1 Contact

- Cell Phone：13701143920
- Email：caoronglu@gmail.com

2 Basic Info

- Ronglu Cao/Male/1981
- Master/Computer Science at Xidian University
- Have been working since 2006, 12 years
- Residence: Chaoyang District, Beijing
- English Speaking: fluent

3 Experience

3.1 Oracle (China) Software Systems Co., Ltd. (2017/11 ~ Now)

Position: Cloud Operations
Working Environments: Linux/Bash/Shell, Python, Chef, Docker, Jira, Jenkins, Job Controller, Xen, Git

3.1.1 Projects

3.1.1.1 Oracle China DC Build

As the first member of China cloud team, I went through the whole process of setting up Oracle China cloud compute site, including both compute infrastructure and compute services build. On joining Oracle, I was assigned to work on building compute site for China cloud, which was not as easy as expected due to China government policy on public cloud infrastructure. With little support from team members from other sites (due to timezone issue), I managed to grow from code study and set up compute site components:

- Infrastructure setup including management node setup, chef server setup, monitoring host setup
- Public cloud services like compute service manager, regional compute service manager, network service in compute
- Microsoft key management service
- Service to service integration including elk, graphite, including supporting services like nimbula diagnostic report service

With proven problem-solving skills, I was assigned as China compute contact point during China cloud build and got a good relationship in the global team. As more members join China cloud team, I take the responsibility of run local daily sync up meeting and team member training.

3.2 Beijing Jingdong Information Technology Co., Ltd. (2017/06 ~ 2017/11)

Position: DevOps Engineer
Working Environment: Linux/Bash/Shell, Python/Perl/Shell, Git, Jenkins, KVM, Docker/Kubernetes, Django/Bootstrap, Mysql

3.2.1 Projects

3.2.1.1 Devops CI/CD Pipeline Build

Without a standard CI/CD pipeline, development work was done at a slow pace. I was hired to build standard CI/CD pipeline as well as train team member on its usage. To gain most from less effort in mind, we used gitlab + kubernetes + Jenkins to as starter:

- Gitlab as code repository with master/development branch management policy in team
- Jenkins with on-demand slaves run on kubernetes with each job using Jenkins file to specify build pipeline
- Application to be deployed in kubernetes after automated testing

This CI/CD pipeline has the following properties:

- Everything in code, controlled in git
- Developer has full control on dev/build/test/prod environment, no one else need to be involved
- Every build can be reproduced at any given time
- Little effort to set up and easy to adopt in team

It was considered as a clean DevOps pipeline after it had been adopted by the team

3.2.1.2 Restful API for Environment Change Request

With CI/CD pipeline setup, this is the first project after adopting the new DevOps process. The project is to setup restful API service by which one can conduct a change in pre-prod/production environment through restful API. In the project, the following techniques were used:

- Django as the web framework, and bootstrap as frontend library
- Celery for task execution management
- Chef to run change update on environments
- Pytest to run unit test/integration test

There were 4 engineers involved in this project, and I was working as lead as well as focusing on restful service/frontend/pytest development. The challenges of the project are:

- Time constraint is 1 month
- Chef recipe consolidation for a large number of modules
- Development process is new to the team

Within 4 weeks, we managed to deploy the service in the production environment on time. And this project was considered a good starter for future projects.

3.3 Cisco Systems (2011/01 ~ 2017/04)

Position: Team Lead & Scrum Master
Working Environment: Linux/Unix, Python/Perl/Shell, Git/SVN/Perforce, Jenkins/Maven/Java, VMware vSphere ESXi, OpenStack, Confluence, Jira, Fisheye, Docker/Kubernetes/Openshift, Cobertura, Rally

3.3.1 Projects

3.3.1.1 Quicksetup Deployment Tool

Background: VBO(Video Back Office - a video product) product deployment is becoming complex along the way, due to factors:

- More hardware resource environment available - 3 vCenter sites for teams in a different location
- Number of nodes varies for different deployment scenarios - simple deployment for smoke test vs performance test

- Different customizations for different customers - node deployment customization for different customers
- New version of the product introduces deployment changes

Overall, the deployment management is becoming complex, even out of control. Specifically, it is getting hard to root cause a deployment issue and difficult to add new features to the deployment tool. To resolve this issue, I designed a next-generation quick setup deployment tool base on DevOps principles, which will:

- Read configuration files including:
  - Hardware resource configuration, which contains vCenter parameters
  - Node deployment definitions file, which defined node types
  - Task definition file, which defined tasks to execute during deployment
- Compose a task dependency graph base on the graph definition from config and execute them in Luigi framework
- Each task will have its dedicated log file and can be re-executed on failure
- The task execution status will be shown on frontend webpage: yellow for pending, green for success and red for failure.

