# Professional Experience

After the research experience at the State Key Laboratory of Networking and Switching Technology (SKLNST) and the High-Performance Computing Institute (HPCI), I found that enterprise customers are why research institutions give their greatest effort, and as such, I am greatly interested in industrial software development. Moreover, rigorous testing and fully standardized software engineering processes are also inaccessible in labs compared to an industry company. Intending to build a broad view of software engineering and train myself to be an all-around engineer, I passed a rigorous selection process and began an internship at IBM in 2016.

I am a member of the DevOps team of Cloud Managed Service (CMS) at the IBM China Software Development Laboratory (CSDL), which undertakes cumbersome tasks that were manually completed every time a new application-deploying requirement was initialized inside or outside of IBM worldwide. Considering the booming market share of cloud services of IBM, changes must take place: a new architecture supporting Continuous Integration and Continuous Deployment (CI/CD) is needed. The ideal workflow can be described as follows: a software development engineer/team can establish a custom test environment, according to the need on our constructed web application, and can easily deploy their target software in the environment within a few steps. Moreover, deployment should also be done automatically every time the source code is modified.

Since we planned to build the whole set on distributed nodes, the status should be monitored. The first project was a web application, illustrating the topologies of the managed computing cluster. I independently completed this project, which lasted for one month. It consisted of two parts: the front-end, and the server-end. The former is responsible for presenting canvas to users and sending HTTP requests to the back-end to retrieve data. And the latter one was in charge of the status information of each node, which was generated by periodically triggered shell script. During the development, the biggest challenge was how to make sure the graphics reflected the latest status of the nodes. Periodically refreshing the pages would result in a terrible user experience, while partly refreshing the pages was a feasible solution, but still not efficient enough. At last, I built the front-end page in AngularJS and bidirectional bound page view with cached data, observing the model-view-view model (MVVM) concept and effectively obliterating the time delay associated with data refreshing.

The second project was the implementation of the microservice project, including front-end web application, back-end middleware, web server, distributed nodes and containers on nodes, etc. The development has lasted for four months, and thus far, it contains about 30K lines of code for the front-end and 3K lines of code for the back-end. Two senior software engineer and I are responsible for the development. I was mainly responsible for writing the front-end pages, the web-server, and the middleware.

The middleware is called API-Gateway, which is used to standardize the communication between the user and services, a proxy also responsible for services discovery, health check and load balance of user queries. It was implemented in Python at the demonstration stage. To avoid the cross-domain problem and decoupling the front-end application with middleware, I developed the web server using Express framework provided in NodeJS. The most important part is how to drive the deployment from the beginning to the end. Our solution was to use a third-party tool named Rancher to manage containers on distributed nodes and use Jenkins as the driver to initiate building, testing and deploying tasks. Now, the overall workflow can be briefly described as follows: users choose environment components on the web pages, which calls the API of containers management tool to get the environment ready. Then, the users locate their source and decide project-building strategy, which will initial a job on Jenkins. As a result, the deployment can be quickly finished as soon as the user makes a request. Currently, I am trying to host the entire set on a local virtual computing network, which will decrease the needs for constructing new testing centers of IBM, significantly cutting down the cost for the company. As the only undergraduate intern within the China Development Laboratory (CDL) in years, I felt incredibly lucky to undertake these challenging tasks, so that I could rapidly build my knowledge and refresh as well as deepen my understanding in the field of computer science and engineering.