



Portfolio

Seunghwan Kim (Allen)
seunghwan.kim1980@gmail.com

Seunghwan Kim (Allen)

Software Quality Engineer, DevOps



Education

Bachelor's degree in Computer Science

School of Computer Science / Kyonggi University
03/1999 – 02/2006 Suwon, Korea



Certificates

CSQE (Certified Software Quality Engineer – ASQ)

ISTQB Foundation (International Software Testing Qualification Board)

CMMI examinant qualification



Personal Development

Certified Kubernetes Administrator(CKA) (12/2019)

Introduction to CMMI Dev V1.3 (06/2016)

Development course using LabView (06/2014)



Language

English - Professional Working Proficiency

Korean - Native Proficiency

Japanese - Limited Working Proficiency



Contact

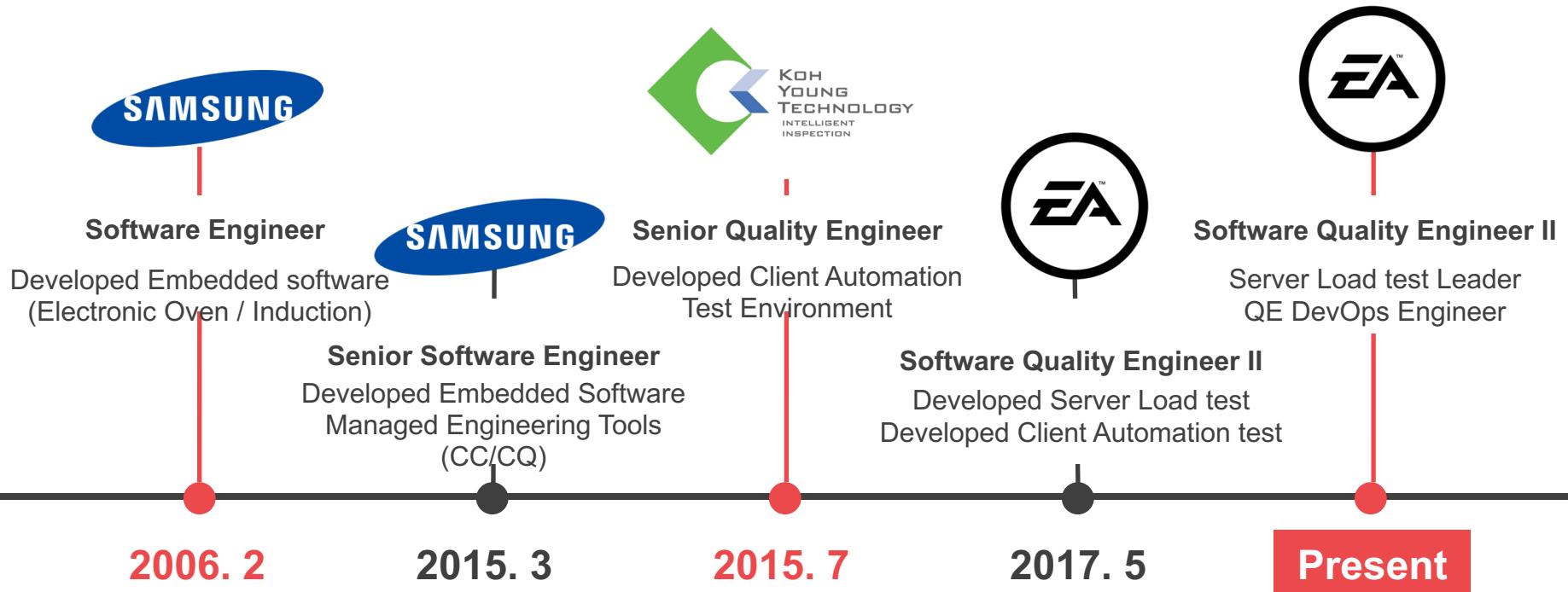
@ seunghwam.kim1980@gmail.com

778-764-5215

linkedin.com/in/allen-kim- 132887124

Work Experience

Focusing on the workplace and roles



Work Achievement



Software Engineer
Development & Management



Software Quality Engineer
30+ Server performance Test
Server Quality improvement
Automation testing



