

Comparative Study of RL Algorithms for Resource Optimization Scheduling in Kubernetes

MSc Research Project
Cloud Computing

Jay Shukla
Student ID: X23113111

National College of Ireland

Supervisor: Mr. Punit Gupta

Student Name:	Jay Milind Shukla
Student ID:	X23113111
Programme:	MSc. Cloud Computing
Year:	2023-24
Module:	MSc Research Project
Supervisor:	Mr. Punit Gupta
Submission Due Date:	12/08/2024
Project Title:	Comparative Study of RL Algorithms for Resource Optimization Scheduling in Kubernetes
Word Count:	2,128
Page Count:	6

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Jay Shukla
Date:	12-08-2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Jay Shukla
X23113111

1 AWS EC2 Instance

To run the Python3 files it is mandatory to launch the EC2 ubuntu instance. It can run on all linux platform. But the command changes according to the linux Flavour.

Follow the below steps to launch the EC2:

1. Login to AWS console.
2. Then Go to the EC2 instance service.
3. Click on Launch Instances.
4. So showing in the Fig. 1 it will open the window.
5. Just launch the EC2 instance.

So firstly launch the EC2 instance.

The screenshot displays the AWS 'Launch an instance' interface. The breadcrumb trail at the top indicates the path: EC2 > Instances > Launch an instance. The main heading is 'Launch an instance' with an 'Info' link. Below this, a brief description states: 'Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.'

The 'Name and tags' section contains a text input field with 'Ubuntu' and an 'Add additional tags' link. The 'Application and OS Images (Amazon Machine Image)' section includes a search bar with the placeholder text 'Search our full catalog including 1000s of application and OS images'. Below the search bar are tabs for 'Recents' and 'Quick Start'.

The 'Summary' panel on the right lists the following configuration details:

- Number of instances: 1
- Software Image (AMI): Canonical, Ubuntu, 24.04 LTS, ...read more (ami-0c38b837cd80f13bb)
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

A 'Free tier' notification is also present, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance'. At the bottom right of the summary panel are 'Cancel' and 'Launch instance' buttons.

Figure 1: Ubuntu Instance Launch From Ec2

So once instance launch then you will need below details of the instance for login. Make sure the port 22 can be open in the Security Group. Security group should be look like added Fig. 2.

<input type="text" value="Filter rules"/>				
Name	Security group rule ID	Port range	Protocol	Source
-	sgr-0fe22896445b334d5	80	TCP	0.0.0.0/0
-	sgr-06fc0b50f037238e2	443	TCP	0.0.0.0/0
-	sgr-0349625d26514510d	22	TCP	0.0.0.0/0

Figure 2: Security Group

2 Instance access with putty

1. To take the instance access we need putty tool on the machine.
2. Putty tool looks like which shows in the Fig. 3.
3. Then we need to take the access to the Ec2 instance for that we need the IP address of the instance.
4. The Fig. 4 shows the IP address and that needs to put in the putty.
5. Then take the access to the putty. It will show the terminal as shown in the Fig. 5.
6. Then add all the files on the terminal.
7. Then run the requirements.txt. For that below is the command.

```
pip install -r requirements.txt
```

Fig 6. shows the exact output for the command.

(Before installing pip need to run below command.)

```
apt-get update
```

8. Then once install all the packages just run below command and that will run the simulation.

```
python3 filename.py
```

