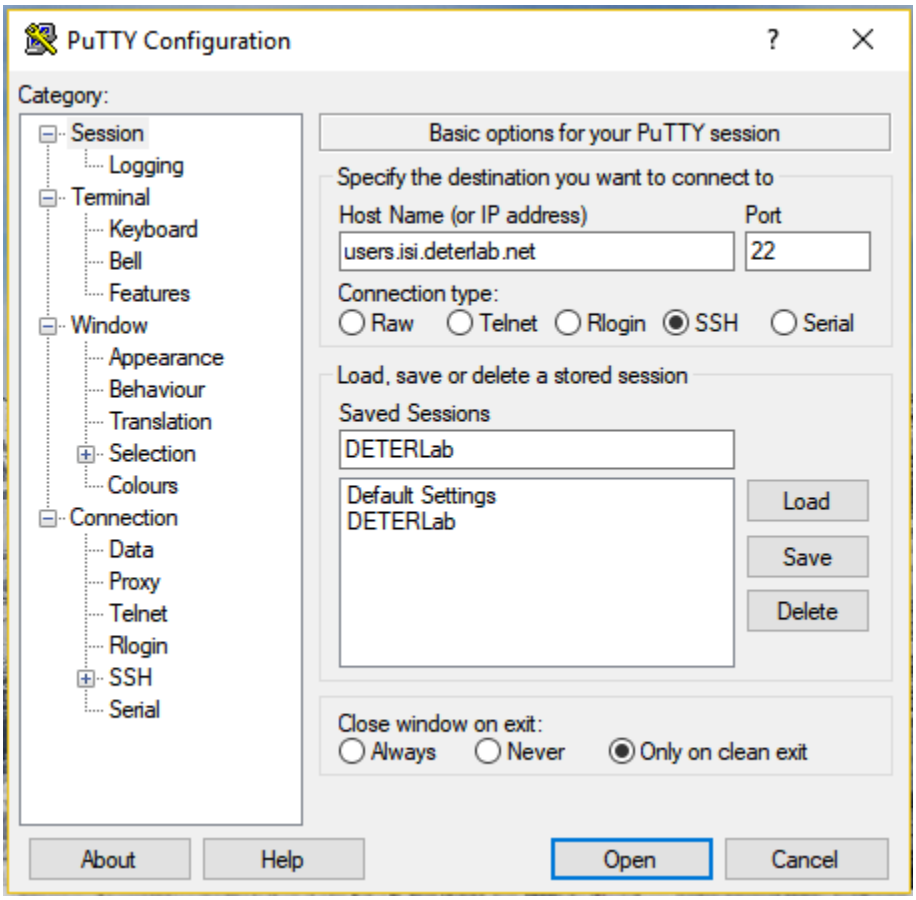
How to Access Windows Nodes using RDP

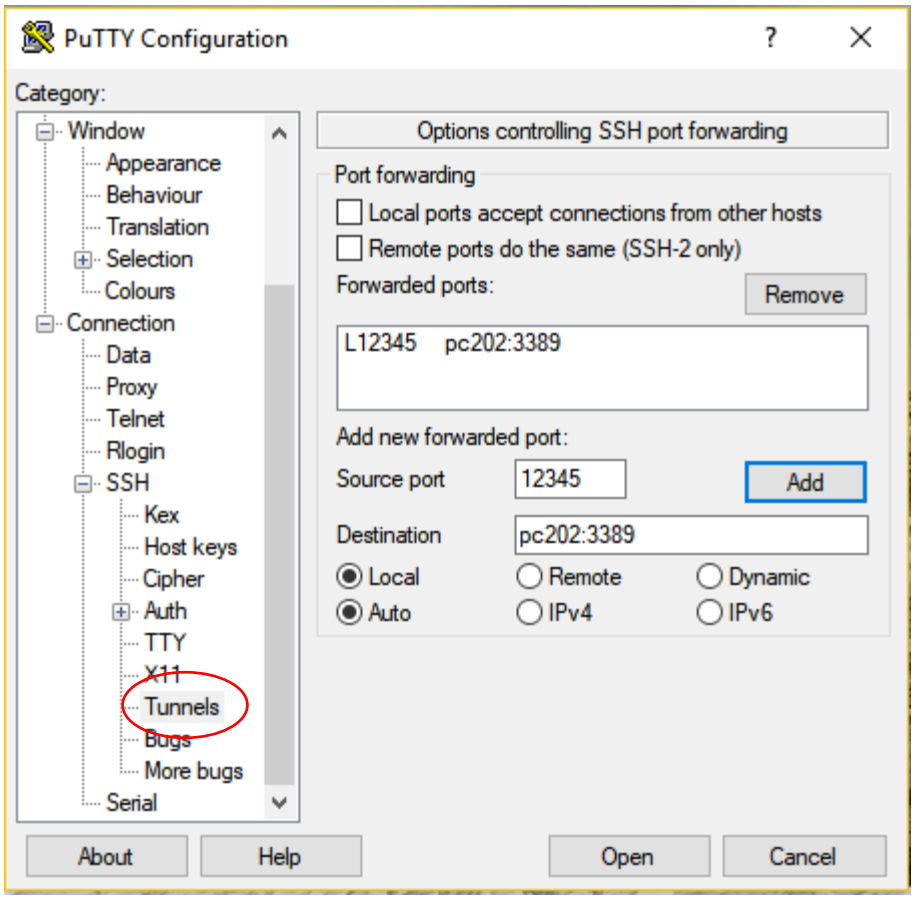
Most malware samples only run on Windows machines, and unpacking the files requires rdp access. Follow the steps below to use rdp and access a Windows node's desktop:

1. Start DETELab experiment. Booting Windows nodes is unreliable, and often requires about 15-20 minutes.
2. Open PuTTY



Use users.isi.deterlab.net as the host name.

3. To create an SSH tunnel, expand the SSH tab on the left. For the source port, you can enter any port you want.



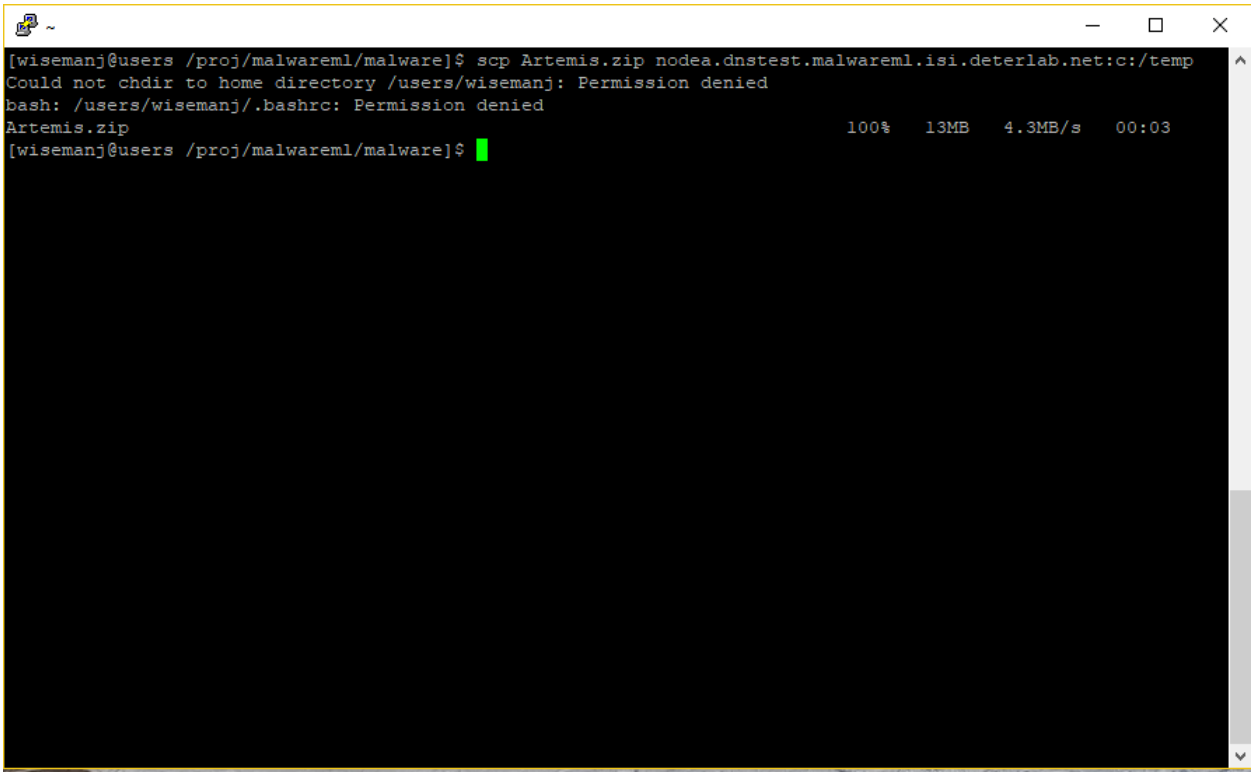
For the destination port, you'll have to get the name of the node that is booted in Windows.

