

Building the VIRNECT REMOTETest Automation Environment

QAOps Implementation

VIRNECT QA Team

Sungtae Kim

I. Project Overview

Purpose

- Enhance quality through test automation implementation.
- Improve development efficiency by integrating CI/CD and Agile processes.
- Systematically manage tests and code based on backlog management.
- Strengthen collaboration between development and QA teams.

Goals

- Write and execute automated test code based on user stories.
- Operate automated tests through Jenkins and Slack integration.
- Optimize backlog management using the Jira QAOps project.
- Improve test code quality and apply the code review process.
- Establish seamless communication using GitHub and Slack.

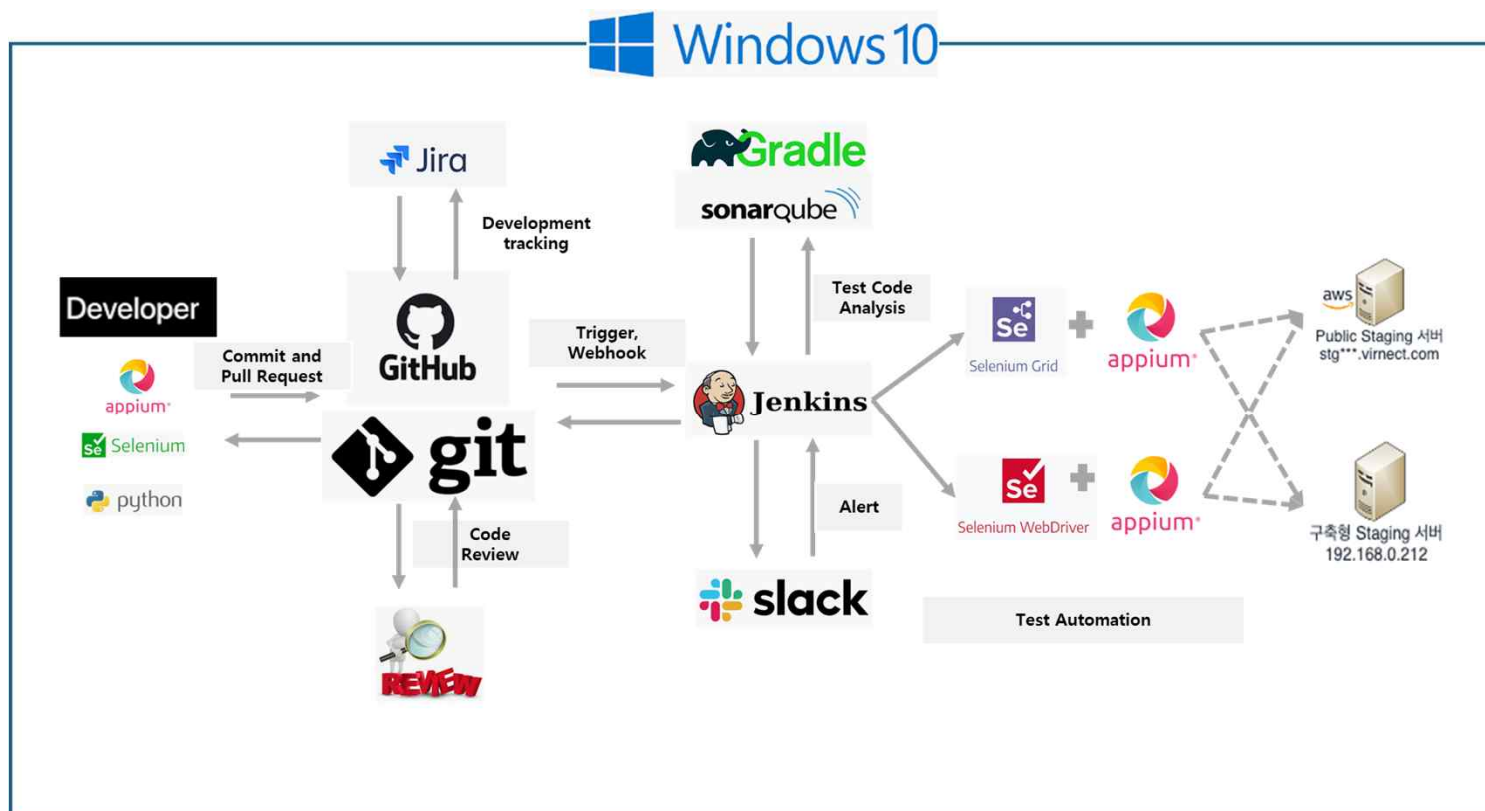
I. Project Overview

Schedule

No.	Schedule	Category	Key Tasks	Responsible Person		Remarks
1	From September 4, 2023 , to September 15, 2023 .	Setup	<ul style="list-style-type: none"> Install Appium, Selenium, Selenium Grid, Python, and Jenkins on the test environment server. 	VIRNECT Co., Ltd. QA Team	Kim Sung-tae, Senior Engineer	
2	From September 18, 2023 , to September 25, 2023	Setup	<ul style="list-style-type: none"> Create the company's official GitHub repository. Integrate GitHub with Jira. Connect Jenkins with Slack. 	VIRNECT Co., Ltd. QA Team	Kim Sung-tae, Senior Engineer	
3	From October 10, 2023 , to December 29, 2023	Setup and Study	<ul style="list-style-type: none"> Study the basic usage of Jenkins. Integrate Jenkins with GitHub. Connect Jenkins with Slack. Study the Groovy language for writing Jenkins Pipeline scripts. 	VIRNECT Co., Ltd. QA Team	Kim Sung-tae, Senior Engineer	
4	From January 8, 2024 , to December 27, 2024	Study and Setup	<ul style="list-style-type: none"> Study Python. Create a QAOps project in the company's Jira. Register the product backlog in Jira. Assign story points to product backlog items. Register user stories in the sprint backlog. Operate sprint cycles in either 2-week or 4-week intervals. Conduct sprint backlog reviews. Review the implemented test automation code and conduct a test automation demo. 	VIRNECT Co., Ltd. QA Team	Kim Sung-tae, Senior Engineer	

