SCP 1.2.0 Change Request

Justin| Lingxing | Ming Yu | Tanny | Keeve | Hong Liang

Section 5, Team 5

Overview:

To first prepare for the change, our team utilized the RACI approach to assign respective tasks that were required to be done. The team then followed up with a risk assessment for potential risks during and after implementation. This process was generally executed on our ticketing system and then tested and documented with putty and git. As we identified potential risks such as security breaches, we set up further monitoring and did additional hardening with Gunicorn. This report gives the highlights of the changes made, and supporting documents can be found in this same folder.

This report covers:

- Summary of changes
- Test and configuration management (in folder)
- RACI for task delegation
- Change Management Workflow Diagram (in folder for sharper image)
- Quality Assurance
- High level system architecture diagram
- Ticket log for change request (in folder as pdf)

Summary of Changes:

- Current corporate website updated with new htm file from Fantasma Web Services
- Implemented improved security script from QA team
- POST CHANGE: Spun up visualping to do content monitoring on the corporate website
- POST CHANGE: Additional hardening using Gunicorn.

Patch notes summary for Change Request:

BUILD: 1.1.0
Made changes to html pages
Patched QA security related scripts
Edited Nginx configurations to only allow external content from Google

Configuration Management:

Deployment Environment	AWS EC2

Instance type	AWS T2.micro
vCPU	1
Memory (GIB)	1
Instance Storage	EBS only
Network	vpc-9fcee6f8
Storage Size	10GB
Volumn Type	General Purpose SSD (gp2)
IOPS (Input/Output Operations Per Second)	100/3000
Encrypted	No

Opertating System	Red/centos (Linux)

Web Server	Nginx 1:1.14.1-2.34

Web framework	Django v2.1.5

Language	Python 2.7
pip	9.0.3

Domain names	http://www.g5t5.tk
	https://www.g5t5.tk

SSL certification	Certificate generated from letsencrypt

Montoring tools	htop
	Cloudwatch
	Nagios

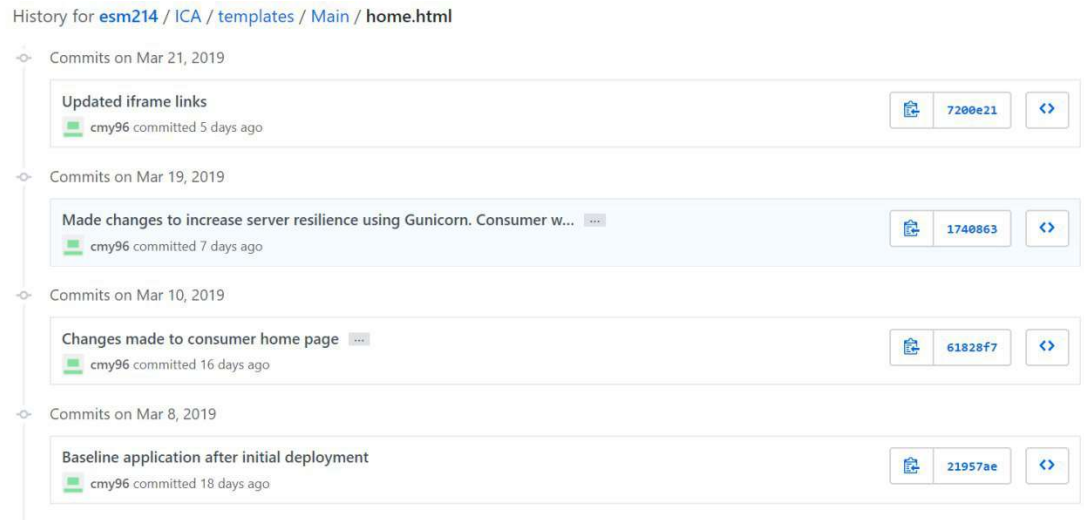
Ports	8000
	8001
	8002
	8999 (For COO monitoring)

Firewall rules	22 (ssh)
	80 (http)
	443 (https)
	8000-8999

The other configuration management documents for other patches before and after this change request are found in this folder as:

