# Lab 3: Continuous Integration

*Software Testing 2021*

*2021/03/18*

# Parameterization & Exception

# Parameterization

- Execute a single test method **multiple times** with different parameters

Reference: https://www.baeldung.com/parameterized-tests-junit-5

# Parameterization - Example 1

Target function:

```java
public class Numbers {
    public static boolean isOdd(int number) {
        return number % 2 != 0;
    }
}
```

Test case:

```java
@ParameterizedTest
@ValueSource(ints = {1, 3, 5, -3, 15, Integer.MAX_VALUE}) // six numbers
void isOdd_ShouldReturnTrueForOddNumbers(int number) {
    assertTrue(Numbers.isOdd(number));
}
```

# Parameterization - Example 2

A static method that returns a **Stream** of Arguments:

```
1  private static Stream<Arguments> provideStringsForIsBlank() {
2      return Stream.of(
3        Arguments.of(null, true),
4        Arguments.of("", true),
5        Arguments.of("  ", true),
6        Arguments.of("not blank", false)
7      );
8  }
```

Test case:

```
1  @ParameterizedTest
2  @MethodSource("provideStringsForIsBlank")
3  void isBlank_ShouldReturnTForNullOrBlankS(String input, boolean expected) {
4      assertEquals(expected, Strings.isBlank(input));
5  }
```

# Exception

- **JUnit 5** Jupiter assertions API introduces the **assertThrows** method for asserting exceptions

Reference: https://www.baeldung.com/junit-assert-exception

# Exception - Example

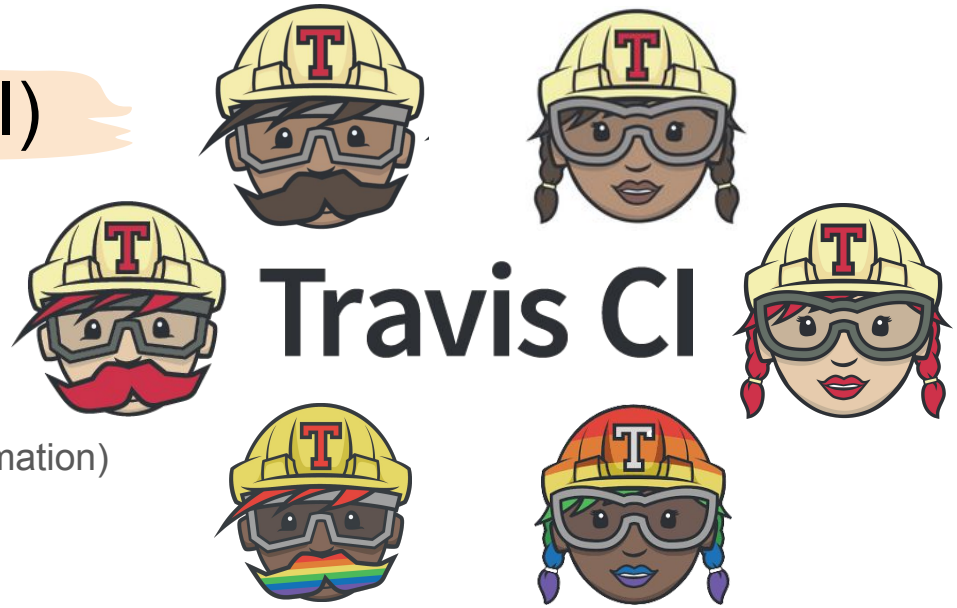
```java
@Test
public void whenExceptionThrown_thenAssertionSucceeds() {
    Exception exception = assertThrows(NumberFormatException.class, () -> {
        Integer.parseInt("1a");
    });

    String expectedMessage = "For input string";
    String actualMessage = exception.getMessage();

    assertTrue(actualMessage.contains(expectedMessage));
}
```

