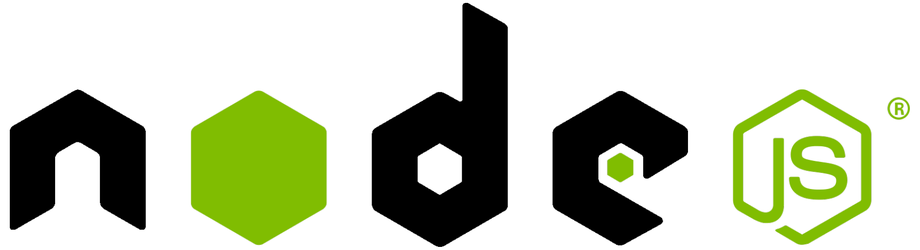
JAVASCRIPT

BACK-END(API WITH DATABASE)

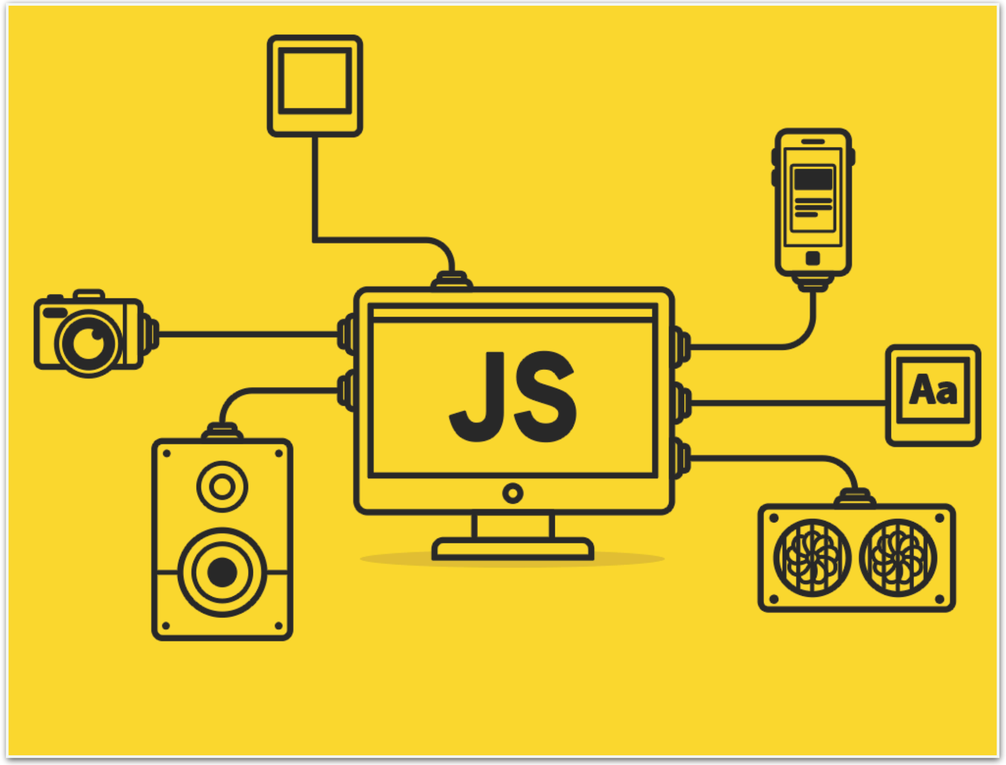
Anirach Mingkhwan

Anirach.m@fitm.kmutnb.ac.th

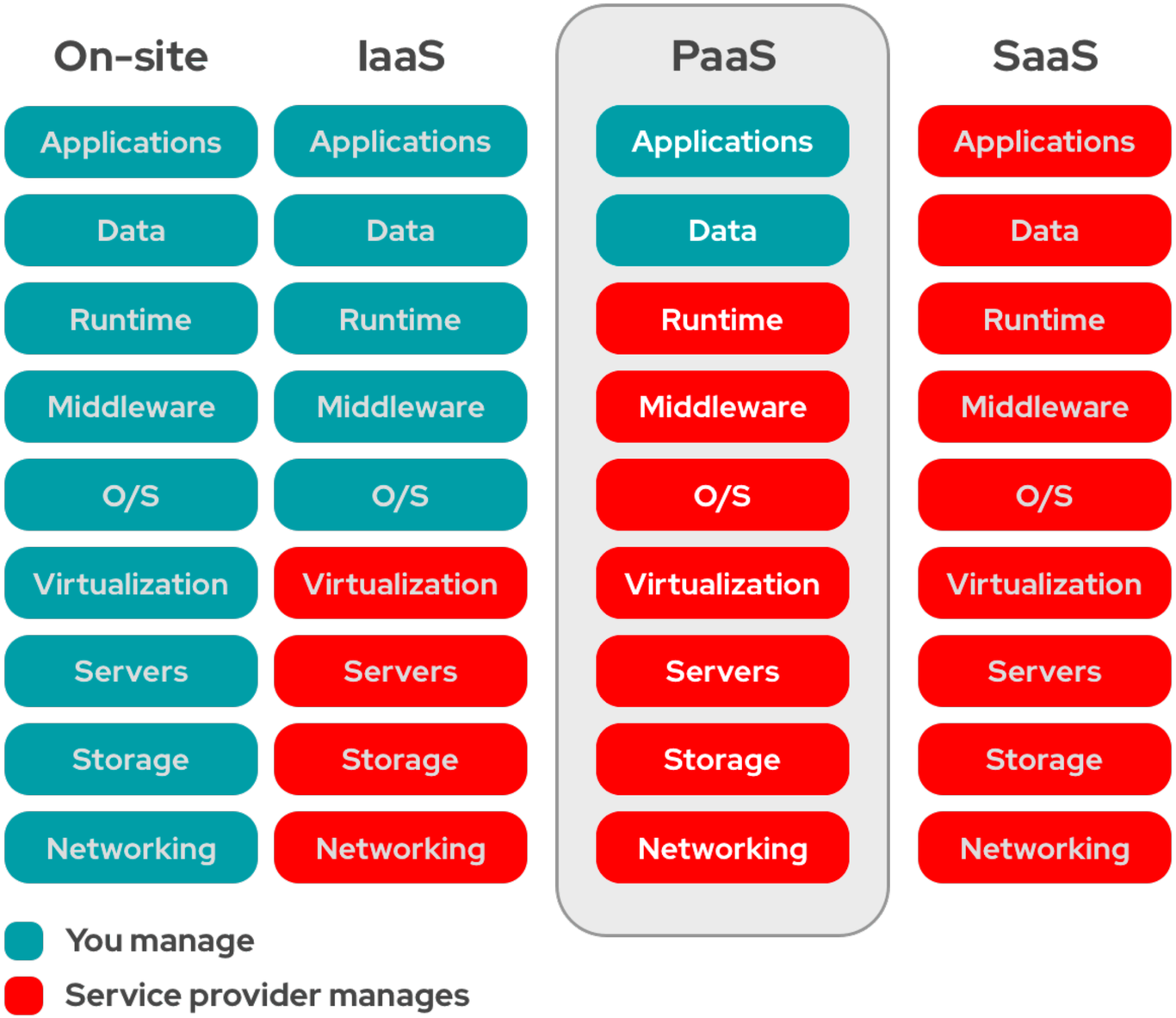


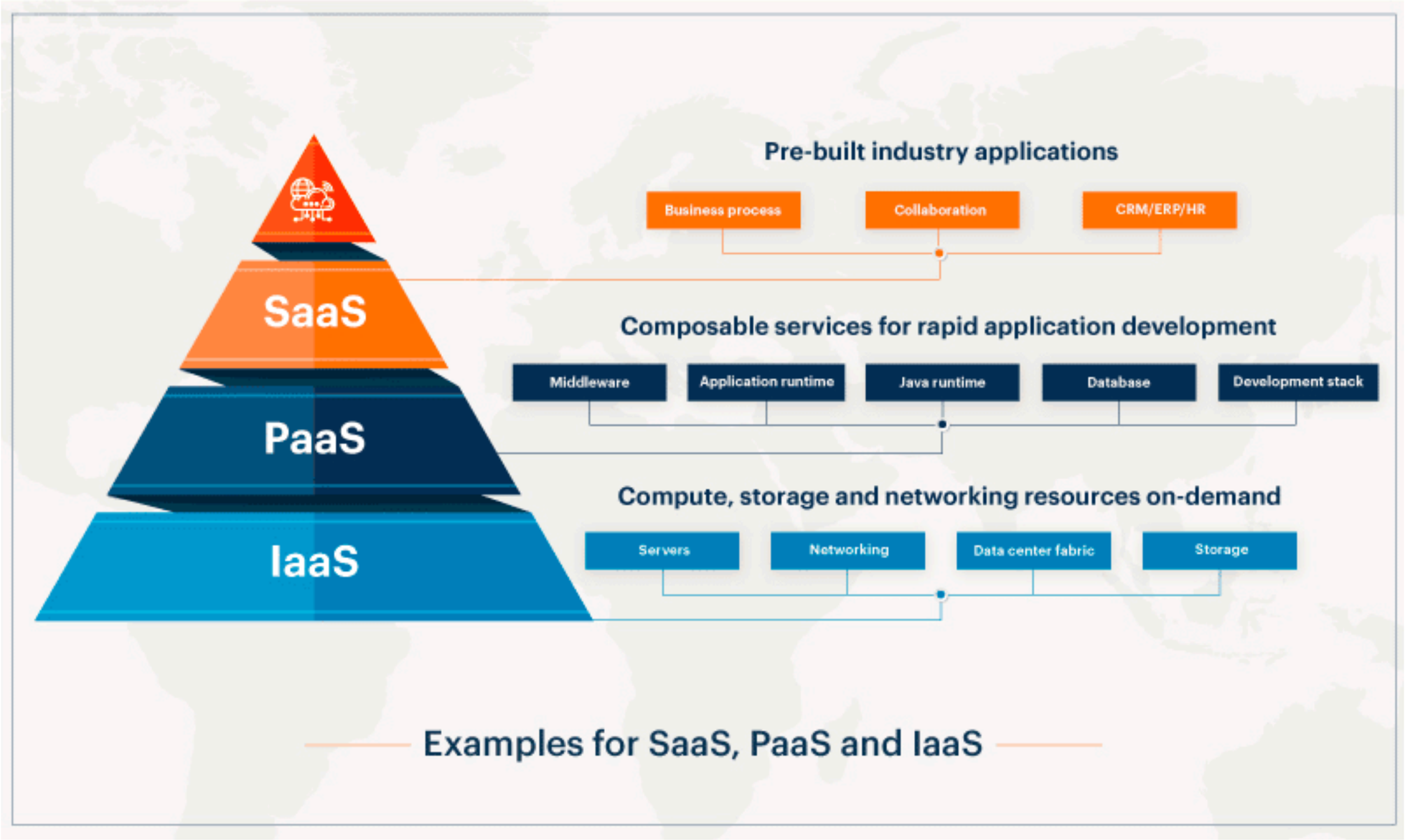
OUTLINE

- PaaS
- Cloud Setting
- Local Database
- SQLite
- SQL command
- API with SQLite DB



CLOUD PAAS


On-site
IaaS
PaaS
SaaS
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
You manage
Service provider manages

