# GCP- Install a VM for an Indy Node

Introduction: The following steps are one way to adhere to the Indy Node guidelines for installing a GCP VM server to host an Indy Node. For the hardware requirements applicable for your network, please refer to the steward technical requirements document or the network governance documents for your network.

1. To prepare for VM creation, there are a few preliminary steps needed. You will need to set up items needed for Node networking. You will also need to create a snapshot schedule so that your VM can be backed up automatically (optional, but this is required for the only method described herein that satisfies the "backup" requirement).
2. From the GCP console (<https://console.cloud.google.com/>) scroll down to the 'Networking' section select 'VPC Network', then 'VPC Networks'
   1. Before you begin, select a 'region' to run your VM in that matches the jurisdiction of your company's corporate offices. Record the region selected for use throughout these instructions.
   2. Create 2 new VPC Networks using the following steps.
      1. Click 'CREATE VPC NETWORK' to create a network for your Client connection on your node.
      2. Name - your choice (e.g. client-vpc-9702)
      3. Description - your choice
      4. Subnets - select 'Custom' tab and create a new subnet.
         1. Name - your choice (e.g. client-subnet-9702)
         2. Region - Select the same region chosen earlier.
         3. IP address range - Type in a valid new subnet block. (e.g. 10.0.1.0/24)
         4. Private Google access - off
         5. Flow logs - your choice (e.g. off)
         6. Click 'Done'
      5. Dynamic routing mode - Regional
      6. DNS server policy - default (No server policy)
      7. Click 'Create'
      8. Repeat the above steps to create a second VPC Network for the Node IP of your server using names node-vpc-9701 and node-subnet-9701 and a range of 10.0.2.0/24
   3. Now set up the firewalls for your new VPC's
   4. Click on the Client VPC in the list of VPC Networks (e.g. client-vpc-9702)
      1. Click 'Firewall rules' in about the middle of the page, and then click 'Add firewall rule' to add SSH access through the Client VPC.
         1. Name - your choice (e.g. ssh-for-admin-access)
         2. Logs - Off
         3. Network - client-vpc-9702 (should already be set)
         4. Priority - default is fine
         5. Direction of traffic - Ingress
         6. Action on match - Allow
         7. Targets - All instances in the network (If you have other VM's using the same VPC as this one, then perform the optional steps listed next)
         8. OPTIONAL: Targets - Specified target tags
            1. Target tags - client9702 (record this value as you will need to associate it later with the VM.)
         9. Source filter - IP ranges
         10. Source IP ranges - Enter the public IP addresses or ranges for your Node Administrators. (e.g. 68.179.145.150)
         11. Protocols and ports - Specified protocols and ports
             1. Check the tcp box and enter 22 for the port.
         12. Click 'Create'
      2. Click 'Firewall rules' in about the middle of the page, and then click 'Add firewall rule' to add port 9702 access through the Client VPC.
         1. Name - your choice (e.g. client-access-9702)
         2. Logs - Off
         3. Network - client-vpc-9702 (should already be set)
         4. Priority - default is fine
         5. Direction of traffic - Ingress
         6. Action on match - Allow
         7. Targets - All instances in the network
         8. Source filter - IP ranges
         9. Source IP ranges - Enter the signification for "all access" (e.g. 0.0.0.0/0)
         10. Protocols and ports - Specified protocols and ports
             1. Check the tcp box and enter 9702 for the port.
         11. Click 'Create'
      3. Click the back arrow to return to the 'VPC networks' view
      4. Click on the node-vpc-9701 network then click 'Firewall rules' to add some rules.
      5. Ask your network administrator for a list of node IPs to add to your 'whitelist' as part of the following steps. For each node IP on the network, do the following:
         1. Click 'Add firewall rule'
         2. Name- Name (alias) of the node you are adding (the next name in the list)
         3. Logs - Off
         4. Network - (should already be set)
         5. Priority - default is fine
         6. Direction of traffic - Ingress
         7. Action on match - Allow
         8. Targets - All instances in the network
         9. Source filter - IP ranges
         10. Source IP ranges - Enter the public IP address matching the Node name that you are adding. (e.g. 68.179.145.150)
         11. Protocols and ports - Specified protocols and ports
             1. Check the tcp box and enter 9701 for the port.
         12. Click 'Create'
      6. Repeat the last set of steps for each node in the node list, changing the node Name and IP address for each new rule.
3. From the GCP 'Compute Engine' console, click 'Snapshots in the left pane
   1. Select the 'SNAPSHOT SCHEDULES' tab then click 'CREATE SNAPSHOT SCHEDULE'
   2. Name - your choice (e.g. 'nodesnapweekly')
   3. Region - Select the same region chosen earlier in this guide.
   4. Snapshot location - Regional (default location)
   5. Schedule frequency - Weekly (then your choice of day and time.)
   6. Autodelete snapshots after - 60 days
   7. Deletion rule - your choice (e.g. Select 'Delete snapshots older than days' to remove the snapshots after you no longer need the VM)
   8. Other options - your choice (defaults are fine)
   9. Click 'CREATE'
4. From the GCP Compute Engine console, click 'VM Instances' in the left pane
5. Click 'Create' (or 'CREATE INSTANCE')
6. WARNING: Do not click enter or return at any time during the filling out of the form that is now displayed. Clicking enter before you are ready might inadvertently create the VM before all setup steps have been completed and you might have to delete the VM and start over. (Don't ask how I know this ...)
7. Select 'New VM instance' in the left pane
8. Name - validatornode1
9. Labels - none needed
10. Region - Select the same region selected earlier in this guide.
11. Machine configuration
    1. Select 'General-purpose' tab
    2. Series - N1 is probably sufficient
    3. Machine Type - Select a type with 8 vCPUs and 32G RAM or greater (n1-standard-8 is close enough)
12. Container - leave unchecked (not needed)
13. Boot disk - Click 'Change'
    1. Select the 'Public images' tab (default)
    2. Operating system - select “Ubuntu”
    3. Version - 'Ubuntu 16.04 LTS' (default)
    4. Boot disk type - your choice (Standard is sufficient)
    5. Size - default is sufficient (10 GB)
    6. Click 'Select'
14. Identity and API access - leave at defaults (not sure this is correct for a Validator Node …)
15. Firewall - leave boxes unchecked
16. Click to expand the ' [Management, security, disks, networking, sole tenancy](https://console.cloud.google.com/)' section
    1. Management Tab
       1. Description - your choice
       2. Deletion protection - Check the box (recommended)
       3. (the rest of the options under Management) - your choice (defaults will work)
    2. Security Tab
       1. Shielded VM - (defaults)
       2. SSH Keys
          1. Check the box to 'Block project-wide SSH keys' (recommended)
          2. Enter a public SSH key for each Admin user (at least your own)
          3. To create an SSH key:
             1. You can use the following command to create a new SSH key pair on Linux or MAC that will work for this step.

