

MICHIGAN STATE

UNIVERSITY

Project Plan

Kubernetes Cluster Inspection Tool

The Capstone Experience

Team Google

Dave Ackley

Linghao Ji

Guillermo Jimenez

Haylee Quarles

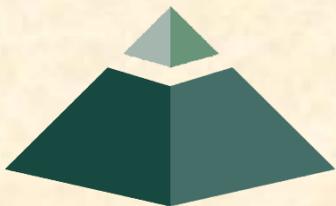
Casey Schneider

Ben Whitelaw

Department of Computer Science and Engineering

Michigan State University

Spring 2019



*From Students...
...to Professionals*

Functional Specifications

- Provide Robust Inspection Tool of Kubernetes Cluster
 - Both Current and Historical State
- Gather All Cluster Data into Single Interactive View
 - Provide Health, Performance, and Resource Changes
 - Ingest with Kubernetes Stackdriver Agent
 - Trace Lifecycle of Nodes, Pods and Containers
- Implement Easily on Any GCP Kubernetes Cluster

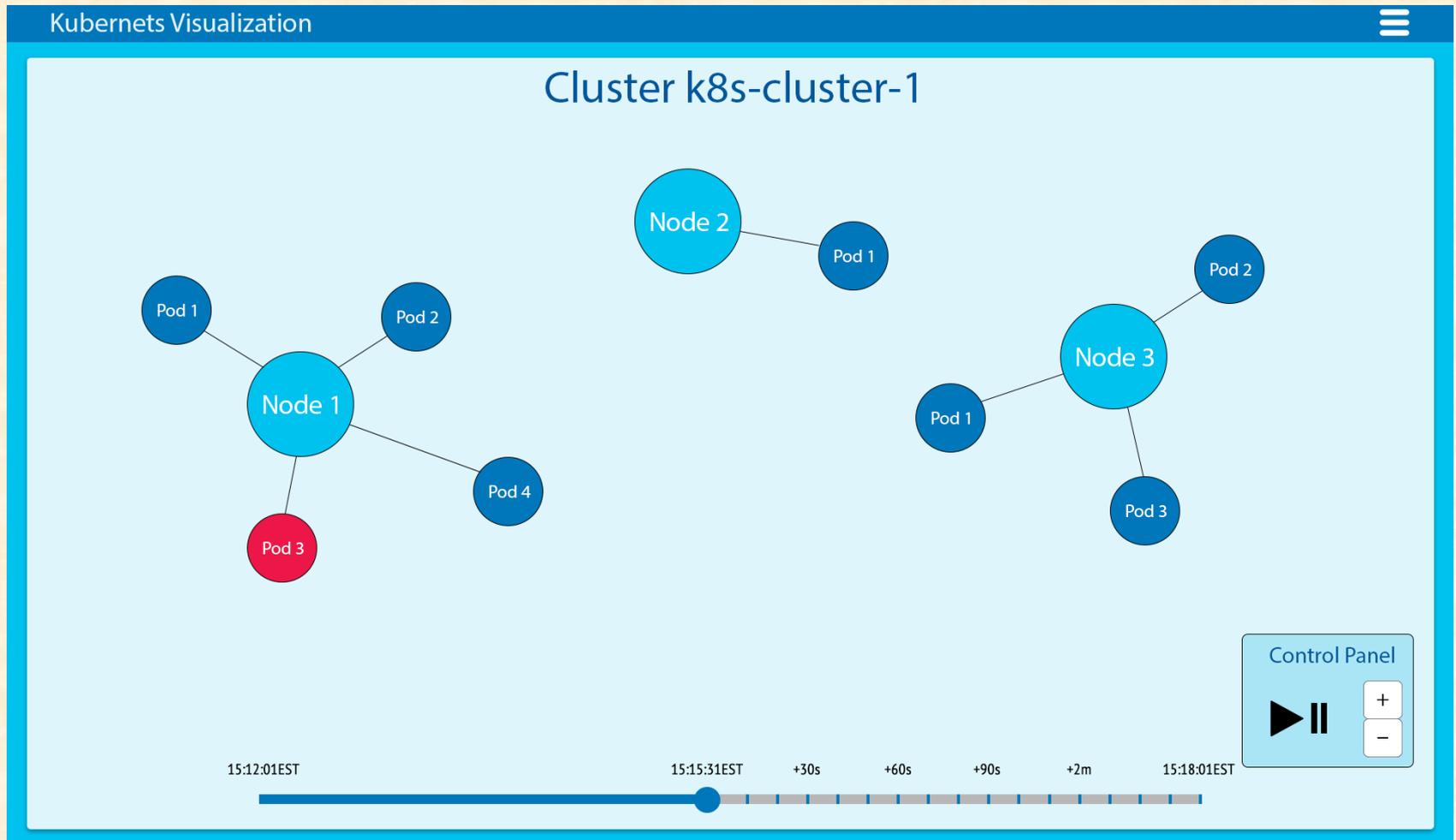


Design Specifications

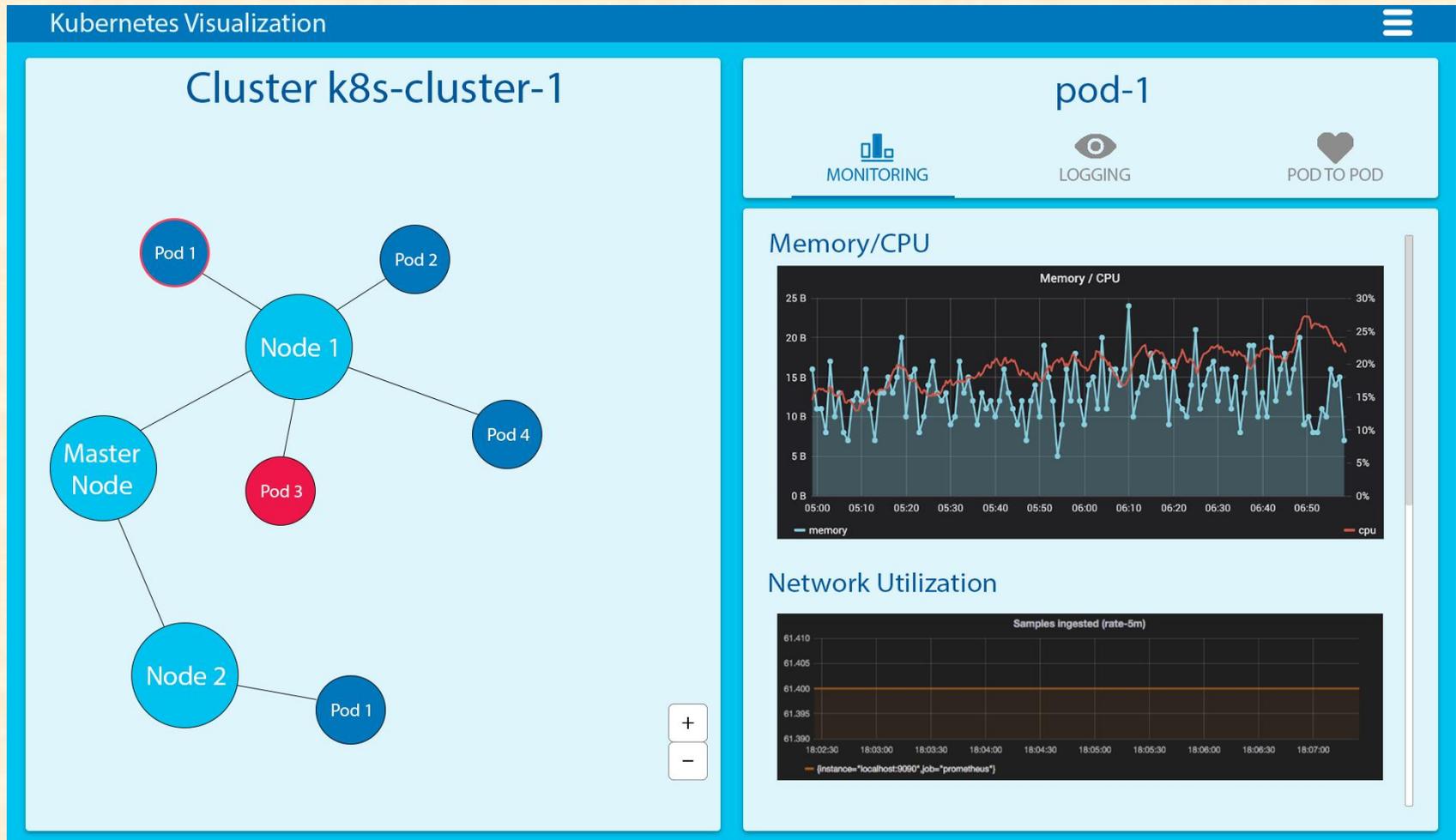
- Reactive Single Page Application
- Cluster Visualization
 - Monitoring and Logging
 - Grafana Dashboard
 - Pod-to-Pod and Service-to-Service Communication
- Historical View Playback
- Focus: Assist Cluster Troubleshooting
- Material Design Framework