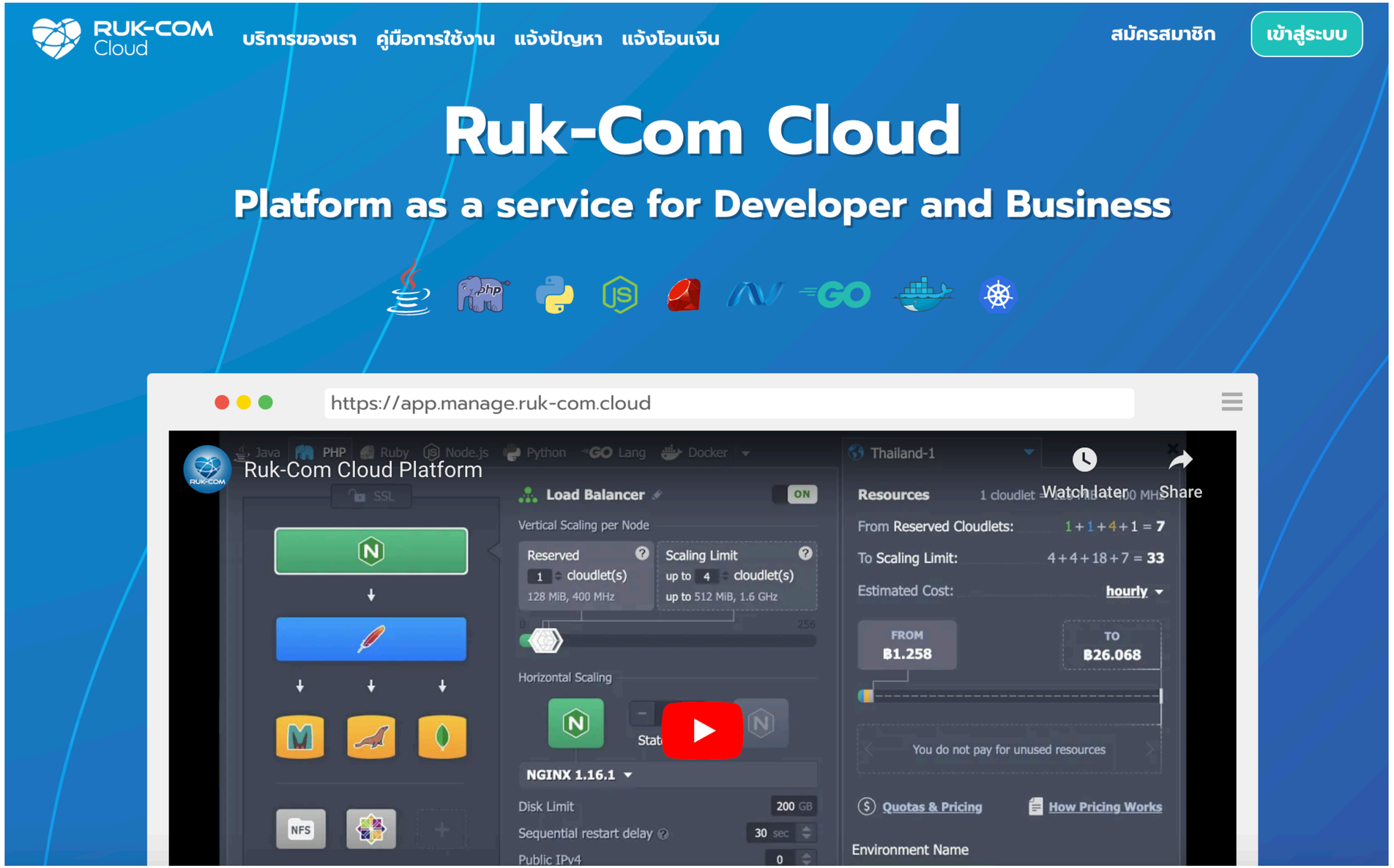


Examples for SaaS, PaaS and IaaS

SETUP CLOUD

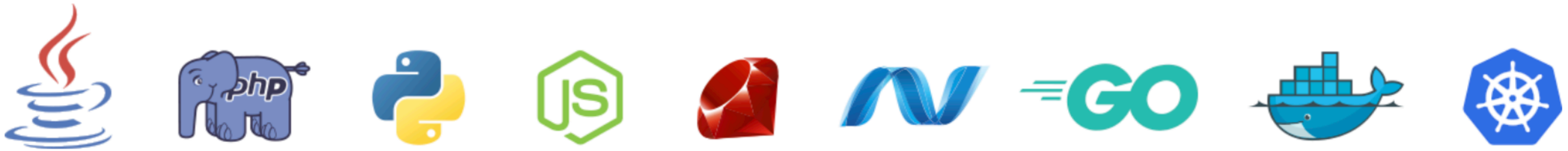
RUK-COM Cloud

https://app.manage.ruk-com.cloud

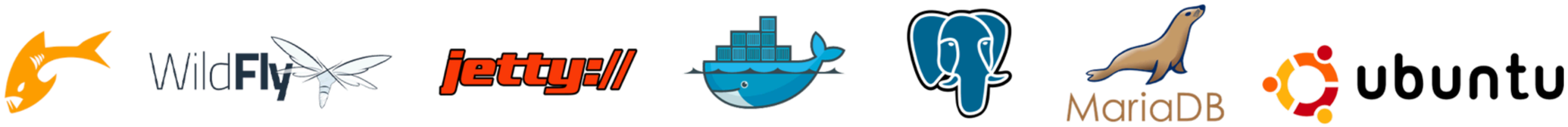


https://paas.ruk-com.cloud/

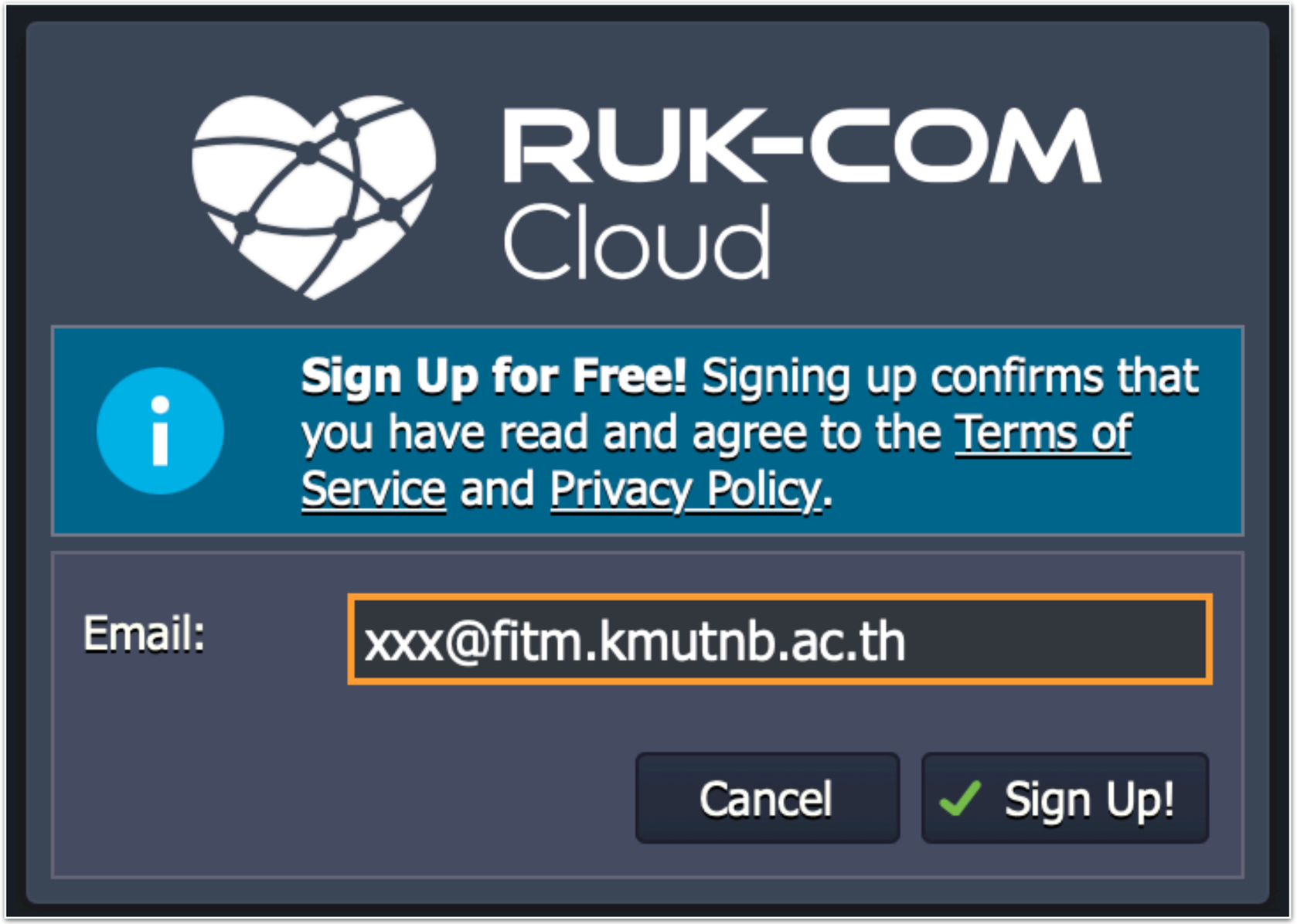
Popular Stacks & Services Out-of-Box Support



New releases and security updates automation, performance and scaling optimization



https://app.manage.ruk-com.cloud/?signup=true&group=kmutnb_students


RUK-COM
Cloud
Sign Up for Free! Signing up confirms that you have read and agree to the Terms of Service and Privacy Policy.
Email:
xxx@fitm.kmutnb.ac.th
Cancel
Sign Up!

