

Cloud Parallel Computing

Evaluating R code in parallel

Wael Mohammed

Table of contents

0.1	Introduction:	1
0.2	Create an account in Linode:	1
0.3	Creating a Linode instance:	1
0.3.1	Distributions:	2
0.3.2	Initialise Linode Instance:	5
0.4	Accessing a Linode instance:	7
0.4.1	Accessing a Linode instance from the terminal:	7
0.5	Moving files to/from the Linode instance:	9
0.6	Working on the Linode instance:	11
0.6.1	Installing R:	11
0.6.2	Running Rscripts on the Linode instance:	13
0.6.3	Processing local R code remotely without moving any files	17

0.1 Introduction:

This document discusses creating, interacting with and using cloud computing.

0.2 Create an account in Linode:

- Go to <https://login.linode.com/signup> and register for a new account. At the time of writing, Linode was running a promotion where new users are given \$100 worth of credit to spend within the platform.

Warning: Remember to shut down and delete all Linode instances once done with them to avoid unintended costs.

- Log in to the Linode account at <https://login.linode.com/login>.

0.3 Creating a Linode instance:

From the Linode dashboard:

- Select **Linodes** from the left side menu.

0.3 Creating a Linode instance:

TABLE OF CONTENTS

- Click on **Create Linode**, and several tabs will populate the page.

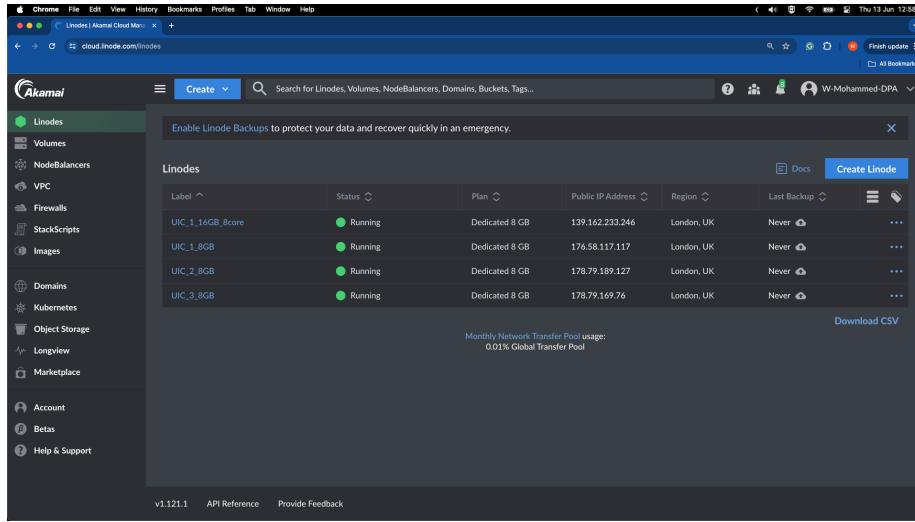


Figure 1: Linode Dashboard

0.3.1 Distributions:

- **Choose a Distribution:** Select a suitable operating system image.
 - Select Ubuntu 24.04 LTS from the list of images.

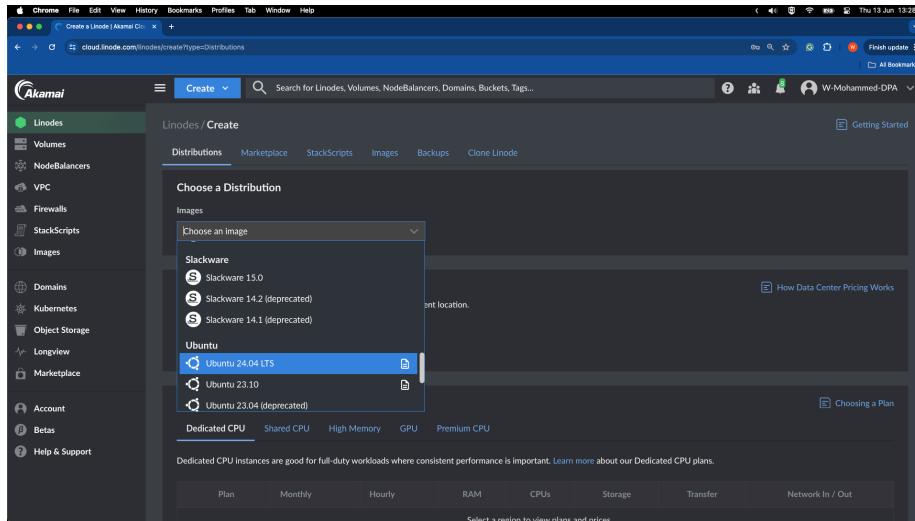


Figure 2: Selecting a Linode Operating System

- **Region:** Choose the closest location.

- Choose London, UK (eu-west) from the list of regions.

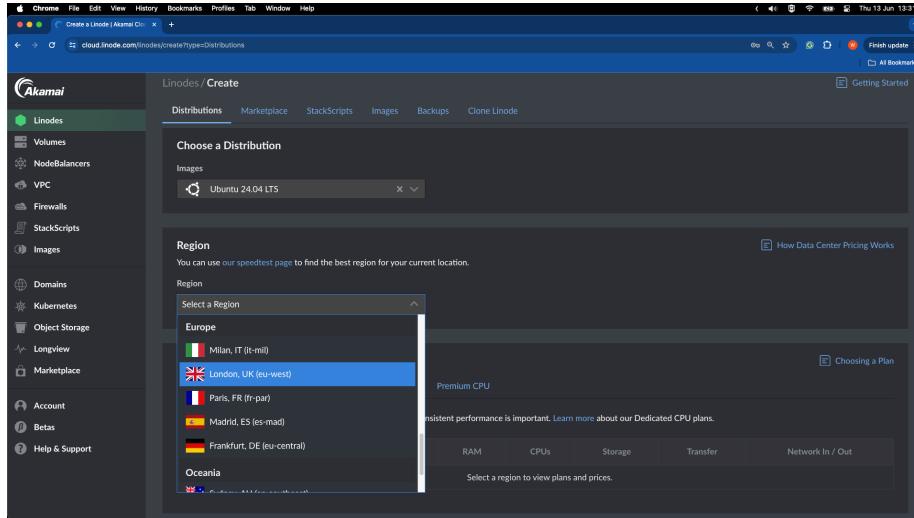


Figure 3: Choosing a Linode Region

- **Linode Plan:** Choose an appropriate computing node.
 - Choose Dedicated 8 GB from the list of plans.

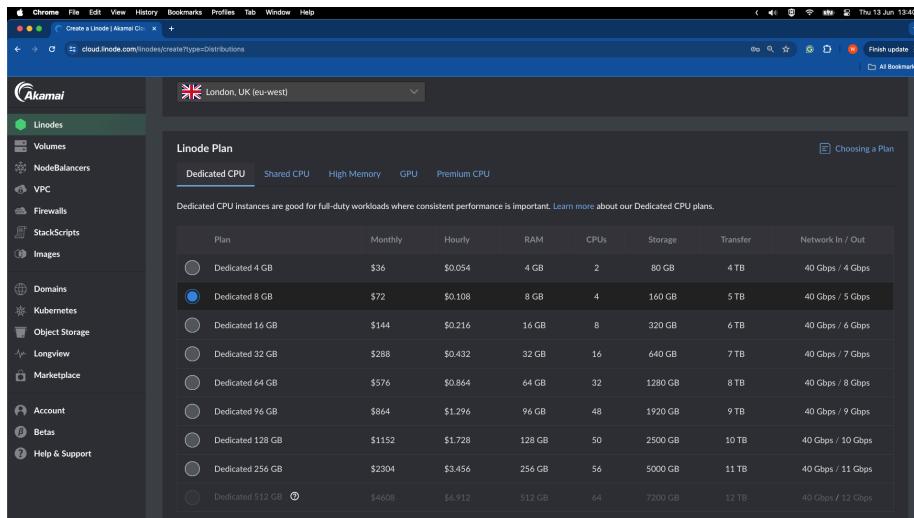


Figure 4: Choosing a Linode Plan

- **Linode Details:**
 - **Linode Label:** Give the cloud computing instance a meaningful label.

