National College of Ireland

A Reinforcement Learning Approach to Real-Time, Cost-Aware Kubernetes Auto-Scaling

MSc Research Project
MSc Cloud Computing

Mageshwaran Kumaresan
Student ID: x23216522

School of Computing
National College of Ireland

Supervisor: Yasantha Samarawickrama

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Mageshwaran Kumaresan
Student ID:	x23216522
Programme:	MSc Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Yasantha Samarawickrama
Submission Due Date:	11/08/2025
Project Title:	A Reinforcement Learning Approach to Real-Time, Cost-Aware Kubernetes Auto-Scaling
Word Count:	XXX
Page Count:	5

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:	Mageshwaran Kumaresan
Date:	11th August 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
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.	☐

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):	

A Reinforcement Learning Approach to Real-Time, Cost-Aware Kubernetes Auto-Scaling

Mageshwaran Kumaresan
x23216522

1 Setting Up a Kubernetes Cluster

- Set up a Kubernetes cluster with Minikube, Kind, or a managed service such as EKS, GKE , AKS or using a container platform called Docker.
- Verify the cluster using the below command:

kubectl cluster-info

2 Install the Prometheus In the local Environment

- Go to the repository where you cloned the code file.
- Enter the below command in the command promnt or in the wsl(Windows Sub linux) or in any terminalto download the prometheus and kube-prometheus-stack charts, which are widely used for deploying comprehensive monitoring stacks.

helm repo add prometheus-community https://prometheus-community.github.io/helm-charts

- Enter the below command to Install Prometheus in the `monitoring` namespace:

helm install prometheus prometheus-community/prometheus –namespace monitoring –create-namespace

- Start up Prometheus at `http://localhost:9090` (or setup port-forwarding).
- Verify the access to the Prometheus dashboard.

kubectl port-forward svc/prometheus-server 9090:80 –namespace monitoring

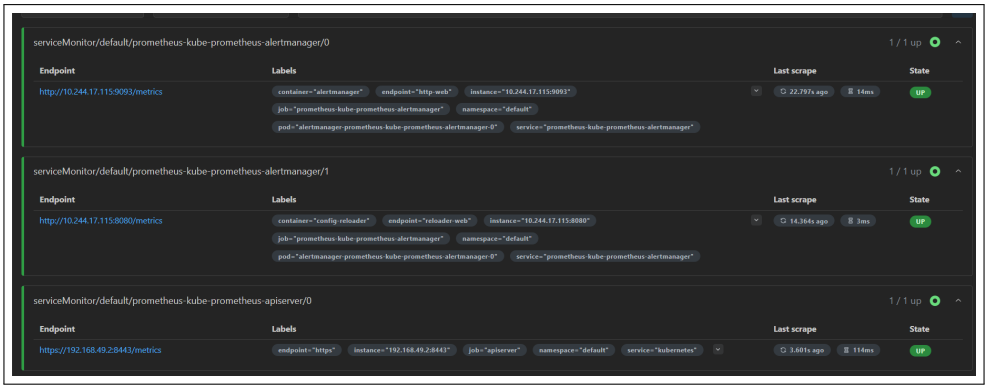

Figure 1: Prometheus dashboard showing the from the local deployement metrics

3 Deploy the Nginx server in the local Environment

- Create the nginx deployment in the namespace which we have created(monitoring):

kubectl create deployment nginx –image=nginx –namespace default

- Set up the initial replica count to 3 by using the bellow command:

kubectl scale deployment nginx –replicas=3

4 Install Python Dependencies in the local Environment

- Create and activate a new virtual environment:

python3 -m venv venv source venv/bin/activate

- Install the Python packages in the newly created environment by using the blow command:

pip install tensorflow==2.10 keras==2.10 keras-rl2 gym prometheus-api-client kubernetes numpy pandas matplotlib dask

5 Configure the Environment Variables

- Ensure the environment variables is set the implementation script:

```
import os
os.environ["PROM_URL"] = "http://localhost:9090"
os.environ["TARGET_DEPLOYMENT"] = "nginx"
os.environ["TARGET_NAMESPACE"] = "default"
```

6 Run the Implementation file

- Launch Jupyter Notebook by using the below command in the terminal.

jupyter notebook

- Open `ML.ipynb` and execute cells sequentially or you can excute it in all in one step by using the Run all button.
- Ensure Prometheus is collecting:
  - `container_cpu_usage_seconds_total`
  - `container_memory_working_set_bytes`
  - `http_request_duration_seconds_bucket`
- Monitor simulation output:
  - Pod counts and latency penalties per minute
  - SLA violation rates and costs
  - Output will be saved to excel `hpa_vpa_rl_simulation.csv`

