Copycloud

Development operations

Introduction

Dev operations is one of the most important parts of software development, yet it is often overlooked. It is crucial for many things, including task distribution and communication between teams, faster development, as well as an easier way to deliver the end product to the customers.

In order to simplify the process and make it easier for end users to get the latest updates, Copycloud uses Azure's DevOps cloud services. Those include a collaborative sprint distribution dashboard, development and deployment pipelines, as well as automated testing and code quality assurance.

In the following chapters follows a breakdown of how each of the above mentioned components works within the system and why/how this benefits both development and end product delivery.

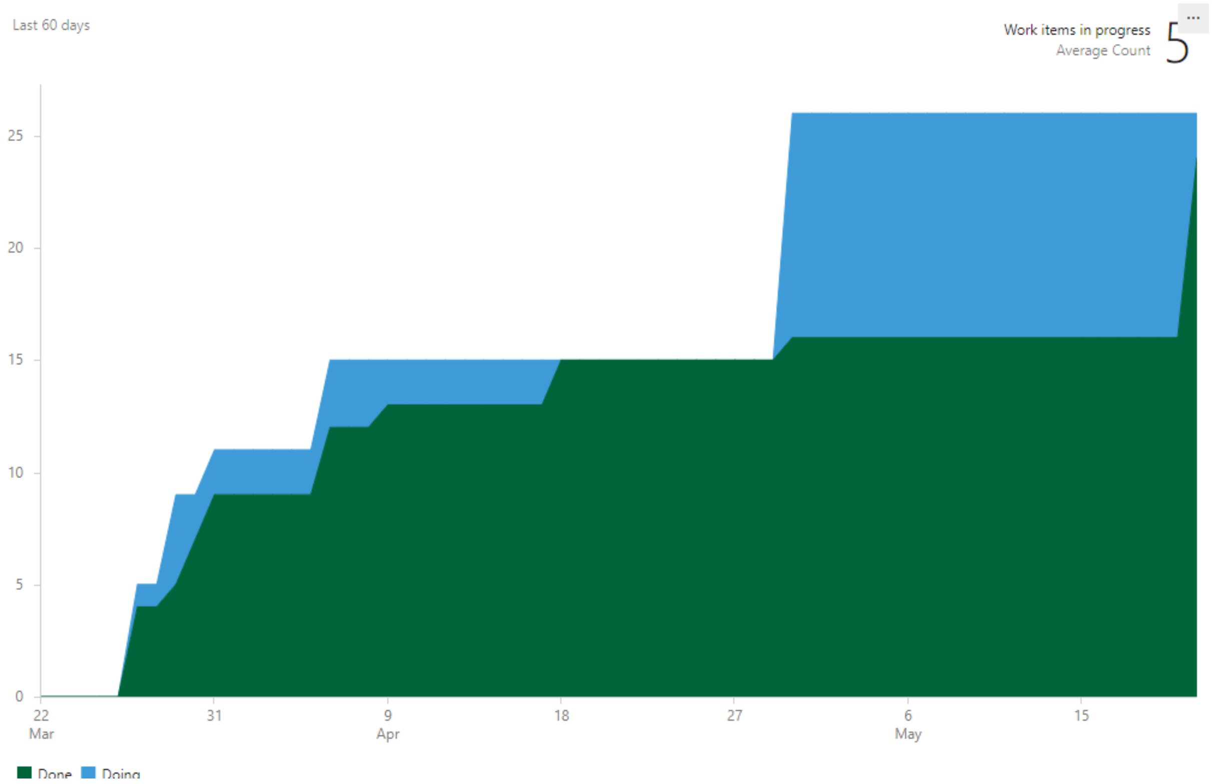
Furthermore, this document concludes with an overview of why Azure for DevOps & the benefits of using a cloud service for monitoring/analytics.

Task Management

In order to make a clear separation of what needs to be done each sprint and describe each task and its importance, Azure boards are used for sprints, as well as analytics.

