**Organizational**

**Service**

The Shopping Cart Service allows Windows users to purchase items online. We provide the application for users to be able to register, login, logout, add items to their respective carts, empty their respective carts, and the ability to checkout. All of these functions are accessed through the API via a REST API. The data is stored on a Postgres database. The service will be available on a public HTTP server.

For administers, we provide the ability to apply roles to accounts (Admin, Employee, Customer), as well as view customer accounts and their associated carts.

We will use Jenkins to automate the build process, from code written in Java/IntelliJ, and Docker Containers.

**Team**

Jayden Rainsey, Chris Powell, Annette Reese

**Architectural**

The service will deploy the Docker container with Kubernetes (six nodes) along with the required files built with Jenkins, to both Github and Dockerhub.

**Environmental dependencies**

A Computer with CPU, HDD, RAM, a monitor, keyboard, mouse, network capabilities, and Windows 10 Enterprise.

**Service dependencies**

Kubernetes, Docker, Jenkins, IntelliJ, Maven, Git(Hub), Grafana, Prometheus, Promtail

**Client-facing**

**Clients**

Window users

**Request Classes**

What are the request classes, and how is a request classified? If your service has only one request class, delete this section.

**Service Level Indicators (SLIs)**

|  |  |  |
| --- | --- | --- |
| **Category: *API*** | **SLI** | **SLO** |
|  |  |  |
|  |  |  |
| Latency | The proportion of sufficiently fast requests, as measured from the micrometer API.  “Sufficiently fast” is defined as < 2s.  count of "api" http\_requests with a duration less than or equal to "1.0" seconds divided by count of all "api" http\_requests | 90% of requests < 1s  99% of requests < 2s |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Grafana*** |  | | |  |
|  |  | | | 99% |
| Availability | The errors reported by Prometheus should not last for more than 5 minutes. | | | 90% of requests < 200 ms  99% of requests < 1,000 ms |
|  | |  |  | |

**Rationale**

Availability and latency SLIs were based on measurement over the period 2018-01-01 to 2018-01-28. Availability SLOs were rounded down to the nearest 1% and latency SLO timings were rounded up to the nearest 50 ms. All other numbers were picked by the author and the services were verified to be running at or above those levels.

No attempt has yet been made to verify that these numbers correlate strongly with user experience.[1](https://sre.google/workbook/slo-document/)

**Error Budget**

Definition: (1-SLO goal) x (eligible events in compliance period)

(1 - 90%) = 10% = Error Budget

10% of events measured by SLI can fail before our SLO goal is missed.

Burn rate tells us how fast we are consuming the allocated Error Budget.

**Clarifications and Caveats**

* Request metrics are measured at the load balancer. This measurement may fail to accurately measure cases where user requests didn’t reach the load balancer.
* We only count HTTP 5XX status messages as error codes; everything else is counted as success.
* The test data used by the correctness prober contains approximately 200 tests, which are injected every 1s. Our error budget is 48 errors every four weeks.

**Operational**

**Monitoring**

This service is monitored by Grafana, with the use of Prometheus and Promtail services for metrics analytics and logging analytics respectively. Alerts will be sent to Slack.

**Troubleshooting**

Troubleshooting will primarily be done with IntelliJ and Postgres.

**Deployment**

The service will be deployed with Kubernetes using six nodes.

**Service Level Objectives**

**Realistic targets**

Realistic targets are 95% availability on all SLIs. This will leave some room for non-fatal errors.

**Ideal targets**

Ideal targets are 99.9999%. If everything worked correctly all the time, we can achieve this target. What is left are the variables that we cannot control such as natural disasters or accidents.

**Reconciliation**

Realistic targets and ideal targets will be evaluated on a as-needed basis.