|  |
| --- |
| **Install and configure local YUM Server on CentOS 7**  **Step 1 – Mount CentOS 7 media**   * Mount the local media like DVD, USB stick etc that contains CentOS 7 / Oracle Linux 7 / RHEL 7 etc. * Here we used the CentOS 7 DVD and mount it. Here we mount DVD media to “/mnt” directory                #mount -t iso9660 -o loop /dev/sr0 /mnt  **Step 2 – Copy media content to the Server**   * Before copying media, we will create a folder inside the server root directory.               #mkdir /localrepo   * Now copy media to the created folder.               #cp -rv /mnt/\* /localrepo/  **Step 3 – Configure the Local Repository**   * Take a backup of the repository folder.               cd /etc   * Backup repository folder.               cp -r yum.repos.d yum.repos.d-bak   * Delete all online repository files.               rm -rf yum.repos.d/\*   * Create locate repository file.                vim yum.repos.d/local.repo   * Add the following line to the file for Centos 7.               [centos7]              name=centos7              baseurl=file:///localrepo/              enabled=1              gpgcheck=0   * Then save and exit the file. For your information, here is what each of the items means in the above command.   [centos7] – Name of the Section.  name = Name of the repository  baseurl = Location of the package  Enabled = Enable repository  gpgcheck= Enable secure installation  gpgkey = Location of the key  gpgcheck is optional (If you set gpgcheck=0, there is no need to mention gpgkey)   * Now update the local repository.   #createrepo /localrepo/    Create Repo   * Now enable the local repository.   #yum clean all  Yum Clean All   * List repository   #yum repolist all |

|  |
| --- |
| **How To Open A Port In CentOS / RHEL 7**  A TCP/IP network connection may be either blocked, dropped, open, or filtered. These actions are generally controlled by the IPtables firewall the system uses and is independent of any process or program that may be listening on a network port.  This post will outline the steps to open a port required by an application. For this example, we will be opening Application Specific (jenkins/http-alt) Port 8080.  Server details are as below:  #uname -a  Linux geeklab 3.10.0-693.17.1.el7.x86\_64 #1 SMP Thu Jan 25 20:13:58 UTC 2018 x86\_64 x86\_64 x86\_64 GNU/Linux  #cat /etc/redhat-release  CentOS Linux release 7.4.1708 (Core)   1. Check Port Status   Check that the port is not open and HTTP is not showing that port:  # netstat -na | grep 8080  # lsof -i -P |grep http  httpd     5823   root    4u  IPv6  42212      0t0  TCP \*:80 (LISTEN)   1. Check Port Status in iptables   Check that iptables are not showing that port open:  # iptables-save | grep 8080   1. Add the port   Add the test port in /etc/services file and allow the port to accept packets. Test port can be added by editing /etc/services file in below format:  # vi /etc/services  service-name  port/protocol  [aliases ...]   [# comment]  # vi /etc/services  Testport   8080/tcp   # Application Name   1. Open firewall ports   Add Firewall rule to allow the port to accept packets:  # firewall-cmd --zone=public --add-port=8080/tcp --permanent  success  # firewall-cmd --reload  success  # iptables-save | grep 8080  -A IN\_public\_allow -p tcp -m tcp --dport 55555 -m conntrack --ctstate NEW -j ACCEPT   1. Check newly added port status   After adding the port for httpd and reloading httpd services, notice now httpd is also listening to newly added port 8080:  # lsof -i -P |grep http  httpd     6595   root    4u  IPv6  43709      0t0  TCP \*:80 (LISTEN)  httpd     6595   root    6u  IPv6  43713      0t0  TCP \*:8080 (LISTEN)  # netstat -na |grep 8080  tcp6       0      0 :::8080                :::\*                    LISTEN |

|  |
| --- |
| **Boot RHEL 7 / CentOS 7 Server in Single User Mode**  For Linux system admins, booting RHEL 7 / CentOS 7 servers in single user mode is the most common day to day activity. Single user mode is considered as maintenance or emergency mode where we can perform our troubleshooting steps. Following are the scenarios where we need to boot our RHEL / CentOS Servers in Single user mode:   * **Reset Root password** – There can be some scenarios where someone from the team has reset the root password and he/she didn’t share it with team and left the organization * **Repairing file system** – There are some situations where OS(operating system) file systems got corrupted and while rebooting server didn’t come up, so in that case we have to repair file system by entering into single user mode. * **Fix Wrong entry of fstab file** – Let’s suppose I have created one file system on RHEL / CentOS Server and update the /etc/fstab file, but there was some typo while updating the file system entries. I got the request to reboot the server, so while booting up it got stuck while mounting that file system. So to resolve and fix such issues, we need to boot Linux servers in Single User mode. * **Disabling / enabling Service** – There can be some service which is taking a lot of time or delaying the boot. So in that case we can disable that service from single user mode. On the contrary we can also enable specific service from single user mode.   There are two methods through which we can boot CentOS 7 / RHEL 7 servers in Single User Mode.    **Method 1:**  Step1: Reboot your Server and go to Grub boot loader menu and choose the appropriate kernel, example is shown below:  RHEL-7-Grub-Menu-screen  Step 2: Press ‘e’ and go to the end of line which starts with ‘linux16’ word.  Grub-Menu-Linux16-RHEL7  Type “rd.break” at end of line which begins with linux16 and then press “ctrl+x”  rd-break-grub-menu-rhel7  In the next window we will get single user mode or emergency mode, something like below:  emergency-mode-rhel7  Step 3: Now remount the /sysroot in ‘rw’ mode  switch\_root:/# mount -o remount,rw /sysroot  switch\_root:/# chroot /sysroot  Let’s assume we want to reset the root password. Run the below commands:  sh-4.2# echo “New-root-password” | passwd --stdin root  sh-4.2# touch /.autorelabel  Reset-Root-Password-RHEL7-single-User-Mode  Now you can edit the files and scan the corrupted file system with fsck command. Once you are done with troubleshooting Type ‘exit‘ command two times or ‘reboot -f‘ to reboot your server.    **Method 2:**  Step 1: Reboot the server and go to the grub menu and select the appropriate kernel  Grub-Menu-CentOS7  Step 2: Press ‘e’ and go the line with starts with ‘linux16’  Replace “ro” with “rw init=/sysroot/bin/sh”  replace-ro-with-rw-init-sysroot-centos7  Once done with the changes press ‘Ctrl+x’  emergency-mode-CentOS7  Step 3: Mount the root file system with chroot command  :/# chroot /sysroot  Let’s assume I want to disable network manager service and rest root password, run the following commands:  :/# systemctl disable NetworkManager  :/# echo “Enter-New-Root-password” | passwd --stdin root  :/# touch /.autorelabel  Once you are done with the changes, type “reboot -f” command to reboot your server  reset-root-password-disable-service-single-user-mode-centos7  Note: In production environment, Single user mode is also password protected. By default root password is the single user mode password on CentOS 7 / RHEL 7 Servers. |