DevOps Engineer
Cluster Environment
Monitoring & Alerting System



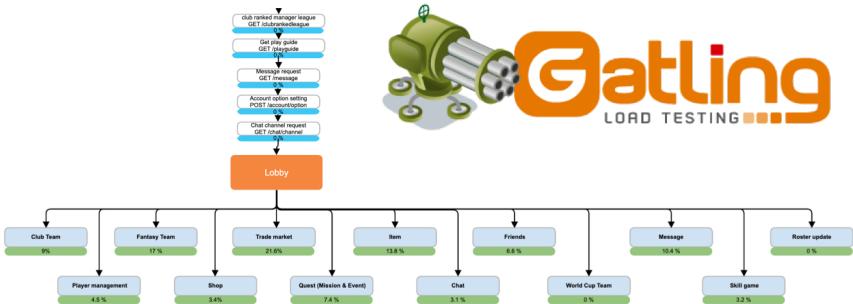
- Developed all electronic oven and induction S/W
Managed sub development team for Oven Products
- Refactored the code as an architectural design
Reduced development period & Increased Quality
- Establish whole server load testing for Game Server
Various types of test – Meta, Relay, and VoIP Server
- Successful game service open without Server issues
Work with various publisher – Tencent, Nexon, Garena
Maximum 300K User concurrent simulation scripts
- Increase service management by Cluster Environment
Increase PC and VM resource management
- Adapt various tool to increase QE development and
Management – CI / CD, Monitoring, Alerting



Project Portfolio

Game Server Load Test

Specific Works



Developed the load test scripts
Gatling based on Scala

Gathered the requirement from dev team
Gathered the actual packet between client and server
Implemented the load generating scripts
Created several extensions to help the test

Established the test environment
Continuous Integration

Created shell script to run the multi docker containers
Created test jobs to run the test automatically
Set up the Linux Virtual Machine using Ansible
Set up the AWS load testing environment



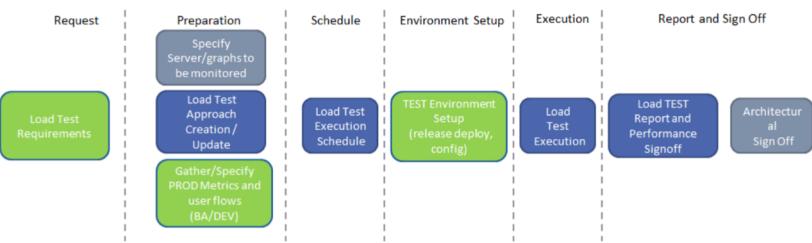
Game Server Load Test

Specific Works

Involved Teams



Phases



Monitoring & Reporting System

Time-based, Structural

Concurrent monitoring system - InfluxDB and Grafana

Created structural reporter for stakeholders

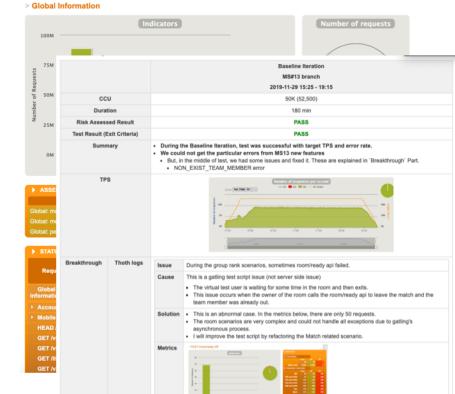
Shared the Official report for Dev team and Publisher

Set up the load test process

R&R, Meeting, Process

Fixed R&R of each team

Established entire processes for the load test
Kick-off, Development, Execution, Retrospective

