Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

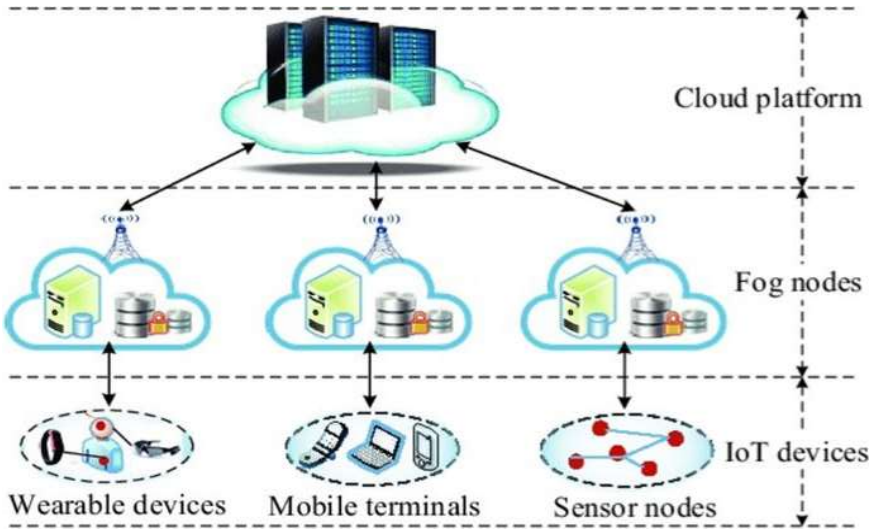
ECEN 404 Final Presentation
Team 21: Load Balancing on an IoT Fog

Sebastian Correa, Christopher Gonzales
TA: Zhuwen Hu
Sponsor: Swarnabha Roy

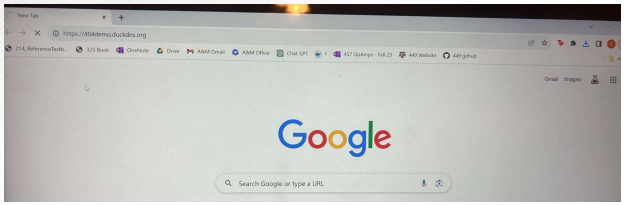
Project Summary

Problem statement:

Load balancing only on the cloud is inefficient for applications, overloads virtual machines, and creates scalability issues.

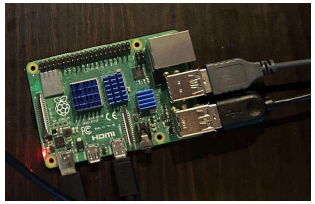


Integrated Subsystems Diagram

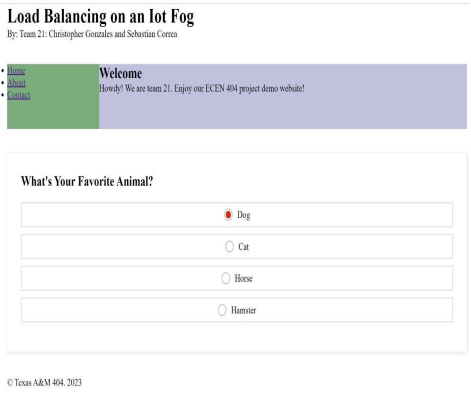


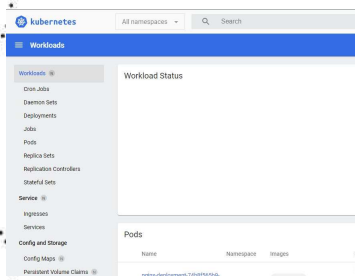
Users → Access → IoT Devices ↔ User Requests ↔ Cloud Layer

