

LinuxONE: Get Access

Get Connected on the LinuxONE Community Cloud

 15 steps  90 minutes

THE CHALLENGE

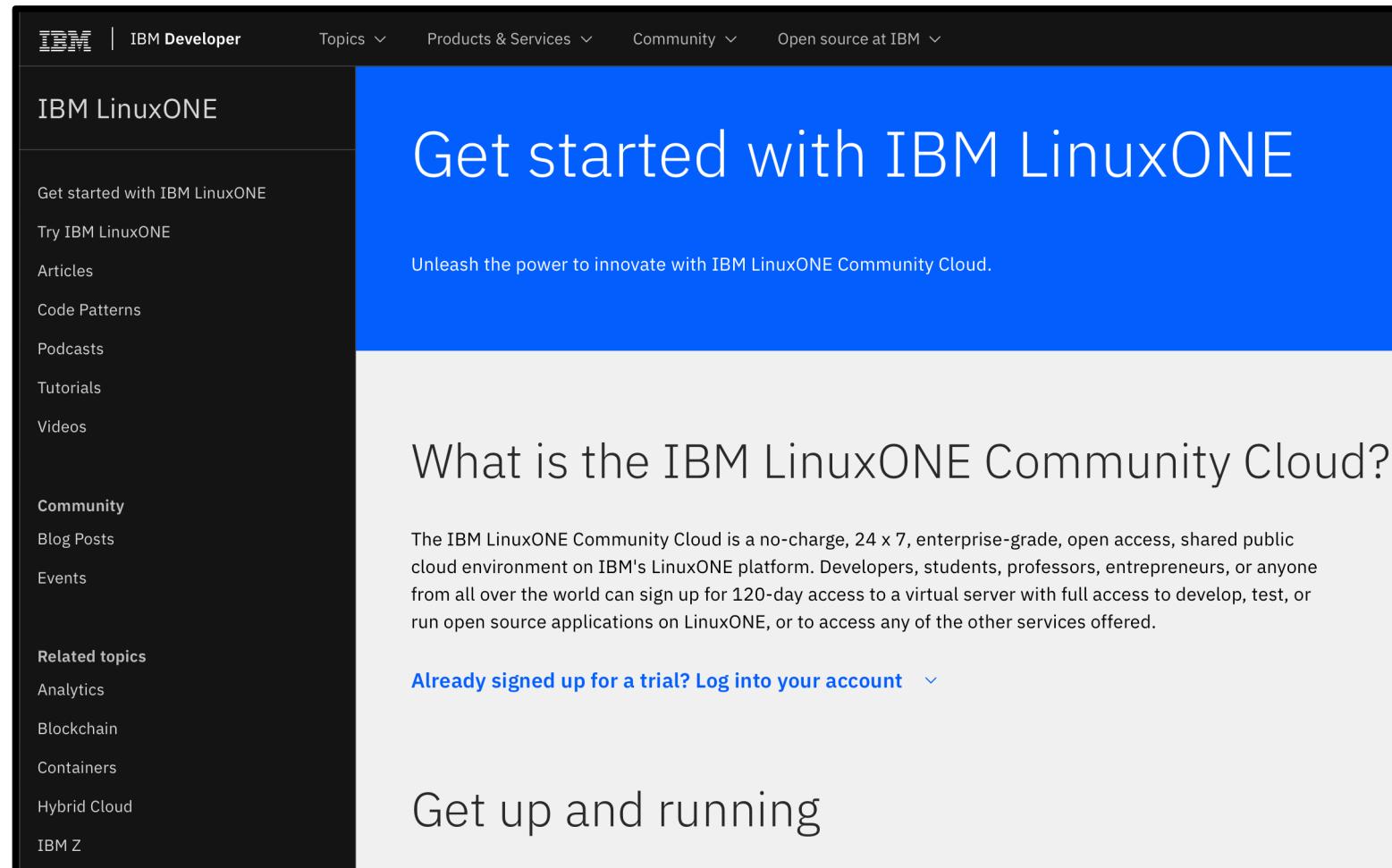
Whether you're interested in Linux but don't know where to start, a skilled shell scripter looking for something different, or just looking for a home for your Part 3 challenge, LinuxONE is here for you, all through the **LinuxONE Community Cloud**.