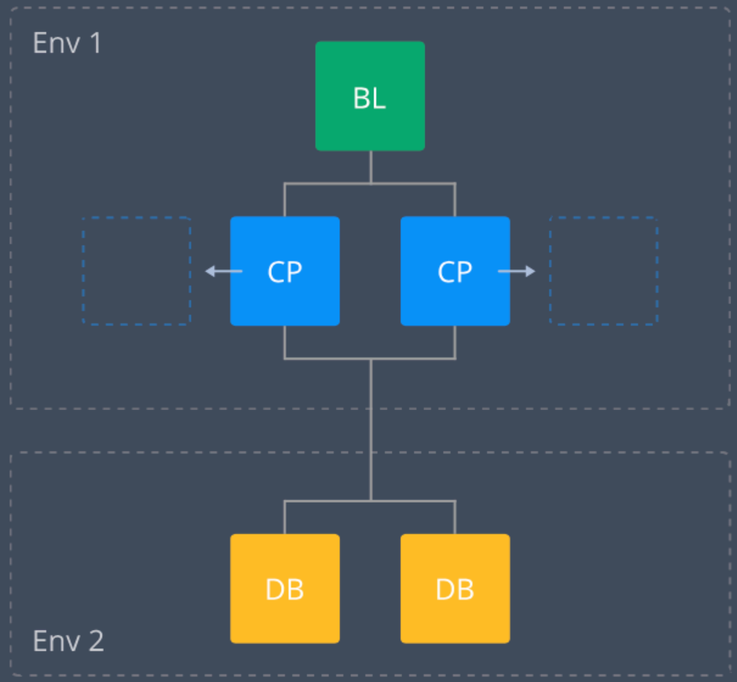
Build Your Application Topology

Container

Layer

Environment

Application

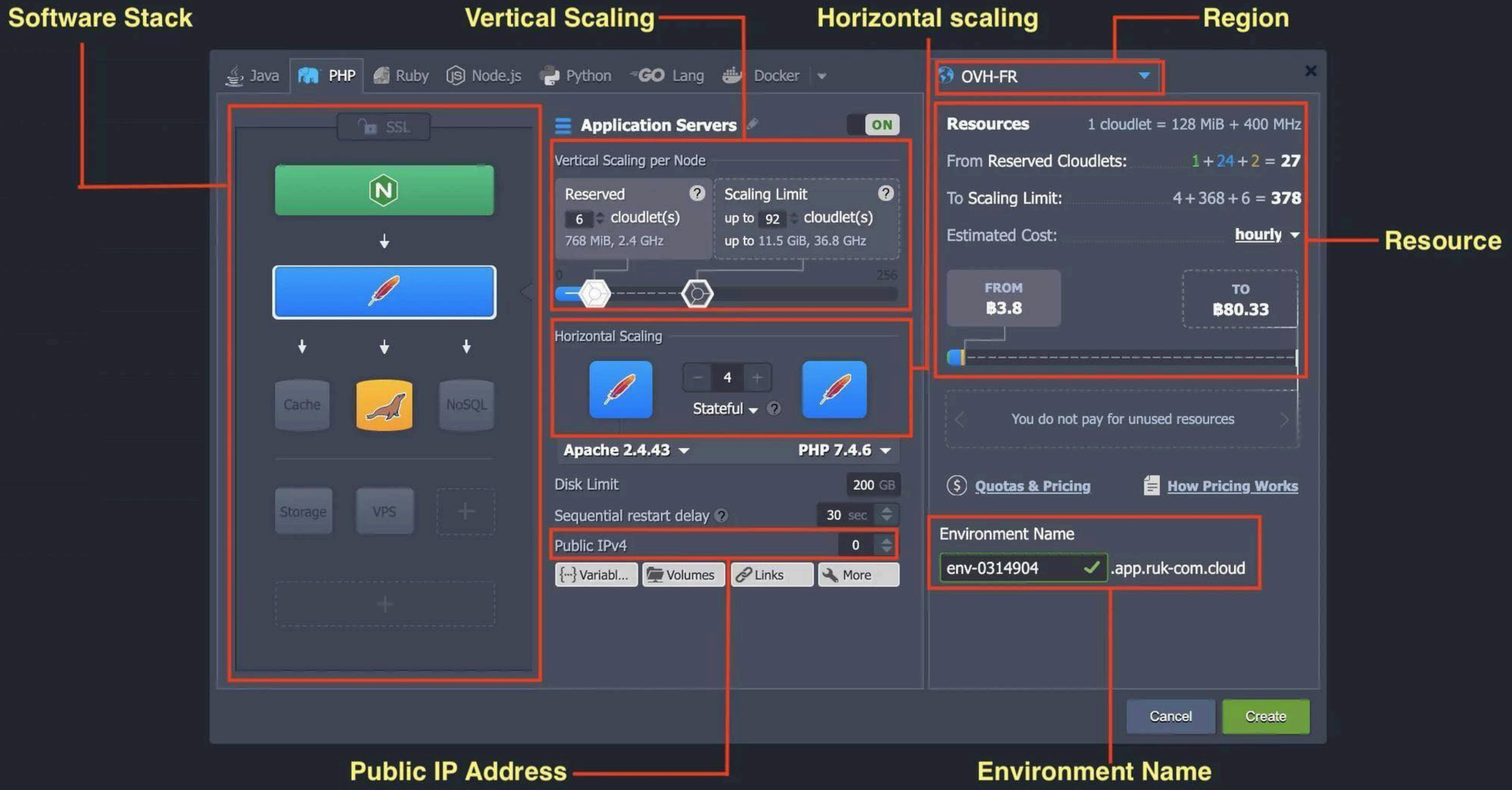


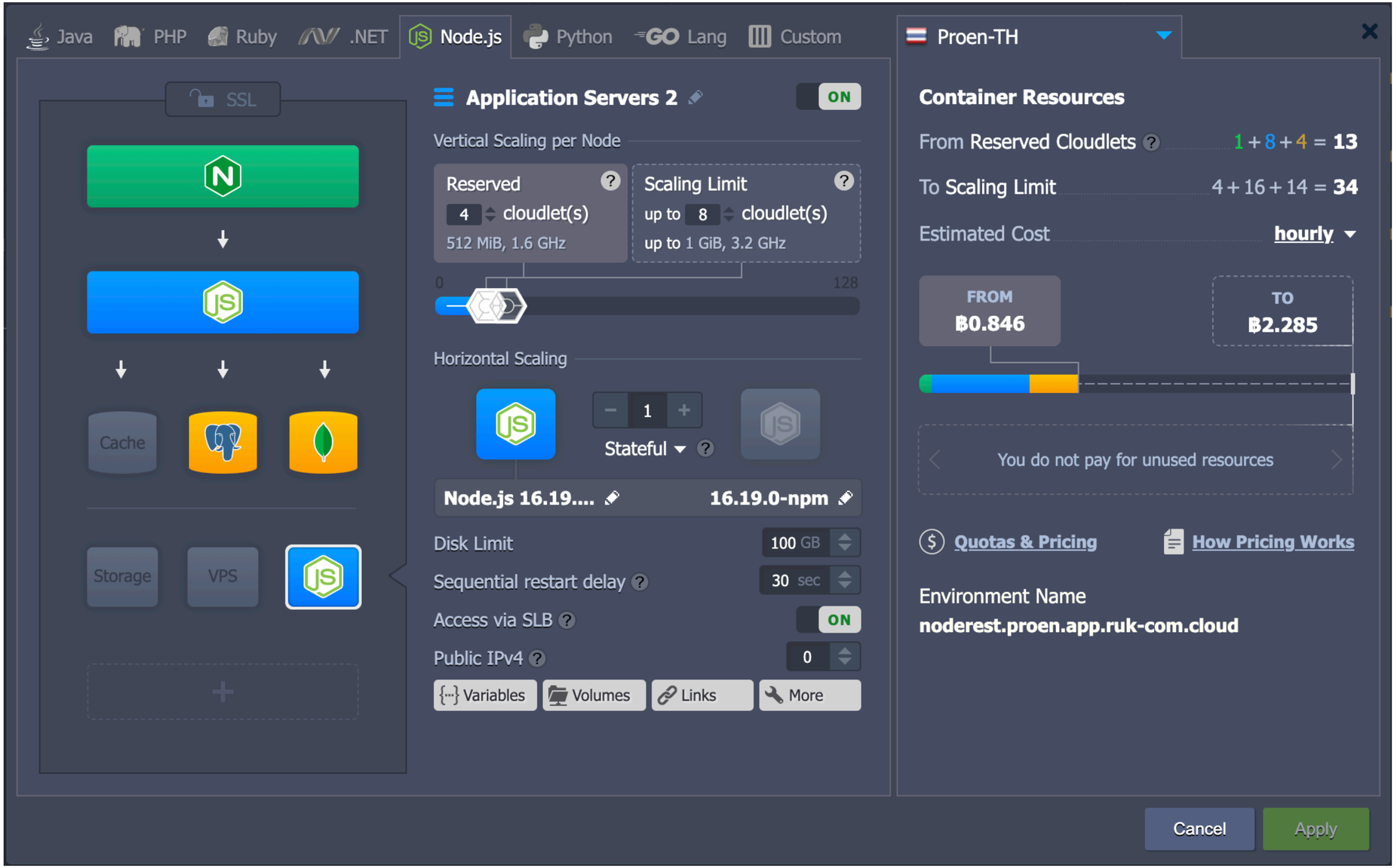
Balancer (BL)

Compute (CP)

Database (DB)

Storage (ST)

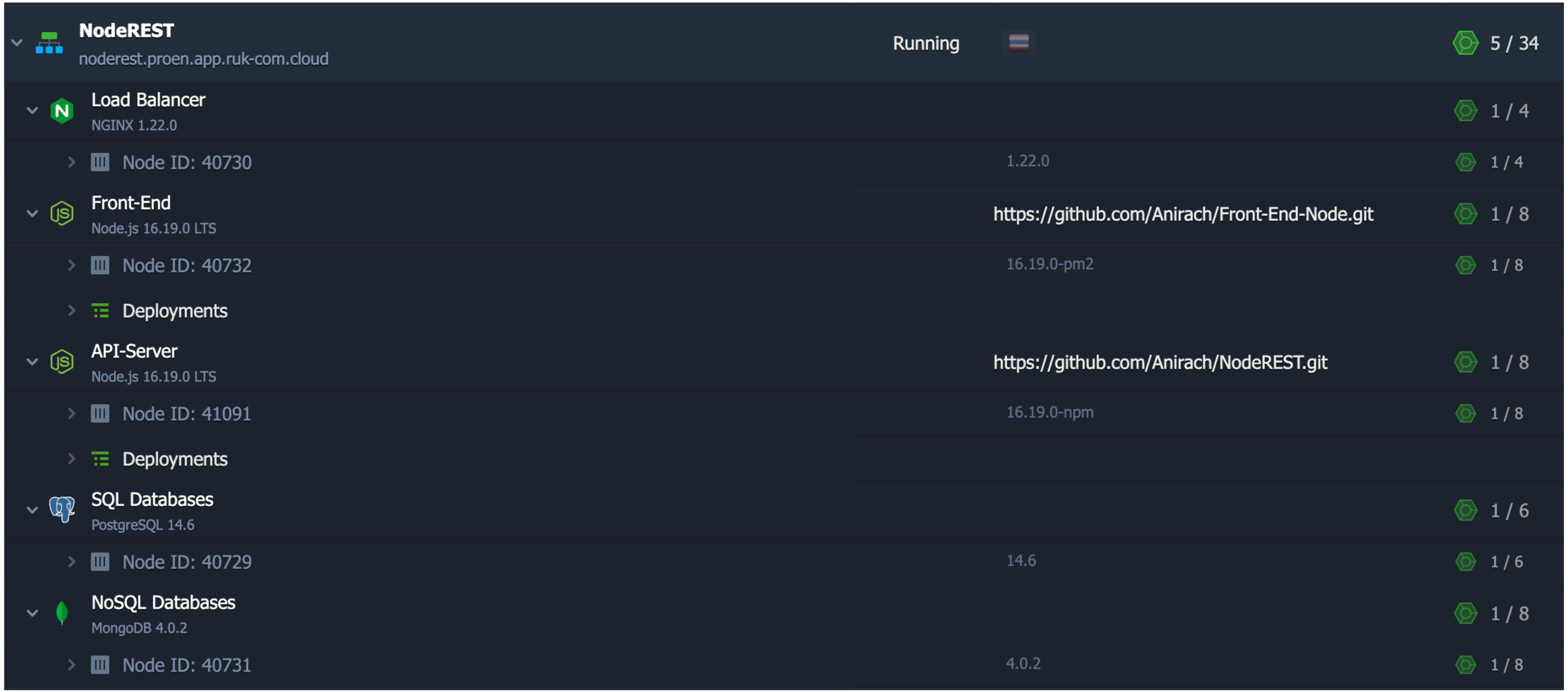

Software Stack
Vertical Scaling
Horizontal scaling
Region
Resource
Public IP Address
Environment Name
Java
PHP
Ruby
Node.js
Python
Lang
Docker
SSL
Cache
NoSQL
Storage
VPS
Application Servers
ON
Vertical Scaling per Node
Reserved
6 cloudlet(s)
768 MiB, 2.4 GHz
Scaling Limit
up to 92 cloudlet(s)
up to 11.5 GiB, 36.8 GHz
0
256
Horizontal Scaling
4
Stateful
Apache 2.4.43
PHP 7.4.6
Disk Limit
200 GB
Sequential restart delay
30 sec
Public IPv4
0
Variabl...
Volumes
Links
More
OVH-FR
Resources
1 cloudlet = 128 MiB + 400 MHz
From Reserved Cloudlets:
1 + 24 + 2 = 27
To Scaling Limit:
4 + 368 + 6 = 378
Estimated Cost:
hourly
FROM
฿3.8
TO
฿80.33
You do not pay for unused resources
Quotas & Pricing
How Pricing Works
Environment Name
env-0314904
.app.ruk-com.cloud
Cancel
Create

