Virtual Server Setup

# **Overview**

This worksheet explains how to set up your virtual server. You will use the virtual server as your “live” environment. It’s where you will deploy your applications for grading.

# Create a Virtual Server (VS)

1. Go to the Virtual Server Manager on the department intranet

Go here: [**https://www.doc.gold.ac.uk/intranet**](https://www.doc.gold.ac.uk/intranet) and then click on the **Student Intranet** tab and then **Virtual Servers**

1. Click **New Server** to create a server

Your username will be set for you. Set an admin password (different to your campus password), and choose options as follows:

Graphical user interface, application

Description automatically generated

## **Important: Make sure you remember your password. There is no way to recover your server if you forget it.**

## **If all goes well, you should see the created server listed in the Virtual Server Manager:**

****

# **Connect to your virtual server via SSH**

Next, you will connect to your web server using SSH. This will allow you to run commands on the server. There are a few ways to do this, so choose the method that works best for you.

****Option 1**: From any computer via a web browser**

1. You can use the doc.gold.ac.uk proxy server by typing the following into your browser's address bar:

http://www.doc.gold.ac.uk/ssh/ID/

In the above, replace **ID** with your server ID.

1. You will be prompted for a username and password. Enter the server administrator username and password you used to set up the virtual server.

|  |
| --- |
|  |

This option launches a bash shell in the browser window.

|  |
| --- |
|  |

****Option 2**: From an SSH client (terminal-bash shell) on the college network**

1. Start a Terminal session and enter the following:

*terminal*

ssh USERNAME@myserver.doc.gold.ac.uk -p PORT

In the above, you should replace: **USERNAME** with your username and **PORT** with the SSH port number of your virtual server, which will be 2 followed by your server ID (e.g. 2350).

1. Then enter your****virtual server** password** when prompted.



If all goes well, you should see the Ubuntu welcome message:

Text

Description automatically generated

and the prompt:



****Option 3**: From an SSH client (terminal-bash shell) outside the college network**

Use this option if you are on a computer that is **not** on the college network.

1. Start a command prompt (on Windows) or terminal (on Mac or Linux) and enter the following:

*local terminal*

ssh -t USERNAME@doc.gold.ac.uk myserver ssh ID

In the above, you should replace: **USERNAME** with your username and **ID** with the ID of your virtual server (e.g. 350)

1. Enter your ****campus password****when prompted. (you won't see the password you type for security reasons but it is working!)
2. Then enter your ****virtual serve**r password** when prompted.

Text

Description automatically generated

If all goes well, you should see the Ubuntu welcome message:

Text

Description automatically generated

and the prompt:



# **Install Node**

Now that your server is running, you need to install some software on it!

Create a minimal stack by installing **node** and **npm**. Node is our runtime environment for server-side Javascript. Npm is the package manager, which we can use to set up the modules we need for our applications.

1. Go to the bash shell on your computer. Run these 3 commands to do the install:

*terminal*

sudo apt-get update

sudo apt install nodejs

sudo apt install npm

What does ‘sudo’ do? Do some research.

1. Check the version of the node installed by:

*terminal*

node --version

Here’s what I get:

|  |
| --- |
|  |

What is the version of the Node.js you have installed?

1. This version is quite old! Let’s update to the latest version. Run the following commands (there are 2 commands here!):

*terminal*

wget <https://raw.githubusercontent.com/nvm-sh/nvm/master/install.sh>

bash install.sh

1. Close the ssh session:

*terminal*

exit

1. and open your ssh session again, then run:

*terminal*

nvm install v16.17.0

1. Now check your node version again:

*terminal*

node --version

You should have version 16.17.0 now.

## Install MySQL interactive shell on your virtual server

1. Install MySQL:

*terminal*

sudo apt-get update  
sudo apt-get install mysql-server

## Set your MySql root password

The root user will have super-user rights on the database engine. We will use this to create new databases and new users.

Note: Don’t use the root user to connect from your app to MySql. You should create new users for your apps to use.

1. To start the MySQL shell as the root user:

*terminal*

sudo mysql

You should see the MySQL shell start:

Text

Description automatically generated

1. Set the password for the root user

*terminal - mysql*

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'goldrhb342';

You can change to whatever password you want, but make sure you remember it! No one else will be able to recover it for you.

Then exit the terminal:

exit

## **Create a simple node test app**

Just to prove that node is working, you can create this simple app.

1. Make a directory named hello-node:

*terminal*

mkdir hello-node

1. then change directory to hello-node:

*terminal*

cd hello-node

1. then intialise this directory as a node app:

*terminal*

npm init

you do not need to answer any of the questions asked. Just press ENTER each time.

1. Then start the editor:

*terminal*

nano index.js

1. Enter the following code:

*index.js*

var http = require('http');

http.createServer(function (req, res) {

    res.writeHead(200, { 'Content-Type': 'text/html; charset=UTF-8' });

    res.end('Hello World!');

}).listen(8000, function () {

    console.log('Node server is running...');

});

1. In order to quit nano and save your file:  Press Ctrl+x to exit.  Press Y in response to "Save modified buffer". Then press ENTER in response to "File Name to Write".
2. Then run your node file:

*terminal*

node index.js

As you can see your server is running:

Graphical user interface, text

Description automatically generated

### **Access your server via HTTP**

Now let’s access your web page from a browser!

1. From any network, enter a URL of this format in the address bar:

*browser*

http://www.doc.gold.ac.uk/usr/ID/

In the above, replace ‘**ID**' with your virtual server's ID (for example, 350)

1. Alternatively, from the college network, you can enter a URL in the format:

*browser*

http://myserver.doc.gold.ac.uk:WWW/

In the above, replace ‘WWW' with the number `5', followed by your server's WWW port number. This will be the number 8000 + the server ID. For example, if your server ID is 199, its WWW port number will be 8199. So with the `5', that would be 58199.

Yes, your first web server is running and says hello to the world!

|  |
| --- |
| Graphical user interface, text, application  Description automatically generated |

1. You can stop your server by pressing Control+C in the SSH session.    
     
   Disconnect from your virtual server
2. Make sure you type 'exit' whenever you are done with your ssh session. This will terminate the session and free up resources.

Your code will stay stored on the server for your next visit!

END