You'll take control of your own Linux system running on IBM Z hardware, giving you the ability to install packages, compile software, and even host a web page. Enjoy your stay!

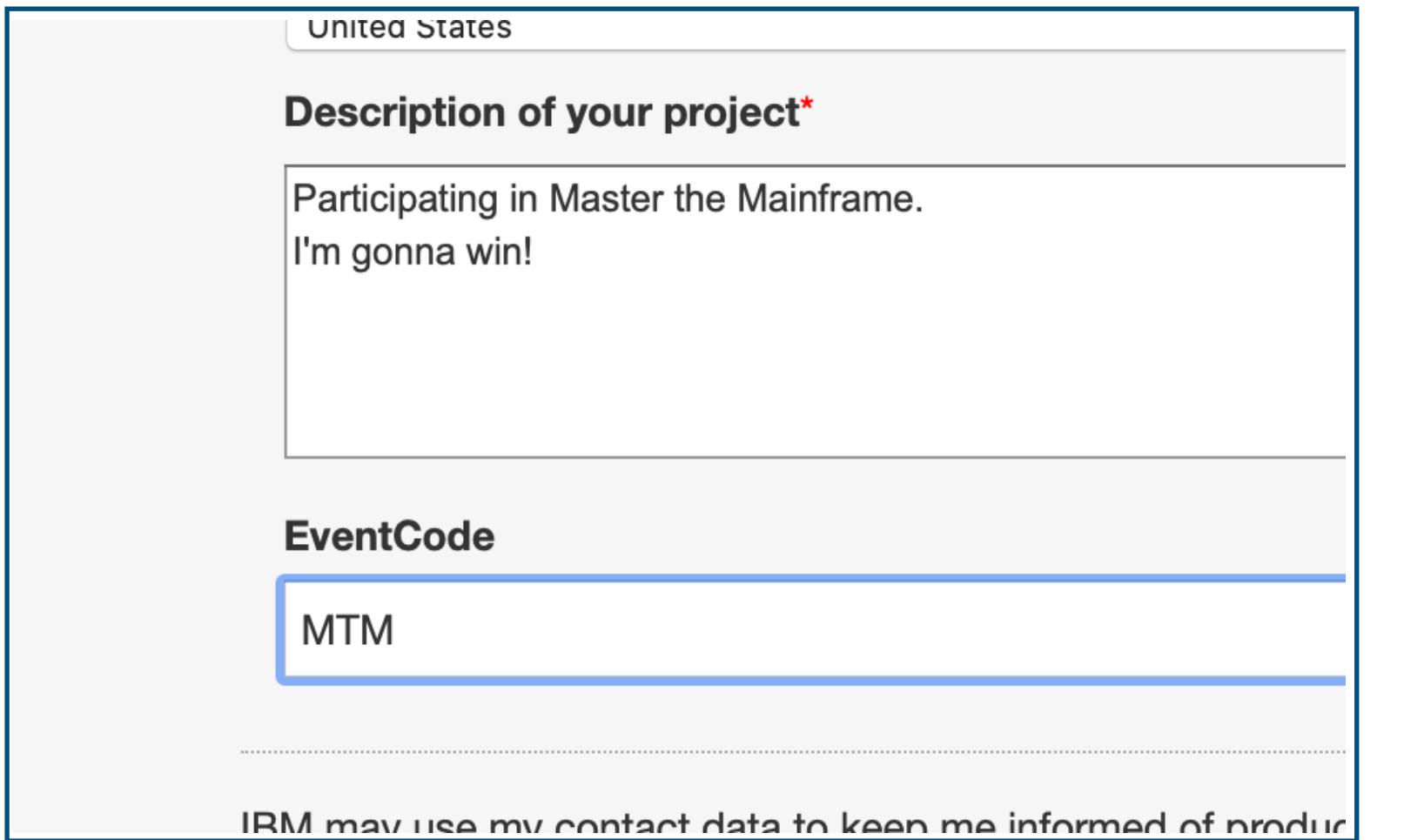
BEFORE YOU BEGIN

You will benefit from completing the USS challenges before jumping into a Linux environment, though it is not required.

