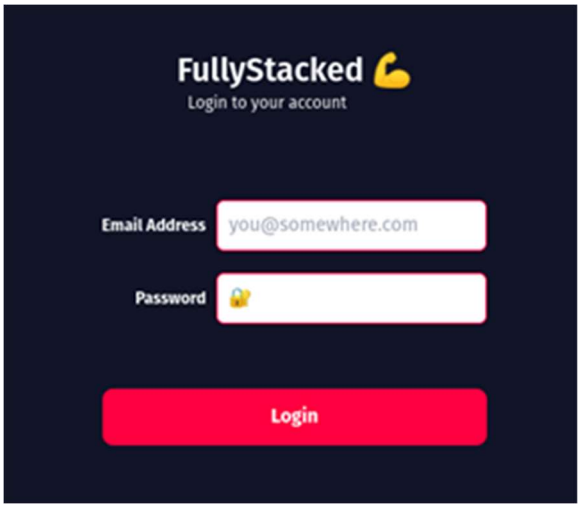
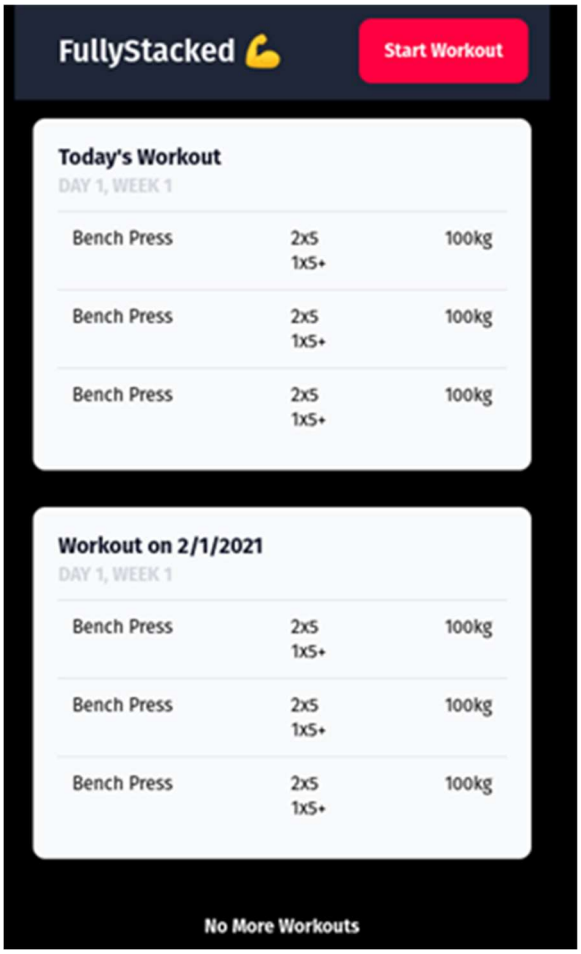
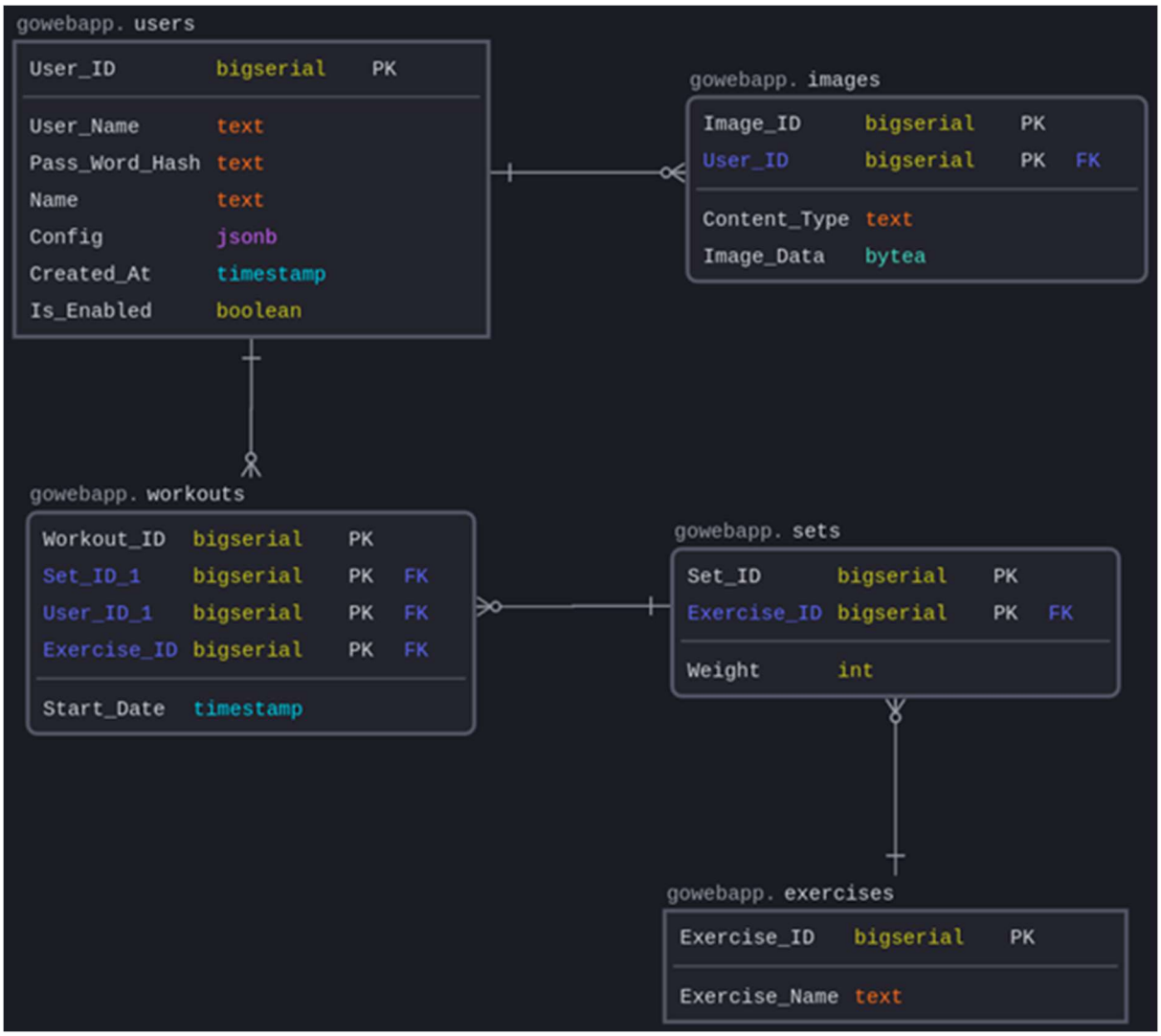
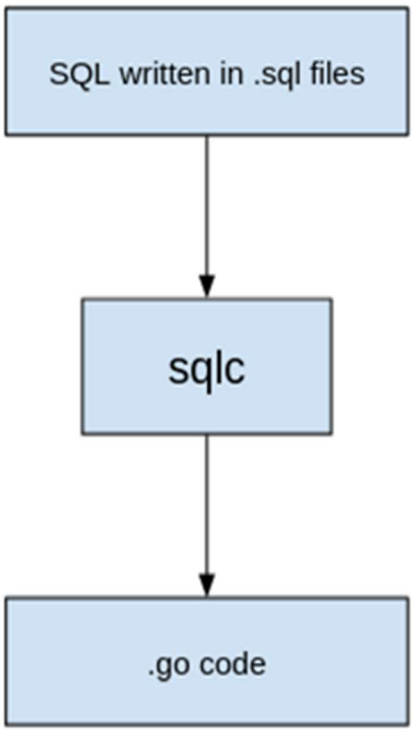
Chapter 1: Building the Database and Model






gowebapp. users
User_ID bigserial PK
User_Name text
Pass_Word_Hash text
Name text
Config jsonb
Created_At timestamp
Is_Enabled boolean
gowebapp. images
Image_ID bigserial PK
User_ID bigserial PK FK
Content_Type text
Image_Data bytea
gowebapp. workouts
Workout_ID bigserial PK
Set_ID_1 bigserial PK FK
User_ID_1 bigserial PK FK
Exercise_ID bigserial PK FK
Start_Date timestamp
gowebapp. sets
Set_ID bigserial PK
Exercise_ID bigserial PK FK
Weight int
gowebapp. exercises
Exercise_ID bigserial PK
Exercise_Name text


SQL written in .sql files
sqlc
.go code

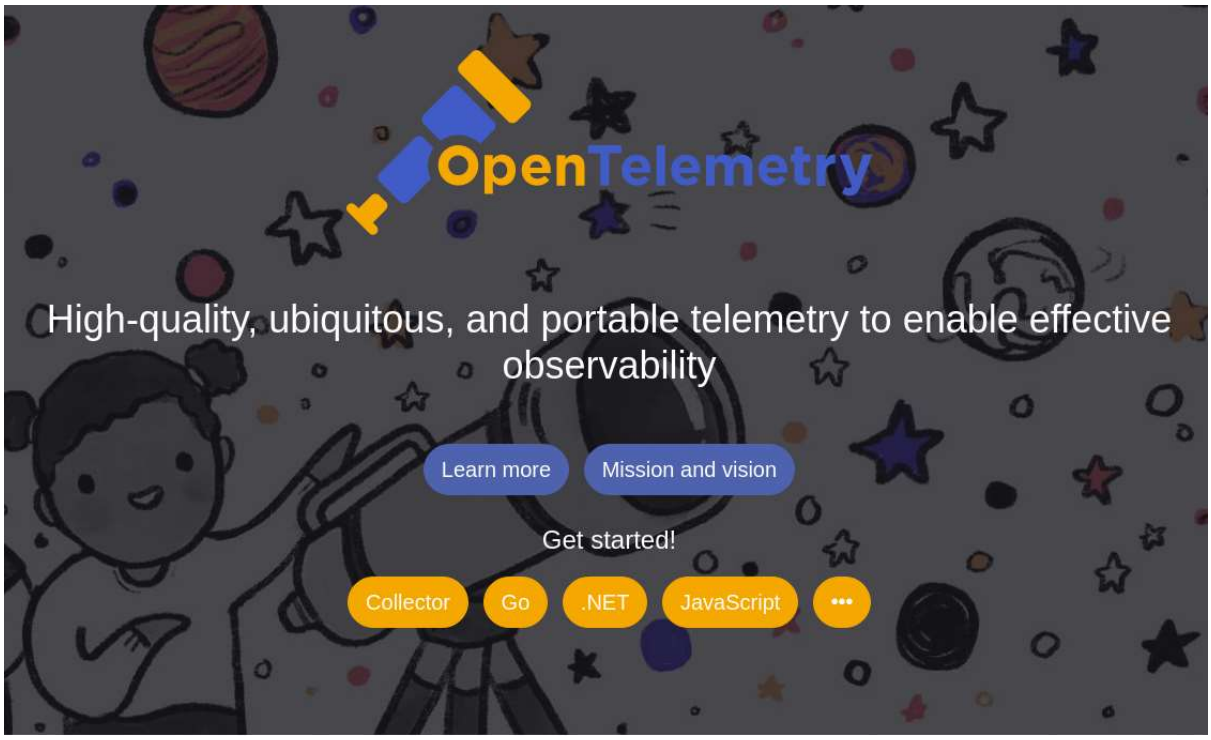
Chapter 2: Application Logging

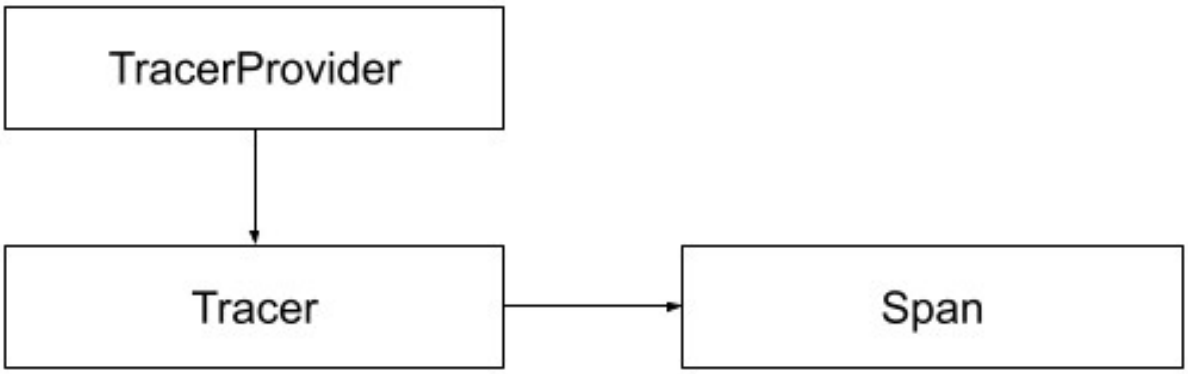
```
2021/10/15 13:43 This is a raw message, no levels, no colors.
[INFO] 2021/10/15 13:43 This is an info message, with colors (if the output is terminal)
[WARN] 2021/10/15 13:43 This is a warning message
[ERRO] 2021/10/15 13:43 This is an error message
[DBUG] 2021/10/15 13:43 This is a debug message
[FTAL] 2021/10/15 13:43 Fatal will exit no matter what
```