* Use UIC_4_8GB

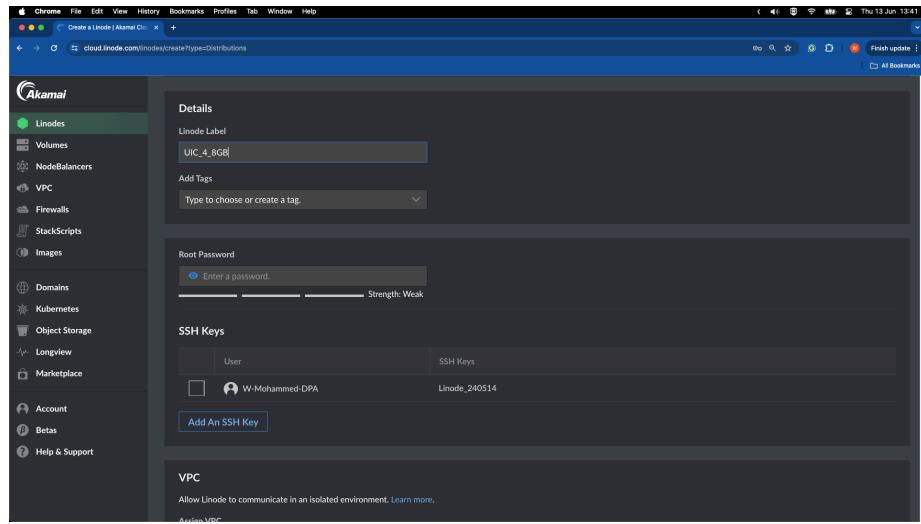


Figure 5: Labelling a Linode Instance

- **Add Tags:** A tag if required.
- **Root Password:** Set a root password.
- **SSH Keys:** Click Add An SSH Key, paste your SSH public key and label it. To learn more about using SSH and SSH keys, check:
 - <https://www.linode.com/docs/guides/connect-to-server-over-ssh-on-windows/>
 - <https://www.youtube.com/watch?v=ZVMckBHd7WA>

0.3 Creating a Linode instance:

TABLE OF CONTENTS

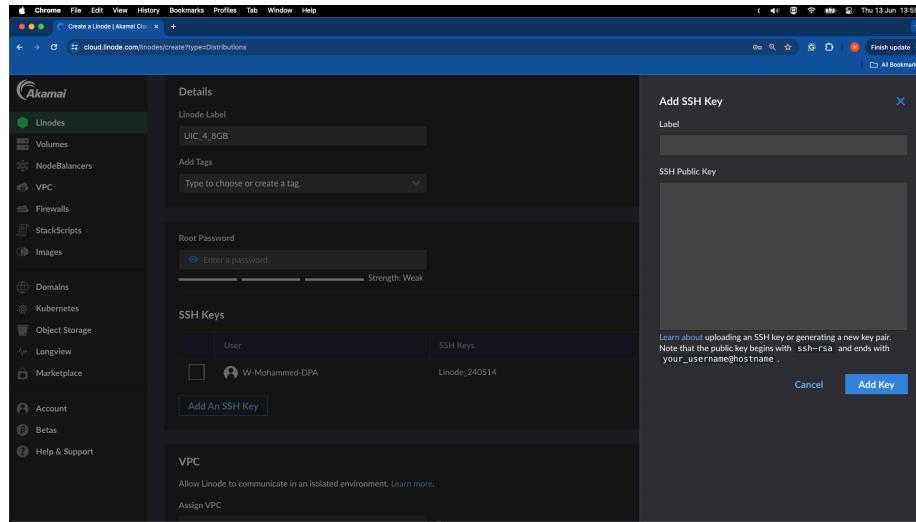


Figure 6: Using SSH Key

0.3.2 Initialise Linode Instance:

Click **Create Linode** to confirm the details and spin the chosen instance.

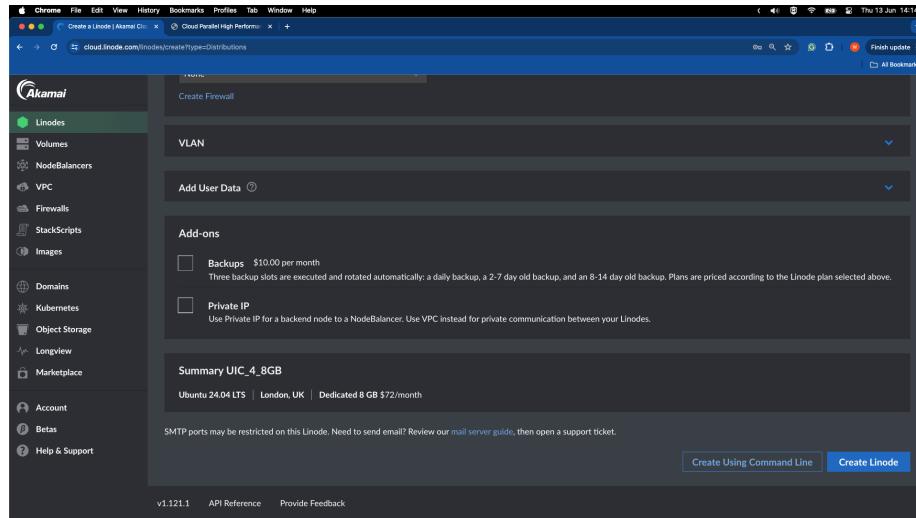


Figure 7: Create Linode

The Linode instance will start spinning, and the page will show “Provisioning”, “Booting” and “Running”.

0.3 Creating a Linode instance:

TABLE OF CONTENTS

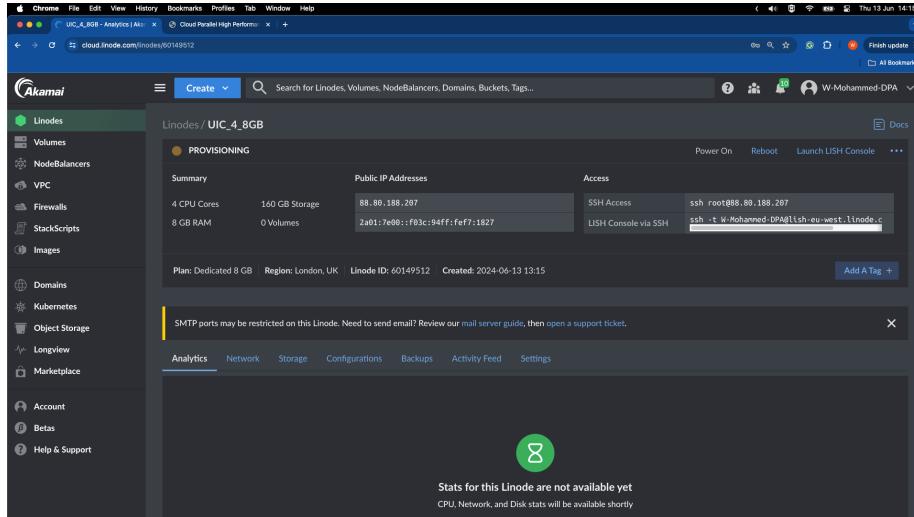


Figure 8: Linode Provisioning

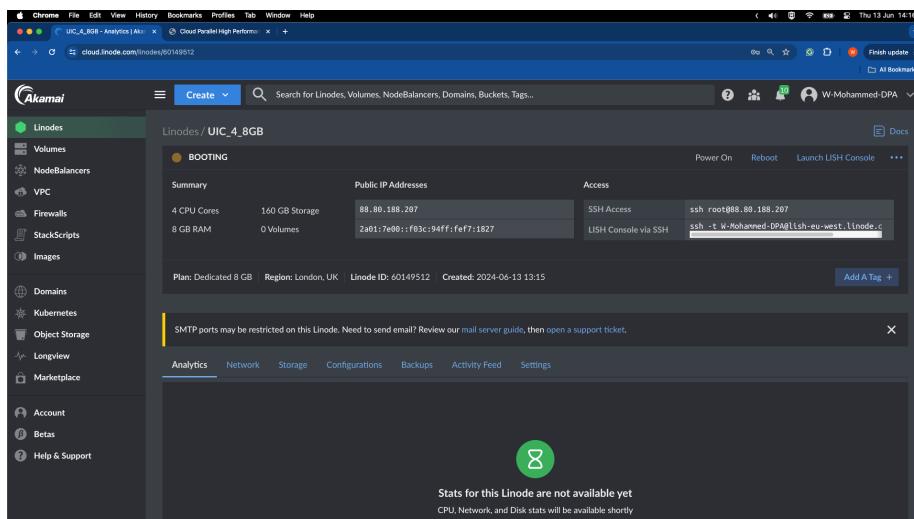


Figure 9: Linode Booting

The Linode instance is ready once it has finished **Booting**; when it is shown that it is **Running**,

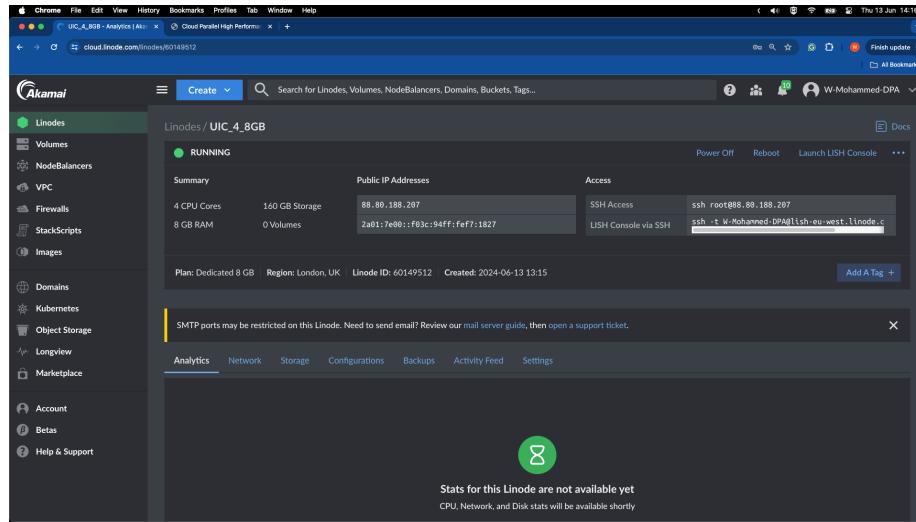


Figure 10: Linode Running

0.4 Accessing a Linode instance:

The Linode instance can be accessed by ssh-ing to the instance. Two methods are demonstrated below.