1. CM_1.0.0_08022019.xlsx
2. CM_1.0.1_10022019.xlsx
3. CM_1.0.2_17022019.xlsx
4. CM_1.0.3_18022019.xlsx
5. CM_1.0.4_21022019.xlsx
6. CM_1.1.0_10032019.xlsx
7. CM_1.1.1_17032019.xlsx

In addition, our team utilized a git repository to help keep track of the baselines of the app version. For this change request we made the following updates to the folder:

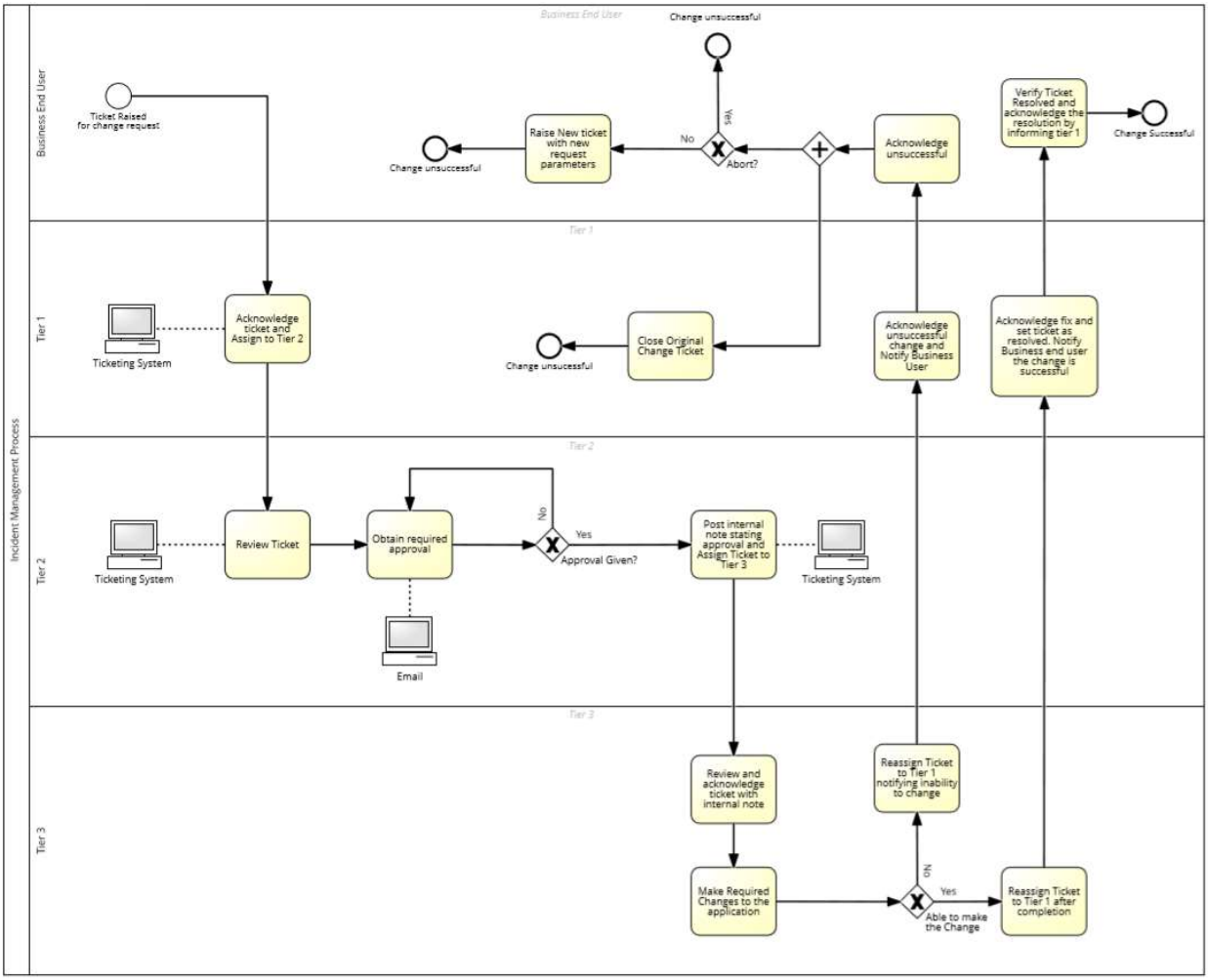


RACI Arrangement:

Workflow Process	Immediate Team					Virtual Team	
	Project Manager	Tier 1: Communications	Tier 2: Business Analyst	Tier 3: Development and Infrastructure	Security	Change Requestor	Business User
Acknowledge change request	I	R	I	I	I	A	-
Review/assess impact and risk of change request	A	I	R	I	C	-	-
Obtain approval for change	A	I	R	I	I	-	-
Implement required changes	A	I	I	R	C	-	-
Post-implementation regression testing	A	I	I	R	C	-	-
Resolve change request ticket	A	R	I	I	I	I	-
Notify Business User and conduct user satisfaction survey	A	R	C	I	I	-	I

In order to ensure clarity of roles and responsibilities, the team delegated the following roles using the RACI approach to manage the change request. We then followed to perform the change using our change request workflow (shown in the next page)

Change Request Workflow (for sharper image refer to pdf in folder)



Quality Assurance:

Quality Objectives

- To achieve 100% uptime during Project Launch (Week 7)
- To achieve 99.9% (3-nines) uptime beyond Project Launch week
- To ensure that the production system is secure and fault tolerant
- To ensure that the production system is able to respond to user requests as fast as a typical commercial website (Avg. time to load for commercial site: ~8.7s)
- To ensure that the production system is able to support at least 50 concurrent users at all times
- To respond to internal & external requests, incidents and changes to the system within 3 working days, or some other reasonable alternative timeframe suitable for the scenario
- To ensure that any changes and updates to the production system meets the change requester's requirements for deployment

Quality Management Approach

The quality management approach is to ensure that all personnel in the project "does the right things, right, the first time". Thus avoiding misunderstandings of what to produce and how, resulting in productivity losses, quality losses and schedule delays. This is going to be achieved by, for example:

- Defining the response strategy to falling quality, or missing the quality objectives stated above
- Reviewing all information that may affect the quality of the product and service e.g. monitoring AWS reports, conducting system load testing, etc.
- Setting automated responses to potential problems, e.g. alerts/notifications to all service personnel when a component of the production system fails, scripts to restart vital services when the system has stopped responding for a predefined timespan, etc.
- Clear and timely communication by Tier 1 & 2 support agents with relevant stakeholders regarding requests, incidents and changes to the production system

Quality Control

The following points sets out and describes the monitoring and processes used to verify that the enterprise solution is of acceptable quality:

IT Infrastructure

- AWS CloudWatch is used to monitor server status and CPU usage. In the event that these metrics vary beyond normal thresholds (e.g. CPU usage < ~3%), email notifications will be sent to all members of the team for further investigation.
- The response time of the production system meets business requirements, as the application webpage has been verified to load under 2s using an external website speed testing service.
- The security team regularly checks for (weekly) and applies patches for applications running on the production system. This will ensure that security loopholes are quickly discovered and closed before they can be exploited by bad actors to compromise the system
- Security of data to/from the server is ensured by HTTPS end-to-end encryption
- Load testing using Apache JMeter and Siege has been performed on the production system to ensure that it is able to meet and surpass the business requirements of the organisation. The team has noted that the SCP application is able to support upward of 2000 user requests per second.

SCP Enterprise Application

- The SCP enterprise application's uptime monitoring is performed by using the Freshping service by Freshworks. This service regularly polls the service ports used by the application, and alerts the team if the application is unresponsive or down.
- To ensure business uptime and continuity, the SCP application is served on multiple ports, and load balanced using nginx. This ensures that the corporate website stays accessible even in the event that any single hosting port goes down.