Sebastian Correa



Remote Access Gateway





Fog Layer

Christopher Gonzales

WireGuard VPN Connection

Netmaker Network

Edge Node Subsystem

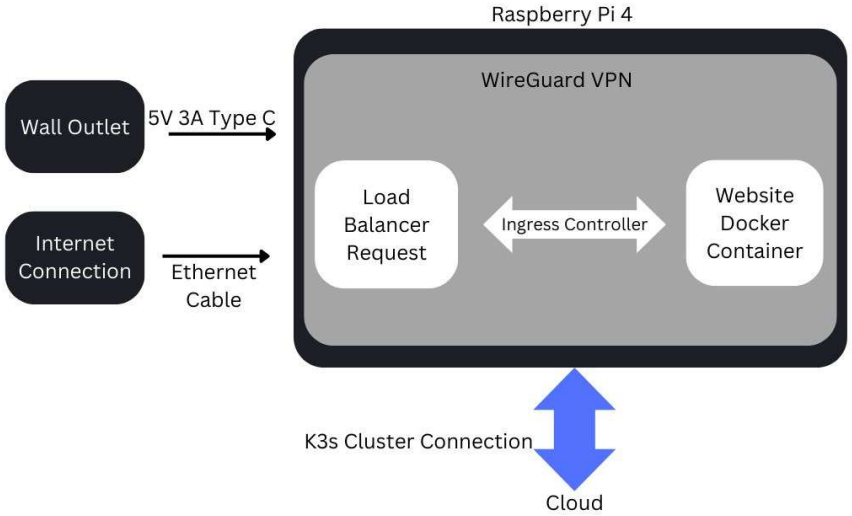
Christopher Gonzales

Function:

- Pi Takes Requests from cloud through k3s cluster
- Request is directed to website container pod
- Request directed back to cloud and user
- If the Pi shuts down then pods get allocated to other worker node

System Requirements:

- Raspbian OS
- Stable Internet Connection
- 5V 3A type C power Supply
- K3s, Docker, Wireguard



Edge Node Subsystem-Challenges/Solutions

Christopher Gonzales

Challenges

- Network and Port Forwarding Issues
- Connection to the k3s Cluster
- Hosting the website
- Accessing Cluster Nodes

Solutions

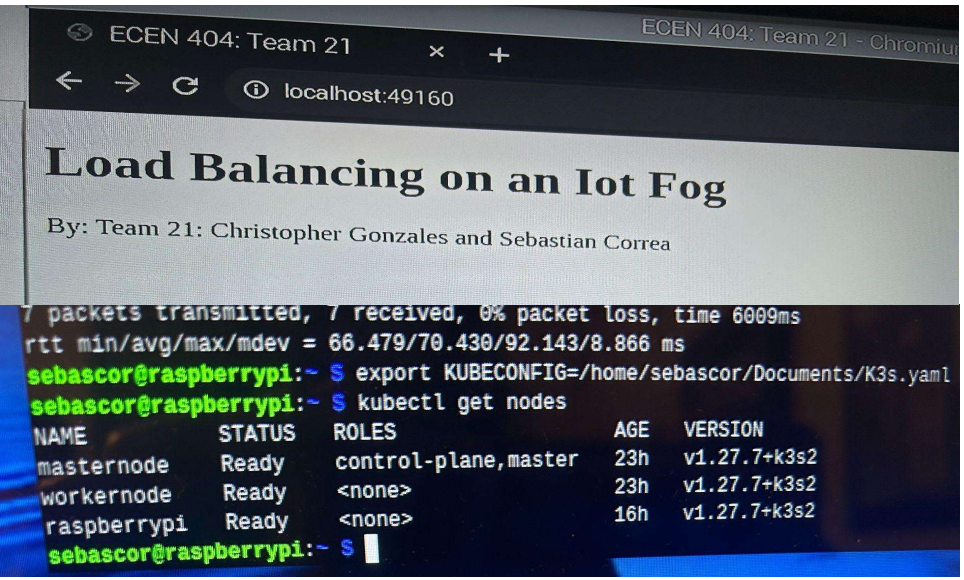
- Netmaker and Wireguard connection (Point-to-Point)
- Ping and Telnet
- Docker Containerization
- K3s.yaml file exported to the worker nodes