0.4.1 Accessing a Linode instance from the terminal:

- From the Linode dashboard copy the **SSH Access** (`ssh root@xxx.xxx.xx.xx`), `ssh root@88.80.188.207` for the *UIC_4_8GB*.
- Paste on and execute the command from macOS/Linux terminal.

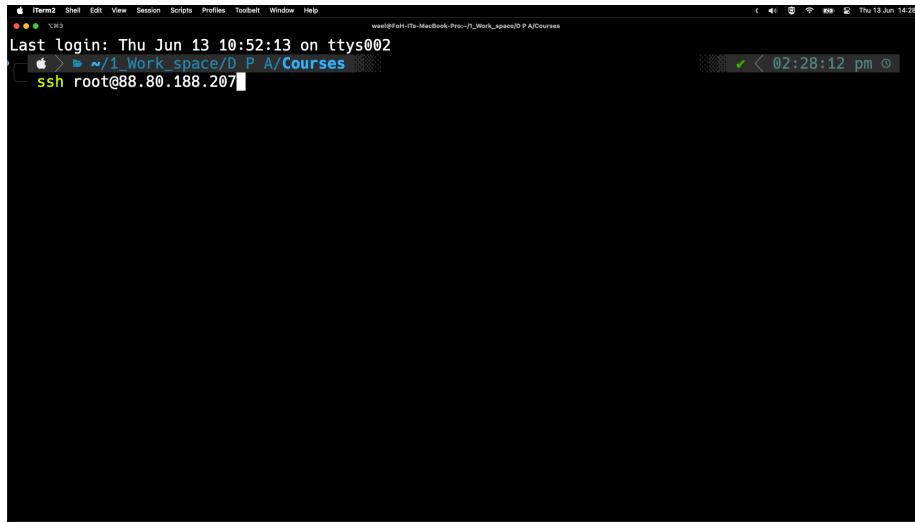


Figure 11: SSH-ing into the Linode Instance

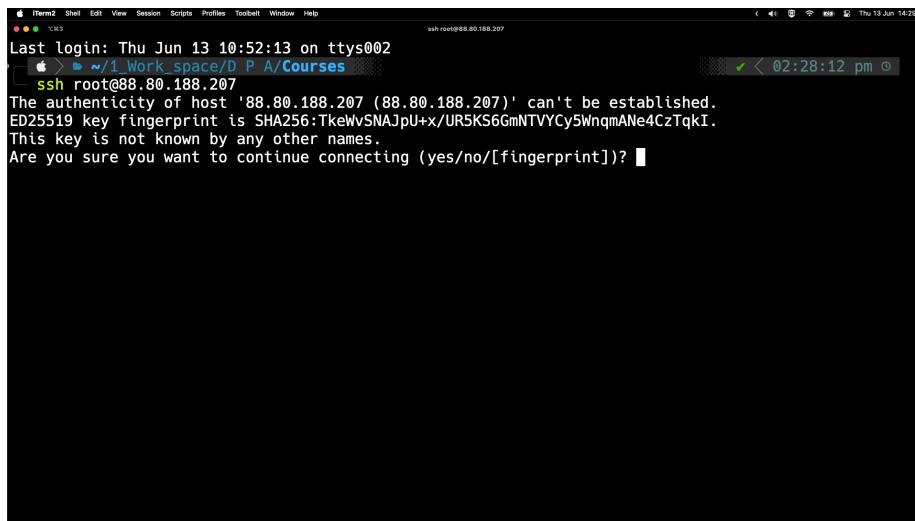


Figure 12: Accept Instance as one of the known hosts

The screenshot shows a terminal window with the following content:

```

root@localhost: ~
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Thu Jun 13 01:29:52 PM UTC 2024

System load: 0.0
Usage of /: 1.3% of 156.93GB
Memory usage: 2%
Swap usage: 0%
Processes: 128
Users logged in: 0
IPv4 address for eth0: 88.80.188.207
IPv6 address for eth0: 2a01:7e00::f03c:94ff:fef7:1827

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@localhost:~# 

```

Figure 13: Linode Instance Terminal

The *LISH Console via SSH* allows users to log in to the Linode instance from the Linode SSH Console.

0.5 Moving files to/from the Linode instance:

The following commands are expected to be run from the local machine's terminal, not the Linode instance.

```

# Move one file from the current folder to the root of the instance:
scp cloud_instance_setup.sh root@xxx.xxx.xxx.xxx:

# Move one file from the current folder to a folder 'microSim' located in the root of the instance
scp cloud_instance_setup.sh root@xxx.xxx.xxx.xxx:microSim/

# To move a folder from the current directory to the root of the instance:
scp -r r4he_uic/ root@xxx.xxx.xxx.xxx:

```

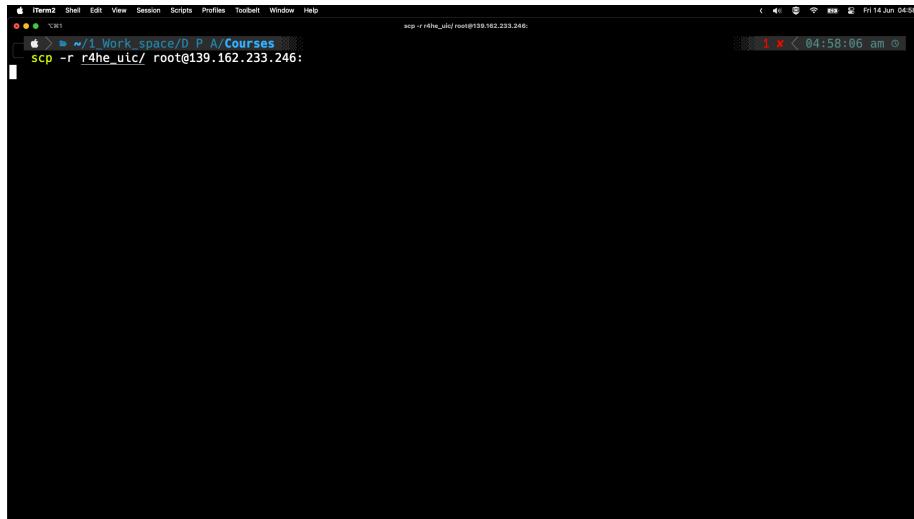


Figure 14: Copy to Linode

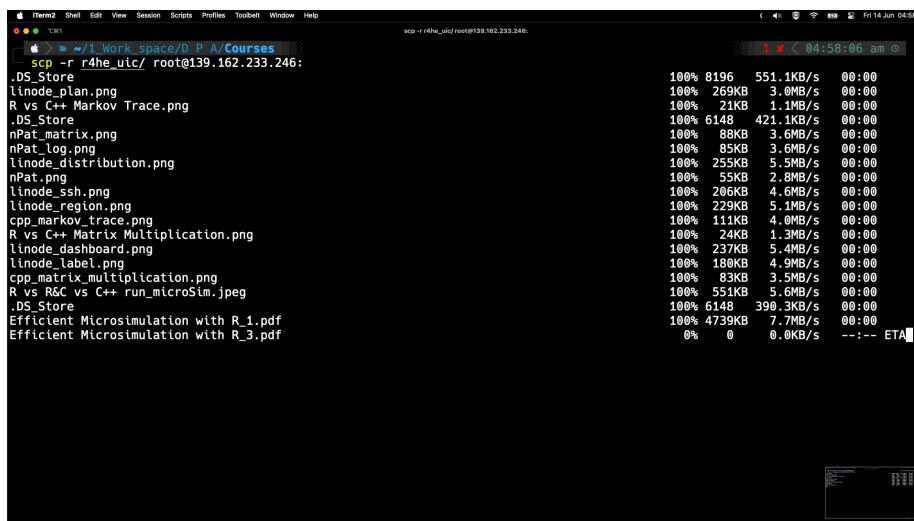


Figure 15: Copy to Linode

```
# To move a file from the Linode instance to our local computer:
scp root@xxx.xxx.xxx.xxx:file_name .
```

```

ls -lah
rwxr-xr-x  6 wael  staff   192 B  Fri Jun 14 05:03:07 2024  ./ 
rwx----- 21 wael  staff   672 B  Thu Apr 25 16:16:15 2024  ../.. 
rwxr--r--  1 wael  staff   10 KB  Fri Jun 14 05:00:23 2024  .DS_Store 
rwxr-xr-x  44 wael  staff    1 KB  Tue Jun 11 22:12:01 2024  R4HE_coursebook/ 
rwxr-xr-x  3 wael  staff   96 B   Tue May 14 23:13:41 2024  R4HE_coursebook_backup/ 
rwx----- 14 wael  staff   448 B  Fri Jun 14 03:43:15 2024  r4he_ulc/ 

ls -lah
rwxr-xr-x  7 wael  staff   224 B  Fri Jun 14 05:03:59 2024  ./ 
rwx----- 21 wael  staff   672 B  Thu Apr 25 16:16:15 2024  ../.. 
rwxr--r--  1 wael  staff   10 KB  Fri Jun 14 05:00:23 2024  .DS_Store 
rwxr-xr-x  44 wael  staff    1 KB  Tue Jun 11 22:12:01 2024  R4HE_coursebook/ 
rwxr-xr-x  3 wael  staff   96 B   Tue May 14 23:13:41 2024  R4HE_coursebook_backup/ 
rwx----- 2  wael  staff   3 KB   Fri Jun 14 05:03:59 2024  README.md 
rwx----- 14 wael  staff   448 B  Fri Jun 14 03:43:15 2024  r4he_ulc/ 

100% 3289  119.5KB/s  00:00

```

Figure 16: Copy from Linode

To learn more about moving files from and to a remote instance, check <https://www.youtube.com/watch?v=LMC5VNoZFhg&t=105s>.

0.6 Working on the Linode instance:

0.6.1 Installing R:

Instructions on how to add the latest R version can be found at <https://cran.r-project.org/bin/linux/ubuntu/fullREADME.html>.

Run the following code from the Linode instance terminal to install R and some of the required R packages.

```

# Update and upgrade the system
sudo apt update && apt upgrade -y

# Install software-properties-common to manage repositories
sudo apt install software-properties-common -y

# Update and upgrade the system again after installing the add-apt-repository
sudo apt update && apt upgrade -y

# Add the R repository appropriate for the Ubuntu release $(lsb_release -cs)
sudo add-apt-repository "deb https://cloud.r-project.org/bin/linux/ubuntu $(lsb_release -cs)/"

# Import the R repository signing key
wget -qO- https://cloud.r-project.org/bin/linux/ubuntu/marutter_pubkey.asc | tee -a /etc/apt

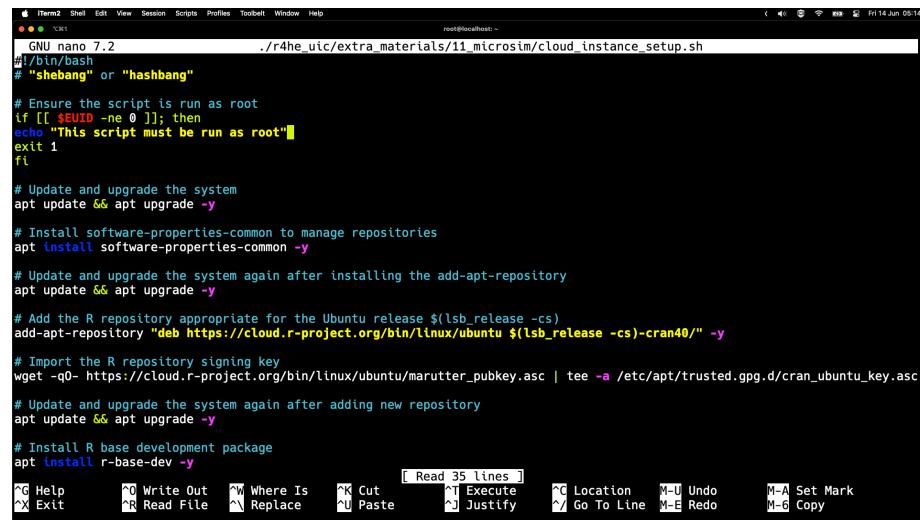
```

```
# Update and upgrade the system again after adding new repository
sudo apt update && apt upgrade -y

# Install R base development package
sudo apt install r-base-dev -y

# Install R packages
R -e "install.packages(c('future', 'purrr', 'furrr', 'ggplot2', 'Rcpp', 'RcppArmadillo', 'he
```

The above commands are saved in the `bash` file `cloud_instance_setup.sh`. The “`cloud_instance_setup.sh`” script allows the execution of all commands by using the command `sudo bash ./cloud_instance_setup.sh`.



The screenshot shows a terminal window titled "Term2" running on a Linode instance. The window displays the code of the `cloud_instance_setup.sh` script. The script starts with a shebang line (`#!/bin/bash`), followed by a root privilege check and an error message if run as anything other than root. It then performs several system updates and installs the R base development package. At the bottom of the terminal, the nano editor's status bar is visible, showing "Read 35 lines".

```
#!/bin/bash
# "shebang" or "hashbang"

# Ensure the script is run as root
if [[ $EUID -ne 0 ]]; then
echo "This script must be run as root"
exit 1
fi

# Update and upgrade the system
apt update && apt upgrade -y

# Install software-properties-common to manage repositories
apt install software-properties-common -y

# Update and upgrade the system again after installing the add-apt-repository
apt update && apt upgrade -y

# Add the R repository appropriate for the Ubuntu release $(lsb_release -cs)
add-apt-repository "deb https://cloud.r-project.org/bin/linux/ubuntu $(lsb_release -cs)-cran40/" -y

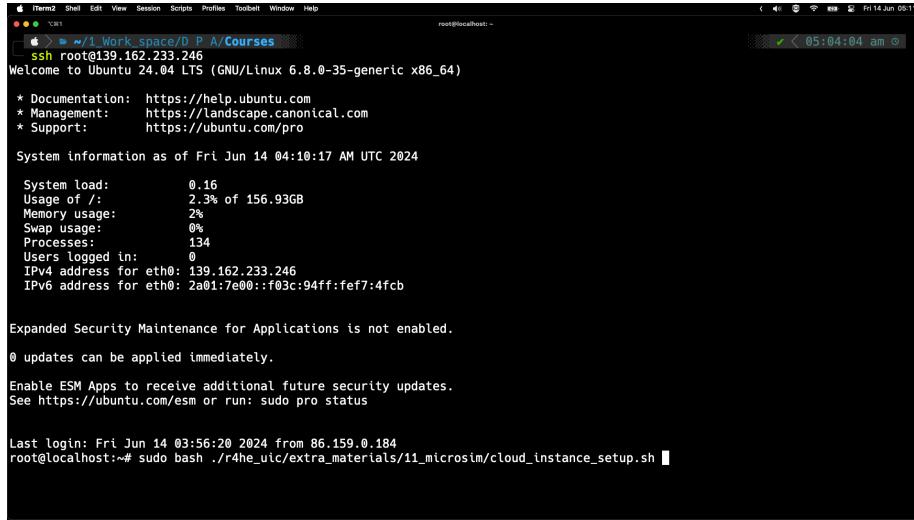
# Import the R repository signing key
wget -qO- https://cloud.r-project.org/bin/linux/ubuntu/marutter_pubkey.asc | tee -a /etc/apt/trusted.gpg.d/cran_ubuntu_key.asc

# Update and upgrade the system again after adding new repository
apt update && apt upgrade -y

# Install R base development package
apt install r-base-dev -y
```

