# Zhiyuan Ouyang

Email: zhiyuan.ouyang.jobs@gmail.com | Tel: +1(412)951-3268
Website: zhiyuanouyang.github.io | zhiyuanouyang.gitlab.io

## Education

**Carnegie Mellon University** Pittsburgh, PA
*Master of Science in Computer Engineering* *Dec 2016*

**Relevant Coursework** Database Application(15615), Distributed System(15640), Cloud Computing(15619), Machine Learning(10701), Computer Vision(16720), Introduction to Computer System(15513), Principles of Software System Construction(15214), Service Oriented Computing(18655), Java Smart Phone Development(18641)

**Katholieke Universiteit Leuven** Leuven, Belgium
*Bachelor of Science, Magna Cum Laude* *July 2015*

**Relevant Coursework** System Software, Software Development, Data Communication and Computer Networks, Object Oriented Programming and Databases

**Beijing Jiaotong University** Beijing, China
*Bachelor of Engineering, Siyuan Honor Program* *July 2015*

**Relevant Coursework** High Level Language Programming, Fundamentals of Computer, JAVA Programming

## Skills

- **Languages**: Java, Python, Ruby, Shell, SQL, etc.
- **Databases**: MySQL, PostgreSQL, Hbase, Cassandra, MongoDB, Redis, Memcached, Neo4j, etc.
- **Frameworks**: Kafka, Fluentd, Spark, MapReduce, JUnit, Cucumber, Dropwizard, etc.
- **Tools**: Jenkins, Git, Kubernetes, Docker, Grafana, Prometheus, Hive, Impala, Hue, Oozie, ElasticSearch, Superset, MetaBase, etc.

## Professional Experience

**Oracle Corporation** Redwood Shores, CA
*Software Engineer, Oracle Cloud Infrastructure Team* *March 2017 - Present*

**Implemented and built up a multi-layer large scale data pipeline/data platform for log/metric/business data across Oracle Cloud Infrastructure, including Collection/Transferring/Transforming/Storaging/Analytics/Visualization layers and helped set up and completed the operation pipeline including Continuous Integration/Continuous Testing/Continuous Deployment/Continuous Monitoring process**

- **Data Collection :** Designed and Implemented multiple data collection modules for collecting data/metrics/logs from various user endpoints/VMs, doing filtering/pre-processing and emitting to various destinations, using Java/Ruby/Fluentd
- **Data Transfering :** Helped set up the Kafka clusters and Implemented modules in Java for transfering and distributing data/metrics/logs to various data sinks
- **Data Transformation :** Impelmented transformation/processing utilities for extracting the value of data based on different use cases in Java/Python to be processed by MR/Spark clusters and wrote shell scripts in Oozie for scheduling
- **Data Storage :** Implemented data accessing modules in Java for Cassandra/Hbase backend
- **Data Analytics :** Wrote SQL for some analytics tasks to extract the value of data for clients using Hive/Impala
- **Data Visualization :** Implemented Web server and provided REST API for data visulization from Cassandra/Hbase backend
- **Continuous Integration :** Implemented Jenkinsfile for multiple modules in datapipeline to enable the continous-integration process
- **Continuous Testing :** Implemented testing scripts for modules in datapipeline using multiple test frameworks JUnit/Ruby Unit Test/Cucumber
- **Continuous Deployment :** Helped created Dockerfiles/Helm Charts for continous-deployment with Kubernetes clusters
- **Continuous Monitoing :** Integrated Prometheus exporter in modules in data pipeline and helped set up the Grafana dashboard for monitoring

## Course Projects

**Carnegie Mellon University** Pittsburgh, PA
*Elastic Cloud Platform* *March 2016 - April 2016*

**Implemented a elastic cloud platform for online shopping system with auto-scaling mechanism in Java**

- Designed the scaling logic and implemented the elastic cloud with a flow-meter to monitor the total load of requests, a load-balancer to balance the load of each instance and a load-meter to monitor the load of each instance

**Carnegie Mellon University** Pittsburgh, PA
*File Caching System* *Feb 2016 - March 2016*

**Implemented a concurrent proxy to support high throughput and low latency file read/write requests**

- Implemented a proxy to handle file read/write requests from multi-tenants concurrently, caching hot-spots in limited resources for low-latency and high throughput. Implemented the caching mechanism with LRU eviction policy, 'write around' invalidation policy, 'check on use' fetching policy and 'last writes win' consistency policy