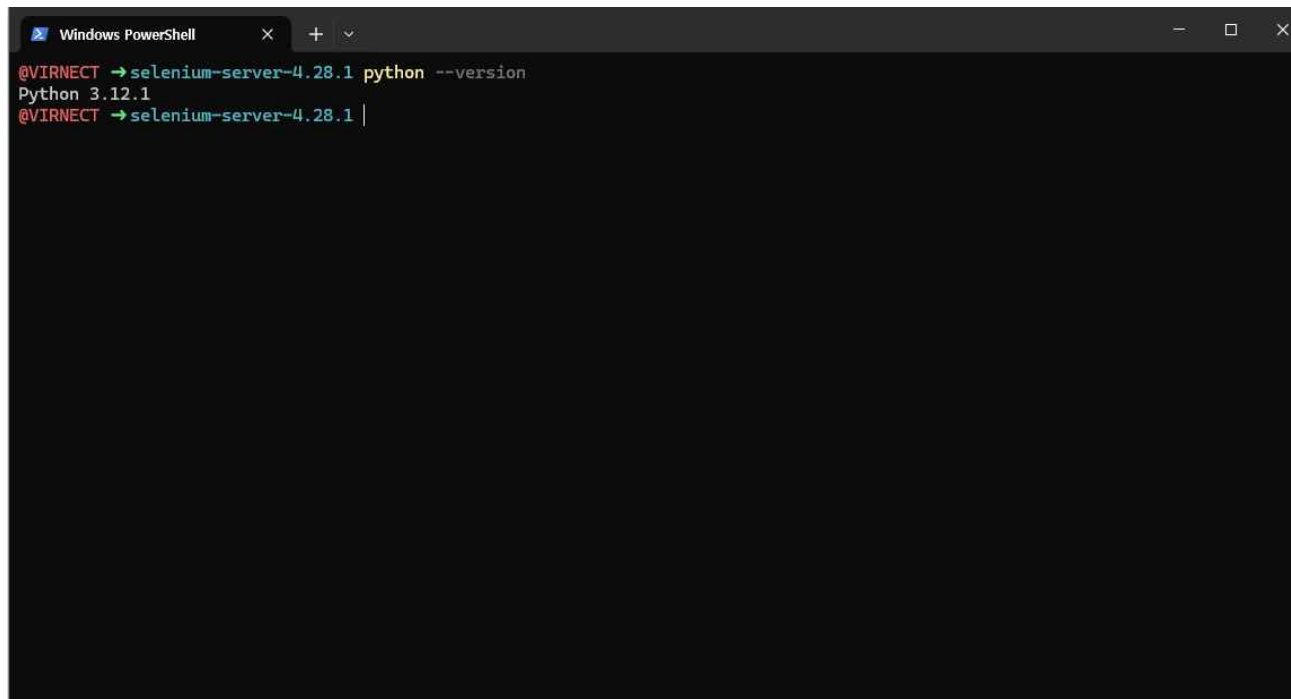
I. Project Overview

CI/ CD Pipeline Architecture in a Team Environment (Windows 10)



I. Project Overview

1. Install and verify Python on the test environment server.



```
Windows PowerShell
@VIRNECT → selenium-server-4.28.1 python --version
Python 3.12.1
@VIRNECT → selenium-server-4.28.1 |
```

I. Project Overview

2. Install and verify Appium on the test environment server.

```
Windows PowerShell
@VIRNECT → selenium-server-4.28.1 appium -v
2.12.1
@VIRNECT → selenium-server-4.28.1 npm list -g --depth=0
C:\Users\VIRNECT\AppData\Roaming\npm
+-- appium@2.12.1
+-- artillery@2.0.21
+-- npm@10.9.0
`-- swagger@0.7.5

@VIRNECT → selenium-server-4.28.1 |
```


I. Project Overview

3. Install and verify Selenium on the test environment server.

```

Windows PowerShell
@VIRNECT → selenium-server-4.28.1 pip show selenium
Name: selenium
Version: 4.25.0
Summary: Official Python bindings for Selenium WebDriver
Home-page: https://www.selenium.dev
Author:
Author-email:
License: Apache 2.0
Location: C:\python312\Lib\site-packages
Requires: certifi, trio, trio-websocket, typing_extensions, urllib3, websocket-client
Required-by: Appium-Python-Client, pytest-webdriver
@VIRNECT → selenium-server-4.28.1 npm list -g selenium-webdriver
C:\Users\VIRNECT\AppData\Roaming\npm
-- (empty)

@VIRNECT → selenium-server-4.28.1 |