References

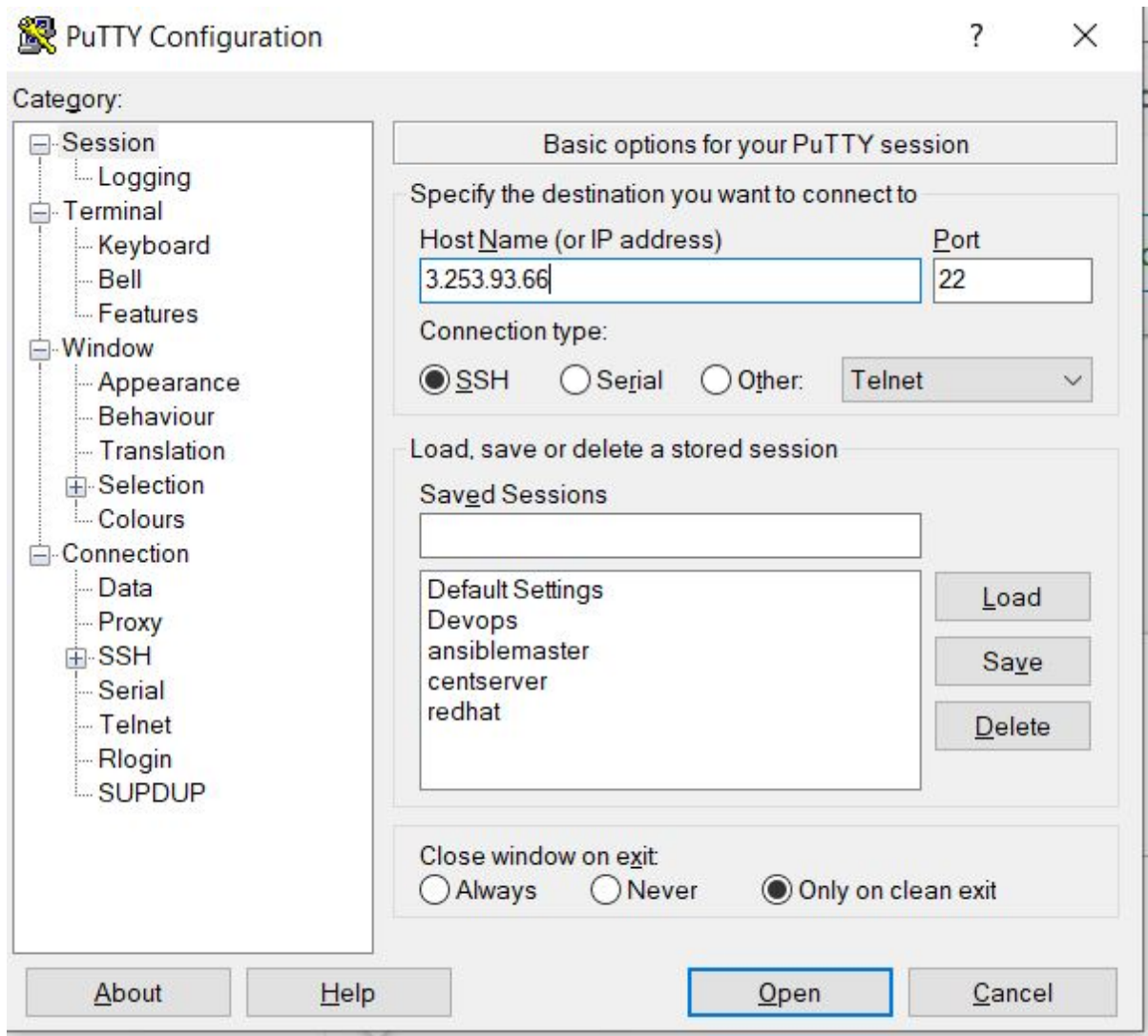


Figure 3: Putty for windows

Instance ID i-07f3355728cfa85fc (Ubuntu)	Public IPv4 address 3.253.93.66 open address
IPv6 address -	Instance state Running
Hostname type IP name: ip-172-31-16-170.eu-west-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-16-170.eu-west-1.compute.internal
Answer private resource DNS name IPv4 (A)	Instance type t2.micro
Auto-assigned IP address -	VPC ID vpc-00e21fc653cd5a6e5 (Default-VPC)
Subnet ID	