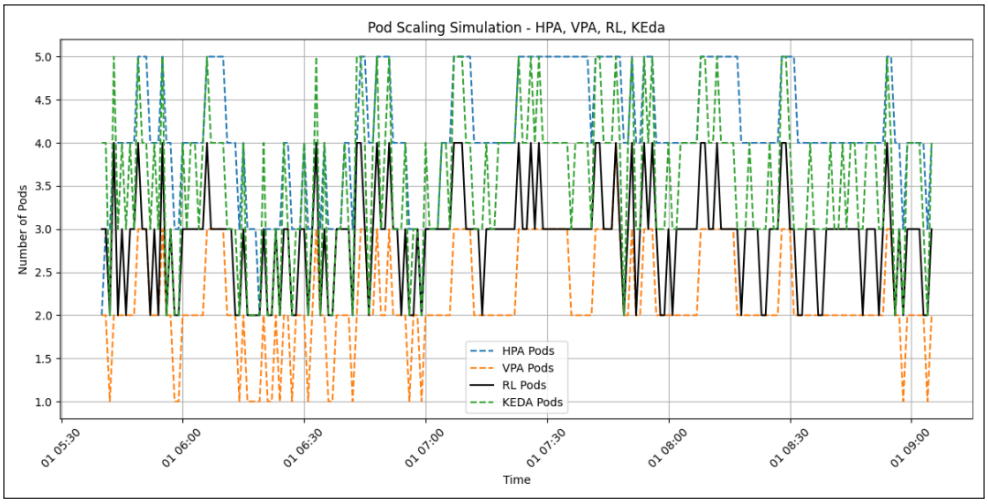

Figure 2: Pod scaling comparison for HPA, VPA, RL, and KEDA over time.

7 Troubleshooting

Prometheus Query Errors

- In case of unavailable metrics, see Prometheus scraping configuration.
- Ensure nginx metrics are exposed (via exporter if needed).

Kubernetes API Errors

- Make sure script is executed within a service account pod.

Model Errors

- Verify `dqn_streaming_model.h5` compatibility with:
  - `tensorflow==2.10`
  - `keras==2.10`

8 Best Practices

- Monitor Prometheus frequently to see whether the metric is available
- Backup `dqn_streaming_model.h5` to avoid retraining.
- Validate `synthetic_load_profile_60_chunks.csv` for realistic RPS values.
- Adjust `SLA_MS` and `KEDA_THRESHOLDS` to match workload behavior.

References

[1] Cloud Native Computing Foundation (CNCF) (2025). *Kubernetes Documentation*. Available at: `https://kubernetes.io/docs/`

[2] Prometheus Authors (2025). *Prometheus Documentation*. Available at: `https://prometheus.io/docs/`

[3] Prometheus Community (2025). *kube-prometheus-stack Helm Chart Documentation*. Available at: `https://github.com/prometheus-community/helm-charts`

[4] NGINX Inc. (2025). *NGINX Documentation*. Available at: `https://nginx.org/en/docs/`

[5] Python Software Foundation (2025). *Python 3 Documentation*. Available at: `https://docs.python.org/3/`

[6] TensorFlow Authors (2025). *TensorFlow 2.10 API Documentation*. Available at: `https://www.tensorflow.org/api_docs`

[7] Keras Team (2025). *Keras 2.10 Documentation*. Available at: `https://keras.io/api/`

[8] Plappert, M. (2020). *keras-rl2 Documentation*. Available at: `https://github.com/wau/keras-rl2`

[9] OpenAI (2025). *Gym Documentation*. Available at: `https://gymnasium.farama.org/`

[10] Prometheus API Client Authors (2025). *prometheus-api-client Documentation*. Available at: `https://github.com/AICoE/prometheus-api-client-python`

[11] Kubernetes Python Client Authors (2025). *Python Client for Kubernetes Documentation*. Available at: `https://github.com/kubernetes-client/python`

[12] NumPy Developers (2025). *NumPy Documentation*. Available at: https://numpy.org/doc/

[13] Pandas Development Team (2025). *pandas Documentation*. Available at: https://pandas.pydata.org/docs/

[14] Matplotlib Development Team (2025). *Matplotlib Documentation*. Available at: https://matplotlib.org/stable/contents.html

[15] Dask Developers (2025). *Dask Documentation*. Available at: https://docs.dask.org/en/stable/