ssh-keygen -P "" -t rsa -b 4096 -m pem -f ~/pems/validatornode.pem

* + - * 1. Once a public key is created the following example can be used on MAC or Linux to display the public key and copy it to the form:

cat ~/pems/validatornode.pem.pub

* + - * 1. Copy the results of the above and paste it into the space provided being careful NOT to copy any leading or trailing whitespace.
      1. Do NOT click 'Create' yet!, Please proceed to the Disks tab.
  1. Disks tab
     1. Defaults work for initial items
     2. Click '+ Add new disk`
        1. Name - your choice (e.g. nodedatadisk)
        2. Description - your choice
        3. Type - 'Standard persistent disk' is fine
        4. Snapshot schedule - nodesnapweekly (created earlier in these instructions) If it does not appear in the list, type the name in and then select it.
        5. Source type - Blank disk
        6. Mode - Read-write
        7. Deletion rule - your choice (e.g. use 'Delete disk' to make sure there are no unseen charges when you no longer need the VM)
        8. Size - 1024
        9. Defaults are fine for the rest of this section
        10. Click 'Done' (at the bottom of the Disks section)
  2. Networking tab
     1. Network tags - leave blank
     2. Hostname - default should be fine
     3. Network interfaces (1) - Fill in the fields for the network interface that will correspond to the Client interface. The Node interface will be the second interface created for this instance.
        1. Network - client-vpc-9702
        2. Subnetwork - default
        3. Primary internal IP - Reserve static internal IP
           1. Name - your choice (e.g. client-internal-ip)
           2. Description - optional
           3. Subnet - default
           4. Static IP address - Assign automatically
           5. Purpose - Non-shared
           6. Click 'RESERVE'
        4. External IP - Create IP address
           1. Name - your choice (client-public-ip)
           2. Description - optional
           3. Network Service Tier - Premium (NOTE: premium costs a bit more, but it was the only option available in my region. Standard is probably okay to use here if you are concerned about the small price increase for premium.)
           4. Click 'RESERVE'
        5. IP forwarding - off
        6. Public DNS PTR Record - unchecked
        7. Click 'Done'
     4. Click '+Add network interface'
        1. (Click the pencil icon to edit the new interface just added, if it is not expanded.)
        2. Network - node-vpc-9701
        3. Subnetwork - default
        4. Primary internal IP - Reserve static internal IP
           1. Name - your choice (e.g. node-internal-ip)
           2. Description - optional
           3. Subnet - default
           4. Static IP address - Assign automatically
           5. Purpose - Non-shared
           6. Click 'RESERVE'
        5. External IP - Create IP address
           1. Name - your choice (node-public-ip)
           2. Description - optional
           3. Network Service Tier - Premium
           4. Click 'RESERVE'
        6. Click 'Done'
  3. Sole Tenancy tab
     1. Defaults should be fine here