Node ID	Name	Type	Default OSID	Node Status	Hours Idle[1]	Startup Status[2]	Disk Image	Snapshot	Log
pc162	nodeB	pc3060	Ubuntu1604-STD	up	0	none	Create New Disk Image	Snapshot Disk to Image	
pc177	nodeC	pc3060	Ubuntu1604-STD	up	0	none	Create New Disk Image	Snapshot Disk to Image	
pc202	nodeA	pc3060	WINXP-UPDATE	up	0	none	Create New Disk Image	Snapshot Disk to Image	

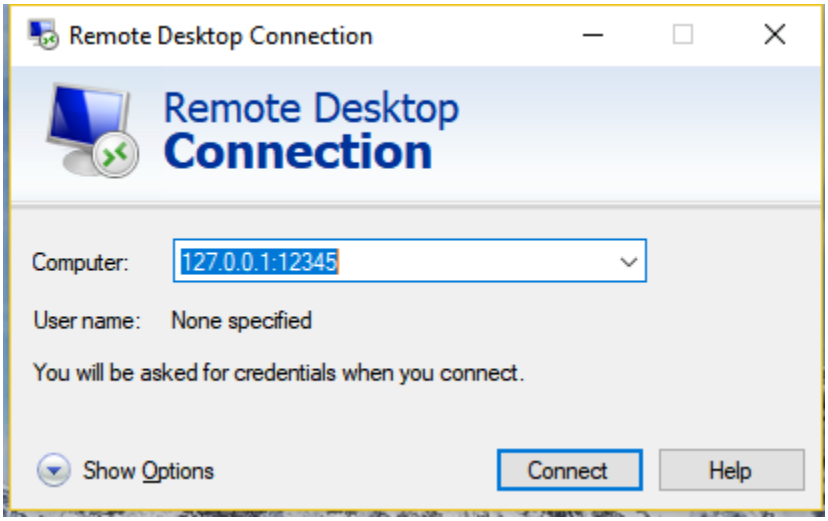
Looking at the project overview page can tell you the id of this node. Use port 3389, the designated RDP port. Press "Add" to open the SSH tunnel.

4. Open the PuTTY connection, and SSH into your Windows node to change your password; Windows passwords are different from your DETERLab password. SSH format for nodes is always: virtualName.experimentName.projectName.isi.deterlab.net (here, that would be nodeA.dnstest.malwareml.isi.deterlab.net). Use net user USER NEWPASS to change your Windows password.