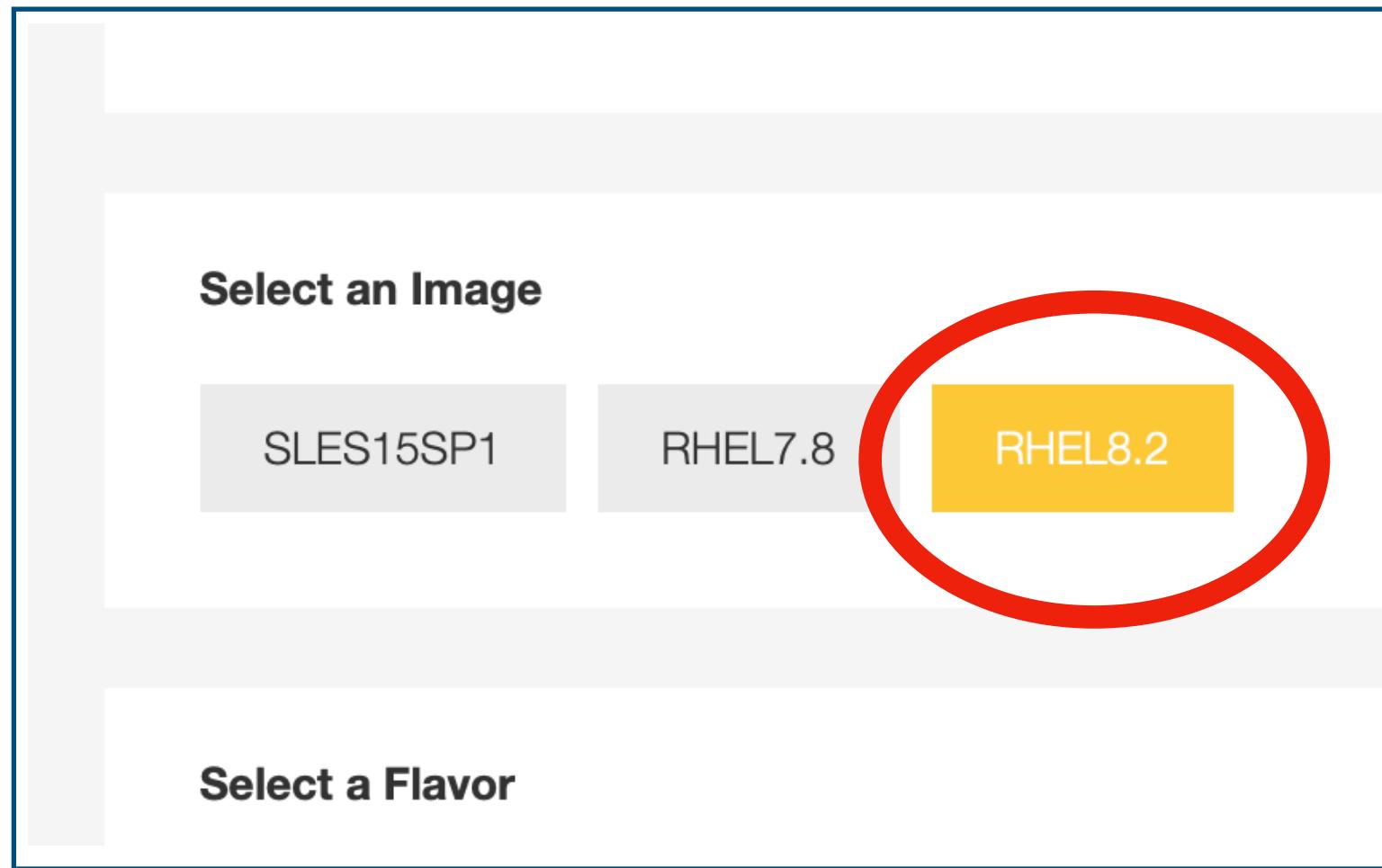
All you need is a computer and an SSH client (The one in VS Code will work) to connect.



The screenshot shows the IBM Developer website with the 'IBM LinuxONE' section highlighted. The main heading is 'Get started with IBM LinuxONE'. Below it is a sub-section titled 'What is the IBM LinuxONE Community Cloud?' which contains a brief description of the service. At the bottom, there's a 'Get up and running' button.



