**CrowSoft**

**User Guide**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Affected Sections(s)** | **Author** |
| 1.0 | 17/3/2019 | Initial Draft | Colin Kenny |
| 1.1 | 18/03/2019 | Amended Connection | Michael McFadden |
| 1.2 | 23/3/2019 | Jenkins/github config | Colin Kenny |
| 1.3 | 27/3/2018 | Added GIT Guide | Colin Kenny |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[VMware Guide 3](#_Toc4623288)

[Prerequisites 3](#_Toc4623289)

[Connection 3](#_Toc4623290)

[Set up 3](#_Toc4623291)

[Creating a VM 5](#_Toc4623292)

[Installing Operating system 7](#_Toc4623293)

[Cloning Ubuntu inside VMware ESXi 9](#_Toc4623294)

[Jenkins User Guide 12](#_Toc4623295)

[Connecting Jenkins to github 12](#_Toc4623296)

[Configure Jenkins job 15](#_Toc4623297)

[Git User Guide 16](#_Toc4623298)

[Prerequisites 16](#_Toc4623299)

[Set up 16](#_Toc4623300)

[Using Git 16](#_Toc4623301)

[Working Commands 16](#_Toc4623302)

[Training: 18](#_Toc4623303)

[.NET User Guide 19](#_Toc4623304)

[Installing .NET on Ubuntu 16.04 Virtual Machine 19](#_Toc4623305)

# VMware Guide

Steps to install and use VMware

### Prerequisites

User must have been given access by the admin.

User needs to install FortiClient on their machine

<https://forticlient.com/>

## Connection

Connection Name LYIT-CDC Letterkenny

Remote Gateway fg-vpn.lyit-cdc.net

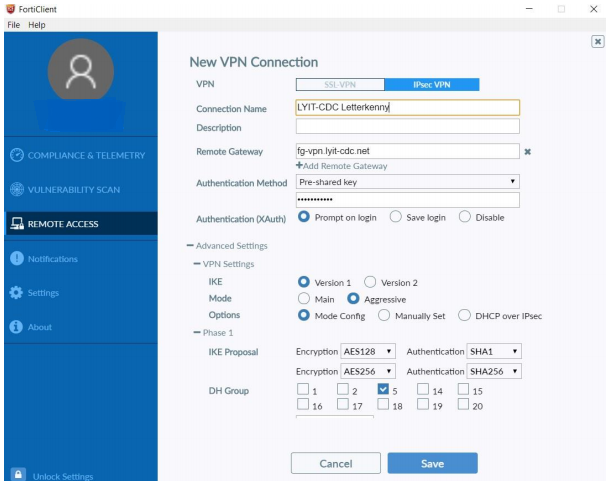
Auth Method Pre-shared Key Link: <pwd\_from \_admin>

CrowSoft IP 172.28.2.46

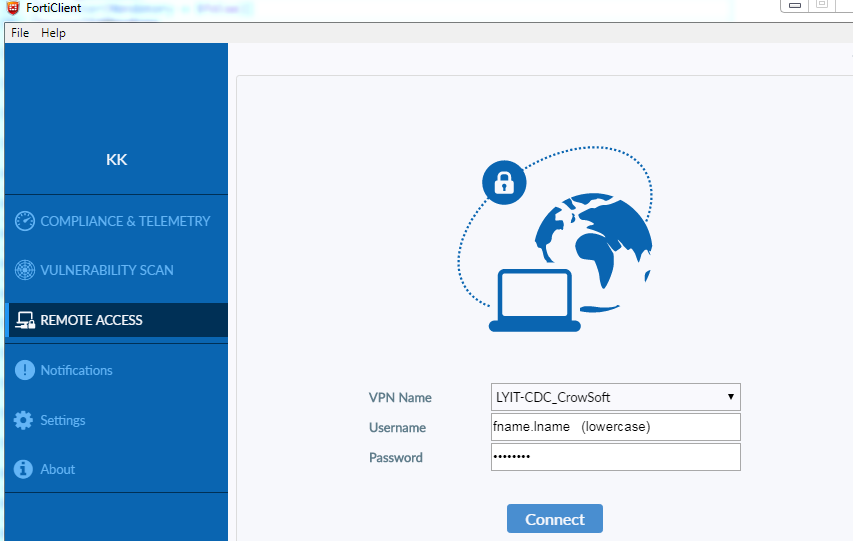
User/password Admin will supply login details

