# [**Nimbus vCAv Testbeds**](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds)

[Skip to end of metadata](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#page-metadata-end)

* Created by [Jeff Mace](https://confluence.eng.vmware.com/display/~jmace), last modified by [Neil Armitage](https://confluence.eng.vmware.com/display/~narmitage) on [May 09, 2017](https://confluence.eng.vmware.com/pages/diffpagesbyversion.action?pageId=183118203&selectedPageVersions=61&selectedPageVersions=62)

[Go to start of metadata](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#page-metadata-start)

**The Nimbus code used for vCAv Testbeds is different from the stock Nimbus made available on other servers. Do not attempt to use this process elsewhere unless you absolutely need to.**

Nimbus allows individuals to create and manage automated testbeds on shared systems. Each testbed is a collection of VMs configured to work together. By default a testbed comes up with a limited lease time but you can extend the lease. Review the information in the [Nimbus QuickTutorial](https://confluence.eng.vmware.com/display/DevToolsDocKB/Nimbus+QuickTutorial) before proceeding.

* [Getting Started](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-GettingStarted)
* [Testbed Errors](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-TestbedErrors)
  + [Unable to find JSON description for the mypod1 testbed](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-UnabletofindJSONdescriptionforthemypod1testbed)
  + [The testbed was not successfully deployed](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-Thetestbedwasnotsuccessfullydeployed)
  + [The testbed postboot process failed. Try running it again](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-Thetestbedpostbootprocessfailed.Tryrunningitagain)
* [Custom Testbeds](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-CustomTestbeds)
* [Using the CPSBU Nimbus Pod](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-UsingtheCPSBUNimbusPod)
* [Testbed Customization](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-TestbedCustomization)
  + [Modify ESXi CPU and RAM](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-ModifyESXiCPUandRAM)
  + [Modify Shared Storage](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-ModifySharedStorage)
  + [Modify Local HDD Storage](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-ModifyLocalHDDStorage)
  + [Modify Local SSD Storage](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-ModifyLocalSSDStorage)
* [Nimbus operations](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-Nimbusoperations)
  + [Extend Lease for All Testbeds](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-ExtendLeaseforAllTestbeds)
  + [List Testbeds](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-ListTestbeds)
  + [Testbed Cleanup](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-TestbedCleanup)
  + [Extend Lease](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-ExtendLease)
* [Additional Products](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-AdditionalProducts)
  + [Loginsight](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-Loginsight)
* [Nimbus Reference](https://confluence.eng.vmware.com/display/vDR/Nimbus+vCAv+Testbeds#NimbusvCAvTestbeds-NimbusReference)

## Authenticating to GitHub

## <https://help.github.com/categories/authenticating-to-github/>

## Creating a personal access token for the command line

<https://help.github.com/articles/creating-a-personal-access-token-for-the-command-line/>

## Checking for existing SSH keys

Before you generate an SSH key, you can check to see if you have any existing SSH keys.<https://help.github.com/articles/checking-for-existing-ssh-keys/>

Error: Agent admitted failure to sign

In rare circumstances, connecting to GitHub via SSH on Linux produces the error "Agent admitted failure to sign using the key". Follow these steps to resolve the problem.

When trying to SSH into GitHub on a Linux computer, you may see the following message in your terminal:

<https://help.github.com/articles/error-agent-admitted-failure-to-sign/>

## Adding a new SSH key to your GitHub account

To configure your GitHub account to use your new (or existing) SSH key, you'll also need to add it to your GitHub account.

Before adding a new SSH key to your GitHub account, you should have:

* [Checked for existing SSH keys](https://help.github.com/articles/checking-for-existing-ssh-keys)
* [Generated a new SSH key and added it to the ssh-agent](https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent)

**Note:** DSA keys were deprecated in OpenSSH 7.0. If your operating system uses OpenSSH, you'll need to use an alternate type of key when setting up SSH, such as an RSA key. For instance, if your operating system is MacOS Sierra, you can set up SSH using an RSA key.

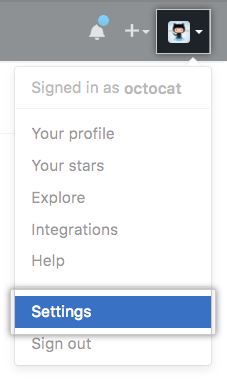
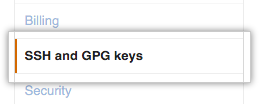
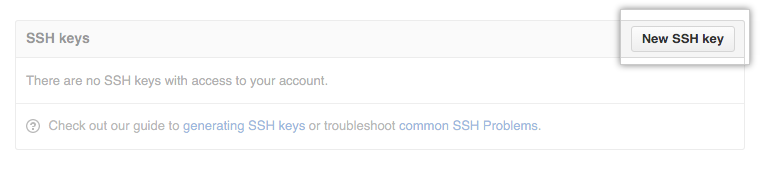
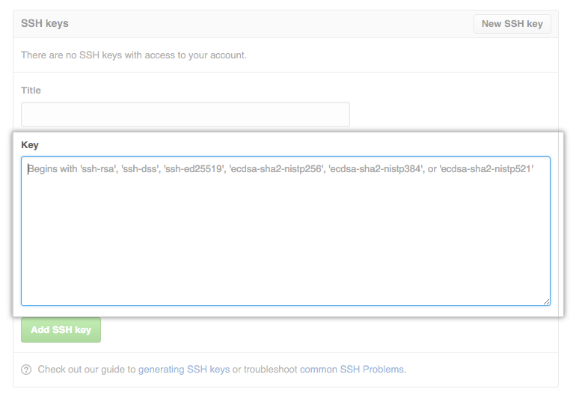
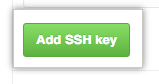
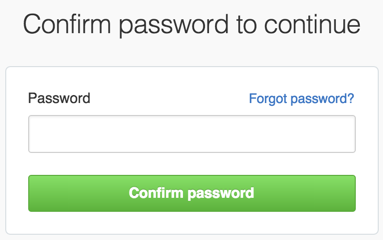
1. Copy the SSH key to your clipboard.

