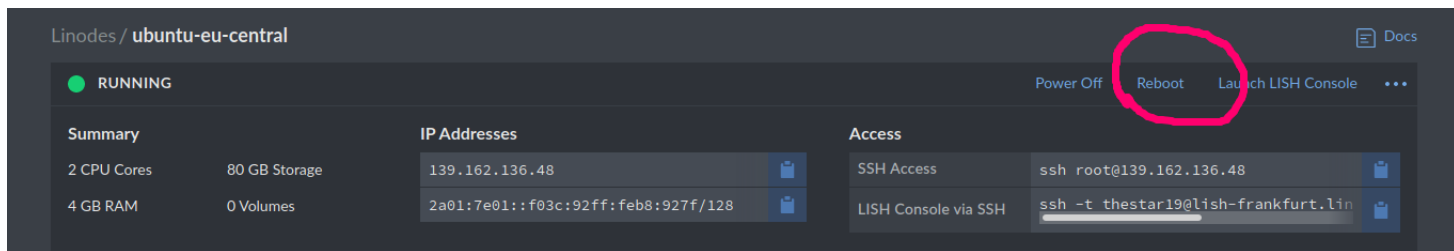
```
#!/bin/sh

# Uncomment the following two lines for normal desktop:
# unset SESSION_MANAGER
# exec /etc/X11/xinit/xinitrc

[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
xsetroot -solid grey
vncconfig -iconic &
x-terminal-emulator -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
x-window-manager &

gnome-panel &
gnome-settings-daemon &
metacity &
nautilus &
```

6.3 Connecting to Linode's Graphical display



To access the Linode server securely, the connection must be linked. To do this, reboot your Linode server from the Linode dashboard on the website. The ssh connection through the terminal should be automatically broken. After the reboot, access it again with the slightly different command compared to before that begins with `ssh -L`. Replace `root@139.162.136.48` with the ssh specific to your installation. This is same one used previously to access Linode.

Listing 5: Linking Linode and home computer

```
HT: $ ssh -L 5901:127.0.0.1:5901 INSERT_SSH_ACCESS
Example:    ssh -L 5901:127.0.0.1:5901 root@139.162.136.48
```

Replace `root@139.162.136.48` with your address available in the top right corner of Linode dashboard.

If everything has gone right so far, the Linode should now be ready to connect. Begin by starting the VNC server through the new ssh connection where the computers are linked. Linode recommends connecting using Remmina which should be available by searching for the application on your local home computer, but any VNC compatible software should work. Use the settings

we provide in Remmina, and the connection should work. If Remmina is unable to connect, either the VNC server did not start correctly or linking between systems did not work.