Figure 17: `cloud_instance_setup.sh`

This command assumes that the `bash` file `cloud_instance_setup.sh` is in the working directory of the Linode instance. See Section 0.5 for how to move files to and from the Linode instance.



```

TERM2 Shell Edit View Session Scripts Profiles Toolbar Window Help
root@localhost: ~
ssh root@139.162.233.246
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-35-generic x86_64)

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

System information as of Fri Jun 14 04:10:17 AM UTC 2024

System load: 0.16
Usage of /: 2.3% of 156.93GB
Memory usage: 2%
Swap usage: 0%
Processes: 134
Users logged in: 0
IPv4 address for eth0: 139.162.233.246
IPv6 address for eth0: 2a01:7e00::f03c:94ff:fef7:4fc

Expanded Security Maintenance for Applications is not enabled.

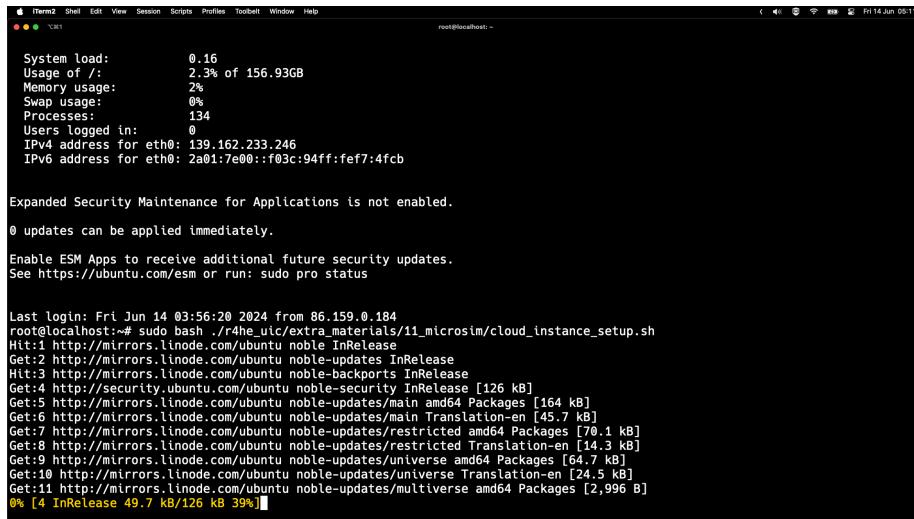
0 updates can be applied immediately.

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

Last login: Fri Jun 14 03:56:20 2024 from 86.159.0.184
root@localhost:~# sudo bash ./4he_uic/extr.../11_microsim/cloud_instance_setup.sh

```

Figure 18: Executing the cloud_instance_setup.sh



```

TERM2 Shell Edit View Session Scripts Profiles Toolbar Window Help
root@localhost: ~
ssh root@139.162.233.246
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-35-generic x86_64)

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

System information as of Fri Jun 14 04:10:17 AM UTC 2024

System load: 0.16
Usage of /: 2.3% of 156.93GB
Memory usage: 2%
Swap usage: 0%
Processes: 134
Users logged in: 0
IPv4 address for eth0: 139.162.233.246
IPv6 address for eth0: 2a01:7e00::f03c:94ff:fef7:4fc

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

Last login: Fri Jun 14 03:56:20 2024 from 86.159.0.184
root@localhost:~# sudo bash ./4he_uic/extr.../11_microsim/cloud_instance_setup.sh
Hit:1 http://mirrors.linode.com/ubuntu noble InRelease
Get:2 http://mirrors.linode.com/ubuntu noble-updates InRelease
Hit:3 http://mirrors.linode.com/ubuntu noble-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://mirrors.linode.com/ubuntu noble-updates/main amd64 Packages [164 kB]
Get:6 http://mirrors.linode.com/ubuntu noble-updates/main Translation-en [45.7 kB]
Get:7 http://mirrors.linode.com/ubuntu noble-updates/restricted amd64 Packages [76.1 kB]
Get:8 http://mirrors.linode.com/ubuntu noble-updates/restricted Translation-en [14.3 kB]
Get:9 http://mirrors.linode.com/ubuntu noble-updates/universe amd64 Packages [64.7 kB]
Get:10 http://mirrors.linode.com/ubuntu noble-updates/universe Translation-en [24.5 kB]
Get:11 http://mirrors.linode.com/ubuntu noble-updates/multiverse amd64 Packages [2,996 B]
0% [4 InRelease 49.7 kB/126 kB 39%]

```

Figure 19: Executing the cloud_instance_setup.sh

0.6.2 Running Rscripts on the Linode instance:

The command `Rscript` is used to execute R scripts. The screenshots below showcase running the R script that compares parallel execution methods, namely “forking” and “PSOCK”. The required code is in the script `cloud_parallel_terminal.R`.

```
root@localhost:~# cd r4he_uic/extramaterials/11_microsim/
root@localhost:~/r4he_uic/extramaterials/11_microsim# ls
cloud_instance_setup.sh          cpp_develop_random.R      nPat.R
cloud_parallel_execution_notes.R  cpp_develop_run_microSim.R  nPat_src.R
cloud_parallel_remote.R          cpp_develop_sample.R    optimise_mat_vs_dataframe.R
cloud_parallel_remote_src.R     cpp_develop_update_probs.R  rates_to_probs.R
cloud_parallel_terminal.R        cpp_markov_trace_RvC.R   recap_vector_sumprod.R
common_go_slows.R               cpp_matrixMult_RvC.R   reprex1.R
compare_vect_cumSum.R           example_trace_LifeYear_calculation.R sampleV_vs_loopsolution.R
cpp_develop_calc_costs.R        linode                      src
cpp_develop_calc_discount_wts.R logical_indexing.R       with_function.R
cpp_develop_calc_effs.R         microsim_surv.R
root@localhost:~/r4he_uic/extramaterials/11_microsim# Rscript cloud_parallel_terminal.R
```

Figure 20: Executing the cloud_instance_setup.sh

The `cloud_parallel_terminal.R` script defines two versions of the `run_psa_parallel()` functions. The first function utilizes the `furrr` and `future` packages; whereas, the second employs the `parallel` function.

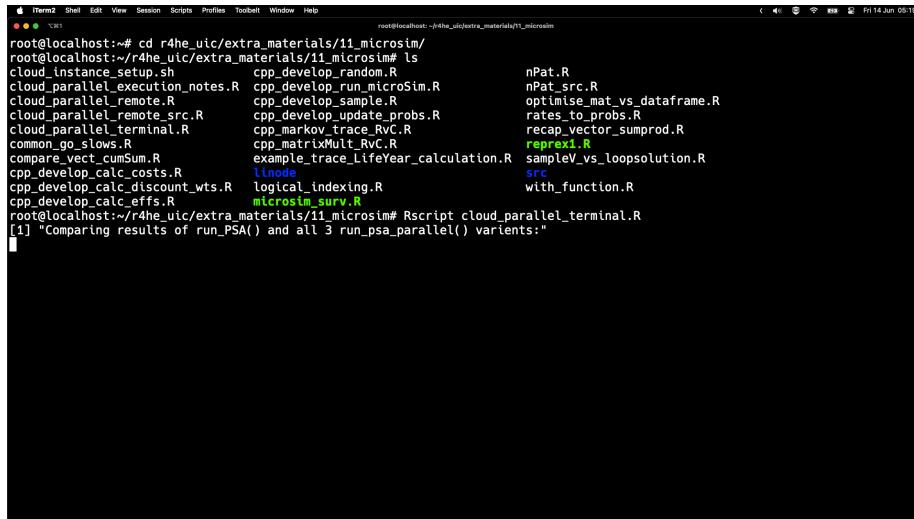
The following screenshots represent the outputs from running the R script `cloud_parallel_terminal.R`.

```
root@localhost:~# cd r4he_uic/extramaterials/11_microsim/
root@localhost:~/r4he_uic/extramaterials/11_microsim# ls
cloud_instance_setup.sh          cpp_develop_random.R      nPat.R
cloud_parallel_execution_notes.R  cpp_develop_run_microSim.R  nPat_src.R
cloud_parallel_remote.R          cpp_develop_sample.R    optimise_mat_vs_dataframe.R
cloud_parallel_remote_src.R     cpp_develop_update_probs.R  rates_to_probs.R
cloud_parallel_terminal.R        cpp_markov_trace_RvC.R   recap_vector_sumprod.R
common_go_slows.R               cpp_matrixMult_RvC.R   reprex1.R
compare_vect_cumSum.R           example_trace_LifeYear_calculation.R sampleV_vs_loopsolution.R
cpp_develop_calc_costs.R        linode                      src
cpp_develop_calc_discount_wts.R logical_indexing.R       with_function.R
cpp_develop_calc_effs.R         microsim_surv.R
root@localhost:~/r4he_uic/extramaterials/11_microsim# Rscript cloud_parallel_terminal.R
```

Figure 21: Executing the cloud_instance_setup.sh

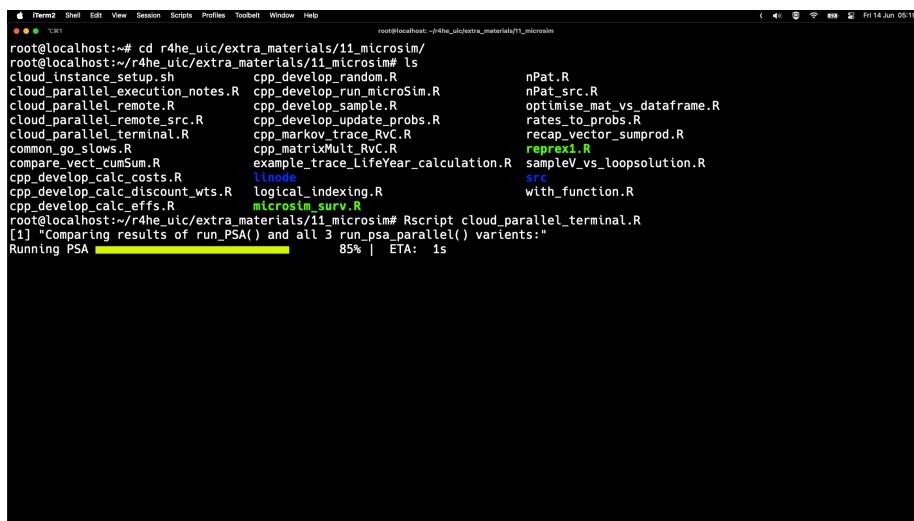
0.6 Working on the Linode instance:

TABLE OF CONTENTS



```
root@localhost:~# cd r4he_uic/extr.../11_microsim/
root@localhost:~/r4he_uic/extr.../11_microsim# ls
cloud_instance_setup.sh          cpp_develop_random.R      nPat.R
cloud_parallel_execution_notes.R  cpp_develop_run_microSim.R  nPat_src.R
cloud_parallel_remote.R          cpp_develop_sample.R     optimise_mat_vs_dataframe.R
cloud_parallel_remote_src.R      cpp_develop_update_probs.R  rates_to_probs.R
cloud_parallel_terminal.R        cpp_markov_trace_RvC.R    recap_vector_sumprod.R
common_go_slows.R               cpp_matrixMult_RvC.R    reprex1.R
compare_vect_cumSum.R           example_trace_LifeYear_calculation.R sampleV_vs_loopsolution.R
cpp_develop_calc_costs.R         linode                      src
cpp_develop_calc_discount_wts.R  logical_indexing.R       with_function.R
cpp_develop_calc_effs.R          microsim_surv.R
root@localhost:~/r4he_uic/extr.../11_microsim# Rscript cloud_parallel_terminal.R
[1] "Comparing results of run_PSA() and all 3 run_psa_parallel() variants:"
```

Figure 22: Executing the cloud_instance_setup.sh



```
root@localhost:~# cd r4he_uic/extr.../11_microsim/
root@localhost:~/r4he_uic/extr.../11_microsim# ls
cloud_instance_setup.sh          cpp_develop_random.R      nPat.R
cloud_parallel_execution_notes.R  cpp_develop_run_microSim.R  nPat_src.R
cloud_parallel_remote.R          cpp_develop_sample.R     optimise_mat_vs_dataframe.R
cloud_parallel_remote_src.R      cpp_develop_update_probs.R  rates_to_probs.R
cloud_parallel_terminal.R        cpp_markov_trace_RvC.R    recap_vector_sumprod.R
common_go_slows.R               cpp_matrixMult_RvC.R    reprex1.R
compare_vect_cumSum.R           example_trace_LifeYear_calculation.R sampleV_vs_loopsolution.R
cpp_develop_calc_costs.R         linode                      src
cpp_develop_calc_discount_wts.R  logical_indexing.R       with_function.R
cpp_develop_calc_effs.R          microsim_surv.R
root@localhost:~/r4he_uic/extr.../11_microsim# Rscript cloud_parallel_terminal.R
[1] "Comparing results of run_PSA() and all 3 run_psa_parallel() variants:"
```

Running PSA [REDACTED] 85% | ETA: [REDACTED]

Figure 23: Executing the cloud_instance_setup.sh

```

root@localhost:~# cd r4he_uic/extramaterials/11_microsim/
root@localhost:~/r4he_uic/extramaterials/11_microsim# ls
cloud_instance_setup.sh          cpp_develop_random.R      nPat.R
cloud_parallel_execution_notes.R  cpp_develop_randomMicroSim.R  nPat_src.R
cloud_parallel_remote.R          cpp_develop_sample.R    optimise_mat_vs_dataframe.R
cloud_parallel_remote_src.R      cpp_develop_update_probs.R rates_to_probs.R
cloud_parallel_terminal.R        cpp_markov_trace_RvC.R   recap_vector_sumprod.R
common_go_slops.R               cpp_matrixMult_RvC.R   reprex1.R
compare_vect_cumSum.R           example_trace_LifeYear_calculation.R sampleV_vs_loopsolution.R
cpp_develop_calc_costs.R        linode                   src
cpp_develop_calc_discount_wts.R logical_indexing.R   with_function.R
cpp_develop_calc_effs.R         microsim_surv.R
root@localhost:~/r4he_uic/extramaterials/11_microsim# Rscript cloud_parallel_terminal.R
[1] "Comparing results of run_PSA() and all 3 run_psa_parallel() variants:"
Number of workers: 1
Number of workers: 2
Number of workers: 2
[1] "Check if all four results, run_microSim() and run_psa_parallel() variants:"
[1] TRUE
[1] TRUE
[1] TRUE
[1] "Without future - 20 sims:"
```

Running PSA ██████████ 30% | ETA: 6s

Figure 24: Executing the cloud_instance_setup.sh

```

root@localhost:~/r4he_uic/extramaterials/11_microsim#
user  system elapsed
0.319  0.019  5.375
[1] "Multisession - 4 workers - 20 sims:"
Number of workers: 4
user  system elapsed
0.365  0.020  4.969
[1] "Multisession - 4 workers - 100 sims:"
Number of workers: 4
user  system elapsed
1.077  0.025  17.108
[1] "Multicore - 2 workers - 20 sims:"
Number of workers: 2
user  system elapsed
4.407  0.196  4.619
[1] "Multicore - 4 workers - 20 sims:"
Number of workers: 4
user  system elapsed
6.304  0.337  3.412
[1] "Multicore - 4 workers - 100 sims:"
Number of workers: 4
user  system elapsed
47.123  0.713  16.087
[1] "Compare Parallel package execution results with run_psa():"
[1] TRUE
[1] TRUE
[1] "multisession - 2 workers - 20 sims - Parallel package:"
user  system elapsed
0.029  0.004  4.559
[1] "Multicore - 2 workers - 20 sims - Parallel package:"
user  system elapsed
4.224  0.145  4.648
[1] "multisession - 4 workers - 100 sims - Parallel package:"
```

Figure 25: Executing the cloud_instance_setup.sh

```

Terminal Shell View Session Scripts Profiles Toolbar Window Help
root@localhost:~/r4he_uic/extr...#!/microsim
[1] 0.365 0.020 4.969
[1] "Multisession - 4 workers - 100 sims:"
Number of workers: 4
  user system elapsed
1.077 0.025 17.108
[1] "Multicore - 2 workers - 20 sims:"
Number of workers: 2
  user system elapsed
4.407 0.196 4.619
[1] "Multicore - 4 workers - 20 sims:"
Number of workers: 4
  user system elapsed
6.304 0.337 3.412
[1] "Multicore - 4 workers - 100 sims:"
Number of workers: 4
  user system elapsed
47.123 0.713 16.087
[1] "Compare Parallel package execution results with run_psa():"
[1] TRUE
[1] TRUE
[1] "multisession - 2 workers - 20 sims - Parallel package:"
  user system elapsed
0.029 0.004 4.559
[1] "Multicore - 2 workers - 20 sims - Parallel package:"
  user system elapsed
4.224 0.145 4.648
[1] "multisession - 4 workers - 100 sims - Parallel package:"
  user system elapsed
0.047 0.007 15.141
[1] "Multicore - 4 workers - 100 sims - Parallel package:"
  user system elapsed
47.372 0.645 16.339
root@localhost:~/r4he_uic/extr...# 

```

Figure 26: Executing the cloud_instance_setup.sh

0.6.3 Processing local R code remotely without moving any files

This section briefly discusses sending R code from an R session running on the local machine (the user's laptop or PC) to be executed on a remote cluster. This process does not require the user to interact directly with the remote cluster (moving files to the cluster and running them as in Section 0.5 and Section 0.6.2).

The process is powered by the `furrr` and `future` packages, and the information needed for the `future` package to establish a connection with the cluster is the instance's **Public IP Address**.

The code required to demonstrate dispatching and processing code from an R session running locally to a Linode instance is described in detail in the R script `cloud_parallel_remote.R`.

