United Airlines Management Cloud User Guide

Angel Petrov (266489)
Chunhui Liu (273452)
Ziad Akram Bathish (273442)

SEP6 Autumn/2020

Supervisors:

Richard Brooks
Jakob Knop Rasmussen
Laurits Ivar Anesen

Software Technology Engineering
6th semester
18.12.2020

Table of contents:

DevOps Tools

Many DevOps tools were used in this project that helped the team during the workflow process, the tools that were used are:

Github Project Board (Automated Kanban template)

After creating a repository on GitHub, you will be able to see these choices in the toolbox:

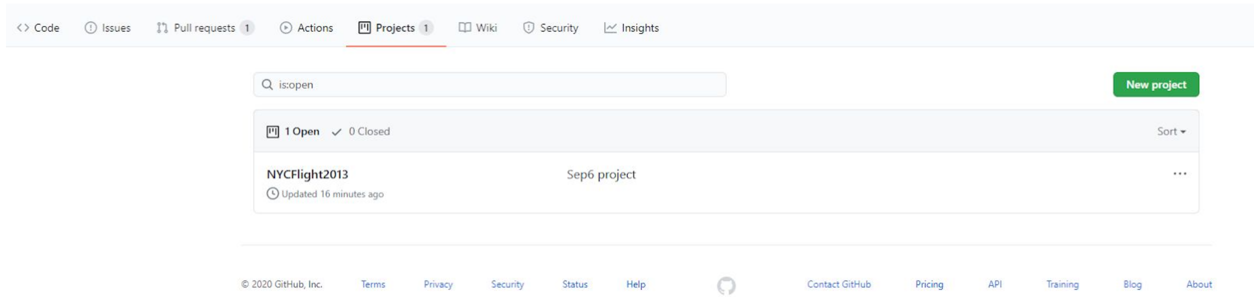
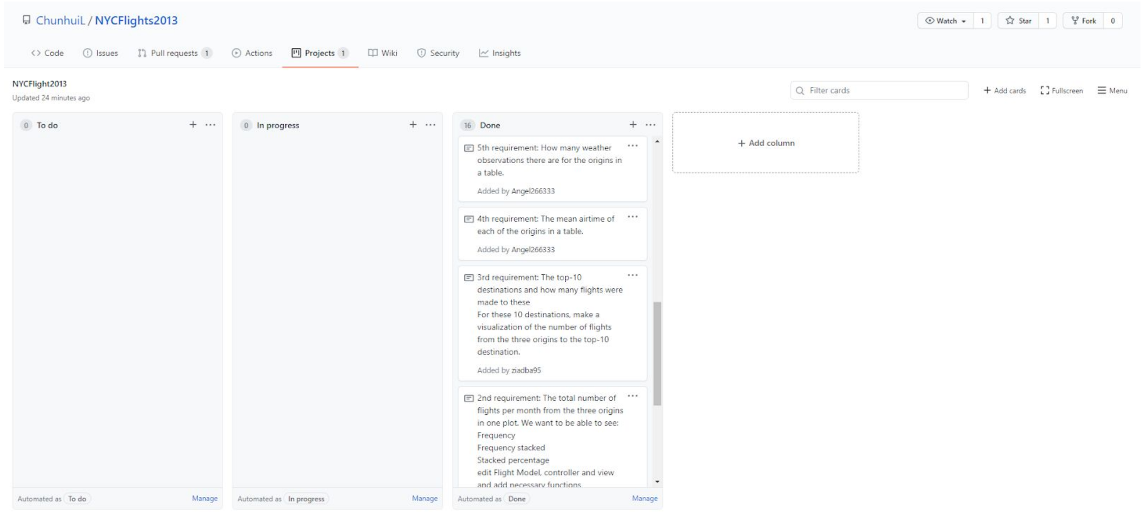


Figure 1: Github Projects Create new Board

After clicking on Projects, you will be able to create a new project by giving it a name and description and you will have a choice to select a template.
In this project, Automated Kanban templates were selected because it makes it easier to move issues and pull requests across the board columns.
After creating the project you can start adding tasks to-do list while working on a task it will be in the "in progress" list when it's done it will be moved to the "Done" list



5th requirement: How many weather observations there are for the origins in a table.
Added by Angel266333

