Google I/O Extended

Deploy and Host your app with Cloud Build and Cloud Run

Google Cloud

Goodie Bag

Relax

Slides, code, and more are available at

the1mattkaufman.com

And yes, this slide will be shown again at the end just in case you're reading this now instead of taking out your phone.

About Me

Matt Kaufman

Founder and CEO, MK Partners
(a Salesforce Consultancy and App Publisher)

12+ year GDG Organizer

Started coding cluelessly at age 4
Could not keep up in college
Coding professionally since 2007

Husband and Father of three

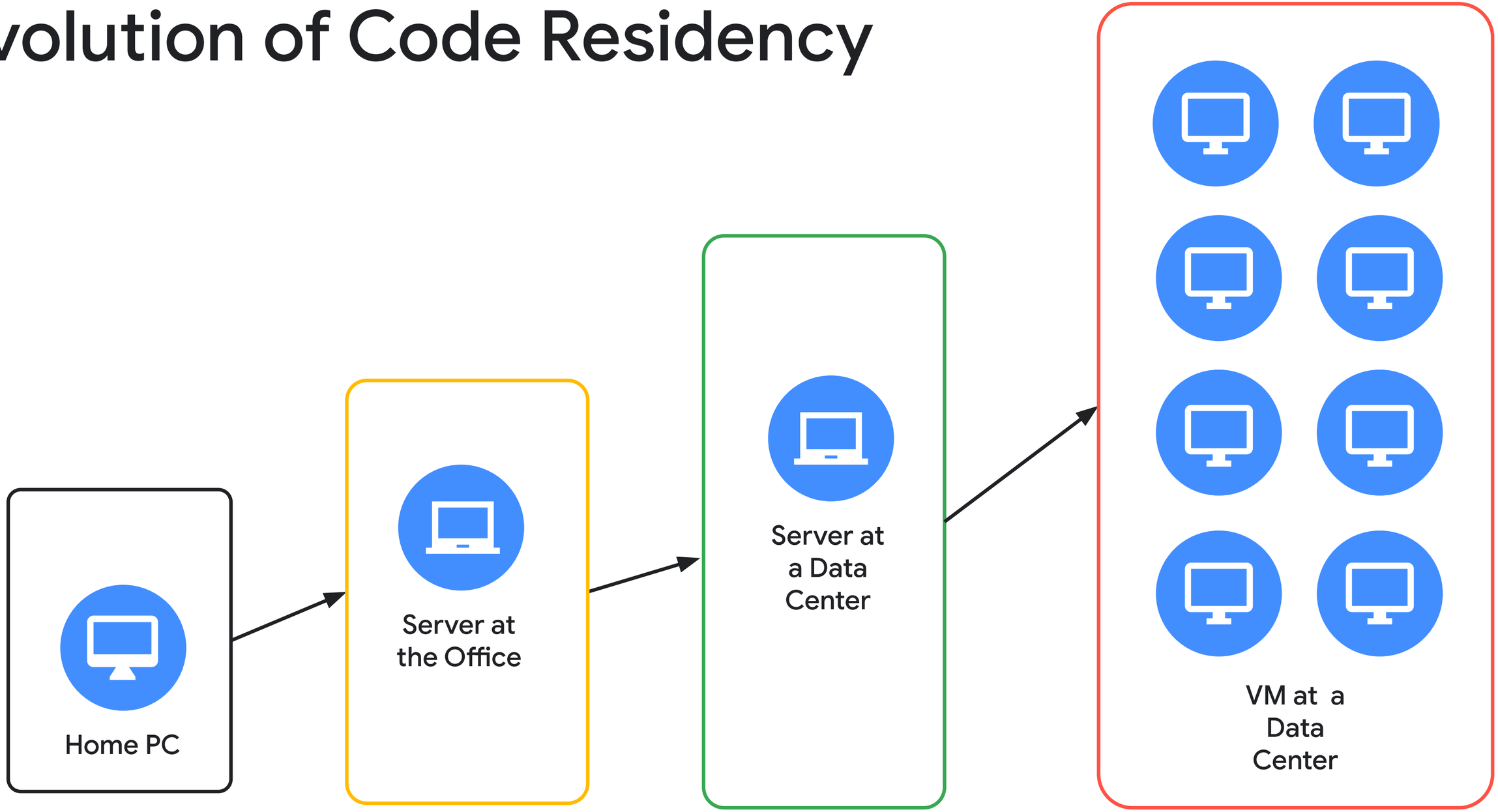
Love puzzles, fixing stuff, and sci-fi

@the1mattkaufman

A Brief History

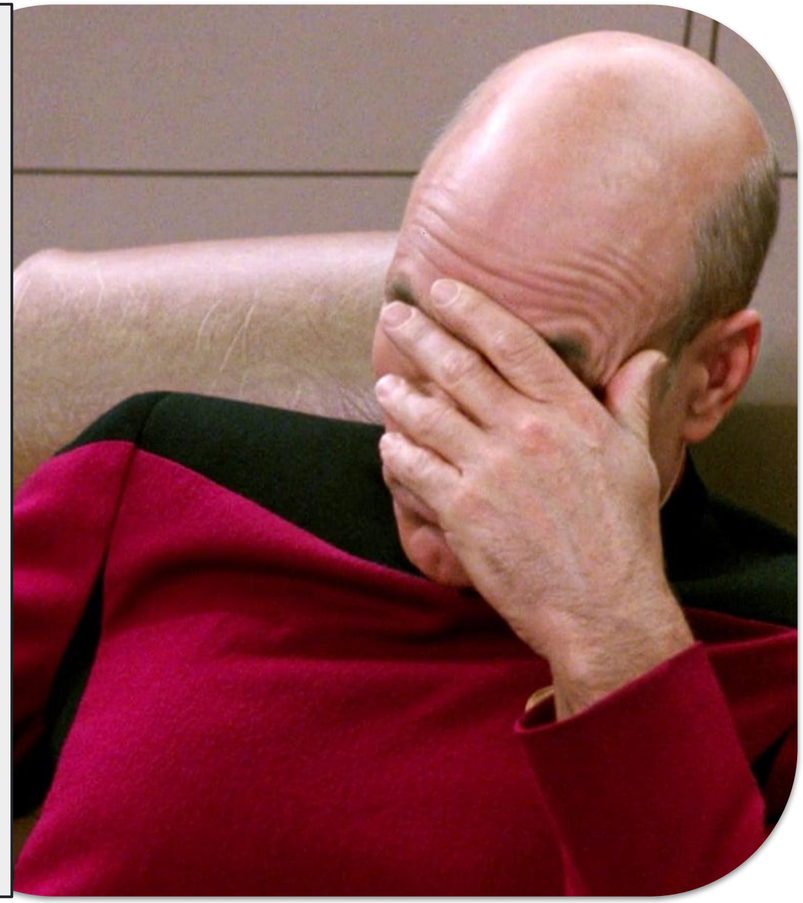
In the beginning

Evolution of Code Residency



0%

Percent of coders that want to volunteer as server engineers in their spare time.