### Set up

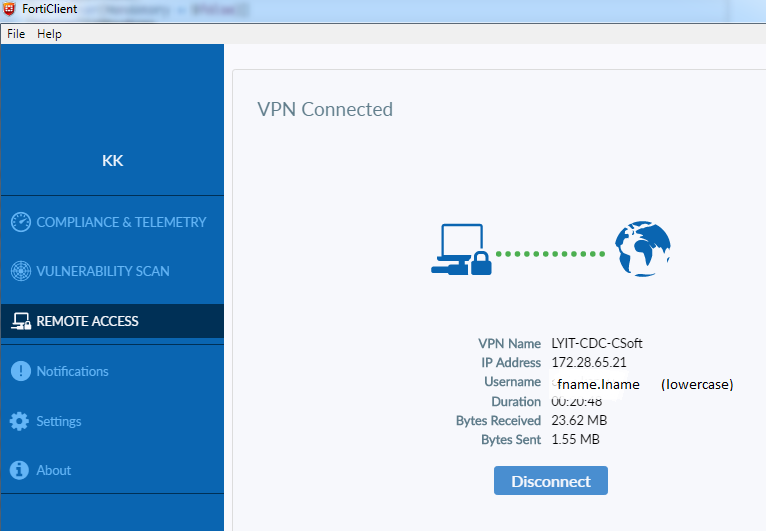
Create a new connection



Connect to LYIT Data Center



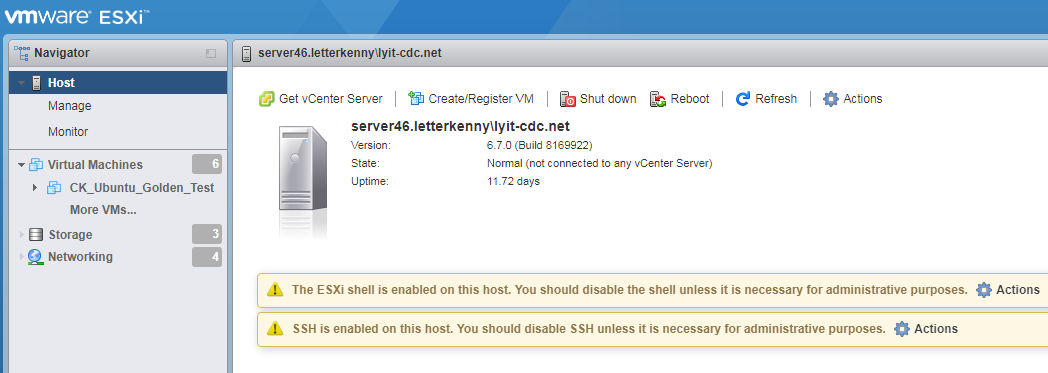
Successful login should display something similar