1. Click 'Create' to create the new GCP VM instance.
2. Wait for your VM to launch.
3. Log in to your VM
   1. From your workstation do the following: (Windows 10 will be different)
   2. ssh -i <public rsa key file> <username>@<Client IP Address>
   3. Where rsa key file was the ssh key .pem file generated earlier
   4. And 'username' is the username from the rsa .pub file.
   5. And 'Client IP Address' is the External IP address showing in the VM list.
   6. for example: ssh -i ~/pems/gpcnode.pem lynnbendixsen@13.58.197.208
   7. NOTE: I got an error the first time I ran the above to login: "Permission denied" because "Permissions are too open" <for your pem file>. To correct the issue I ran chmod 0600 ~/pems/gpcnode.pem and then I was able to login successfully.
4. Configure networking to the second NIC
   1. From your instance's command prompt, run the command $ ip a and verify that you have 2 internal IP addresses that match what you have in your Node Installation Info spreadsheet. Note the names of the network interfaces. (mine were ens4 and ens5) The remaining instructions in this section assume ens4 is your original primary NIC (Client NIC) and ens5 is the secondary NIC (Node NIC).
   2. route -n
      1. Record the default gateway for later use. (e.g. mine is 10.0.1.1)
   3. Disable automatic network management by GCP. Run the following:
      1. sudo su -
      2. echo 'network: {config: disabled}' > /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg
   4. cd /etc/network/interfaces.d
   5. vim 50-cloud-init.cfg
      1. Cut the existing ens4 lines from this file in preparation for moving them to a new file in this same directory. (Mine only had 2 lines to remove.)
      2. Save the changed file.
      3. Example 50-cloud-init.cfg now looks like:   
            
          auto lo  
          iface lo inet loopback
   6. Create 2 new network interface files using the following suggestions. These example files are configured so that a GCP VM can use 2 IP addresses on 2 different interfaces and subnets.
   7. vim ens4.cfg (use <interface name>.cfg if your interface name is not ens4)
      1. For the following, substitute the Gateway and interface names recorded earlier for <Gateway> and <interface name> respectively. <interface name> should also match the name of the file it is in. Paste the ens4 lines cut from the 50-cloud-init.cfg file and then add the following lines, indented 3 spaces:  
           
          up ip route add default via <Gateway> dev <interface name> tab 1  
          up ip rule add from <local IP addr of <interface name>>/32 tab 1  
          up ip rule add to <local IP addr of <interface name>>/32 tab 1  
          up ip route flush cache

* 1. Example ens4.cfg  
       
     auto ens4

iface ens4 inet dhcp

up ip route add default via 10.0.1.1 dev ens4 tab 1

up ip rule add from 10.0.1.2/32 tab 1

up ip rule add to 10.0.1.2/32 tab 1

up ip route flush cache

* 1. Repeat previous steps but for the second network interface. The simplest way to do that is probably:
     1. cp ens4.cfg ens5.cfg
     2. vi ens5.cfg
        1. Replace all instances of ens4 with ens5
        2. Change <local IP addr> to the one corresponding to ens5
        3. Change ‘tab 1’ to ‘tab 2’
        4. Change the Gateway to the correct value (I used 10.0.2.1, but see below for how I found out what it was)
  2. Example ens5.cfg  
       
     auto ens5

iface ens5 inet dhcp

up ip route add default via 10.0.2.1 dev ens5 tab 2

up ip rule add from 10.0.2.2/32 tab 2

up ip rule add to 10.0.2.2/32 tab 2

up ip route flush cache

* 1. ifup ens5
     1. Check to make sure ens5 came up and is working properly using the command ip a. NOTE: When I ran ifup, I got the error "Failed to bring up ens5." because I had the wrong Gateway in ens5.cfg. At this point if you run route -n again, you will see the correct gateway and can adjust ens5.cfg to use it.
     2. If the ens4 interface becomes unusable, you should now be able to log in through ens5 to fix it (if you first allow ssh to the second NIC in the VPC network console).
  2. Restart your instance.
     1. reboot
  3. ssh to your instance again as described earlier.
     1. ssh -i <public rsa key file> ubuntu@<Client IP Address>