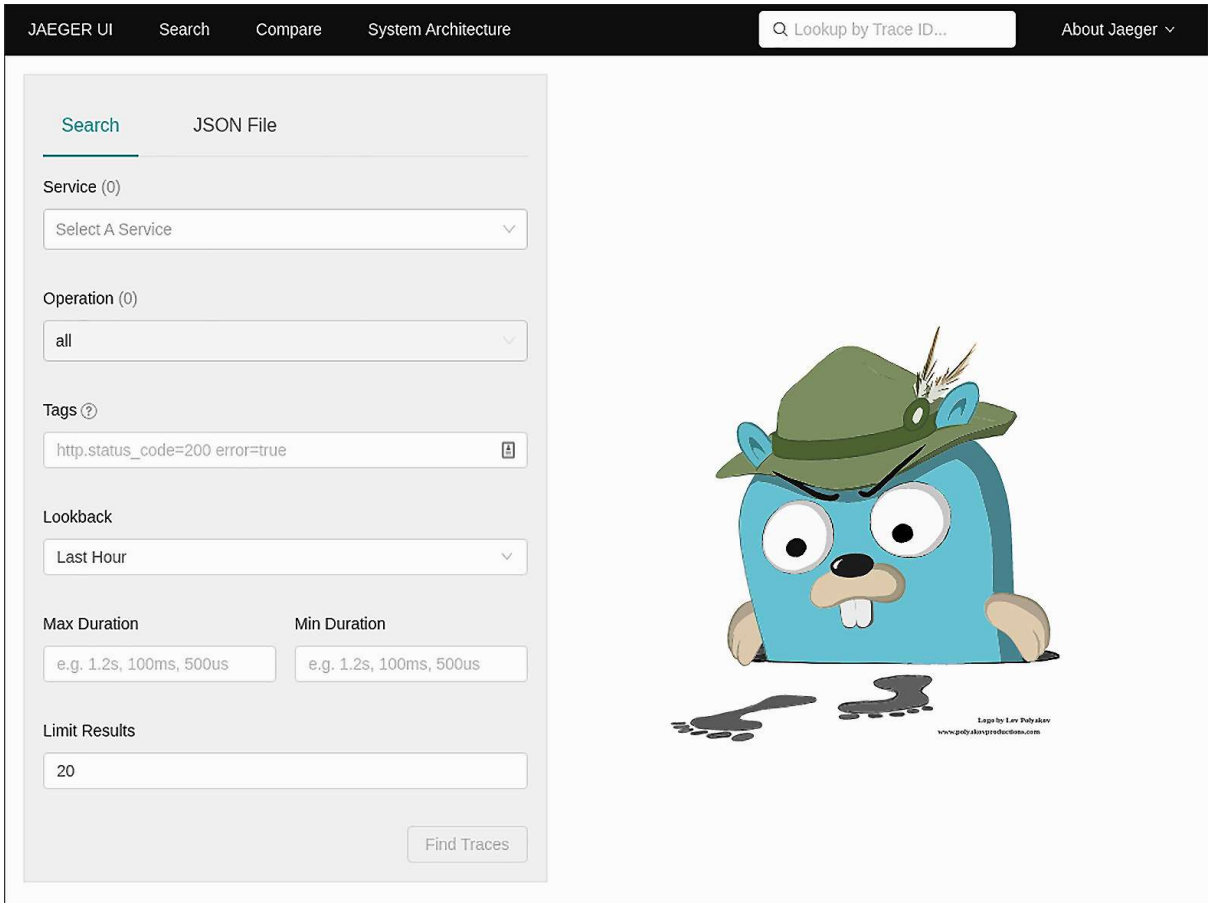
```
[DBUG] 2021/10/17 19:15 Application logging to stdout = true
```

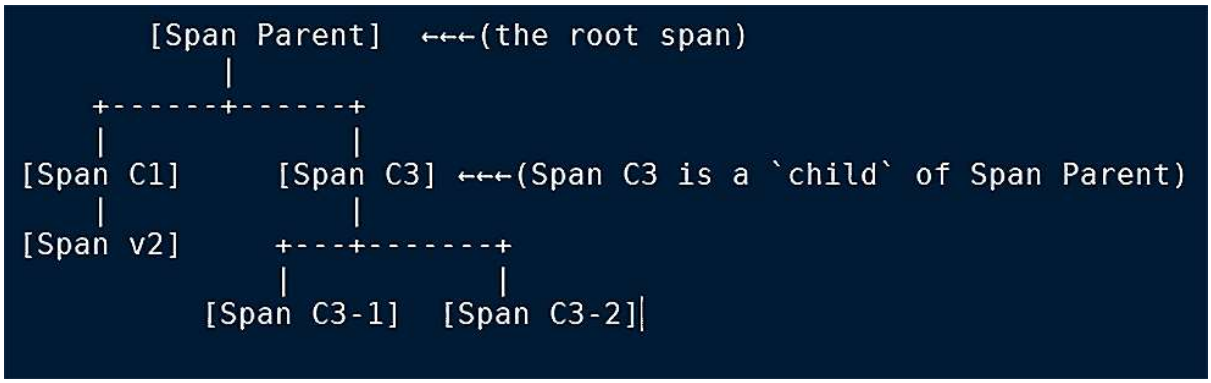
```
[INFO] 2021/10/17 19:16 Starting the application...
[INFO] 2021/10/17 19:16 Database connection fine
[INFO] 2021/10/17 19:16 Success - user creation
[INFO] 2021/10/17 19:16 Success - exercise creation
[INFO] 2021/10/17 19:16 Success - updating workout
[INFO] 2021/10/17 19:16 Application complete
```

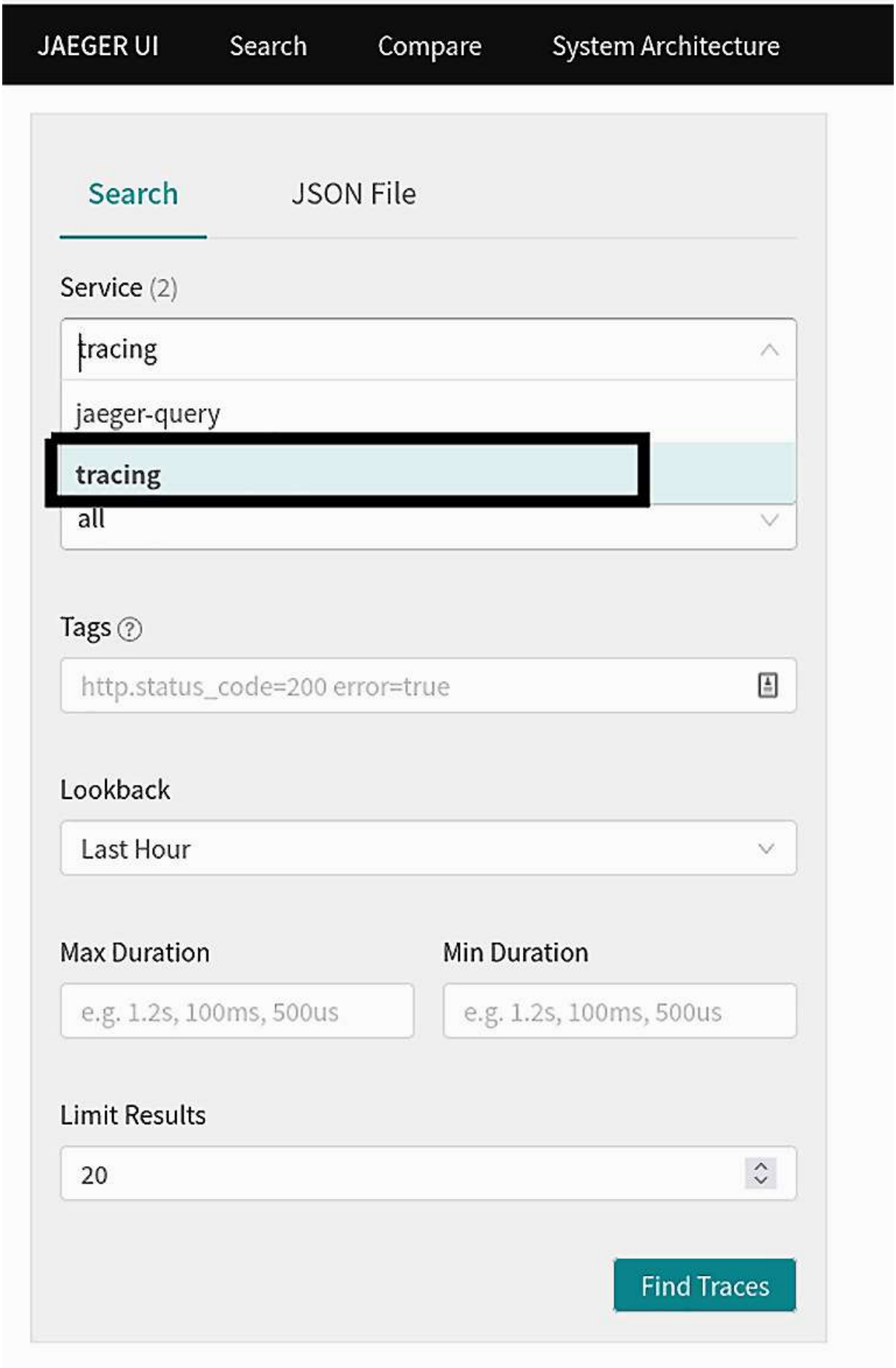
Chapter 3: Application Metrics and Tracing

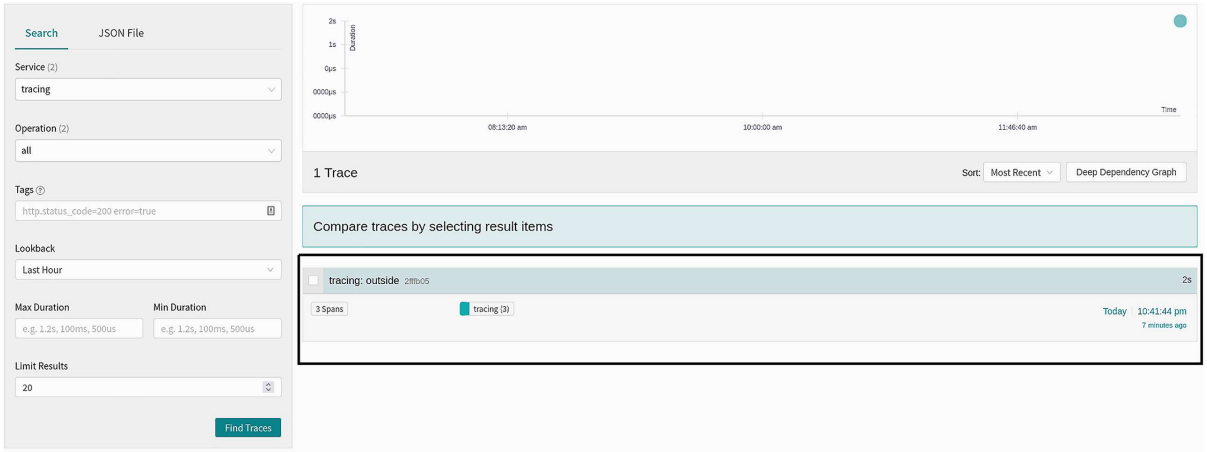


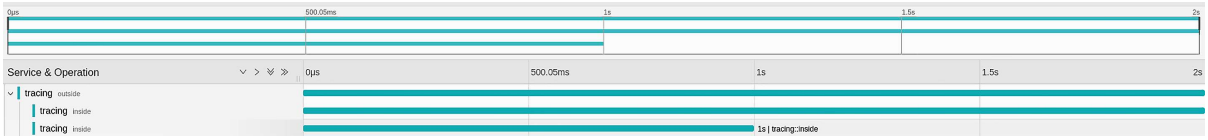


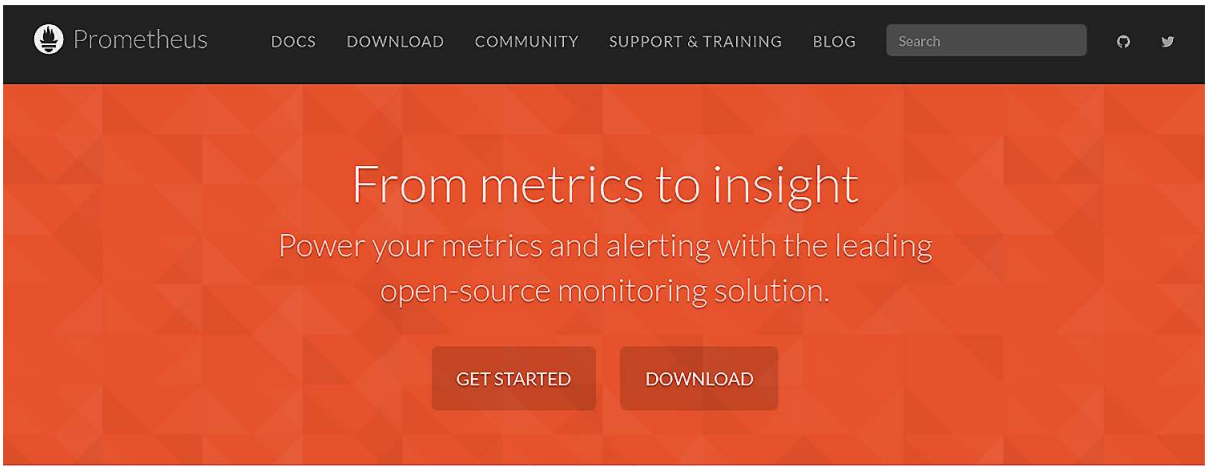

JAEGER UI
Search
Compare
System Architecture
Lookup by Trace ID...
About Jaeger
Search
JSON File
Service (0)
Select A Service
Operation (0)
all
Tags
http.status_code=200 error=true
Lookback
Last Hour
Max Duration
e.g. 1.2s, 100ms, 500us
Min Duration
e.g. 1.2s, 100ms, 500us
Limit Results
20
Find Traces


[Span Parent] ←←←(the root span)
[Span C1]
[Span C3] ←←←(Span C3 is a `child` of Span Parent)
[Span v2]
[Span C3-1]
[Span C3-2]


JAEGER UI
Search
Compare
System Architecture
Search
JSON File
Service (2)
tracing
jaeger-query
tracing
all
Tags
http.status_code=200 error=true
Lookback
Last Hour
Max Duration
e.g. 1.2s, 100ms, 500us
Min Duration
e.g. 1.2s, 100ms, 500us
Limit Results
20
Find Traces







Dimensional data

Prometheus implements a highly dimensional data model. Time series are identified by a metric name and a set of key-value pairs.

Powerful queries

PromQL allows slicing and dicing of collected time series data in order to generate ad-hoc graphs, tables, and alerts.

Great visualization

Prometheus has multiple modes for visualizing data: a built-in expression browser, Grafana integration, and a console template language.

Efficient storage

Prometheus stores time series in memory and on local disk in an efficient custom format. Scaling is achieved by functional sharding and federation.

Simple operation

Each server is independent for reliability, relying only on local storage. Written in Go, all binaries are statically linked and easy to deploy.

Precise alerting

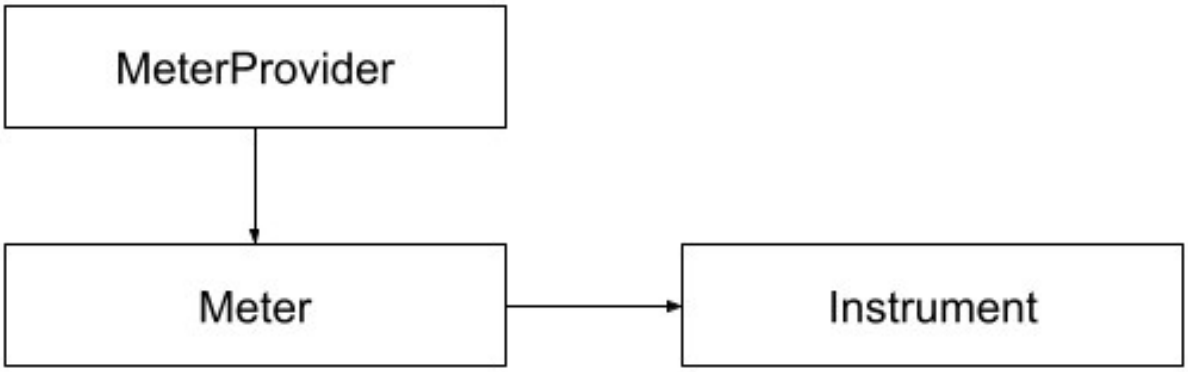
Alerts are defined based on Prometheus's flexible PromQL and maintain dimensional information. An alertmanager handles notifications and silencing.