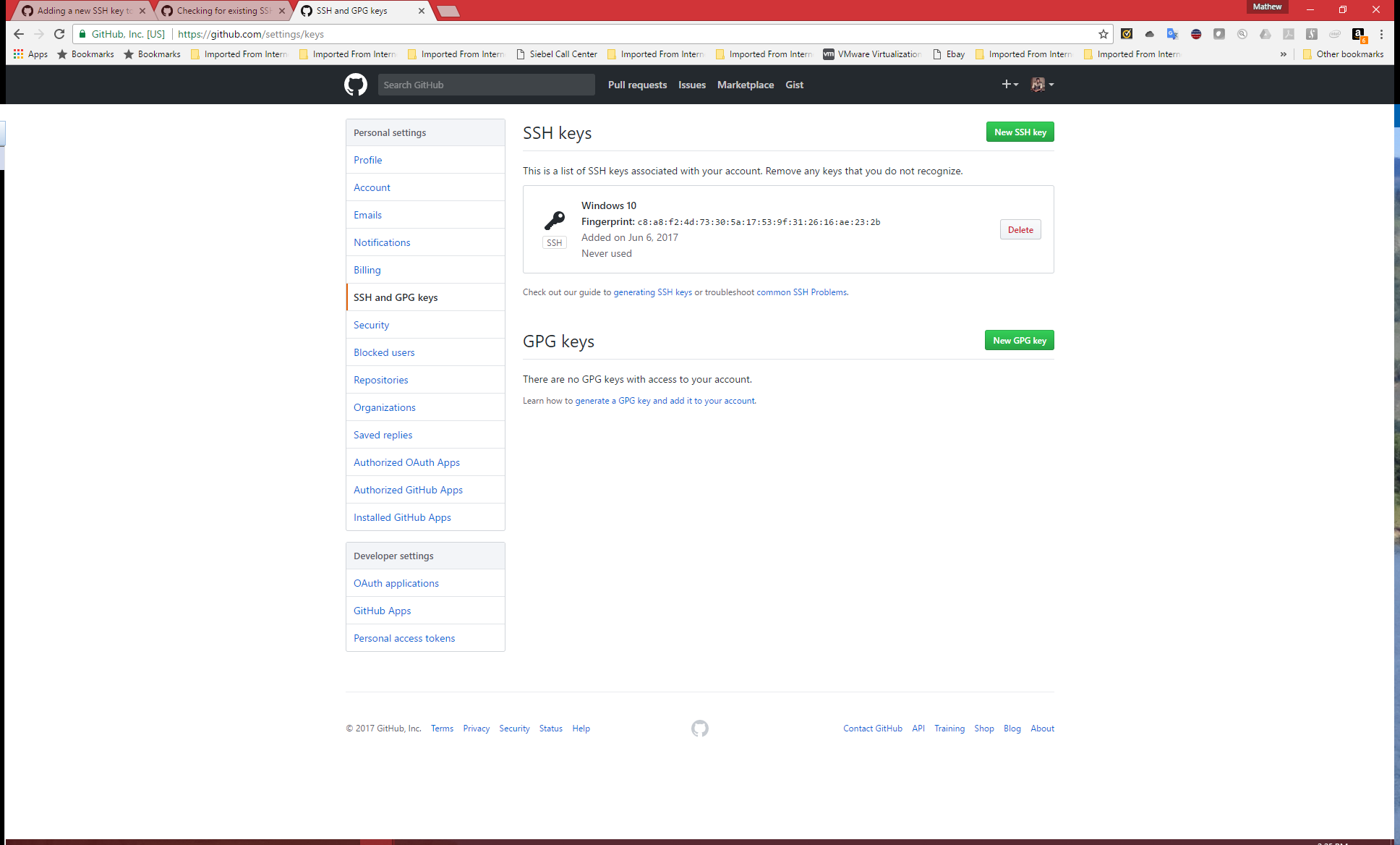
If your SSH key file has a different name than the example code, modify the filename to match your current setup. When copying your key, don't add any newlines or whitespace.

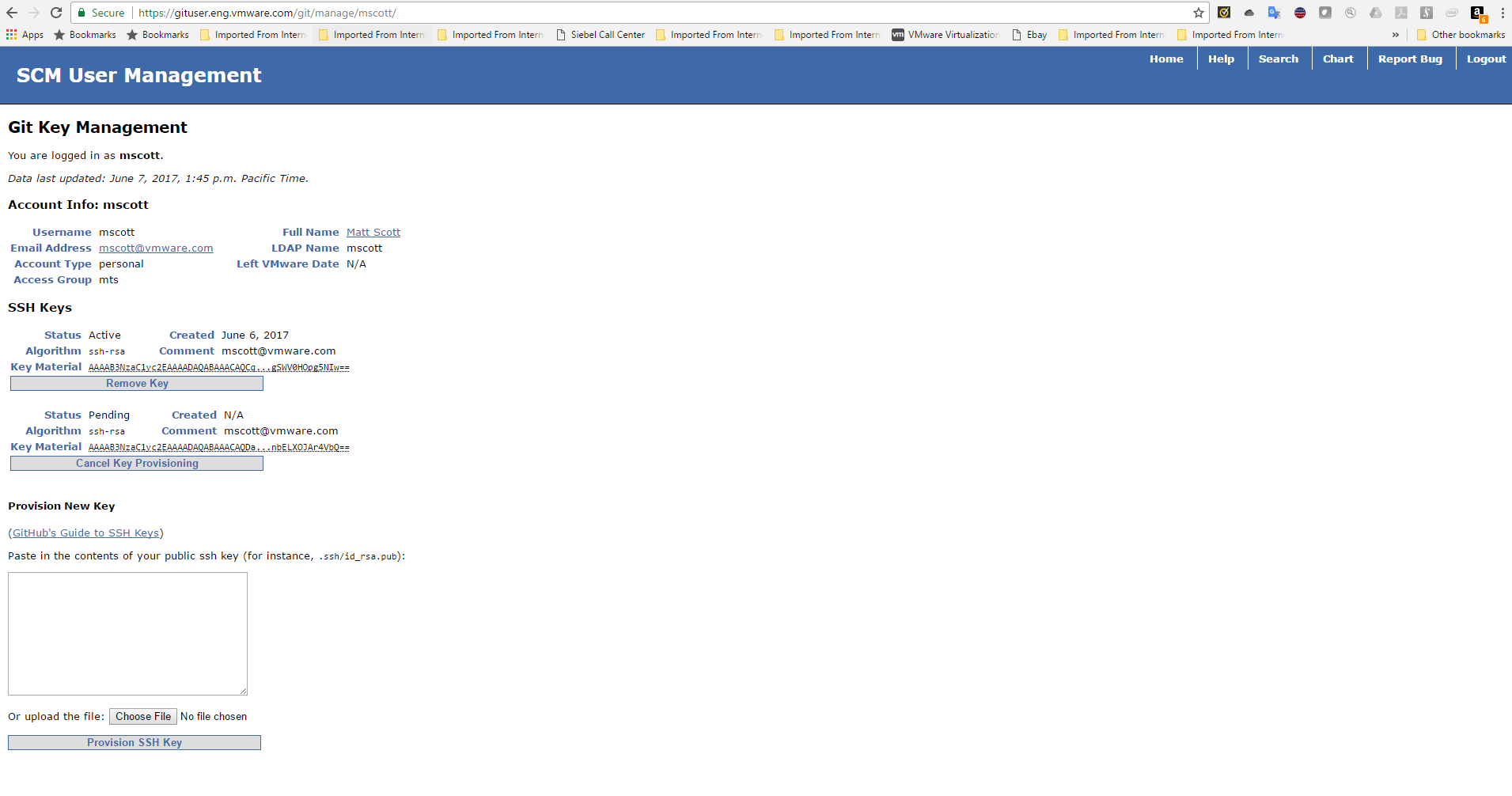
$ clip < ~/.ssh/id\_rsa.pub

# Copies the contents of the id\_rsa.pub file to your clipboard

**Tip:** If clip isn't working, you can locate the hidden .ssh folder, open the file in your favorite text editor, and copy it to your clipboard.