1. Configure and mount the data disk.
   1. Find the name of your data disk:
      1. sudo fdisk -l
      2. In most cases **/dev/sdb** will be the name of the 1Tib data disk created during the GPC VM setup.
   2. The following steps assume that your disk size is less than 2 TiB, that your disk is /dev/sdb and that you will be using MBR partitioning.
   3. sudo fdisk /dev/sdb
      1. Create a new partition
         1. n
         2. p
         3. <defaults for the rest> TIP: press enter 3 times to accept the defaults and complete the process of creating a partition.
         4. Now, print and write the partition and exit.
         5. p
         6. w
   4. Update the kernel:
      1. partprobe
   5. Add a filesystem to your new disk partition:
      1. sudo mkfs -t ext4 /dev/sdb1
   6. Mount the disk to the directory where the Node software does the most writing (/var/lib/indy):
      1. sudo mkdir /var/lib/indy
      2. sudo mount /dev/sdb1 /var/lib/indy
   7. Add the drive to /etc/fstab so that it mounts at server startup.
      1. sudo blkid
      2. Record the UUID of /dev/sdb1 for use in the /etc/fstab file.
      3. sudo vim /etc/fstab
      4. Add the following line to the end of the fstab file (substituting in your own UUID):
         1. UUID=70cfcc21-230b-4d25-bb41-596412d0ca7f /var/lib/indy ext4 defaults,nofail 1 2
         2. Vim Hint: In vim, arrow down to the last line of the file, press the ‘o’ key and then paste in the above line. As before, <esc> then :wq will write and exit the file.
2. Restart the instance to check for NIC and Disk persistence.
   1. From the GCP Compute Engine console, click 'VM Instances' in the left pane, select your VM and click 'Reset'.
   2. Login to your VM as before:
      1. ssh -i <public rsa key file> <username>@<Client IP Address>
   3. Check the NIC
      1. ip a
      2. The output of the above command should have 2 NICS with the correct IP addresses displayed.
      3. df -h
      4. The output of the above command should show /var/lib/indy mounted to the /dev/sdb1 disk with the correct size (1T).
      5. More NIC and disk verifications will occur during the Indy Node install process.
3. Optional: Add a temporary administrative user as a safety net during Two Factor Authentication (2FA) setup.
   1. sudo adduser tempadmin
      1. You can safely ignore messages like “sent invalidate(passwd) request, exiting“
   2. sudo usermod -aG sudo tempadmin
   3. Setup sshd\_config to temporarily allow password login for the tempadmin user.
      1. sudo vim /etc/ssh/sshd\_config
      2. Comment out the line containing ‘ChallengeResponseAuthentication’.
         1. #ChallengeResponseAuthentication no
      3. Make sure this line exists and is set to yes:
         1. PasswordAuthentication yes
      4. :wq to save and exit.
      5. sudo systemctl restart sshd
      6. The above lines will be altered again when you set up 2FA.
   4. To be able to login, you will also likely need to setup an ssh key
      1. sudo mkdir /home/tempadmin/.ssh
      2. sudo chown tempadmin:tempadmin /home/tempadmin/.ssh
      3. sudo vim /home/tempadmin/.ssh/authorized\_keys
      4. Paste the users public key into the open file and then save it (:wq) (You can use the same key as you used for the base admin user in this case, since it is a temporary user)
      5. sudo chown tempadmin:tempadmin /home/tempadmin/.ssh/authorized\_keys
4. Setup 2FA for SSH access to the Node for your base user.
   1. Optional: Login in a separate terminal as your tempadmin user (that has sudo privileges) to have a backup just in case something goes wrong during setup.
      1. ssh tempadmin@<Client IP Addr>
   2. Install Google Authenticator, Duo, or Authy on your phone.
   3. As your base user on the Node VM, run the following to install the authenticator:
      1. sudo apt-get install libpam-google-authenticator
   4. Configure the authenticator to allow both password and SSH key login with 2FA by changing 2 files:
      1. sudo vim /etc/pam.d/common-auth
      2. Add the following line as the first uncommented line in the file
         1. auth sufficient pam\_google\_authenticator.so
         2. :wq
      3. sudo vim /etc/ssh/sshd\_config
         1. add/configure the following lines:
            1. ChallengeResponseAuthentication yes
            2. UsePAM yes
            3. AuthenticationMethods publickey,keyboard-interactive
            4. PasswordAuthentication no
         2. If you see any of the above lines commented out, remove the # to uncomment them. If you don't see any of the above lines, make sure to add them. If you see those lines configured in any different way, edit them to reflect the above.
         3. :wq
      4. sudo systemctl restart sshd
   5. Setup your base user to use 2FA by running the following from a terminal:
      1. google-authenticator
      2. Answer ‘y’ to all questions asked during the setup