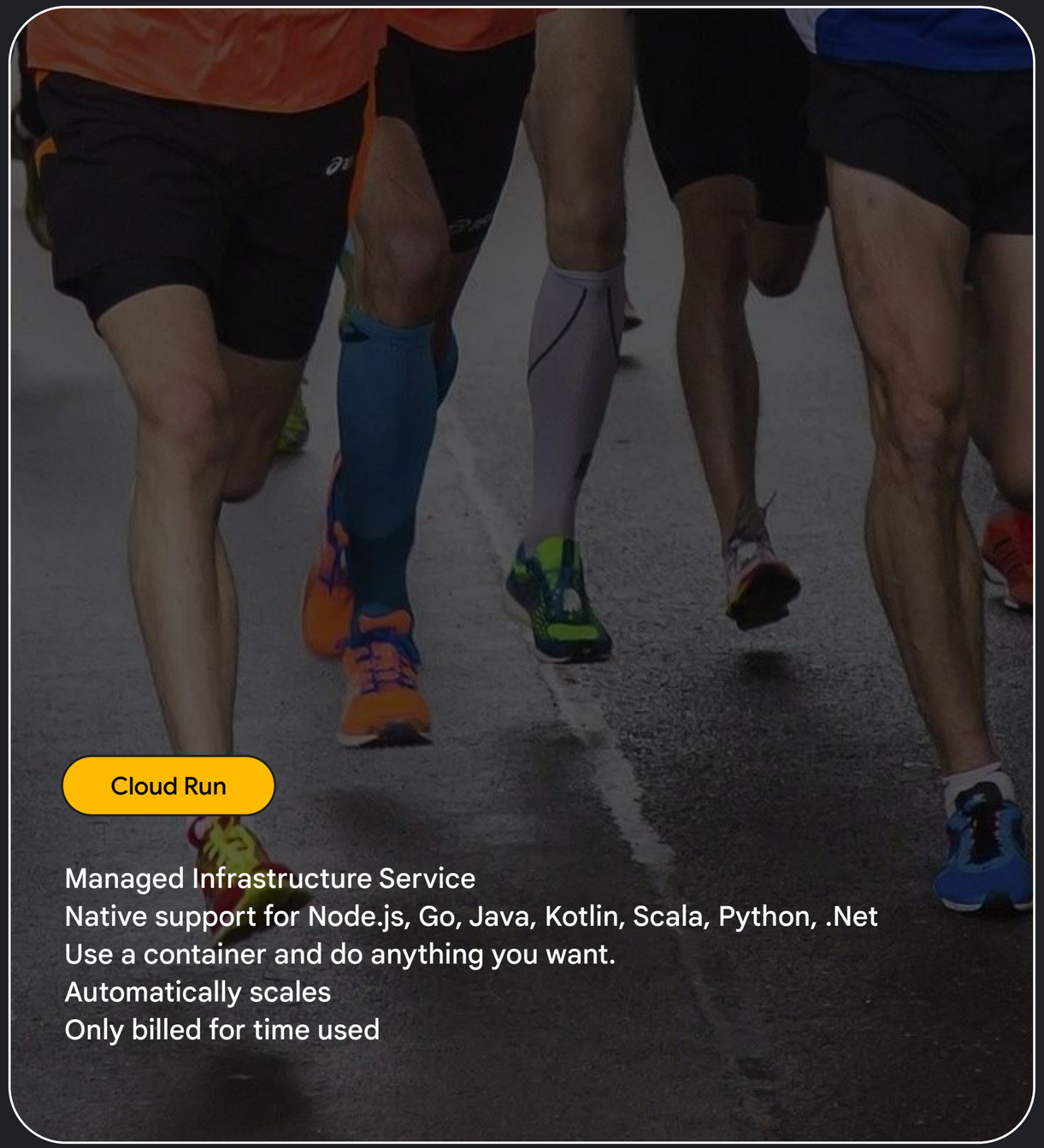
This is the way

Serverless

Why Serverless

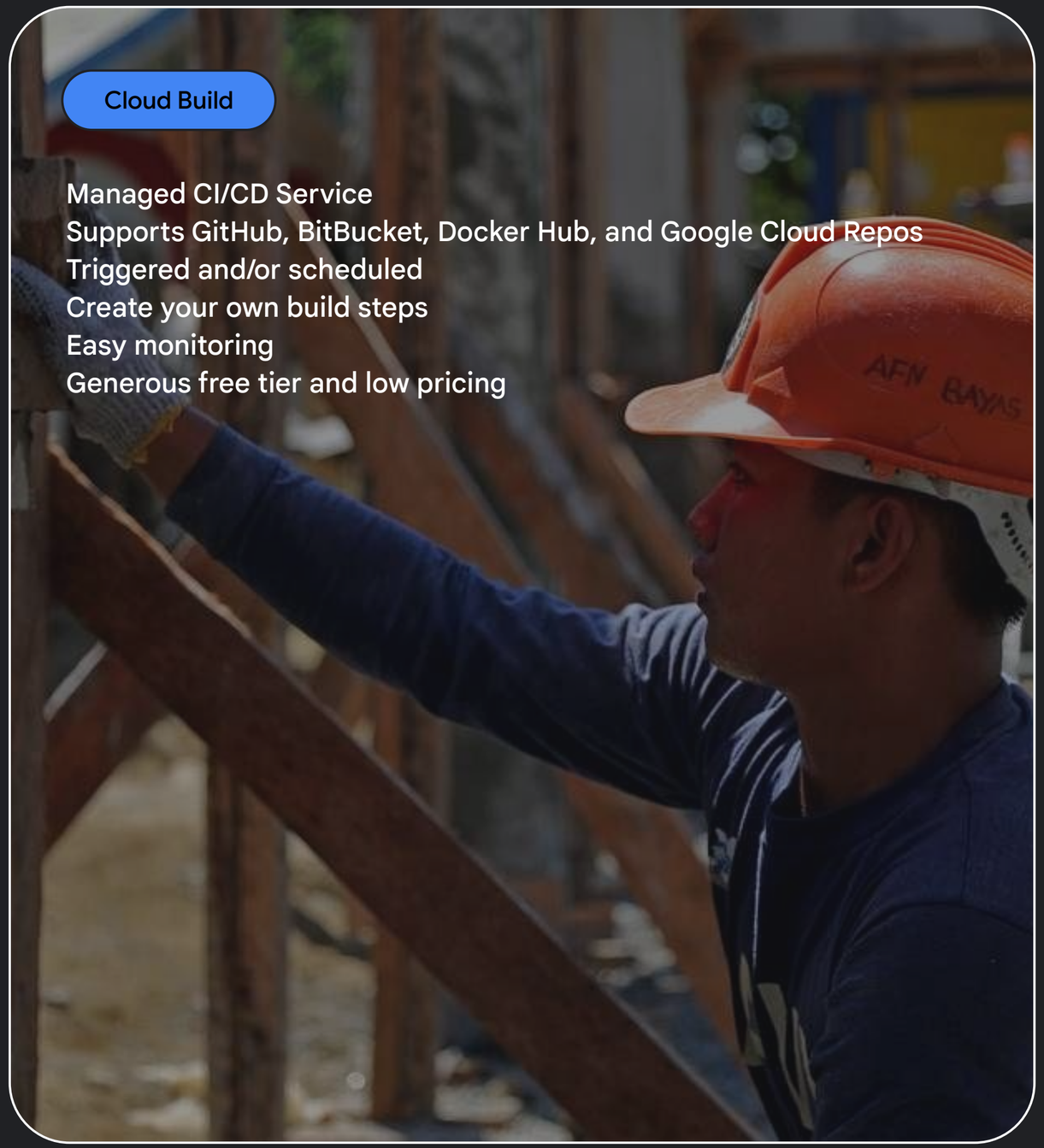
Serverless provides the fastest path to cloud native applications, bringing speed and scalability without worrying about managing infrastructure

Source: cloud.google.com



Cloud Run

Managed Infrastructure Service
Native support for Node.js, Go, Java, Kotlin, Scala, Python, .Net
Use a container and do anything you want.
Automatically scales
Only billed for time used



Cloud Build

Managed CI/CD Service
Supports GitHub, BitBucket, Docker Hub, and Google Cloud Repos
Triggered and/or scheduled
Create your own build steps
Easy monitoring
Generous free tier and low pricing