The screenshot shows a form step titled 'Description of your project*' with the value 'Participating in Master the Mainframe.
I'm gonna win!'. Below this is another field labeled 'EventCode' with the value 'MTM'.



The screenshot shows a 'Select an Image' section with three options: 'SLES15SP1', 'RHEL7.8', and 'RHEL8.2'. The 'RHEL8.2' option is highlighted with a red circle. Below this is a 'Select a Flavor' section with the same three options.

1. GET YOUR VIRTUAL MACHINE

Head over to the [LinuxONE developer page](#) and do a little reading about the IBM LinuxONE system and Community Cloud. Then follow [these instructions](#) to request your own virtual system.

<https://github.com/linuxone-community-cloud/technical-resources/blob/master/faststart/deploy-virtual-server.md>

You will follow the instructions in this guide pretty closely, but please read ahead through steps 2-4 here, as they contain information specific to Master the Mainframe.

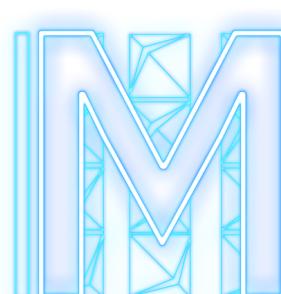
2. SET THE EVENT CODE

To help us keep track of Master the Mainframe contestants on the system, and to ensure systems remain available throughout the contest, make sure you enter **MTM** for your event code.

3. SELECT RHEL82

There are a number of Linux Distributions available for Z hardware, including SUSE Linux Enterprise Server (SLES) and Redhat Enterprise Linux (RHEL). For these challenges, we'll be using RHEL8.2, so make sure you select that here.

Afterwards, if you want to try out another flavor, just go back and de-provision your RHEL server and request a SLES instance.



```
mtm >chmod 600 linux-mtm.pem
mtm >ssh -i linux-mtm.pem linux1@148.100.113.117
Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

Last login: Mon Jul 27 21:17:23 2020 from 67.251.217.36
[linux1@ibm-jeff-bisti ~]$
```

4. SSH KEY MANAGEMENT

Good news, you may be able to use the terminal built right into VS Code to ssh into your LinuxONE instance. Once the machine has started and you've downloaded your .pem file (keep that in a safe place) chmod it to 600 so no one else can see it.

If you can't, then please follow the instructions and use PuTTY

Then use an ssh command similar to the one above to log in. You'll use `-i` to point it to your pem file. It will ask you if you want to continue (Yes), and then you're in! This step can be tricky, so take your time.

“WAIT, THIS IS MY OWN SYSTEM? HOW DOES THAT WORK?

One of the many things IBM Z is good at is virtualization. Basically, taking a whole lot of physical hardware and turning it into virtual resources. In this case, the system you're using is a virtual system running on top of what's called a Hypervisor. That's the thing that manages the translation between real, physical devices and logical or virtual resources.

Of course, unless you really go poking around, this appears just like a regular server, and that's the point. IBM Z hardware is able to make the most of resources. In fact, there are probably hundreds, if not THOUSANDS of other people just like you (well, not JUST like you) with their own *very own* system operating on the same hardware right now.

```
MTM> cd ~
MTM> mkdir my_directory
MTM> cd my_directory/
MTM> touch a_file
MTM> touch another_file
MTM> ls
a_file  another_file
MTM> rm a_file
MTM>
```

```
MTM> cd /
MTM> ls -l
total 16
lrwxrwxrwx.  1 root root  7 Aug 12 2018 bin -> usr/bin
dr-xr-xr-x.  4 root root 4096 Jul 27 16:34 boot
drwxr-xr-x. 13 root root 2420 Aug  5 21:50 dev
drwxr-xr-x. 89 root root 8192 Jul 27 16:48 etc
drwxr-xr-x.  3 root root  20 Jul 27 16:33 home
lrwxrwxrwx.  1 root root  7 Aug 12 2018 lib -> usr/lib
lrwxrwxrwx.  1 root root  9 Aug 12 2018 lib64 -> usr/lib64
drwxr-xr-x.  2 root root  6 Aug 12 2018 media
drwxr-xr-x.  2 root root  6 Aug 12 2018 mnt
drwxr-xr-x.  2 root root  6 Aug 12 2018 opt
dr-xr-xr-x. 100 root root  0 Aug  5 21:50 proc
dr-xr-x---.  3 root root 149 Jul 27 16:33 root
drwxr-xr-x. 27 root root 800 Aug  5 21:50 run
lrwxrwxrwx.  1 root root  8 Aug 12 2018 sbin -> usr/sbin
drwxr-xr-x.  2 root root  6 Aug 12 2018 srv
dr-xr-xr-x. 13 root root  0 Aug  5 21:50 sys
drwxrwxrwt.  3 root root  85 Aug 14 09:30 tmp
drwxr-xr-x. 12 root root 144 Oct 29 2019 usr
drwxr-xr-x. 20 root root 278 Jul 27 16:32 var
```