|  |
| --- |
| **SSH Passwordless Login Using SSH Keygen in 5 Easy Steps**  SSH (Secure SHELL) is an open source and most trusted network protocol that is used to login into remote servers for execution of commands and programs. It is also used to transfer files from one computer to another computer over the network using secure copy (SCP) Protocol.  In this article we will show you how to setup password-less login on RHEL/CentOS 7.x/6.x/5.x and Fedora using ssh keys to connect to remote Linux servers without entering password. Using Password-less login with SSH keys will increase the trust between two Linux servers for easy file synchronization or transfer.  If you are dealing with a number of Linux remote servers, then SSH Password-less login is one of the best ways to automate tasks such as automatic backups with scripts, synchronization files using scp and remote command execution.  In this example we will setup SSH password-less automatic login from server 192.168.0.12 as user tecmint to 192.168.0.11 with user sheena.  **Step 1: Create Authentication SSH-Kegen Keys on – (192.168.0.12)**  First login into server 192.168.0.12 with user tecmint and generate a pair of public keys using the following command.  [tecmint@tecmint.com ~]$ ssh-keygen -t rsa  Generating public/private rsa key pair.  Enter file in which to save the key (/home/tecmint/.ssh/id\_rsa): [Press enter key]  Created directory '/home/tecmint/.ssh'.  Enter passphrase (empty for no passphrase): [Press enter key]  Enter same passphrase again: [Press enter key]  Your identification has been saved in /home/tecmint/.ssh/id\_rsa.  Your public key has been saved in /home/tecmint/.ssh/id\_rsa.pub.  The key fingerprint is:  5f:ad:40:00:8a:d1:9b:99:b3:b0:f8:08:99:c3:ed:d3 tecmint@tecmint.com  The key's randomart image is:  +--[ RSA 2048]----+  |        ..oooE.++|  |         o. o.o  |  |          ..   . |  |         o  . . o|  |        S .  . + |  |       . .    . o|  |      . o o    ..|  |       + +       |  |        +.       |  +-----------------+    **Step 2: Create .ssh Directory on – 192.168.0.11**  Use SSH from server 192.168.0.12 to connect server 192.168.0.11 using sheena as user and create .ssh directory under it, using following command.  [tecmint@tecmint ~]$ ssh sheena@192.168.0.11 mkdir -p .ssh  The authenticity of host '192.168.0.11 (192.168.0.11)' can't be established.  RSA key fingerprint is 45:0e:28:11:d6:81:62:16:04:3f:db:38:02:la:22:4e.  Are you sure you want to continue connecting (yes/no)? yes  Warning: Permanently added '192.168.0.11' (ECDSA) to the list of known hosts.  sheena@192.168.0.11's password: [Enter Your Password Here]  **Step 3: Upload Generated Public Keys to – 192.168.0.11**  Use SSH from server 192.168.0.12 and upload new generated public key (id\_rsa.pub) on server 192.168.0.11 under sheena‘s .ssh directory as a file name authorized\_keys.  [tecmint@tecmint ~]$ cat .ssh/id\_rsa.pub | ssh sheena@192.168.0.11 'cat >> .ssh/authorized\_keys'  sheena@192.168.1.2's password: [Enter Your Password Here]    **Step 4: Set Permissions on – 192.168.0.11**  Due to different SSH versions on servers, we need to set permissions on .ssh directory and authorized\_keys file.  [tecmint@tecmint ~]$ ssh sheena@192.168.0.11 "chmod 700 .ssh; chmod 640 .ssh/authorized\_keys"  sheena@192.168.0.11's password: [Enter Your Password Here]  **Step 5: Login from 192.168.0.12 to 192.168.0.11 Server without Password**  From now onwards you can log into 192.168.0.11 as sheena user from server 192.168.0.12 as tecmint user without password.  [tecmint@tecmint ~]$ ssh sheena@192.168.0.11 |