Live Demo

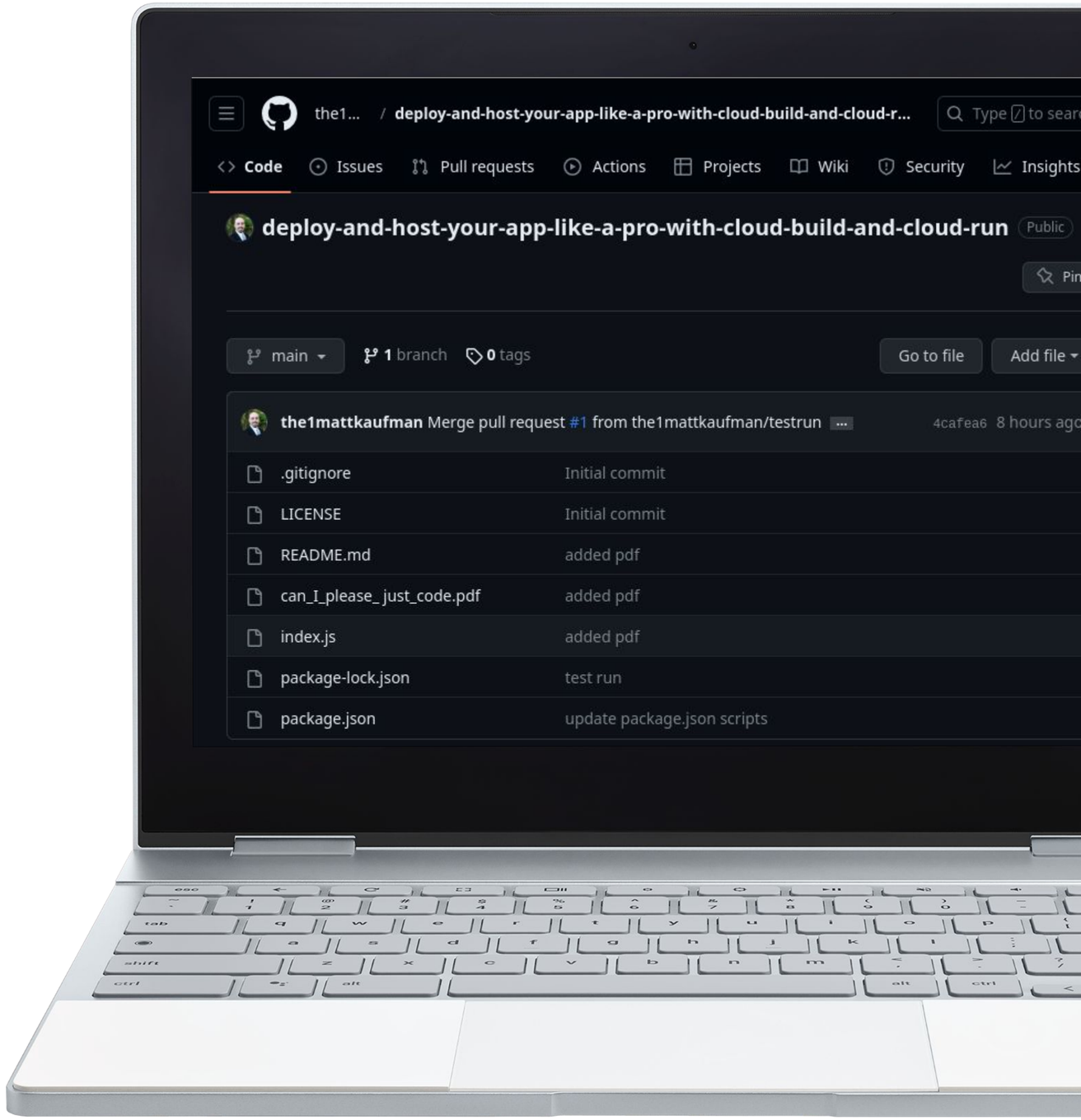
Let’s deploy a simple app

index.js

```
app.get('/', (req, res) => {
  const response = {};
  response.numerator = parseFloat(req.query.numerator,10);
  response.denominator = parseFloat(req.query.denominator,10);
  response.result = response.numerator / response.denominator;
  if ( !req.query.numerator || !req.query.denominator ) {
    throw new Error('Invalid numerator or denominator');
  }
  res.json(response);
})
```