5. EXPLORE!!

This is *your* system, take a look around. Most of the commands you learned in the USS challenges are here, so commands like ls, cd, mv, mkdir and touch should work just like they did under USS, because Linux is a UNIX-like operating system.

6. PUT YOUR ROOT DOWN

Since you have full access, why not start by going into the top-level directory, also known as the root directory. Do that by entering `cd /`

Follow that up with an `ls -l` to look around. Curious what all those new directories mean?

bin - where programs (binaries) are kept

etc - System-wide configuration files

home - Where users keep their own files, including you!

lib - Short for “library”. Shared code and kernel libraries

root - Where the system administrator (root) keeps their files

usr - Shared, read-only files, which includes some binaries and documentation.

var - Short for “variable” files. Items in here change frequently, including logs, printer spools, and some caches.



```
MTM> ls /root
ls: cannot open directory '/root': Permission denied
MTM>
```

7. PERMISSION DENIED

You are currently logged into the system as user 'linux1'. This is a standard user that does not have the authority to do much damage. This is by design, to keep users from inadvertently messing up the system for everybody else.

Try to do an ls of **/root**. As a non-root user, you will get a Permission denied error because you're not allowed to look around in there.

“OH NO I BROKE IT. CAN I GET ANOTHER?

Of course! Head back to the the Linux Community Cloud page, delete the virtual machine that's currently running, and then make a new one. You can use the same keys, no need to set that up again.

Some tips to keep your system from going belly-up: Stay in the linux1 user unless you absolutely need to perform an action as root. That's when the sudo command gets used, and ideally you should double, or even triple-check your commands before hitting Enter. Don't be afraid of ruining a mainframe by entering commands, everything is kept safely isolated through the hypervisor and virtualization. (Although you should always follow the usage guidelines and restrictions set forth by the Linux ONE Community Cloud agreement. Nothing malicious or too network-intensive, please.)

```
MTM> ls /root
ls: cannot open directory '/root': Permission denied
MTM> sudo ls /root
anaconda-ks.cfg  original-ks.cfg
MTM>
```

8. ELEVATE YOUR AUTHORITY

Try the same thing, but with sudo at the beginning. This is shorthand for **S**witch **U**sers and **D**O this. By default, you become the root user (if you have authority) which gives you full privileges.

Enter **sudo ls /root** and you should see something different this time.

```
MTM> sudo yum install nano
rhel8 base          591 kB/s | 4.1 kB   00:00
rhel8 supp          3.7 MB/s | 3.8 kB   00:00
rhel8 appstream     4.4 MB/s | 4.5 kB   00:00
rhel8 codeready     4.4 MB/s | 4.5 kB   00:00
rhel8 ansible2.9    3.9 MB/s | 4.0 kB   00:00
Dependencies resolved.
=====
Package           Architecture Version      Repository  Size
=====
Installing:
nano              s390x      2.9.8-1.el8  rhel8-base  581 k
Transaction Summary
=====
Install 1 Package

Total download size: 581 k
Installed size: 2.3 M
Is this ok [y/N]: y
Downloading Packages:
nano-2.9.8-1.el8.s390x.rpm          18 MB/s | 581 kB   00:00
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : nano-2.9.8-1.el8.s390x 1/1
Running scriptlet: nano-2.9.8-1.el8.s390x 1/1
Verifying  : nano-2.9.8-1.el8.s390x 1/1
Installed:
nano-2.9.8-1.el8.s390x
Complete!
```

9. INSTALL SOFTWARE

One of the reasons you might want to use root authority briefly is to install software. Type **sudo yum install nano**

When it asks if you want to Continue, enter **y**

Yum is a software package manager for Redhat Linux. We need to first say sudo to issue a privileged command, then we are simply telling yum to install the editor called nano.