Screen Mockup: Interactive Cluster Map



Screen Mockup: Drilldown Monitoring View



Screen Mockup: Drilldown Logging View

Kubernetes Visualization

Cluster k8s-cluster-1

```
graph TD; MasterNode((Master Node)) --- Node1((Node 1)); MasterNode --- Node2((Node 2)); Node1 --- Pod1_1((Pod 1)); Node1 --- Pod2((Pod 2)); Node1 --- Pod3((Pod 3)); Node1 --- Pod4((Pod 4)); Node2 --- Pod1_2((Pod 1));
```

pod-1

MONITORING LOGGING POD TO POD

Logs

- 2019-01-27 15:15:31 EST Killing container with id docker://metadata-agent:Container failed liveness
- 2019-01-27 15:15:31 EST Created container
- 2019-01-27 15:15:33 EST Started container
- 2019-01-27 15:16:20 EST Liveliness probe failed: HTTP probe failed with statuscode: 500
- 2019-01-27 16:20:00 EST Created container
- 2019-01-27 16:20:00 EST Started container
- 2019-01-27 16:20:00 EST Other k8s logs



Screen Mockup: Pod to Pod View

The screen mockup is titled "Kubernetes Visualization" and is divided into two main sections. The left section, titled "Cluster k8s-cluster-1", displays a network diagram. It features a central "Node 1" connected to a "Master Node" and "Node 2". "Node 1" is also connected to "Pod 1", "Pod 2", and "Pod 3". "Node 2" is connected to "Pod 1". A red line connects "Pod 1" and "Pod 2". The right section, titled "pod-1 to pod-2", contains three tabs: "MONITORING", "LOGGING", and "POD TO POD". The "POD TO POD" tab is active, showing a list of "Pod to Pod Logs". The logs include several entries of "Pod 2 received 2kb from Pod 1", one entry of "Pod 1 recieved error 500 while trying to reach Pod 1" (highlighted in pink), and one entry of "Pod 2 received 4kb from Pod 2".

Kubernetes Visualization

Cluster k8s-cluster-1

pod-1 to pod-2

MONITORING LOGGING **POD TO POD**

Pod to Pod Logs

- Pod 2 received 2kb from Pod 1
- Pod 1 sent 2kb to Pod 2
- Pod 2 received 4kb from Pod 2
- Pod 1 recieved error 500 while trying to reach Pod 1
- Pod 2 recieved 2kb from Pod 1