```

I. Project Overview

4. Install and verify Selenium Grid on the test environment server.

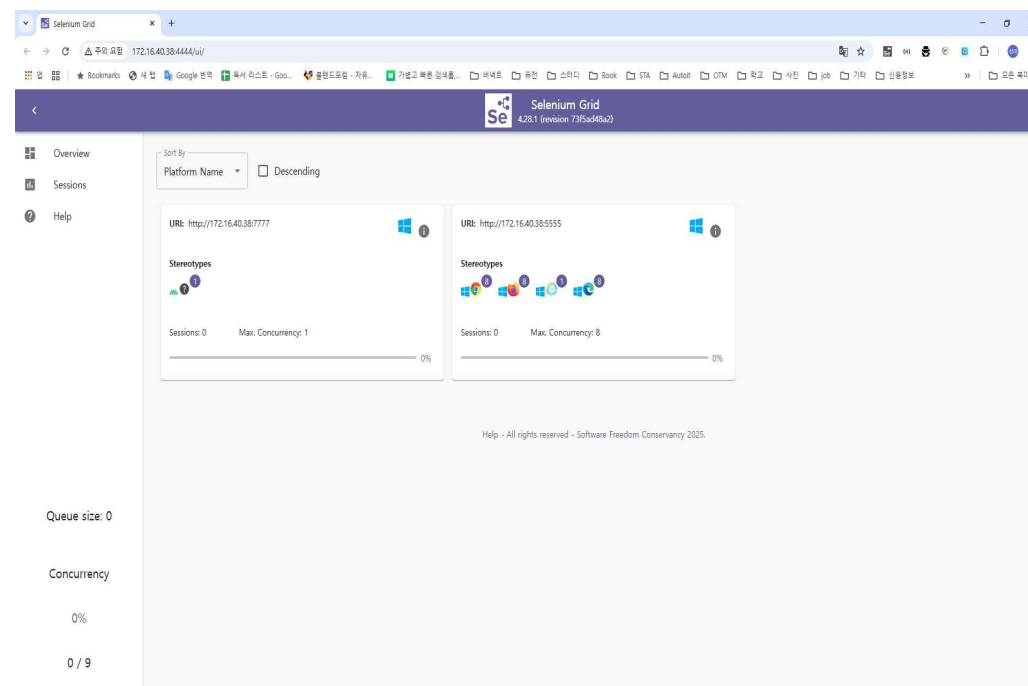
```
Windows PowerShell
@VIRNECT → selenium-server-4.28.1 dir selenium-server-*.jar

디렉터리: C:\selenium-server-4.28.1

Mode                LastWriteTime         Length Name
----                -
-a-----         2025-02-07 오전 10:12         43301857 selenium-server-4.28.1.jar

@VIRNECT → selenium-server-4.28.1 |
```

그림의 제목이나 설명을 입력하세요



그림의 제목이나 설명을 입력하세요

I. Project Overview

5. Install and verify Jenkins on the test environment server.

```

Windows PowerShell
@VIRNECT → C:\ wmic service get name,displayname,state | findstr /I "jenkins"
Jenkins
Running
@VIRNECT → C:\
  
```

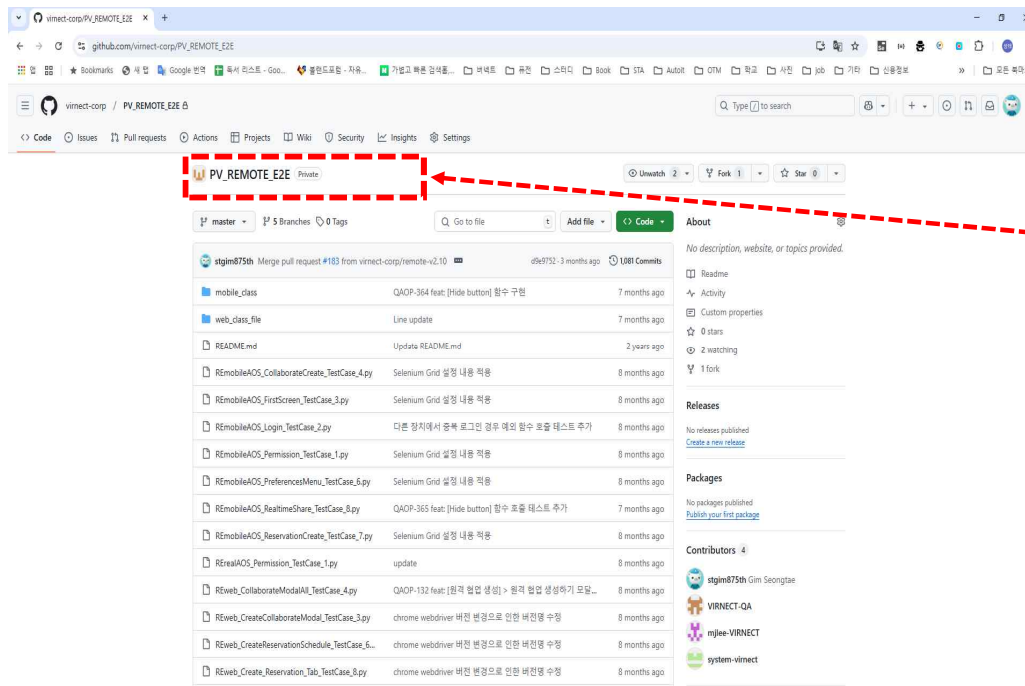
그림의 제목이나 설명을 입력하세요

S	W	Name ↓	최근 성공	최근 실패	최근 소요 시간
🔄	☀️	artillery-load-test	—	—	—
✅	☀️	Remote_Mobile_AOS_v2.10	6 mo 27 days #14	7 mo 9 days #6	4 min 59 sec
✅	☁️	REMOTE_Mobile_AOS_v2.10_E2E_REGRESSION_TEST	8 mo 7 days #7	7 mo 9 days #8	22 min
✅	☀️	REMOTE_Mobile_AOS_v2.10_E2E_REGRESSION_TEST_Coverage	7 mo 22 days #1	—	22 min
✅	☁️	Remote_Web_v2.10	7 mo 17 days #18	7 mo 17 days #16	4 min 3 sec
✅	☁️	REMOTE_WEB_v2.10_E2E_REGRESSION_TEST	8 mo 8 days #5	8 mo 12 days #6	30 min
✅	☁️	REMOTE_WEB_v2.10_E2E_REGRESSION_TEST_Coverage	7 mo 22 days #5	7 mo 22 days #2	33 min