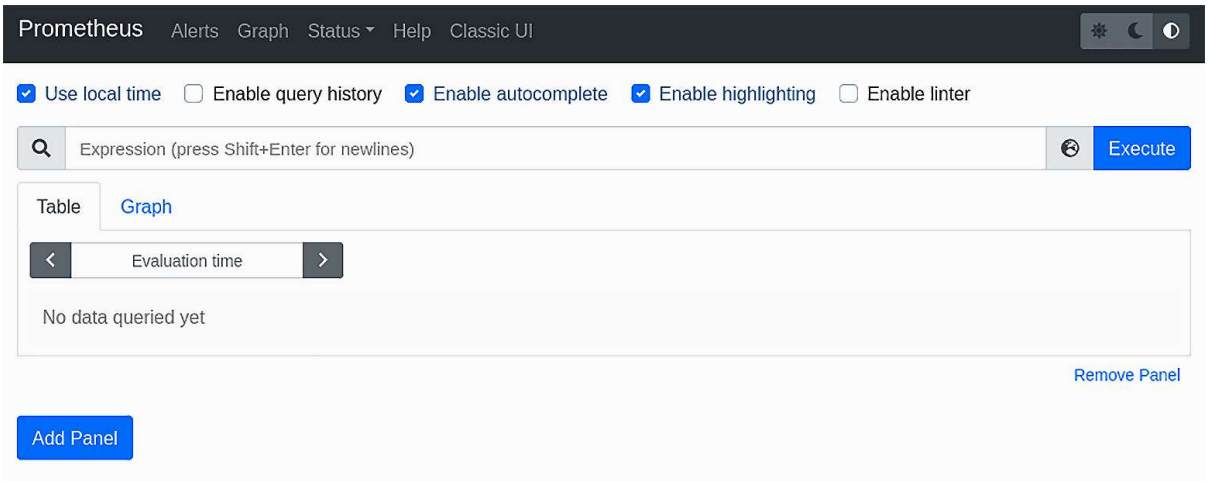
Many client libraries

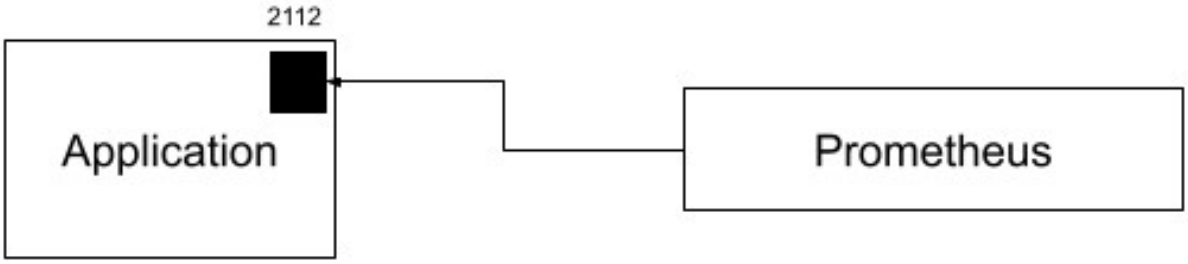
Client libraries allow easy instrumentation of services. Over ten languages are supported already and custom libraries are easy to implement.

Many integrations

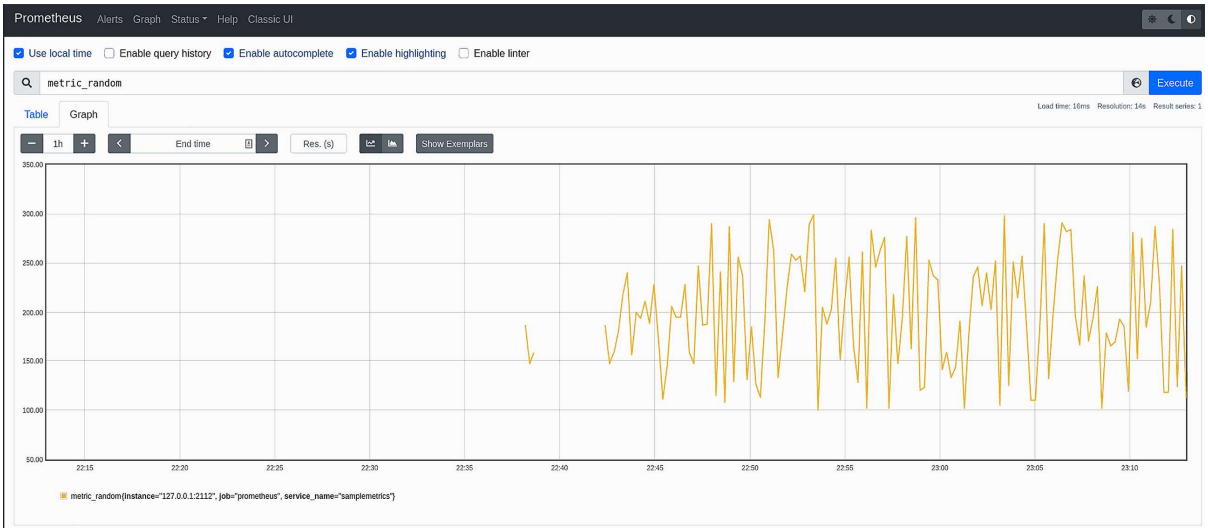
Existing exporters allow bridging of third-party data into Prometheus. Examples: system statistics, as well as Docker, HAProxy, StatsD, and JMX metrics.


MeterProvider
Meter
Instrument

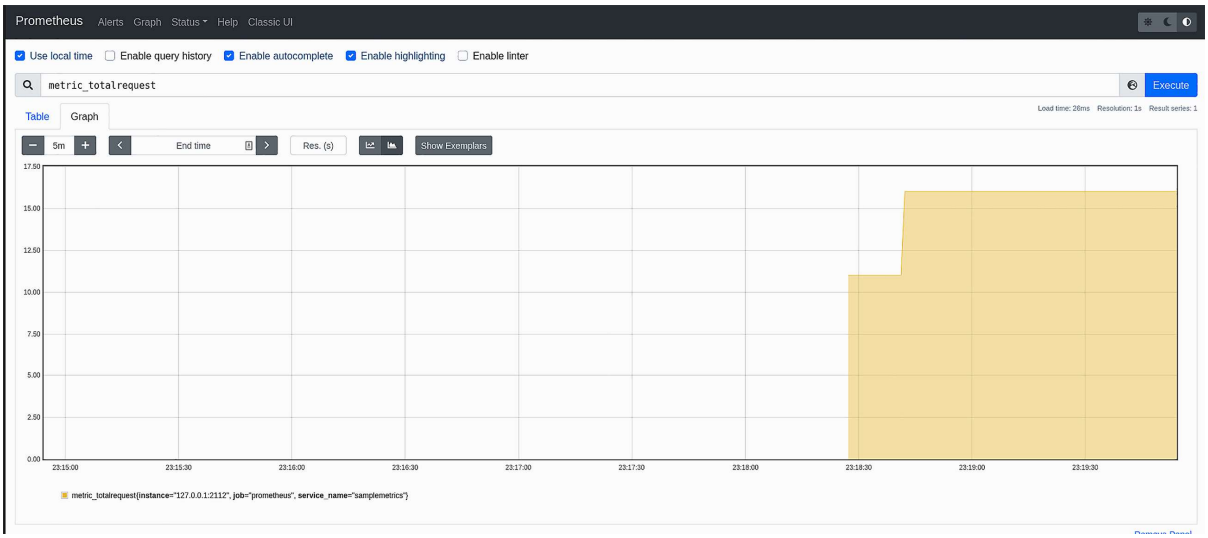

Prometheus
Alerts
Graph
Status
Help
Classic UI
Use local time
Enable query history
Enable autocomplete
Enable highlighting
Enable linter
Expression (press Shift+Enter for newlines)
Execute
Table
Graph
Evaluation time
No data queried yet
Remove Panel
Add Panel


2112
Application
Prometheus

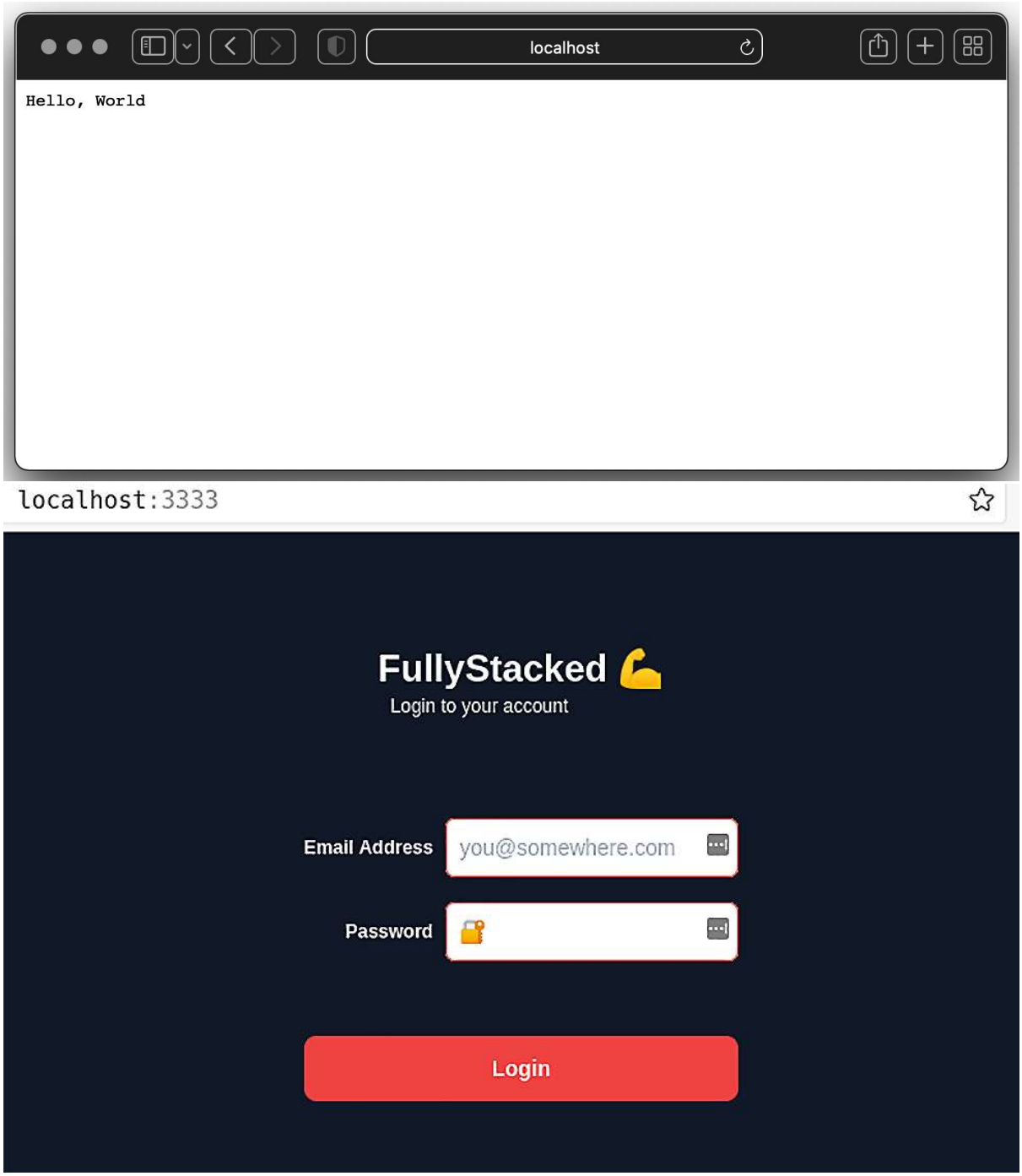

metric_random
metric_random
gauge
Random numbers
Table
Graph


Prometheus
Alerts
Graph
Status
Help
Classic UI
Use local time
Enable query history
Enable autocomplete
Enable highlighting
Enable linter
metric_random
Execute
Table
Graph
Load time: 16ms
Resolution: 14s
Result series: 1
1h
End time
Res. (s)
Show Exemplars
350.00
300.00
250.00
200.00
150.00
100.00
50.00
22:15
22:20
22:25
22:30
22:35
22:40
22:45
22:50
22:55
23:00
23:05
23:10
metric_random{instance="127.0.0.1:2112", job="prometheus", service_name="samplemetrics"}

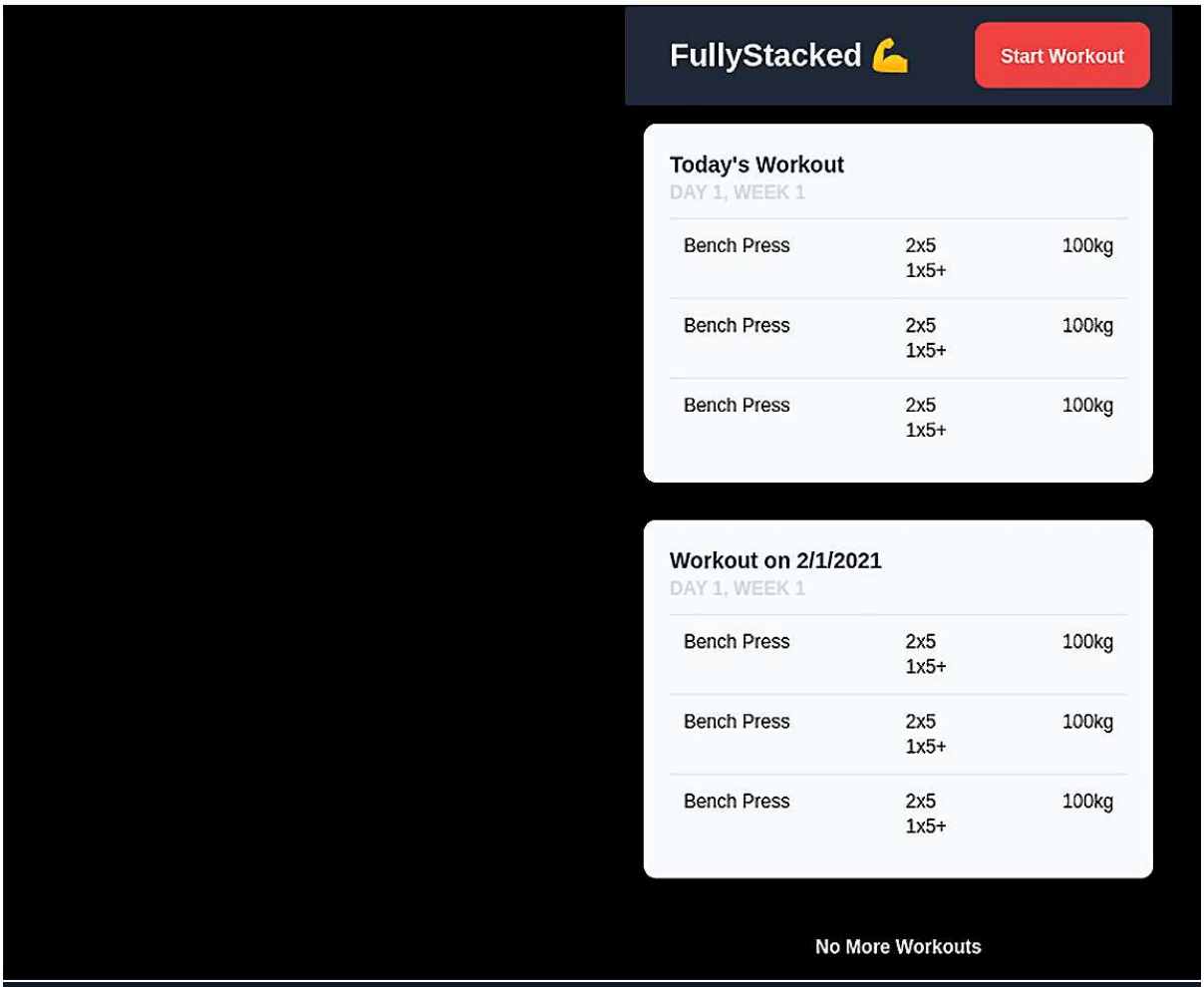

metric_
metric_random
gauge
metric_totalrequest
histogram_quantile(__quantile__, sum by(le) (rate(__histogram_metric__[5m]))) snippet
scrape_samples_post_metric_relabeling
Execute
Table
Load time: 16ms
Resolution: 14s
Result series: 1