# Continuous Integration

# Continuous Integration (CI)

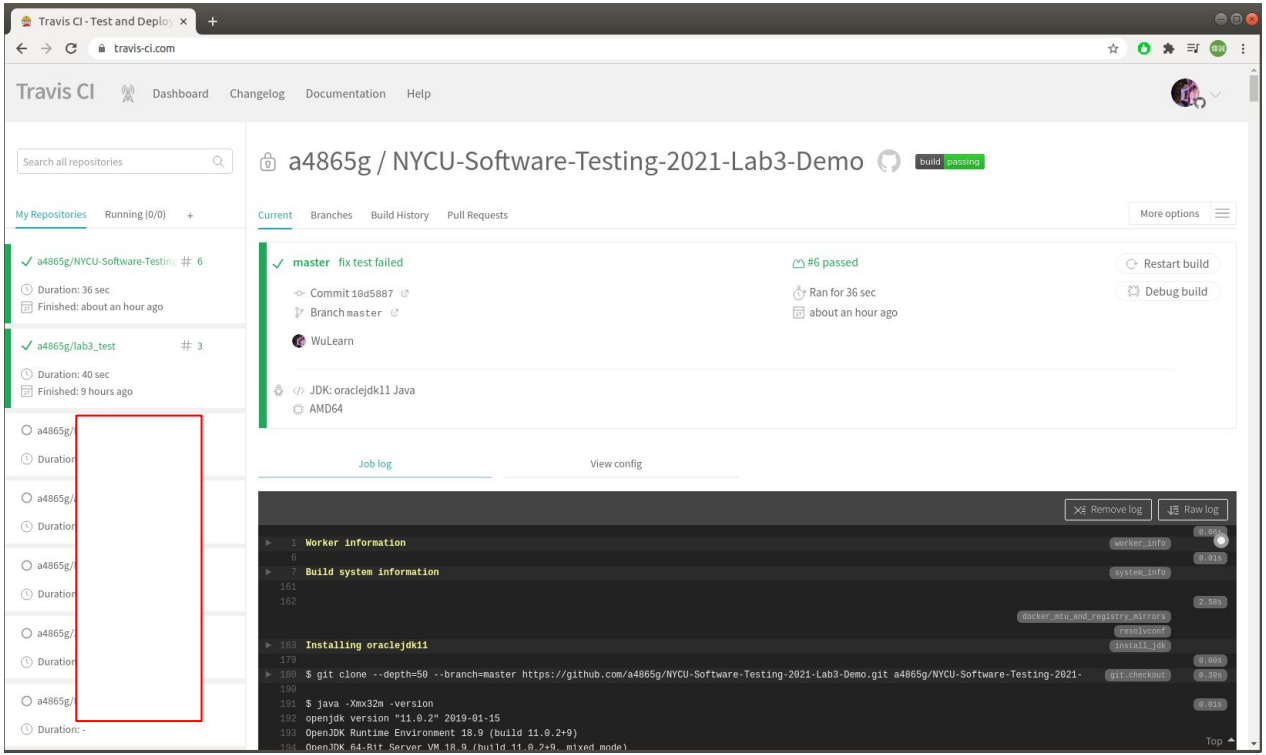
- **Features**
  - Build
  - Test
  - Source code analysis
  - Other related work ... (deployment automation)

- **Advantages**
  - Reduces risk
  - Reduces manual and complex processes
  - A deployable version can be generated at any time
  - Increases system transparency
  - Builds team confidence