To demonstrate the importance of such tools, a diagram of the last 60 days of work can be seen below:

Breakdown of the period 22nd of march until 21st of May

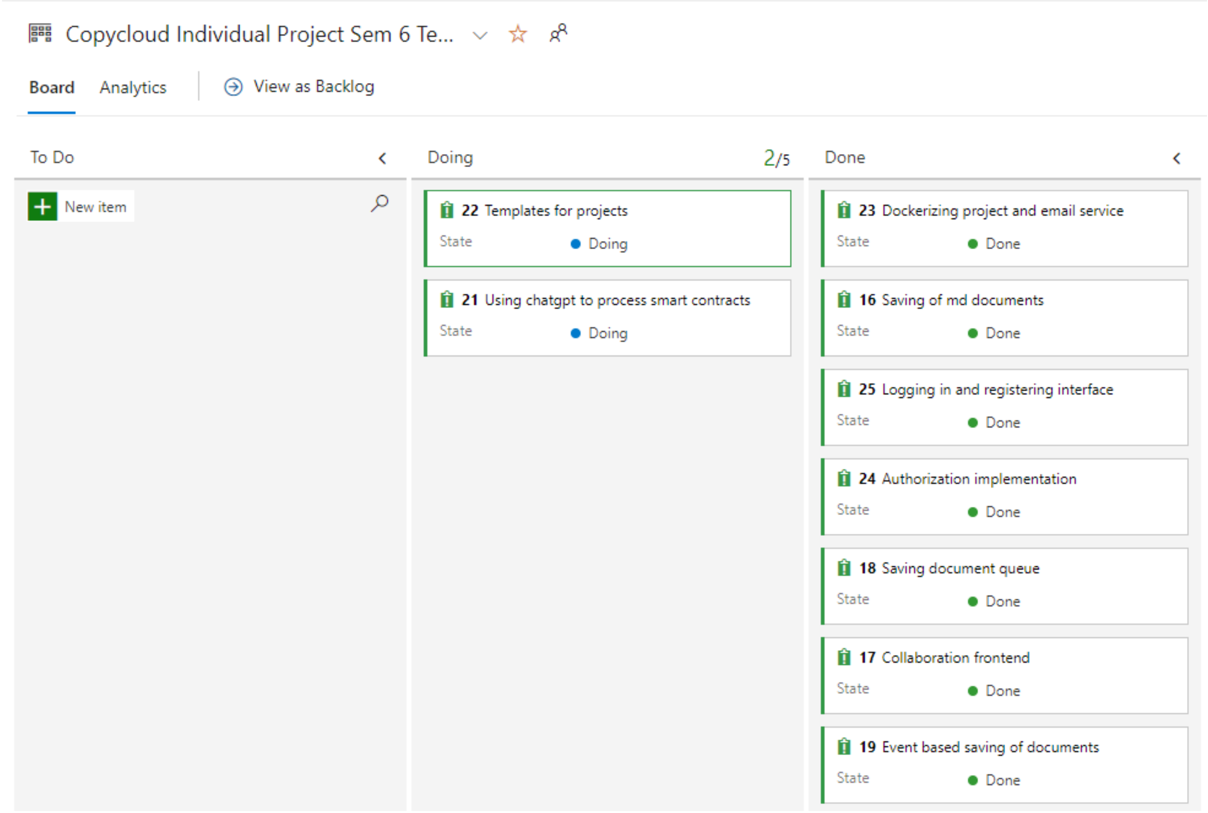


This information is useful as it shows that in the beginning, the graphs of doing and done are very similar in shape, but as the sprint approaches the end, there are many more tasks that are still a work of progress.

This information can be used to draw a conclusion that tasks should be split up into more groups or distributed more evenly over the sprints. This can also suggest that at the end of the sprint, time management becomes increasingly more difficult, which ends up having an impact of how many tasks are actually completed.

To visually represent each task and what state it is currently in, the most common way is to use a simple “To do - doing - done” board. There are many different tools, such as Trello/Github or Jira, but for this project, Azure boards were used. The reason behind this is that it allows to easily integrate boards with the actual codebase and see more insight/analytics about tasks or different sprints.