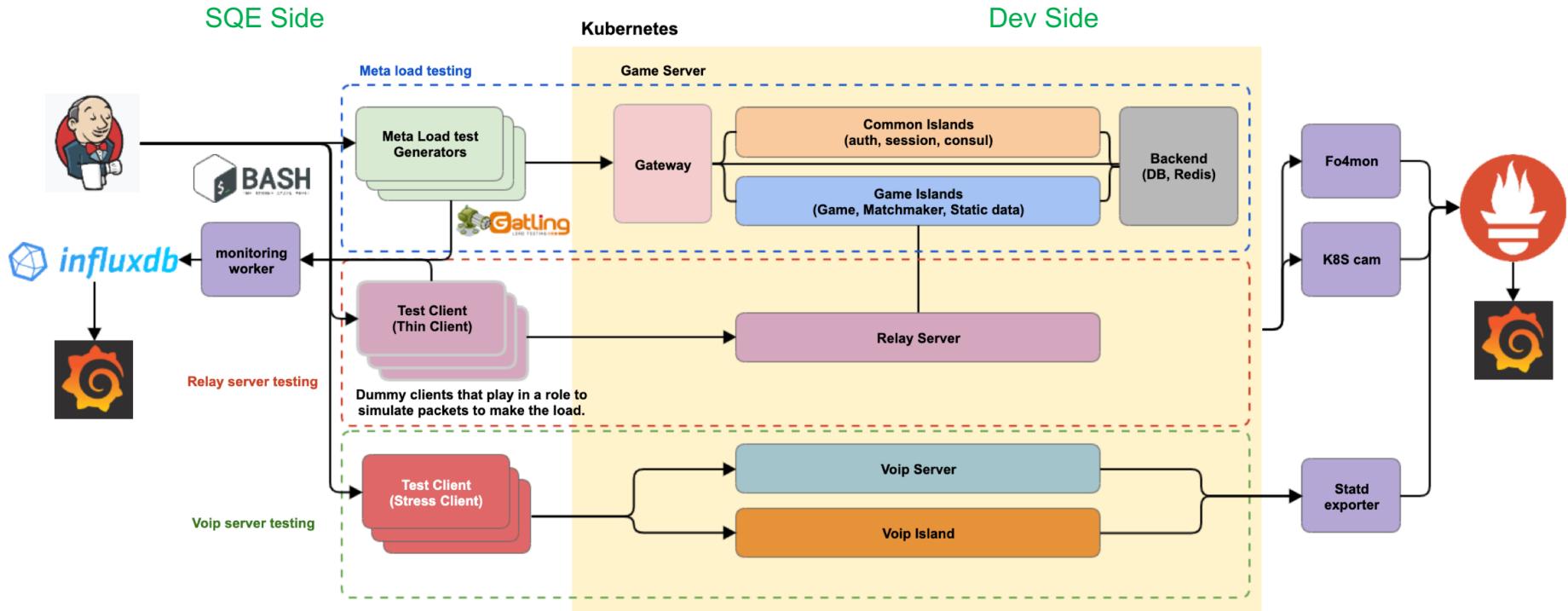


Game Server Load Test

Architecture

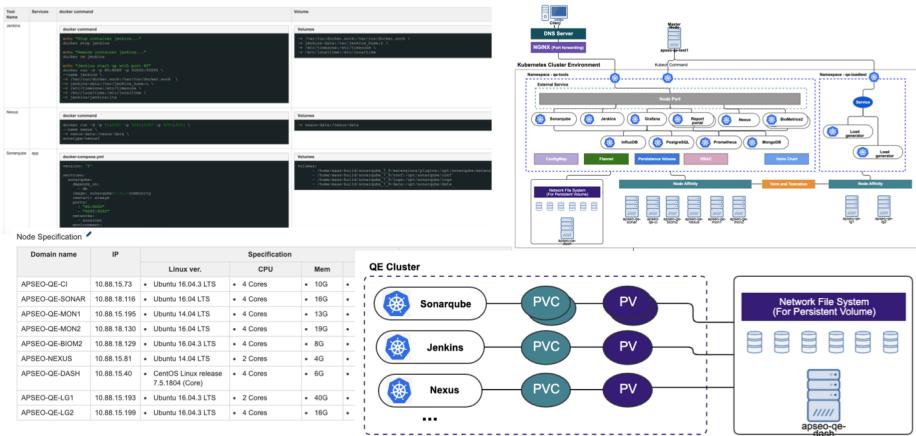


Established and Developed the Load testing Environment for Game



QE Infrastructure

Specific Works



Established Cluster tools

Version Control, Package manager, Monitoring

Adapted Helm chart to install the Kubernetes services
Managing the definition version by Helm charts
Set up the Cluster monitoring system and Alert system

