



National
College of
Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Vikrant Sonawane
Student ID: x21210403

School of Computing
National College of Ireland

Supervisor: Rashid Mijumbi

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: ...Vikrant Sonawane.....

Student ID: ...x21210403.....

Programme: ...MSc. Cloud Computing..... **Year:** ...2023.....

Module: ...Research Project.....

Lecturer:

Submission Due Date: ...14/07/2023.....

Project Title: ...Serverless Auto-scaling mechanism using Reinforcement learning.....

Word Count:6430..... **Page Count:**17.....

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:Vikrant Sonawane.....

Date:14/07/2023.....

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

Vikrant Sonawane
Student ID: x21210403

1 Tools Used

The following are the tools used for the development of this research:

1. Pycharm (Python application development and testing)
2. Python
3. OpenAI Gym
4. Docker
5. Kubernetes
6. Knative
7. Ingress
8. Prometheus and Grafana
9. Anaconda (Jupyter Labs)

2 Setup Process

After installing the above software, we can work on setting up the project:

The project is uploaded to git and linked here (Sonawane, n.d.)

The folder structure of the project and description is as follows:

```
+ hello-python
+ ingress-nginx
+ knative
+ knative-docs
+ prometheus-config
+ rl-implmentation
```

Python Application:

- To create a Python application image, we can run the Docker file.
- But the Python application image is already present in the Docker hub. It is linked here (Sonawane, n.d.).
- Based on this Python application, we have created a Kubernetes service using the deployment.yaml file present in **research-project\hello-python\app** folder.

Ingress Configuration:

- For setting up ingress we need to follow steps linked here (Ingress, 2023)
- Install helm using the link (HELM, n.d.)
- Create Kubernetes namespace:
`kubectl create namespace ingress-nginx`
- Add the ingress repository.
`helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx`
- Install nginx-ingress through Helm
`helm install ingress-controller ingress-nginx/ingress-nginx --namespace ingress-nginx`

Install Prometheus:

- **Setup prometheus**
kubect1 apply --kustomize github.com/kubernetes/ingress-nginx/dep1oy/prometheus/
- **Reconfigure nging**
helm upgrade ingress-controller ingress-nginx/ingress-nginx \ -- namespace ingress-nginx \ --set controller.metrics.enabled=true \ --set-string controller.podAnnotations."prometheus\.io/scrape"="true" \ --set-string controller.podAnnotations."prometheus\.io/port"="10254"
- **create ingress for hello-python Find ingress manifest at path and deploy it:**
kubect1 apply -f ingress-example.yaml
- **create hpa for hello-python app**
kubect1 autoscale deployment hello-python --cpu-percent=80 --min=1 --max=3

Knative Configuration:

- **Install the required custom resources by running the command:**
kubect1 apply -f <https://github.com/knative/serving/releases/download/knative-v1.11.0/serving-crds.yaml>
- **Install the required custom resources by running the command:**
kubect1 apply -f <https://github.com/knative/serving/releases/download/knative-v1.11.0/serving-core.yaml>
- **Install the Knative Kourier controller by running the command:**
kubect1 apply -f <https://github.com/knative/net-kourier/releases/download/knative-v1.11.1/kourier.yaml>
- **Configure Knative Serving to use Kourier by default by running the command:**
kubect1 patch configmap/config-network \ --namespace knative-serving \ --type merge \ --patch '{"data":{"ingress-class":"kourier.ingress.networking.knative.dev"}}'
- **Fetch the External IP address or CNAME by running the command:**
kubect1 --namespace kourier-system get service courier
- **Verify:**
kubect1 get pods -n knative-serving
- **Publish the python application:**
kubect1 apply research-project\knative\service.yaml

Install Anaconda:

To install Anaconda, a popular distribution for Python and data science packages, follow these steps:

- **Download Anaconda Installer:** Go to the official Anaconda download page: <https://www.anaconda.com/products/distribution> Choose the appropriate installer for your operating system (Windows, macOS, or Linux) and for the Python version you prefer (usually Python 3.x).
- **Run the Installer:** Once the installer is downloaded, navigate to the directory where it's located and execute the installer script. Replace <installer-filename> with the actual name of the installer you downloaded.
- **Follow the Installation Wizard:** The installation wizard will guide you through the installation process. You can choose the installation location, whether to add Anaconda to your system's PATH (recommended), and other preferences.
- **Verify Installation:** To verify that Anaconda is installed correctly, open a new terminal/command prompt window and type:
conda -version
- **Launch Jupyter labs**

- Upload "**research-project\rl-implementation\agents\env-discrete-state-discrete-action\env-discrete-state-discrete-action.ipynb**" in Jupyter labs and run the python code.

References

HELM. (n.d.). *HELM*. Retrieved from HELM: <https://helm.sh/docs/intro/install/>

Ingress, N. (2023). *NGINX Ingress* . Retrieved from NGINX Ingress : Installation with Manifests | NGINX Ingress Controller. (n.d.). Installation With Manifests | NGINX Ingress Controller. <https://docs.nginx.com/nginx-ingress-controller/installation/installation-with-manifests/>

Sonawane, V. (n.d.). Retrieved from <https://github.com/Vikrant2691/research-project>

Sonawane, V. (n.d.). *docker*. Retrieved from <https://hub.docker.com/r/vison91/helloworld-python>