Listing 6: Connect via ssh and start vncserver

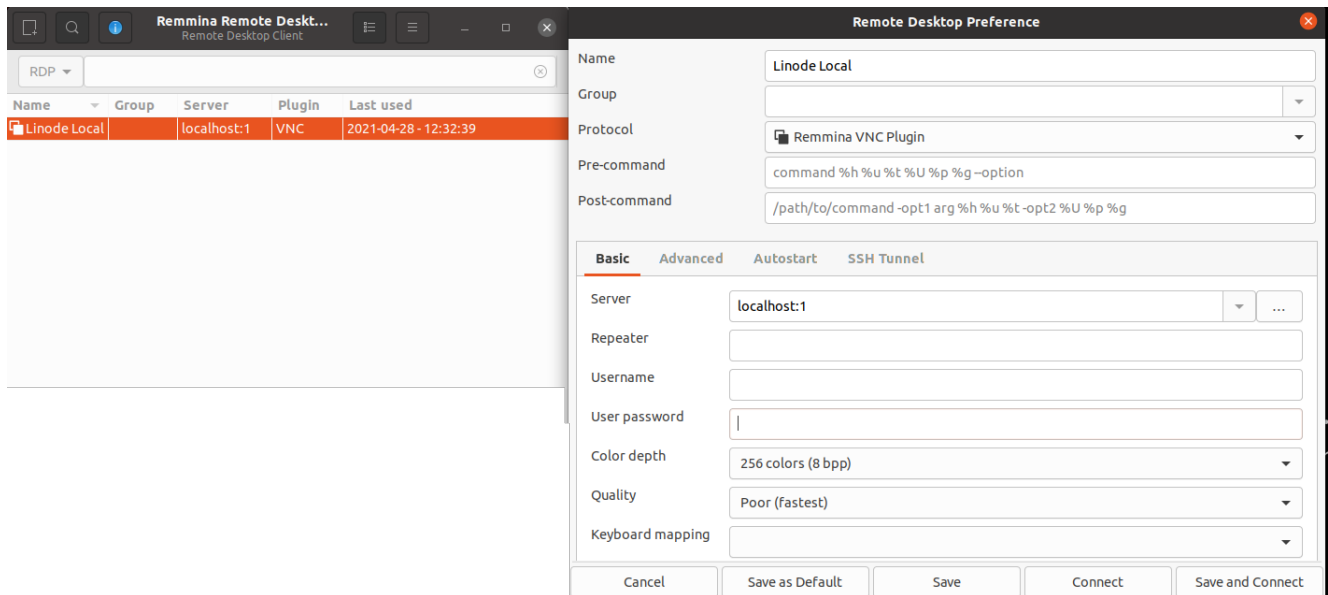
```
LT: $ vncserver :1
```

Remmina settings:

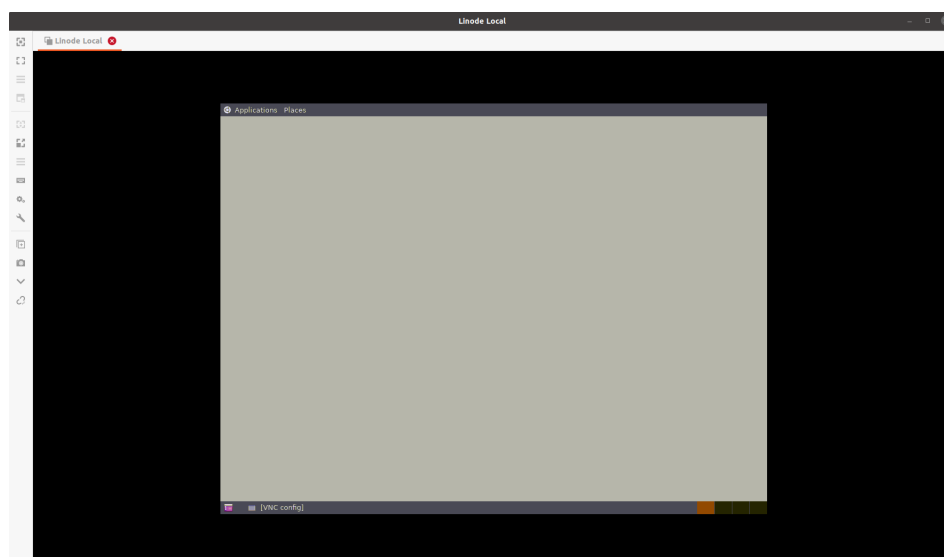
Name = Linode

Protocol = Remmina VNC Plugin

Server = localhost:1



6.4 Abstruse



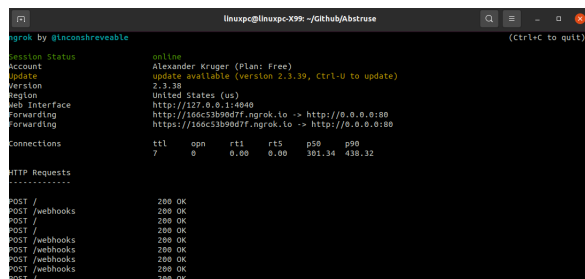
The graphical interface presented by Linode is simple, but this is an advantage. Beautiful interfaces are computationally expensive and the VNC interface won't matter when it's acting as a CI tool. The first step is installing docker and docker-compose. Once this is done, move on to cloning the Abstruse git. Abstruse needs a docker image as it's base, this can be decided later as pertaining to the particular repository, but Ubuntu is fine to start with.