Migrated Infra into Cluster Kubernetes (Linux VMs)

Investigated exist services in the environment
Designed cluster architecture based on the Kubernetes
Established Persistent volume strategy



kubernetes



QE Infrastructure

Specific Works



Improved Code Quality

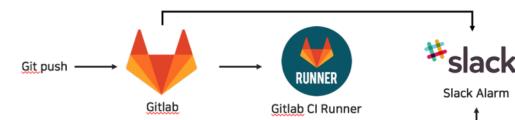
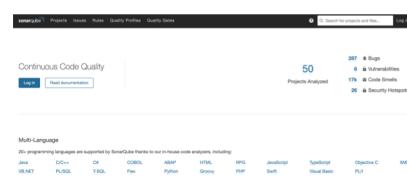
SonarQube, Gitlab CI Pipeline

Set up the Sonarqube project to check the code quality
Set up automated CI pipeline to run the every merge
Set up the result notification by an e-mail and Slack

Migrated Infra for Windows VMs

Monitoring, Alert system

Set up the monitoring system using Telegraf, Influxdb, Grafana
Set up the notification from Grafana to Slack
Improved the way to trouble shooting

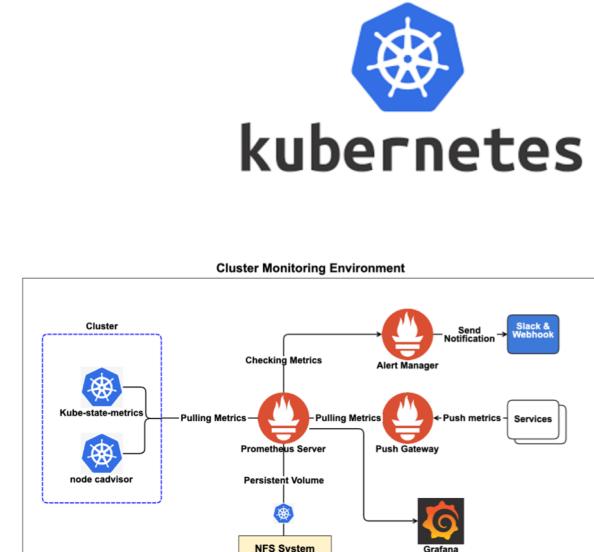
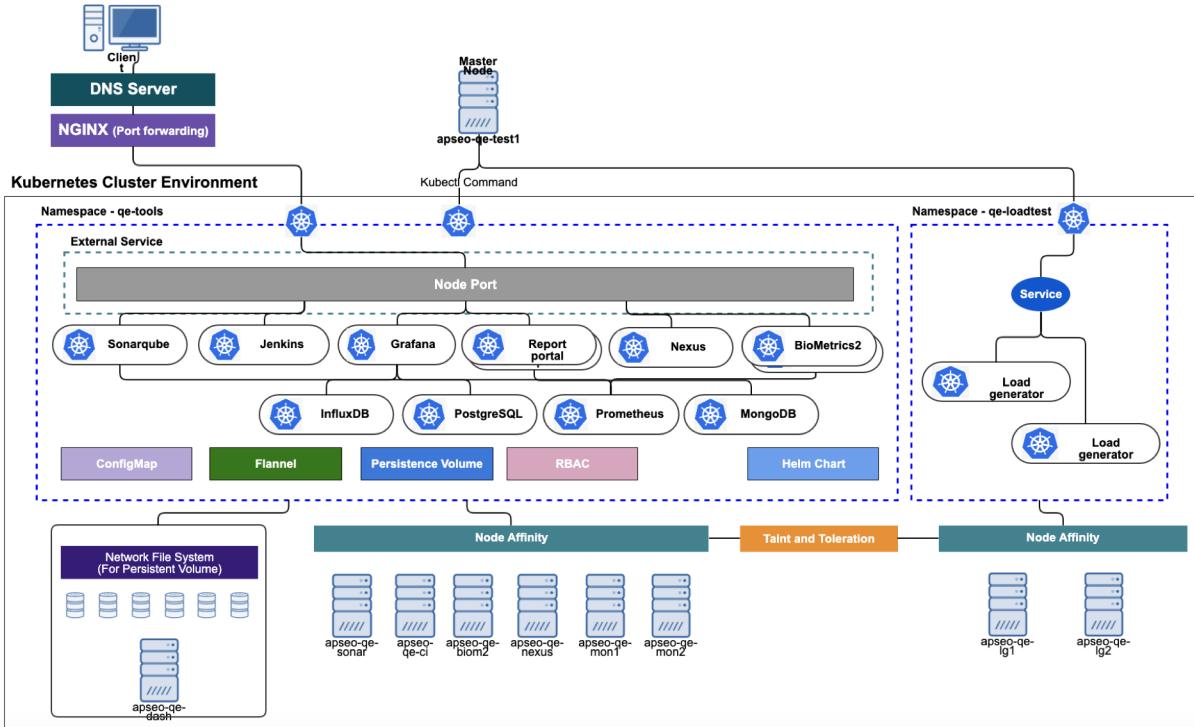


