

## Detailed guide on installing Mininet and Ryu over Baadal VM:

Created by: Prateek Bhaisora (2023JCS254)

- Setting up Baadal VM

**Step 1:** Request for a Baadal VM and wait for its approval by Faculty and Admin.

- (FYI **Baadal** is the **orchestrator/manager** for IITD's private Cloud System. Currently, you will be renting a Virtual Machine from IITD's Cloud Service via Baadal.)

**Step 2:** Once your request gets accepted, you can see your VM in the **My VMs** tab on Baadal's Homepage.

The screenshot shows the Baadal homepage with a yellow header bar containing the logo 'BAADAL' and 'An Academic Cloud'. Below the header, there are navigation links for 'About', 'FAQ', 'Team Baadal', and 'Contact'. On the right, it says 'Welcome Prateek [ Logout ]'. The main content area has a 'USER MENU' sidebar with links for 'Home', 'Request VM', 'Pending Requests', 'My VMs', 'My Tasks', and 'Mail Admin'. The main panel is titled 'My VMs' and shows a table of VM details. The table has columns for Name, Owner, Private IP, RAM, vCPUs, Status, and Settings. A single row is visible: 'prateek-col334' (Owner: Tarun Mangla), Private IP: 10.17.50.98, RAM: 4.0 GB, vCPUs: 4 CPU, Status: Running, and a Settings icon.

Name	Owner	Private IP	RAM	vCPUs	Status	Settings
prateek-col334	Tarun Mangla	10.17.50.98	4.0 GB	4 CPU	Running	

**Step 3:** Click on the **Wrench and Screwdriver icon** under the **Settings** tab. Now click on the **Vnc icon** to get a **VNC Link** for your **VM**. Note the Private IP address you see here (*We will use it later to SSH into Baadal VM*).

- (FYI, **VNC (Virtual Network Computing)** is a graphical desktop-sharing system that uses the [RFB](#) protocol to remotely control another computer. It transmits the keyboard and mouse inputs from one computer to another, relaying the graphical-screen updates, over a network.)

**USER MENU**

Home  
Request VM  
Pending Requests  
My VMs  
My Tasks  
Mail Admin

**VM Configuration**

Name	HDD	RAM	VCPUs	Operating System	Private IP	Security Domain	Status
prateek-col334	80 GB	4096 MB	4 CPU	Ubuntu 22.04 Server amd64	10.17.50.98	Research	Running

VNC Link : [http://10.7.2.101:6080/vnc\\_auto.html?path=?token=663dfdf5018e95797555](http://10.7.2.101:6080/vnc_auto.html?path=?token=663dfdf5018e95797555)

Purpose : COL334 Assignments

**VM Operations**

Snapshot(s)

Configure Snapshots

**Step 4:** Now login using the credentials you got on mail and change your password, as prompted. (*Please be cautious while setting up the new password. In case you forget your password, you might not be able to reset it and will have to request for a new VM.*)

**Possible Errors:**

- In case, you get a Failed to Connect to Server issue. Simply stop your VM, destroy it and restart it using the **icons in VM Operations** (refer above screenshot). If the issue persists, then mail the Baadal Admin or request for another VM.

**Step 5:** Open a new terminal (or, command prompt) and ssh into the Baadal VM using your user name and Private IP.

ssh username@ip

```

  ssh baadalvm@10.17.50.98
baadalvm@10.17.50.98's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-78-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wednesday 25 September 2024 12:08:34 PM IST

System load: 0.0          Processes:           147
Usage of /: 6.5% of 70.31GB   Users logged in:      1
Memory usage: 7%
Swap usage:  0%          IPv4 address for enp1s0: 10.17.50.98
                           IPv4 address for enp1s0: 169.254.3.216

Expanded Security Maintenance for Applications is not enabled.

205 updates can be applied immediately.
129 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Sep 24 17:25:49 2024 from 10.194.48.116
baadalvm@baadalvm:~$ █

```

**Step 5:** Notice that my Baadal VM is Ubuntu 22.04.2 LTS and comes with pre-installed Python 3.10.12 but no pip3, mininet and ryu.

```
baadalvm@baadalvm:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.2 LTS
Release:        22.04
Codename:       jammy
baadalvm@baadalvm:~$ python3 --version
Python 3.10.12
baadalvm@baadalvm:~$ pip3 --version
Command 'pip3' not found, but can be installed with:
sudo apt install python3-pip
baadalvm@baadalvm:~$ mn --version
Command 'mn' not found, but can be installed with:
sudo apt install mininet
baadalvm@baadalvm:~$ ryu-manager --version
ryu-manager: command not found
baadalvm@baadalvm:~$
```

- (FYI, I ran into some eventlet dependency errors between ryu and eventlet when using python 3.10.12 and some mininet compatibility issues with Python 3.10.12, so we will first downgrade the Python version to 3.9.6 and eventlet to 0.30.2).

- **Setting up Proxy over BaadalVM to enable WAN Access**

**Step 1:** Download the proxy.sh script to setup proxy via CLI. Notice that I used –no-check-certificate flag with wget as otherwise my certificate validation process by the proxy server would fail.

```
wget --no-check-certificate https://csc.iitd.ac.in/uploads/proxy.sh
```

```
baadalvm@baadalvm:~$ wget --no-check-certificate https://csc.iitd.ac.in/uploads/proxy.sh
--2024-09-25 12:39:28--  https://csc.iitd.ac.in/uploads/proxy.sh
Resolving csc.iitd.ac.in (csc.iitd.ac.in)... 10.10.211.211, 2001:df4:e000:29::211
Connecting to csc.iitd.ac.in (csc.iitd.ac.in)|10.10.211.211|:443... connected.
WARNING: cannot verify csc.iitd.ac.in's certificate, issued by 'CN=GlobalSign RSA OV SSL CA 2018,0=GlobalSign nv-sa,C=BE':
  Unable to locally verify the issuer's authority.
HTTP request sent, awaiting response... 200 OK
Length: 1577 (1.5K) [text/x-sh]
Saving to: 'proxy.sh'

proxy.sh                                100%[=====] 1.54K --.-KB/s   in 0s

2024-09-25 12:39:28 (502 MB/s) - 'proxy.sh' saved [1577/1577]

baadalvm@baadalvm:~$ ls
proxy.sh
baadalvm@baadalvm:~$
```

- (FYI, IITD uses a **Proxy server** to monitor and filter traffic using **IITD's WiFi/LAN** services. What is a proxy server, you say? Well, refer back to you COL334/672 Minor Exam question ;) and try solving it.)