Listing 7: Installing docker

```
LT: $ sudo apt-get install docker
LT: $ sudo apt-get install docker-compose
```

Listing 8: Cloning Abstruse

```
LT: $ cd ..
LT: $ cd home
LT: $ git clone https://github.com/bleenco/abstruse.git
LT: $ cd abstruse
LT: $ docker pull ubuntu
```

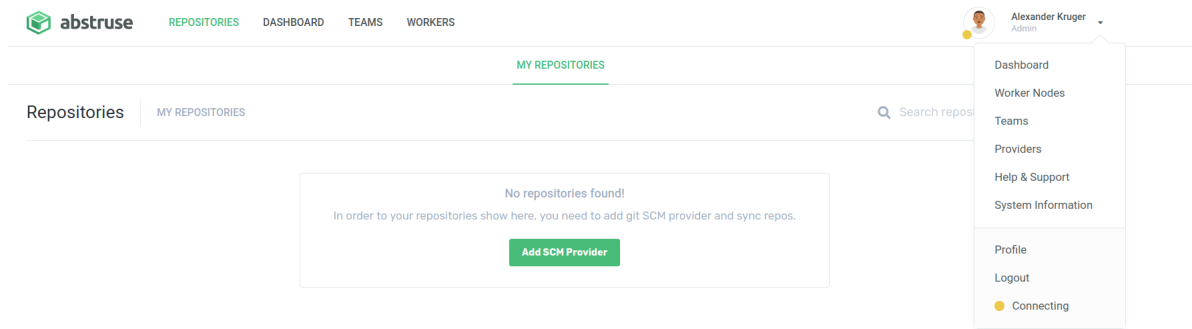


Exposing a server to the internet by opening ports can bring considerable problems, therefore this tutorial will use ngrok. Ngrok tunnels the data so that a public IP can be used without exposing more than needed. If the server is running Ubuntu, ngrok is available on snap. Without registering an account, a ngrok session is limited to two hours. Registering is free and worth doing for any server which is meant to run continually. To launch Abstruse, let docker compose the image. Afterwards, run ngrok on an available terminal window. Abstruse documentations is sparse and does hardly mention that ngrok needs to be pointed at 0.0.0.0:80 through http to work correctly without modifying Abstruse. At this stage, copy the http address presented by ngrok, which should look something like <http://e88aa3d6d4a7.ngrok.io>. You will need this later.

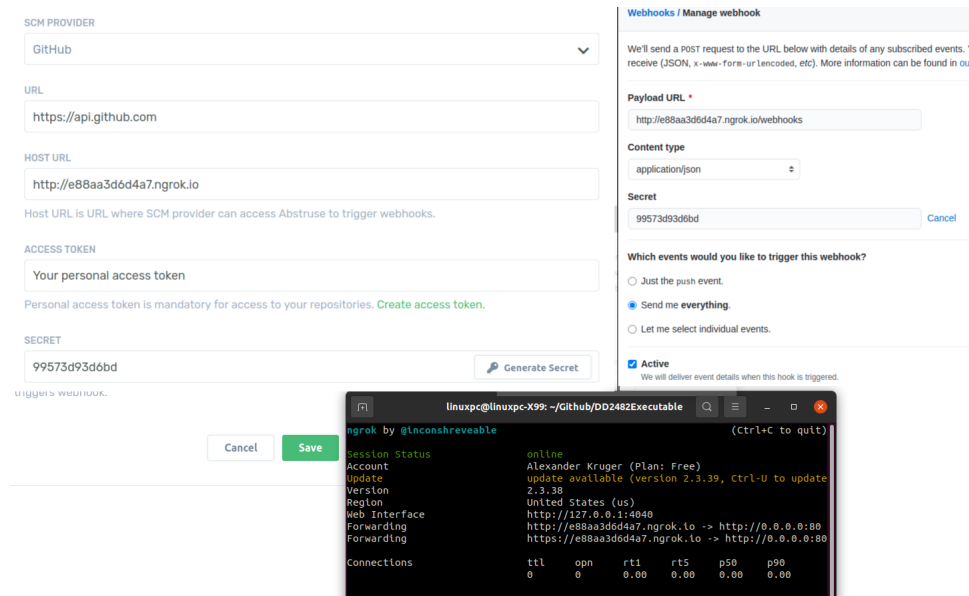
Listing 9: Docker compose and ngrok

```
LT: $ snap install ngrok
LT: $ docker-compose -f configs/demo/default/docker-compose.yml up -d
LT: $ ngrok http 0.0.0.0:80
L-VNC Browser: http://localhost
```