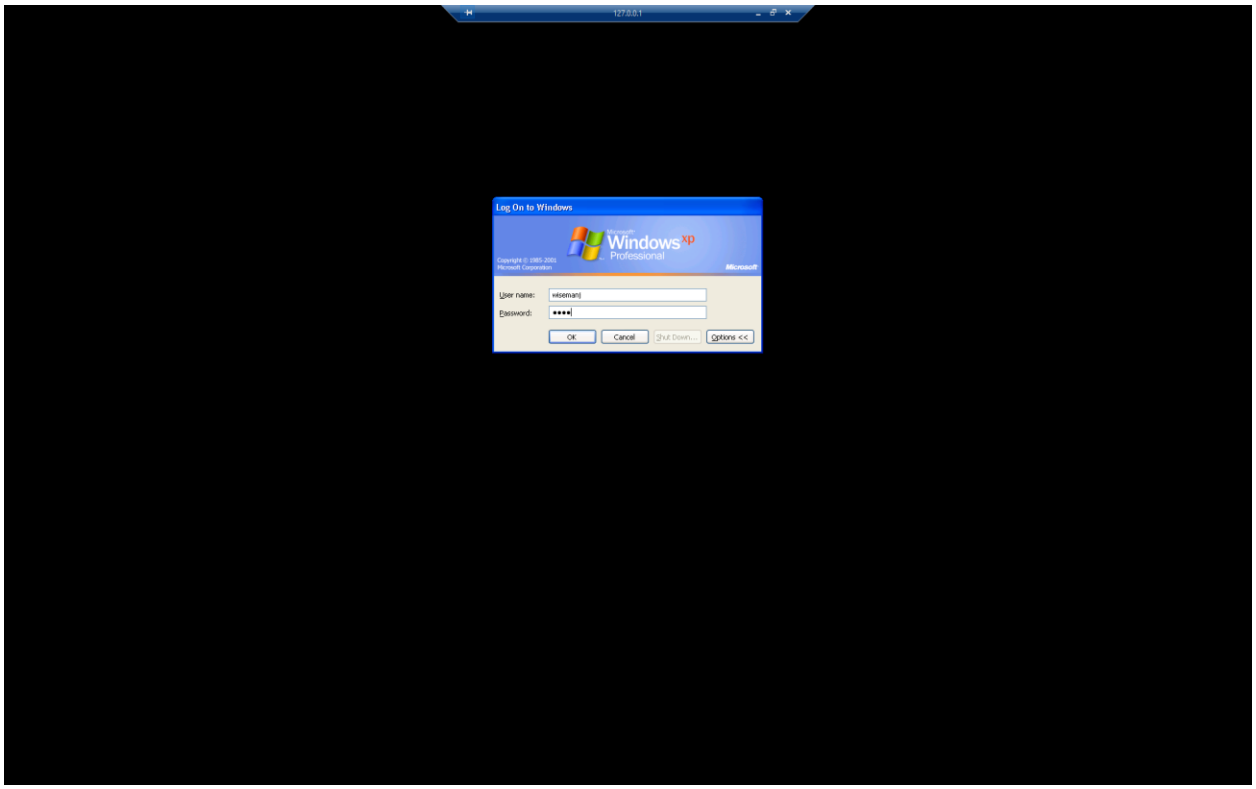
5. Exit the Windows SSH and transfer whatever files you want on the Windows machine using scp:



6. Open the RDP client and specify the localhost and port that you entered into PuTTY earlier:



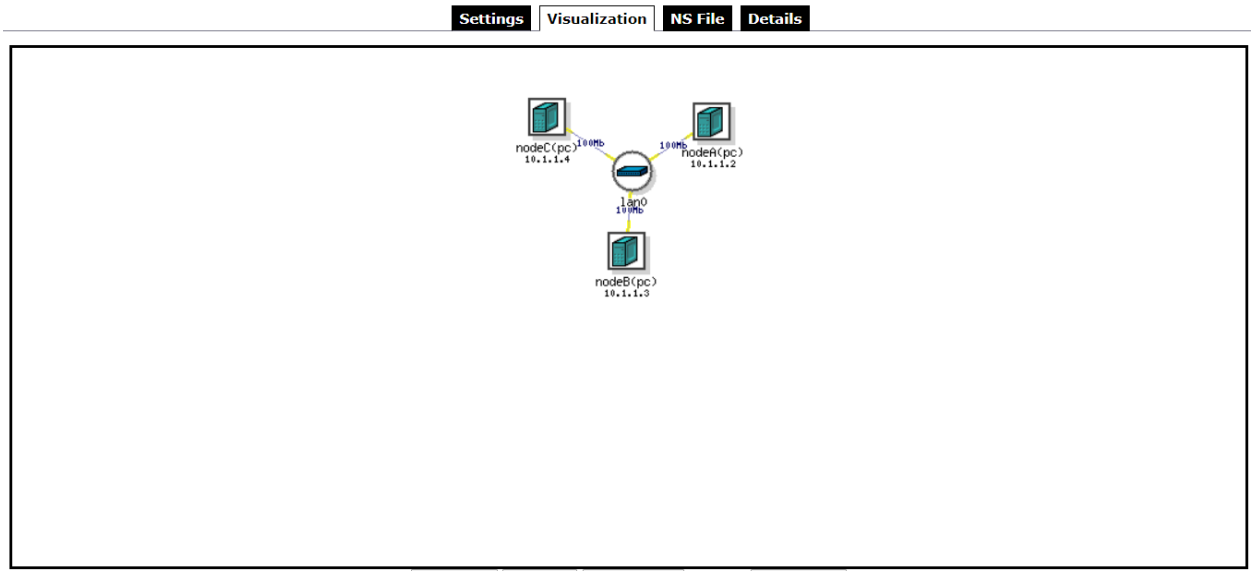
Connect and enter your login information in the remote desktop:



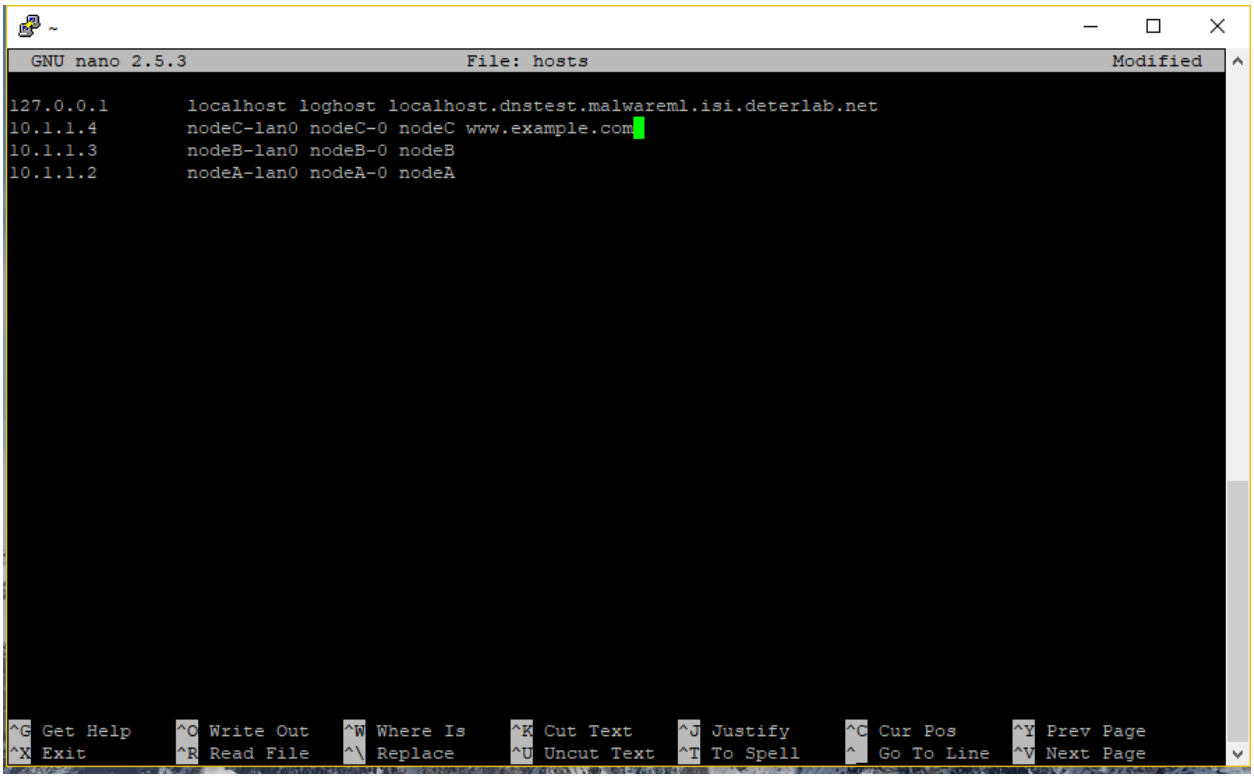
7. From the remote desktop, you can unzip malware files and run whatever programs you need.