Enter url and proceed to VMWARE ESXi login entering same login credentials

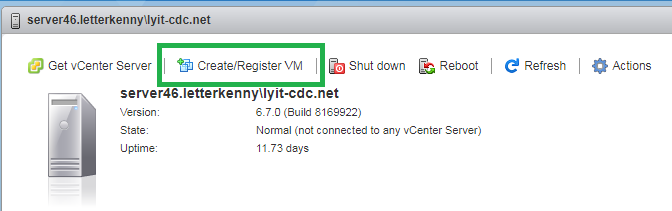
<https://172.28.2.46/>

You are now connected to the LYIT Computing Data Center

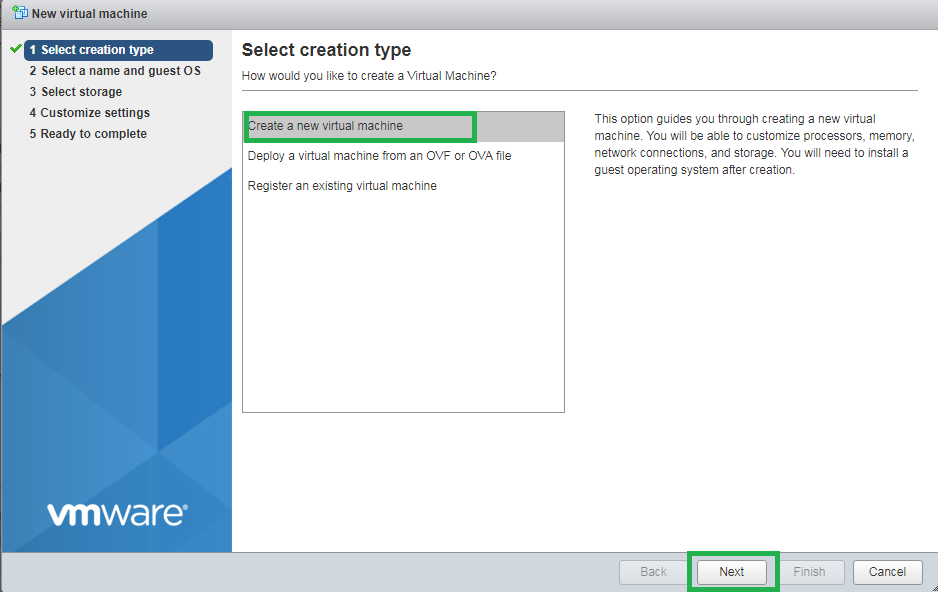


## Creating a VM

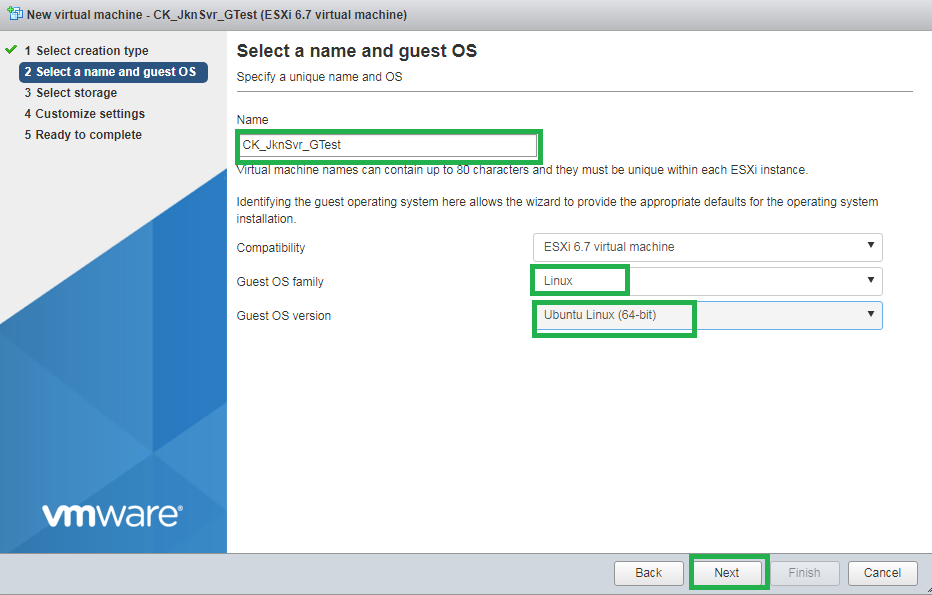
Click on **Create/Register VM** ­­on Host display



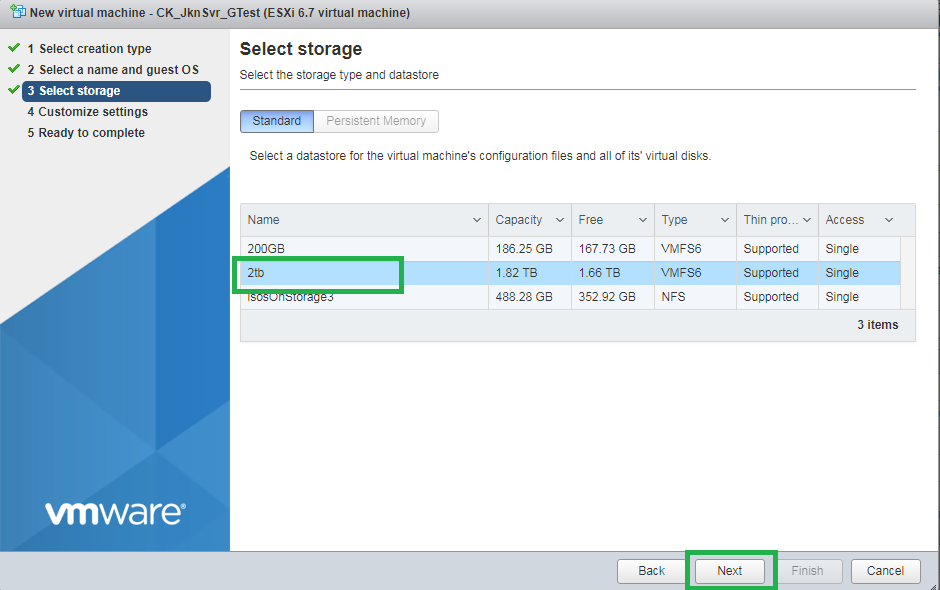
Create a new Virtual machine and click next