**Step 2:** Thereafter, you can follow the steps provided by CSC [here](#). If you are unsure of your proxy URL and category (e.g. for MTech, the proxy URL is <http://proxy62.iitd.ac.in:3128>), please refer [here](#).

**Step 3:** Once you have successfully set up your proxy. You won't get any certificate validation failed issues, on trying to access external websites. Now you can access WAN from your BaadalVM.

- (FYI, a digital certificate is an authentication mechanism used to authenticate the actual sender of the message. e.g. IITD is doing this certificate check to ensure that data is sent by prateek and not by any other user spoofing himself/herself as prateek! Refer to [Man-In-the-Middle \(MITM\)](#) attack for more details.)

```
baadalvm@baadalvm:~$ wget google.com
--2024-09-25 12:47:23-- http://google.com/
Resolving proxy62.iitd.ac.in (proxy62.iitd.ac.in)... 10.10.78.62
Connecting to proxy62.iitd.ac.in (proxy62.iitd.ac.in)|10.10.78.62|:3128... connected.
Proxy request sent, awaiting response... 301 Moved Permanently
Location: http://www.google.com/ [following]
--2024-09-25 12:47:23-- http://www.google.com/
Reusing existing connection to proxy62.iitd.ac.in:3128.
Proxy request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html                                              [ =>                               ] 24.27K  --.-KB/s   in 0.04s

2024-09-25 12:47:23 (558 KB/s) - 'index.html' saved [24849]
baadalvm@baadalvm:~$
```

## ● Installing Mininet From Source

**Step 1:** Type **Mininet GitHub** on google and open up Mininet's source code homepage. Copy the HTTPS URL and clone the repo on Baadal VM.

```

baadalvm@baadalvm:~$ git clone https://github.com/mininet/mininet.git
Cloning into 'mininet'...
remote: Enumerating objects: 10388, done.
remote: Counting objects: 100% (234/234), done.
remote: Compressing objects: 100% (142/142), done.
remote: Total 10388 (delta 129), reused 170 (delta 90), pack-reused 10154 (from 1)
Receiving objects: 100% (10388/10388), 3.36 MiB | 10.71 MiB/s, done.
Resolving deltas: 100% (6911/6911), done.
baadalvm@baadalvm:~$ ls
index.html mininet proxy.sh
baadalvm@baadalvm:~$ rm -rf index.html
baadalvm@baadalvm:~$ ls
mininet proxy.sh
baadalvm@baadalvm:~$ 

```

- (FYI, the index.html was the file downloaded by wget when I ran wget google.com earlier to check internet access. Feel free to remove it as I did.)

**Step 2:** Now, we will build the mininet 2.3.1b4 binary from source.

```
baadalvm@baadalvm:~$ cd mininet/
baadalvm@baadalvm:~/mininet$ git tag
1.0.0
2.0.0
2.1.0
2.1.0p1
2.1.0p2
2.2.0
2.2.1
2.2.2
2.3.0
2.3.0b1
2.3.0b2
2.3.0d3
2.3.0d4
2.3.0d5
2.3.0d6
2.3.0rc1
2.3.0rc2
2.3.1b2
2.3.1b3
2.3.1b4
cs244-spring-2012-final
baadalvm@baadalvm:~/mininets git checkout -b 2.3.1b4
Switched to a new branch '2.3.1b4'
baadalvm@baadalvm:~/mininets mkdir build
baadalvm@baadalvm:~/mininet$ ls
bin build CONTRIBUTORS custom debian doc examples INSTALL LICENSE Makefile mininet mnexec.c README.md setup.py util
baadalvm@baadalvm:~/mininets$
```

**Step 3:** Now, we will use the install.sh script to install mininet. Run the following command, enter your password and wait for the script to install mininet. This might take a while.

```
./util/install.sh -s ./build/ -a
```

```

Connected (unencrypted) to: QEMU (IITD_tmangla_prateek-col334)
libtool: install: /usr/bin/install -c .libs/libof_snmp_cpu.lai /usr/local/lib/libof_snmp_cpu.la
libtool: install: /usr/bin/install -c .libs/libof_snmp_cpu.a /usr/local/lib/libof_snmp_cpu.a
libtool: install: chmod 644 /usr/local/lib/libof_snmp_cpu.a
libtool: install: ranlib /usr/local/lib/libof_snmp_cpu.a
libtool: finish: PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin:/snap/bin:/sbin" ldconfig -n /usr/local/lib

Libraries have been installed in:
/usr/local/lib

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-L' option
flag during linking and do at least one of the following:
- add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
  during execution
- add LIBDIR to the 'LD_RUN_PATH' environment variable
  during linking
- use the '-Wl,-rpath -Wl,LIBDIR' linker flag
- have your system administrator add LIBDIR to '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.

make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/home/baadalvm/mininet/build/oflops/example_modules/snmp_cpu'
make[2]: Leaving directory '/home/baadalvm/mininet/build/oflops/example_modules/snmp_cpu'
make[2]: Entering directory '/home/baadalvm/mininet/build/oflops/example_modules'
make[3]: Entering directory '/home/baadalvm/mininet/build/oflops/example_modules'
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/home/baadalvm/mininet/build/oflops/example_modules'
make[2]: Leaving directory '/home/baadalvm/mininet/build/oflops/example_modules'
make[1]: Leaving directory '/home/baadalvm/mininet/build/oflops/example_modules'
Making install in cbench
make[1]: Entering directory '/home/baadalvm/mininet/build/oflops/cbench'
make[2]: Entering directory '/home/baadalvm/mininet/build/oflops/cbench'
/usr/bin/mkdir -p '/usr/local/bin'
/bin/bash ../libtool --mode=install /usr/bin/install -c cbench '/usr/local/bin'
libtool: install: /usr/bin/install -c cbench /usr/local/bin/cbench
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/baadalvm/mininet/build/oflops/cbench'
make[1]: Leaving directory '/home/baadalvm/mininet/build/oflops/cbench'
Making install in doc
make[1]: Entering directory '/home/baadalvm/mininet/build/oflops/doc'
make[1]: Nothing to be done for 'install'.
make[1]: Leaving directory '/home/baadalvm/mininet/build/oflops/doc'
Enjoy Mininet!
baadalvm@baadalvm:~$ mininet$
```

- (FYI, for some reason my Baadal ssh pipe broke, so I switched back to the VNC. You might see me switching back and forth between VNC UI and my Local MAC Terminal throughout this tutorial.)

