

Infidels Project

1. Background

2. Scope of Work

(1) Preparation

(2) Solution Architecture Design

4. Timeline

5. Effort Estimation

1. Background

Web server presenting the information inserted by the user through API to a Data Base.
Full CI-CD pipeline to test, tag and deploy the components.

1st product – API server

REST API server: POST data from user to the Data Base

- Server will be dockerized
- Post Data will contain: user_id, feedback
- Logbook.xml will contain: [info]@timestamp feedback
- Config will contain:
 1. db_address
 2. db_user
 3. db_pwd
 4. api_url

2nd product – Web server

- Server will be dockerized
- Login field: will receive a username that will pull the feedback data using the user_id from Data Base
- Secretfile will contain:
 1. db_user
 2. db_pwd
- Logs will be out to a txt file

3rd QA

- Will check for a successful response from API server
- Check Web server is accessible
- Check successful Login

4rd DBA

- Data Base server will be dockerized
- Decide on Data Base type
- Prepare:
 1. Data Base config
 2. Structure
 3. db_user
 4. db_pwd

5th DevOps:

- Docker Image for:
 1. API
 2. Web
 3. DB
- Ansible Playbook roles:
 1. Common (pre-config)
 2. API server
 3. Web server
 4. Data Base server
 5. Elasticserach
 6. Kibana
 7. Metricbeat – System logs
 8. Filebeat – API logs
- Git structure build and versioning strategy
- Jenkins server with HA in mind
- Docker Hub configuration

No effort estimation for this part as everything should run by the dead line.

2.Scope of Work

(1) Preparation

- Fill pre requirements checklist: versions, application specific configurations
- Design the GIT repository structure
- Decide on the versioning strategy
- Understand the dependencies from other teams: variables, ports etc.

(2) Solution Architecture Design

The focus will be on the following:

- Fault Tolerance and High Availability – CI/CD pipeline
- Auto-scalability and Elasticity – Deployment to multiple hosts / Single one

4. Timeline

- The project start date: 04.09.19 should be defined
- The project end date: 25.09.19

5. Effort Estimation

5.5.1.API Server

Step	Tasks	Hours
API server	<ul style="list-style-type: none">Choose framework and write the code	1.5
External config	<ul style="list-style-type: none">Create and use an external config file	1
DB	<ul style="list-style-type: none">Connect to Data Base	1
SubTotal, hours		3.5

5.5.2.Web Server

Step	Tasks	Hours
Web Server	<ul style="list-style-type: none">Develop the Server	1
Login Mechanism	<ul style="list-style-type: none">Develop the mechanism	3
DB	<ul style="list-style-type: none">Develop an operational flow to DB	1
Secret file	<ul style="list-style-type: none">Create a secret file	0.5
UI	<ul style="list-style-type: none">Develop UI for feedback view	1
Logs	<ul style="list-style-type: none">Push logs to a txt file	0.5
SubTotal, hours		7

5.5.3.QA

Step	Tasks	Hours
Test API	<ul style="list-style-type: none"> Create a test file for the API server 	2
Test Web server	<ul style="list-style-type: none"> Create a test file for the Web server 	3
Test Login	<ul style="list-style-type: none"> Create a test file for the Login process 	5
SubTotal, hours	-	10

5.5.3.DBA

Step	Tasks	Hours
Data Base	<ul style="list-style-type: none"> Set up the DB 	2
Infrastructure	<ul style="list-style-type: none"> Create a Dockerfile Create a db.config Decide on the structure Create the username and password 	2
SubTotal, hours	-	4