Server spec VM 2G / 15g disk 2 cores
Install the server image
https://releases.ubuntu.com/22.04/ubuntu-22.04.1-live-server-amd64.iso

Install with all updates and create an ordinary user, this user will have some sudo privileges and we do not need to grant any sudo access to the sysop user we will create later for dxspider.
use strong passwords, this is a server that will be exposed to the internet.

Install the Standard server version of ubuntu, not the minimised version
Enable SSH
No snaps need be installed

In these notes command executed as root are prefixed with #
commands executed as the user 'sysop' are prefixed with $

In terms of editors 'nano' is an option for those unfamiliar with 'vim' ;-)
(and if 'unfamiliar with 'vim' here is how to get out !)
https://thenewstack.io/how-do-you-exit-vim-a-newbie-question-turned-tech-meme

log in as the user you created above and become root

$ sudo su -

Create user sysop
```
# su - sysop
```

Add your normal login user to the sysop group, this allows the normal login user to interact with the cluster via the 'dx' command etc
Note however the sysop user must be used for administration involving file system access such as the creation of new connection scripts, git updates etc.

```
# usermod -a -G sysop
```

Install packages that will be required for dxspider, we are going to go directly to mojo branch. all the packages required are in the apt packaging system so we shall install from there

apt install make libtimedate-perl libnet-telnet-perl libdigest-sha-perl libdata-structure-util-perl libmojo-ioloop-readwriteprocess-perl libjson-perl gcc libmath-round-perl libnet-cidr-lite-perl libcurses-perl

we will make some links now, whilst we are still root

ln -s /home/sysop/spider /spider
ln -s /spider/perl/console.pl /usr/local/bin/dx
ln -s /spider/perl/*dbg /usr/local/bin

```
Now would also be a very good time to consider one's firewall
requirements, for example

# ufw allow 7300/tcp
# ufw allow ssh
# ufw enable
```

We have done everything we need to do for now as the root user, for the rest of the install we will be the user "sysop"

change to the sysop user
root@ei7mre-2:~# su - sysop

We want to install dxspider, Mojo branch

```
$ git clone -b mojo  git://scm.dxcluster.org/scm/spider
```

change into the newly created spider directory

$ cd spider

Make our local directories

```
$ mkdir local_cmd
$ mkdir local_data
$ mkdir local
```

$ cp `perl/DXVars.pm.issue local/DXVars.pm`

edit local/DXVars.pm to define the various parameters that are appropriate for your cluster

```
$ cp perl/Listeners.pm local/
```

edit local/`Listeners.pm` and follow instructions, generally if you do have a static ipv6 address it's worth configuring for dual stack these days, however before you will see ipv6 users you will also need to ensure that you have the appropriate AAAA DNS record in place for the v6 address of your dxcluster

Create a connect script to connect to your peer node(s)

$ cd /spider/connect/
$ cp gb7tlh <peer-node>

edit this file with your chosen editor so that the telnet address and port are correct for your peer node(s) and set your own node callsign

Change to the base spider directory

$ cd /spider

run the cluster

```
$ perl/cluster.pl
```

assuming there are no errors, it will start up

Now it's time to connect to the node and check that it's working ok.
open a second ssh session to your server and become sysop

```
$ sudo su - sysop
```

issue the 'dx' command to access the dxcluster administration console

We need to tell the node the callsign (including any SSID) of it's peer node(s), this has to be done both ends so you will need to have agreement in place with other node(s) you wish to peer with

set/spider <peer-node>

now it's time to connect and see if things are working

connect/<peer-node>

if all is good you should see spots start arriving
at thai point you may wish to confirm remote telnet access is working also

Time to tidy up the installation and set up dxspider to start as a service, exit the console by issuing the 'bye' command

Firsty, as the user sysop we will set up the dxpsider crontab file. this controls things that dxpsider should do on a regular schedule, like check that it is connected to upstream nodes, update databases etc

$ cd /spider/local_cmd/
now use an editor to create a file called crontab
$ vim crontab

with the contents below (replacing <peer-node> with the call of the upstream peer node. this will set up a simple schedule that will check very minute to see if the node is connected to it's peer, if not connection will be attempted

* * * * * start_connect('<peer-node>') unless connected('<peer-node>')

further help on crontab scheduling can be found here
https://crontab.guru/

Now we will install the systemd service file for the dxpsider service and enable it.

We must become root, however since the sysop user does not have sudo privileges, we will first have to drop back to our login user

```
sysop@ei7mre-2:~$ exit
bminish@ei7mre-2:~$ sudo su -
[sudo] password for bminish:
root@ei7mre-2:~#
```

now we will copy the service file to /etc/systemd/system/

```
# cp /spider/dxspider.service /etc/systemd/system/
```
and start the service
```
# systemctl start dxspider.service
```

become the sysop user
```
#  su - sysop
```

verify the cluster is running by opening the admin console
```
$ dx
```