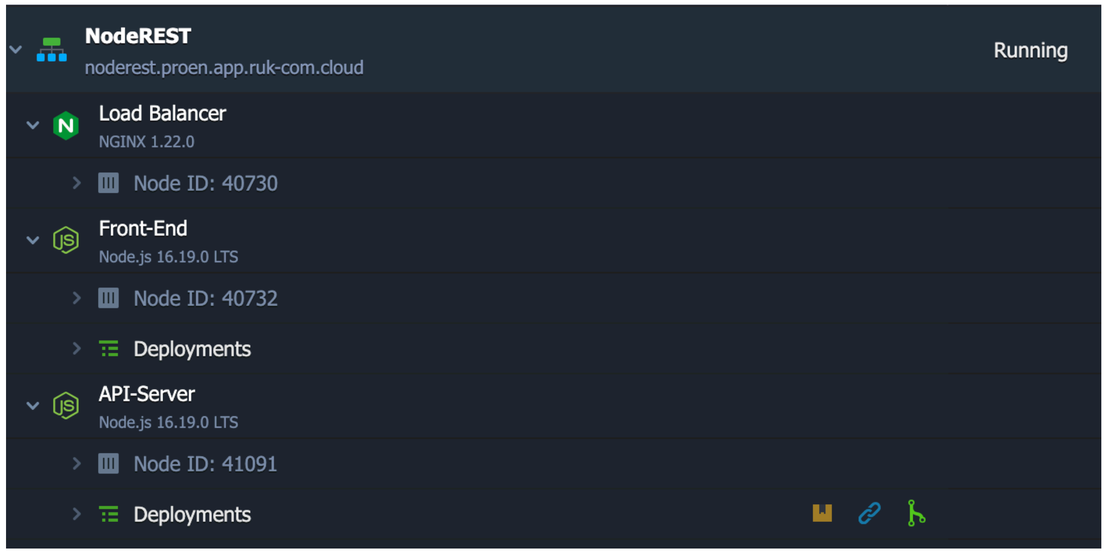

Java
PHP
Ruby
.NET
Node.js
Python
Lang
Custom
SSL
Cache
Storage
VPS
Application Servers 2
ON
Vertical Scaling per Node
Reserved
4 cloudlet(s)
512 MiB, 1.6 GHz
Scaling Limit
up to 8 cloudlet(s)
up to 1 GiB, 3.2 GHz
0
128
Horizontal Scaling
1
Stateful
Node.js 16.19....
16.19.0-npm
Disk Limit
100 GB
Sequential restart delay
30 sec
Access via SLB
ON
Public IPv4
0
Variables
Volumes
Links
More
Proen-TH
Container Resources
From Reserved Cloudlets
1 + 8 + 4 = 13
To Scaling Limit
4 + 16 + 14 = 34
Estimated Cost
hourly
FROM
฿0.846
TO
฿2.285
You do not pay for unused resources
Quotas & Pricing
How Pricing Works
Environment Name
noderest.proen.app.ruk-com.cloud
Cancel
Apply

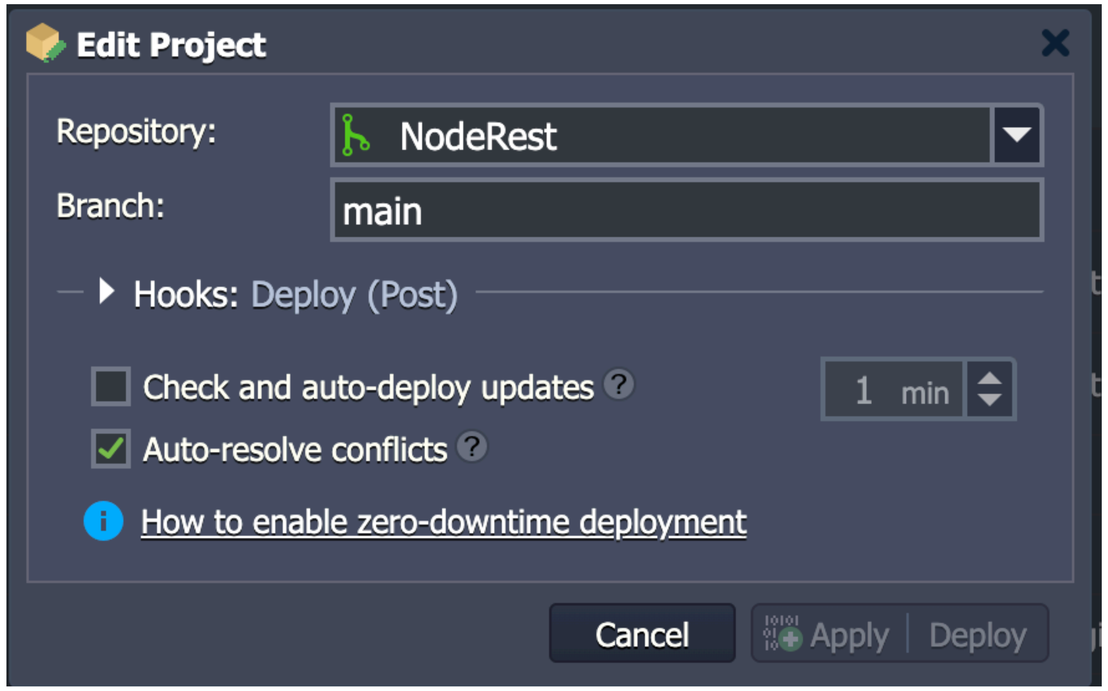
CONTAINERS	FROM Reserved	FROM Price	TO Scaling Limit	TO Price
Load Balancer		฿0.069		฿0.272
└ Cloudlets (RAM, CPU)	1	฿0.069	4	฿0.272
Application Servers		฿0.274		฿0.537
└ Cloudlets (RAM, CPU)	4	฿0.274	8	฿0.537
SQL Databases		฿0.206		฿0.403
└ Cloudlets (RAM, CPU)	3	฿0.206	6	฿0.403
NoSQL Databases		฿0.069		฿0.548
└ Cloudlets (RAM, CPU)	1	฿0.069	8	฿0.548
Application Servers 2		฿0.274		฿0.537
└ Cloudlets (RAM, CPU)	4	฿0.274	8	฿0.537
Total		~~฿0.892~~		~~฿2.297~~
Save		฿0.046		฿0.012
Pay Hourly ▼		฿0.846		฿2.285