Once successfully launched ngrok, open a browser on the Remmina Linode VNC window by clicking in the top left corner and selecting firefox from the dropdown menu. To find the abstruse installation, go to <http://localhost>. The browser should display a modern UI. Enter the personal



details, and then head to the Provider section which can be accessed by clicking on your previously entered name in the top right corner. Begin creating a SCM provider and copy the Secret to a texteditor such as Notepad++. Switch to the host computer, go to GitHub, and create a personal access token under Settings/Developer settings. Select “repo” at the top of all the check buttons and copy the token to the texteditor. Head to the repo that Abstruse is to work with, create a Webhook under settings with the secret from Abstruse, set the address from ngrok + “/webhooks” as Payload URL, and choose “send all events”. The content type should be JSON.



Having configured the repo, continue creating a SCM provider in Abstruse by adding the ngrok URL without any “/webhooks” to the “Host URL” field. Add the access token you created previously and save the integration. Synchronize and the repo should appear. Enable it by pushing the radio button to right.

Listing 10: Settings for Abstruse and Webhook

L-VNC: SCM Provider edit on Abstruse:

```
SCM Provider = GitHub
URL = Autofills , but https://api.github.com
Host URL = Address from ngrok. Copy and paste might not work.
Access token = Noted earlier , generated at
                Git/Developer Settings/ Personal Access Tokens
Secret = Must be same as Webhook, is autofilled.
```

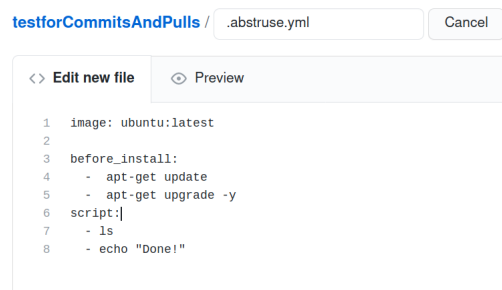
H-Browser: Webhook on Git:

```
Payload URL = Address from ngrok + /webhooks
Example: http://e88aa3d6d4a7.ngrok.io/webhooks
Content type = application/json
Secret = Secret from SCM provider from Abstruse config
Which events = Send me everything
```

Getting this step to work can be challenging at best. Sometimes the commits do not appear at all, even after having a stable record for multiple days. Ngrok does not seem to be the problem here as the log at <http://127.0.0.1:4040> often clearly shows that it is receiving and submitting web requests. The developers are aware of the problems. To fix any issues, keep an eye on Ngrok and edit the configs on both GitHub and Abstruse.

Before we can push something to GitHub, Abstruse needs a way to understand what to do with the files. For this, it uses a .yml file. Upload a .abstruse.yml file to the root directory of the repo that is being configured. Once committed, go into the Linode VNC connection to the Abstruse web GUI setup at localhost. Open the repo and fetch yml from settings. Click trigger build and a new job should appear in the repo on Abstruse.

You might need to go into the repo area in Abstruse to activate “Build pushed branches”.



Listing 11: .abstruse.yml

```
image: ubuntu:latest
```

```
before_install:
```

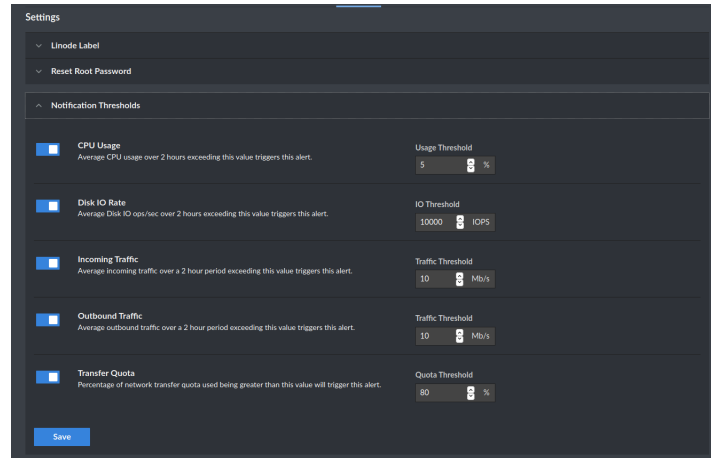
```
  - apt-get update
  - apt-get upgrade -y
```

```
script:
```