1. In the upper-right corner of any page, click your profile photo, then click **Settings**.
2. In the user settings sidebar, click **SSH and GPG keys**.
3. Click **New SSH key** or **Add SSH key**.
4. In the "Title" field, add a descriptive label for the new key. For example, if you're using a personal Mac, you might call this key "Personal MacBook Air".
5. Paste your key into the "Key" field.
6. Click **Add SSH key**.
7. If prompted, confirm your GitHub password.





<https://gituser.eng.vmware.com/git/manage/mscott/>

# [**Prepare server to run Nimbus**](https://confluence.eng.vmware.com/display/vDR/Prepare+server+to+run+Nimbus)

[Skip to end of metadata](https://confluence.eng.vmware.com/pages/viewpage.action?spaceKey=vDR&title=Prepare+server+to+run+Nimbus#page-metadata-end)

* Created by [Jeff Mace](https://confluence.eng.vmware.com/display/~jmace), last modified on [May 25, 2017](https://confluence.eng.vmware.com/pages/diffpagesbyversion.action?pageId=183120652&selectedPageVersions=4&selectedPageVersions=5)

[Go to start of metadata](https://confluence.eng.vmware.com/pages/viewpage.action?spaceKey=vDR&title=Prepare+server+to+run+Nimbus#page-metadata-start)

This is based on the steps at <https://wiki.eng.vmware.com/ToolsEng/Projects/Nimbus/FAQ#Using_nimbus_locallay_without_.2Fmts.2Fgit_mount>

1. Setup required environment variables

|  |
| --- |
| echo "export GIT\_BIN=git  export RUBY\_ROOT=/build/nimbus-toolchain/lin64  export PATH=$PATH:/build/mts-git/bin" | sudo tee /etc/profile.d/mts.sh |

1. Close the current session and login again to reload the shell environment
2. Setup curl at the expected path

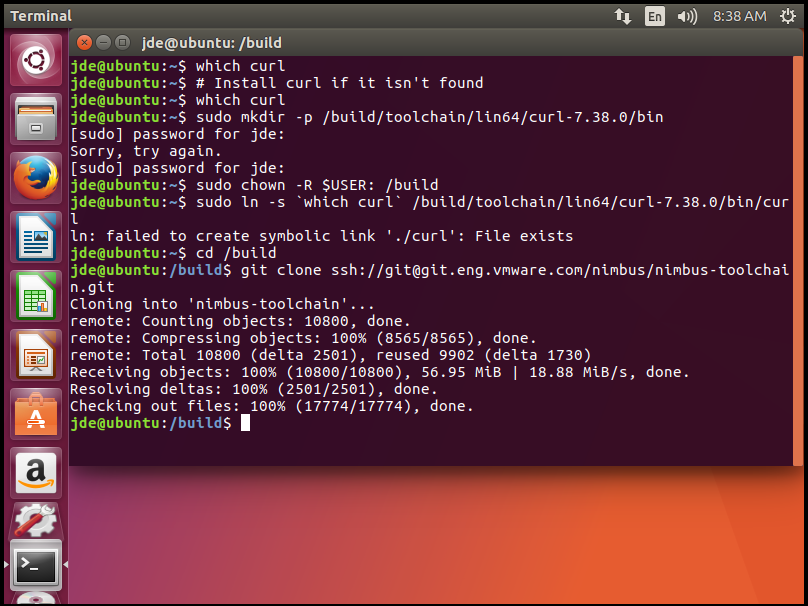
|  |
| --- |
| which curl  # Install curl if it isn't found    sudo mkdir -p /build/toolchain/lin64/curl-7.38.0/bin  sudo chown -R $USER: /build  sudo ln -s `which curl` /build/toolchain/lin64/curl-7.38.0/bin/curl |

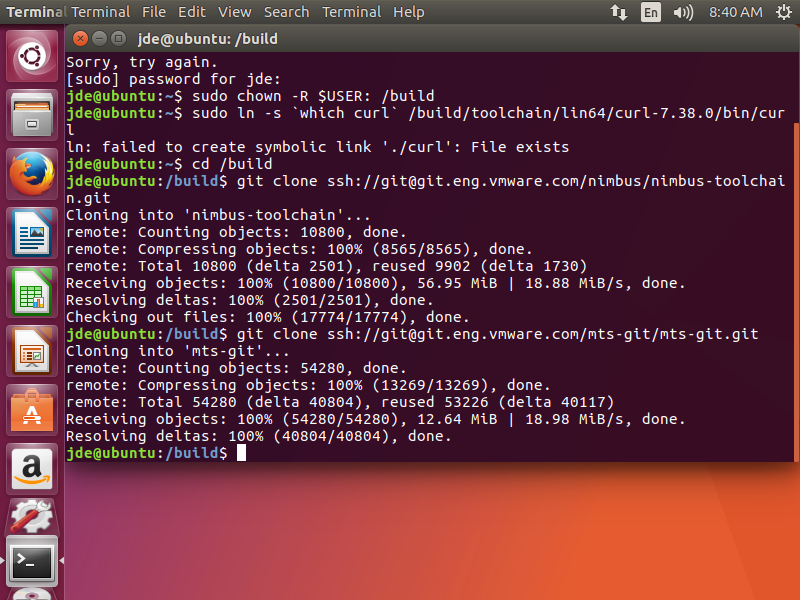
1. Setup the Nimbus toolchain

|  |
| --- |
| cd /build  git clone ssh://git@git.eng.vmware.com/nimbus/nimbus-toolchain.git  git clone ssh://git@git.eng.vmware.com/mts-git/mts-git.git |

1. Checkout all subtrees under mts-git

|  |
| --- |
| echo "staging" | tee /build/mts-git/.env  /build/mts-git/sbin/sync |





jde@ubuntu:/build$ /build/mts-git/sbin/sync

Fri Jun 9 08:41:08 PDT 2017 - updating /build/mts-git ...

Syncing from /build/mts-git/sbin/../repos

Fri Jun 9 08:41:09 PDT 2017 - cloning rlui ...

Fri Jun 9 08:41:09 PDT 2017 - cloning ddt ...

Fri Jun 9 08:41:09 PDT 2017 - cloning rlui-vmware ...

Fri Jun 9 08:41:09 PDT 2017 - cloning nimbus ...

Cloning into '/build/mts-git/sbin/../rlui'...

Cloning into '/build/mts-git/sbin/../rlui-vmware'...

Cloning into '/build/mts-git/sbin/../nimbus'...

Fri Jun 9 08:41:09 PDT 2017 - cloning rbvmomi-utils ...

Fri Jun 9 08:41:09 PDT 2017 - cloning rbvmomi ...

Fri Jun 9 08:41:09 PDT 2017 - cloning rbvmomi-misc ...

Cloning into '/build/mts-git/sbin/../ddt'...

Cloning into '/build/mts-git/sbin/../rbvmomi'...

Cloning into '/build/mts-git/sbin/../rbvmomi-misc'...

Cloning into '/build/mts-git/sbin/../rbvmomi-utils'...

Fri Jun 9 08:41:09 PDT 2017 - cloning rbvcloud ...