4th requirement: The mean airtime of each of the origins in a table.
Added by Angel266333

3rd requirement: The top-10 destinations and how many flights were made to these
For these 10 destinations, make a visualization of the number of flights from the three origins to the top-10 destination.
Added by ziadba95

2nd requirement: The total number of flights per month from the three origins in one plot. We want to be able to see:
Frequency
Frequency stacked
Stacked percentage
edit Flight Model, controller and view
and add necessary functions

Figure 2: Github Automated Kanban Board

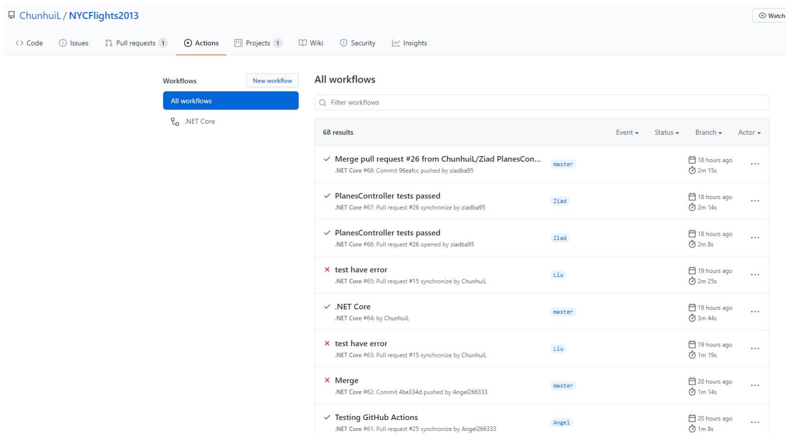
Github Actions

Allows the project team to automate the workflow by providing the possibility to build and test code in GitHub.

It helps with doing automated tests while it has a built-in CI/CD using npm and .net libraries, so every time a team member makes a pull request it will automatically run commands to execute the testing script.

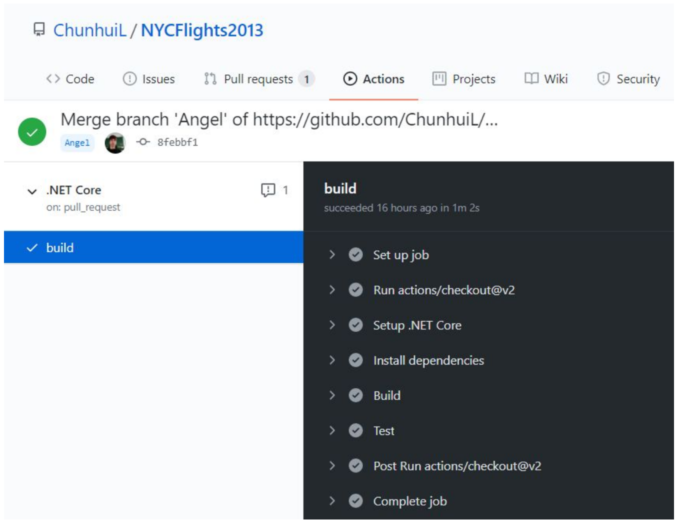
In this project, Actions was used to build the ASP.NET Core application. A file called "dotnet-core.yml" is generated and committed to the main branch. After creating the file, in the Actions section of GitHub, we are able to see that we can now run an Actions workflow.

Here is an example of how all pull requests were tested before merging to the master.



68 results

✓ Merge pull request #26 from ChunhuiL/Ziad PlanesCon... .NET Core #68: Commit 96eafcc pushed by ziadba95	master	18 hours ago	2m 15s
✓ PlanesController tests passed .NET Core #67: Pull request #26 synchronize by ziadba95	Ziad	18 hours ago	2m 14s
✓ PlanesController tests passed .NET Core #66: Pull request #26 opened by ziadba95	Ziad	18 hours ago	2m 8s
✗ test have error .NET Core #65: Pull request #15 synchronize by ChunhuiL	Liu	19 hours ago	2m 25s
✓ .NET Core .NET Core #64: by ChunhuiL	master	19 hours ago	3m 44s
✗ test have error .NET Core #63: Pull request #15 synchronize by ChunhuiL	Liu	19 hours ago	1m 19s
✗ Merge .NET Core #62: Commit 4be334d pushed by Angel266333	master	20 hours ago	1m 14s
✓ Testing GitHub Actions .NET Core #61: Pull request #25 synchronize by Angel266333	Angel	20 hours ago	1m 8s