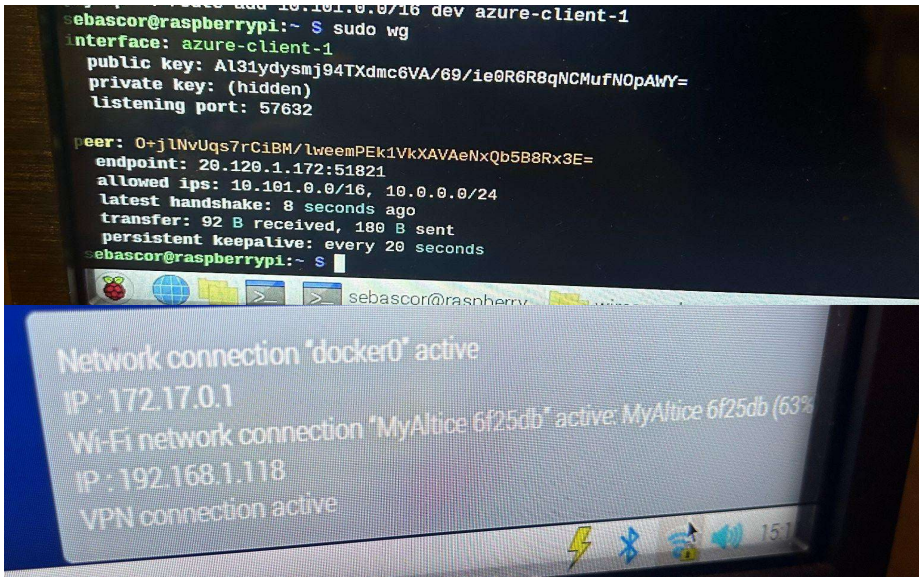
Edge Node Subsystem-Accomplishments

Christopher Gonzales

Accomplishments:

- Pi Hosts Website using ports specified
- VPN connection to the master node network
- Ping master node ip (average time = 70.4 ms)
- Connects to master node port specified
- Access Cluster Nodes From the Pi