Prometheus
Alerts
Graph
Status
Help
Classic UI
Use local time
Enable query history
Enable autocomplete
Enable highlighting
Enable linter
metric_totalrequest
Execute
Table
Graph
Load time: 26ms
Resolution: 1s
Result series: 1
5m
End time
Res. (s)
Show Exemplars
17.50
15.00
12.50
10.00
7.50
5.00
2.50
0.00
23:15:00
23:15:30
23:16:00
23:16:30
23:17:00
23:17:30
23:18:00
23:18:30
23:19:00
23:19:30
metric_totalrequest{instance="127.0.0.1:2112", job="prometheus", service_name="samplemetrics"}

Chapter 4: Serving and Embedding HTML Content

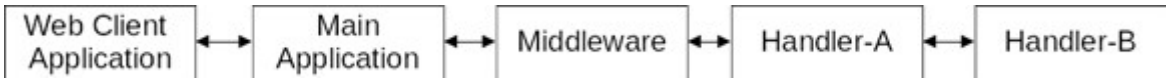


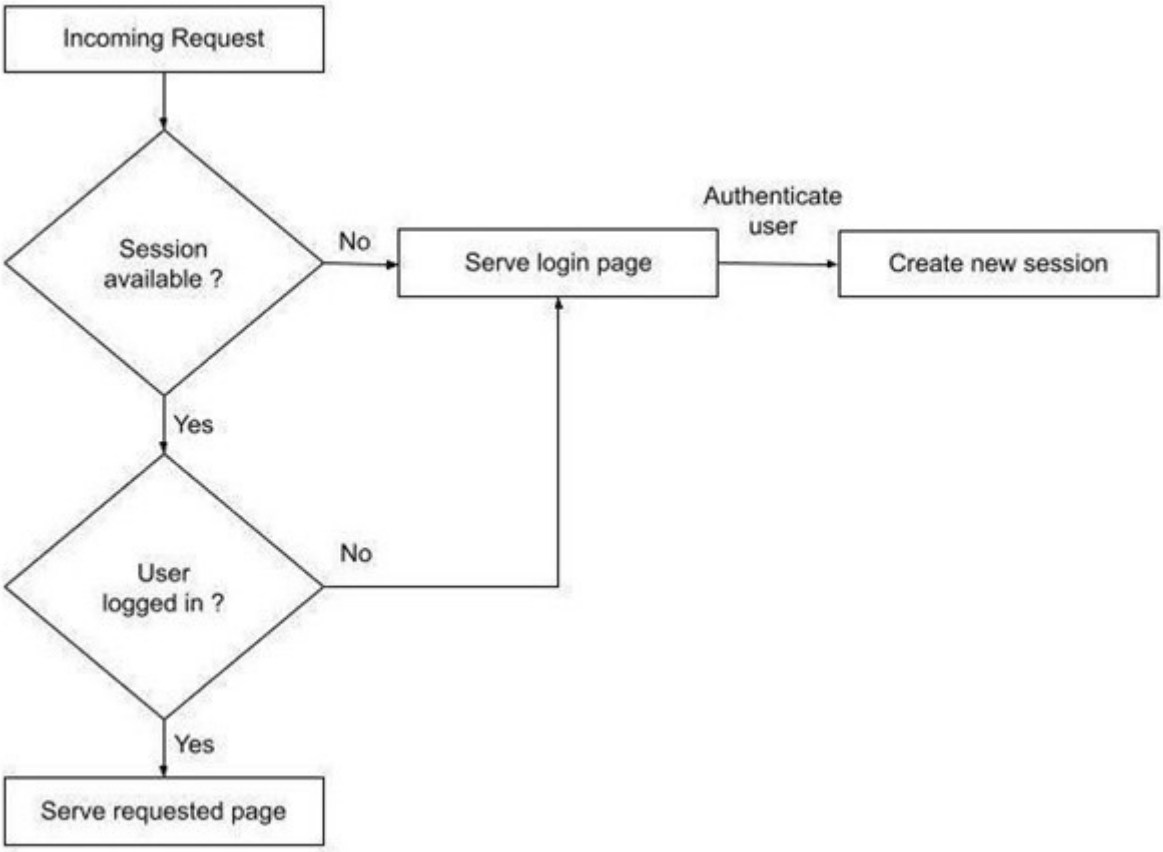
localhost:3333/dashboard.html





Chapter 5: Securing the Backend and Middleware



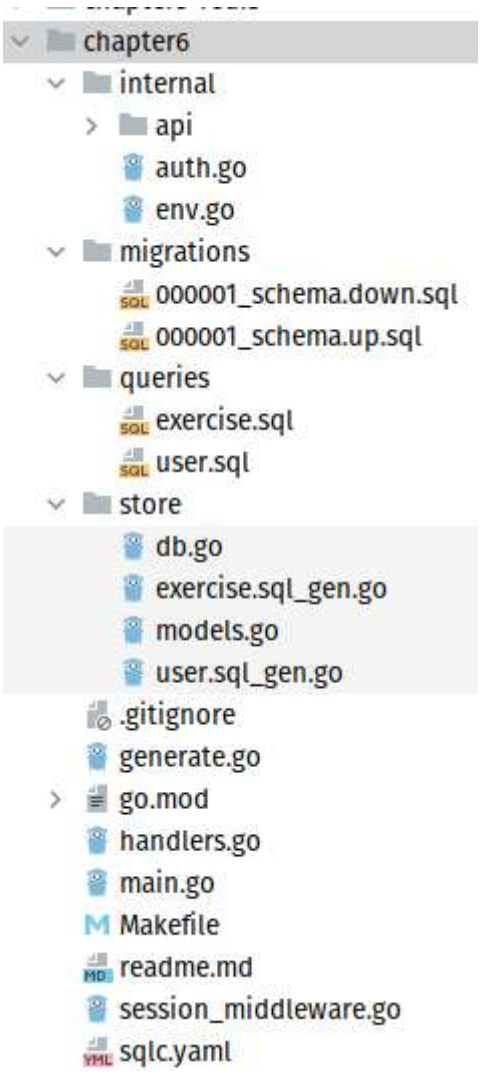


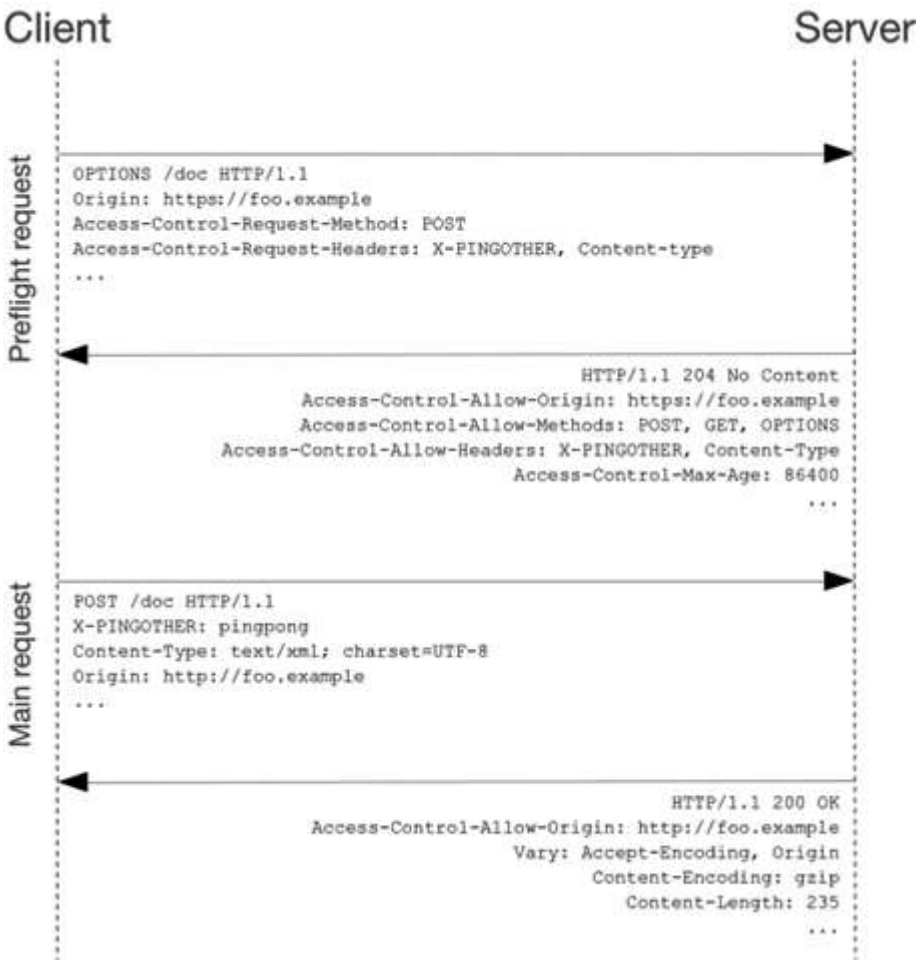


Request Headers (472 B) Raw

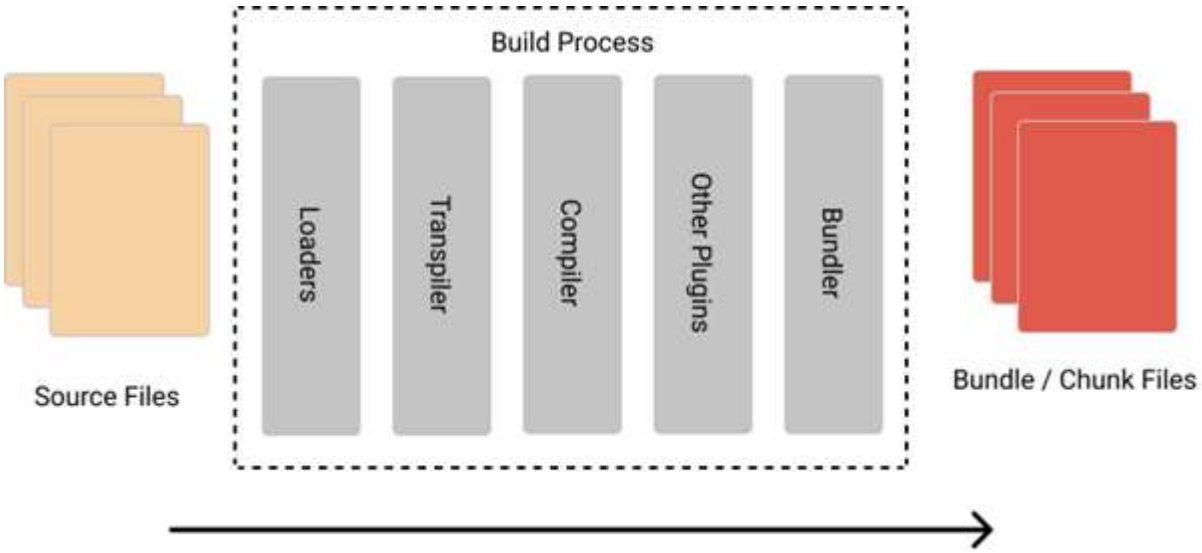
```
Accept: image/webp,*/*
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Cookie: session_token=MTY0MzIwMDQ0OXxEdi1CQkFFQ180SUFBUkFCRUFBQUpmLUNBQUVHYzNSeWFXNW5EQThBRFdG
MWRHaGxiblJwWTJGMFpXUUVZbTl2YkFJQ0FBQT18f6UKeoMoR1MOfLWETjfQd3YislSGPiuK0Q_6brfFVuk=
DNT: 1
Host: localhost:3333
Referer: http://localhost:3333/login
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:89.0) Gecko/20100101 Firefox/89.0
```

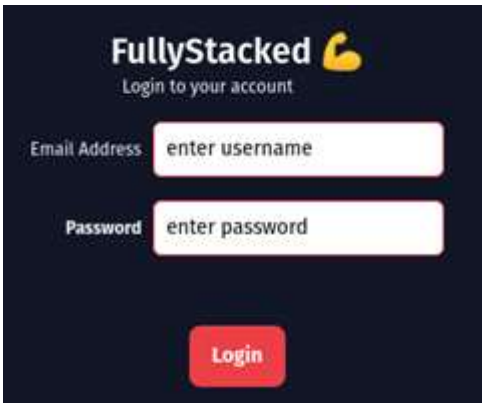
Chapter 6: Moving to API-First




Client
Server
Preflight request
OPTIONS /doc HTTP/1.1
Origin: https://foo.example
Access-Control-Request-Method: POST
Access-Control-Request-Headers: X-PINGOTHER, Content-type
...
HTTP/1.1 204 No Content
Access-Control-Allow-Origin: https://foo.example
Access-Control-Allow-Methods: POST, GET, OPTIONS
Access-Control-Allow-Headers: X-PINGOTHER, Content-Type
Access-Control-Max-Age: 86400
...
Main request
POST /doc HTTP/1.1
X-PINGOTHER: pingpong
Content-Type: text/xml; charset=UTF-8
Origin: http://foo.example
...
HTTP/1.1 200 OK
Access-Control-Allow-Origin: http://foo.example
Vary: Accept-Encoding, Origin
Content-Encoding: gzip
Content-Length: 235
...