Merge branch 'Angel' of https://github.com/ChunhuiL/...
Angel 8febbf1

.NET Core 1
on: pull_request
✓ build

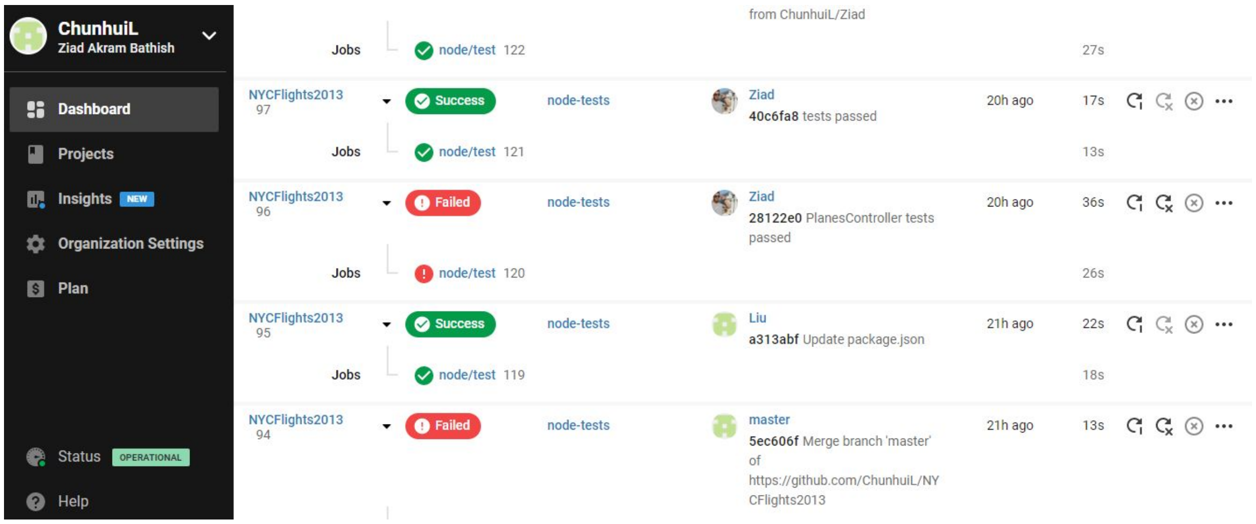
build
succeeded 16 hours ago in 1m 2s

- Set up job
- Run actions/checkout@v2
- Setup .NET Core
- Install dependencies
- Build
- Test
- Post Run actions/checkout@v2
- Complete job

Figure 3: Example of GitHub Actions build passing successfully.

CircleCI

It does the same job as Github actions but it was used just to have extra support and it has a faster building process than Github actions.
After creating the repository, CircleCI was added by authenticating a GitHub account in CircleCI website, then selecting the wanted repository.
Then it will be activated and will test all pull requests created by team members and give a report about the failed tests to have a better idea of what went wrong during the building process.



				from ChunhuiL/Ziad		
Jobs	node/test 122					27s
NYCFlights2013 97	Success	node-tests	Ziad 40c6fa8 tests passed		20h ago	17s
Jobs	node/test 121					13s
NYCFlights2013 96	Failed	node-tests	Ziad 28122e0 PlanesController tests passed		20h ago	36s
Jobs	node/test 120					26s
NYCFlights2013 95	Success	node-tests	Liu a313abf Update package.json		21h ago	22s
Jobs	node/test 119					18s
NYCFlights2013 94	Failed	node-tests	master 5ec606f Merge branch 'master' of https://github.com/ChunhuiL/NYCFlights2013		21h ago	13s

Figure 4: CircleCI Dashboard with building & testing history

Setting up cloud provider, project management, version control and testing:

Project Setup, Scripts and Libraries

The project setup consists of two projects linked together – NYCFlights2013 and TestNYCFlights2013. For NYCFlights2013, a CircleCI configuration file was added to the project setup as well as for GitHub Actions. The extensions used for this project are:

- Angular Basic Project Template version **1.0.4**
- Angular Item Templates version **1.7.1**
- Angular8Core version **1.0**
- ASP.NET Core 2.2 with Angular 7 version **2.0.1**
- ASPNetCore Web Api Starter Template version **2.0**
- Google Cloud Tools for Visual Studio version **2.0.4.0**
- MySQL for Visual Studio version **1.2.9**
- MySQL Connector Net version **8.0.22**

- NPM Task Runner version **1.4.90**
- NodeJS version **14.15.3**

Connection to the database is established in a class called ConnectorDB. The credentials for the database are as follows:

Database name: NYCFlight2013

IP: 34.76.189.231

Username: root / Password: 12345

Port: 3306

The database is facilitated by Google Cloud which acts as a Cloud provider. In Google Cloud, a new project has been created with billing enabled from the given free trial period with a debit card as the development team was unable to utilize their 50 dollars of given credit which was marked as expired in the Overview section for Billing for an unknown reason.

First, we need to create the database instance by expanding the list under the hamburger button on the left side of the main overview page and selecting **SQL** from the dropdown list. Then, we click on **Create Instance** and select MySQL. From there, we provide all necessary details and select the closest region to where we are located - in our case Europe-west1(Belgium)

Moving forward with setting up the project, we need to navigate to the **API Library** section and enable the **Compute Engine API** for our project. This API would allow us to run a virtual machine in the cloud.

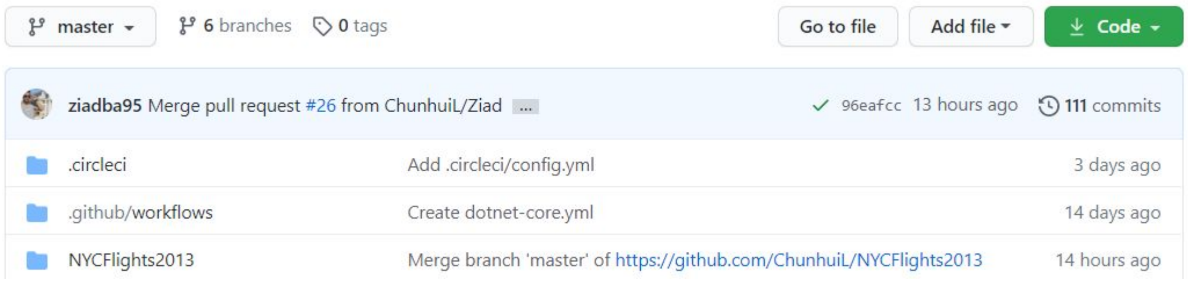
Next thing that we needed to do is set up a static IP address. A static IP address is an external IP address that is reserved for the project. To reserve a static IP address, we must find the section **VPC network** and select the sub-option **External IP addresses**. From there, we would provide our specified name, IP protocol to use, and region, which, in our case, are IPv4 and Europe-west1(Belgium) for closer proximity to the remote server. If we were to look up this address using whatismyipaddress.com, we can see that our settings are in fact correctly applied.

Azure Web Deployment

The application project is deployed to Microsoft Azure. There are no particular settings that need to be adjusted other than having a Microsoft Azure account and credits on it. We have used the free credits that come one-time in order to keep the project up and running for demonstration purposes. We have to right-click on our project and click Publish. From there, select Azure and the application will shortly be deployed.