How to Start DNS and Web Servers

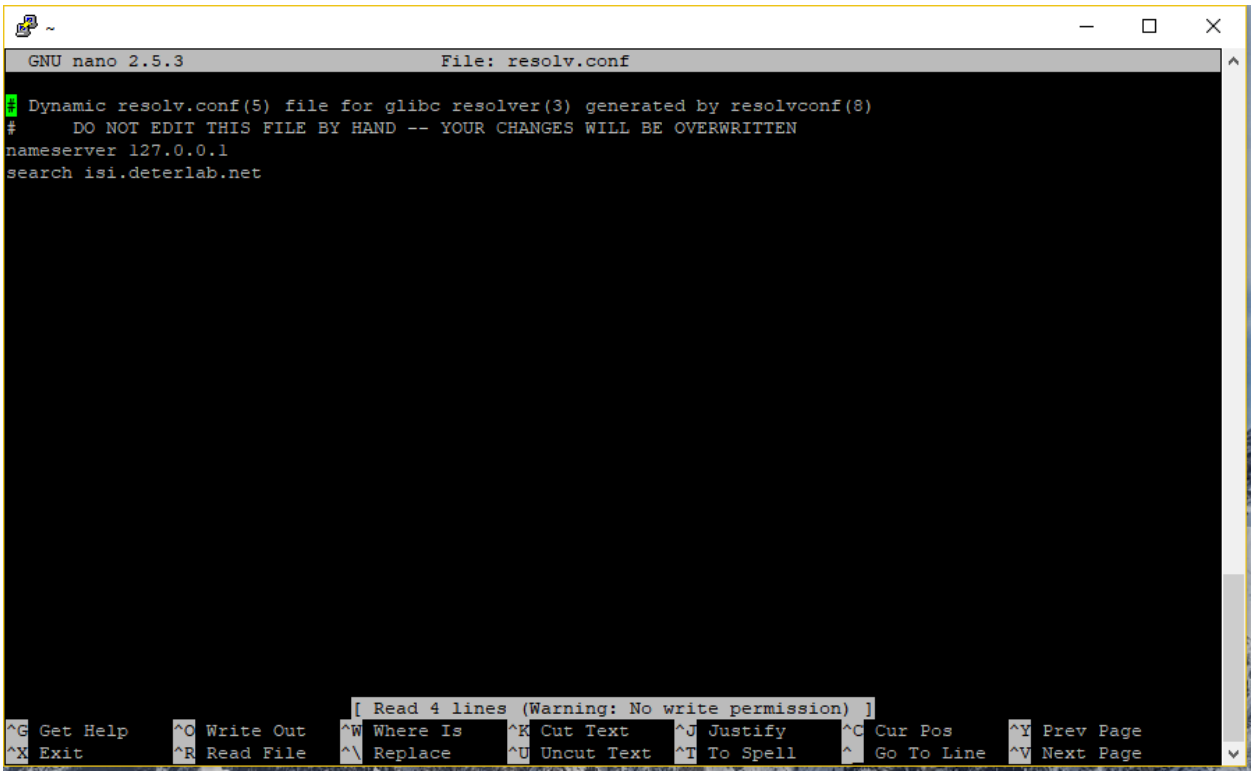
Node IPs can be found either using ipconfig/ifconfig or by checking the experiment page on DETERLab. The experiment page can give you a visualization of the topology:



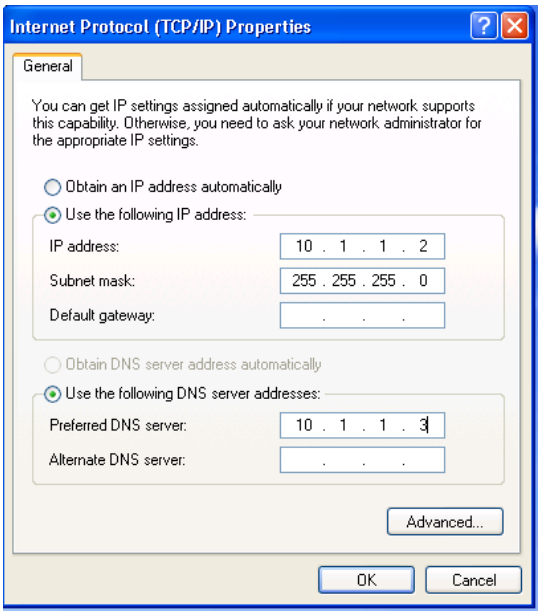
1. SSH into your Linux node and install the dnsmasq package.
2. dnsmasq will use the node's /etc/hosts file to resolve DNS queries. Update your hosts file as desired:



3. Start the dnsmasq service using `sudo service dnsmasq start`. To check that dnsmasq is running, see if the /etc/resolv.conf file has been updated. If it has, then the nameserver will be set to the localhost:

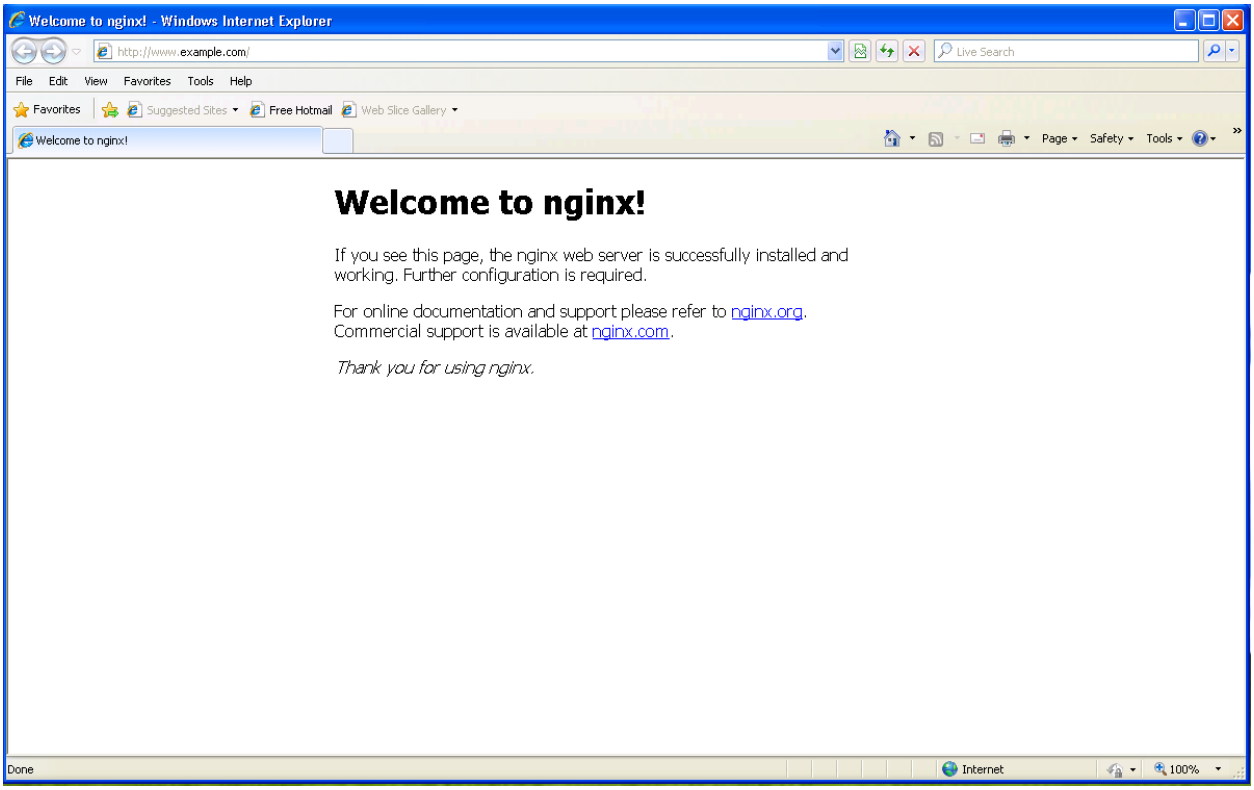


4. Now you can edit the nameservers of any other nodes using the DNS node's IP. On Windows, this is done by configuring network connections:



On Linux nodes, edit the /etc/resolv.conf file to change the nameserver.

5. To start a web server, SSH into a Linux node and install the nginx package. The steps for installing and configuring nginx can be found at: https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-16-04. Essentially: install the package, specify connections using `sudo ufw allow 'Nginx Full,'` and make sure it is running using `systemctl status nginx`.
6. Test your DNS and web server by going back into your Windows node, opening Internet Explorer, and navigating to any address specified in your DNS server's hosts file:



If everything is working, then you will be brought to the nginx welcome page. Any subdomain will return a 404 error. If everything is not working, then there will be no connection or a timeout. You could also use ping to test that the DNS server is working.