Chapter 7: Frontend Frameworks





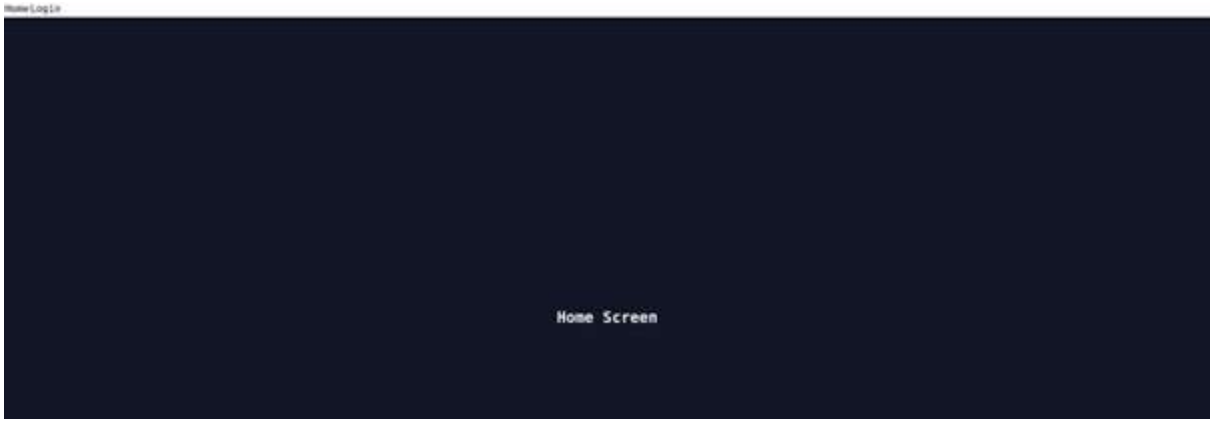

```
> test@1.0.0 start
> http-server ./ -p 3000

Starting up http-server, serving ./
Available on:
  http://127.0.0.1:3000
  http://192.168.1.3:3000
  http://172.17.0.1:3000
Hit CTRL-C to stop the server
```

```
vite v2.8.6 dev server running at:

> Local: http://localhost:3000/
> Network: use `--host` to expose

ready in 202ms.
```

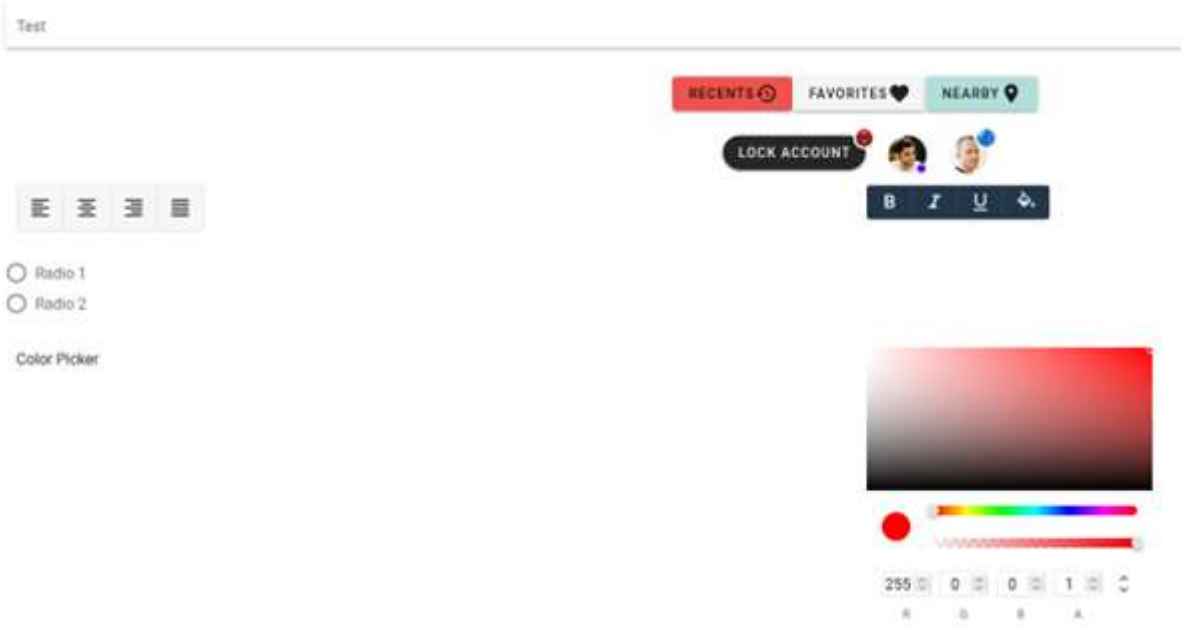


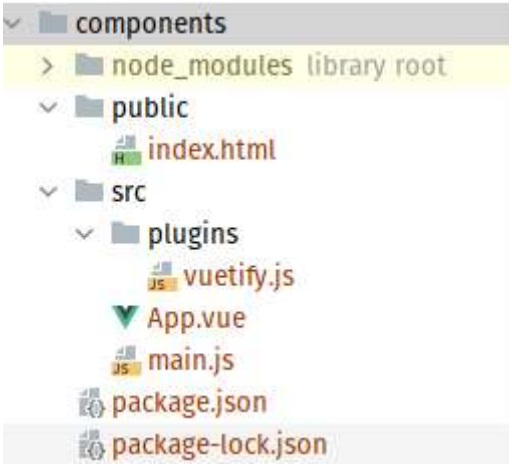
Chapter 8: Frontend Libraries

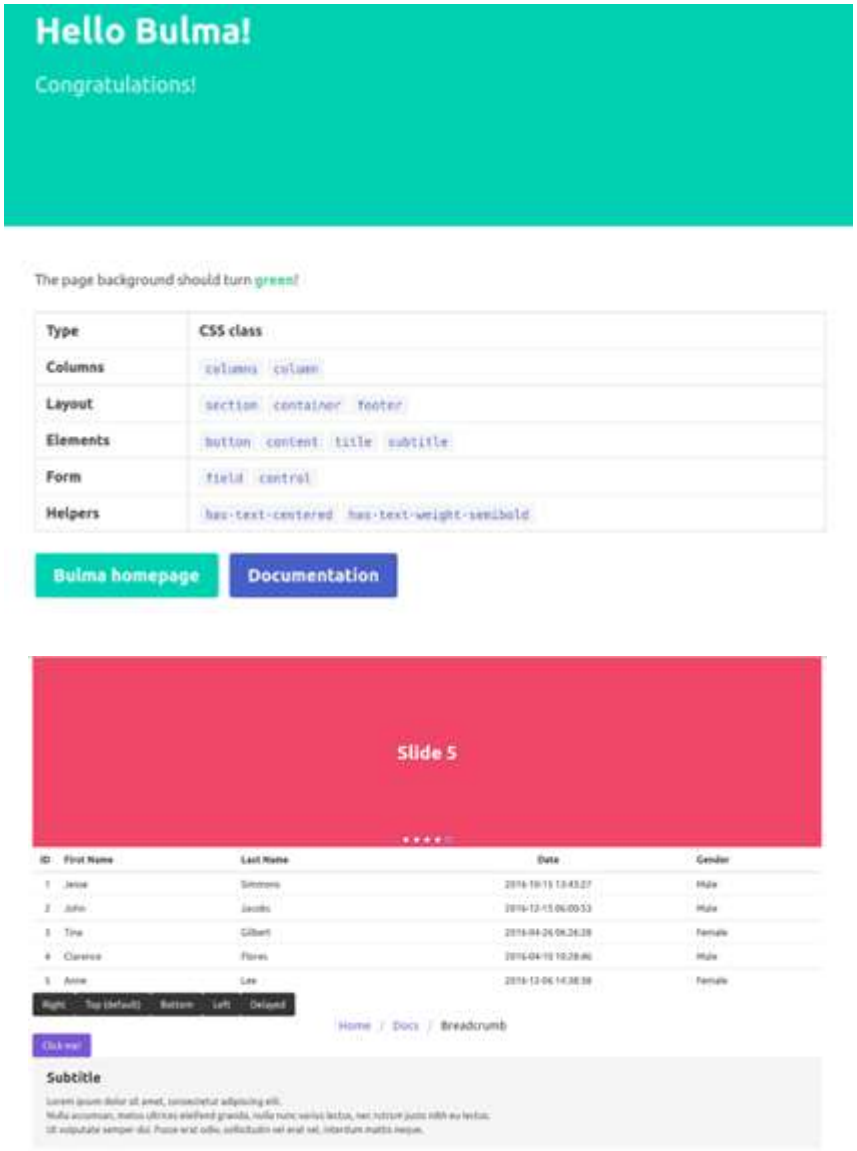
```
 DONE  Compiled successfully in 7784ms


App running at:
- Local:   http://localhost:8080/
- Network: http://192.168.1.5:8080/

Note that the development build is not optimized.
To create a production build, run npm run build.
```







The Email Address must be at least 6 characters.
The Email Address must be a valid email.
Email Address
nick
Send
email: nick

nick@bar.com


```
{
  "text": {
    "required": true,
    "email": true,
    "minLength": true,
    "$model": "nick@bar.com",
    "$invalid": false,
    "$dirty": true,
    "$anyDirty": true,
    "$error": false,
    "$anyError": false,
    "$pending": false,
    "$params": {
      "required": {
        "type": "required"
      },
      "email": {
        "type": "email"
      },
      "minLength": {
        "type": "minLength",
        "min": 2
      }
    }
  },
  "$model": null,
  "$invalid": false,
  "$dirty": true,
  "$anyDirty": true,
  "$error": false,
  "$anyError": false,
  "$pending": false,
  "$params": {
    "text": null
  }
}
```