Example Azure board with 2 tasks in doing and 7 done



To Do

Doing 2/5

22 Templates for projects
State ● Doing

21 Using chatgpt to process smart contracts
State ● Doing

Done

23 Dockerizing project and email service
State ● Done

16 Saving of md documents
State ● Done

25 Logging in and registering interface
State ● Done

24 Authorization implementation
State ● Done

18 Saving document queue
State ● Done

17 Collaboration frontend
State ● Done

19 Event based saving of documents
State ● Done

Automated builds and deployment

The world of software moves extremely fast and updates are needed all the time. In order to keep up with increasing demands, an automation pipeline (or multiple) is usually needed for a development team to be able to deliver fast and reliably.

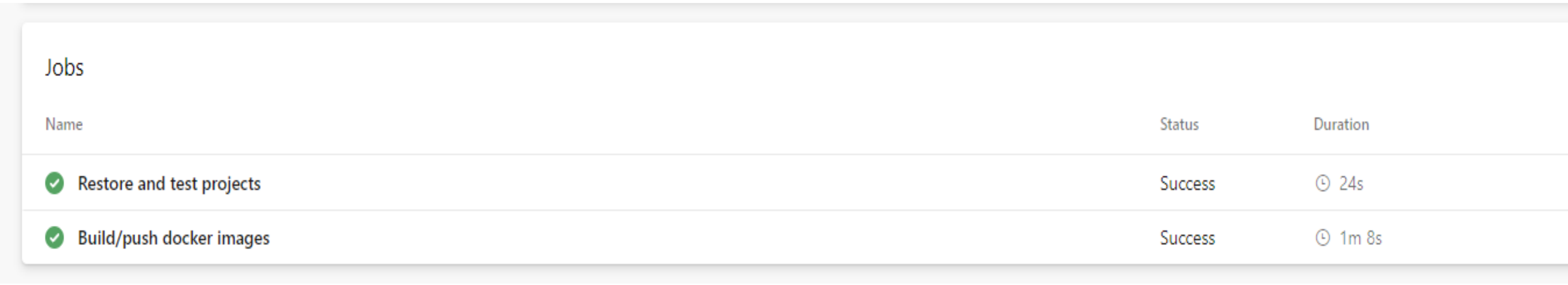
To achieve this, Copycloud uses Azure pipelines for automatically building the application, testing its codebase, assuring code quality with an external tool and finally - creating a virtual image (virtual computer) and deploying it for end customers.

In order to have a clear separation of what state the pipeline is at, there are 4 different stages that it goes through:

1) Automated application build + automated tests
2) Code quality assurance
3) Creating virtual images & publishing them to a repository
4) Automatically deploying

Currently, Copycloud covers ¾ of the above mentioned steps, where the last one remaining being automatic deployment.

To see what is going on visually, Azure pipelines provides a great overview of what state the pipeline is currently in, as well as anything being logged directly from the application:



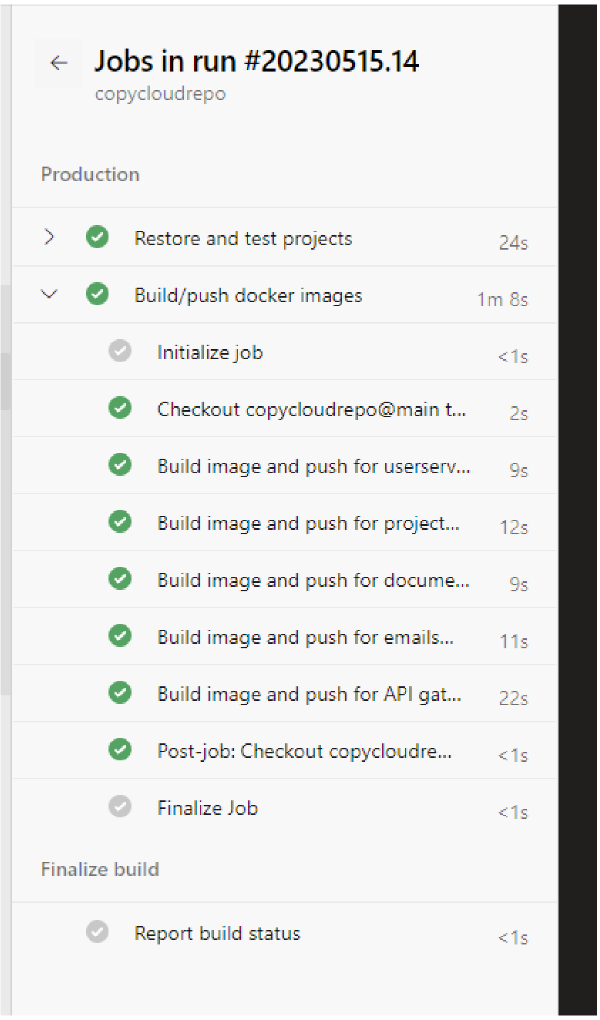
Jobs

Name	Status	Duration
✓ Restore and test projects	Success	24s
✓ Build/push docker images	Success	1m 8s

It is clear and shows how long it takes for each process, which can also derive clear issues such as memory leaks/problems that case the application to take very long to test or build.

An even more clear breakdown of each of the processes can be seen when either of the jobs is opened. This reveals what each step of the job goes through. For reference, Azure cancels the entire job if any of the substeps fails, ensuring that the software is not deployed in case of any bugs/issues.

Build/Image creation & Build/Test pipelines respectfully



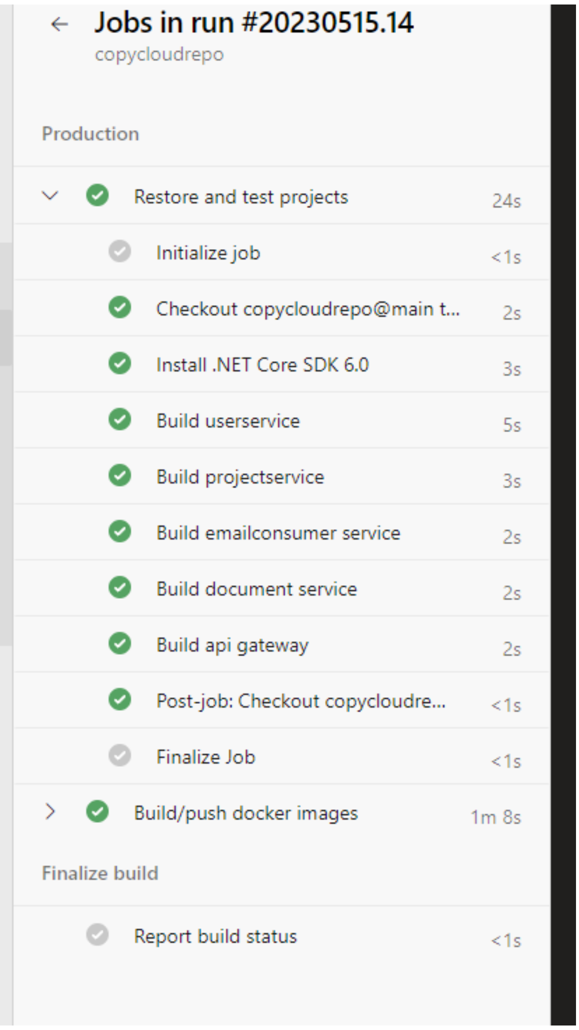
Jobs in run #20230515.14
copycloudrepo

Production

Restore and test projects 24s
Build/push docker images 1m 8s
Initialize job <1s
Checkout copycloudrepo@main t... 2s
Build image and push for userserv... 9s
Build image and push for project... 12s
Build image and push for docume... 9s
Build image and push for emails... 11s
Build image and push for API gat... 22s
Post-job: Checkout copycloudre... <1s
Finalize Job <1s