Fri Jun 9 08:41:09 PDT 2017 - cloning nimbusvc ...

Fri Jun 9 08:41:09 PDT 2017 - cloning rbvcloud-rewrite ...

Cloning into '/build/mts-git/sbin/../nimbusvc'...

Cloning into '/build/mts-git/sbin/../rbvcloud'...

Cloning into '/build/mts-git/sbin/../rbvcloud-rewrite'...

remote: Counting objects: 225, done.

remote: Compressing objects: 100% (105/105), done.

remote: Total 225 (delta 112), reused 225 (delta 112)

Receiving objects: 100% (225/225), 169.11 KiB | 0 bytes/s, done.

Resolving deltas: 100% (112/112), done.

Fri Jun 9 08:41:11 PDT 2017 - cloning rbbuildweb ...

Cloning into '/build/mts-git/sbin/../rbbuildweb'...

remote: Counting objects: 362, done.

remote: Compressing objects: 100% (126/126), done.

remote: Total 362 (delta 169), reused 362 (delta 169)

Receiving objects: 100% (362/362), 42.51 KiB | 0 bytes/s, done.

Resolving deltas: 100% (169/169), done.

remote: Counting objects: 180, done.

remote: Compressing objects: 100% (67/67), done.

remote: Total 180 (delta 113), reused 180 (delta 113)

Receiving objects: 100% (180/180), 47.79 KiB | 0 bytes/s, done.

Resolving deltas: 100% (113/113), done.

remote: Counting objects: 3221, done.

Fri Jun 9 08:41:11 PDT 2017 - cloning rvc ...

Cloning into '/build/mts-git/sbin/../rvc'...

Fri Jun 9 08:41:11 PDT 2017 - cloning rvc-vmware ...

Cloning into '/build/mts-git/sbin/../rvc-vmware'...

remote: Counting objects: 236, done.

remote: Compressing objects: 100% (91/91), done.

remote: Total 236 (delta 110), reused 198 (delta 92)

Receiving objects: 100% (236/236), 55.37 KiB | 0 bytes/s, done.

Resolving deltas: 100% (110/110), done.

remote: Counting objects: 1485, done.

Fri Jun 9 08:41:11 PDT 2017 - cloning tools ...

Cloning into '/build/mts-git/sbin/../tools'...

remote: Counting objects: 6, done.

remote: Compressing objects: 100% (4/4), done.

remote: Total 6 (delta 1), reused 0 (delta 0)

Receiving objects: 100% (6/6), done.

Resolving deltas: 100% (1/1), done.

remote: Counting objects: 3264, done.

remote: Counting objects: 225, done.

Fri Jun 9 08:41:11 PDT 2017 - cloning amqp ...

Cloning into '/build/mts-git/sbin/../amqp'...

remote: Compressing objects: 100% (1421/1421), done.

remote: Total 3221 (delta 1929), reused 2931 (delta 1752)

Receiving objects: 100% (3221/3221), 1.04 MiB | 0 bytes/s, done.

Resolving deltas: 100% (1929/1929), done.

Fri Jun 9 08:41:11 PDT 2017 - cloning eventmachine ...

Cloning into '/build/mts-git/sbin/../eventmachine'...

remote: Counting objects: 67947, done.

remote: Compressing objects: 100% (105/105), done.

remote: Total 225 (delta 112), reused 225 (delta 112)

Receiving objects: 100% (225/225), 169.11 KiB | 0 bytes/s, done.

Resolving deltas: 100% (112/112), done.

Fri Jun 9 08:41:12 PDT 2017 - cloning rbvmomi-vmware ...

Cloning into '/build/mts-git/sbin/../rbvmomi-vmware'...

remote: Counting objects: 204, done.

remote: Compressing objects: 100% (131/131), done.

remote: Total 204 (delta 67), reused 122 (delta 40)

Receiving objects: 100% (204/204), 37.03 KiB | 0 bytes/s, done.

Resolving deltas: 100% (67/67), done.

Fri Jun 9 08:41:12 PDT 2017 - cloning vsan-testbed ...

Cloning into '/build/mts-git/sbin/../vsan-testbed'...

remote: Counting objects: 1640, done.

remote: Counting objects: 1293, done.

remote: Compressing objects: 100% (631/631), done.

remote: Counting objects: 9686, done.

remote: Counting objects: 4921, done.

remote: Total 1293 (delta 556), reused 1151 (delta 482)

remote: Compressing objects: 100% (1744/1744), done.

Receiving objects: 100% (1293/1293), 439.93 KiB | 0 bytes/s, done.

Resolving deltas: 100% (556/556), done.859)

remote: Compressing objects: 100% (859/859), done.

Fri Jun 9 08:41:13 PDT 2017 - cloning vmknet-team ...

Cloning into '/build/mts-git/sbin/../vmknet-team'...

remote: Total 1640 (delta 1003), reused 1215 (delta 736)

Receiving objects: 100% (1640/1640), 518.51 KiB | 0 bytes/s, done.

Resolving deltas: 100% (1003/1003), done.840)

remote: Total 4921 (delta 3096), reused 4901 (delta 3084)

Receiving objects: 100% (4921/4921), 2.40 MiB | 0 bytes/s, done.

Fri Jun 9 08:41:13 PDT 2017 - cloning drac ...

Cloning into '/build/mts-git/sbin/../drac'...

Resolving deltas: 100% (3096/3096), done.840)

Fri Jun 9 08:41:13 PDT 2017 - cloning pyvpx-5\_0 ...

Cloning into '/build/mts-git/sbin/../pyvpx-5\_0'...

remote: Counting objects: 3510, done.

remote: Counting objects: 253, done.

remote: Counting objects: 262, done.

remote: Compressing objects: 100% (79/79), done.

remote: Total 262 (delta 182), reused 262 (delta 182)

Receiving objects: 100% (262/262), 61.69 KiB | 0 bytes/s, done.

Resolving deltas: 100% (182/182), done.

Fri Jun 9 08:41:14 PDT 2017 - cloning terminal-table ...

Cloning into '/build/mts-git/sbin/../terminal-table'...

remote: Compressing objects: 100% (183/183), done.

remote: Counting objects: 843, done.

remote: Total 253 (delta 147), reused 131 (delta 62)

Receiving objects: 100% (253/253), 1.02 MiB | 0 bytes/s, done.

remote: Compressing objects: 100% (464/464), done.

remote: Total 843 (delta 544), reused 588 (delta 375)

Receiving objects: 100% (843/843), 152.03 KiB | 0 bytes/s, done.

Resolving deltas: 100% (147/147), done.

Resolving deltas: 100% (544/544), done.

remote: Counting objects: 24, done.

remote: Compressing objects: 100% (15/15), done.

Receiving objects: 100% (24/24), 3.27 KiB | 0 bytes/s, done.

remote: Total 24 (delta 4), reused 0 (delta 0)

Resolving deltas: 100% (4/4), done.

Fri Jun 9 08:41:14 PDT 2017 - cloning hostdsim-bench ...

Cloning into '/build/mts-git/sbin/../hostdsim-bench'...

Fri Jun 9 08:41:14 PDT 2017 - cloning rbvapi ...