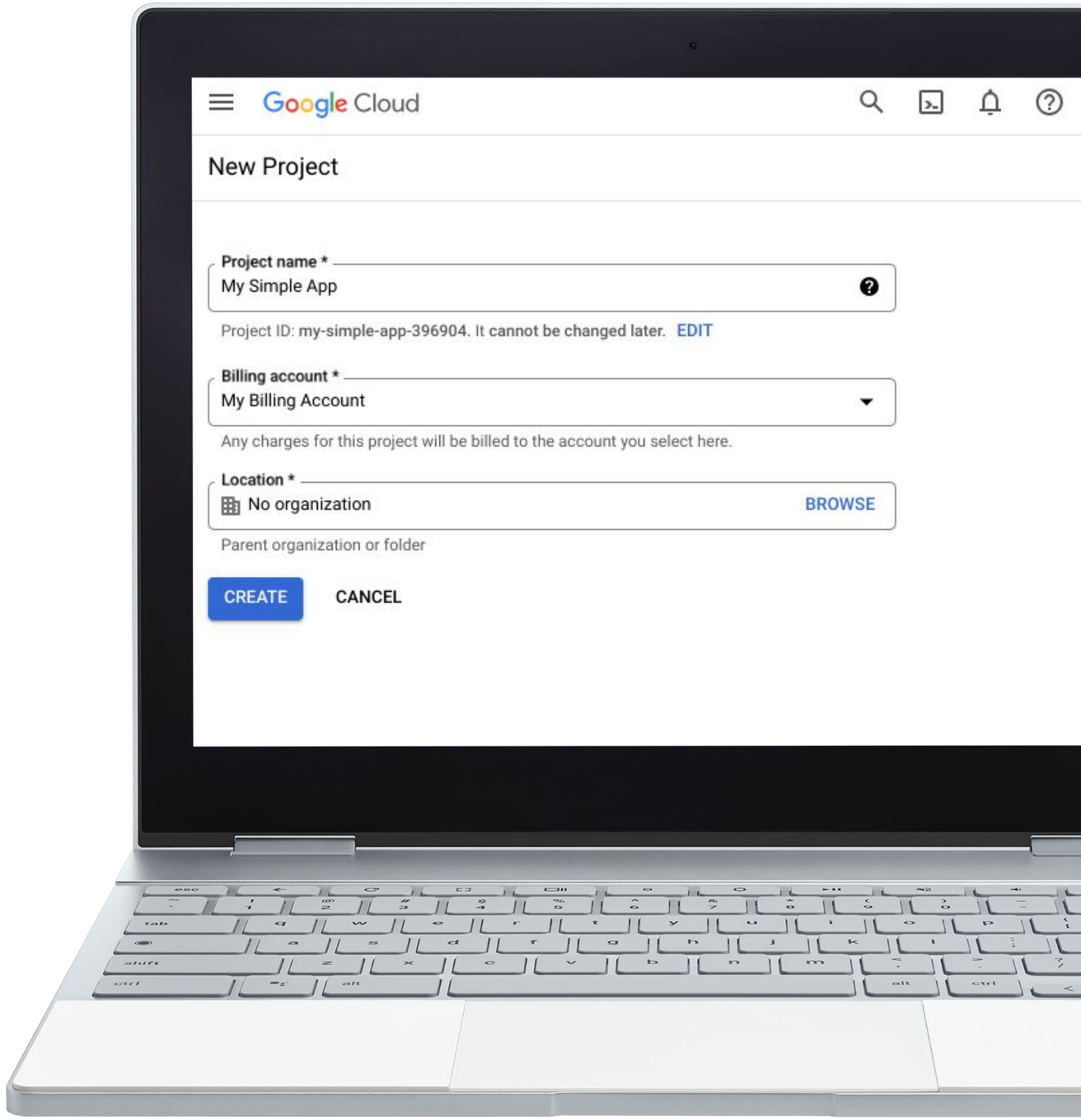
Create a Github Repo

1. Signup for a Github account if you don't already have one
2. Fork the sample repo for this talk. You may also create your own new one, but then you're on your own to make sure your code works.



