Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

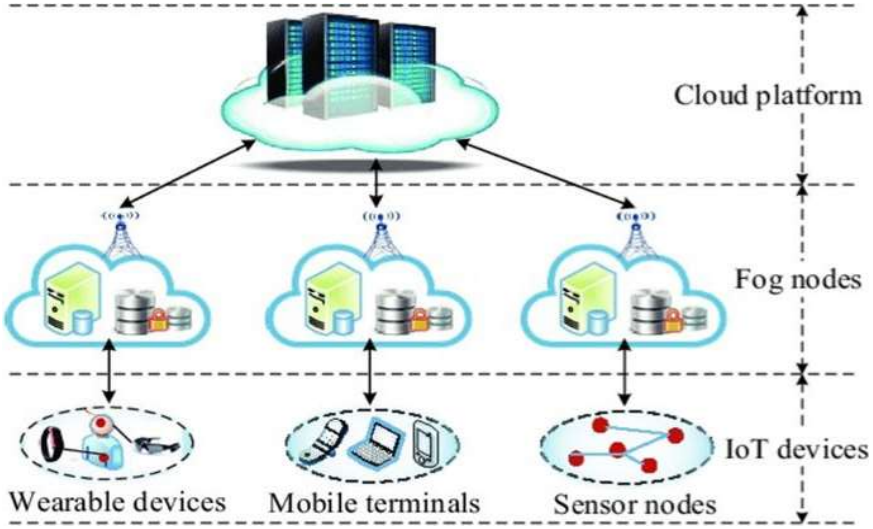
Team 21: Load Balancing on an IoT Fog
Bi-Weekly Update 2

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

Project Summary

Problem statement:

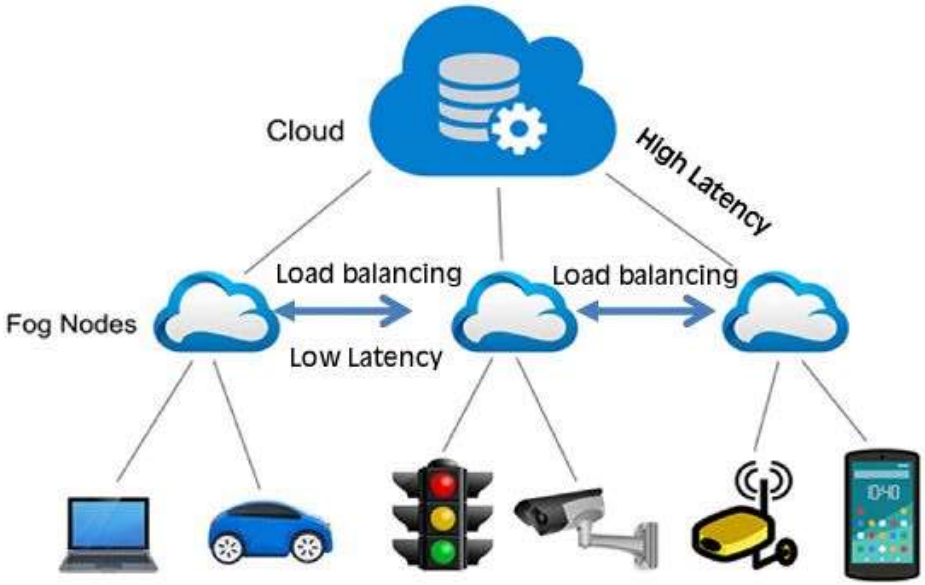
- IoT devices are becoming more common
- Fog Computing
- Need to distribute workload amongst edge nodes, servers, and IoT devices