Name new VM adding OS & Version you require

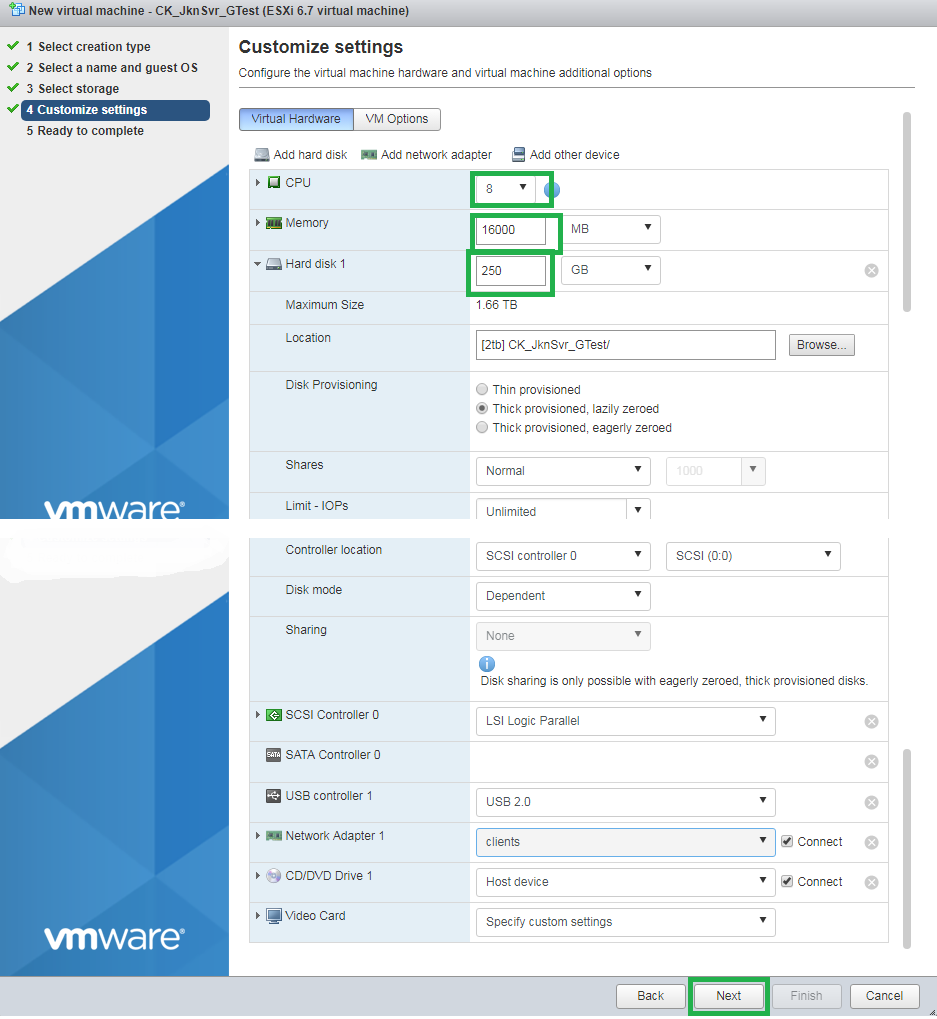


Select Storage option **2tb**



Configure the hardware for the VM click next and Finish

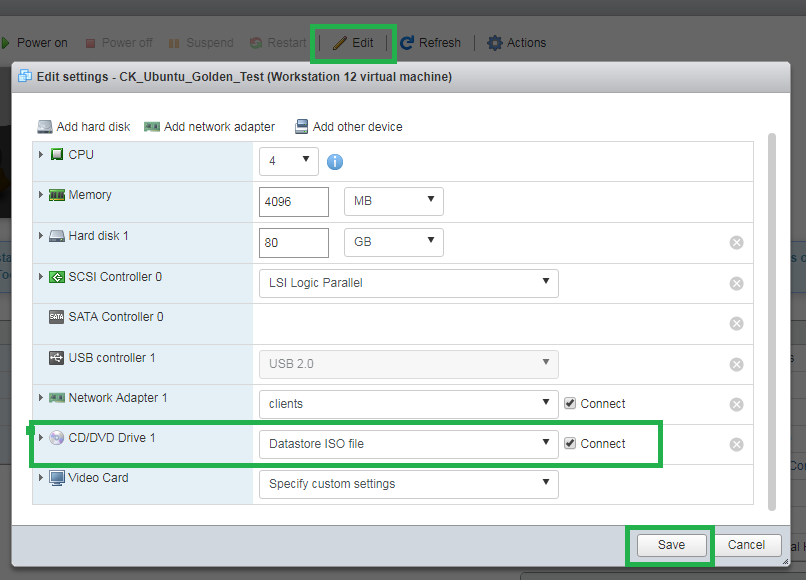
CD/DVD drive can be pointed to Datastore ISO file here (done in next step)



### Installing Operating system

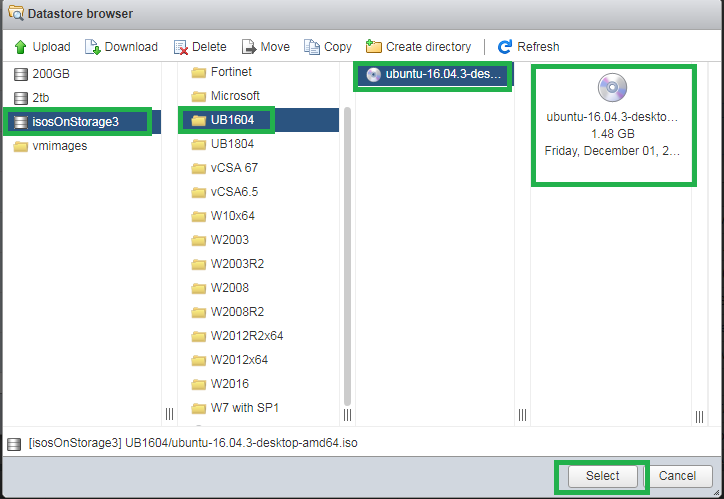
Open newly created VM click on Edit on toolbar at the top

CD/DVD option choose **Datastore ISO file** checking the **Connect** box



Select **isosOnStorage3** choosing you ISO image this example **UB1604**

Click **Select** and **Save**



Power on the VM and proceed to Install OS

Revert CD/DVD setting when install is complete



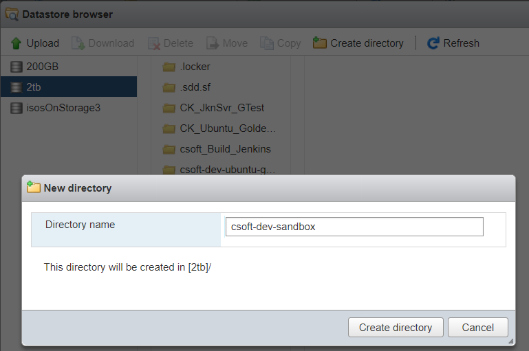
## Cloning Ubuntu inside VMware ESXi

Step1

After logging into VMware ESXi click on the storage icon, then click on the Datastore Browser.