QE Infrastructure

Architecture



Established effective resource management environment



Automation Test

Client and Server Automation Test

Selector

S	W	Name	최근 성과	최근 실패	최근 소요 시간
QA	★	QA_Automation Test Selector	6 days 2 hr - [ms13]_331939_Z64	1 mo 1 day - #715	3 hr 13 min

PvP test

S	W	Name	최근 성과	최근 실패	최근 소요 시간
Yellow	Yellow	Manual FO4 PvP Test Automation (Club Friendly Mode)	28 days - [ms13]_323142_75	2 mo 14 days - #65	9 min 1 sec
Blue	Yellow	Manual FO4 PvP Test Automation Fantasy Match	28 days - [ms13]_323142_91	1 mo 13 days - #65	2 hr 13 min
Blue	Yellow	Manual FO4 PvP Test Automation Invitational Match	5 mo 9 days - [dex]	5 mo 9 days - #661_227236_8	17 min
Red	Yellow	Manual FO4 PvP Volta Live	1 mo 9 days - [dex]	#3	1 hr 54 min
Blue	Yellow	Soak FO4 PvP FantasyMatch	7 hr 39 min - [dex]	-	3 hr 36 min
Yellow	Yellow	Soak FO4 PvP FantasyManager(dex)	3 days 23 hr - [dex]	332132_6322	3 hr 49 min

```
graph TD; Scheduler["Scheduler cron H/30"] --> IsBuilding{Is building?}; IsBuilding -- No --> PvPEorPVP["PvE or PvP Automation"]; PvPEorPVP -- Parallel execution --> GameServer[Game Server]; GameScript[GameScript] --> JenkinsSlave[... Jenkins Slave]; JenkinsSlave --> Matchmaking[Matchmaking]; Matchmaking --> GameServer;
```

Server Automation test

Java, Serenity, Behavior Driven Development

Set up the Server automation test environment
Implemented scripts of BDD based on Serenity BDD
Supporting the CI / CD Daily test Jobs for Dev and QA team