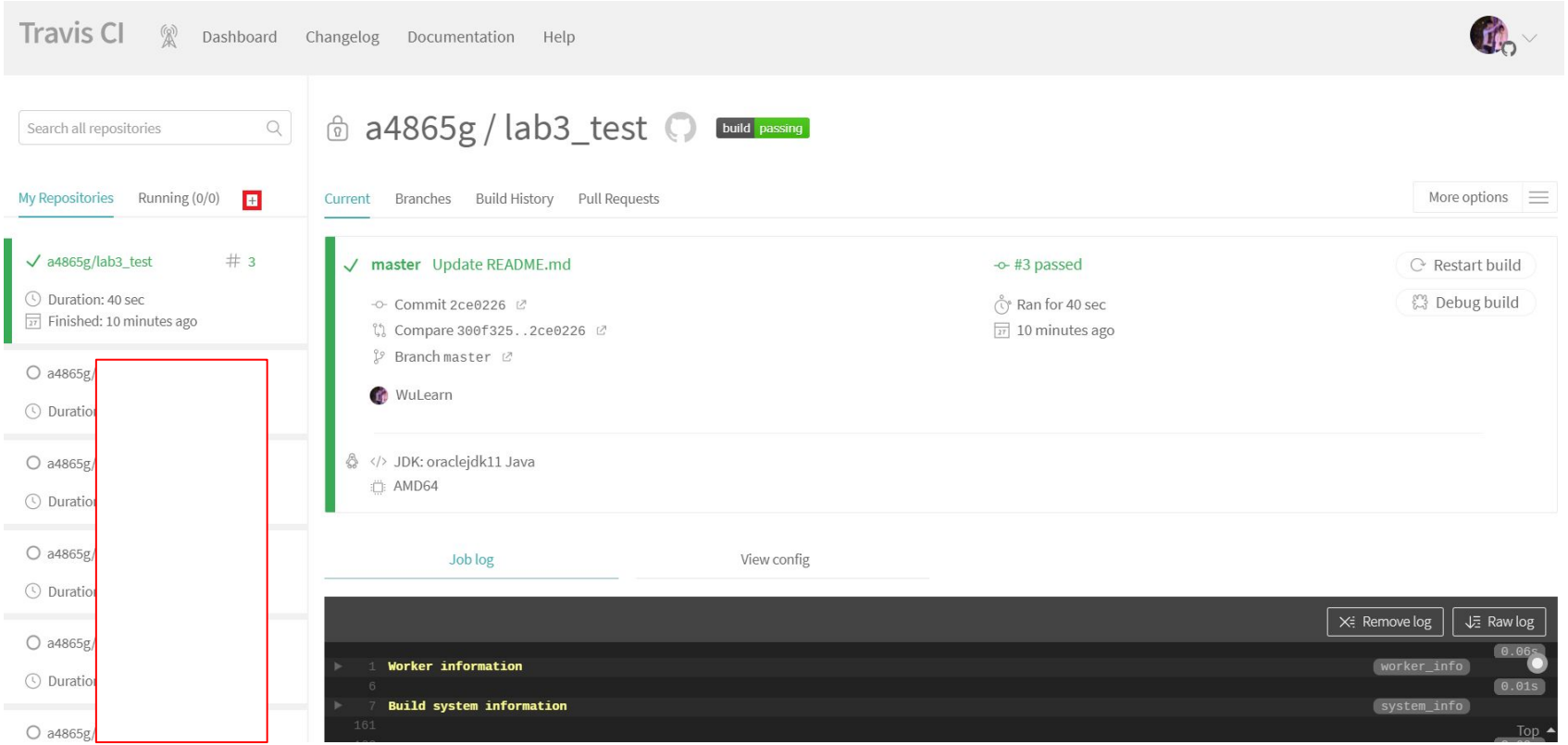


**Jenkins**

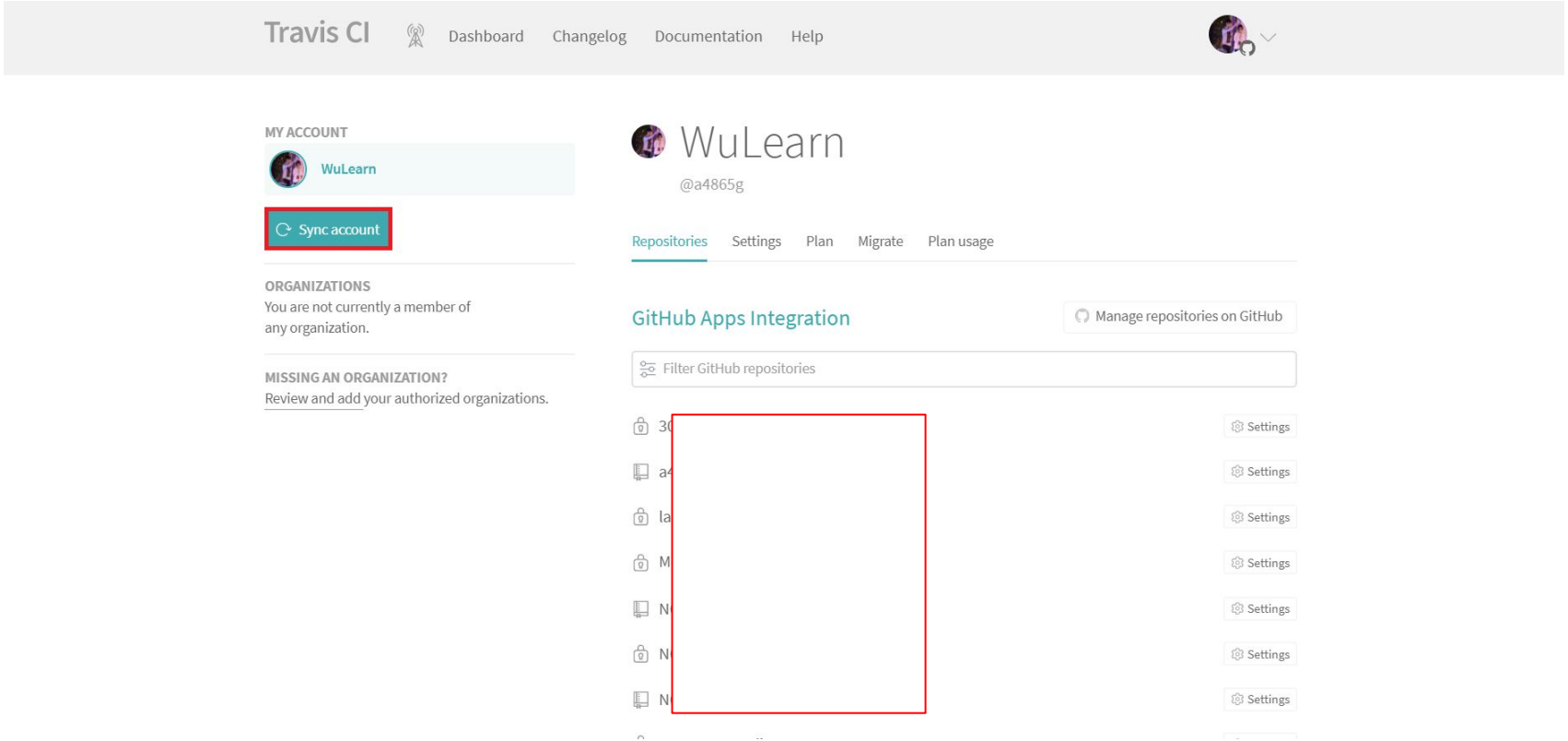
# Travis CI


Travis CI - Test and Deplo
travis-ci.com
Travis CI
Dashboard
Changelog
Documentation
Help
Search all repositories
My Repositories
Running (0/0)
a4865g/NYCU-Software-Testing # 6
Duration: 36 sec
Finished: about an hour ago
a4865g/lab3_test # 3
Duration: 40 sec
Finished: 9 hours ago
a4865g / NYCU-Software-Testing-2021-Lab3-Demo
build passing
Current
Branches
Build History
Pull Requests
More options
master fix test failed
Commit 10d5887
Branch master
WuLearn
#6 passed
Ran for 36 sec
about an hour ago
Restart build
Debug build
JDK: oraclejdk11 Java
AMD64
Job log
View config
Remove log
Raw log
Worker information
Build system information
Installing oraclejdk11
$ git clone --depth=50 --branch=master https://github.com/a4865g/NYCU-Software-Testing-2021-Lab3-Demo.git a4865g/NYCU-Software-Testing-2021-
$ java -Xmx32m -version
openjdk version "11.0.2" 2019-01-15
OpenJDK Runtime Environment 18.9 (build 11.0.2+9)
Top