A screenshot of a social media post

Description automatically generated



When it opens up create a new directory where you want to store the cloned VM.

Step 2

A screenshot of a cell phone

Description automatically generatedAfter creating the new directory click on the VM which you want to clone. Right click on the VMX and VMDK files, copy both of these files individually. Note : the last snapshot vmdk has to be also copied . This will take you back out to the first part but just click back into the list of directories.

A screenshot of a social media post

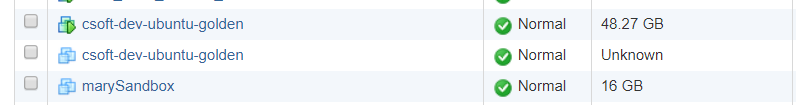
Description automatically generated

Click on the file where you want to paste the copied file. Here you have to click the copy button for it to work. Each file is copied individually and this part takes a few minutes.

Step 3

A screenshot of a cell phone

Description automatically generatedThe new VM must be registered. Right click on the VMX file which has been just copied and Register VM.



Step 4

This will leave two copies of the VM in the dashboard .

A screenshot of a computer

Description automatically generated

Here the cloned VM must be renamed. For the storage to be visible to Ubuntu.

Step 5

A screenshot of a cell phone

Description automatically generatedWhen powering on the cloned machine for the first time a dialog box pops up. Click, I copied it, and Answer

Now your cloned machine is ready to go.

# Jenkins User Guide

Documented steps to install and use Jenkins on Project VM’s.

## Connecting Jenkins to github

To allow Jenkins to connect using SSH client Putty is installed on VM.

Git needs to be installed on the Jenkins server.

* Sudo apt install git

Terminal commands

* sudo apt-get install –y putty
* cd /var/lib/Jenkins - change into Jenkins directory
* sudo su – Jenkins
* mkdir ./.ssh
* cd ./.ssh
* ssh-keygen –t rsa –C “admin@172.28.25.136”
* eval “$(ssh-agent –s)” - returns agent pid
* ssh-add id\_rsa
* cat id\_rsa.pub

**copy generated key into github setting sshkey**

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC/zbZzBy9PzOsn29fLwJUl5PDXplzWgkGBy8Drn8xLCBnw5H8sydii2uwD9+aP4GFegsMEtZG9ogHwr7fUqQWm0YvxCKriKNINgrBXCX9pAwyn3gMw80sSppYEcJsbWlDb42uflKbf80Spea2PfmgoIF9dnvFWauWjt1UNK3rzT6viq49HQ7jBexvA049wZKeVtPnq7LRPUWlQp4OgP+JjYKaOLsUaY7GmLEtmSBxcUbmKnqpOh8d7oAr+DkGblrFifN+zHvjcNRzPi1+r60pMg01ESzI4Z1ecAAIKgY8YOYGqoMhz5cT8dMIdvQBVhOl3oNC85mQjJBQLvNt9jHrd admin@172.28.25.136

* ssh –T [git@github.com](mailto:git@github.com)
* sudo apt-get install openssh-server
* sudo apt install openssh-client
* sudo service ssh status
* sudo nano /etc/ssh/sshd\_config - will display config file

Setting up Firewall and opening ports

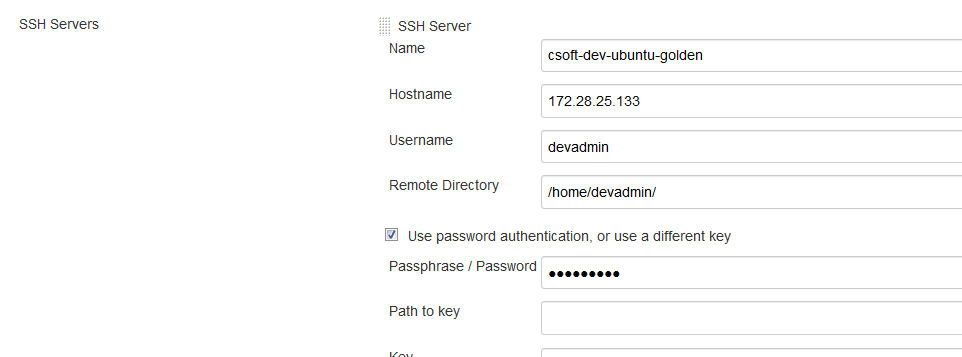
* sudo ufw status
* sudo ufw allow 8080
* sudo ufw allow 22