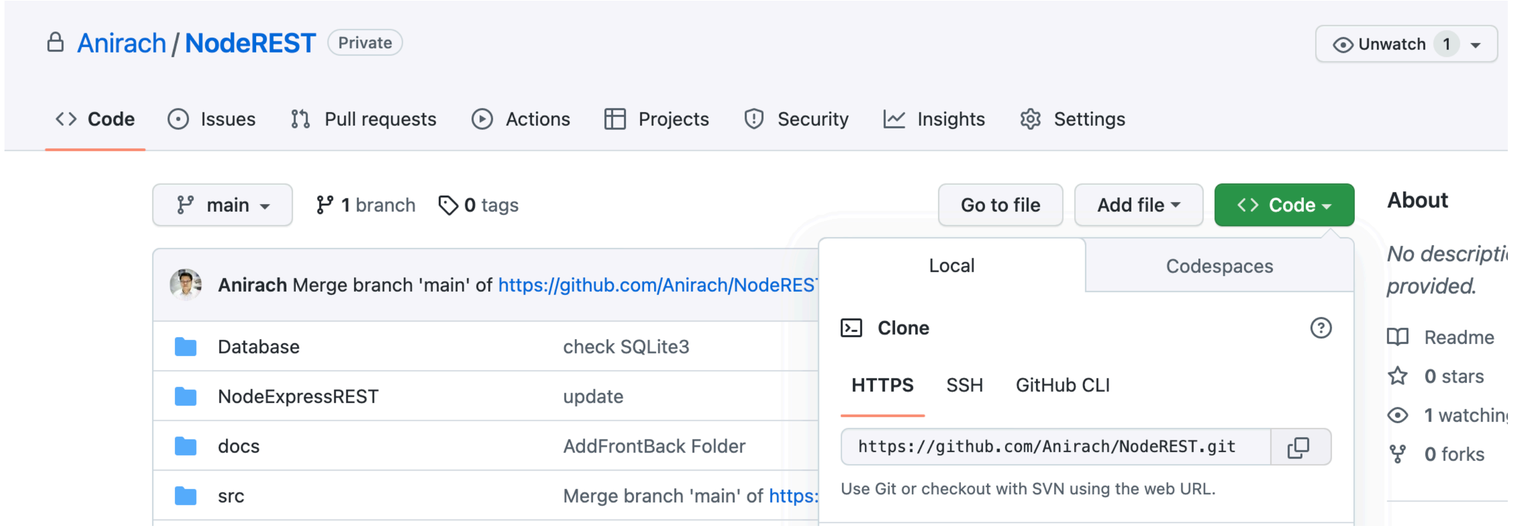
How to track your spends?

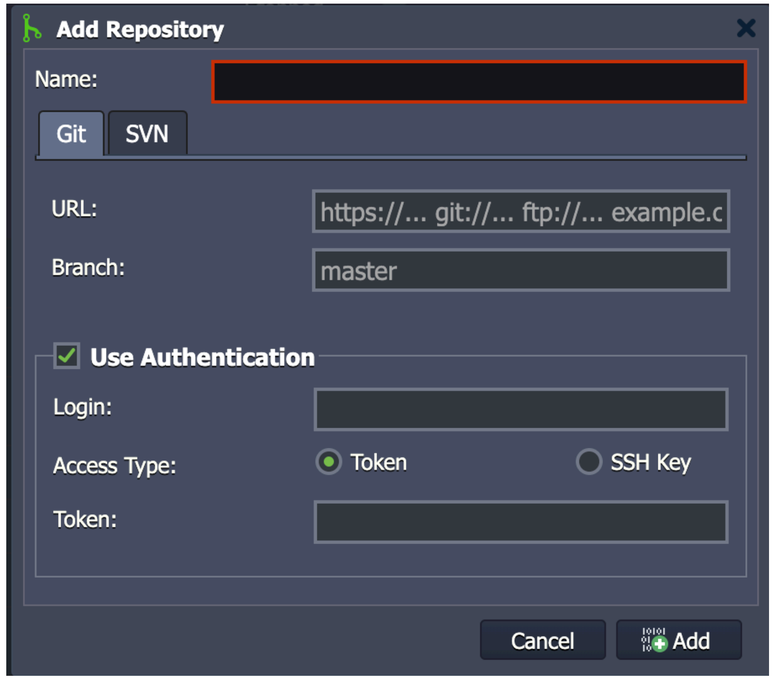
Disk Space and Traffic are charged based on consumption.