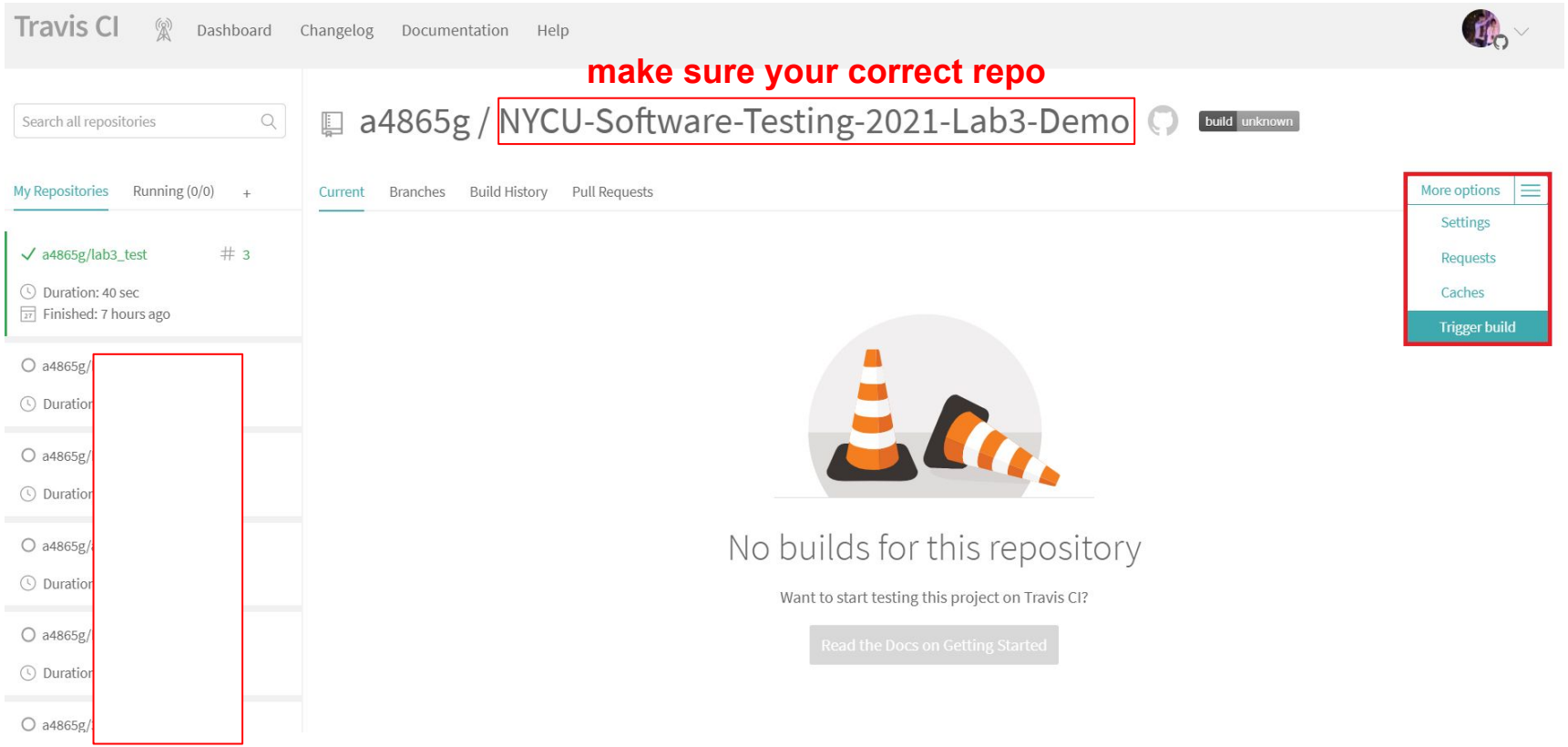
# Travis CI - Homepage



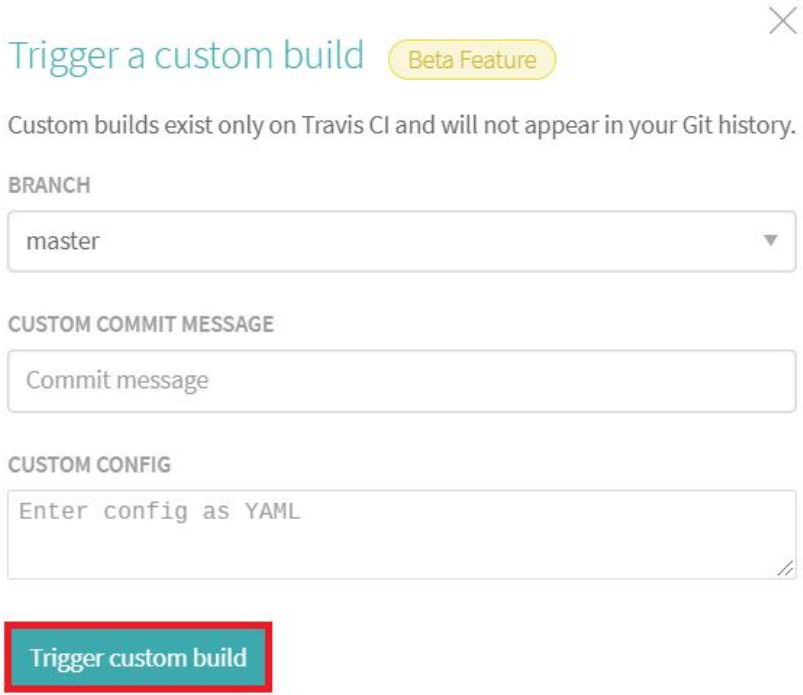
# Travis CI - Sync



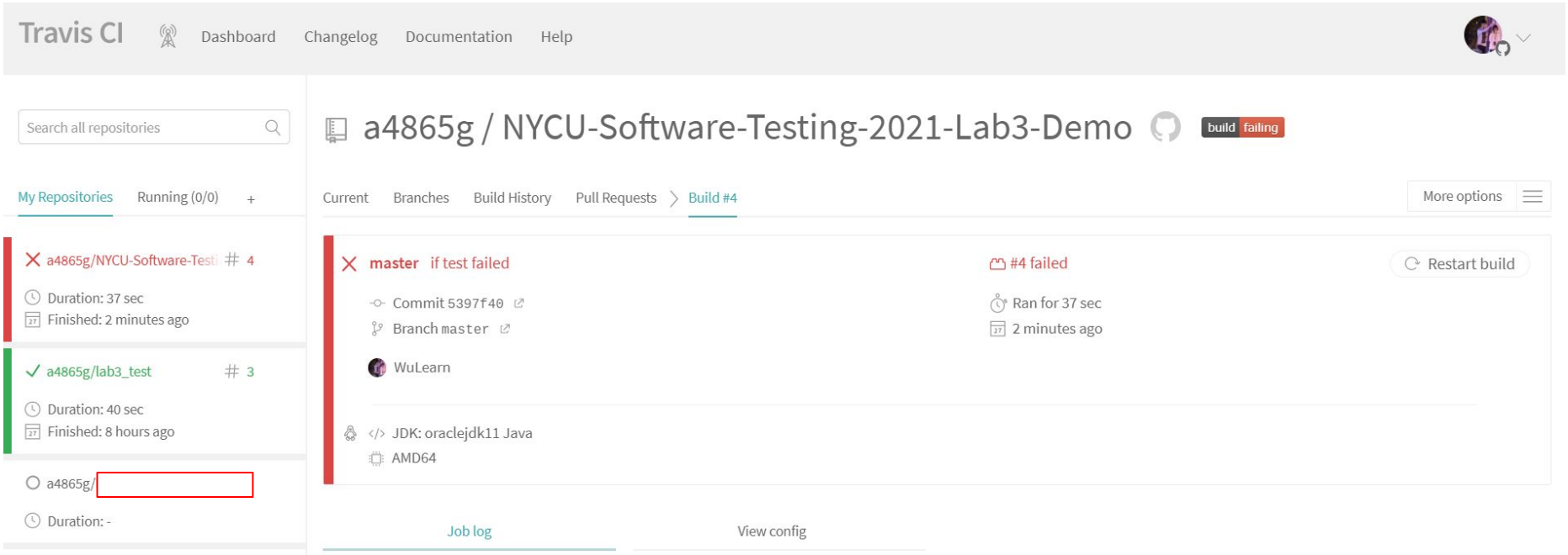
# Travis CI - Build



# Travis CI - Build (Trigger)



# Travis CI - Build (Failed)



# Travis CI - Build (Success)

# Travis CI

- Travis CI
  - https://travis-ci.com/ (For: private / public repo) **(Please use it for this Lab)**
  - https://travis-ci.org/ (For: public repo) (Will be shutting down in several weeks)
- [DevOps] Travis CI - Step/Job/Stage