Roles and responsibilities

Name	Role	Responsibilities
Justin Choy	Operations Manager	• Oversees and coordinates the entire team • Monitor AWS CloudWatch & Freshping for abnormal system statuses
Chua Ming Yu	Security	• Installation and hardening of web server & relevant applications (e.g. security patching) • Configuration of server encryption, security groups and access control
Tanny Lai	Tier 1 - Communications	• Communicate and provide end-users with technical support • Escalate requests to proper parties for solutioning • Provide clear and timely updates to relevant stakeholders
Keeve Quek	Tier 2 - Business Analyst	• Monitor and analyze system utilisation charts, and identify the root cause of abnormal chart patterns • Set up of monitoring for the web application • Setting alarms in AWS to alert the team of System Failure, Instance Fail and abnormal CPU usage
Guo Lingxing	Tier 3 - Development and Infrastructure	• Test build for business functionality with Quality Assurance • Maintain configuration management log to track changes to the production system • Configure ports and load balancing activities of the system
Ng Hong Liang	Quality Assurance	• Identify the posing risks and address the risks by providing suitable recommendations • Checks and ensure that the production & support environment meets business requirements • Test build for business functionality with Development and Infrastructure

Tools

The list below sets out the tools that the project will utilize to help manage and control quality of the enterprise solution:

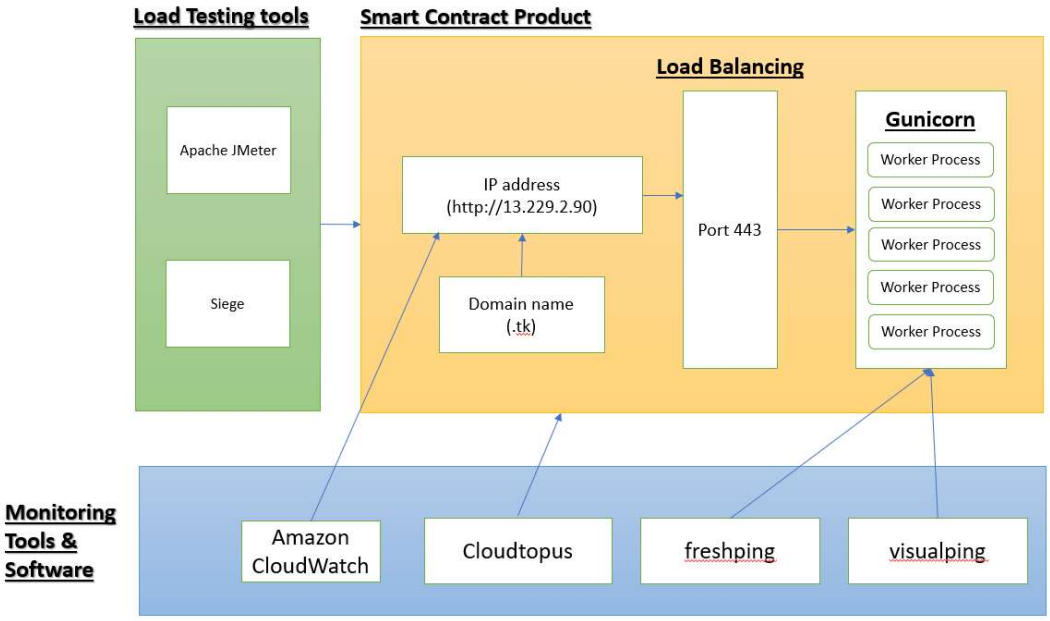
- Google Drive/GitHub [Storage of Project Documentation]
- Google Sheets/Excel spreadsheet application [Logging of Deployment and Change Management]
- Signavio Process Modeling Platform [Define incident response process]
- IS214 ESM Ticketing System (osTicket Support Ticket System) [To facilitate incident/change requests and tracking]
- Apache JMeter [Load testing of Infrastructure]
- Siege [Load testing of Infrastructure]
- https://www.webpagetest.org/ [Test application website response time]
- AWS CloudWatch [Monitor Infrastructure uptime]
- Freshping by Freshworks [Monitor Application uptime]

Change Request Testing

For this change request, our group conducted the following tests on the updated corporate website to prove that the change was effective and met the identified qualities. All of which were successful:

Smoke Test	
web search of http://g5t5.tk returns successful http response	Success
use server checker https://smallseotools.com/check-server-status/ to ensure that the deployed server is available to general public	Success (HTTP response: 200)
Functional Test	
the index page loads as expected and the iframe link loads properly as stated by Fantasma Serrvices	Success
Load Testing	
Executed the following siege command with 0 failed transactions: `siege -c250 -t60s http://g5t5.tk`	Success

High Level architectural diagram:



This is the overview diagram after implementing the change request and managing our identified risks.