Cloud Subsystem

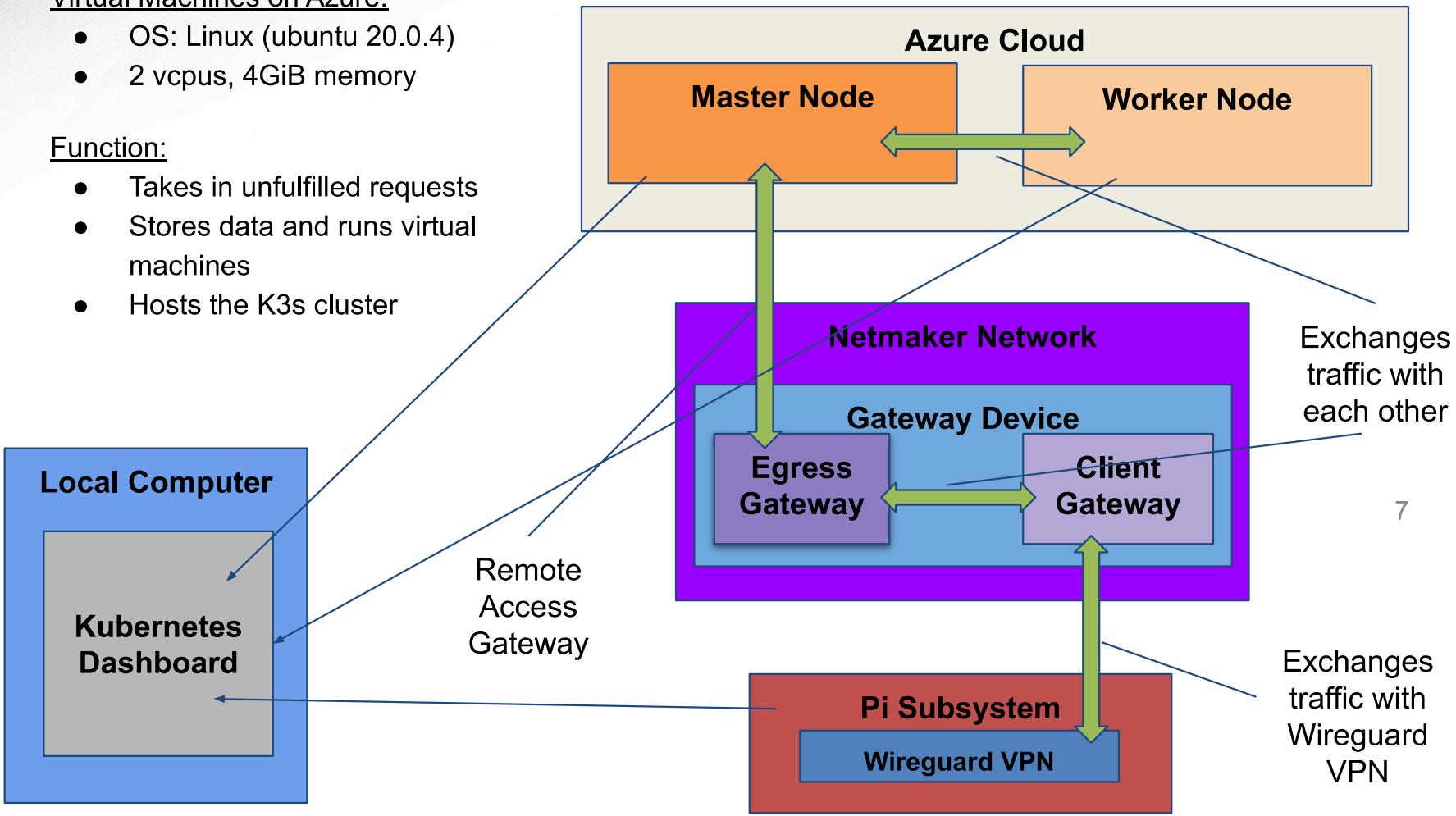
Sebastian Correa

Virtual Machines on Azure:

- OS: Linux (ubuntu 20.0.4)
- 2 vcpus, 4GiB memory

Function:

- Takes in unfulfilled requests
- Stores data and runs virtual machines
- Hosts the K3s cluster



Cloud Subsystem-Challenges/Solutions

Sebastian Correa

Challenges

- Connection with the Pi and the Azure Virtual Network
- Monitoring the K3s cluster
- Deploying the website to end users

Solutions

- Netmaker and Wireguard connection (Point-to-Point) connection
- Kubernetes Dashboard to monitor the cluster
- Nginx-Ingress-Controller to deploy website

Cloud Subsystem-Accomplishments

Sebastian Correa

Accomplishments:

- Switched to new cloud service
- Created VPN for Point-to-Point connection
- Applied new Load Balancer (Nginx Service with NodePort)
- Generated the Kubernetes Dashboard for monitoring
- Organized resources in designated namespaces

Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

Cloud Subsystem-Netmaker

Sebastian Correa

View All Networks

netmaker

Network Settings

Overview | Hosts (1) | Clients (2) | Egress (1) | DNS | Access Control | Graph

Search hosts

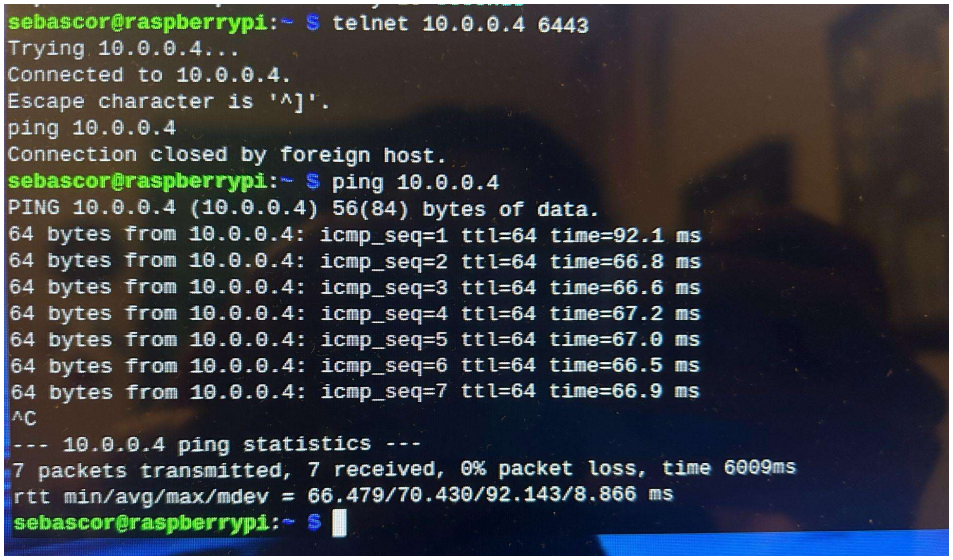
+ Add New Host

Host Name	Private Address (IPv4)	Public Address	Connectivity	Health Status
MasterNode	10.101.0.1/16	20.127.241.65	Connected	Healthy