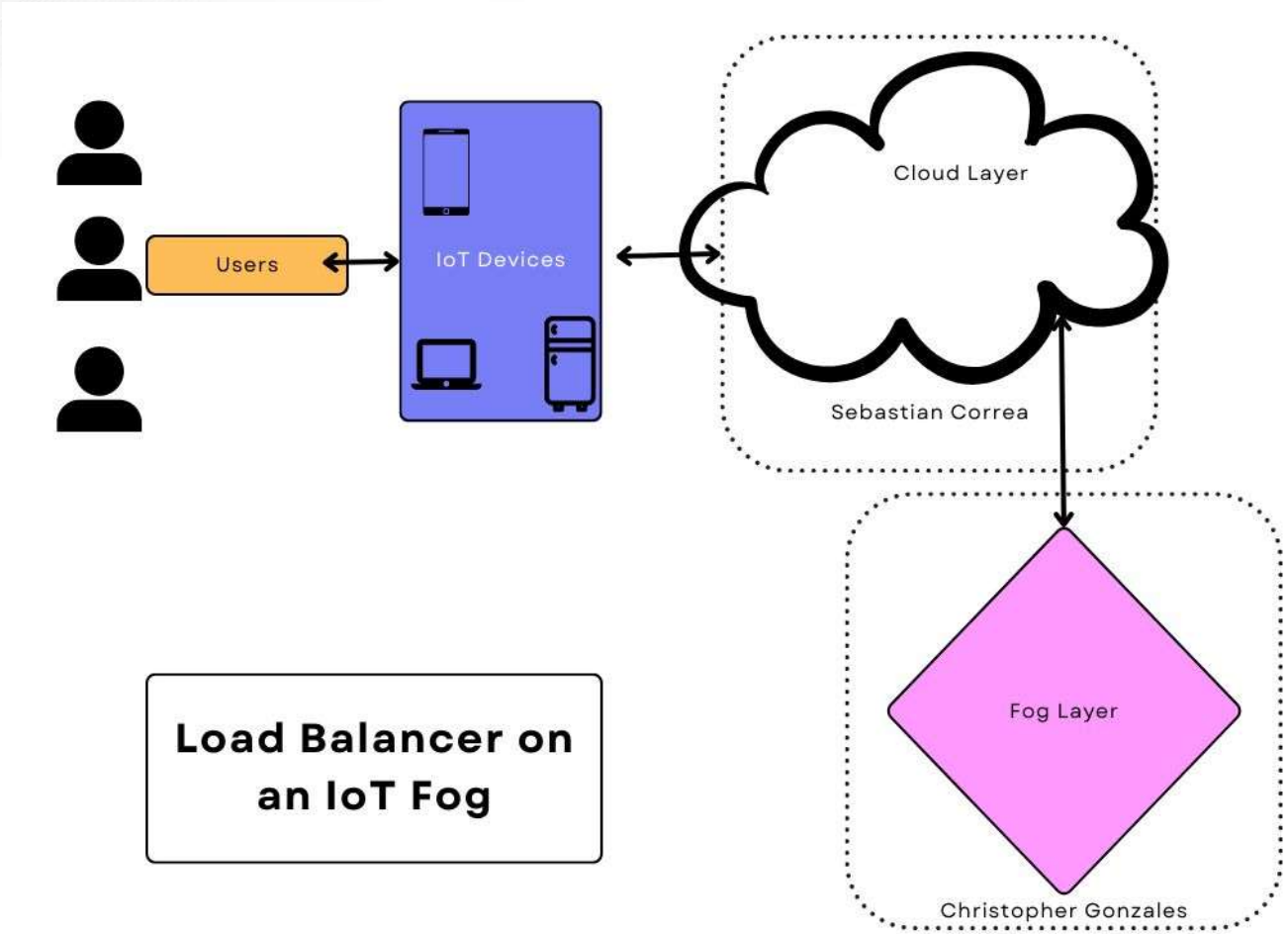
Project Summary

Load-Balancing Solution:

- Takes input from IoT devices
- Distributes the workload
- Sends data to the cloud



Subsystems Diagram



Project Timeline

Test Pi's condition and choose new cloud service (completed 8/31/23)	Test Website on Pi and convert VM's to new cloud service (completed 9/14)	Re-establish the connection between the Pi and the cloud (to complete by 9/28)	Finalize Integration (to complete by 10/12)	Test system's capabilities (to complete by 11/2)	Validation (To complete by 11/23)	Demo and Report (To complete by 12/4)