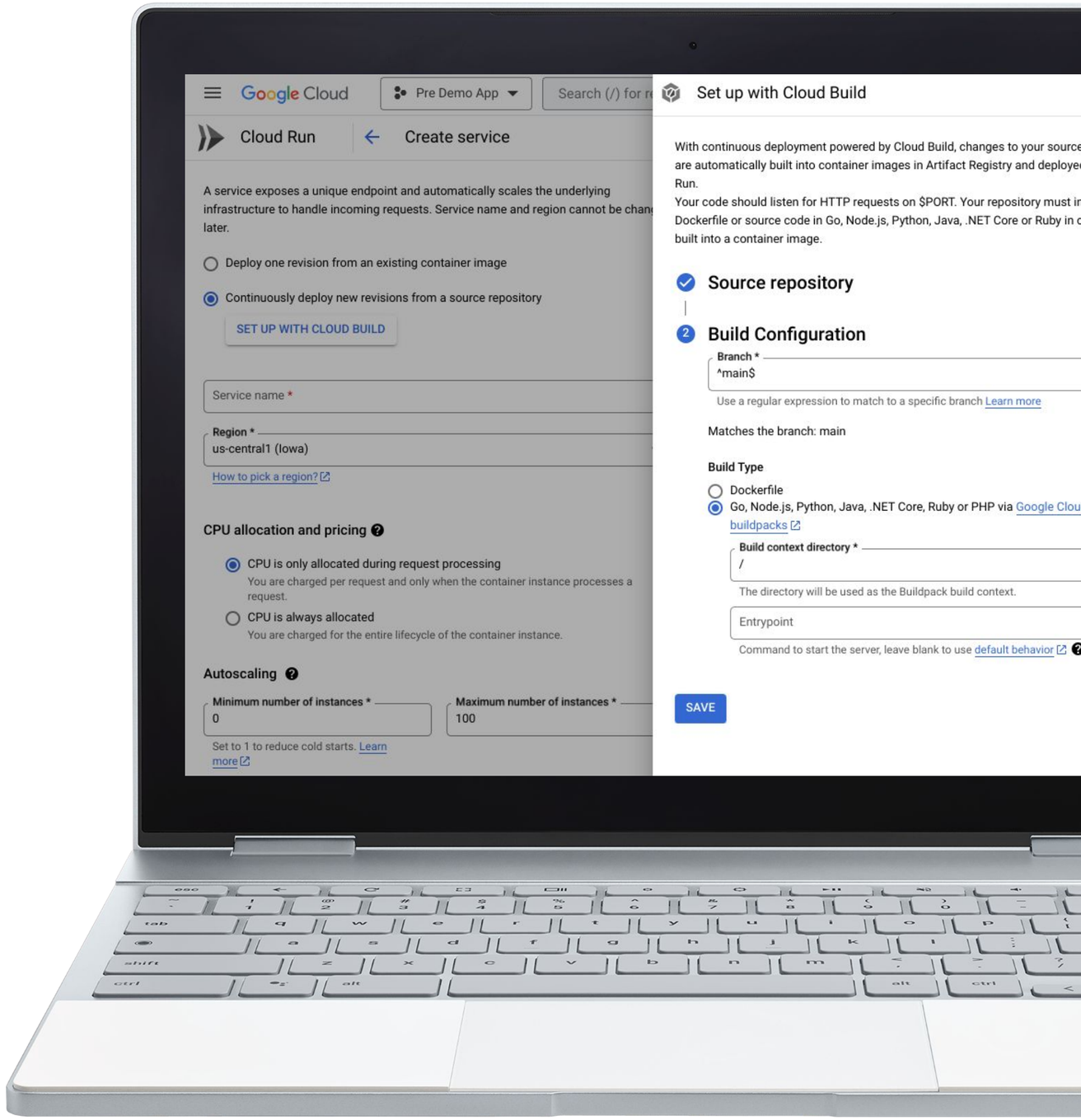
Create a Project

1. In the Google Cloud console, go to Menu > IAM & Admin > Create a Project.
   Go to Create a Project
2. In the Project Name field, enter a descriptive name for your project.
3. In the Location field, click Browse to display potential locations for your project. Then, click Select.
4. Click Create. The console navigates to the Dashboard page and your project is created within a few minutes.