Version Control

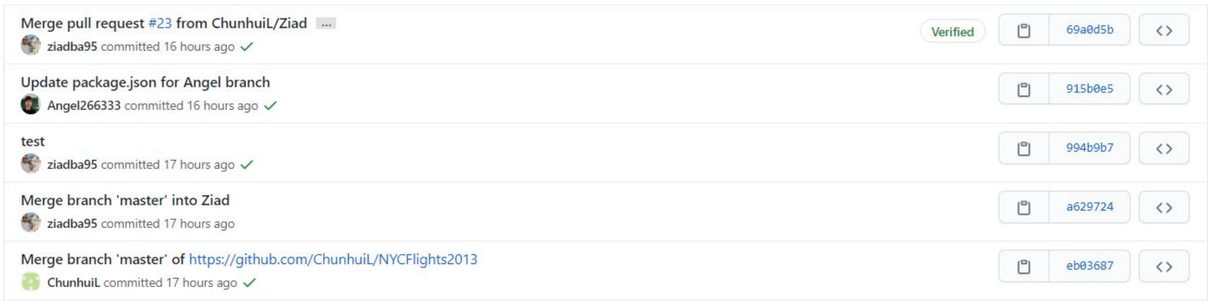
A need for version control was a necessity for the development team due to the scale of the project. GitHub proved to be a reliable version control tool by allowing each individual in the project to work in his own individual branch. An individual is able to make a pull request after pushing from his local branch to his remote branch. Then, he is able to create a pull request to let other collaborators know of the changes that they have made. The contributor and reviewers can then add comments and remarks to the potential changes before the changes are merged into the base branch in this case master. Other major pros of using GitHub include version control for identifying changes that are traceable with version history. If a potential bug was introduced during the span of the project and was merged into the project's main branch the project team is able to look over the commit history of the affected file and revert to a working version. Version history also provides the possibility to see a snapshot of the entire selected branch and the commits done to it at the point that the snapshot was taken.



ziadba95 Merge pull request #26 from ChunhuiL/Ziad ... ✓ 96eafcc 13 hours ago 111 commits

.circleci	Add .circleci/config.yml	3 days ago
.github/workflows	Create dotnet-core.yml	14 days ago
NYCFlights2013	Merge branch 'master' of https://github.com/ChunhuiL/NYCFlights2013	14 hours ago

Figure 5: Project team's repository in master branch.


Merge pull request #23 from ChunhuiL/Ziad …
ziadba95 committed 16 hours ago ✓
Verified 69a0d5b

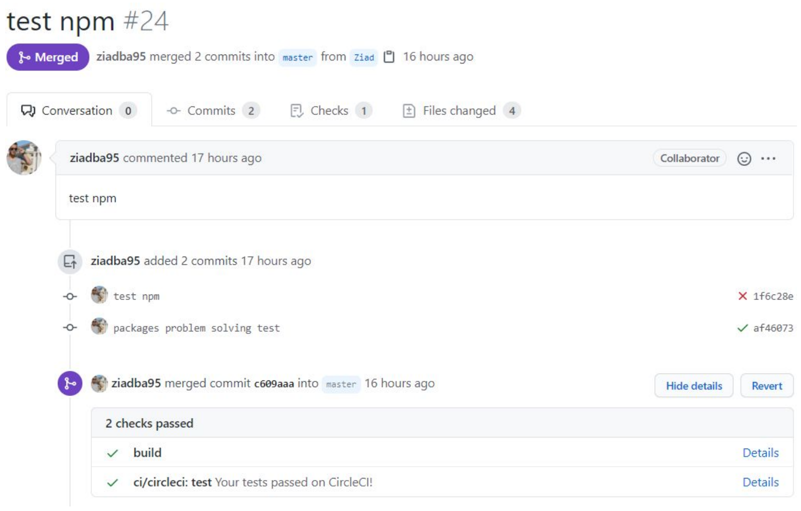
Update package.json for Angel branch
Angel266333 committed 16 hours ago ✓
915b0e5

test
ziadba95 committed 17 hours ago ✓
994b9b7

Merge branch 'master' into Ziad
ziadba95 committed 17 hours ago
a629724

Merge branch 'master' of https://github.com/ChunhuiL/NYCFlights2013
ChunhuiL committed 17 hours ago ✓
eb03687

Figure 6: Selected section of commit history. Notice that pull requests have a "Verified" attribute to make contributors aware that the commit has been approved.


test npm #24
Merged ziadba95 merged 2 commits into master from Ziad 16 hours ago

ziadba95 commented 17 hours ago
test npm

ziadba95 added 2 commits 17 hours ago
test npm ✗ 1f6c28e
packages problem solving test ✓ af46073

ziadba95 merged commit c609aaa into master 16 hours ago
2 checks passed
✓ build Details
✓ ci/circleci: test Your tests passed on CircleCI! Details

Figure 7: Example Pull Request being approved and merged into master.

Automating parts of the workflow through of CI/CD:

Continuous Integration: Github actions and CircleCi were used to automate builds and testing on the completed and committed code.
Continuous Delivery: while there is no end-user there was no need for CD. in case there is an end-user the process will be to develop, build, test and deploy same as CI just with one step further.