Cloning into '/build/mts-git/sbin/../rbvapi'...

remote: Counting objects: 576, done.

Fri Jun 9 08:41:14 PDT 2017 - cloning ddns ...

Cloning into '/build/mts-git/sbin/../ddns'...

remote: Compressing objects: 100% (509/509), done.

remote: Compressing objects: 100% (4840/4840), done.

remote: Compressing objects: 100% (880/880), done.

remote: Compressing objects: 100% (789/789), done.

remote: Total 3510 (delta 2504), reused 3510 (delta 2504)

Receiving objects: 100% (3510/3510), 781.20 KiB | 0 bytes/s, done.

remote: Total 1485 (delta 785), reused 870 (delta 448)

Receiving objects: 100% (1485/1485), 215.63 KiB | 0 bytes/s, done.

Resolving deltas: 100% (785/785), done. MiB | 15.62 MiB/s

remote: Total 576 (delta 59), reused 568 (delta 55)

Receiving objects: 100% (576/576), 8.87 MiB | 15.62 MiB/s, done.

Resolving deltas: 100% (2504/2504), done.

Resolving deltas: 100% (59/59), done.

Fri Jun 9 08:41:15 PDT 2017 - cloning vsan-tools ...MiB/s

Cloning into '/build/mts-git/sbin/../vsan-tools'...

Fri Jun 9 08:41:15 PDT 2017 - cloning nimbus-gems ...

Cloning into '/build/mts-git/sbin/../nimbus-gems'...

remote: Compressing objects: 100% (1276/1276), done.

remote: Total 3264 (delta 1941), reused 3264 (delta 1941)

Fri Jun 9 08:41:15 PDT 2017 - cloning nimbus-stats ...

Receiving objects: 100% (3264/3264), 338.20 KiB | 0 bytes/s, done.

Cloning into '/build/mts-git/sbin/../nimbus-stats'...

Resolving deltas: 100% (1941/1941), done.20887)

Fri Jun 9 08:41:15 PDT 2017 - cloning nimbus-configs ...

Cloning into '/build/mts-git/sbin/../nimbus-configs'...

remote: Counting objects: 869, done.

remote: Compressing objects: 100% (327/327), done.

remote: Total 869 (delta 514), reused 869 (delta 514)

Receiving objects: 100% (869/869), 109.81 KiB | 0 bytes/s, done.

Resolving deltas: 100% (514/514), done.9/20887)

Fri Jun 9 08:41:15 PDT 2017 - cloning internal-scripts ...

Cloning into '/build/mts-git/sbin/../internal-scripts'...s

remote: Total 9686 (delta 6294), reused 7160 (delta 4631)

Receiving objects: 100% (9686/9686), 6.53 MiB | 9.16 MiB/s, done.

remote: Compressing objects: 100% (20887/20887), done.

Resolving deltas: 100% (6294/6294), done.

remote: Counting objects: 65, done.

remote: Compressing objects: 100% (61/61), done.

remote: Counting objects: 54, done.

remote: Total 65 (delta 21), reused 0 (delta 0)

Receiving objects: 100% (65/65), 9.39 KiB | 0 bytes/s, done.

remote: Compressing objects: 100% (39/39), done.

Resolving deltas: 100% (21/21), done.

remote: Total 54 (delta 13), reused 0 (delta 0)

Receiving objects: 100% (54/54), 7.21 KiB | 0 bytes/s, done.

Resolving deltas: 100% (13/13), done.

remote: Counting objects: 8, done.

remote: Compressing objects: 100% (4/4), done.

remote: Total 8 (delta 1), reused 0 (delta 0)

Receiving objects: 100% (8/8), done.

Resolving deltas: 100% (1/1), done.

Fri Jun 9 08:41:16 PDT 2017 - cloning nimbus-vra ...

Fri Jun 9 08:41:16 PDT 2017 - cloning ntf ...

Cloning into '/build/mts-git/sbin/../nimbus-vra'...

Cloning into '/build/mts-git/sbin/../ntf'...

Fri Jun 9 08:41:16 PDT 2017 - cloning devtools-pxe-library ...

Cloning into '/build/mts-git/sbin/../devtools-pxe-library'...

remote: Counting objects: 2999, done.

remote: Total 67947 (delta 49036), reused 64315 (delta 46041)

Receiving objects: 100% (67947/67947), 14.71 MiB | 16.05 MiB/s, done.

remote: Counting objects: 9359, done.

remote: Counting objects: 1338, done.

remote: Counting objects: 6400, done.

remote: Counting objects: 3654, done.

remote: Counting objects: 31, done.

remote: Compressing objects: 100% (19/19), done.

remote: Total 31 (delta 8), reused 31 (delta 8)

Receiving objects: 100% (31/31), 9.80 KiB | 0 bytes/s, done.

Resolving deltas: 100% (8/8), done.)

remote: Counting objects: 144, done.

remote: Compressing objects: 100% (142/142), done.

remote: Counting objects: 1367, done.

remote: Compressing objects: 100% (423/423), done.

remote: Total 144 (delta 60), reused 0 (delta 0)

Receiving objects: 100% (144/144), 124.00 KiB | 0 bytes/s, done.

remote: Total 1367 (delta 907), reused 1346 (delta 892)

Resolving deltas: 100% (60/60), done.

Receiving objects: 100% (1367/1367), 202.60 KiB | 0 bytes/s, done.

Resolving deltas: 100% (907/907), done.

Resolving deltas: 100% (49036/49036), done.

remote: Compressing objects: 100% (1832/1832), done.

remote: Total 2999 (delta 1406), reused 2545 (delta 1119)

Receiving objects: 100% (2999/2999), 3.96 MiB | 0 bytes/s, done.

Resolving deltas: 100% (1406/1406), done.

remote: Compressing objects: 100% (627/627), done.

remote: Total 1338 (delta 853), reused 1071 (delta 681)

Receiving objects: 100% (1338/1338), 1.13 MiB | 0 bytes/s, done.

Resolving deltas: 100% (853/853), done.

remote: Compressing objects: 100% (5472/5472), done.

remote: Compressing objects: 100% (2872/2872), done.

remote: Total 3654 (delta 1965), reused 908 (delta 498)

Receiving objects: 100% (3654/3654), 1.14 MiB | 0 bytes/s, done.

remote: Total 9359 (delta 3619), reused 9305 (delta 3603)

Resolving deltas: 100% (1965/1965), done.

Receiving objects: 100% (9359/9359), 6.08 MiB | 0 bytes/s, done.

remote: Compressing objects: 100% (5254/5254), done.

