National College of Ireland

Configuration Manual

MSc Research Project
Programme Name

Amal Abi
Student ID: x23274719

School of Computing
National College of Ireland

Supervisor: Khadija Hafeez

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……Amal Abi……

Student ID: …… x23274719……

Programme: ……MSc Cyber Security…… **Year:** ……2024-25

Module: ……MSc Research Project……

Lecturer: …… Khadija Hafeez ……

Submission Due Date: ……11/08/2025……

Project Title: …… Zero Trust for Cloud-Native Web Apps……

Word Count: …487……..…… **Page Count:** ……4……..….

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: ……Amal Abi……

Date: ……15/09/2025……

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):	

Configuration Manual

Amal Abi
Student ID: x23274719

1 Prerequisites

1.1 Hardware and OS

- Host machine with at least: ○ 2 CPUs
  - 4 GB RAM
  - 10 GB free disk space
- OS: Windows 10/11, macOS, or Linux

1.2 Software Requirements

- **Minikube** (latest stable version)
- **kubectl** (CLI for Kubernetes)
- **Container runtime**: Docker or built-in Minikube runtime
- **CNI Plugin**: Calico (required for NetworkPolicy enforcement)

2 Environment Setup

1. **Install Minikube** and `kubectl` as per your OS instructions.
2. **Start Minikube with Calico CNI**:

   `minikube delete minikube start --cni=calico` This ensures NetworkPolicies will be enforced.

3. **Verify cluster status**:

```
kubectl get nodes
kubectl get pods -A
```

3 Application Deployment

3.1 Deploy NGINX Web Server

1. Apply the deployment:

```
kubectl apply -f nginx-deployment.yaml
```

This launches an NGINX container listening on port 80.

2. Apply the ClusterIP service: `kubectl apply -f nginx-service.yaml`

   This exposes the NGINX deployment internally within the cluster.

3. Verify:

```
kubectl get svc,pods
kubectl rollout status deploy/nginx-deployment
```

4 Zero Trust Policy Enforcement

4.1 Apply NetworkPolicy

The `nginx-network-policy.yaml` file enforces Zero Trust ingress control:

```
kubectl apply -f nginx-network-policy.yaml
```

- **Selector:** Targets pod with `app=nginx`
- **Rule:** Allows ingress only from pods labelled `role=trusted`
- All other pods are implicitly denied.

5 Test Pod Deployment

1. **Trusted Pod** (allowed by policy): `kubectl`

   `apply -f trusted-busybox.yaml`

2. **Untrusted Pod** (denied by policy): `kubectl`

   `apply -f untrusted-busybox.yaml`

3. **Unlabeled Pod** (denied by policy): `kubectl`

   `apply -f busybox-pod.yaml`

4. Verify all pods: `kubectl get pods --`

   `show-labels`

6 Functional Testing

6.1 From Trusted Pod (Expected: Success)

```
kubectl exec -it trusted-busybox -- sh -c 'wget --spider --timeout=2
http://nginx-service && echo TRUSTED_OK'
```

6.2 From Untrusted and Unlabeled Pods (Expected: Denied)

```
kubectl exec -it untrusted-busybox -- sh -c 'wget --spider --timeout=2
http://nginx-service || echo UNTRUSTED_BLOCKED'

kubectl exec -it busybox -- sh -c "wget --spider --timeout=2
http://nginxservice || echo UNLABELED_BLOCKED"
```

6.3 Server-Side Log Validation

Check NGINX logs: `kubectl logs deploy/nginx-`

`deployment | tail -n 50`

Expected: Only `trusted-busybox` connections appear (HTTP 200 responses). Blocked traffic leaves no logs, proving enforcement at the network layer.

7 Optional Attacker Simulation

Deploy the attacker pod to simulate malicious probing:

```
kubectl apply -f attacker-pod.yaml kubectl
logs -f attacker-pod
```

Expected: Continuous timeouts and connection refusals. No entries in NGINX logs.

8 Troubleshooting

Issue	Possible Cause	Solution
Trusted pod cannot connect	Wrong label or service name	Ensure pod has `role=trusted` and service is `nginx-service`
No blocking occurs	NetworkPolicy not enforced	Ensure Minikube started with `–cni=calico`
DNS errors in pods	Cluster-wide deny-all blocking DNS	Apply `allow-dns-egress.yaml`

9 Clean-Up

```
kubectl delete -f attacker-pod.yaml --ignore-not-found
 kubectl delete -f busybox-pod.yaml -f untrusted-busybox.yaml -f
trustedbusybox.yaml
```

```
kubectl delete -f nginx-network-policy.yaml -f nginx-service.yaml -f
nginxdeployment.yaml
 minikube
delete
```

10 Observations

The Zero Trust policy worked as expected: only labelled trusted pods accessed the protected service.

Untrusted and unlabeled pods were denied at the CNI layer, leaving no trace in application logs.

This limitation can be addressed in production by enabling CNI flow logging or Kubernetes audit logs.