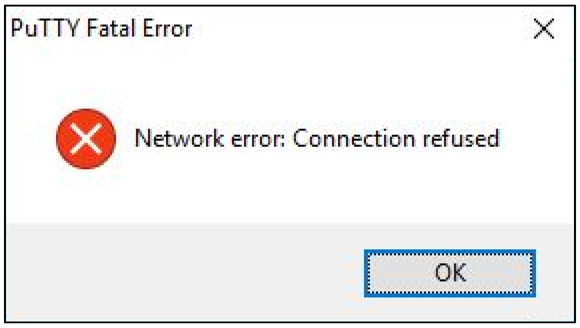
Troubleshooting - Network error: Connection refused

Objective :- Fixing the OpenSSH server daemon issue when failed to start with System start/reboot

SSH connection refused error means that the request to connect to the server is routed to the SSH host, but the host doesn't accept that request and send an acknowledgement also it can mean the SSH daemon is not running.



Network error: Connection refused

Checked the authorized ppk file added to your PuTTY connection or not ?
Checked the Security Groups of the machine are allowing SSH port 22 or not ?
Checked the Subnet attached to Internet Gateway or not ?
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html#TroubleshootingInstancesConnecting

Got the issue after checking the System Log from Instances Actions

We were able to see that the reason was we cannot connect via SSH was because the sshd service failed to run

In order to troubleshoot, launched a new t2.micro Amazon Linux 2 instance(Recovery Instance) in the same subnet with a Public IP.
https://aws.amazon.com/getting-started/tutorials/launch-a-virtual-machine/

Stopped the original instance and detached the root volume and attached it to the recovery instance and mapped as /dev/sdf

Once the volume is attached to the recovery instance, createed a mount point and mounted the volume to the recovery instance as below

> *sudo -i* (change to root)
> *mkdir /mnt/recovery* (create mountpoint)
> *mount /dev/xvdf1 /mnt/recovery* (mount the volume to the mountpoint)

Next, we checked the 'messages' file for any errors of sshd and found that the reason sshd failed to run was due to permissions being 'world-writable'... we checked the logs with the command as below

> *cat messages | grep sshd*

As we discussed, permissions in Linux are something very finicky which shouldn't be changed unless absolutely necessary. For example, when sshd tried to run it was expecting a certain set of permissions. From the error we see they are 'world-writable' (too open), which prevents sshd from running as the permissions set do not match the permissions it expects to be there.

To move forward, I proposed to copy the permissions from the Recovery Instance (as they are correct) and overwrite the existing ones in the broken volume.

> *cd /mnt/recovery* (to switch to the mountpoint)
> *find . -exec sh -c "chmod --reference=/{} {} ; chown --reference=/{} {}" \;*

Note: When permissions are broken in this manner it is nearly impossible to ever fix. So even if we regained SSH to our instance it doesn't mean that our other applications would be unaffected.

In these situations we highly recommend launching a new instance and migrating your web files to the new instance.

After the script finished, we unmounted the volume

> *cd* (to get out of the broken volume)
> *umount /mnt/recovery* (to unmount)

Now we go back to the AWS Console, detached and then reattached volume to the original instance as /dev/xvda (as this is the root volume) and started the original instance.

AWS highly recommend setting up regular snapshots of our instances which would allow us to revert any unintended changes to our system. We can automate this using the lifecycle manager which will take manage the creation, retention and deletion of snapshots.

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/snapshot-lifecycle.html

Lastly, here are the instructions for the SSM agent but remember, as in this scenario this is not something we could have done by SSM alone.

https://docs.aws.amazon.com/systems-manager/latest/userguide/session-manager-getting-started.html