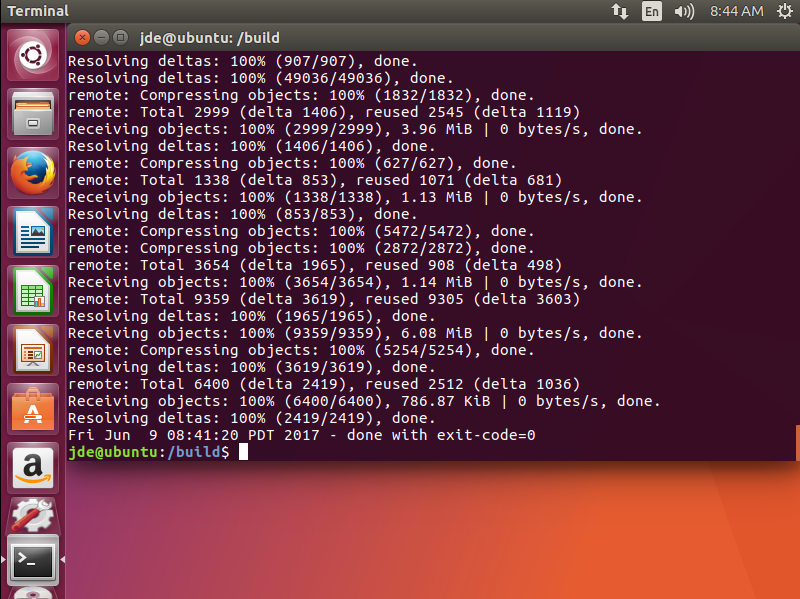
Resolving deltas: 100% (3619/3619), done.

remote: Total 6400 (delta 2419), reused 2512 (delta 1036)

Receiving objects: 100% (6400/6400), 786.87 KiB | 0 bytes/s, done.

Resolving deltas: 100% (2419/2419), done.

Fri Jun 9 08:41:20 PDT 2017 - done with exit-code=0



# [**Photon NTP**](https://confluence.eng.vmware.com/display/vDR/Photon+NTP)

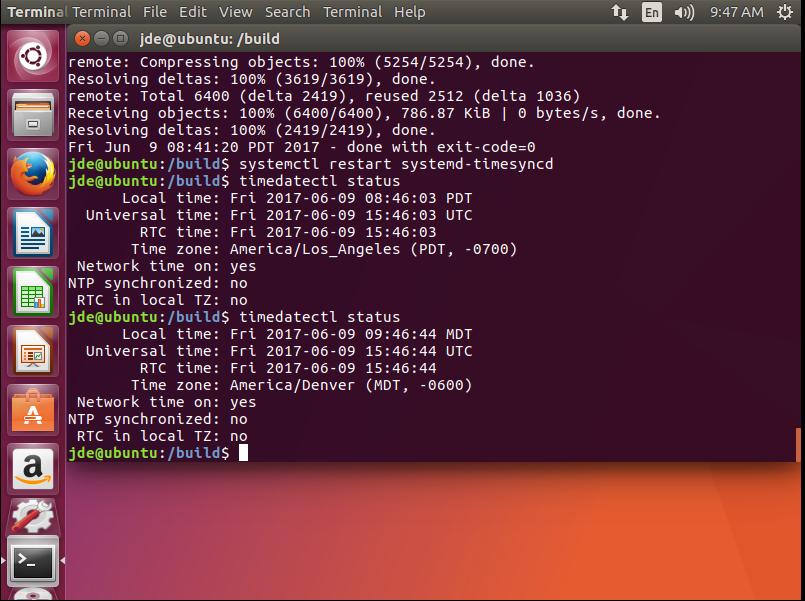
<https://confluence.eng.vmware.com/display/vDR/Photon+NTP>

NTP is controlled via Systemd Timesync

Configured via /etc/systemd/timesyncd.conf

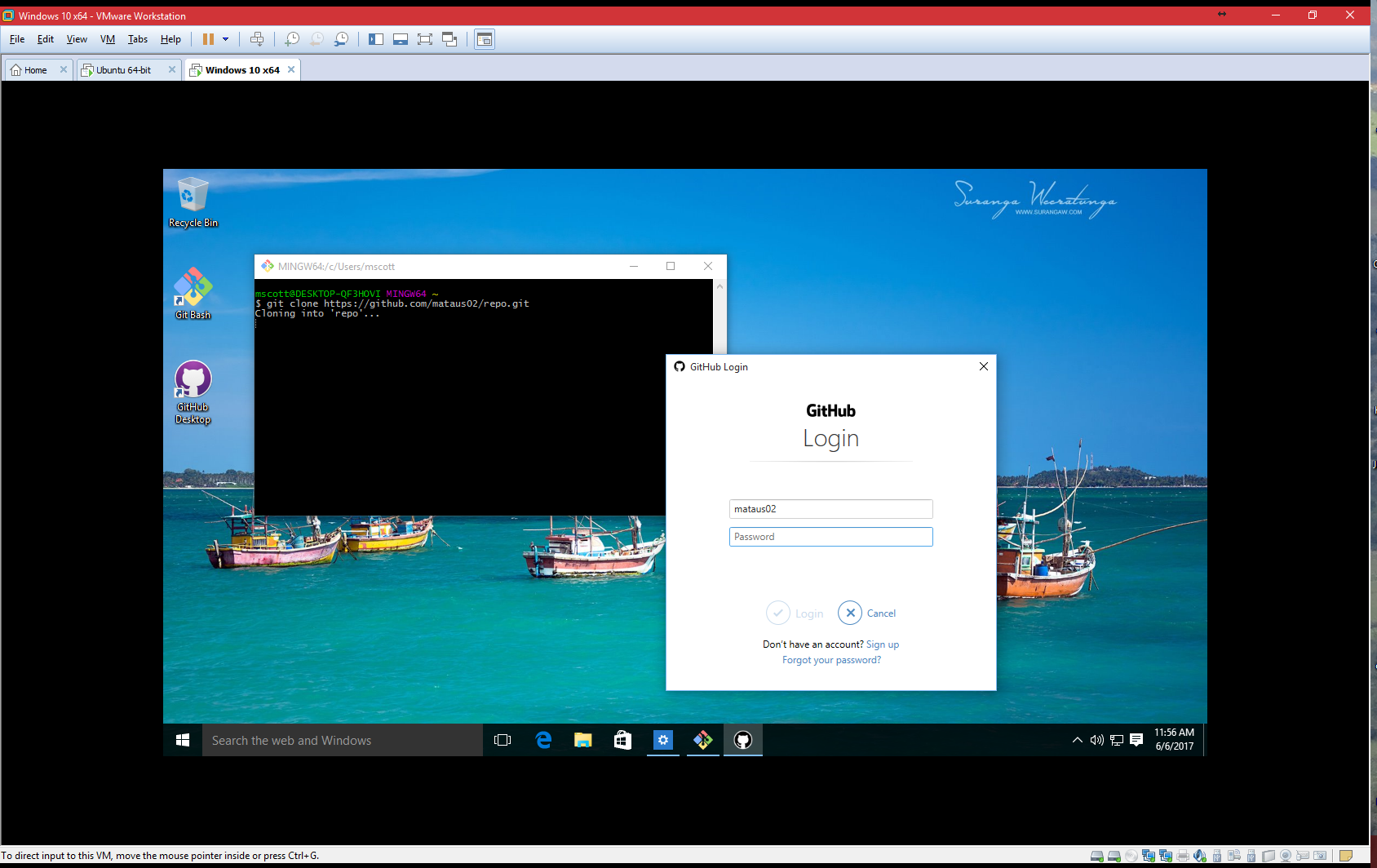
Restart Service

|  |
| --- |
| systemctl restart systemd-timesyncd    root@ova-dhcp [ ~ ]# timedatectl status        Local time: Mon 2016-06-20 09:14:08 UTC    Universal time: Mon 2016-06-20 09:14:08 UTC          RTC time: Mon 2016-06-20 09:14:08         Time zone: UTC (UTC, +0000)   Network time on: yes  NTP synchronized: yes   RTC in local TZ: no  root@ova-dhcp [ ~ ]# |