This solution has the following properties:

- All configuration files are version-controlled
- Common config files are shared among team members and treated as code
- Easy to pinpoint the root cause of a deployment failure
- Re-execute whole deployment after a root cause is fixed to restore from deployment failure

The new quick setup tool reduces the product deployment turn around time by executing independent tasks in parallel, and it enhances the whole team's performance by clear feedback on each deploy task.

3.3.2 Other Achievement and Responsibilities

- ELK orchestration: migrate ELK subsystem from virtual machine to openshift environment, components include Elasticsearch, Logstash, Kibana and Kafka
- VMware image generation tool: create loopback disk, mount/chroot and install product rpm files, after which it will be packed as ova format image template
- Run daily standup meeting as scrum master in our agile development process
- Product platform related responsibilities, such as OS upgrade, database upgrade, HA setup of nodes etc

3.4 Synopsys (Shanghai) Ltd. Co (2006/02 ~ 2011/01)

Position: SCM Engineer - Build Environment Support
Work Environment: C/C++, Make, Perl, Expect, Linux/Unix, Bash/Shell, Java

3.4.1 Projects

3.4.1.1 Purecov Report Generation Tool

To make sure the regression test covers all code lines, we run purecov(an IBM code instrument tool) to generate the coverage report. After running a regression on purecov instrumented software, we need to merge all purecov reports from each test to generate an overall report. It takes days to get the purecov report generated in the beginning. However, it takes much longer time after a major change in the product - introducing multi-process, which caused a single test to generate a set of purecov data files, one for each process. To solve the longtime problem, I consolidated script to merge purecov script once a test finished its running. After this change, a purecov report can be generated within a week.

3.4.1.2 Internal Tools Developed

- Perforce Code Change History Webpage - show code changes on a webpage in color
- Disk Usage Status Report Page - Collecting disk usage data, Pruning Obsolete files, Image management etc
- Product Auto-Installer - Automatic installation of products using expect
- Hardware Manager - Collect and show hardware data on the web portal

3.5 Xi'an 262 Nuclear Medical Instrument Company (2004/07 - 2005/05)

Position: Software Engineer

3.5.1 Project

3.5.1.1 Medical Instrument Software

The company was manufacturing Nuclear Medical Instrument. The project was to implement the software used in the medical instrument. The software reads data from serial port. Typically the data have 5 standard sample data besides real patient data. Given the sample value of standard sample data, it uses curve fitting to figure out the curve and use the curve parameter to calculate the corresponding value for the patient. I was responsible for modules:

- Curve fitting calculation to predict a patient result
- The communication protocol between instrument and computer, and interface to the upper layer
- Patient information and result management module include patient report generation and report format customization

4 Machine Learning Knowledge

Recently, I have been spending my time building expertise in machine learning. Machine Learning algorithms already learned:

- Supervised Learning
  - Linear Regression
  - Logistic Regression
  - Deep Learning Neural Networks
  - Support Vector Machine
- Unsupervised Learning
  - K-Means Clustering
  - Dimensionality Reduction
  - Anomaly Detection
  - PCA - Principal Components Analysis
  - SOM - Self Organizing Maps

5 Technical Skill List

- Programming Languages: Python/Perl/Bash/Java/Golang
- Web Development: django/bootstrap/rest framework
- Frameworks: Twisted/Luigi/Celery
- Logging: ELK/Splunk
- CI/CD Tools: Jenkins Pipeline/Maven/Nexus/Artifactory/Jira/Pytest/Checkstyle/Cobertura
- Database: MySQL/SQLite
- Source Code Controll: Git/SVN/Perforce

- Cloud Platforms：AWS/OPC/K8S/VMware vCenter/Hadoop/Spark

6 Education

- XiDian University (2003/08 ~ 2006/04) - Computer Science | Computer Software and Technology | Master | Xi'an
- XiDian University (1999/08 ~ 2003/07) - Computer Science | Computer Science and Technology | Bachelor | Xi'an

7 Acknowledge

Thanks for your time reading my resume and I am looking forward to working with you and building a brighter future together