3412 212112 11121

AMERICAN EXPRESS

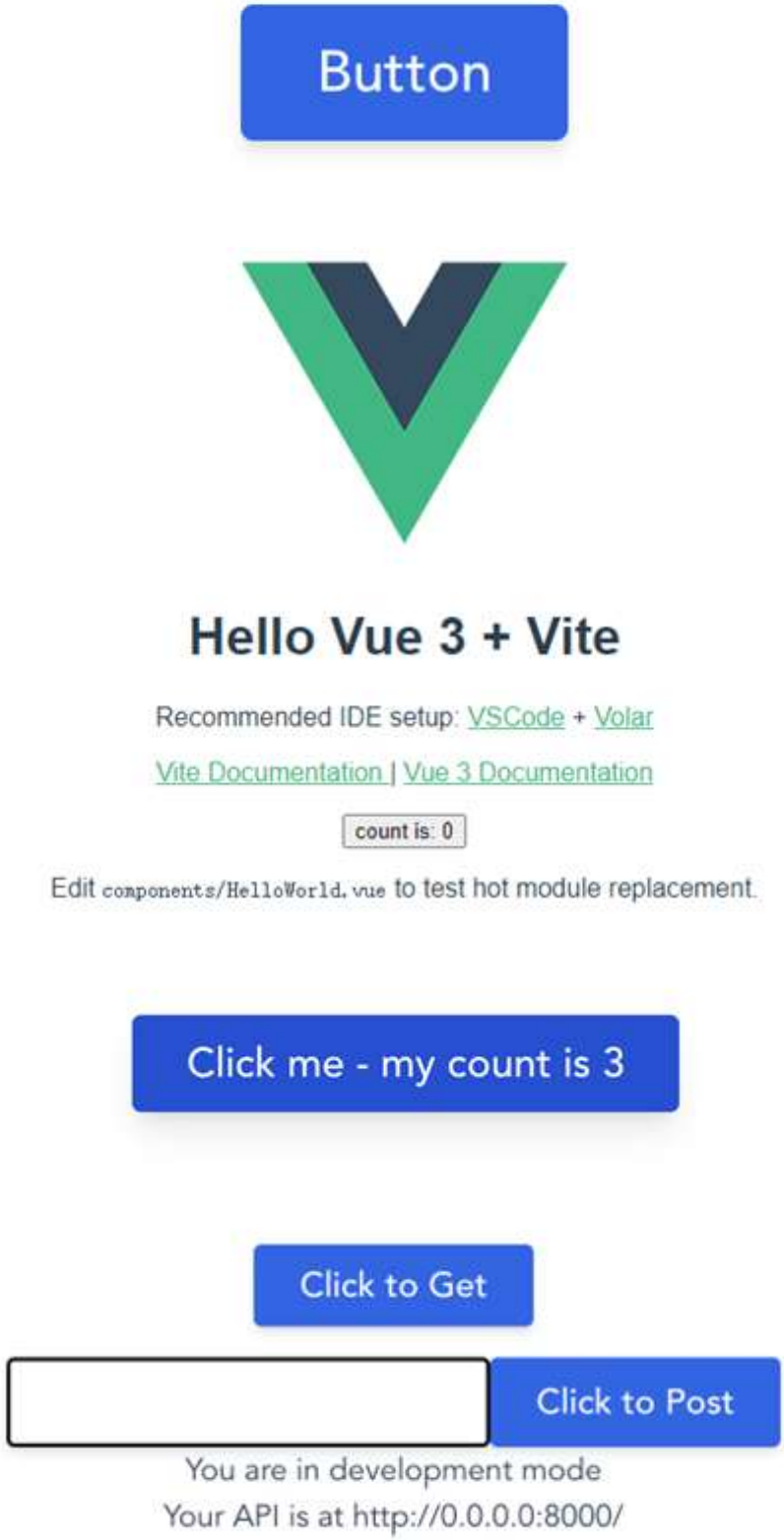
American Express: starts with 34/37

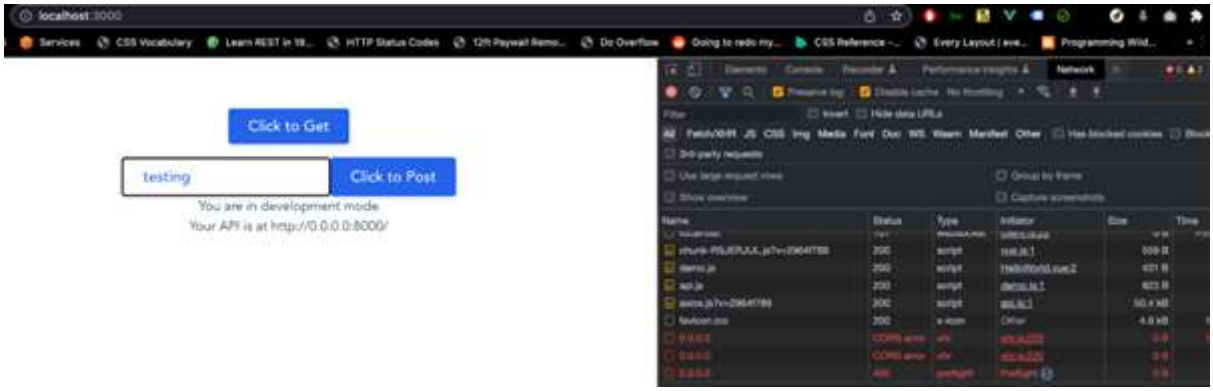
34•• •••••• •••••

4242 4222 4242 4244

4242 4222 4242 4244 visa

Chapter 9: Tailwind, Middleware, and CORS





Click to Get

eMesreveR

Click to Post

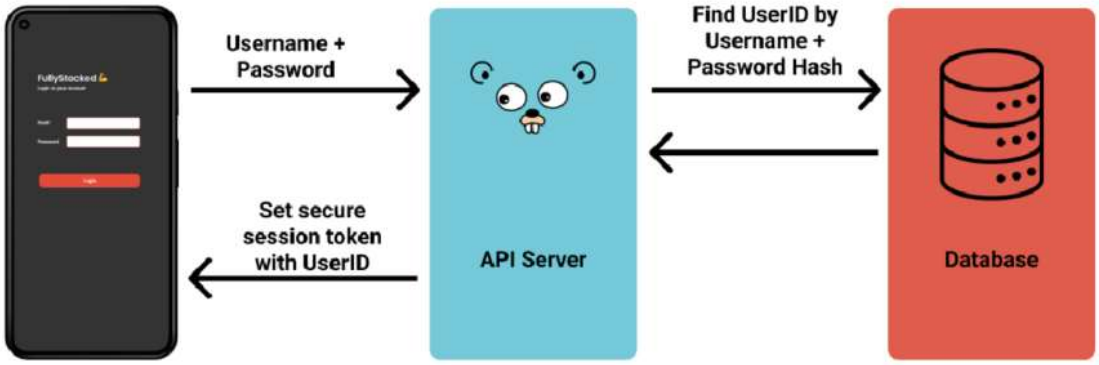
You are in development mode

Your API is at http://0.0.0.0:8000/

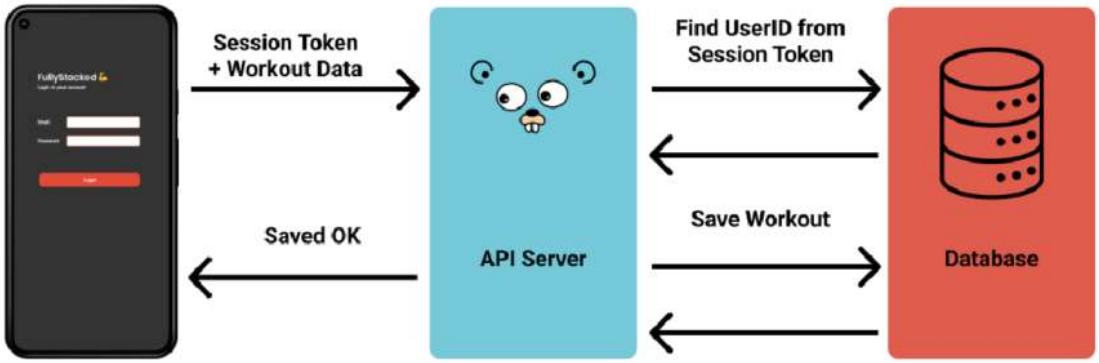
- gnitset
- Hello World
- eMesreveR
- ReverseMe

Chapter 10: Session Management

Login



Save Workout



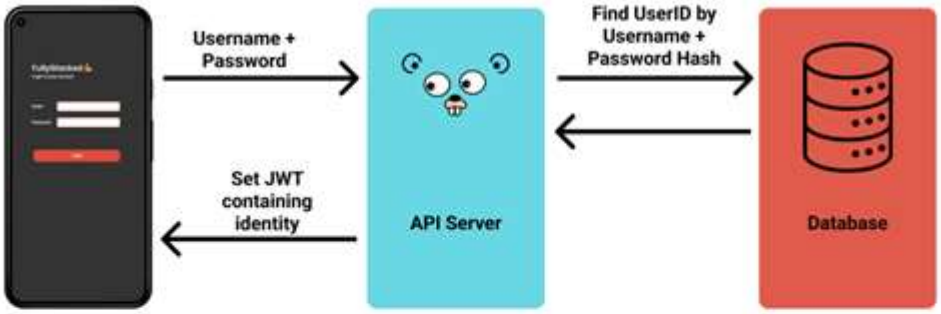
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwidGl0bGUiOiJGdWxsIFN0YWNrIEdvIiwiaWF0IjoxNTE2MjM5MDIyfQ.Bph4Comr-law35-d1kJSDS3riNiS7GsF2NwJNB8I-0M

Header - algorithm used and the token type

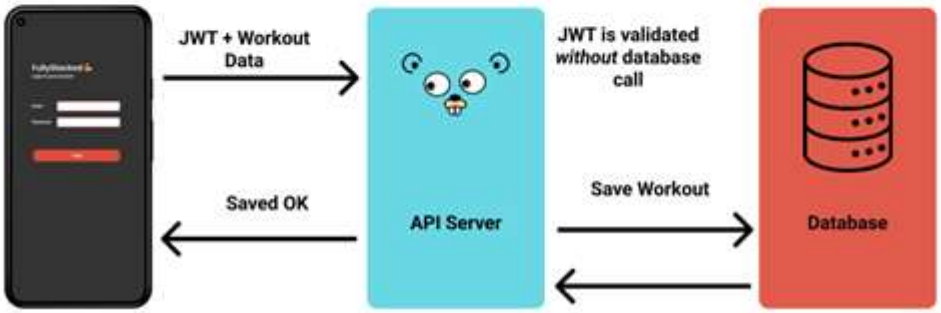
Payload - The data and claims

Signature - Used to verify authenticity

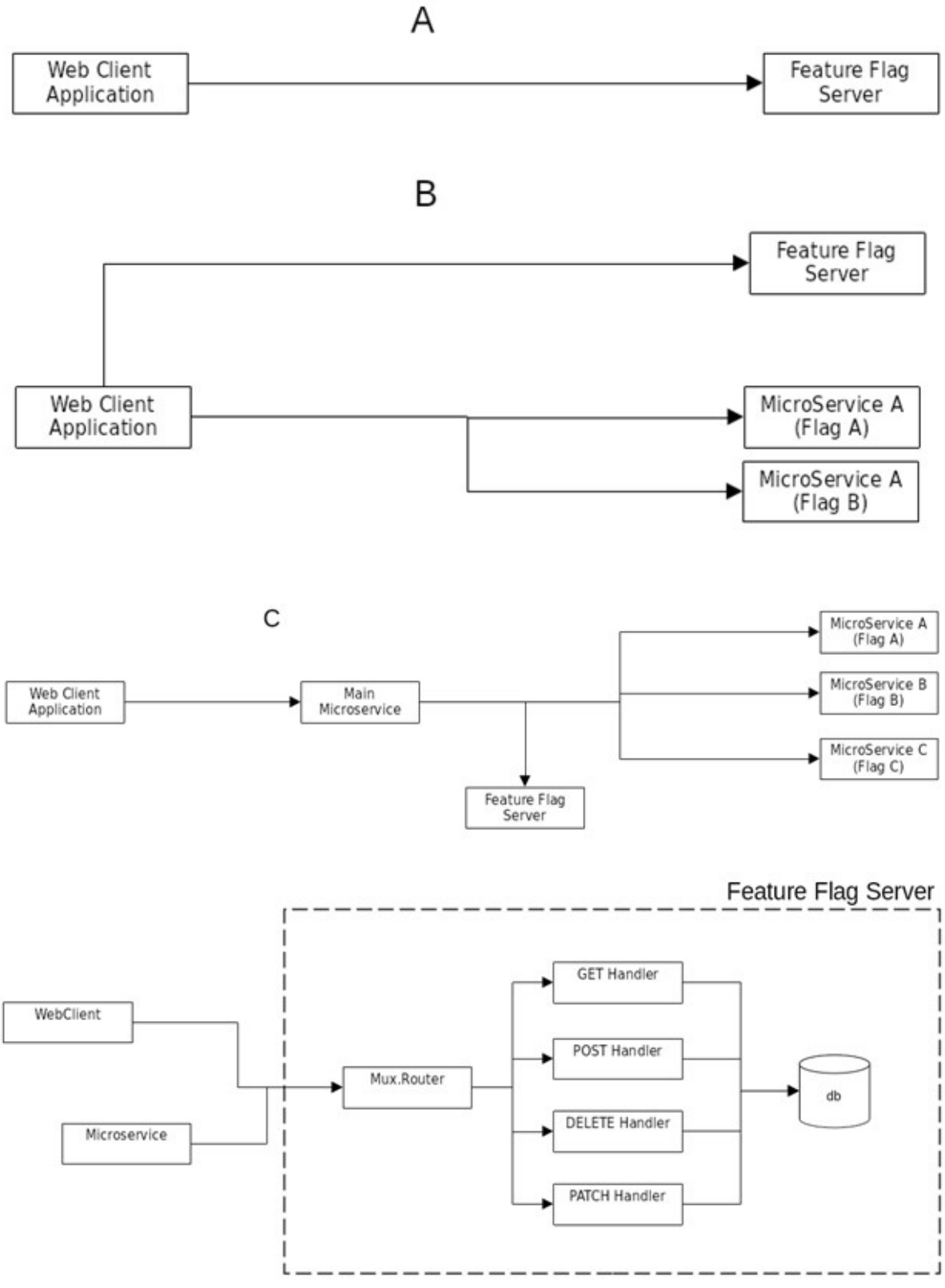
Login



Save Workout

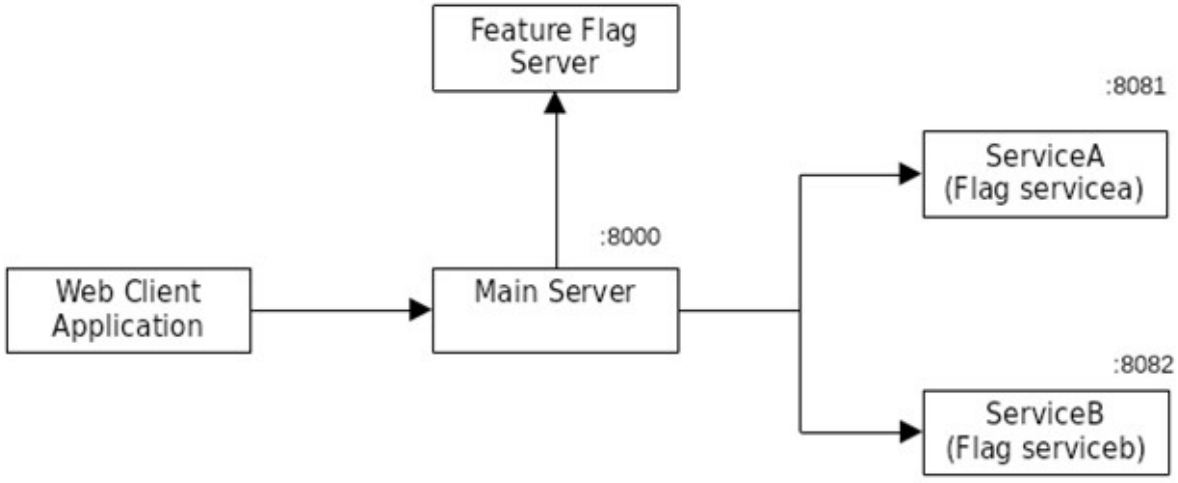


Chapter 11: Feature Flags

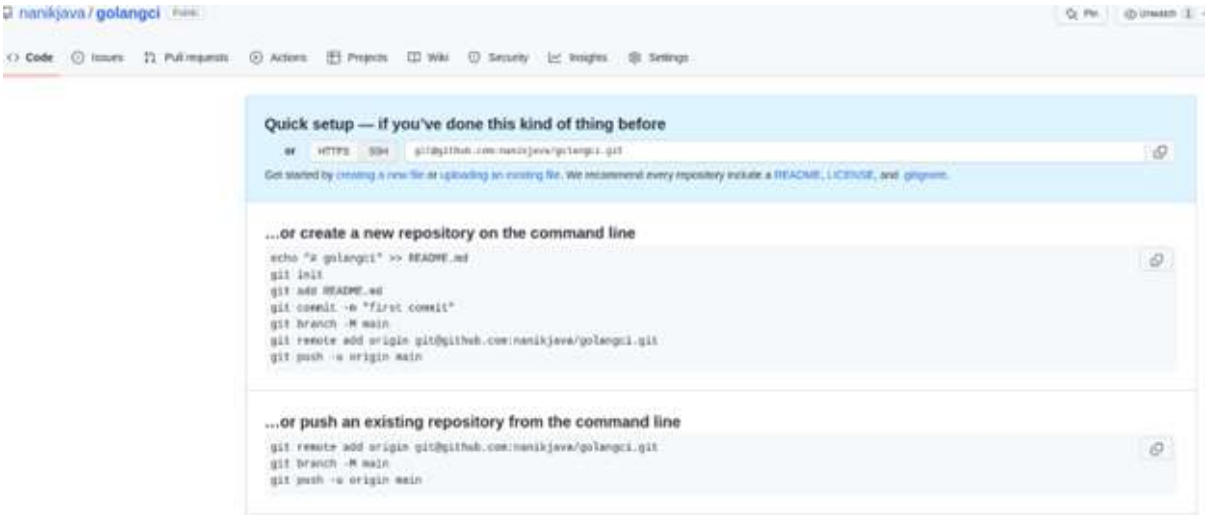


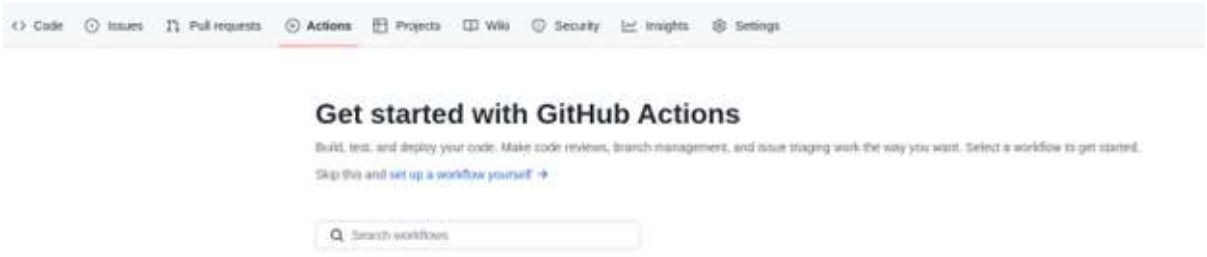

localhost:3000
Click to Post
You are in development mode
Your API is at http://0.0.0.0:8080/


localhost:3000
Click to Get
Click to Post


Feature Flag Server
:8081
ServiceA (Flag servicea)
:8000
Web Client Application
Main Server
:8082
ServiceB (Flag serviceb)