Adding this towards the Top for DASH Usage

<https://dash.eng.vmware.com/#/report/nimbus/usage?user=mscott>



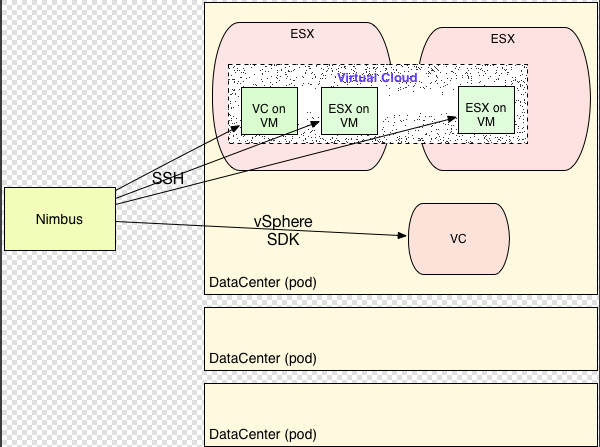
## What is Nimbus?

As a core VMware product, many Devs and QEs work on vSphere. Both need a testing environment for VC, ESX, etc. To run a test case with one VC controlling 10 ESX, a VC Dev would need 11 physical servers. Hundreds of developers with similar server requirements would obviously be unaffordable from cost, space, power, and network bandwidth perspectives. In addition, the time to install and provision 11 physical servers would lead to very low work productivity for each developer.

Nimbus solves this problem. The basic idea is to setup a vSphere testing environment on vSphere. The key points are:

* Both VC and ESX can be installed on a virtual machine.
* vSphere provides an API to enable third party software (Nimbus) to create/edit/delete virtual machines in a datacenter, which can be thought of as a cloud.
* As a cloud software provider, VMware has a number of internal clouds for R&D.

The following diagram shows how Nimbus works:



* There are a number of datacenters (Nimbus calls them **pods**) that Nimbus can use.
* Nimbus uses a load balance mechanism to pick a pod.
* Nimbus connects to the VC in the pod using RBVmomi. RBVmomi is a Ruby implementation of vSphere SDK. (see its dumped interfaces: [ToolsEng/Projects/Nimbus/RbVmomi\_Interfaces](https://confluence.eng.vmware.com/ToolsEng/Projects/Nimbus/RbVmomi_Interfaces))
* Nimbus creates the virtual machine(s) on the pod using vSphere API.
* After the virtual machine(s) are created, Nimbus runs test scripts on them over SSH. Then it collects the results and output and sends it back to the user.

[ToolsEng/Projects/Nimbus/UserGuide](https://confluence.eng.vmware.com/ToolsEng/Projects/Nimbus/UserGuide) provides additional information about Nimbus.

Nimbus comprises as a set of independent commands that are implemented with shell and Ruby scripts. A typical Nimbus command looks like this:

|  |
| --- |
| # nimbus-esxdeploy vmname ob-131234 |

Don't be confused with syntax of this command, it's only a sample.

Anurag provided the following statistics for the week of 11/11/2013, showing the number of Nimbus-deployed virtual machines:

* 166K in total
* 77k ESX
* 40K VC
* 17k Nimbus-workers
* 12K ISCSI
* 7K NSF
* Little bit of others

There are around 1 million VMs a month!

## A Hands-on Introduction to the Nimbus Gateway

The Nimbus gateway is a server that contains the latest complete Nimbus installation, including all scripts, library dependencies, toolchain, and mounts. <i>It is important to note that the Nimbus gateway is only a place to issue Nimbus commands; virtual machines are created on **pods**.

Use your standard VMware account credentials to ssh to the nimbus gateway:

|  |
| --- |
| $ ssh yourname@nimbus-gateway.eng.vmware.com |

Before trying to deploy an instance of ESX, you need to get a build number from buildweb. Go to: <https://buildweb.eng.vmware.com/login/> and login with your VMware credentials. For this example, on the "Released" menu, select "Released Builds". Under Product, scroll down to select "server (VMware ESX Server )". Under Version, select "5.5". Under Milestone, select "GA". Click "Search". Pick one of the builds, in this case "1331820".

Then run the nimbus-esxdeploy command on nimbus-gateway:

|  |
| --- |
| chaol@nimbus-gateway:~$ /mts/git/bin/nimbus-esxdeploy chaol-try 799733  2013-11-17 23:03:35 -0800: deploying build 799733 to /mts-pek/home/chaol/public\_html/pxe/799733   ......  2013-11-17 23:03:39 -0800: creating VM chaol-try ...  2013-11-17 23:03:41 -0800: vm hw version: vmx-09  2013-11-17 23:03:41 -0800: mac: [&amp;quot;00:50:56:86:3d:80&amp;quot;]  2013-11-17 23:03:41 -0800: configuring PXE servers in rdinfra3...   ......  Configuration complete.  2013-11-17 23:04:01 -0800: powering on VM...  2013-11-17 23:04:05 -0800: VM got placed on host: pa-rdinfra3-aesx197.eng.vmware.com  2013-11-17 23:04:05 -0800: Waiting for ESX to boot - done  2013-11-17 23:08:50 -0800: Determined IP address: 10.67.198.146, type: guestIp  2013-11-17 23:08:50 -0800: To manage this VM use &amp;quot;NIMBUS=rdinfra3-vc6-1 /mts/git/bin/nimbus-rvc&amp;quot;  2013-11-17 23:08:50 -0800: You can kill the VM with &amp;quot;NIMBUS=rdinfra3-vc6-1 /mts/git/bin/nimbus-ctl kill chaol-try&amp;quot;  2013-11-17 23:08:50 -0800: done |

A virtual machine named chaol-try has been created.

Before running the nimbus-rvc command, you must set the environment variable NIMBUS to specify the pod,  rdinfra3-vc6-1. The previous nimbus-esxdeloy command execution has already chosen the pod to create the virtual machine. The VM chaol-try will only be found on pod rdinfra3-vc6-1. Note that when you run nimbus-esxdeloy, a different pod might be chosen, so check the output of nimbus-esxdeploy.