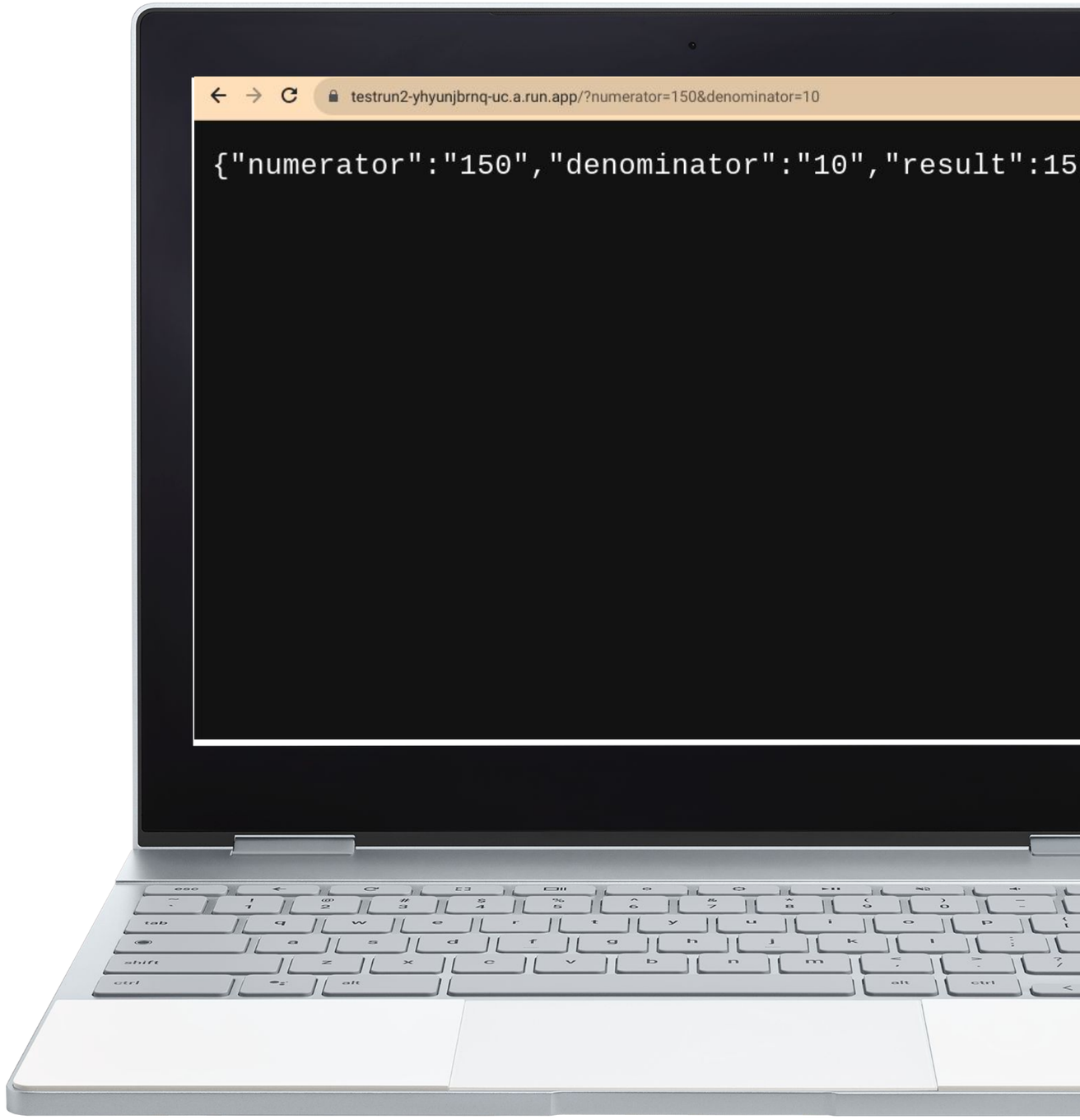
Build a new Cloud Run Service

1. In the Google Cloud console, go to Menu > Cloud Run > Create service.
2. In the form, select Continuously deploy new revisions from a source repository
3. In the Service settings page, click Set up with Cloud Build (enable the necessary API(s) if prompted)
4. Select GitHub as the Provider. If you are not yet authenticated, click Authenticate and follow the instructions.
5. Select the Repository.
6. Click Next.
7. Enter the Branch to indicate what source should be used when running the trigger. You can use a regex
8. Select the Google Cloud Buildpacks Build Type option
9. Click Save
10. Enter a Service name