      3. Save the secret key, verification code and scratch codes in a safe place. These are all just for your user and can be used to login or to recover as needed.
   6. On your phone app add an account and then scan the barcode or enter the 16 character secret key from the previous steps output.
   7. You should now be able to login using 2FA. First, check that login still works for your base user in a new terminal. If that doesn’t work, double check all of the configuration steps above and then restart sshd again. If it still doesn’t work, it’s possible that a server restart is required to make 2FA work (NOTE: It is dangerous to restart at this point, because then all of your backup terminals that are logged in will be logged out and there is a chance that you will lose access. Please check that all other steps have been executed properly before restarting.)
5. Add other administrative users:
   1. Send the other new admin users the following instructions for generating their own SSH keys:
      1. ssh-keygen -P "" -t rsa -b 4096 -m pem -f ~/pems/validatornode.pem
      2. Have the new users send you their public key (e.g. validatornode.pem.pub if they do the above command)
      3. Also have them send you their Public IP address so that you can add it to the GCP firewall to allow them access. Optionally, have them send a preferred username also.
   2. Add their IP addresses to the GCP firewall:
      1. From the GCP VPC Networks screen (GCP main menu -> VPC network->VPC networks), click on your Client VPC (e.g. client-vpc-9702)
      2. Click the 'Firewall rules' tab (in about the middle of the screen).
      3. Click on the name of the rule that allows port 22 access for your admins (e.g. ssh-for-admin-access)
      4. Click 'EDIT' at the top of the screen.
      5. Scroll down to the list of Source IP ranges and add the new Admins' IP addresses.
      6. Click ‘SAVE’ (Note: Restart is not needed. As soon as you save, they should have access.)
   3. Add the users to the server:
      1. Login to the Node as the base user.
      2. Run the following commands, substituting the username in for <newuser>
      3. sudo adduser <newuser>
         1. You can safely ignore messages like “sent invalidate(passwd) request, exiting“
      4. sudo usermod -aG sudo <newuser>
      5. Then create a file in the newusers home directory:
         1. sudo mkdir /home/<newuser>/.ssh
         2. sudo chown <newuser>:<newuser> /home/<newuser>/.ssh
         3. sudo vim /home/<newuser>/.ssh/authorized\_keys
         4. Paste the users public key into the open file and then save it (:wq)
         5. sudo chown <newuser>:<newuser> /home/<newuser>/.ssh/authorized\_keys
      6. Repeat the above for each new admin user you create.
   4. The new users are now able to login. Since 2FA is required, when you send the password to each of the new users, also send the following instructions (HINT: fill in the username, Client IP address, and password for them with the correct values):
      1. Thanks for agreeing to help with the administration of our Indy Validator Node. Please login to the node, change your password, and setup Two Factor Authentication (2FA) using the following instructions:
         1. ssh -i <your private SSH key file> <username>@<Client IP Addr>
         2. Type in <password> for your password
         3. On successful login, type in ‘passwd’ to change your password on the Validator Node. Please use a unique password of sufficient length and store it in a secure place (i.e. a password manager).
         4. To set up 2FA, type in ‘google-authenticator’
            1. Answer ‘y’ to all questions asked during the setup
            2. Save the secret key, verification code, and scratch codes in a safe place. These are all for your user and can be used to login or to recover as needed.
         5. Install Google Authenticator, Duo, Authy, or other google-authenticator compatible app on your phone or device.
         6. On your 2FA phone app, add an account, and then scan the barcode or enter the 16 character secret key from step 4’s output.
         7. Log out and then log back in to check and make sure it worked!
   5. All of your secondary admin users should be setup now.
6. You can now begin the Indy Node installation.

# Troubleshooting, Tips, and Acknowledgements

1. Helpful links (and acknowledgements to the authors of the content on these sites).
   1. <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-ubuntu-secondary-network-interface/>
   2. <https://www.digitalocean.com/community/tutorials/how-to-add-and-delete-users-on-ubuntu-16-04>
   3. <https://www.techrepublic.com/article/how-to-combine-ssh-key-authentication-and-two-factor-authentication-on-linux/>
   4. For adding a second user’s 2FA for them<https://www.linux.com/topic/desktop/how-set-2-factor-authentication-login-and-sudo/>