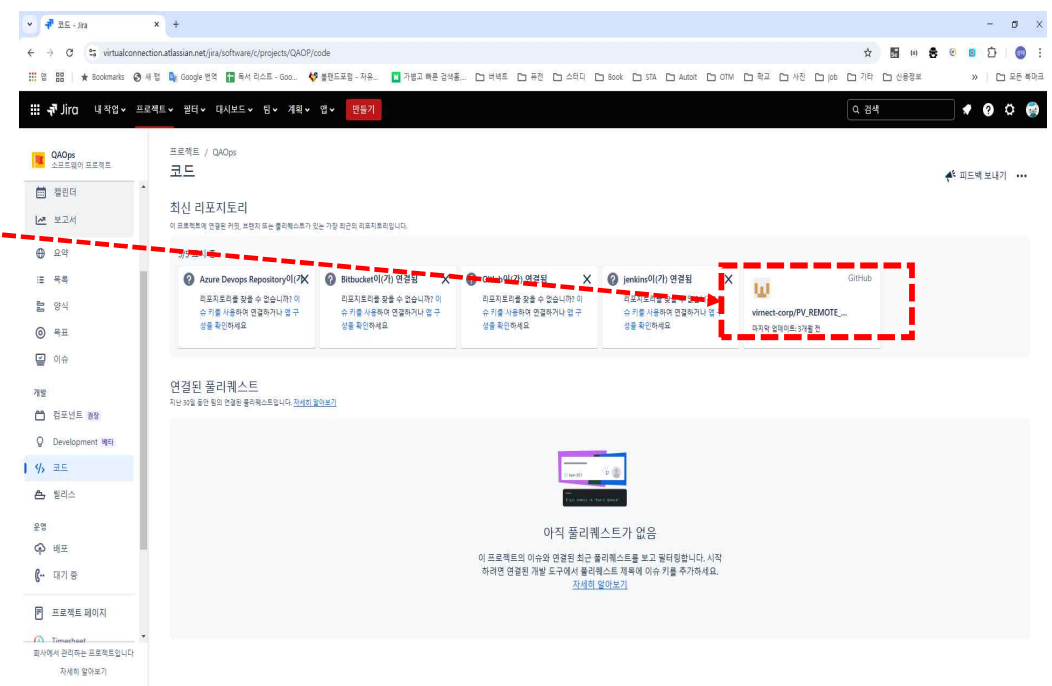
그림의 제목이나 설명을 입력하세요

I. Project Overview

6. Create a repository on the company's official GitHub to store code, register it in the company's Jira project, and establish a connection.



