CS Capstone Problem Statement

November 2, 2018

Software product life cycle transformation with Docker container technology

Prepared for

HP

Bryan Crampton _____ Signature _____ Date

Prepared by

Group 58

Brendan Byers _____ Signature _____ Date

Austin Sanders _____ Signature _____ Date

Daniel Grocki _____ Signature _____ Date

David Jansen _____ Signature _____ Date

Owen Loughran _____ Signature _____ Date

Abstract

HP PageWide printers are large, room-sized printers that are usually used for 3 different types of print jobs: corrugated packaging, high-volume commercial, and sign/display. These printers are extremely complex, and because of that require large amounts of computational power, both inside the printers themselves and in the tech stack running them. There are many different services that run in the backend that manage the print jobs and make sure each step of the process is properly managed and performed. The complexity of the system has reached a point where it is difficult to develop and manage services during their lifecycle. HP is looking for a better way to build, manage, test and deploy required services that also reduce the amount of computing resources needed to operate the printers.

CONTENTS

1 Introduction 2

2 Piece 1: Project Management Software 2

2.1 Jira . . . 2

2.2 Trello . . . 2

2.3 Agilean . . . 2

3 Piece 2: Inter-Service Communication 3

3.1 HTTP GET/POST . . . 3

3.2 WebSockets . . . 3

3.3 Long Polling . . . 3

4 Piece 3: Logging Collection and Viewing 4

4.1 Loggly . . . 4

4.2 Splunk . . . 4

5 Conclusion 4

1 Introduction

2 Piece 1: Project Management Software

In order to keep the project organized and moving smoothly, it is important to track what needs to done. In our project we are practicing the agile software development style. The software we choose should provide tools structured around the agile method. We should be able to organize and track things like scrum boards, issue summaries, performance reports, and a roadmap. These help us visualize and understand what needs to be worked on and help us follow the agile development methodologies. The software we choose needs to also provide issue tracking functionality. To make sure we spend our time on things that really matter, there should also be ways to categorize, organize, and assign issues to specific people. Additionally, the software should be accessible and easy to use. The software should fit into our workflow as seamlessly as possible. It should allow us to work on the project without needing to spend precious time on figuring out complicated software. We would also be more inclined to select software that the other group members have experience with. Project members can focus fully on the project itself instead of having to re-learn or troubleshoot unfamiliar software. Finally, because it is a small project with a total team size of 5 members, we want it to be cost efficient. It shouldnt cost more than a few dollars per person to use the service. This is three different options that are available.

2.1 Jira

Jira is a project and issue tracking service run by Atlassian. It is a part of other products such as Bitbucket and Confluence. Jira is designed to cater to the agile development style, providing tools to properly design and track sprints that are easy to understand. It is also reasonably customizable, allowing teams to modify the workflow to best fit with how they operate.

For our project, there is no cost to use Jira. It is already in use at HP, and so our project can easily be added without needing large amounts of setup and coordination between team members, project leads, and HP devops.

2.2 Trello

Trello is produced by Atlassian, same as Jira. It provides teams a way to collaborate on projects using boards. It allows users to create different types of tasks and attach them to a board. While it doesnt have as many tools as Jira, it provides a simple interface to manage small projects. With larger projects, its simplicity is a detriment because it becomes difficult to organize things and the interface can become chaotic.

For our purposes, I believe it would be free. There are paid pricing tiers, but they are primarily geared toward multi-team and enterprise scale projects.

2.3 Agilean

Agilean is designed for Scrum type development. It provides a few additional tools beyond that of Jira. There are a few more automation tools available that can automate some of the workflow around the project. This would cut down on the time we would spend configuring and moving things around. Our project should be relatively simple and straightforward, so we probably wouldnt take full advantage of the options though. It also has monitoring tools that provide insight into the development plan, allowing users to identify impediments they are facing. These are incorporated into the project flow and make it easier to identify and solve issues that team members are facing. It

also provides functionality to coordinate and streamline scrum standup meetings. With it you can automatically plan, track, and summarize standups. It provides automated meeting minutes for later reference in addition to recording and following up on action points discovered during the meetings. This would help a team get more out of standups and be more productive.