0.6 Working on the Linode instance:

TABLE OF CONTENTS

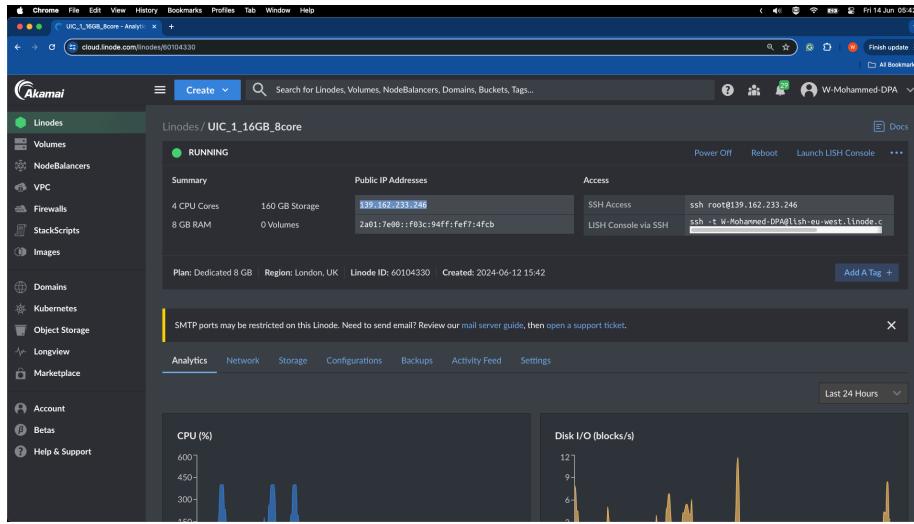


Figure 27: Linode Instance Public IP Address

The command `top -i` or `htop` can be executed on the Linode instance to monitor when or check that the remote execution of R code is running on the Linode.

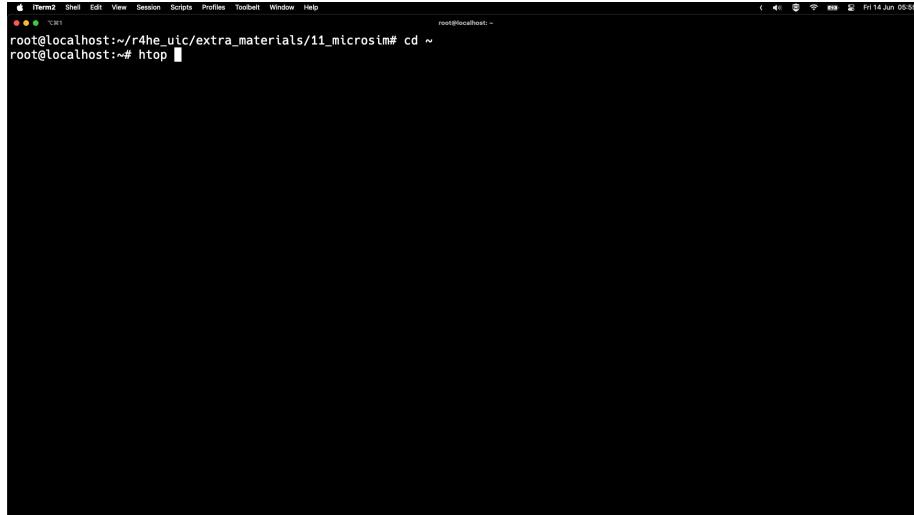


Figure 28: Running the htop command

The `htop` command displays a dynamic task-manager-like interface to show the running processes and their CPU and memory usage, among other statistics.

0.6 Working on the Linode instance:

TABLE OF CONTENTS

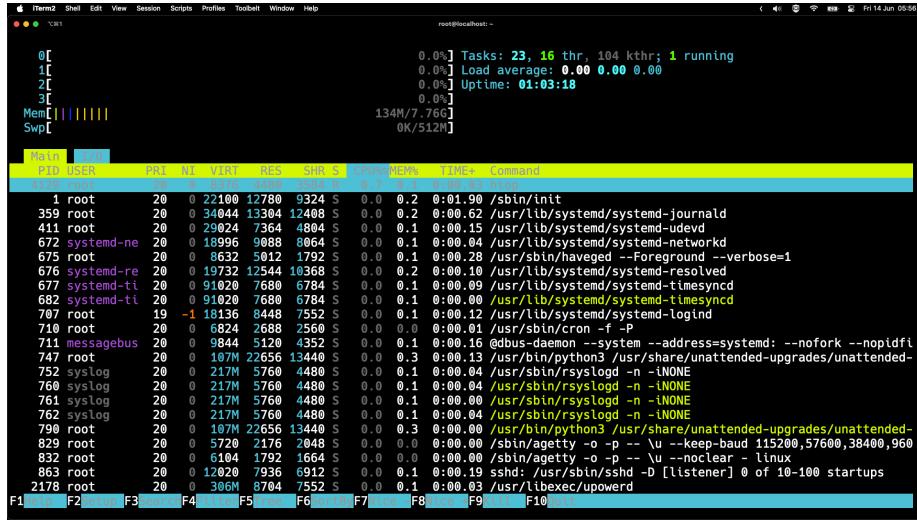


Figure 29: Linode Instance - htop

The command `htop` is used below to monitor the execution of the code sent from the R script `cloud_parallel_remote.R`. This code involves running 100 PSA iterations on the Linode instance.

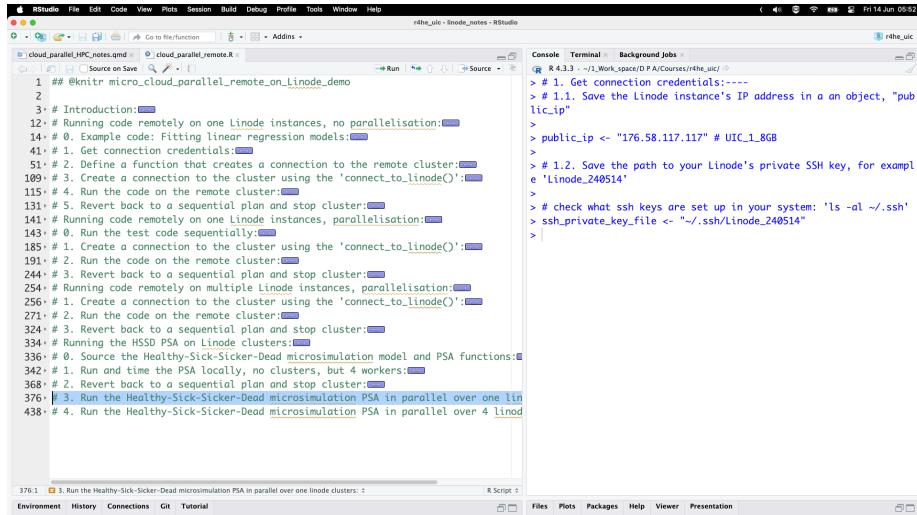


Figure 30: `cloud_parallel_remote.R`

The connection credentials include the SSH private key and the Linode instance Public IP Address.

0.6 Working on the Linode instance:

TABLE OF CONTENTS

The screenshot shows an RStudio interface. The left pane displays an R script titled "cloud_parallel.R" containing code for parallel processing. The right pane shows a terminal window with the command "htop" running, displaying system resource usage.

```

 376: # 3. Run the Healthy-Sick-Sicker-Dead microsimulation PSA in parallel over one li
377: # 3.1. Add one cluster to the 'public_ip' vector:
378:
379: public_ip <- c(
380:   "139.162.233.246" # UIC_1_16GB_8core
381: )
382:
383: # 3.2. Create the cluster and connect to it:
384:
385: # print the cluster object
386: if(exists("cl")) cl
387: cl <- connect_to_linode(public_ip, ssh_private_key_file)
388: # print the cluster object
389: cl
390:
391: # 3.3. Define the parallel processing strategy:
392:
393: future::plan(
394:   strategy = list(
395:     # The outer plan instructs the outer future loop to distribute over the insta
396:     future::tweak(
397:       strategy = future::cluster, # the outer future loop will be over clusters
398:       workers = cl # pass the cluster(s) connection defined above
399:     ),
400:
401:     # The inner plan instructs the inner future loop to run in parallel on each i
402:     future::tweak(
403:       strategy = future::multisession, # the inner future loop will distribute ov
404:       workers = 4 # set the number of workers to parallel ov
405:   )
406:
407:   # Run the Healthy-Sick-Sicker-Dead microsimulation PSA in parallel over one Linode clusters:
408:   cloud_parallel()
409:
410:   # 3. Run the Healthy-Sick-Sicker-Dead microsimulation PSA in parallel over one Linode clusters: t
411: 
```

Figure 31: Adding the Linode Instance IP Address

The screenshot below displays the `htop` command (running on the Linode instance terminal) next to the R session (running on the local PC). The parallel execution code appears in R's console but has yet to be executed.