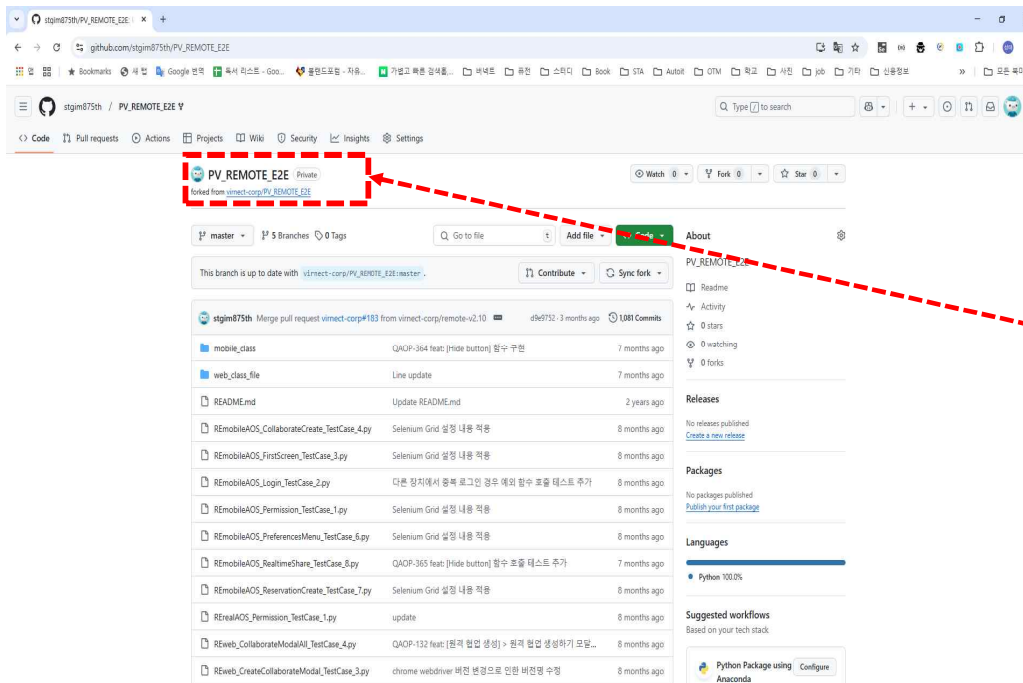
그림의 제목이나 설명을 입력하세요



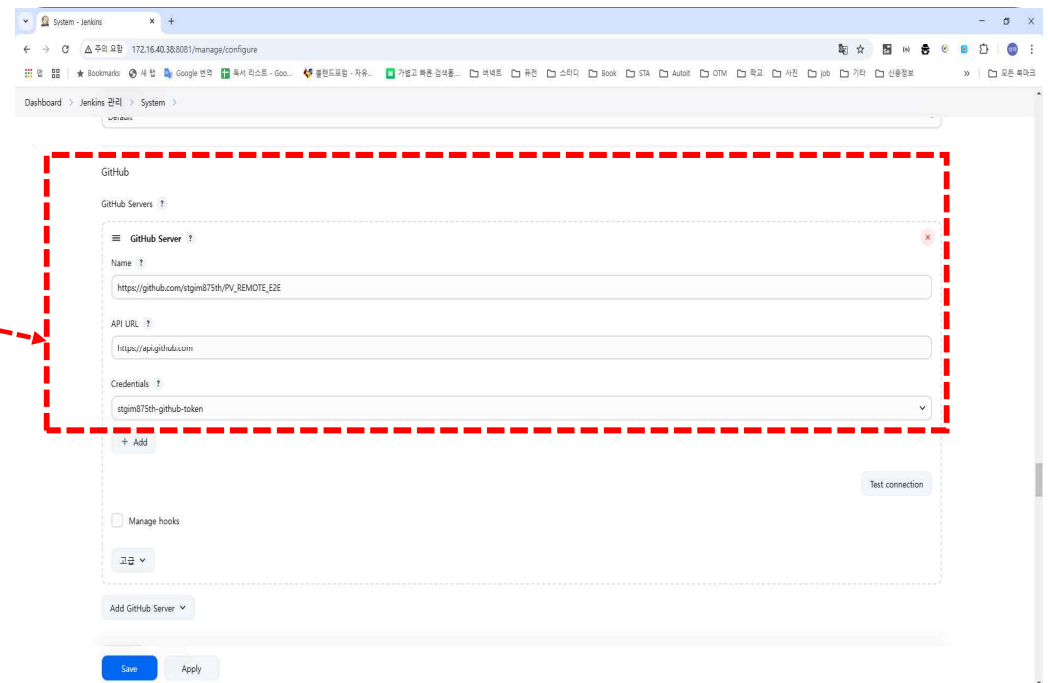
그림의 제목이나 설명을 입력하세요

I. Project Overview

7. Fork the code repository created on the company's official GitHub to your personal GitHub account, then register it in Jenkins installed on the test environment server.



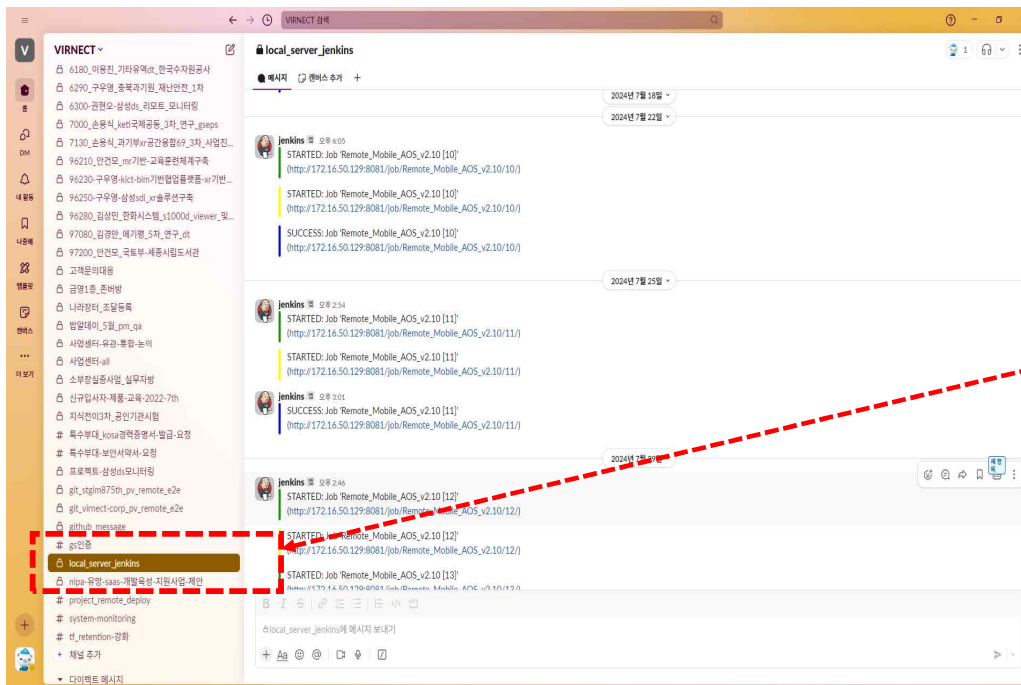
그림의 제목이나 설명을 입력하세요



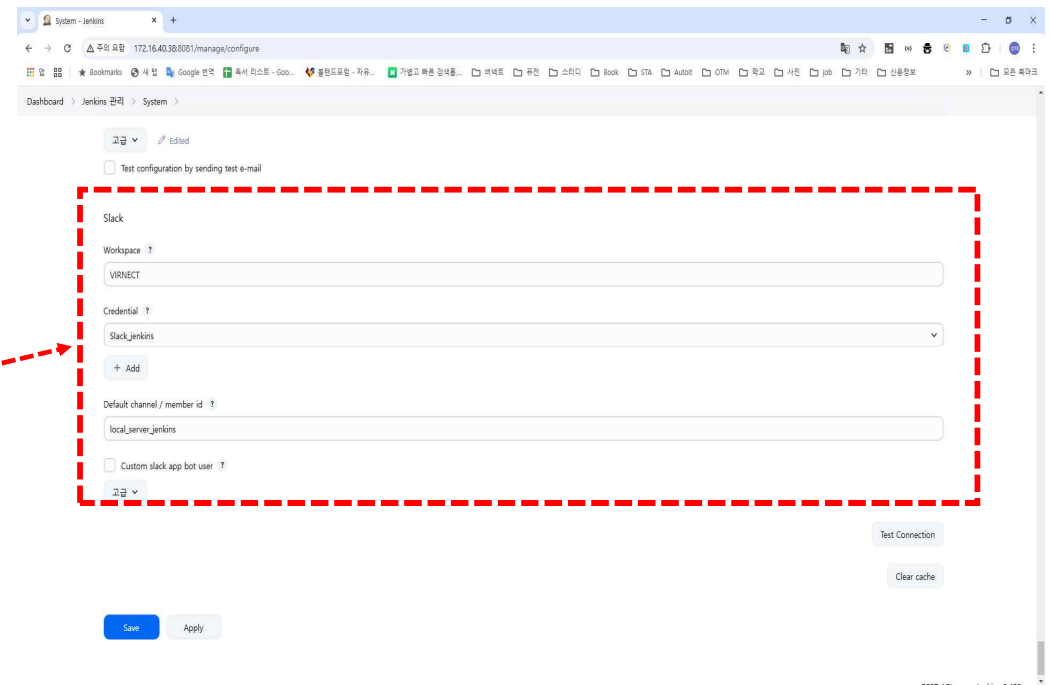
그림의 제목이나 설명을 입력하세요

I. Project Overview

8. After installing Jenkins in the company's Slack channel, authenticate and connect the Slack channel's Jenkins channel with the Jenkins installed in the test environment.