**Step 3:** Now, you should see an **Enjoy Mininet!** message being displayed at the end. You can also check the mininet version installed as:

```

baadalvm@baadalvm:~$ mn --version
2.3.1b4
baadalvm@baadalvm:~$
```

**Step 4:** Feel free to run a custom topology using mininet to test its working.

```
sudo mn --switch ovs --controller ref --topo tree,depth=2,fanout=8 --test pingall
```

- (FYI, the command (*refer screenshot below*) creates a network with tree topology of depth 2 and fanout 8 (i.e. 64 hosts connected to 9 switches), using Open vSwitch switches under the control of OpenFlow/Stanford reference controller, and runs the test to check connectivity between every pair of nodes with pingall. This might blow up your Terminal. Enjoy! 😊)

```
[root@baadalvm ~]# sudo mn --switch ovs --controller ref --topo tree,depth=2,fanout=8 --test pingall
[sudo] password for baadalvm:
*** Creating network
*** Adding controller
*** Adding hosts:
s1 s2 s3 s4 s5 s6 s7 s8 s9
*** Adding switches:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64
*** Adding links:
(s1, s2) (s1, s3) (s1, s4) (s1, s5) (s1, s6) (s1, s7) (s1, s8) (s1, s9) (s2, h1) (s2, h2) (s2, h3) (s2, h4) (s2, h5) (s2, h6) (s2, h7) (s2, h8) (s3, h9) (s3, h10) (s3, h11) (s3, h12) (s3, h13) (s3, h14) (s3, h15) (s3, h16) (s4, h17) (s4, h18) (s4, h19) (s4, h20) (s4, h21) (s4, h22) (s4, h23) (s4, h24) (s5, h25) (s5, h26) (s5, h27) (s5, h28) (s5, h29) (s5, h30) (s5, h31) (s5, h32) (s6, h33) (s6, h34) (s6, h35) (s6, h36) (s6, h37) (s6, h38) (s6, h39) (s6, h40) (s7, h41) (s7, h42) (s7, h43) (s7, h44) (s7, h45) (s7, h46) (s7, h47) (s7, h48) (s8, h49) (s8, h50) (s8, h51) (s8, h52) (s8, h53) (s8, h54) (s8, h55) (s8, h56) (s9, h57) (s9, h58) (s9, h59) (s9, h60) (s9, h61) (s9, h62) (s9, h63) (s9, h64)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64
*** Starting controller
c0
*** Starting 9 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9 ...
*** Waiting for switches to connect
s1 s2 s3 s4 s5 s6 s7 s8 s9
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64
```

```
[root@baadalvm ~]# h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64
h60 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64
h61 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64
h62 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64
h63 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64
h64 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64
*** Results: 0% dropped (4032/4032 received)
*** Stopping 1 controllers
c0
*** Stopping 72 links
.....
*** Stopping 9 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9
*** Stopping 64 hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64
*** Done
completed in 43.627 seconds
[root@baadalvm ~]#
```

## • Installing Python to 3.9.6 from source

**Step 1:** As promised above, now we will install a lower and compatible version of python on Ubuntu. First install the prerequisites using the following command:

```
sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev  
libssl-dev libreadline-dev libffi-dev wget
```

```
baadalvm@baadalvm:~$ sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libreadline-dev libffi-dev wget  
[sudo] password for baadalvm:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
build-essential is already the newest version (12.9ubuntu3).  
build-essential set to manually installed.  
libssl-dev is already the newest version (3.0.2-0ubuntu1.18).  
libssl-dev set to manually installed.  
zlib1g-dev is already the newest version (1:1.2.11.dfsg-2ubuntu9.2).  
zlib1g-dev set to manually installed.  
The following additional packages will be installed:  
  libncurses-dev libnspr4 libnspr4-dev libnss3  
Suggested packages:  
  ncurses-doc readline-doc  
The following NEW packages will be installed:  
  libffi-dev libgdbm-dev libncurses-dev libncurses5-dev libnspr4-dev libnss3-dev libreadline-dev  
The following packages will be upgraded:  
  libnspr4 libnss3 wget  
3 upgraded, 7 newly installed, 0 to remove and 170 not upgraded.  
Need to get 2,997 kB of archives.  
After this operation, 6,653 kB of additional disk space will be used.  
Do you want to continue? [Y/n] Y
```

(Press Tab and Space if you enter a restart services window.)

```
Unpacking libffi-dev:amd64 (3.4.2-4) ...  
Setting up libcurses-dev:amd64 (6.3-2ubuntu0.1) ...  
Setting up wget (1.21.2-2ubuntu1.1) ...  
Setting up libreadline-dev:amd64 (8.1.2-1) ...  
Setting up libffi-dev:amd64 (3.4.2-4) ...  
Setting up libnspr4:amd64 (2:4.35-0ubuntu0.22.04.1) ...  
Setting up libcurses5-dev:amd64 (6.3-2ubuntu0.1) ...  
Setting up libgdbm-dev:amd64 (1.23-1) ...  
Setting up libnspr4-dev (2:4.35-0ubuntu0.22.04.1) ...  
Setting up libnss3:amd64 (2:3.98-0ubuntu0.22.04.2) ...  
Setting up libnss3-dev:amd64 (2:3.98-0ubuntu0.22.04.2) ...  
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...  
Processing triggers for man-db (2.10.2-1) ...  
Processing triggers for install-info (6.8-4build1) ...  
Scanning processes...  
Scanning candidates...  
Scanning linux images...  
  
Running kernel seems to be up-to-date.  
  
Restarting services...  
  systemctl restart udisks2.service  
  
No containers need to be restarted.  
  
No user sessions are running outdated binaries.  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
baadalvm@baadalvm:~$ █
```

