**User stories:**

**Epic:** Add Prometheus alerts for cluster conditions

**User story:** As a DevOps Engineer I want to know when my clusters are running out of resources so I can prevent crashes before they happen and allocate the appropriate resources

**Story Points: 2**

**Epic:** Integrate Alertmanager into the monitoring deployment

**User story:** As a DevOps Engineer I want to be able to keep track of my alerts without opening Grafana so that I’m aware of alerts sooner and more consistently

**Story Points: 5**

**Epic:** Configure Planetarium Cluster so monitoring tools and planetarium pods are hosted on separate nodes

**User story:** As a DevOps Engineer I want my monitoring tools and applications to be deployed separately so that one crashing does not affect the other

**Story Points: 6**

**Epic:** Store and reference sensitive information needed for Planetarium and monitoring tools to work as secrets in the cluster

**User story:** As a user I want to know that my information is secure so that I don’t have to worry about or deal with a data breach

**Story Points: 5**

**Epic:** Store data saved by Prometheus and Loki outside of the cluster in a persistent volume

**User story:** As a DevOps Engineer I need to have my data saved in a persistent volume so that I can access the data even if the containers fail and the data is lost locally.

**Story Points: 3**

**Epic:** Configure Prometheus to alert using a multi-window multi-burn rate strategy for relevant SLIs

**User story:** As a DevOps Engineer I want to ensure that I have an alerting method that has both a good recall and precision so that I can respond to incidents that occur and not waste time on issues that are not significant incidents.

**Story Points: 1**

**Epic:** Add a DevOps pipeline to automate image deployment to cluster

**User story:** As a DevOps Engineer I want the DevOps pipeline to be as automated as reasonable to reduce toil

**Story Points: 4**

**Sprint 1:**

**Intended velocity: 9**

**Epics:**

Integrate Alertmanager into the monitoring deployment

Store data saved by Prometheus and Loki outside of the cluster in a persistent volume

Configure Prometheus to alert using a multi-window multi-burn rate strategy for relevant SLIs

**Daily Scrum 1/31:**

**Jiawen: Working on setting up persistent volume in cluster. He has finished latency recording rules.**

**Colton: No issues so far. Will be helping setting up the persistent volume outside of the cluster.**

**Trevor: Also working on setting up the persistent volume. No issues**

**Daily Scrum 2/1:**

**Colton: Plans to study for the CoE representative coming in. Afterwards will work on setting up S3 storage for Loki logs.**

**Jiawen: Plans to do similar work. Is working on the documentation for the S3 bucket. Investigating endpoints for the S3 bucket.**

**Trevor: Will be looking into alternative ways to store logs other than S3. Having issues applying a Loki configuration to send logs to S3 bucket in AWS.**

**Daily Scrum 2/2:**

**Colton: Will help with the S3 bucket and the Alertmanager. Yesterday worked on the S3 bucket**

**Jiawen: Will be creating the database to use for the S3 bucket. Got a configuration for the database yesterday and will work on implementing it**

**Trevor: Worked on implementing AlertManager yesterday. Trying to set up email notification but was unable to get that working. Will continue to work on setting up email notifications today.**

**Sprint 2:**

**Intended Velocity: 10**

**Epics:**

Change Kubernetes deployment to allow for a separate testing and production environment

Store and reference sensitive information needed for Planetarium and monitoring tools to work as secrets in the cluster

**Daily Scrum 2/6:  
Colton: Will start working on a canary deployment for the Kubernetes cluster.**

**Jiawen: Will start working on a canary deployment for the Kubernetes cluster.**

**Trevor: Will start working on a canary deployment for the Kubernetes cluster.**