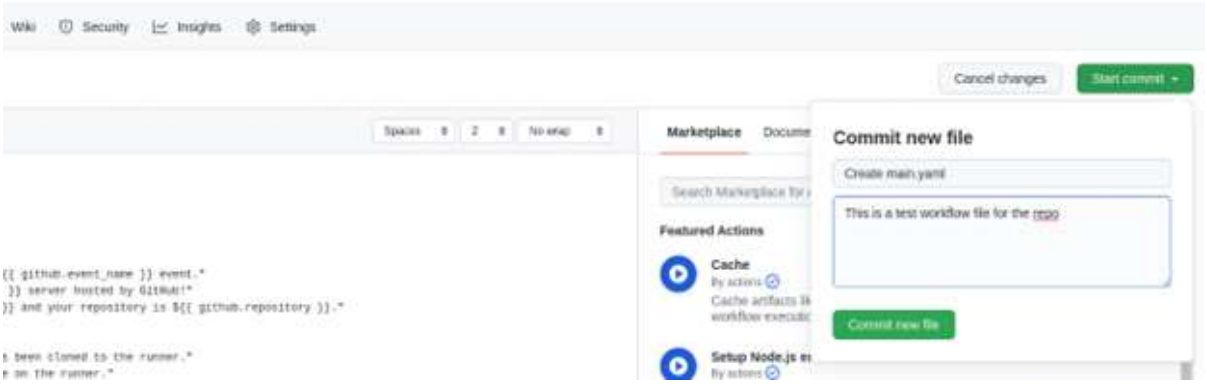
Chapter 12: Building Continuous Integration

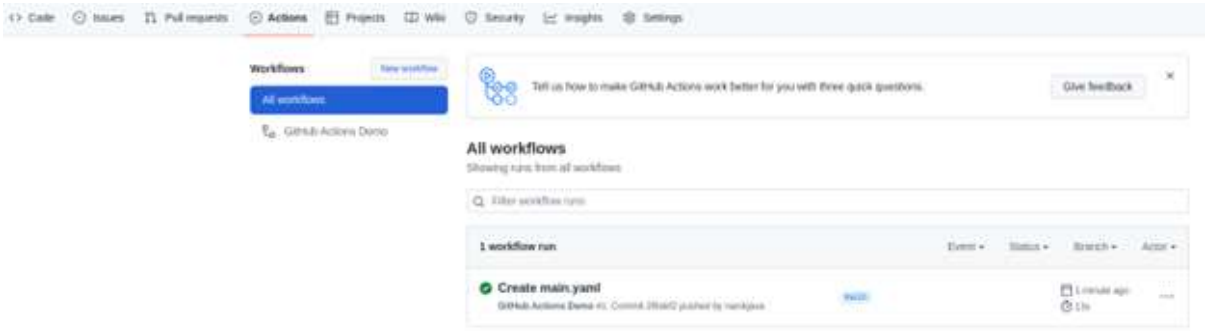


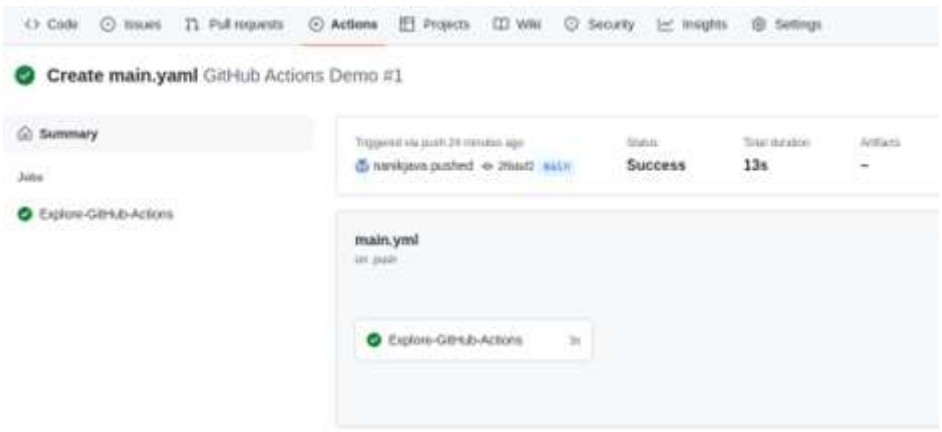


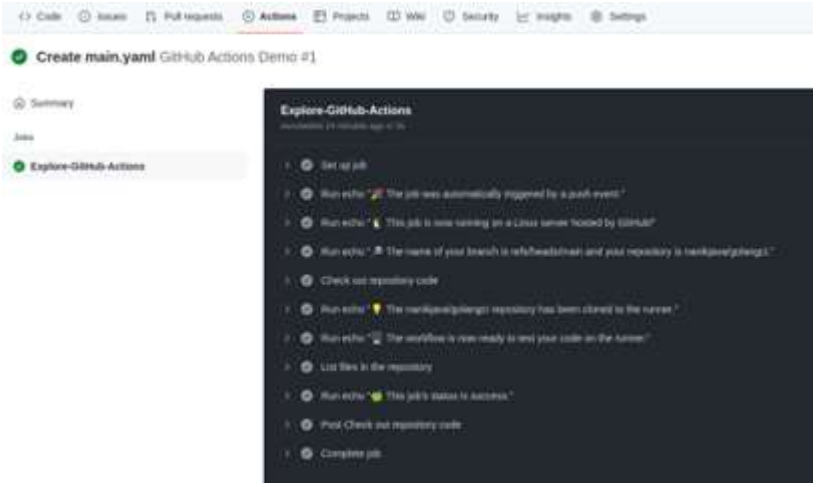


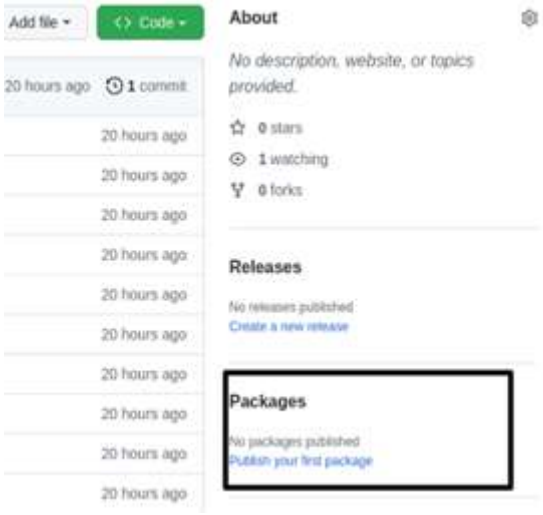

Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
golangci / .github / workflows / main.yml in main
Edit new file
Preview
Spaces
2
No wrap
name: GitHub Actions Demo
on: [push]
jobs:
Explore-GitHub-Actions:
runs-on: ubuntu-latest
steps:
- run: echo "🎉 The job was automatically triggered by a ${{ github.event_name }} event."
- run: echo "🐧 This job is now running on a ${{ runner.os }} server hosted by GitHub!"
- run: echo "🔎 The name of your branch is ${{ github.ref }} and your repository is ${{ github.repository }}."
- name: Check out repository code
uses: actions/checkout@v3
- run: echo "💡 The ${{ github.repository }} repository has been cloned to the runner."
- run: echo "🖥️ The workflow is now ready to test your code on the runner."
- name: List files in the repository
run: |
ls ${{ github.workspace }}
- run: echo "🍏 This job's status is ${{ job.status }}."


Wiki
Security
Insights
Settings
Cancel changes
Start commit
Marketplace
Commit new file
Create main.yaml
This is a test workflow file for the repo
Commit new file
Featured Actions
Cache
Setup Node.js


Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Workflows
All workflows
Give feedback
All workflows
1 workflow run
Create main.yaml


Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Create main.yaml GitHub Actions Demo #1
Summary
Jobs
Explore-GitHub-Actions
Status
Success
Total duration
13s
Artifacts
–
main.yml
Explore-GitHub-Actions



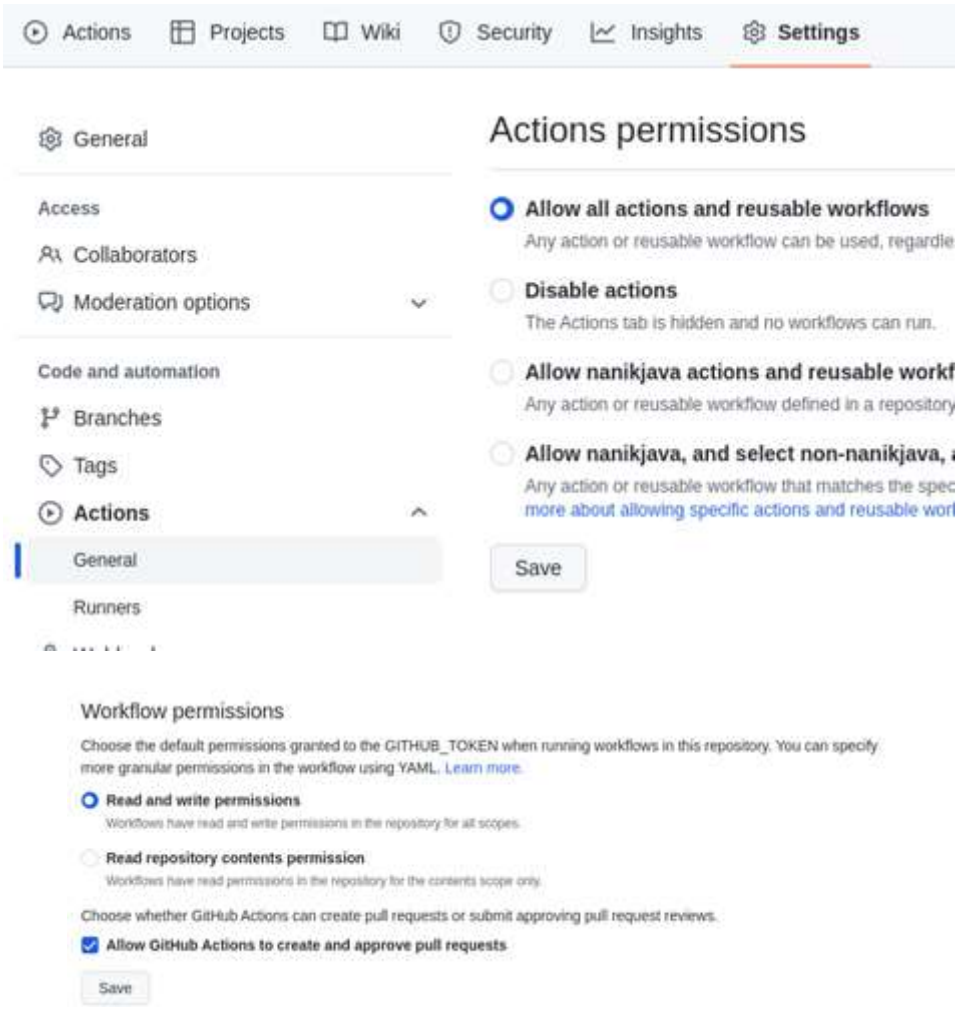


Clear current search query, filters, and sorts

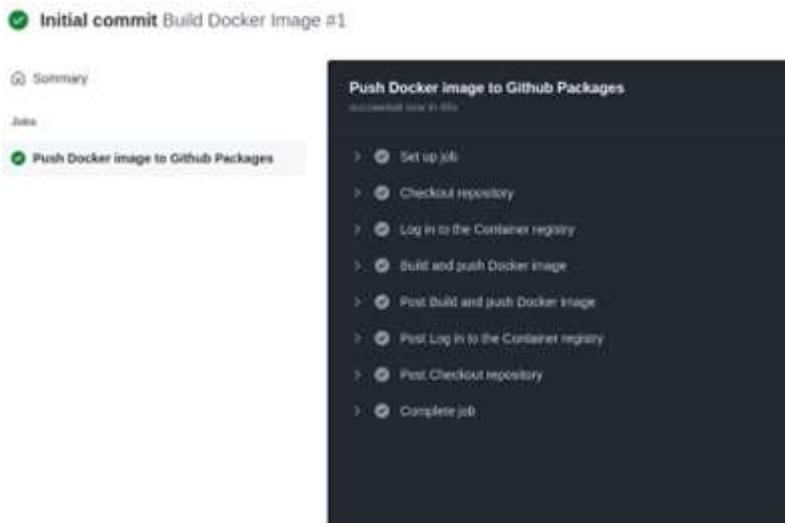
0 packages

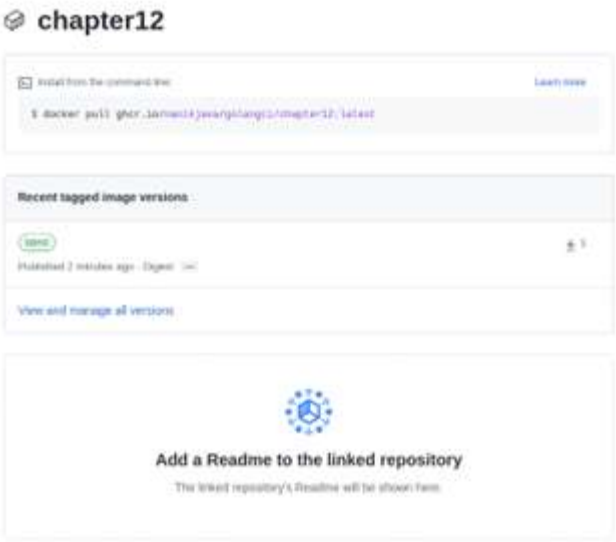
No results matched your search.

Try browsing all packages to find what you're looking for.