**Step 2:** Now fetch and extract the Python source code using tar as:

```
wget https://www.python.org/ftp/python/3.9.6/Python-3.9.6.tgz  
tar -xf Python-3.9.6.tgz  
cd Python-3.9.6
```

```
baadalvm@baadalvm:~$ ls
mininet proxy.sh
baadalvm@baadalvm:~$ wget https://www.python.org/ftp/python/3.9.6/Python-3.9.6.tgz
--2024-09-25 14:15:03-- https://www.python.org/ftp/python/3.9.6/Python-3.9.6.tgz
Resolving proxy62.iitd.ac.in (proxy62.iitd.ac.in)... 10.10.78.62
Connecting to proxy62.iitd.ac.in (proxy62.iitd.ac.in)|10.10.78.62|:3128... connected.
Proxy request sent, awaiting response... 200 OK
Length: 25640094 (24M) [application/octet-stream]
Saving to: 'Python-3.9.6.tgz'

Python-3.9.6.tgz          100%[=====] 24.45M   110MB/s    in 0.2s

2024-09-25 14:15:04 (110 MB/s) - 'Python-3.9.6.tgz' saved [25640094/25640094]

baadalvm@baadalvm:~$ ls
mininet proxy.sh Python-3.9.6.tgz
baadalvm@baadalvm:~$ tar -xvf Python-3.9.6.tgz
baadalvm@baadalvm:~$ ls
mininet proxy.sh Python-3.9.6 Python-3.9.6.tgz
baadalvm@baadalvm:~$ cd Python-3.9.6/
baadalvm@baadalvm:~/Python-3.9.6$ ls
aclocal.m4      config.sub  Doc    install-sh  Mac      Modules    Parser    Programs  README.rst
CODE_OF_CONDUCT.md  configure  Grammar  Lib      Makefile.pre.in  netlify.toml  PC       pyconfig.h.in  setup.py
config.guess    configure.ac  Include  LICENSE   Misc      Objects    PCbuild  Python    Tools
baadalvm@baadalvm:~/Python-3.9.6$
```

### Step 3: Build Python from source. This might take a while.

./configure --enable-optimizations

```
baadalvm@baadalvm:~/Python-3.9.6$ ./configure --enable-optimizations
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for python3.9... no
checking for python3... python3
checking for --enable-universalsdk... no
checking for --with-universal-archs... no
checking MACHDEP... "linux"
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for a sed that does not truncate output... /usr/bin/sed
checking for --with-cxx-main=<compiler>... no
checking for g++... no
configure:
```

```

checking for -O2... yes
checking for glibc _FORTIFY_SOURCE/memmove bug... no
checking for gcc ipa-pure-const bug... no
checking for stdatomic.h... yes
checking for GCC >= 4.7 __atomic builtins... yes
checking for ensurepip... upgrade
checking if the dirent structure of a d_type field... yes
checking for the Linux getrandom() syscall... yes
checking for the getrandom() function... yes
checking for library containing shm_open... none required
checking for sys/mman.h... (cached) yes
checking for shm_open... yes
checking for shm_unlink... yes
checking for pkg-config... /usr/bin/pkg-config
checking whether compiling and linking against OpenSSL works... yes
checking for X509_VERIFY_PARAM_set1_host in libssl... yes
checking for --with-ssl-default-suites... python
checking for --with-builtin-hashlib-hashes... md5,sha1,sha256,sha512,sha3,blake2
configure: creating ./config.status
config.status: creating Makefile.pre
config.status: creating Misc/python.pc
config.status: creating Misc/python-embed.pc
config.status: creating Misc/python-config.sh
config.status: creating Modules/ld_so_aix
config.status: creating pyconfig.h
creating Modules/Setup.local
creating Makefile
baadalvm@baadalvm:~/Python-3.9.6$ 

```

**Step 4:** Install the alternate version (in this case, 3.9.6) of python using Make. This might take a while. You can verify the installation as depicted below.

sudo make altinstall

- (FYI, read more about [Make](#) here).

```

baadalvm@baadalvm:~/Python-3.9.6$ sudo make altinstall
gcc -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initializers -Werror=implicit-function-declaration -fvisibility=hidden -I./Include/internal -I. -I./Include -DPy_BUILD_CORE -o Programs/python.o ./Programs/python.c
gcc -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initializers -Werror=implicit-function-declaration -fvisibility=hidden -I./Include/internal -I. -I./Include -DPy_BUILD_CORE -o Parser/acceler.o Parser/acceler.c
gcc -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initializers -Werror=implicit-function-declaration -fvisibility=hidden -I./Include/internal -I. -I./Include -DPy_BUILD_CORE -o Parser/grammar1.o Parser/grammar1.c
gcc -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initializers -Werror=implicit-function-declaration -fvisibility=hidden -I./Include/internal -I. -I./Include -DPy_BUILD_CORE -o Parser/listnode.o Parser/listnode.c
gcc -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initializers -Werror=implicit-function-declaration -fvisibility=hidden -I./Include/internal -I. -I./Include -DPy_BUILD_CORE -o Parser/node.o Parser/node.c
gcc -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initializers -Werror=implicit-function-declaration -fvisibility=hidden -I./Include/internal -I. -I./Include -DPy_BUILD_CORE -o Parser/parser.o Parser/parser.c
gcc -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initializers -Werror=implicit-function-declaration -fvisibility=hidden -I./Include/internal -I. -I./Include -DPy_BUILD_CORE -o Parser/token.o Parser/token.c
gcc -c -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -std=c99 -Wextra -Wno-unused-result -Wno-unused-parameter -Wno-missing-field-initializers -Werror=implicit-function-declaration -fvisibility=hidden -I./Include/internal -I. -I./Include -DPy_BUILD_CORE -o Parser/pegen.o Parser/pegen/pegen.c

```

```
[1] Looking in links: /tmp/tmpg06exu5h
Processing /tmp/tmpg06exu5h/setuptools-56.0.0-py3-none-any.whl
Processing /tmp/tmpg06exu5h/pip-21.1.3-py3-none-any.whl
Installing collected packages: setuptools, pip
Successfully installed pip-21.1.3 setuptools-56.0.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
baadalvm@baadalvm:~/Python-3.9.6$ python3.9 --version
Python 3.9.6
baadalvm@baadalvm:~/Python-3.9.6$ pip3 --version
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
baadalvm@baadalvm:~/Python-3.9.6$
```

**Step 5:** Create a new python 3.9.6 virtual environment and source into it.

```
cd ~
python3.9 -m venv py_3.9.6
source py_3.9.6/bin/activate
```

```
baadalvm@baadalvm:~$ ls
mininet proxy.sh Python-3.9.6 Python-3.9.6.tgz
baadalvm@baadalvm:~$ rm -rf Python-3.9.6.tgz
baadalvm@baadalvm:~$ python3.9 -m venv py_3.9.6
baadalvm@baadalvm:~$ ls
mininet proxy.sh py_3.9.6 Python-3.9.6
baadalvm@baadalvm:~$ source py_3.9.6/bin/activate
(py_3.9.6) baadalvm@baadalvm:~$
```