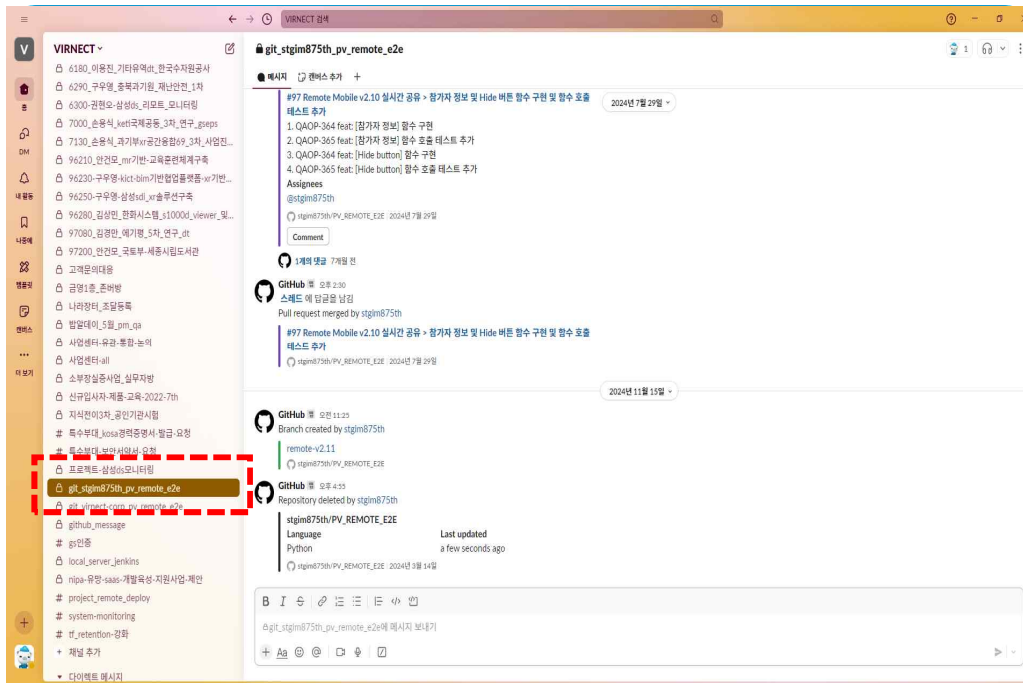
그림의 제목이나 설명을 입력하세요



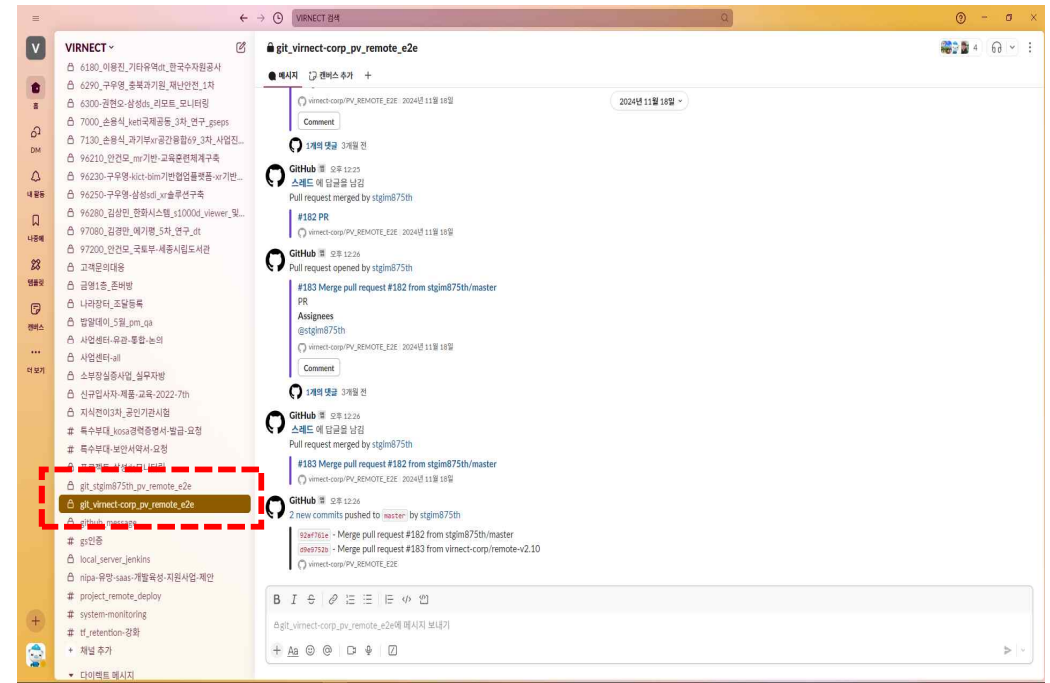
그림의 제목이나 설명을 입력하세요

I. Project Overview

9. Connect the official GitHub repository and the forked personal account repository to the company's Slack channel.



그림의 제목이나 설명을 입력하세요



그림의 제목이나 설명을 입력하세요

I. Project Overview

10. Create a **QAOps project** in the company's **Jira project**, and manage the **product backlog** and **sprint backlog** with sprint cycles operating in **2- week or 4- week intervals**.