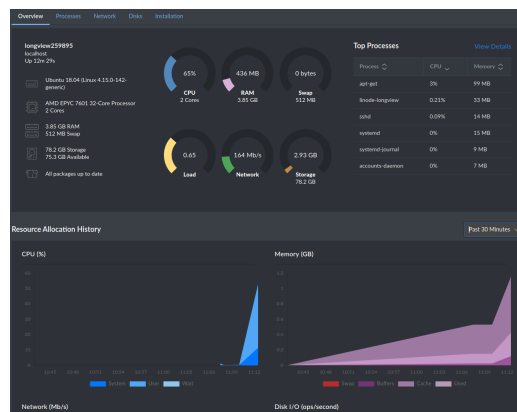
```
  - ls
  - echo "Done!"
```

7 Automatic monitoring

7.1 Notification Thresholds



Linode offers two kinds of performance monitoring that are included with the server. The most basic kind is a setting to configure warnings if for example CPU usage stays high for long. Configuring this is simple, go to the Nodes/Servers dashboard on LW and change the settings under Settings/Notification Thresholds. Note that Linode will only notify you, generally via email, if the average CPU usage stays higher for a 2 hour period.



7.2 Longview

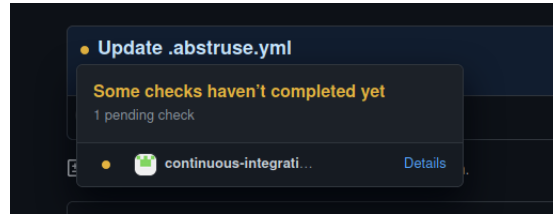
The other option is called "Longview". This gives you a good overview over the last 12 hours on the Linode webpage. Open the Linode dash and go to Longview. Run the link on your server's terminal and check back a couple of minutes later if Longview is working. Otherwise, try run these commands:

Listing 12: Longview troubleshooting

```
LT: $ sudo systemctl status longview
LT: $ sudo systemctl start longview
```

8 Example usage

8.1 Testing mypy



When Abstruse runs the job associated with the push or pull request, it begins by cloning the associated repo and then drops the .yml file into that cloned folder. From here, Abstruse hands over the reigns to the yml file and this is where many options lie.

Let's say a developer is working on some fixes for the static testing library mypy. Each time the dev commits some new code to the repo, a test should be run to see if the new code causes any problems. As it happens, mypy comes with extensive testing. The developer builds a .yml file for Abstruse and commits some code. A small notification on the commit page on github will report the result of the testing.