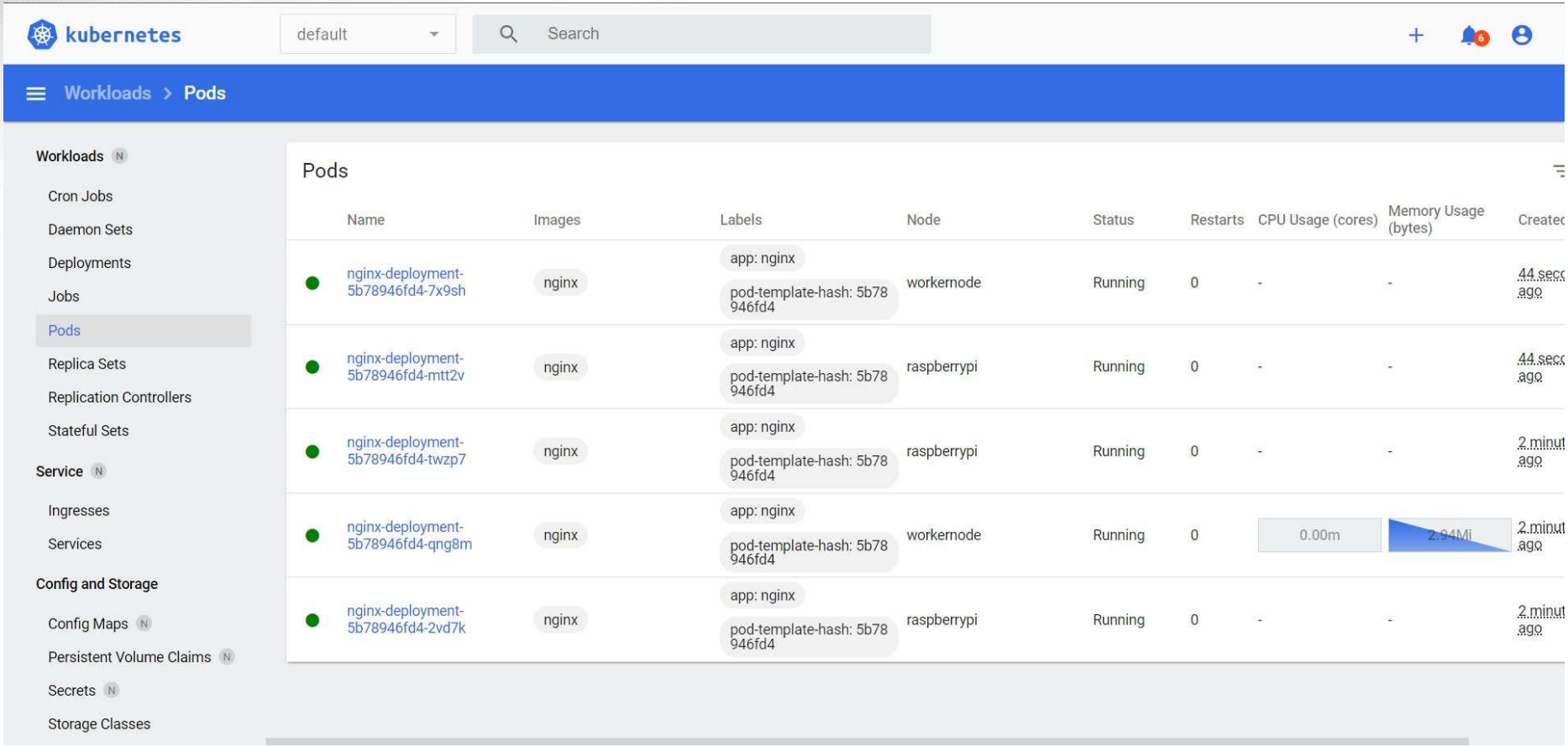
1

Connection Tests

```
C:\Users\sebas\Downloads\MasterNodeFiles>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=69ms TTL=64
Reply from 10.0.0.4: bytes=32 time=88ms TTL=64
Reply from 10.0.0.4: bytes=32 time=94ms TTL=64
Reply from 10.0.0.4: bytes=32 time=107ms TTL=64

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 69ms, Maximum = 107ms, Average = 89ms

C:\Users\sebas\Downloads\MasterNodeFiles>
```