While you're there, you can also issue **yum search nano** to find out about that package, or any other software you may be interested in. Try **yum search text** to see a list of installable packages that mention "text" in their name or description.

For the curious, yum stands for Yellowdog Updater, Modified. Yellowdog Linux was a Redhat-based Linux distribution popular with older PowerPC-equipped Macintoshes.



```
GNU nano 2.9.4
my_file
I'm typing on this line
Now I'm typing on this line
Skipped that last line
And now I'm done! M

^G Get Help ^Q Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-
^X Exit ^R Read File ^V Replace ^U Uncut Text ^T To Spell ^_ Go To Line M-
```

10. EDIT A FILE

Change directory to /home/linux1 (`cd /home/linux1`) and type **nano my-file**. You are now using the software you just installed! Type some stuff (keep it clean, the admins are watching) and then type Control-O to save the file (hit enter to accept the name) and then Control-X to exit out. Nano is one of the simpler text editors available on Linux

```
tail -f /var/log/messages
```

11. TAIL A FILE

Sometimes you want to see a file that's under construction. Sound strange? What about keeping track of all of your system's messages in realtime, as they get added to the log file. We can find that in the /var directory, since that's where variable files like logs and spools live.

Try out this command:
`tail -f /var/log/messages`

```
systemd[1]: Starting dnf makecache...
dnf[7388]: Metadata cache refreshed recently.
systemd[1]: Started dnf makecache.
systemd[1]: Starting dnf makecache...
dnf[7417]: Metadata cache refreshed recently.
systemd[1]: Started dnf makecache.
systemd[1]: Starting dnf makecache...
dnf[7444]: rhe18 base
dnf[7444]: rhe18 supp
dnf[7444]: rhe18 appstream
dnf[7444]: rhe18 codeready
dnf[7444]: rhe18 ansible2.9
dnf[7444]: Metadata cache created.
systemd[1]: Started dnf makecache.
systemd[1]: Starting dnf makecache...
dnf[7475]: Metadata cache refreshed recently.
systemd[1]: Started dnf makecache.
systemd[1]: Starting dnf makecache...
dnf[7502]: Metadata cache refreshed recently.
systemd[1]: Started dnf makecache.
systemd[1]: Started /run/user/1000 mount wrapper.
systemd[1]: Created slice User Slice of UID 1000.
systemd[1]: Starting User Manager for UID 1000...
systemd-logind[6951]: New session 3 of user linux1.
systemd[1]: Started Session 3 of user linux1.
systemd[7513]: Reached target Timers.
systemd[7513]: Reached target Paths.
systemd[7513]: Starting D-Bus User Message Bus Socket.
systemd[7513]: Listening on D-Bus User Message Bus Socket.
```

12. FIX, OBSERVE AND EXIT

Did you get a Permission Denied message? You know how to fix that, right?

Tail shows us the bottom part (get it) of a file, and the -f flag says to “follow” it, showing any updates as they’re added. If anything happens on your system that it thinks is important, it will show up here. Control-c cancels the command.

No validate steps here, just enjoy and keep on going!

NICE JOB! LET'S RECAP

You may have been asking yourself “I thought this was a mainframe contest... why did I log into another system?”

As you may know, an IBM Z mainframe is able to run many different operating systems at the same time. One of those operating systems is z/OS (which you’ve been using for the rest of the contest) and it also happens to excel at running Linux. In most big companies, there is a mix of z/OS, Linux, and other operating systems being used. This is Linux on IBM Z hardware, so you’re still mastering the mainframe.

NEXT UP...

Keep it moving with the next Linux challenge, LNX2. This will take what you’ve learned and elevate them to the next level.



Want to talk? Join our Slack
ibm.biz/mtm_slack

Tweet about it!
[#MasterTheMainframe](#)