Web hook needs to be set on github

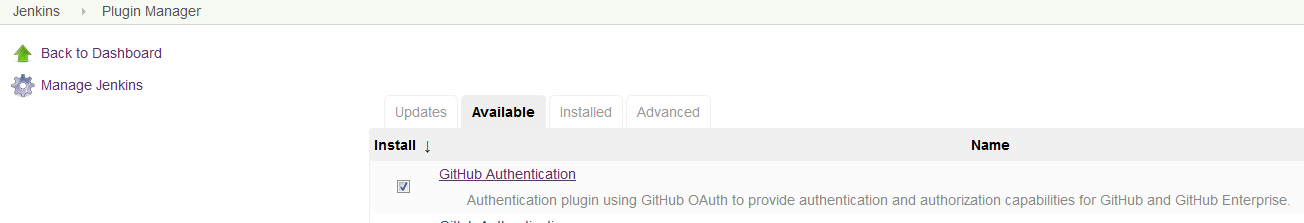
172.28.25.136:8080/github-webhook/

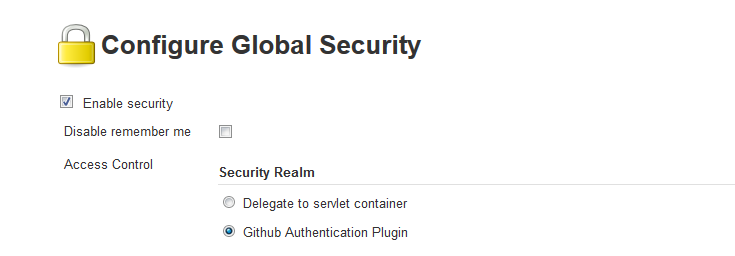
Jenkins

Installed Publish over SSH plugin

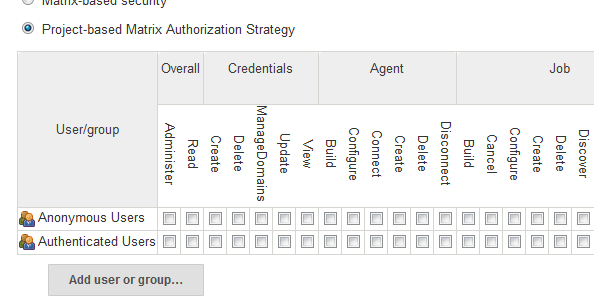


Install GitHub Authentication plugin and restart Jenkins





Authorization: Logged-in users can do anything



Option is available to lock down privileges and create user groups.

## Configure Jenkins job

# Git User Guide

Steps to install and configure git for project.

## Prerequisites

Install git to machine

Link:

<http://git-scm.com/download/win>

### Set up

GitHub project has been created

<https://github.com/rlennon/CrowSoft>

### Using Git

Create a folder for your workspace e.g. C:/CrowSoft

Open Git terminal

Change directory into newly created folder:

Cd CrowSoft

Clone down the repository:

git clone https://github.com/rlennon/CrowSoft.git

##### Set global parameters:

git config --global user.email [L000XXXXX@student.lyit.ie](mailto:L000XXXXX@student.lyit.ie)

git config --global user.name "L000XXXXX"

### Working Commands

git status New/Updated view files will be displayed in red.

git add –all Adds all files to staging area.

git commit -m "Add useful comment" Commits files with comment

git push -u origin master push changes to Master

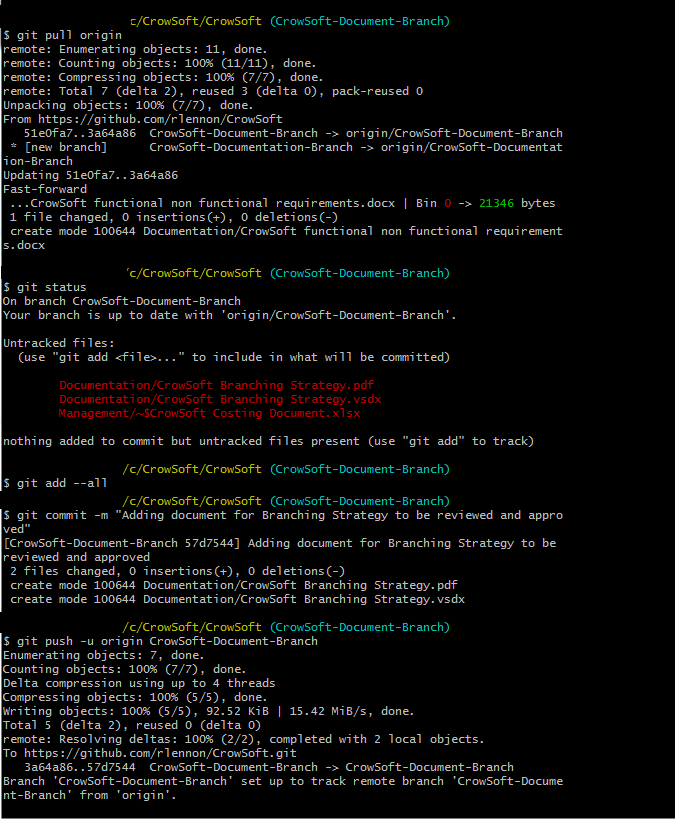
git checkout CrowSoft-Document-Branch Checkout Branch