Finalize build

Report build status <1s



Jobs in run #20230515.14
copycloudrepo

Production

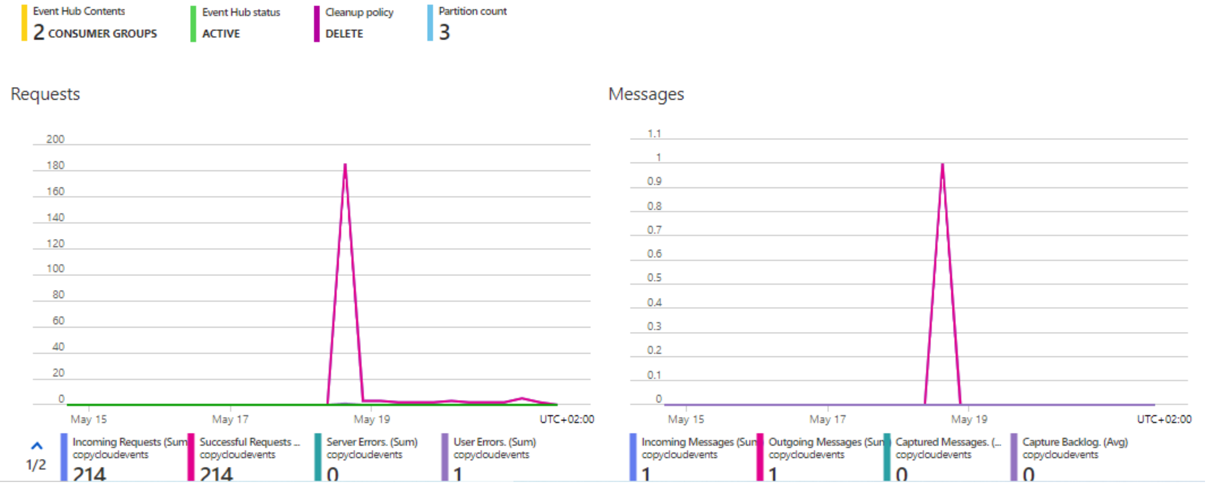
Restore and test projects 24s
Initialize job <1s
Checkout copycloudrepo@main t... 2s
Install .NET Core SDK 6.0 3s
Build userservice 5s
Build projectservice 3s
Build emailconsumer service 2s
Build document service 2s
Build api gateway 2s
Post-job: Checkout copycloudre... <1s
Finalize Job <1s
Build/push docker images 1m 8s

Finalize build

Report build status <1s

One of the biggest benefits of using Azure for dev operations is the in-built monitoring of each cloud service that is delegated to Azure. This is extremely useful, as things such as event-driven architecture, file storages, email systems or even deployment itself can be closely tracked with very insightful analytics for each component.

Event hub tracking



Event Hub Contents 2 CONSUMER GROUPS | Event Hub status ACTIVE | Cleanup policy DELETE | Partition count 3

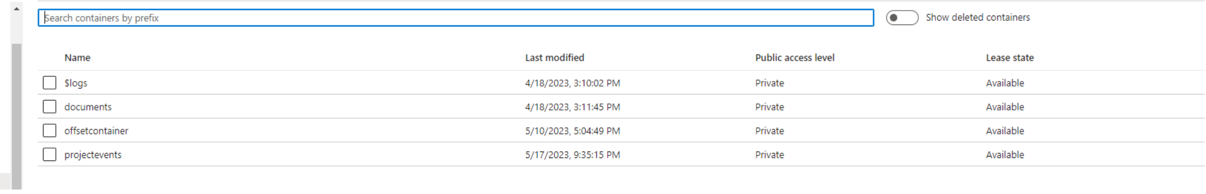
Requests

Incoming Requests (Sum copycloudevents 214 | Successful Requests ... copycloudevents 214 | Server Errors. (Sum) copycloudevents 0 | User Errors. (Sum) copycloudevents 1