Figure 4: IP address Section

```

root@ip-172-31-16-170: /home/ubuntu#

```

Figure 5: Putty Terminal

```

See /usr/share/doc/python3.12/README.venv for more information.
note: If you believe this is a mistake, please contact your Python installation or OS distribution provider. You can override this, at the r
ur Python installation or OS, by passing --break-system-packages.
hint: See PEP 668 for the detailed specification.
root@ip-172-31-16-170:/home/ubuntu# pip install -r requirements.txt --break-system-packages
Collecting certifi==2024.7.4 (from -r requirements.txt (line 1))
  Downloading certifi-2024.7.4-py3-none-any.whl.metadata (2.2 kB)
Collecting charset-normalizer==3.3.2 (from -r requirements.txt (line 2))
  Downloading charset-normalizer-3.3.2-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (33 kB)
Collecting cloudpickle==3.0.0 (from -r requirements.txt (line 3))
  Downloading cloudpickle-3.0.0-py3-none-any.whl.metadata (7.0 kB)
Collecting contourpy==1.2.1 (from -r requirements.txt (line 4))
  Downloading contourpy-1.2.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.8 kB)
Collecting cycler==0.12.1 (from -r requirements.txt (line 5))
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting Farama-Notifications==0.0.4 (from -r requirements.txt (line 6))
  Downloading Farama-Notifications-0.0.4-py3-none-any.whl.metadata (558 bytes)
Collecting filelock==3.15.4 (from -r requirements.txt (line 7))
  Downloading filelock-3.15.4-py3-none-any.whl.metadata (2.9 kB)
Collecting fonttools==4.53.1 (from -r requirements.txt (line 8))
  Downloading fonttools-4.53.1-cp312-cp312-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2014_x86_64.whl.metadata (16
----- 162.6/162.6 kB 9.7 MB/s eta 0:00:00
Collecting fsspec==2024.6.1 (from -r requirements.txt (line 9))
  Downloading fsspec-2024.6.1-py3-none-any.whl.metadata (11 kB)
Collecting gym==0.26.2 (from -r requirements.txt (line 10))
  Downloading gym-0.26.2.tar.gz (721 kB)
----- 721.7/721.7 kB 54.6 MB/s eta 0:00:00
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Collecting gym-notices==0.0.8 (from -r requirements.txt (line 11))
  Downloading gym-notices-0.0.8-py3-none-any.whl.metadata (1.0 kB)
Collecting gymnasium==0.29.0 (from -r requirements.txt (line 12))
  Downloading gymnasium-0.29.0-py3-none-any.whl.metadata (10 kB)
Collecting idna==3.7 (from -r requirements.txt (line 13))
  Downloading idna-3.7-py3-none-any.whl.metadata (9.9 kB)
Collecting Jinja2==3.1.4 (from -r requirements.txt (line 14))
  Downloading Jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting kiwisolver==1.4.5 (from -r requirements.txt (line 15))
  Downloading kiwisolver-1.4.5-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.4 kB)
Requirement already satisfied: MarkupSafe==2.1.5 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 16)) (2.1.5)
Collecting matplotlib==3.9.1 (from -r requirements.txt (line 17))

```

Figure 6: Run requirement.txt file

```

:/home/ubuntu/jay1# python3 gymhpa_cost_optimised.py
3.10/dist-packages/stable_baselines3/common/vec_env/patch_gym.py:49: UserWarning: Using a
to Gymnasium environments. Stable-Baselines3 is automatically wrapping you
a `Monitor` wrapper
DummyVecEnv.
-----
| 1048 |
| 1 |
| 1 |
| 2048 |
-----
| 795 |
| 2 |
| 5 |
| 4096 |
| 0.003029359 |
| 0 |
| 0.2 |
| -1.1 |
ce | -5.77e-05 |
| 0.0003 |
| 9.56e+06 |
| 10 |
loss | -0.00239 |
| 1.83e+07 |
-----
| 751 |
| 3 |
| 8 |
| 6144 |
| 0.0038546824 |

```

Figure 7: Running Simulation