Edge Node Subsystem

Christopher Gonzales

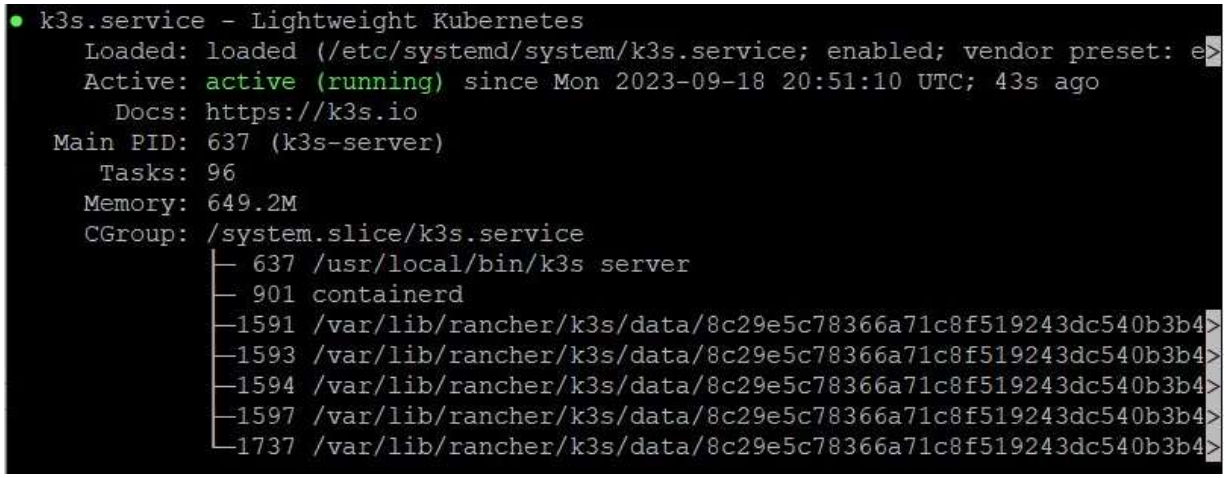
Accomplishments since 403 13 hrs of effort	Ongoing progress/problems and plans until the next presentation
- Created simple website code - Added Website Functionality - downloaded software to host website - Created Dockerfile for website	- Simulate Website Traffic to test website container locally - Register Pi as Worker Node in k3s cluster

Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

Cloud Subsystem

Sebastian Correa

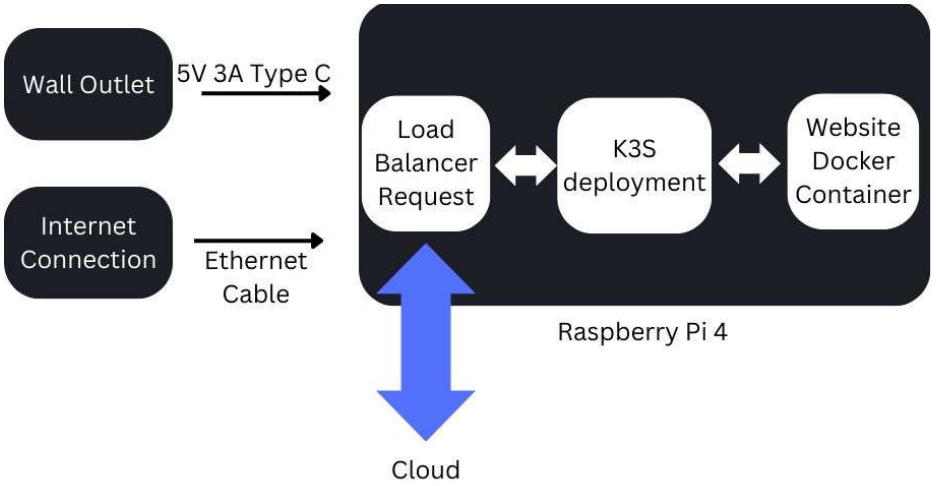
Accomplishments since 403 6-7 hrs of effort	Ongoing progress/problems and plans until the next presentation
• Created VM's in Microsoft Azure • Installed K3s on Master and worker nodes in VM's • Looked into KubEdge • Updated Master node	• Decide between Kubernetes Dashboard or Prometheus • Ongoing integration with Pi • Connect other nodes with master node