git push -u origin CrowSoft-Document-Branch Push to Branch - Current branch is the origin

git pull origin Pull to make sure your branch is up to date

rm -rf <folder\_name> Remove Folder

git branch Display current branch



## Training:

<https://www.katacoda.com/courses/git>

<https://www.git-tower.com/blog/git-cheat-sheet/>

<https://datasift.github.io/gitflow/IntroducingGitFlow.html>

# .NET User Guide

## Installing .NET on Ubuntu 16.04 Virtual Machine

* Log into VM using admin credentials.
* Open “Terminal”.
* Type in following commands in order:
  + wget -q https://packages.microsoft.com/config/ubuntu/16.04/packages-microsoft-prod.deb
  + sudo dpkg -i packages-microsoft-prod.deb\*
  + sudo apt-get install apt-transport-https
  + sudo apt-get update
  + sudo apt-get install dotnet-sdk-2.2\*\*

\*Will ask for admin password after this step  
\*\* Click ‘Y’ at prompt

* Once all commands have completed test the installation by checking for the version. Use the command “dotnet --version”. If .NET was installed correctly the terminal will return the version number.
* Close and turn off the VM.
* Take a snapshot of the VM by clicking “Actions” -> “Take Snapshot”. Give the snapshot a name and description and save.

# Connect to Development Server & MySQL Remotely

## How Ubuntu 16.04 was configured

**Please note: This is already configured on the dev server. Do not do this again!**

Edit the mysqld.cnf file, type in the following command



Comment out 127.0.0.1, and add the address 0.0.0.0, and save the file.

****

I installed openssh using the following command: *sudo apt install -y openssh-server*

Check if ssh is running: sudo systemctl status ssh

I added our devadmin admin user as a sudo user, using the following usermod command

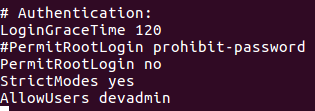


Edit the ssh config file: nano edit /etc/ssh/sshd\_config

Make sure Port 22 is not commented.

Then change the following settings:

* PermitRootLogin no
* AllowUsers devadmin



Save file, then restart ssh server: service ssh reload

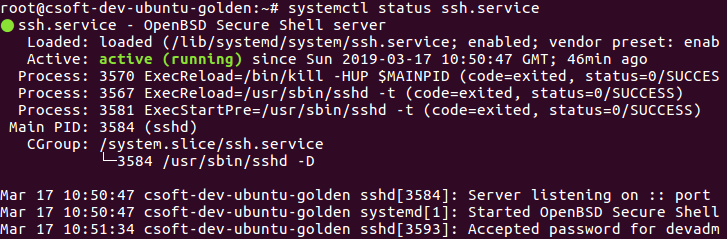


Then restart ssh service: service ssh restart



Check if ssh is running: 

You should get the following status: active (running)



I added a dbadmin user. To log into mysql on Ubuntu, use the following command: mysql -u root -p

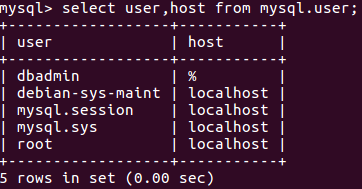


Type the following commands (press enter after each line):

***CREATE USER 'dbadmin'@'localhost';***

***GRANT ALL PRIVILEGES ON \*.\* TO 'dbadmin'@'localhost' IDENTIFIED BY 'refer to password in Slack';***

To check if your user is added, type: ***select user, host from mysql.user;***



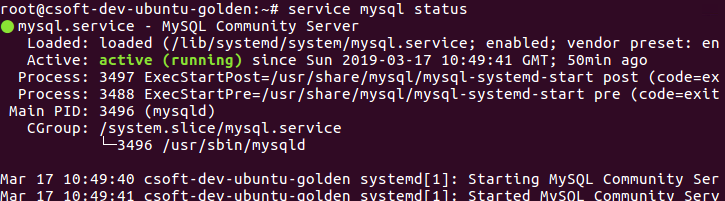
Now you need to restart the mysql service: service mysql reload (this reloads the config file)



Then restart mysql service using the following command:

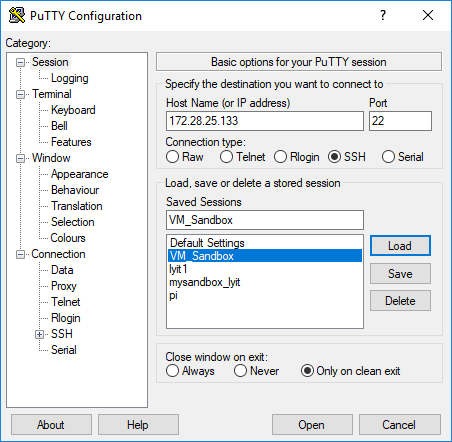


Check the mysql status: service mysql status



## Test connection using Putty

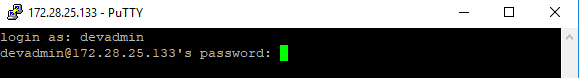
* Open Putty
* IP Address: 172.28.25.133
* Port: 22