- Installing Ryu

**Step 1:** Install ryu as a python package using pip. You can later verify its installation as shown.

```
pip3 install ryu
```

```
(py_3.9.6) baadalvm@baadalvm:~$ pip3 install ryu
Collecting ryu
  Downloading ryu-4.34.tar.gz (1.1 MB)
    |████████| 1.1 MB 4.4 MB/s
Collecting eventlet!=0.18.3,!=0.20.1,!=0.21.0,!=0.23.0,>=0.18.2
  Downloading eventlet-0.37.0-py3-none-any.whl (361 kB)
    |████████| 361 kB 16.1 MB/s
Collecting msgpack>=0.3.0
  Downloading msgpack-1.1.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (377 kB)
    |████████| 377 kB 27.2 MB/s
Collecting netaddr
  Downloading netaddr-1.3.0-py3-none-any.whl (2.3 MB)
    |████████| 2.3 MB 55.8 MB/s
Collecting oslo.config>=2.5.0
  Downloading oslo.config-9.6.0-py3-none-any.whl (132 kB)
    |████████| 132 kB 74.3 MB/s
Collecting ovs>=2.6.0
  Downloading ovs-3.4.0.tar.gz (132 kB)
    |████████| 132 kB 85.2 MB/s
Collecting routes
  Downloading Routes-2.5.1-py2.py3-none-any.whl (40 kB)
    |████| 40 kB 7.3 MB/s
Collecting six>=1.4.0
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Collecting tinyrpc
```

```
(py_3.9.6) baadalvm@baadalvm:~$ pip3 show ryu
Name: ryu
Version: 4.34
Summary: Component-based Software-defined Networking Framework
Home-page: http://osrg.github.io/ryu/
Author: Ryu project team
Author-email: ryu-devel@lists.sourceforge.net
License: Apache License 2.0
Location: /home/baadalvm/py_3.9.6/lib/python3.9/site-packages
Requires: eventlet, msgpack, netaddr, oslo.config, ovs, routes, six, tinyrpc, webob
Required-by:
(py_3.9.6) baadalvm@baadalvm:~$ █
```

- [Fixing Ryu and Eventlet dependency](#)

**Step 1:** As you can see above ryu requires eventlet but trying to use ryu-manager might lead into an eventlet dependency error as shown below:

```
(py_3.9.6) baadalvm@baadalvm:~$ ryu-manager --version
Traceback (most recent call last):
  File "/home/baadalvm/py_3.9.6/bin/ryu-manager", line 33, in <module>
    sys.exit(load_entry_point('ryu==4.34', 'console_scripts', 'ryu-manager')())
  File "/home/baadalvm/py_3.9.6/bin/ryu-manager", line 25, in importlib_load_entry_point
    return next(matches).load()
  File "/usr/local/lib/python3.9/importlib/metadata.py", line 77, in load
    module = import_module(match.group('module'))
  File "/usr/local/lib/python3.9/importlib/_init_.py", line 127, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
File "<frozen importlib._bootstrap>", line 1030, in _gcd_import
File "<frozen importlib._bootstrap>", line 1007, in _find_and_load
File "<frozen importlib._bootstrap>", line 986, in _find_and_load_unlocked
File "<frozen importlib._bootstrap>", line 680, in _load_unlocked
File "<frozen importlib._bootstrap_external>", line 850, in exec_module
File "<frozen importlib._bootstrap>", line 228, in _call_with_frames_removed
File "/home/baadalvm/py_3.9.6/lib/python3.9/site-packages/ryu/cmd/manager.py", line 33, in <module>
  from ryu.app import wsgi
File "/home/baadalvm/py_3.9.6/lib/python3.9/site-packages/ryu/app/wsgi.py", line 109, in <module>
  class _AlreadyHandledResponse(Response):
File "/home/baadalvm/py_3.9.6/lib/python3.9/site-packages/ryu/app/wsgi.py", line 111, in _AlreadyHandledResponse
  from eventlet.wsgi import ALREADY_HANDLED
ImportError: cannot import name 'ALREADY_HANDLED' from 'eventlet.wsgi' (/home/baadalvm/py_3.9.6/lib/python3.9/site-packages/eventlet/wsgi.py)
(py_3.9.6) baadalvm@baadalvm:~$ █
```

**Step 2:** To fix this, we will first change our eventlet version.

`pip install eventlet==0.30.2`

```
(py_3.9.6) baadalvm@baadalvm:~$ pip install eventlet==0.30.2
Collecting eventlet==0.30.2
  Using cached eventlet-0.30.2-py2.py3-none-any.whl (224 kB)
Requirement already satisfied: dnspython<2.0.0,>=1.15.0 in ./py_3.9.6/lib/python3.9/site-packages (from eventlet==0.30.2) (1.16.0)
Requirement already satisfied: greenlet>=0.3 in ./py_3.9.6/lib/python3.9/site-packages (from eventlet==0.30.2) (3.1.1)
Requirement already satisfied: six>=1.10.0 in ./py_3.9.6/lib/python3.9/site-packages (from eventlet==0.30.2) (1.16.0)
Installing collected packages: eventlet
Successfully installed eventlet-0.30.2
WARNING: You are using pip version 21.1.3; however, version 24.2 is available.
You should consider upgrading via the '/home/baadalvm/py_3.9.6/bin/python3.9 -m pip install --upgrade pip' command.
(py_3.9.6) baadalvm@baadalvm:~$ ryu-manager --version
ryu-manager 4.34
(py_3.9.6) baadalvm@baadalvm:~$ █
```