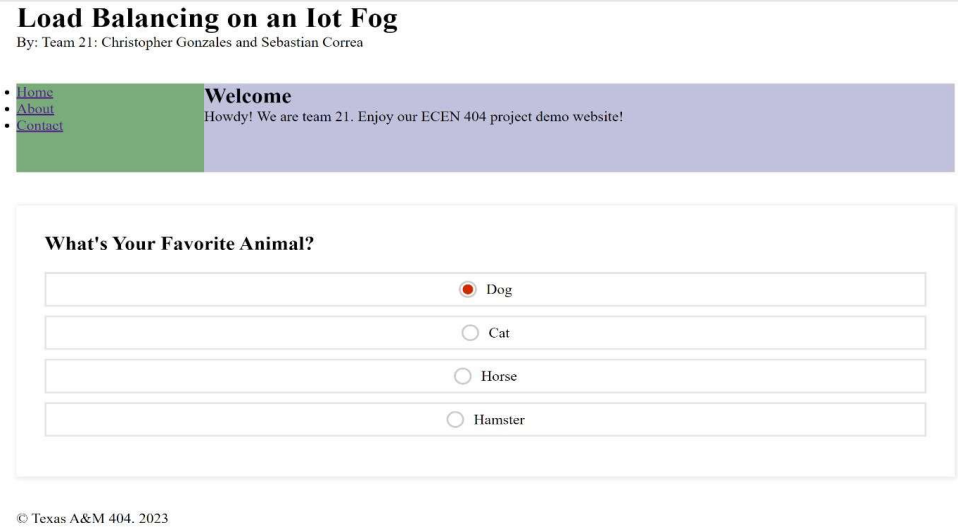
Edge Node Subsystem

Christopher Gonzales

- 5V 3A Type C Power Supply
- Ethernet Connection
- Load Balancer Hosted in the Cloud




```
sebascor@raspberrypi:~/403/website $ docker buildx build -t my-website --load -f website.dockerfile .
[+] Building 373.4s (11/11) FINISHED                                    docker-container:intelligent_ramanujan
 => [internal] load .dockerignore                                                                        0.1s
 => => transferring context: 2B                                                                          0.0s
 => [internal] load build definition from website.dockerfile                                             0.1s
 => => transferring dockerfile: 529B                                                                     0.0s
 => [internal] load metadata for docker.io/library/node:14                                               0.5s
 => [1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa   0.2s
 => => resolve docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa   0.1s
 => [internal] load build context                                                                        0.1s
 => => transferring context: 192B                                                                        0.0s
 => CACHED [2/5] WORKDIR /usr/src/app                                                                    0.0s
 => CACHED [3/5] COPY package*.json ./                                                                   0.0s
 => CACHED [4/5] RUN npm install                                                                         0.0s
 => CACHED [5/5] COPY . .                                                                                0.0s
 => exporting to docker image format                                                                   371.4s
 => => exporting layers                                                                                  1.1s
 => => exporting manifest sha256:52ea4cdd9ef2a4a71cedae3ef65659d0016f79c5e52ab6eacc1abcca474b2914        0.1s
 => => exporting config sha256:47b24b8a29d19e98d6b990c602e6005cf11ba12a80ff647e496a98732a39e353          0.1s
 => => sending tarball                                                                                 370.1s
 => importing to docker                                                                                274.7s
sebascor@raspberrypi:~/403/website $
```



Load Balancing on an Iot Fog

By: Team 21: Christopher Gonzales and Sebastian Correa

- Home
- About
- Contact

Welcome

Howdy! We are team 21. Enjoy our ECEN 404 project demo website!