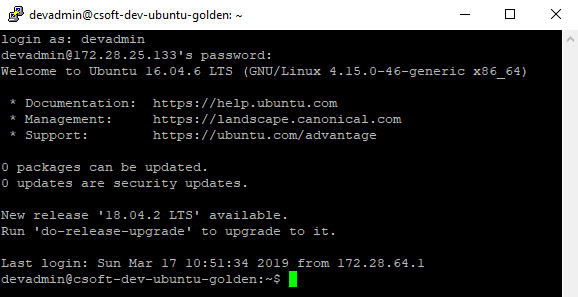
Type in username: devadmin



Type in password: Password available on Slack only



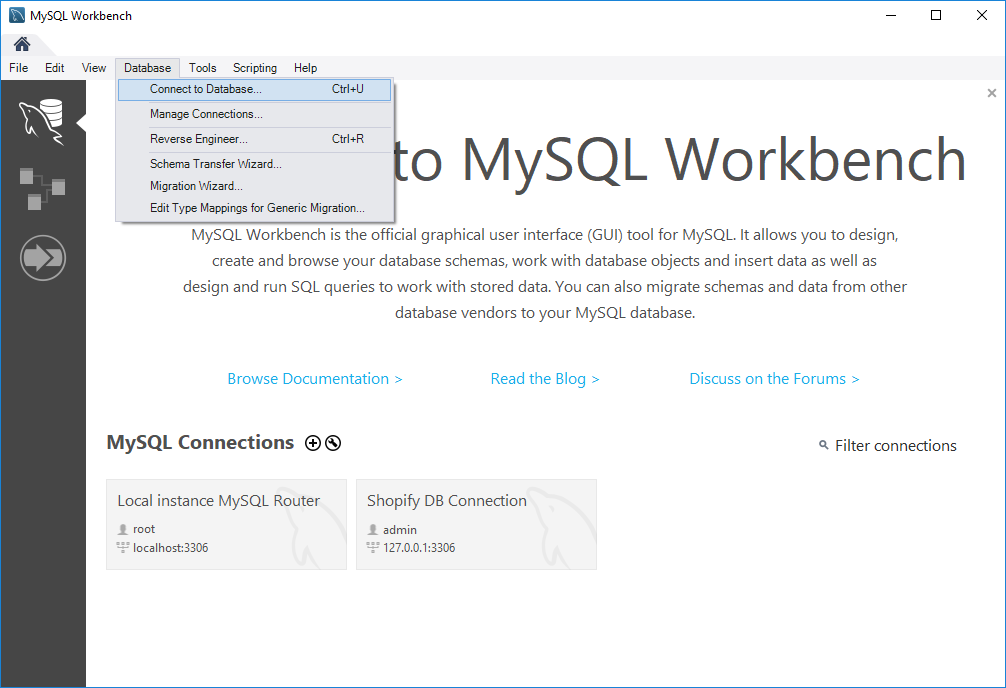
You’re in!



## Log into MySQL Database Remotely

You will need MySQL Workbench if you would like to access the MySQL database from you desktop / home pc. Here is the link (It’s open source): <https://dev.mysql.com/downloads/workbench/>

Open MySQL Workbench, and go to Database, Connect to Database

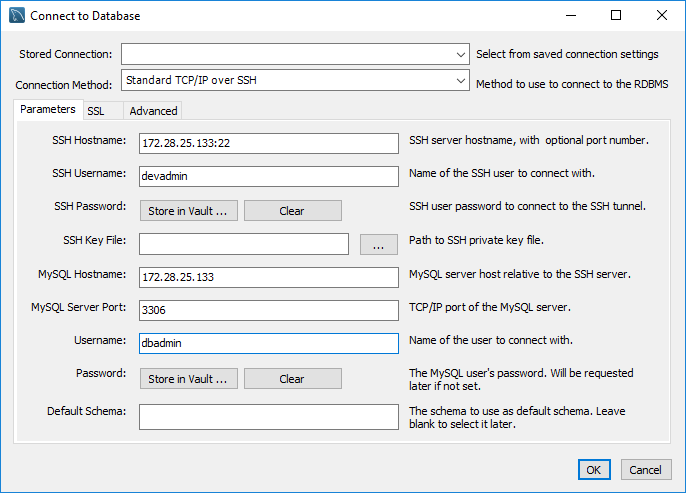


Click on Connection Method and select: Standard TCP/IP over SSH

Change the following settings

* SSH Hostname: 172.28.25.133:22
* SSH Username: devadmin
* SSH Password: refer to dev server password posted on Slack (You can store the password in Vault)
* MySQL Hostname: 172.28.25.133
* MySQL Server port: 3306
* Username: dbadmin
* Password: refer to dev server password posted on Slack (You can store the password in Vault)

Then click on OK



If you see the following screen, you’re in! If not, check your settings. 