Mount Windows Share On Boot in Ubuntu 20.04

These steps assume you are root or using `sudo`.

Install cifs-utils

```
apt install install cifs-utils
```

Save your credentials in /root

Create `/root/.smbcredentials` with the following contents ...

```
username=username
password=password
```

Change username and password to your username and password.

Add fstab entry to mount on boot

Edit `/etc/fstab` and add the following line to the end.

```
//192.168.1.111/software /mnt/clyde cifs
vers=3.0,credentials=/root/.smbcredentials
```

`192.168.1.111` should be changed to your host and `software` should be changed to your share. `/mnt/clyde` is the location where the remote share is mounted. This folder must already be created.

Make sure networking is started before mounting

```
systemctl enable systemd-networkd-wait-online
```

Restart

linux:mount_windows_share_on_boot_in_ubuntu_20.04 · 2021/05/18 05:18 · 2021/05/18 11:54

Install GNS3 on UNRAID on your local network

GNS3 allows network engineers to model networks. Unlike network simulators, GNS3 uses real device images. The devices run on a hypervisor. Unraid is an easy to use NAS operating system based on Slackware that supports docker images and virtual machines.

Create a VM in Unraid

We will be using Ubuntu 20.04 as the base of GNS3. Install it as you would normally install it. After or before installation, you will need to edit the Unraid XML file of VM to allow nested VMs.

* In Unraid, go to the “VM” tab.

* Click the VM icon and then click on “Edit”.

* On the upper right of the page, click on the switch 

, to switch to XML view.

* In the `cpu mode='host-passthrough'` area, add `<feature policy='require' name='vmx'/>`

```
<cpu mode='host-passthrough' check='none' migratable='on'>
  <topology sockets='1' dies='1' cores='4' threads='1'/>
  <cache mode='passthrough'/>
  <feature policy='require' name='vmx'/>
</cpu>
```

Install GNS3

Install GNS3. All commands need to be run as root or sudo. You can install GNS3 with OpenVPN if you plan to access it from the Internet. However, we will only be accessing GNS3 through the local network.

```
cd /tmp
curl
https://raw.githubusercontent.com/GNS3/gns3-server/master/scripts/remote-ins
tall.sh > gns3-remote-install.sh
bash gns3-remote-install.sh --with-iou --with-i386-repository
```

This will take some time to finish.

After the install finishes, edit `/etc/gns3/gns3_server.conf` to change the default IP the GNS3 listens on. By default, it listens on the IP that OpenVPN would normally be on.

Edit the `host` line to the IP of your server.

```
[Server]
host = 192.168.1.13
port = 3080
images_path = /opt/gns3/images
projects_path = /opt/gns3/projects
report_errors = True

[Qemu]
enable_kvm = True
require_kvm = True
```

Restart the GNS3 service.

```
systemctl restart gns3
```

Done!

You should now have a working GNS3 installation. You will now need to add your server as a remote server for your GNS3 installation.

networking:install_gns3_on_unraid_on_your_local_network · 2021/04/29 15:23 · 2021/04/29 18:02

Python Code Snippets

List of files, just filenames, no directories.

```
filenames_no_dirs = next(os.walk(directory))[2]
```

List files, no directories, full path

```
filenames = [os.path.join(directory, fn) for fn in
next(os.walk(directory))[2]]
```

Indicate that your string variable is a raw string.

```

print(r'%s' % raw_string)
```

python:python_code_snippets · 2021/04/09 12:19 · 2021/04/11 09:21

Useful Linux CLI Commands

`crontab -u www-data -e` - Edit crontab of user where www-data is the user.

`rsync -a /dir1/ /dir2/` - Sync two directories. This overwrites files with the same names.

`stat file.txt` - Show information about a file or directory

`du -sh directory/` - show file or directory size

`zip -r zip_filename.zip directory/` - zip up a directory

`unzip zip_filename.zip` - unzip file to current directory

`zipinfo zip_filename.zip` - list files and directories in a zip file

`grep -i "whatever" file.txt` - search for "whatever" in file.txt

`!!` - run last command

`sudo !!` - run last command as root

One of the easiest tools to use if you don't know the syntax of a command is TLDR. On Ubuntu you can simply install it using apt, `apt install tldr`. Simply use `tldr zip` where `zip` is the command you wish to know more about.

linux:useful_linux_cli_commands · 2021/03/26 02:47 · 2021/04/08 21:34

Install Gogs on Ubuntu 20.04 with MariaDB

These instructions will install Gogs on an Ubuntu 20.04 server using the MariaDB back-end. Gogs is a self-hosted front-end to Git. Gogs is extremely lightweight and comes in a single binary file. It also supports multiple back-end databases including PostgreSQL, MySQL, or SQLite3.

These commands assume you are running as *root* or *sudo*. All scripts also rely on the install directory being */home/git*. Remember to set your own password wherever a *password* is needed.

Install MariaDB

```
apt install mariadb-server
mysql_secure_installation
```

Download and extract the latest/desired version of Gogs from dl.gogs.io/

```
wget https://dl.gogs.io/0.12.3/gogs_0.12.3_linux_amd64.tar.gz
tar xzvf gogs_0.12.3_linux_amd64.tar.gz
```

Create a systemd service file in */lib/systemd/system/gogs.service*

```
[Unit]
Description=Gogs
After=syslog.target
After=network.target

[Service]
LimitMEMLOCK=infinity
```

```
LimitNOFILE=4000
RestartSec=2s
Type=simple
User=git
Group=git
WorkingDirectory=/home/git
ExecStart=/home/git/gogs/gogs web
Restart=always
Environment=USER=git HOME=/home/git GOGS_WORK_DIR=/home/git

[Install]
WantedBy=multi-user.target
```

Enable Gogs service

```
systemctl enable gogs
```

Start Gogs service

```
systemctl start gogs
```

Install nginx

```
apt install nginx
```

Log in to MariaDB to create a user and database

```
mysql -u root -p
```

Create database

```
CREATE DATABASE IF NOT EXISTS gogs CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;
```

Create user and grant privileges

```
CREATE USER 'gogs'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON gogs.* TO 'gogs'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Create nginx proxy file. Note that I am using an SSL certificate here. If you do not have your own certificate, you can use certbot.

```
server {
    listen 80;
    server_name gogs.domain.edu;
    return 301 https://$server_name$request_uri;
}
```

```
server {
    listen 443 ssl;
    server_name gogs.domain.edu;

    ssl_certificate /etc/nginx/ssl/cert.pem;
    ssl_certificate_key /etc/nginx/ssl/key.key;

    location / {
        proxy_set_header X-Real-IP $remote_addr;
        proxy_pass http://127.0.0.1:3000/;
    }
}
```

Test your nginx configuration file

```
nginx -t
```

Restart nginx

```
systemctl restart nginx
```

Set the external URL by editing */home/git/custom/conf/app.ini*.

Continue the installation using the web interface https://gogs.domain.edu.

linux:install_gogs_on_ubuntu_20.04_with_mariadb · 2021/03/01 05:41 · 2021/04/10 18:34

Older entries >>