- **Verify installation and version:**

**Step 1:** Use the following commands to verify the installed versions over Baadal VM. You should receive a similar output.

```
(py_3.9.6) baadalvm@baadalvm:~$ python --version
Python 3.9.6
(py_3.9.6) baadalvm@baadalvm:~$ pip3 --version
pip 21.1.3 from /home/baadalvm/py_3.9.6/lib/python3.9/site-packages/pip (python 3.9)
(py_3.9.6) baadalvm@baadalvm:~$ ryu-manager --version
ryu-manager 4.34
(py_3.9.6) baadalvm@baadalvm:~$ mn --version
2.3.1b4
(py_3.9.6) baadalvm@baadalvm:~$ pip show eventlet
Name: eventlet
Version: 0.30.2
Summary: Highly concurrent networking library
Home-page: http://eventlet.net
Author: Linden Lab
Author-email: eventletdev@lists.secondlife.com
License: UNKNOWN
Location: /home/baadalvm/py_3.9.6/lib/python3.9/site-packages
Requires: dnspython, greenlet, six
Required-by: ryu
(py_3.9.6) baadalvm@baadalvm:~$ pip show ryu
Name: ryu
Version: 4.34
Summary: Component-based Software-defined Networking Framework
Home-page: http://osrg.github.io/ryu/
Author: Ryu project team
Author-email: ryu-devel@lists.sourceforge.net
License: Apache License 2.0
Location: /home/baadalvm/py_3.9.6/lib/python3.9/site-packages
Requires: eventlet, msgpack, netaddr, oslo.config, ovs, routes, six, tinyrpc, webob
Required-by:
(py_3.9.6) baadalvm@baadalvm:~$ █
```

- **Moving Assignment files into BaadalVM:**

**Step 1:** Once you have installed the required softwares for this assignment, you need to move the files into the Baadal VM. First download the files from Moodle into your Downloads folder.

The screenshot shows the Moodle assignment interface for 'Assignment-3'. On the left, there's a navigation sidebar with links like 'Dashboard', 'Site home', 'My courses' (expanded to show '2401-COL334'), and 'Assignment-3'. The main area is titled 'Assignment-3' and lists the submitted files: 'Assignment-3.pdf', 'p1\_topo.py', 'p2\_topo.py', and 'p3\_topo.py'. Below this is the 'Grading summary' section, which includes fields for Participants (182), Submitted (0), Needs grading (0), Due date (Sunday, 6 October 2024, 11:59 PM), and Time remaining (11 days 9 hours). At the bottom, there are buttons for 'View all submissions' and 'Grade'.

**Step 2:** Open a new terminal in your Downloads folder and SCP the files from your local machine to your remote Baadal VM.

```
scp Assignment-3.pdf p1_topo.py p2_topo.py p3_topo.py baadalvm@10.17.50.98:~/
```



- (FYI, Secure copy protocol (*SCP*) is a means of securely transferring computer files between a local host (e.g. your machine, here) and a remote host (e.g. Baadal VM, here) or between two remote hosts via Secure Shell (*SSH*) protocol).

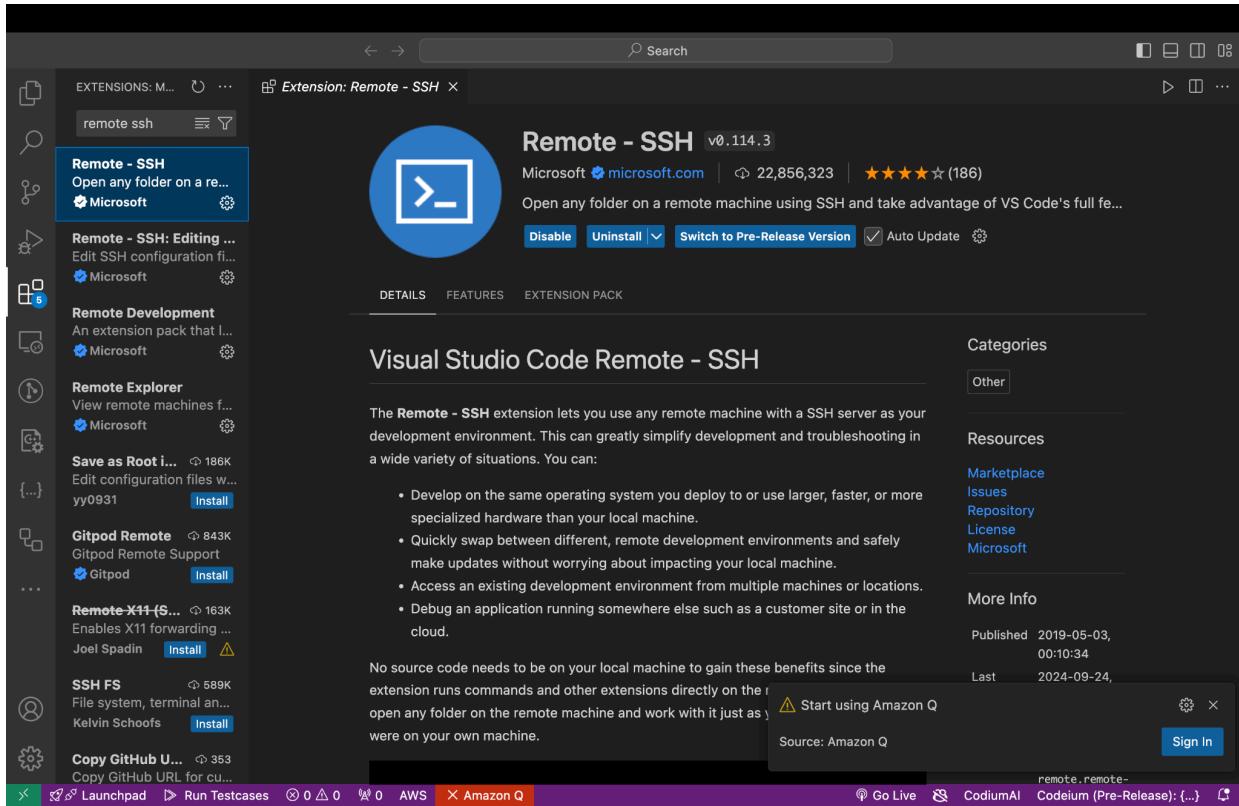
**Step 3:** Test the topology files on your BaadalVM.

```
sudo python3 p1_topo.py
```

```
(py_3.9.6) baadalvm@baadalvm:~$ ls
Assignment-3.pdf mininet p1_topo.py p2_topo.py p3_topo.py proxy.sh py_3.9.6 Python-3.9.6
(py_3.9.6) baadalvm@baadalvm:~$ sudo python3 p1_topo.py
Unable to contact the remote controller at 127.0.0.1:6633
*** Creating network
*** Adding hosts:
h1 h2 h3 h4 h5
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s2) (h5, s2) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4 h5
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Running CLI
*** Starting CLI:
mininet> █
```