Listing 13: Link to mypy github repo

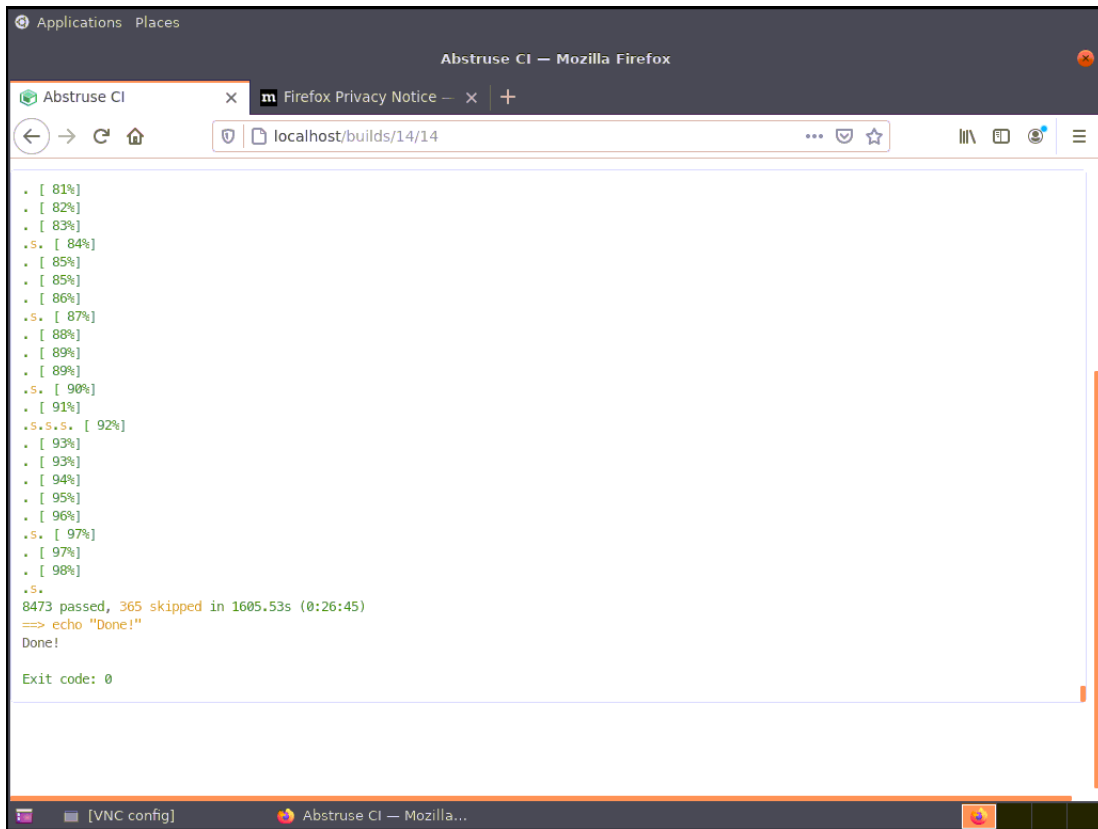
`https://github.com/python/mypy`

Download the entire mypy github repo and upload it to a new repo so that we have complete access. Add the yml file, configure Abstruse and a git repo to point to each other and push a minor change. If everything is configured correctly, Abstruse should now run the tests for mypy and report the results. It may take some time for Abstruse to finish and report the results, upwards of 27 minutes is not uncommon; Mypy has over 8300 tests.

Listing 14: .abstruse.yml for the new repo mypy

```
image: ubuntu:latest

before_install:
  - apt-get update
  - apt-get upgrade -y
  - apt update
  - apt upgrade -y
  - apt-get install python3-distutils -y
  - apt install curl -y
  - curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
  - python3 get-pip.py
  - pip3 install -r test-requirements.txt
  - apt install python2 -y
  - curl https://bootstrap.pypa.io/pip/2.7/get-pip.py --output get-pip.py
  - python2 get-pip.py
script:
  - ls
  - pytest -q mypy
  - echo "Done!"
```



9 Where to go now?

Having completed this tutorial, Abstruse should now be up and running and connected to Github. To make this a more permanent install, make sure to register for ngrok to not be limited to two hours. Right now, bugs in Abstruse prevent some options normally associated with CI tools. In the webhook message sent by Github, many details regarding the push or pull request that launched the build are unavailable. Example of this can be the commit ID. Despite this, Abstruse offers many features that can be further explored in the git repo's documentation. There are many options when configuring the .yml file; Abstruse's repo has a page dedicated to the options that Abstruse offers.

Listing 15: Links to further reading

<https://github.com/bleenco/abstruse/tree/master/docs>

<https://github.com/bleenco/abstruse/blob/master/docs/ABSTRUSE.YML.md>

References

- [1] AWS. Aws service terms. <https://aws.amazon.com/service-terms/>, April 2021. Accessed: 2021-04-29.
- [2] GeekWire. Amazon will cut off parler from aws for violating terms of service. <https://www.geekwire.com/2021/amazon-will-cut-off-parler-aws-violating-terms-service/>, January 2021. Accessed: 2021-04-29.
- [3] GitHub. Abstruse ci. <https://github.com/bleenco/abstruse>. Accessed: 2021-04-29.
- [4] Linode. Aws vs. linode. <https://www.linode.com/aws-vs-linode/>. Accessed: 2021-04-29.
- [5] Linode. The cloud developers trust. https://www.linode.com/company/about/?_gl=1*1ex5fcx*_gcl_aw*R0NMLjE2MTg0NzUyNDUuRUJYU1Rb2JDae1JdXF2WGp1c183d0lWalFXaUF4MV9vQWNyR. Accessed: 2021-04-29.
- [6] Linode. Cpu, transfer, storage, and ram bundled into one simple price. <https://www.linode.com/pricing/>. Accessed: 2021-04-29.
- [7] Linode. Install vnc on ubuntu 18.04. <https://www.linode.com/docs/applications/remote-desktop/install-vnc-on-ubuntu-18-04/>, October 2020. Accessed: 2021-04-28.