Then run the nimbus-rcv command, which will allow you use "vm.ip" to check the ip address of the VM:

|  |
| --- |
| chaol@nimbus-gateway:~$ NIMBUS=rdinfra3-vc6-1 /mts/git/bin/nimbus-rvc  Welcome to RVC. Try the &amp;apos;help&amp;apos; command.  0 chaol-try: poweredOn  /rdinfra3-compute-vc6.eng.vmware.com/RDinfra3 VC6/vm/users/chaol&amp;gt; vm.ip chaol-try  chaol-try: 10.67.198.146  /rdinfra3-compute-vc6.eng.vmware.com/RDinfra3 VC6/vm/users/chaol&amp;gt; exit |

The last step is to delete the virtual machine:

|  |
| --- |
| chaol@nimbus-gateway:~$ NIMBUS=rdinfra3-vc6-1 /mts/git/bin/nimbus-ctl kill chaol-try |

## Developer Environment

### Setup

Nimbus can be developed and tested on a local laptop. For developers who use a Macbook Pro, refer to [Setup Nimbus on Macbook](https://confluence.eng.vmware.com/ToolsEng/Projects/Nimbus/Development/SetupOnMacbook).

Sample workflows are found:

* [Nimbus/Development#Example\_Development\_Workflows](https://confluence.eng.vmware.com/Nimbus/Development#Example_Development_Workflows)
* [Fbelemezov/GitWorkflow](https://confluence.eng.vmware.com/Fbelemezov/GitWorkflow)

### Reviewboard

Review board is used to generate a code review package before you check in your code changes to GIT. Go to <https://reviewboard.eng.vmware.com/account/login/>. You should be able to login to Reviewboard with your standard VMware account. However, many new hires found they couldn't login and had to open a Bugzilla ticket for help. If you cannot login, use the following information in Bugzilla:

* Product - "internal"
* Category - "Review Board"
* Component - "other"

### Bugzilla

Bugzilla is where Nimbus user to report bug or look for help. Go to <http://bugzilla.eng.vmware.com>, and login with your VMware credentials.

**To subscribe to email notification of Nimbus bugs**

From the "My Account" menu, select "Field Subscriptions". In the "Products, Categories and Components" tab, expand the "Internal" tree, and go to "Nimbus". On the "Nimbus" row, check the second radio box (Subscription Type to see all Nimbus Changes), then click on "Submit Changes" button.

From the "My Account" menu, select "Email Preferences". Check all check-boxes in the "Email me when ..." table, and uncheck all check-boxes in the "But don' t email me when ..." table, then click on "Submit Changes" button. Later on after you get familiar with the Bugzilla system, you may tune your settings on this page.

**To query Nimbus bugs**

From the "My Account" menu, select "Saved Searches". If your "My Saved Searches" table is empty, you may import searches from other developers. In "View Another User's Saved Searches", type in "chaol" or "pkatyal" to get a list of saved searches. Click "Add" to import each search you are interested in to your own saved list, then click "Submit Changes". You can either run a search from the "Saved Searches" tab by selecting "Run", or you can directly access any of your saved searches from the top-level dropdown menu, "Saved Searches".

## Peer systems: CAT, BAT, SVS, etc. (todo)

* What's them?
* How to access them?
* How Nimbus cooperates with them?

### Continuous Automated Testing (CAT)

The CAT ([ToolsEng/Projects/CAT](https://confluence.eng.vmware.com/ToolsEng/Projects/CAT)) system is a framework to continuously run automated tests. When a CAT test fails, it is relatively easy to identify the changeset that caused the failure. This increases branch stability by identifying bad checkins quickly, so that they can be fixed or backed out within a specific period of time.

* CAT continuously produces product builds to validate source code changes and gate official builds.
* CAT initiates tests to verify stability and provide reliability status reports.
* CAT tracks issues that are integrated into Bugzilla.
* CAT results are stored and easily viewed. Branch stability can be viewed at the changeset level.

Builder and Testers

These are two important concepts in CAT. A builder creates builds. It is a software application/process that monitors the source code repository of a project for new changesets. When one is detected it will initiate a build using this change set. The tester is a software application that performs tests on new builds that are installed on test machines. The tester will monitor the test and send email notification when it fails.

REST API

CAT is using REST API, through which Nimbus talks to it.

Add tester on INFRA-CAT([1](http://vim-cat.eng.vmware.com/)):

Login first with your vmware account to make sure the admin panel is enabled. 1. Add a workload (set Skipboot to true if no target machine required). 2. Add a new area to specify the test sets (or use a existing one). 3. Add a new tester (owner: infra-cat-spam, machines list can be empty only if Skipboot is true). 4. After the tester is submitted, associate it with the previous workload by adding a "work unit". 5. Start the tester. (the option "stop gracefully" means stops the tester after it finishes, when "stop" means stop it immediately).

URLs

* <http://cat.eng.vmware.com> - For ESXi testing
* <http://vim-cat.eng.vmware.com> - VC/VCVA testing
* <http://infra-cat.eng.vmware.com> - For devtools internal testing

Please refer to CAT's User Guide([ToolsEng/Projects/CAT/UserGuide](https://confluence.eng.vmware.com/ToolsEng/Projects/CAT/UserGuide)) for details.

### BAT

Put your name here, then start your contribution.

### Submit Verification System (SVS)

SVS (ToolsEng/Projects/SVS) pushes code submission through a pipeline to validate, build, and test, before it is committed to SCM.

SVS versus CAT:

* SVS is pre-checkin, preventing easy-to-catch errors from going forward (e.g. build breakages, sanity tests)
* [CAT](https://confluence.eng.vmware.com/CAT) is post-checkin, bringing value with a large battery of tests and the coverage it offers.
* SVS handles patches individually (either the base or the patch can be blamed for failure)
* [CAT](https://confluence.eng.vmware.com/CAT) may not test every changeset (which changeset should be blamed in case of failure?)

Please refer to SVS's User Guide([ToolsEng/Projects/SVS/UserGuide](https://confluence.eng.vmware.com/ToolsEng/Projects/SVS/UserGuide)) for details.

### Any else?

Put your name here, then start your contribution.

## Useful References

* [Nimbus Homepage](https://confluence.eng.vmware.com/ToolsEng/Projects/Nimbus) - Nimbus Project homepage.
* [Nimbus User Guide](https://confluence.eng.vmware.com/display/DevToolsDocKB/Nimbus+User+Guide)
* [Nimbus Index](https://confluence.eng.vmware.com/ToolsEng/Projects/Nimbus/Nimbusindex) - This page contains almost everything about Nimbus.
* [vSphere API Reference](http://pubs.vmware.com/vsphere-50/index.jsp?topic=%2Fcom.vmware.wssdk.apiref.doc_50%2Fright-pane.html)
* [Safari Online Bookstore](http://premium.safaribooksonline.de/home) - Use your VMware email to register, then you will automatically be granted to read all books.
* <https://help.github.com/articles/changing-a-remote-s-url/#switching-remote-urls-from-ssh-to-https>
* <https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/#generating-a-new-ssh-key>
* <https://guides.github.com/activities/hello-world/>
* <https://gituser.eng.vmware.com/git/manage/mscott/>
* <https://4.bp.blogspot.com/-AxaLzVO0rkI/V-u164mX_ZI/AAAAAAAABec/RLhtEhsWAmgOBN9pNOWBEY5BRWNICxG5QCLcB/s1600/NSX_Poster.png>
* <https://confluence.eng.vmware.com/pages/viewpage.action?spaceKey=vDR&title=Nimbus+vCAv+Testbeds>
* <https://git.eng.vmware.com/>