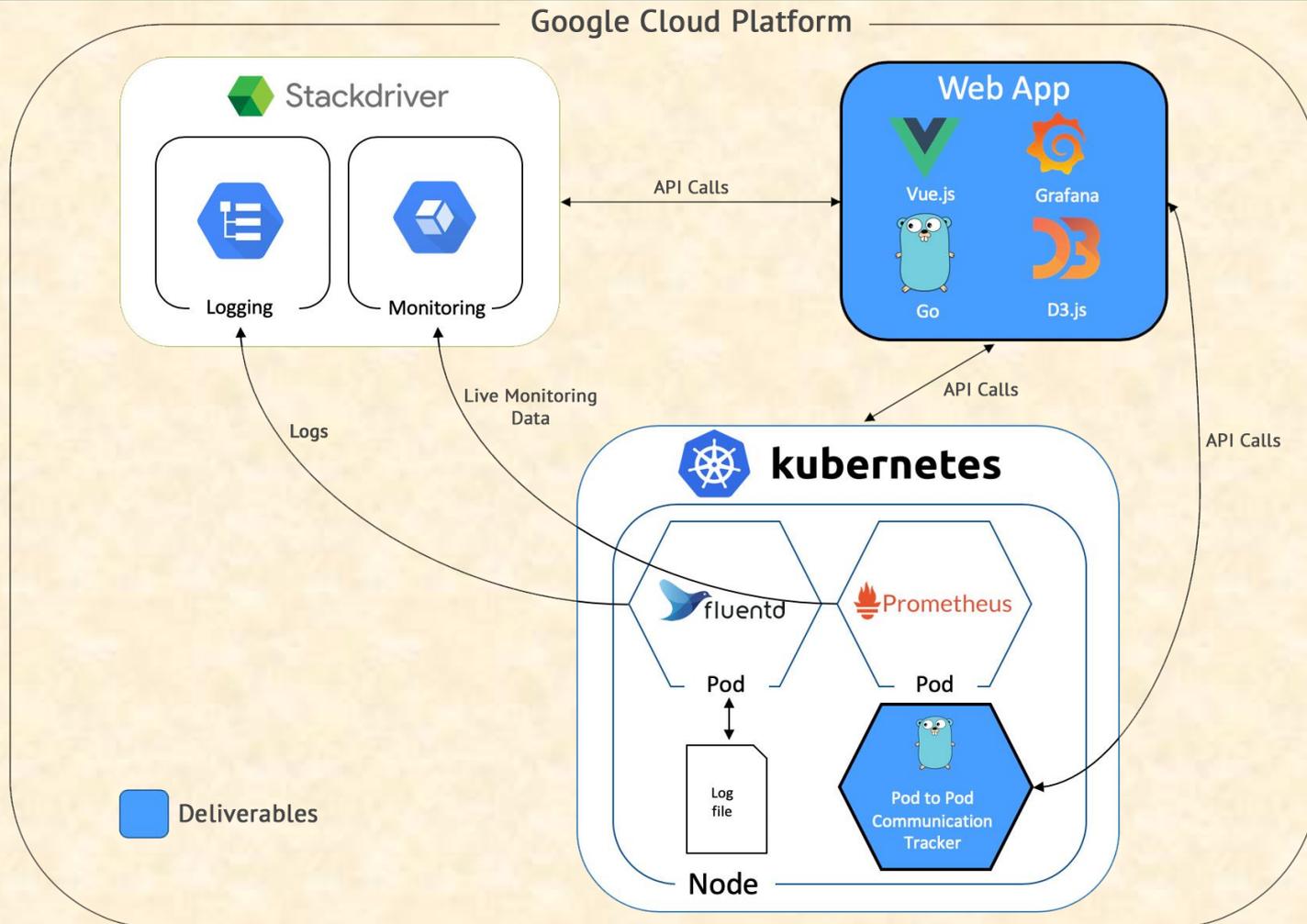


Technical Specifications

- Web Application
 - Metrics and Logging Visualization Dashboard
 - Interactive Cluster Map and Events Timeline
- Pod-to-Pod Communication Tracker
 - Daemon Set on each node in the cluster
 - RESTful API for the web app to access the data



System Architecture



System Components

- Frontend
 - Vue.js
 - Grafana
 - D3.js
- Backend
 - Go
- Software Platforms / Technologies
 - Kubernetes
 - Stackdriver APIs
 - Monitoring
 - Logging
 - Fluentd and Prometheus



Risks

- Feasibility of Pod-to-Pod Communication Monitoring
 - Description: There is no other product on the market that is able to achieve this feature – not sure if it is even possible
 - Mitigation: Adding a Daemon Set to each node in the cluster
- Cost Effective Data Caching Algorithm
 - Description: Making calls to the Stackdriver API racks up a charge
 - Mitigation: Data obtained from the APIs will be stored in a cache for easy retrieval
- Unfamiliarity with Needs of the End User
 - Description: No one on the team uses Kubernetes in our daily lives, can't easily anticipate what features would be most useful
 - Mitigation: Reach out to Kubernetes users for feedback and input
- Playback Button to See Historical Cluster Events
 - Description: Not sure if this feature is possible with available technology
 - Mitigation: Use other applications with similar features as a model, and do some testing of this feature on a small scale using javascript



Questions?

?

?

?

?

?

?

?

?

?