For teams less than 50 people, it is completely free. Our team doesnt plan on adding people beyond who we already have, so we would be able to use it for free.

3 Piece 2: Inter-Service Communication

To process a print job, the file to be printed goes through various transformation done by different services. Everything is coordinated by a Work Manager service. This service receives jobs and coordinates job preparation by Worker A The services often run on different servers and require some sort of inter-server communication. While Worker A is working, it may require supplementary information or data to complete the job. There needs to be a way to communicate the request and the information between the Work Manager and Worker. The entire printing process is extremely time sensitive, so it is important to get the workers what they need to complete the job quickly. When the work is complete, Worker A signals that to the work manager stores its completed work in a place accessible for the next step of the process. To accomplish this, a combination of HTTP GET/POST and websockets is used. They are each outlined below in addition to Long polling, which is a legacy protocol similar to websockets.

3.1 HTTP GET/POST

HTTP is a request-response protocol that is unidirectional. In order to receive data, a client must make a GET request and the server will POST its responds to the client. With this protocol, the server is unable to initiate a connection with a client.

Currently it is used to send commands to services, while websockets are used to communicate status and more time sensitive communications. HTTP was designed to be stateless and to communicate through a request/response model. It is best suited for retrieving things from a server that dont require ongoing updates. If constant contact is required or asynchronous communication needs to take place, websockets are better suited for the task.

3.2 WebSockets

The primary purpose of websockets is to send and receive real-time data from another service. When initialized, a two way pipe is create between the client and server which they can communicate through. The initial handshake between client and server uses HTTP, but once that is complete they communicate using the WebSocket protocol. Instead of streaming data, websockets is built on a message based transport system that is bi-directional. Entities are able to quickly communicate with each other little very low latency. After the initial handshake, there is little overhead beyond the messages themselves. Websockets provide a stateful connection between two entities, meaning that information about the connection is saved and can be used later.. It also support duplex communication, meaning both client and server can send and receive simultaneously.

3.3 Long Polling

Long polling is a modification of the HTTP GET/POST protocol. In this mode, the server holds a request open until new data is available and then sends that to the client. The client makes an initial GET request to initiate the connection

and the server will respond opening a POST response. Instead of closing the POST, it remains open to allow the server to push data to the client.

Long polling was created before websockets existed. It created a rudimentary way for servers to continually communicate and push data to a client in a semi-synchronous manner. For all intensive purposes, websockets should be used instead.

4 Piece 3: Logging Collection and Viewing

In addition to running services, we also want to manage logs so they can be used to monitor and record performance. Large amounts of logs will be produced and we also need a way to comprehend the logs. An open source service is preferred because of the open source nature of the rest of our software. At a minimum we want to record and interpret logs from development and testing servers, but it would also be nice to have in depth log monitoring functionality in production. We will be interpreting logs from a variety of sources. At a minimum we want to monitor the resources used by the servers. This includes CPU activity and usage, memory usage, and network activity. We would also like to monitor the services themselves, recording information like performance, crashes, and individual resource usage. The docker containers provide a standardized way to record this information, so it is important that the logging tool that is chosen is able to interact with them.

4.1 Loggly

Loggly is a cloud based log management service. Once it is hooked up, it allows a user to monitor logs from a variety of sources and congregate them all in one place. It also provides functionality to create graphs and charts based off of logs. One reason for choosing this is its preconfigured dashboards. It comes setup and ready for 9 different log sources, including Linux and Docker. This would integrate cleanly into what we plan to create. Because it is cloud based, we would need to be able to hook up the services we want to monitor to the external servers hosting our Loggly instance.

One drawback of this service is its cost. It costs around $45 a month and is limited to 1 gigabyte of logs per day. If we chose to get an enterprise license, it would only be $1.50/GB per day. Not knowing how much logs we will generate makes it difficult to calculate the total cost.

Loggly also isnt able to be self-hosted. The logs themselves can be hosted on premises, but most of the heavy calculations go through Logglys servers. It is not open source, while most of the other technologies we are using are. If it was added to the servers that go out to HPs customers, it would be a headache to support because it is meant for use by a single business, not as a service provided to clients.

4.2 Splunk

5 Conclusion