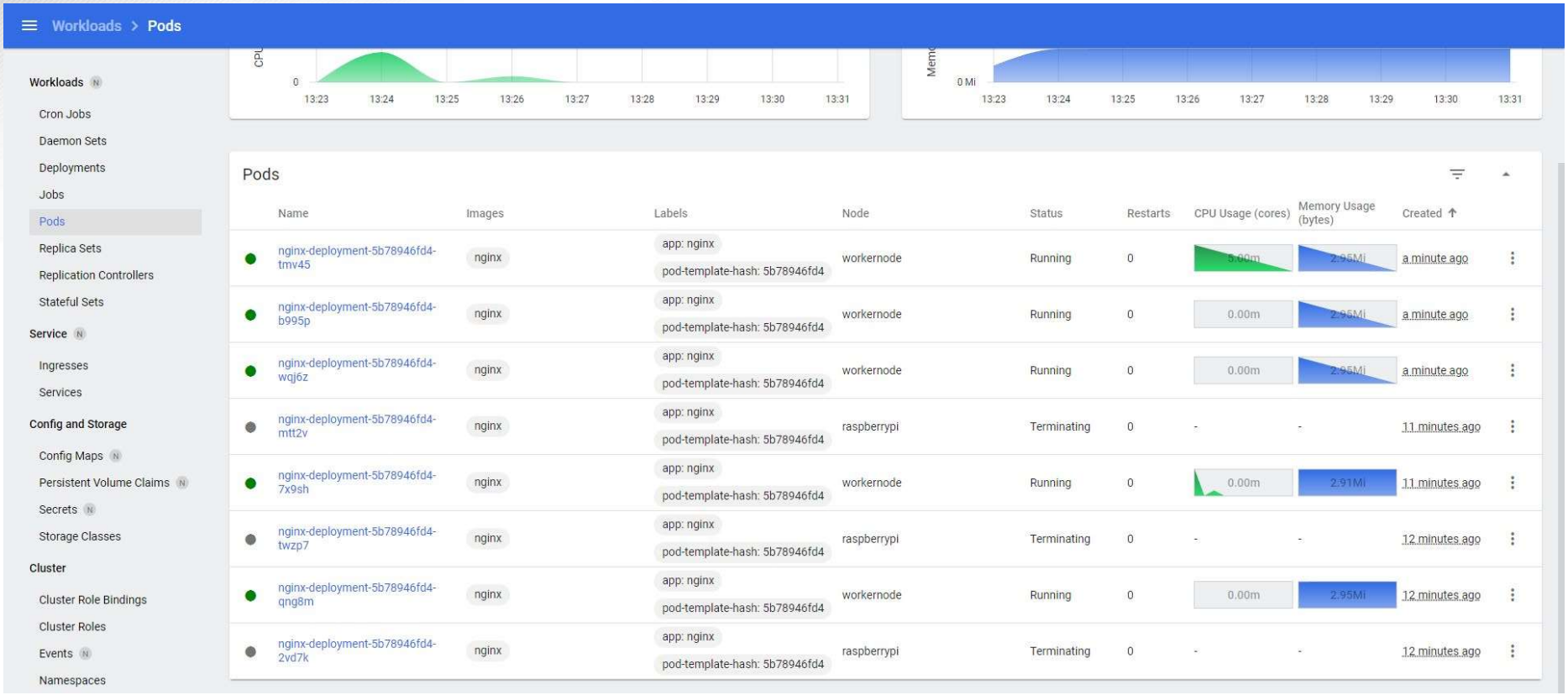


Pods Deployed



60% of pods distributed to the Pi 40% to VM Worker node

Fault Tolerance



Time to redistribute workload: 5 mins

Reading Traffic

```
sebascor@raspberrypi:~/Documents/403/website $ wrk -t2 -c50 -d30s http://192.16
8.1.118:30303/
Running 30s test @ http://192.168.1.118:30303/
  2 threads and 50 connections
  Thread Stats   Avg      Stdev     Max   +/- Stdev
    Latency     5.63ms    5.00ms  59.07ms   84.52%
    Req/Sec     1.25k   491.55     3.46k    73.91%
  64000 requests in 30.09s, 52.06MB read
Requests/sec:   2126.70
Transfer/sec:      1.73MB
```

Nginx Website

Conclusions

Changes

- Changed to website Demo
- Required Wireguard VPN Connection
- New Cloud Service

Current status:

- Integration Complete
- Load Balancer Not Fully Validated
- Demo website Not Fully Validated
- Validation to be Completed in 1 week

Questions?