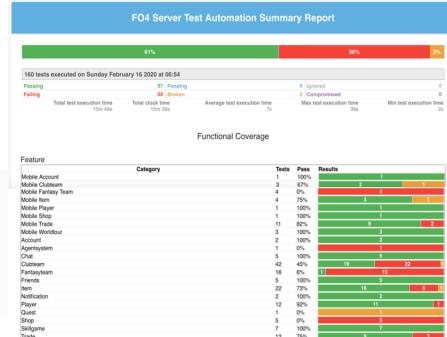
Client Automation test

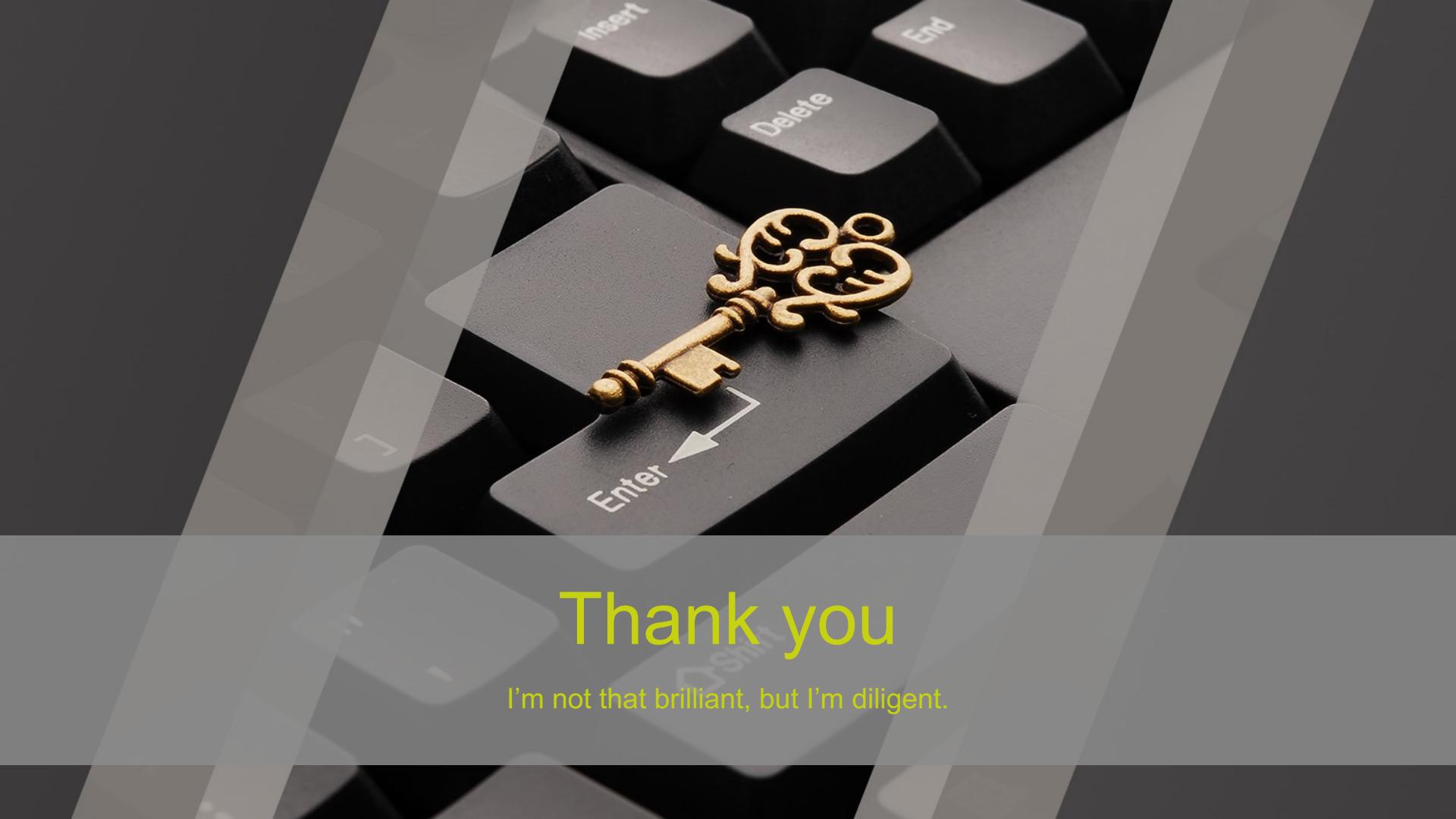
Jenkins Pipeline, PowerShell, Python, C#

Set up the Autosmoke test and Soak test by Jenkins Pipeline
Implemented PC and Mobile Client Scripts based on C#
Supporting the CI / CD Daily test Jobs for Dev and QA team



Jenkins





Thank you

I'm not that brilliant, but I'm diligent.