The screenshot shows the Jira QAOps project backlog. The left sidebar contains navigation options like Backlog, Sprint Backlog, Calendar, and Reports. The main area displays a list of tasks with the following columns: 유형 (Type), # 키 (Key), 요약 (Summary), 상태 (Status), Sprint, @ 담당자 (Assignee), 우선 순위 (Priority), 레이블 (Labels), 만들 (Created), and 업데이트 (Updated).

유형	# 키	요약	상태	Sprint	@ 담당자	우선 순위	레이블	만들	업데이트
QAOP-3	Remote_Web_Mobile E2E 자동화 테스트 실행을 위해 Local J...	완료		김성태_QA			2024년 1월 31일	2024년 4월 26일	
QAOP-50	Remote_Web_Mobile Sonarqube 설치 및 환경 구성을해야한...	완료		김성태_QA			2024년 2월 8일	2024년 4월 26일	
QAOP-93	Remote Web 로그인 메뉴를 함수로 구현해야한다.	완료		김성태_QA			2024년 3월 14일	2024년 3월 15일	
QAOP-129	Remote Mobile(AOS) 환경 설정 하위 메뉴들에 Location serc...	완료	QAOP 1 스프린트	김성태_QA			2024년 3월 27일	2024년 4월 5일	
QAOP-146	Local PC에 Windows_Jenkins 설치 및 자동화 진행에 필요한 p...	완료	QAOP 1 스프린트	김성태_QA			2024년 3월 20일	2024년 4월 15일	
QAOP-147	Remote Web 자동화 테스트 코드 실행을 위한 Github와 Jenki...	완료	QAOP 1 스프린트	김성태_QA			2024년 3월 20일	2024년 4월 15일	
QAOP-148	Remote Mobile 자동화 테스트 코드 실행을 위한 Github와 Je...	완료	QAOP 1 스프린트	김성태_QA			2024년 3월 20일	2024년 4월 15일	
QAOP-149	Sonarqube에 Local Jenkins pipeline 구성	완료	QAOP 1 스프린트	김성태_QA			2024년 3월 20일	2024년 4월 15일	
QAOP-145	Local PC Docker에 Sonarqube 설치 진행	완료	QAOP 1 스프린트	김성태_QA			2024년 3월 20일	2024년 3월 20일	
QAOP-388	Remote_Web_E2E 함수 구현	해야 할 일		김성태_QA			2024년 7월 29일	2024년 11월 22일	
QAOP-389	Remote_Web_E2E 함수 호출 테스트 설계	해야 할 일		김성태_QA			2024년 7월 29일	2024년 11월 22일	
QAOP-390	Remote_Mobile(AOS)_E2E 함수 구현	해야 할 일		김성태_QA			2024년 7월 29일	2024년 11월 22일	
QAOP-391	Remote_Mobile(AOS)_E2E 함수 호출 테스트 설계	해야 할 일		김성태_QA			2024년 7월 29일	2024년 11월 22일	

I. Project Overview

11. Write automation code for the company's product features based on **user stories** from the **sprint backlog** registered in the company's **Jira QAOps project**.

```

1  import inspect
2  import pytest
3
4  from appium.webdriver.common.appium import Appium
5  from selenium.webdriver.common.action_chains import ActionChains
6  from selenium.webdriver.common.actions.action_builder import ActionBuilder
7  from selenium.common.exceptions import TimeoutException
8  from selenium.webdriver.common.actions import interaction
9  from selenium.webdriver.common.actions.action_builder import PointerInput
10 from selenium.webdriver.support.ui import WebDriverWait
11 from selenium.webdriver.support import expected_conditions as EC
12
13 # Remote Login 화면
14 class RemoteLogin:
15     def __init__(self, driver):
16         self.driver = driver
17
18     # 로그인 화면
19     self.login_screen = (Appium.XPATH, "//android.widget.FrameLayout/android.widget.FrameLayout/android.view.View/android.view.View/android.view.View")
20
21     # Remote 로그인 > 아이디 입력 필드
22     self.login_field = (Appium.XPATH, "//android.widget.ImageView[@content-desc='로그인 화면']//android.widget.EditText[1]")
23
24     # Remote 로그인 화면 > 비밀번호 필드
25     self.password_field = (Appium.XPATH, "//android.widget.ImageView[@content-desc='로그인 화면']//android.widget.EditText[2]")
26
27     # Remote 로그인 화면 > 로그인 버튼
28     self.login_button = (Appium.XPATH, "//android.widget.Button[@content-desc='로그인']")
29
30     # 워크스페이스 모양
31     self.workspace_model = (
32         Appium.XPATH, "//android.widget.FrameLayout/android.widget.FrameLayout/android.view.View/android.view.View/android.view.View"
33     )
34
35     # 워크스페이스 모양
36     self.my_workspace = (Appium.XPATH, "//android.view.View[@content-desc='LV UP RISING']")
37
38     # 다른 장치에서 원격 로그인 > 접속 종료 모양
39     self.disconnect_terminated = (
40         Appium.XPATH, "//android.view.View[@package='com.virnect.remote.mobile2.openrealise' and @bounds='[120,914][968,1580]']"
41     )
42
43     # 다른 장치에서 원격 로그인 > 접속 종료 모양 > [확인]
44     self.disconnect_terminated_confirm = (
45         Appium.XPATH, "//android.widget.Button[@content-desc='확인']"
46     )
47
48     # also: 실패 및 예외처리 발생 시 Fail 처리 자동 함수
49     def handle_exceptions(method):
50         def wrapper(*args, **kwargs):