# Lab

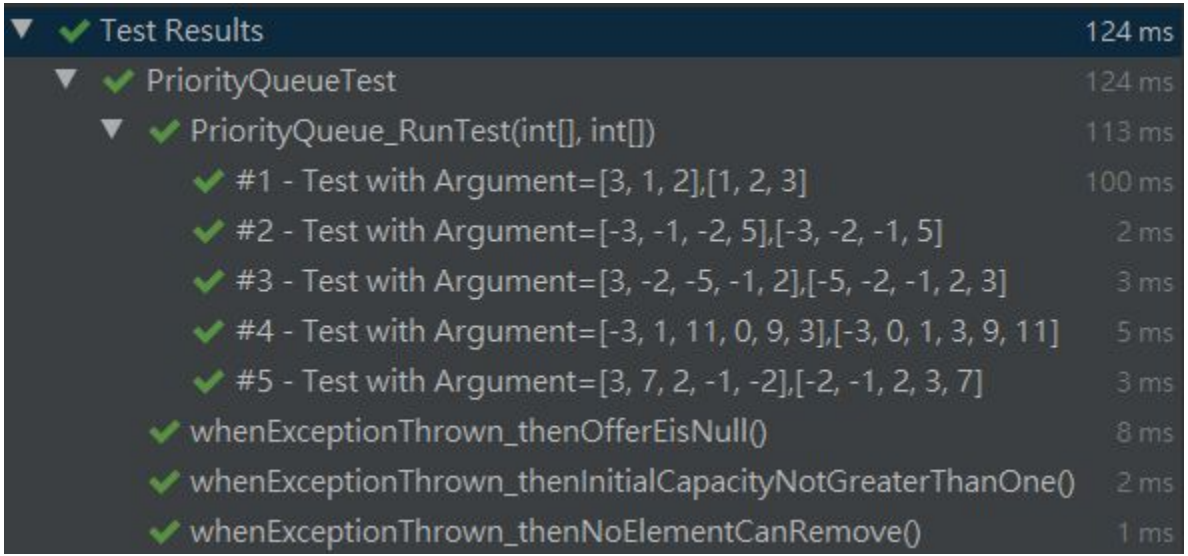
# Lab - Part A

- Test *java.util.PriorityQueue class* with Junit.
  - import java.util.PriorityQueue;
- Parameterization
  - Each testcase : { *random array* } , { *correct array* }.
    - *Random array* means to specify your own input array.
    - *Correct array* means the PriorityQueue polling correct order instead of using toArray().
    - *e.g. arguments(new int[]{5, 4, 2, 3}, new int[]{2, 3, 4, 5});*
  - Your test must have at least 5 unique testcases.
- Exception
  - You have to test at least 3 unique Exceptions that is thrown by PriorityQueue.
  - The same type of exception in different ways is accepted.

# Lab - Part A (Example)

```java
public class PriorityQueueTest {
    static Stream<Arguments> stringIntAndListProvider() {
        //TODO return Stream
    }
    @ParameterizedTest(name="#{index} - Test with Argument={0},{1}")
    @MethodSource("stringIntAndListProvider")
    public void PriorityQueue_RunTest(int[] random_array,int[] correct_array){
        PriorityQueue<Integer> test = new PriorityQueue<Integer>();
        int index=0;
        Integer s;
        int[] result=new int[random_array.length];

        //TODO random_array add to PriorityQueue

        //TODO get PriorityQueue result

        assertArrayEquals(correct_array,result);
    }
    //TODO 3 unique Exceptions
}
```

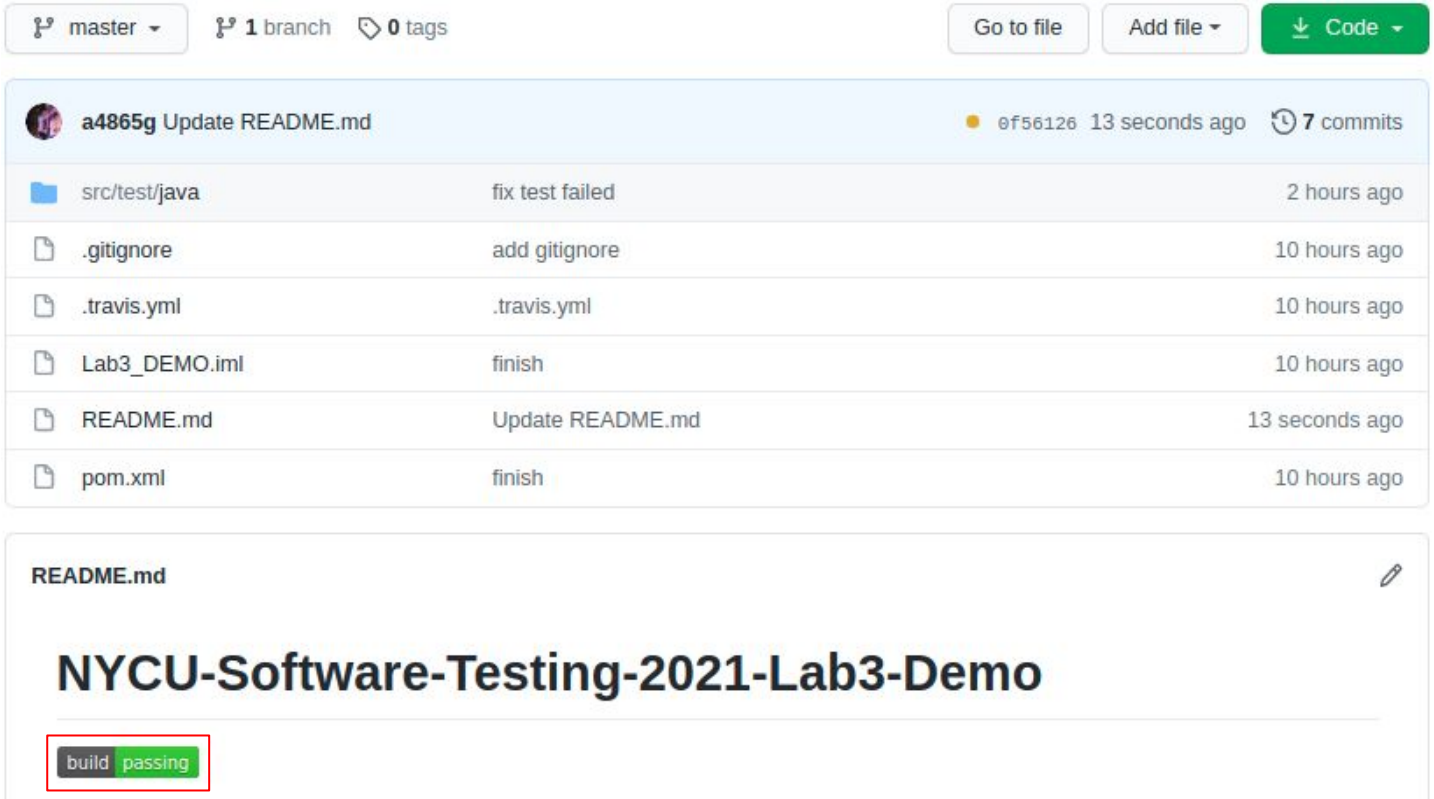
# Lab - Part A (Example)