11. Select Allow unauthenticated invocations as the Authentication
12. Click Create



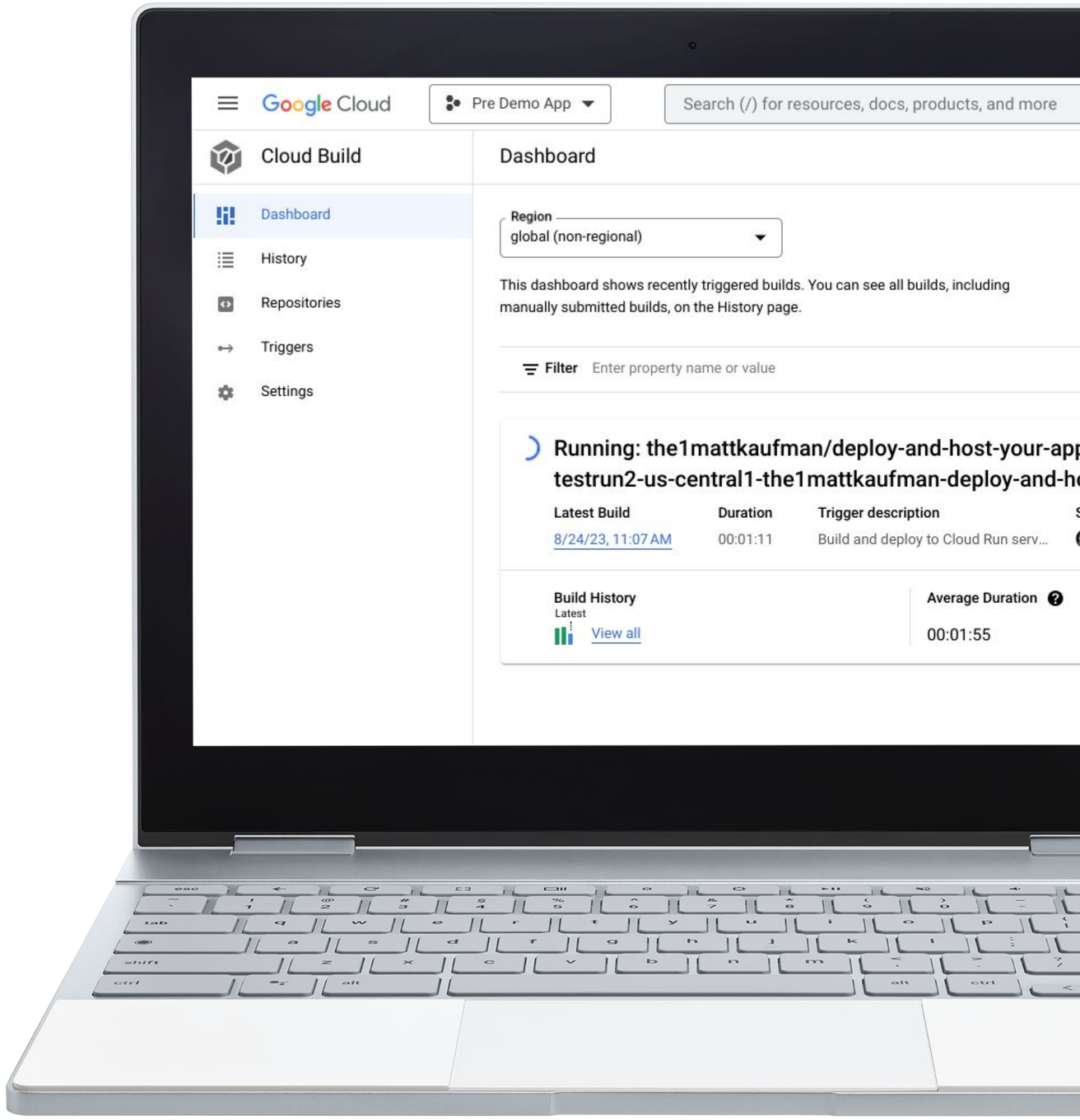
Testing our App

1. In the Google Cloud console, go to Menu > Cloud Run
2. Click on the name of your service.
3. Copy the URL for your service
4. Open a new browser tab and paste the URL followed by *?numerator=150&denominator=10 then hit Enter to visit the URL*.
5. You should see JSON output displaying the numerator, denominator and the result of doing the division.
6. Try different values. You should receive an Internal Server Error when not specifying values or specifying one as 0.
7. Back on your Service Details in the Google Cloud Console, click the LOGS link to confirm that your Error was logged.