Messages

Incoming Messages (Sum copycloudevents 1 | Outgoing Messages (Sum copycloudevents 1 | Captured Messages. (... copycloudevents 0 | Capture Backlog. (Avg) copycloudevents 0

Storage containers

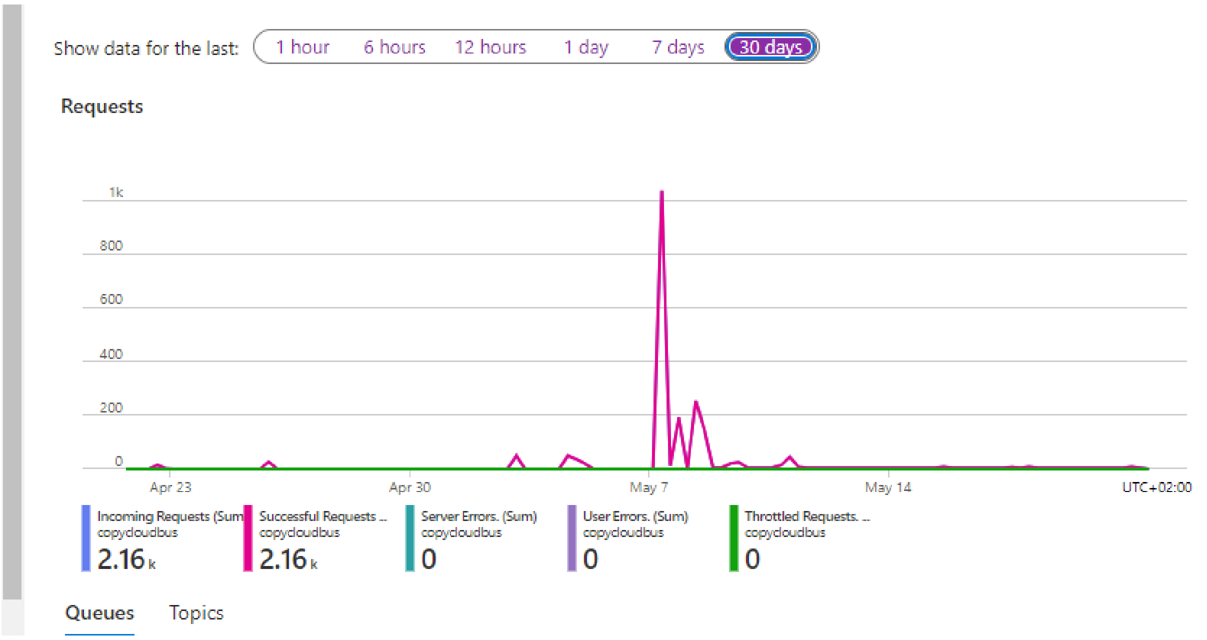


Search containers by prefix

○ Show deleted containers

	Name	Last modified	Public access level	Lease state
□	$logs	4/18/2023, 3:10:02 PM	Private	Available
□	documents	4/18/2023, 3:11:45 PM	Private	Available
□	offsetcontainer	5/10/2023, 5:04:49 PM	Private	Available
□	projectevents	5/17/2023, 9:35:15 PM	Private	Available

Message queues



Show data for the last: 1 hour | 6 hours | 12 hours | 1 day | 7 days | [✓ 30 days]

Requests

Incoming Requests (Sum copycloudbus 2.16 k | Successful Requests ... copycloudbus 2.16 k | Server Errors. (Sum) copycloudbus 0 | User Errors. (Sum) copycloudbus 0 | Throttled Requests. ... copycloudbus 0

In conclusion, using a delegated 3rd party service for dev operations (building, testing, deploying, tracking analytics) is usually the most recommended path for any small to midsize software company. This changes if the requirements involve using home-brewn tools for testing, deployment and architecture.

For Copycloud, Azure DevOps provides all of the tools necessary and does this at an affordable price. Furthermore, as a solo/small development team, Azure lets you focus time on the product and marketing, while all of the technicalities are taken care of seamlessly.