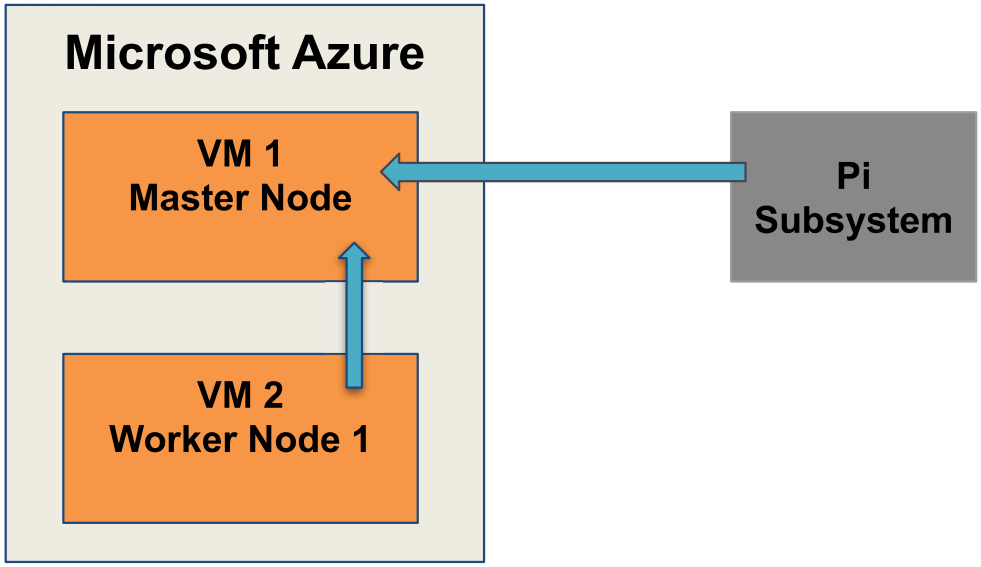
What's Your Favorite Animal?

- (•) Dog
- ( ) Cat
- ( ) Horse
- ( ) Hamster

Cloud Subsystem

Sebastian Correa

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



Execution plan

Work	End Date	Owner	Status	Completion Date
Check pi condition	8/24	Christopher	Complete	8/24
Choose new cloud service	8/24	Sebastian	Complete	8/24
Convert all virtual machines to Microsoft Azure	9/7	Sebastian	complete	9/11
Build website	9/7	Christopher	complete	9/7
Install K3s on VMs	9/14	Sebastian	complete	9/14
Test website	9/14	Christopher	Incomplete	9/11
Add more features to the website	9/21	Christopher	Incomplete	9/18
Create connections between VMs	9/21	Sebastian	Incomplete	N/A
Connect the Pi with the Cloud	9/28	Sebastian	Incomplete	N/A
Convert Pi connection into worker node	10/5	Sebastian	Incomplete	N/A
Configure containerization in Pi	10/5	Christopher	Incomplete	N/A
Finalize Integration	10/12	Christopher	Incomplete	N/A

Execution Plan

Work	End Date	Owner	Status	Completion Date
Test Fault Tolerance	10/19	Sebastian	Incomplete	N/A
Demonstrate movement of loads	10/26	Sebastian	Incomplete	N/A
Bug fix the VM's	11/2	Sebastian	Incomplete	N/A
Bug fix the website	11/2	Christopher	Incomplete	N/A
Configure cloud validation	11/9	Sebastian	Incomplete	N/A
Configure pi validation	11/9	Christopher	Incomplete	N/A
Finalize validation	11/23	Christopher	Incomplete	N/A

Validation plan

Task	Specification	Summary	Result	Owner
Cloud Response Time	<500ms	Amount of time it takes the Cloud to respond to Load Balancer		Sebastian
Edge Device Runtime	<500ms	Amount of time it takes the Edge Device to respond to Load Balancer		Christopher
Edge and Cloud Transmission Time	<500ms	Amount of time it takes for the edge device and cloud to respond to one another		Sebastian
Reading Traffic	<500ms	Amount of time it takes the K3s cluster to read the incoming traffic		Sebastian
Minimum Number of Test Cases	50	50 test cases, or traffic data, being sent to our system for testing		Christopher
Minimum Number of applications to run	1	Our application we're going to use for testing		Christopher
Load Balancing Test	40%-60%	Ensure that the load is distributed equally between rpi website container and VM container		Sebastian
Failover Test	<500 ms	Shut down one edge node and ensure traffic is redirected to working node within seconds		Christopher

Questions?