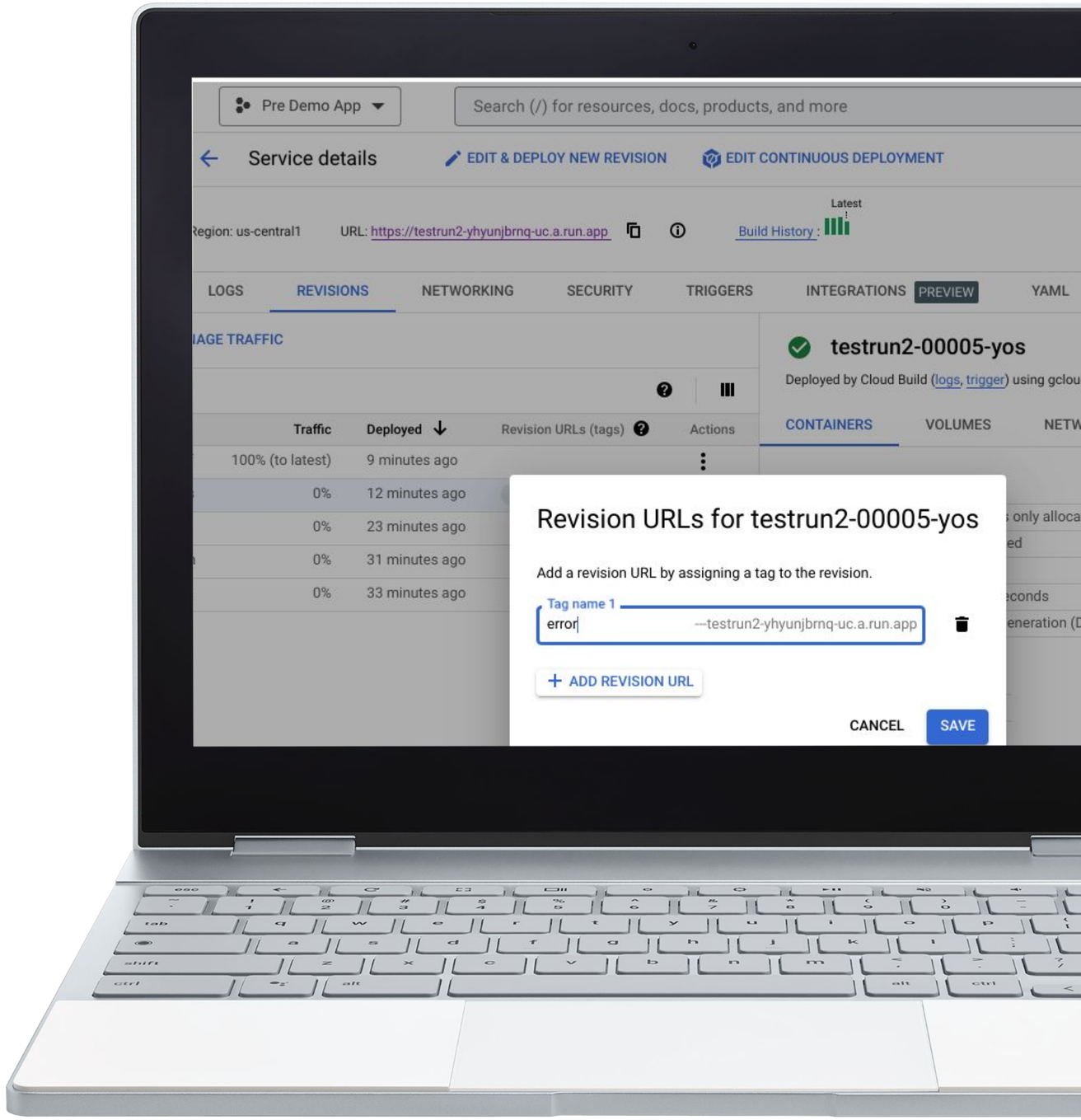
Trigger Building a Revision

1. Using either your preferred IDE or directly in GitHub, make the changes specified in the TODOs in index.js.
2. Commit the changes to the same branch you specified in Build a new Cloud Run Service.
3. In the Google Cloud console, go to Menu > Cloud Build > History.
4. You should see the build just triggered by your commit.
5. Click into the build to view the steps execute.
6. Once the build has completed successfully, repeat the steps previously performed in Testing our App
7. You should now receive valid JSON when testing with a 0 denominator.



Route Traffic

1. In the Google Cloud console, go to Menu > Cloud Run
2. Click on the name of your service.
3. Click the plus icon in the Revision URLs (tags) column for the second revision in the list
4. Enter error as the Tag name 1
5. Click Save
6. Click on the link to open the new Revision URL. The Internal Server Error should appear
7. Confirm the Internal Server Error does not appear when visiting the URL for the service at the top of the page.



Final Thoughts

Efficiency

There tends to be a lot of focus on the performance or efficiency of code.

I prefer to focus on the performance of coders.

By delegating infrastructure and builds to managed services, we’re less distracted and have more time to code.

Groundhog Day

Relax

Slides, code, and more are available at

the1mattkaufman.com

If you joined this talk late or aren't into time travel movies then you have no idea why this slide is funny.

Google I/O Extended

Thank You

Matt Kaufman He/Him

CEO / Coder