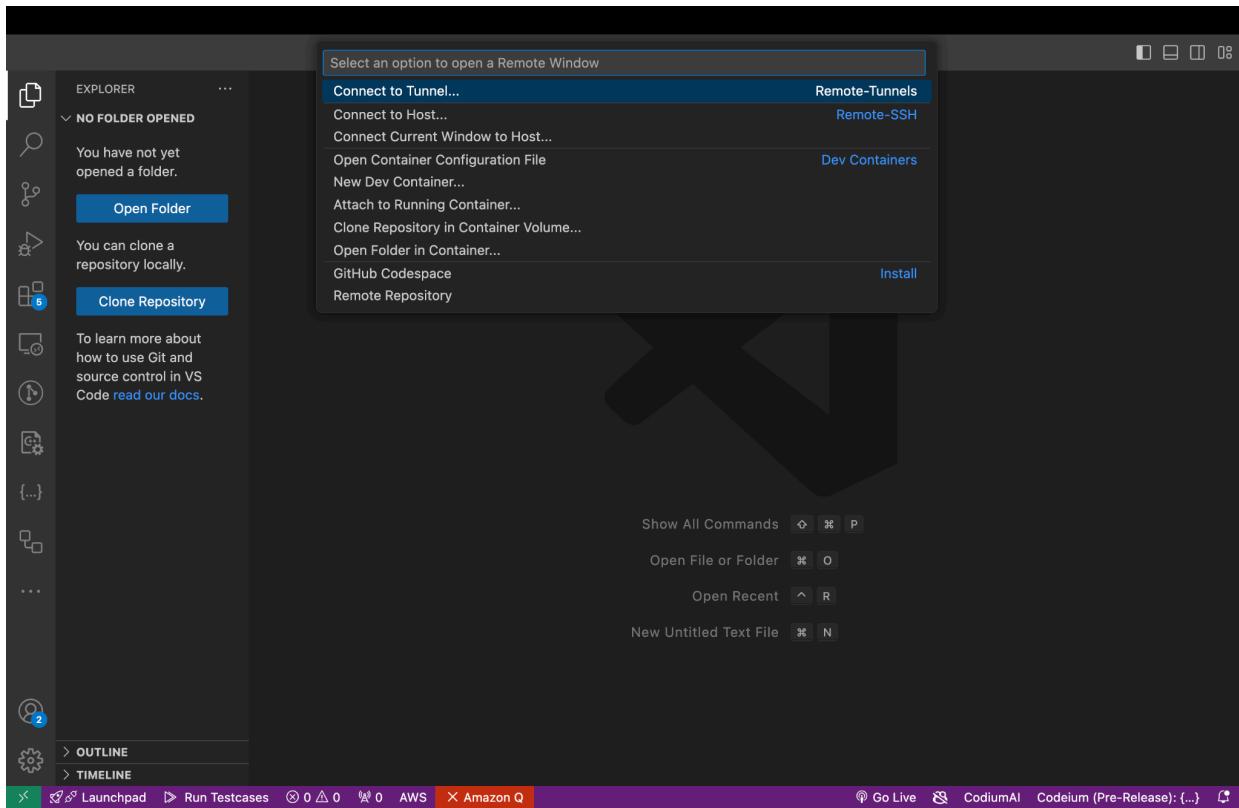
- (FYI, the **Unable to contact the remote controller** error is due to you not starting the Ryu SDN Controller in another terminal. Please refer to the docs shared by Tarun Mangla Sir to learn how to use and program using Mininet and Ryu).
- **Setting up Remote SSH Development Environment using VS Code: (Optional)**

**Step 1:** Yeah I know, vim and nano are not everyone's cup of tea! So, let's set up a remote ssh dev environment using VSCode. Go to extensions tab in vs code and download the Remote-SSH extension.

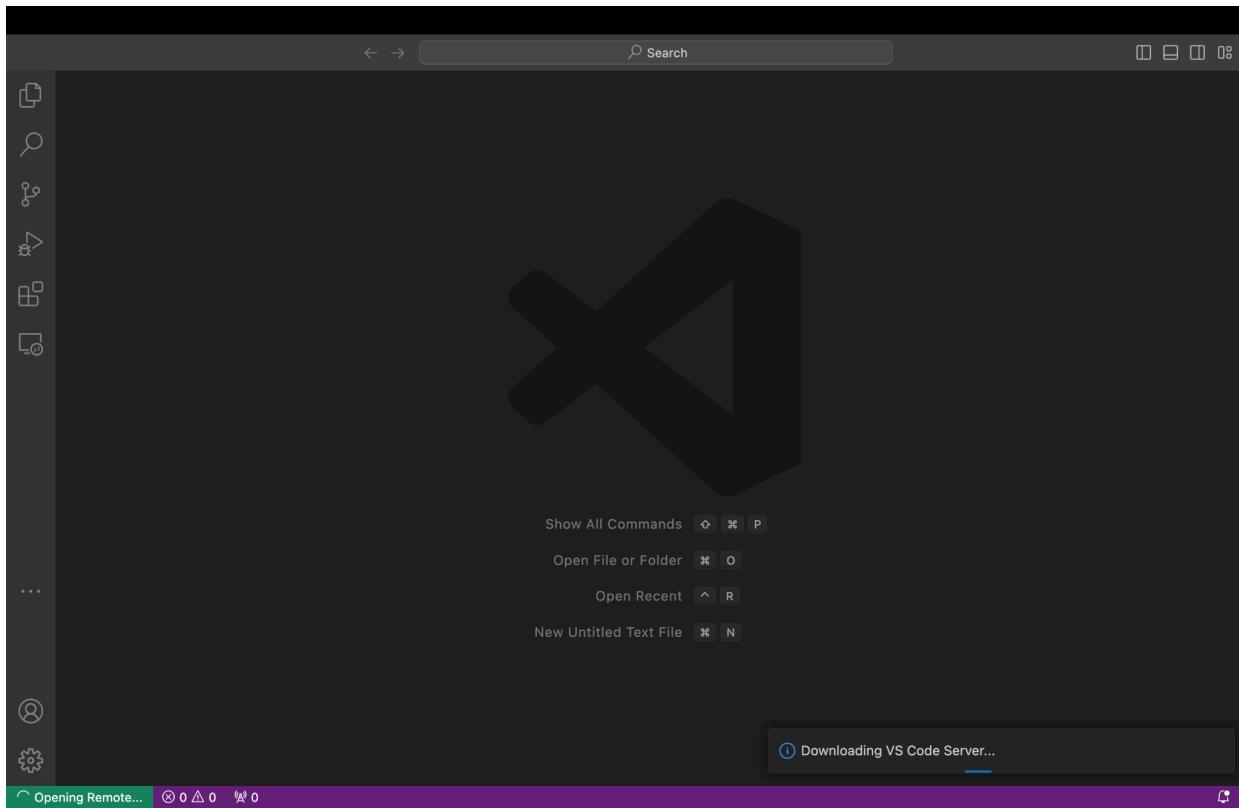


**Step 2:** Click on the small icon on bottom left of VSCode and select connect to host and Add a new host. Fill in your SSH command here and select any config file.

