Second Deliverables
Team 8: Kevin Jones, Chris Knieste, Evan Shone, Joshua Streger
Dr. Bohacek

GitHub link:
https://github.com/jdstregz/sky-scraper
Schema received is schema.txt

Deliverable 1
Our first steps were to gather requirements, find possible web services, choose appropriate tools, and build prototypes for how we might solve this problem. Each prototype was designed for a different web service by a different member of the team. This was done to allow everyone to learn about the problem and the tools, and come up with unique solutions that we can look over. Our next step will be to analyze our different solutions to come up with the most simple and efficient method. After that we will start working with an actual database to store the scraping data, and then create an environment with a simple command line utility.

Deliverable 2
Each prototype was pretty similar. Start with a list of URL's and pull all the info then look for the specific region of data using various techniques such as css, xpath, and regular expressions. The amazon prototype had some workaround because of the website's use of javascript. We like that each of the prototypes work, but dislike that the methods are not easily extensible nor permanent. As soon as the website changes the spiders will break.

We got the schema from Dr. Bohacek on Monday (11/7) and worked to hook up the scraped data from the prototypes with a PostgreSQL database. We also worked on implementing Docker to allow us to have a uniform way to access the databases. Docker is a software that allows containerization of our PostgreSQL and splash images so we can have consistent testing on all of our systems. It also allows for easy setup. Preliminary research was conducted for a command line utility and setup script. Our final steps will be to finish hooking up the prototypes and databases and get the command line utility and setup scripts functional.

Requirements
- Scrape pricing for various cloud services - Completed
    - Virtual machine and instance store services in particular - Completed
- Command line utility
- To help match customers with best cloud compute services
- Should be able to track history of pricing

Scraping tools
- Scrapy (Python)

Major web services to scrape data
- **Google**

- **Amazon**
- **Microsoft Azure**
- Alibaba Cloud
- CenturyLink
- IBM Softlayer

Create prototypes for the major cloud services using Scrapy

- Google - **Josh**
- Amazon - **Kevin**
- Microsoft Azure - **Chris**
- Alibaba - **Evan**

Testing

See screen-shots in tests directory on GitHub