```

그림의 제목이나 설명을 입력하세요

```

1  class CreateReservation:
2
3     # 예약 생성 화면 > 예약 시간 > 시간 버튼을 순차적으로 클릭하기
4     @handle_exceptions
5     def select_time_hour(self, time_hour):
6         try:
7             time_hour_btn = WebDriverWait(self.driver, 5).until(
8                 EC.presence_of_element_located((self.time_hour_open)))
9             actions = ActionChains(self.driver)
10            actions.move_to_element(time_hour_btn)
11            actions.click(time_hour_btn)
12            actions.perform()
13            print(f"선택 시간 {time_hour}을 클릭했습니다.")
14            return True
15        except TimeoutException:
16            print(f"선택 시간 {time_hour}을 클릭하지 못했습니다.")
17            return False
18
19     # 예약 생성 화면 > 진행 기간 > 시간 버튼을 순차적으로 클릭하기
20     @handle_exceptions
21     def select_time_hour_click(self, time_hour_click, time_hour_message):
22         try:
23             time_hour_selection = WebDriverWait(self.driver, 5).until(
24                 EC.presence_of_element_located(
25                     by_xpath="//div[@class='modal reservation-modal']/div[@role='tooltip' and @class='popover select-options scroll']/div[@class='popover--body']/div[@class='ps ps--active']/div[@class='select-optionbox']/button[contains(text(), '{time_hour_click}')]"))
26             actions = ActionChains(self.driver)
27             actions.move_to_element(time_hour_selection)
28             actions.click(time_hour_selection)
29             actions.perform()
30             print(time_hour_message)
31             time.sleep(1)
32             return True
33         except TimeoutException:
34             print(f"선택 시간 {time_hour_click}을 클릭하지 못했습니다.")
35             return False
36
37     # 예약 생성 화면 > 진행 기간 > 마우스 이동하여 화면을 하 컨트롤 > 화면을 아래로 progress bar 이동하기
38     @handle_exceptions
39     def reservation_time_hour_progress_bar_down_move(self):
40         result = self.scroll_element(self.scroll_control, self.time_hour_progress_bar)
41         if result:
42             print("예약 생성 화면 > 진행 기간 [시간 Select box]에서 마우스로 [아래로 이동]했습니다.")
43         else:
44             print("예약 생성 화면 > 진행 기간 [시간 Select box]에서 마우스로 [아래로 이동]하지 못했습니다.")
45         return result
46
47     # 예약 생성 화면 > 진행 기간 > 마우스 이동하여 화면을 상 컨트롤 > 화면을 위로 progress bar 이동하기
48     @handle_exceptions
49     def reservation_time_hour_progress_bar_up_move(self):
50         result = self.scroll_element(self.scroll_control, self.time_hour_progress_bar)
51         if result:
52             print("예약 생성 화면 > 진행 기간 [시간 Select box]에서 마우스로 [아래로 이동]했습니다.")
53         else:
54             print("예약 생성 화면 > 진행 기간 [시간 Select box]에서 마우스로 [아래로 이동]하지 못했습니다.")
55         return result

```

그림의 제목이나 설명을 입력하세요

I. Project Overview

12. Push and merge the written test codes into the company's GitHub repository and the personal account repository.

The screenshot displays the Jira QAOps project overview. The left sidebar contains navigation links for various project components. The main area shows a table of tasks with the following data:

유형	# 키	요약	상태	Sprint	@ 담당자	우선 순위	레이블	만들	업데이트
QAOP-3	Remote_Web_Mobile E2E 자동화 테스트 실행을 위해 Local J...	완료		김성태_QA		2024년 1월 31일	2024년 4월 26일		
QAOP-50	Remote_Web_Mobile Sonarqube 설치 및 환경 구성을해야한...	완료		김성태_QA		2024년 2월 8일	2024년 4월 26일		
QAOP-93	Remote Web 로그인 메뉴를 함수로 구현해야한다.	완료		김성태_QA		2024년 3월 14일	2024년 3월 15일		
QAOP-129	Remote Mobile(AOS) 환경 설정 하위 메뉴들에 Location serc...	완료	QAOP 1 스프린트	김성태_QA		2024년 3월 27일	2024년 4월 5일		
QAOP-146	Local PC에 Windows_Jenkins 설치 및 자동화 진행에 필요한 p...	완료	QAOP 1 스프린트	김성태_QA		2024년 3월 20일	2024년 4월 15일		
QAOP-147	Remote Web 자동화 테스트 코드 실행을 위한 Github와 Jenki...	완료	QAOP 1 스프린트	김성태_QA		2024년 3월 20일	2024년 4월 15일		
QAOP-148	Remote Mobile 자동화 테스트 코드 실행을 위한 Github와 Je...	완료	QAOP 1 스프린트	김성태_QA		2024년 3월 20일	2024년 4월 15일		
QAOP-149	Sonarqube에 Local Jenkins pipeline 구성	완료	QAOP 1 스프린트	김성태_QA		2024년 3월 20일	2024년 4월 15일		
QAOP-145	Local PC Docker에 Sonarqube 설치 진행	완료	QAOP 1 스프린트	김성태_QA		2024년 3월 20일	2024년 3월 20일		
QAOP-388	Remote_Web_E2E 함수 구현	해야 할 일		김성태_QA		2024년 7월 29일	2024년 11월 22일		
QAOP-389	Remote_Web_E2E 함수 호출 테스트 설계	해야 할 일		김성태_QA		2024년 7월 29일	2024년 11월 22일		
QAOP-390	Remote_Mobile(AOS)_E2E 함수 구현	해야 할 일		김성태_QA		2024년 7월 29일	2024년 11월 22일		
QAOP-391	Remote_Mobile(AOS)_E2E 함수 호출 테스트 설계	해야 할 일		김성태_QA		2024년 7월 29일	2024년 11월 22일		

2. Project Results

- Established an in-house test automation process
- Built a collaborative environment by integrating Jira, GitHub, Jenkins, and Slack
- Enabled automated quality validation through CI/CD pipeline integration
- Developed and reviewed test automation scripts based on user stories
- Reduced effort and costs for repetitive and regression testing through test automation adoption