# Lab - Part B

- Travis-CI
  - Authorize Travis CI with your Github account.
  - Make sure that Travis CI can be built and test automatically.
- Github
  - Deploy your ***Part A (PriorityQueueTest.java)*** on Github in public.
  - The name of repository is your student id.
  - You have to commit twice:
    - Pass all test cases
    - Unable to pass one of the test cases
  - Add *Travis CI status image* and *screenshots* (both pass and fail) to your *README.md*.
- Submit your Github repository link to E3.

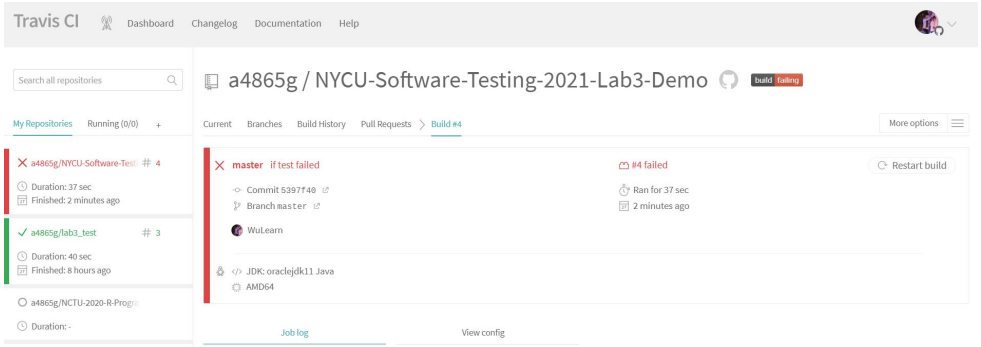
# Lab - Part B (Status Image)



# Lab - Part B (Screenshot)

```
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running PriorityQueueTest
[INFO] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.151 s - in PriorityQueueTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ Lab3_DEMO ---
```

# Lab - Part B (Screenshot)




```
[ERROR] Failures:
[ERROR]   PriorityQueueTest.PriorityQueue_RunTest:37 array contents differ at index [0], expected: <1> but was: <2>
[ERROR]   PriorityQueueTest.PriorityQueue_RunTest:37 array contents differ at index [0], expected: <-3> but was: <-2>
[ERROR]   PriorityQueueTest.PriorityQueue_RunTest:37 array contents differ at index [0], expected: <-5> but was: <-4>
[ERROR]   PriorityQueueTest.PriorityQueue_RunTest:37 array contents differ at index [0], expected: <-3> but was: <-2>
[ERROR]   PriorityQueueTest.PriorityQueue_RunTest:37 array contents differ at index [0], expected: <-2> but was: <-1>
[INFO]
[ERROR] Tests run: 8, Failures: 5, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  12.096 s
[INFO] Finished at: 2021-03-17T17:27:46Z
[INFO] ------------------------------------------------------------------------
```

# Lab - Part B (Hint)

- Create your own **pom.xml** and **.travis.yml**.
  - *Maven: pom.xml*
  - *Travis-CI: .travis.yml*
- In the pom.xml, version of **junit-jupiter** would be **5.5.2** with **JDK 11.**