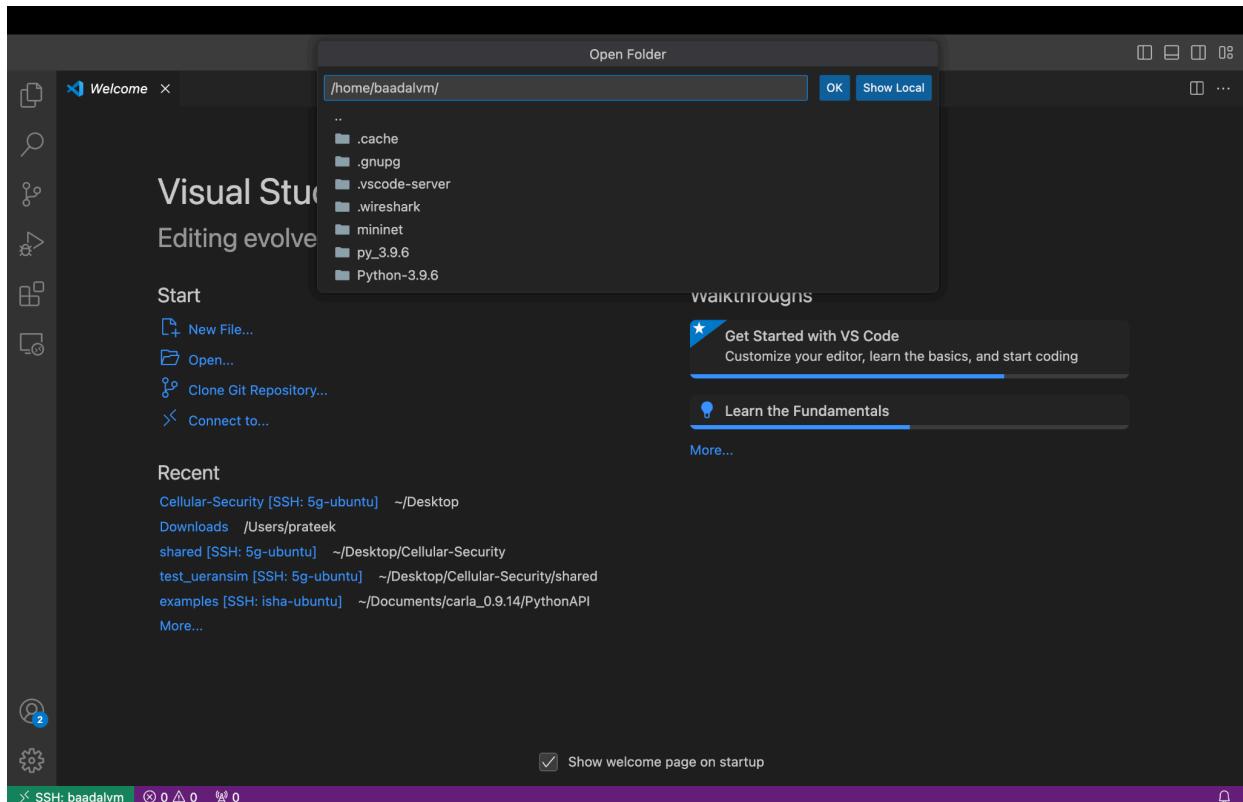
**Step 3:** Now again select icon and choose the connect to host option. Choose your BaadalVM IP and enter your password as prompted.



**Step 4:** Wait for VSCode to download a VSCode server on your Baadal VM. Note that this is a one time process and might take a while.



**Step 5:** Once the VSCode server is downloaded, you will get automatically SSH'ed into your BaadalVM. Click on File > Open Folder to use your new dev environment. All changes you do here will reflect on your Baadal VM. You can use the inbuilt terminal of VSCode (using Ctrl+`) or, code here and run Baadal VM on a separate terminal.



The screenshot shows the VS Code interface with the 'EXPLORER' and 'EDITOR' panes visible. The Explorer pane on the left shows a file tree with a folder named 'BAADALV...' containing various files like '.cache', '.gnupg', '.vscode-server', '.wireshark', 'mininet', 'py\_3.9.6', 'Python-3.9.6', '.bash\_history', '.bash\_logout', '.bashrc', '.gitconfig', '.lessht', '.profile', '.python\_history', '.sudo\_as\_admin\_su...', '.viminfo', '.wget-hsts', 'Assignment-3.pdf', 'p1\_topo.py', 'p2\_topo.py', 'p3\_topo.py', and 'proxy.sh'. The 'p1\_topo.py' file is selected and shown in the Editor pane on the right. The code in 'p1\_topo.py' is a Python script for creating a network topology using the Mininet library:

```
1  from mininet.topo import Topo
2  from mininet.net import Mininet
3  from mininet.log import setLogLevel, info
4  from mininet.cli import CLI
5  from mininet.node import RemoteController, OVSSwitch
6
7  class CustomTopo(Topo):
8      def build(self):
9          # Add two switches
10         s1 = self.addSwitch('s1')
11         s2 = self.addSwitch('s2')
12
13         # Add hosts
14         h1 = self.addHost('h1')
15         h2 = self.addHost('h2')
16         h3 = self.addHost('h3')
17         h4 = self.addHost('h4')
18         h5 = self.addHost('h5')
19
20         # Connect hosts to switch S1
21         self.addLink(h1, s1)
22         self.addLink(h2, s1)
23         self.addLink(h3, s1)
24
25         # Connect hosts to switch S2
26         self.addLink(h4, s2)
27         self.addLink(h5, s2)
28
29         # Optionally, add a link between the two switches
30         self.addLink(s1, s2)
31
32     def run():
33         """Create the network, start it, and enter the CLI."""
34         topo = CustomTopo()
35         net = Mininet(topo=topo, switch=OVSSwitch, build=False)
```

The status bar at the bottom indicates the file is 2.3.1b4, has 0 changes, and is 0% complete. It also shows the current encoding as UTF-8 and the language as Python.