The screenshot shows an RStudio interface. The left pane displays an R script titled "cloud_parallel.R" containing code for parallel processing. The right pane shows a terminal window with the command "htop" running, displaying system resource usage.

```

 376: # 3. Run the Healthy-Sick-Sicker-Dead microsimulation PSA in parallel over one li
377: # 3.1. Add one cluster to the 'public_ip' vector:
378:
379: public_ip <- c(
380:   "139.162.233.246" # UIC_1_16GB_8core
381: )
382:
383: # 3.2. Create the cluster and connect to it:
384:
385: # print the cluster object
386: if(exists("cl")) cl
387: cl <- connect_to_linode(public_ip, ssh_private_key_file)
388: # print the cluster object
389: cl
390:
391: # 3.3. Define the parallel processing strategy:
392:
393: future::plan(
394:   strategy = list(
395:     # The outer plan instructs the outer future loop to distribute over the insta
396:     future::tweak(
397:       strategy = future::cluster, # the outer future loop will be over clusters
398:       workers = cl # pass the cluster(s) connection defined above
399:     ),
400:
401:     # The inner plan instructs the inner future loop to run in parallel on each i
402:     future::tweak(
403:       strategy = future::multisession, # the inner future loop will distribute ov
404:       workers = 4 # set the number of workers to parallel ov
405:   )
406:
407:   # Run the Healthy-Sick-Sicker-Dead microsimulation PSA in parallel over one Linode clusters:
408:   cloud_parallel()
409:
410:   # 3. Run the Healthy-Sick-Sicker-Dead microsimulation PSA in parallel over one Linode clusters: t
411: 
```

Figure 32: Local R Session Vs Remote Linode Instance Terminal

The screenshot below shows the start of the execution of the R code. The second line shown in the console (in the screenshot below) represents the part of the

code where R establishes a connection with the Linode instance based on its IP Address.

```
R 4.3.3 - ~/Work/space/D PA/Courses/r4he_uic/
> public_ip <- c(
+   "139.162.233.246" # UIC_1_16GB_8core
+ )
>
> cl <- connect_to_linode(public_ip, ssh_private_key_file)
```

State	PID	User	PR	NI	VIRT	RES	SIRQ	SPC	WNL
RUNN	1	root	20	0	22100	12780	9324	5	0.0
RUNN	359	root	20	0	34944	13304	12488	5	0.0
RUNN	411	root	20	0	29924	7364	4894	5	0.0
RUNN	672	systemd-ne	20	0	18996	9088	8664	5	0.0
RUNN	675	root	20	0	8632	5012	1792	5	0.0
RUNN	676	systemd-re	20	0	19732	12544	10368	5	0.0
RUNN	677	systemd-tl	20	0	91020	7680	6784	5	0.0
RUNN	682	systemd-ti	20	0	91020	7680	6784	5	0.0
RUNN	707	root	19	-1	18136	8448	7552	5	0.0
RUNN	710	root	20	0	6824	2680	2560	5	0.0
RUNN	711	messagebus	20	0	9844	5120	4352	5	0.0
RUNN	747	root	20	0	1074	22654	13440	5	0.0
RUNN	752	syslog	20	0	217M	5760	4480	5	0.0
RUNN	760	syslog	20	0	217M	5760	4480	5	0.0
RUNN	761	syslog	20	0	217M	5760	4480	5	0.0
RUNN	762	syslog	20	0	217M	5760	4480	5	0.0
RUNN	790	root	20	0	1074	22654	13440	5	0.0
RUNN	829	root	20	0	5720	2176	2048	5	0.0
RUNN	832	root	20	0	6104	1792	1664	5	0.0
RUNN	863	root	20	0	12020	7936	6912	5	0.0
RUNN	2178	root	20	0	306M	8704	7552	5	0.0

Figure 33: Connecting to the Linode Instance

The parallel execution plan `future::plan`, the code in the console just below the remote connection line, represents the instructions for executing the PSA code on the Linode instance. This plan would apply to multiple clusters if more than one instance IP Address is saved in the “`public_ip`” object (each representing a cluster). This point is discussed further and showcased in the `cloud_parallel_remote.R` script.

0.6 Working on the Linode instance:

TABLE OF CONTENTS

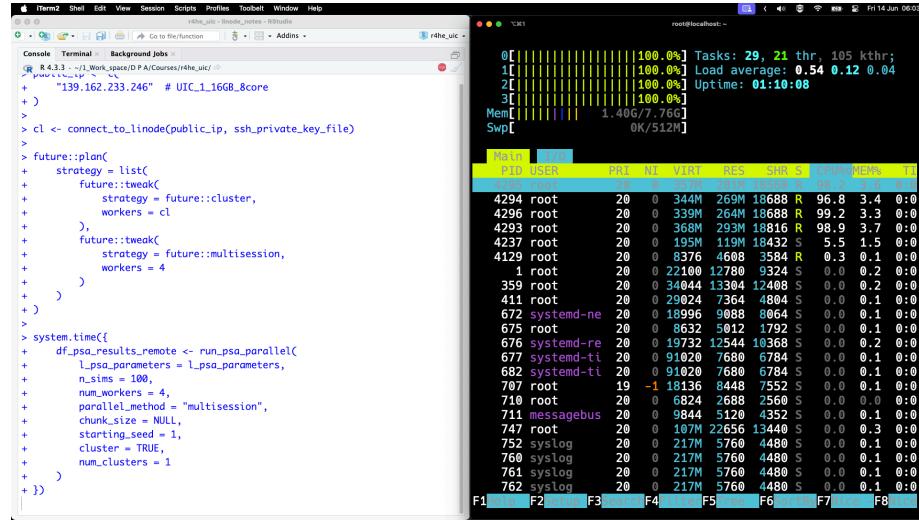


Figure 34: Running Local Code on Remote Instance

The execution of 100 PSA iterations from the local R session onto the Linode instance took around 77 seconds.

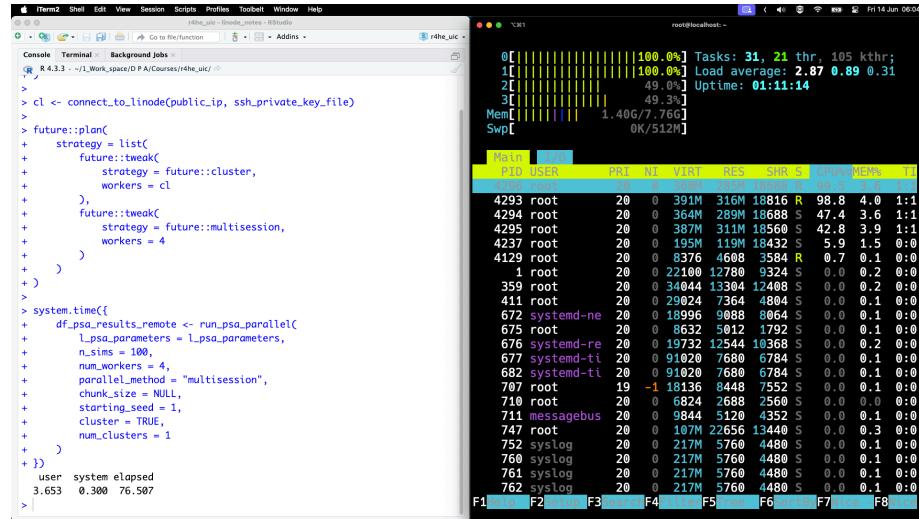


Figure 35: Returning Results from Remote Instance to Local R Session

After the parallel code execution finishes, the Linode instance becomes idle, and the R processes (running on the remote instance, seen in the `htop` charts in the previous screenshots) are terminated. The 'future' package controls the termination of the children R processes.

0.6 Working on the Linode instance:

TABLE OF CONTENTS

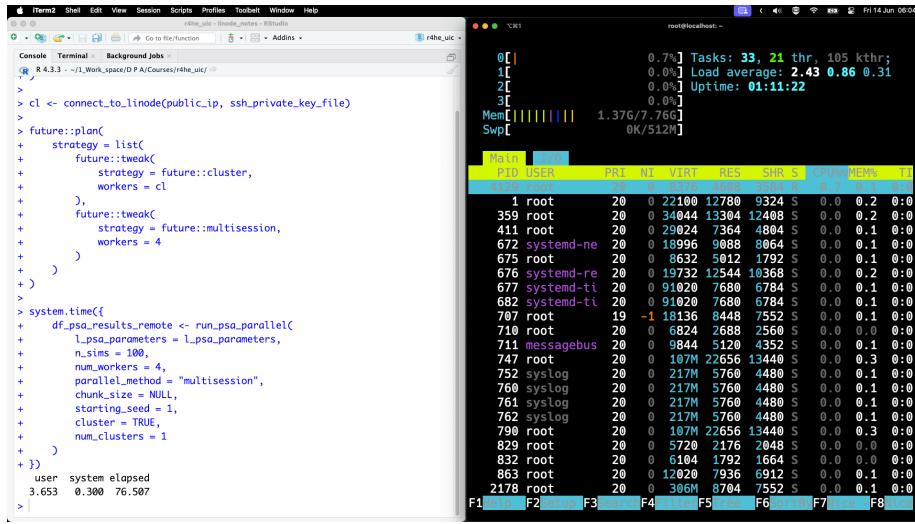


Figure 36: Linode Instance Idle

We hope you find the content useful!

Robert Smith, Wael Mohammed & Paul Schneider

Copyright © 2024 Dark Peak Analytics Ltd