Actions
Projects
Wiki
Security
Insights
Settings
General
Access
Collaborators
Moderation options
Code and automation
Branches
Tags
Actions
General
Runners
Actions permissions
Allow all actions and reusable workflows
Any action or reusable workflow can be used, regardle
Disable actions
The Actions tab is hidden and no workflows can run.
Allow nanikjava actions and reusable workf
Any action or reusable workflow defined in a repository
Allow nanikjava, and select non-nanikjava,
Any action or reusable workflow that matches the spec
more about allowing specific actions and reusable wor
Save

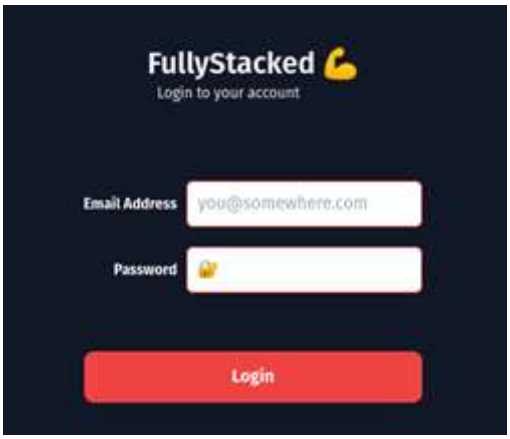
Workflow permissions
Choose the default permissions granted to the GITHUB_TOKEN when running workflows in this repository. You can specify more granular permissions in the workflow using YAML. Learn more.
Read and write permissions
Workflows have read and write permissions in the repository for all scopes.
Read repository contents permission
Workflows have read permissions in the repository for the contents scope only.
Choose whether GitHub Actions can create pull requests or submit approving pull request reviews.
Allow GitHub Actions to create and approve pull requests
Save


Initial commit Build Docker Image #1
Summary
Jobs
Push Docker image to Github Packages
Push Docker image to Github Packages
Set up job
Checkout repository
Log in to the Container registry
Build and push Docker image
Post Build and push Docker image
Post Log in to the Container registry
Post Checkout repository
Complete job

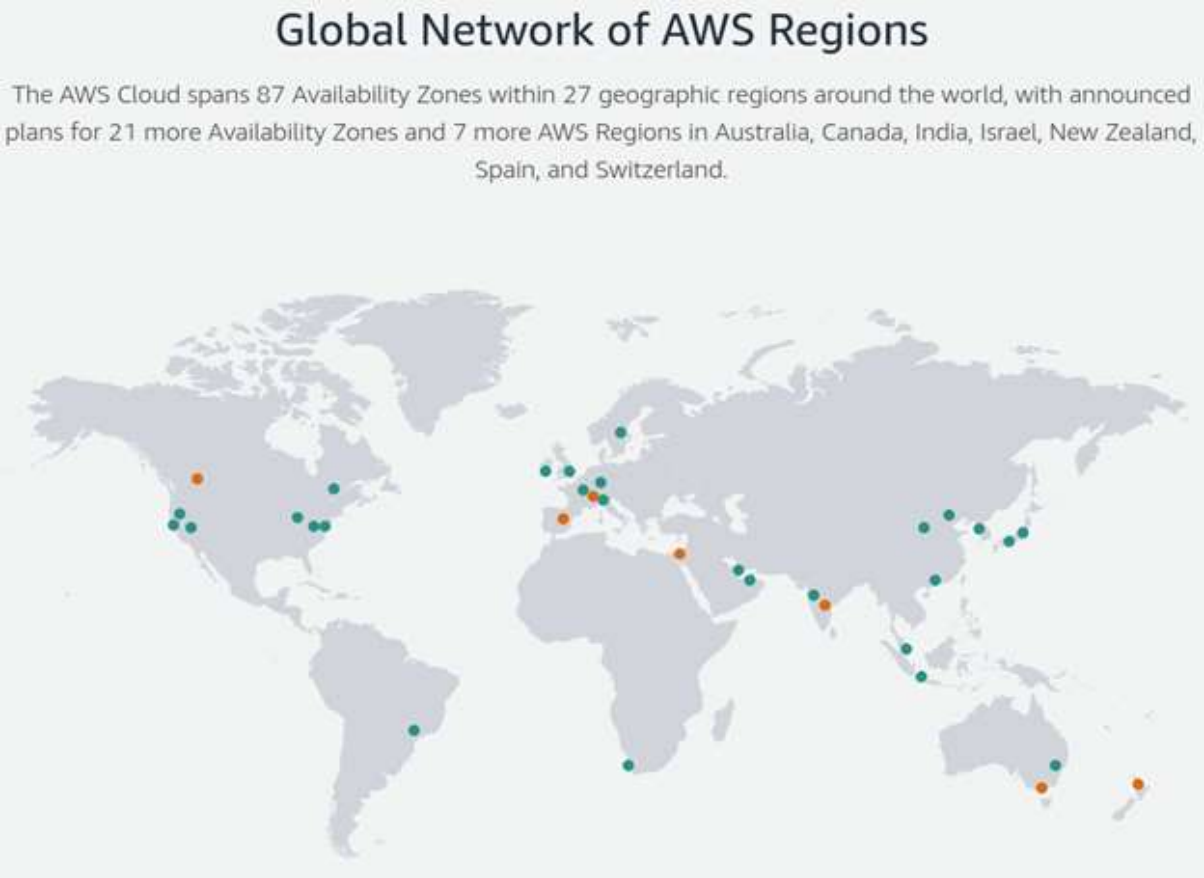

chapter12
Recent tagged image versions
latest
View and manage all versions
Add a Readme to the linked repository

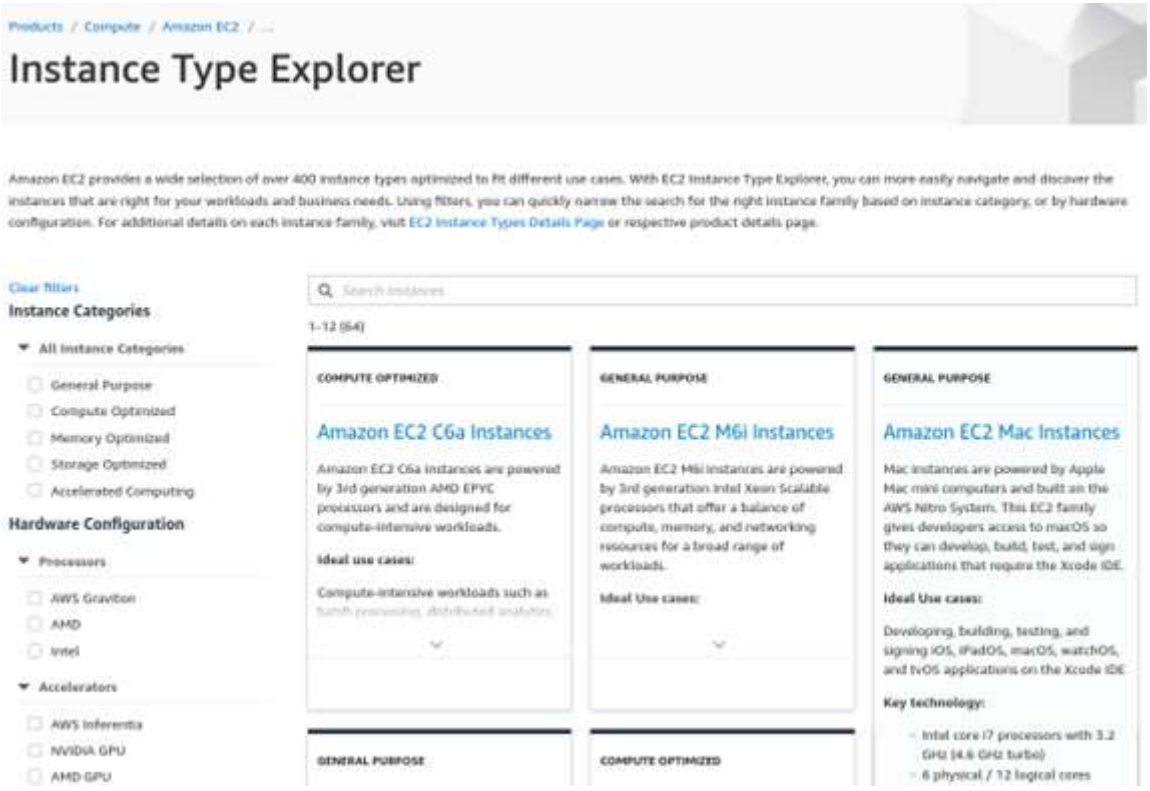
Chapter 13: Dockerizing an Application

```
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
redis        latest    bba24acba395   4 weeks ago    113MB
postgres     latest    1ee973e26c65   4 weeks ago    376MB
```



Chapter 14: Cloud Deployment

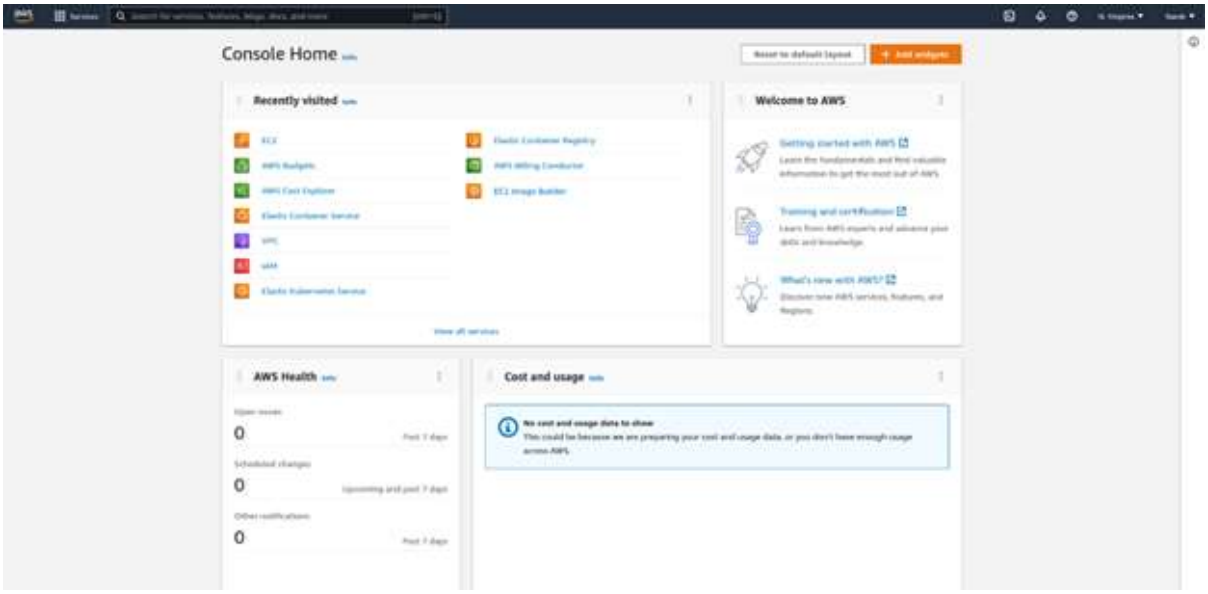





Amazon Elastic Block Store
SSD-based
General Purpose
Provisioned IOPS
HDD-based
Throughput Optimized
Cold
Select your volume type
Select the right volume type based on your application's needs and cost
Attach to your Amazon EC2 Instance
Attach your new Amazon EBS volume(s) to either new or existing EC2 instances. Multi-Attach supported for Linux on the AWS Nitro System
Run your application
Run your applications with persistent block volumes that offer 99.999% availability


Region 1
VPC Subnet in AZ 1
Amazon EC2 Instance
Amazon EC2 Instance
VPC Subnet in AZ 2
Amazon EC2 Instance
Amazon EC2 Instance
Region 2
VPC Subnet in AZ 1
Amazon EC2 Instance
Amazon EC2 Instance
VPC Subnet in AZ 2
Amazon EC2 Instance
Amazon EC2 Instance


Connect
Amazon Relational Database Service
Amazon RDS managed features

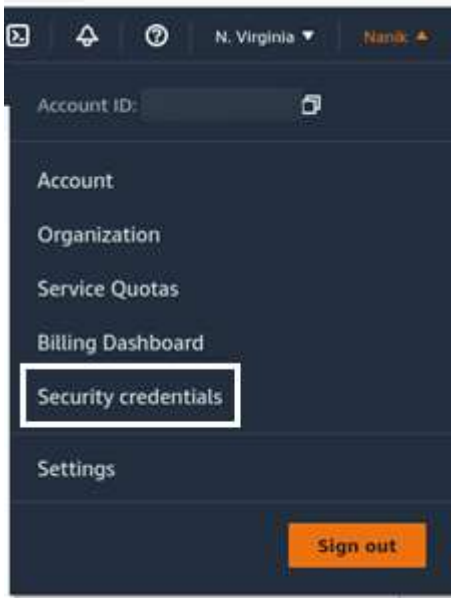


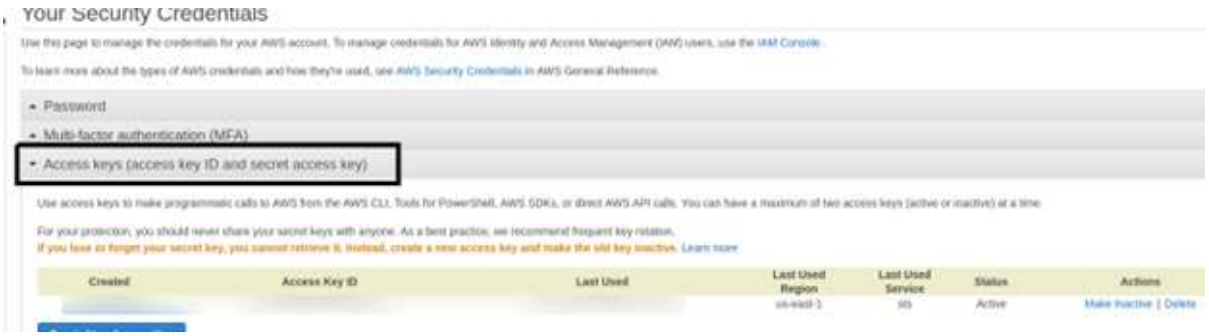
Welcome to nginx!

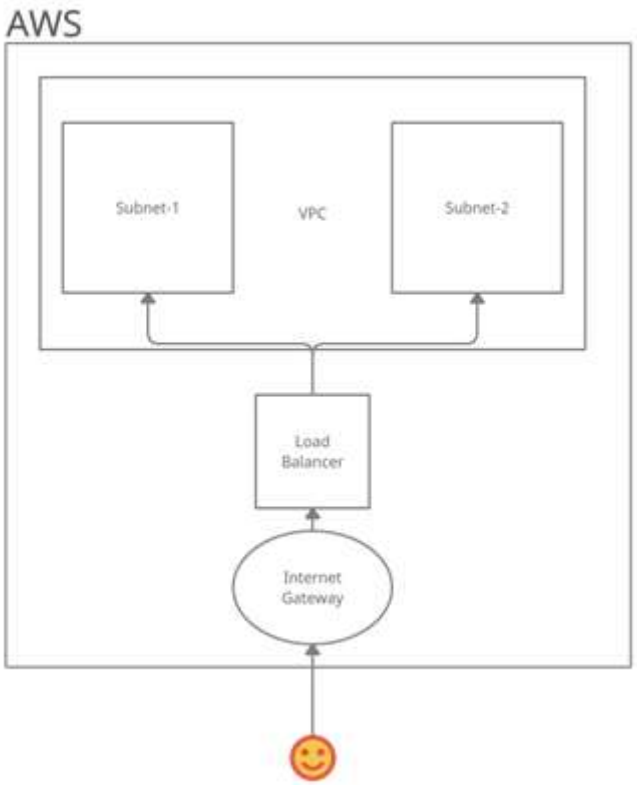
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.




Your Security Credentials
Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.
To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.
Password
Multi-factor authentication (MFA)
Access keys (access key ID and secret access key)
Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time.
For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation.
If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key and make the old key inactive. Learn more
Created
Access Key ID
Last Used
Last Used Region
Last Used Service
Status
Actions
us-east-1
ses
Active
Make Inactive | Delete


AWS
Subnet-1
VPC
Subnet-2
Load Balancer
Internet Gateway