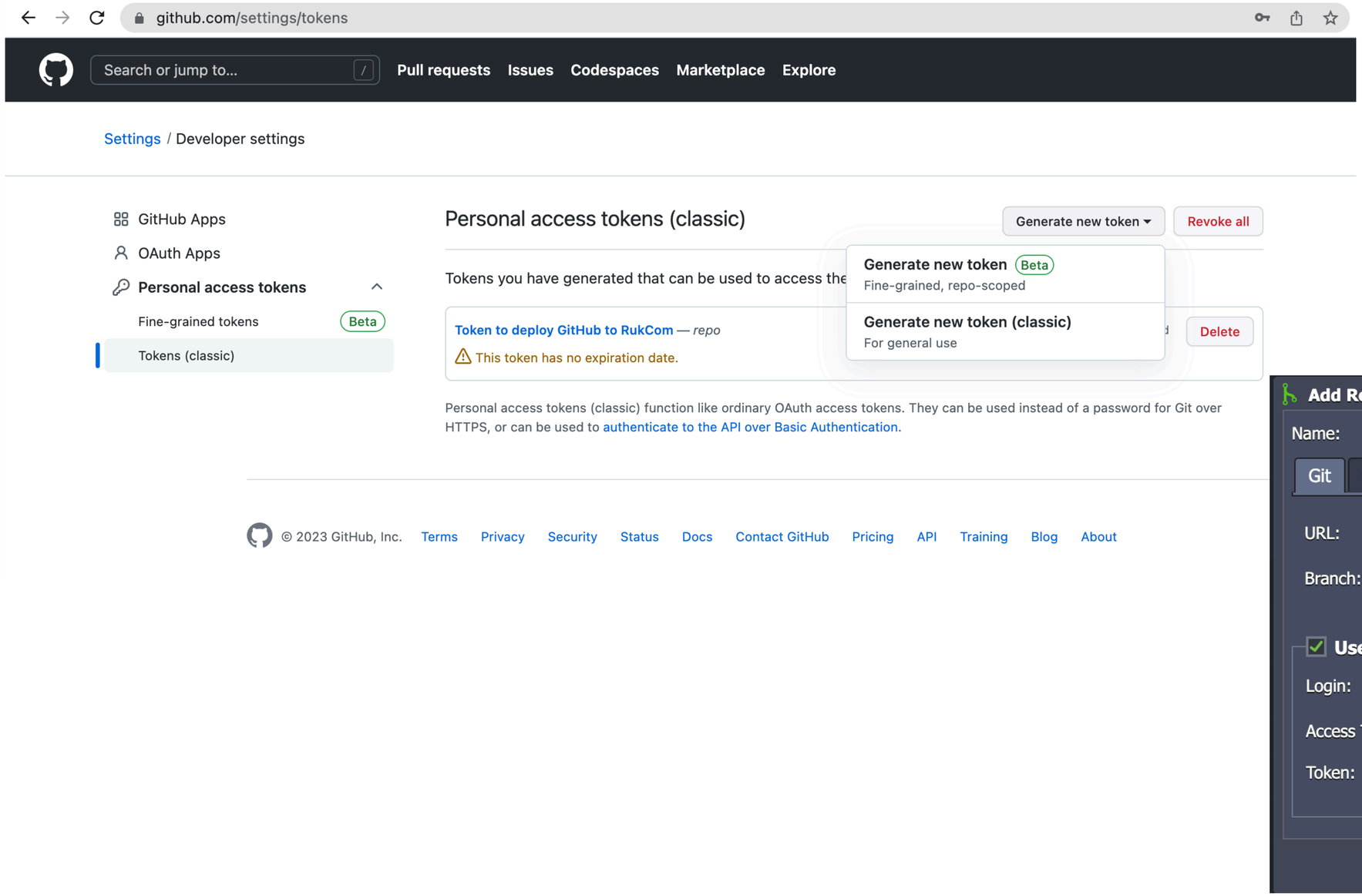

NodeREST
noderest.proen.app.ruk-com.cloud
Running
5 / 34
Load Balancer
NGINX 1.22.0
1 / 4
Node ID: 40730
1.22.0
1 / 4
Front-End
Node.js 16.19.0 LTS
https://github.com/Anirach/Front-End-Node.git
1 / 8
Node ID: 40732
16.19.0-pm2
1 / 8
Deployments
API-Server
Node.js 16.19.0 LTS
https://github.com/Anirach/NodeREST.git
1 / 8
Node ID: 41091
16.19.0-npm
1 / 8
Deployments
SQL Databases
PostgreSQL 14.6
1 / 6
Node ID: 40729
14.6
1 / 6
NoSQL Databases
MongoDB 4.0.2
1 / 8
Node ID: 40731
4.0.2
1 / 8

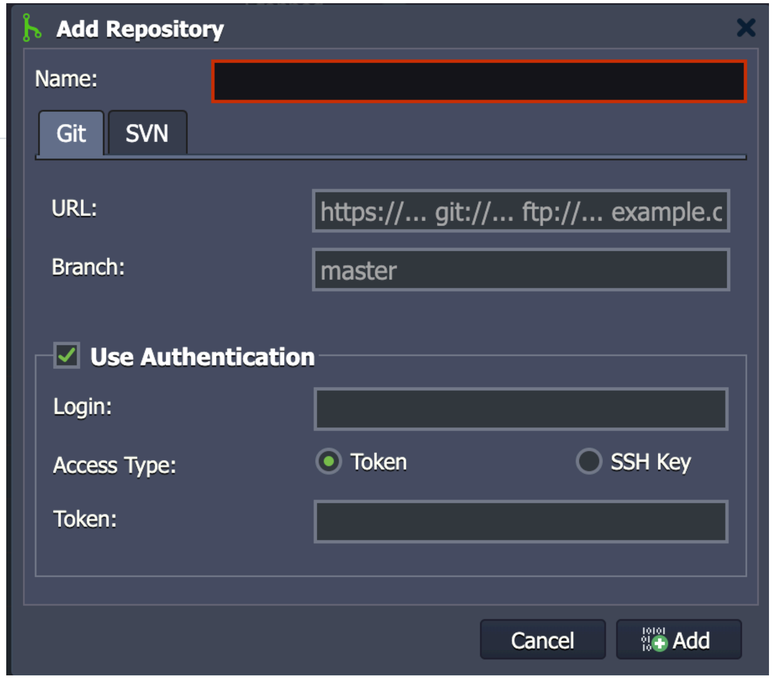

NodeREST
noderest.proen.app.ruk-com.cloud
Running
Load Balancer
NGINX 1.22.0
Node ID: 40730
Front-End
Node.js 16.19.0 LTS
Node ID: 40732
Deployments
API-Server
Node.js 16.19.0 LTS
Node ID: 41091
Deployments


Edit Project
Repository:
NodeRest
Branch:
main
Hooks: Deploy (Post)
Check and auto-deploy updates
1 min
Auto-resolve conflicts
How to enable zero-downtime deployment
Cancel
Apply
Deploy

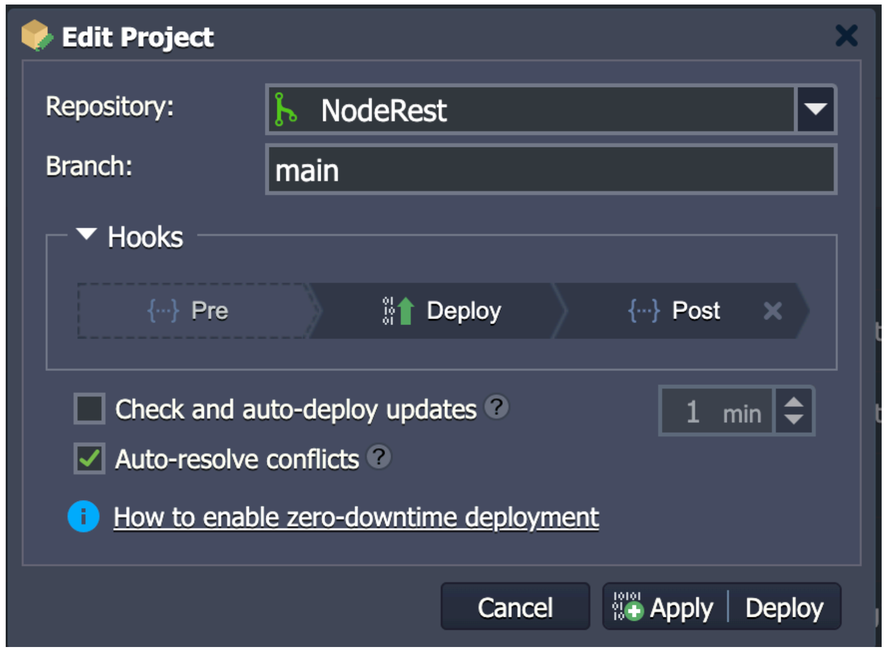

Anirach / NodeREST
Private
Unwatch 1
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
Local
Codespaces
Clone
HTTPS
SSH
GitHub CLI
https://github.com/Anirach/NodeREST.git
Use Git or checkout with SVN using the web URL.
Anirach Merge branch 'main' of https://github.com/Anirach/NodeREST
Database
check SQLite3
NodeExpressREST
update
docs
AddFrontBack Folder
src
Merge branch 'main' of https:
About
No descripti
provided.
Readme
0 stars
1 watchin
0 forks

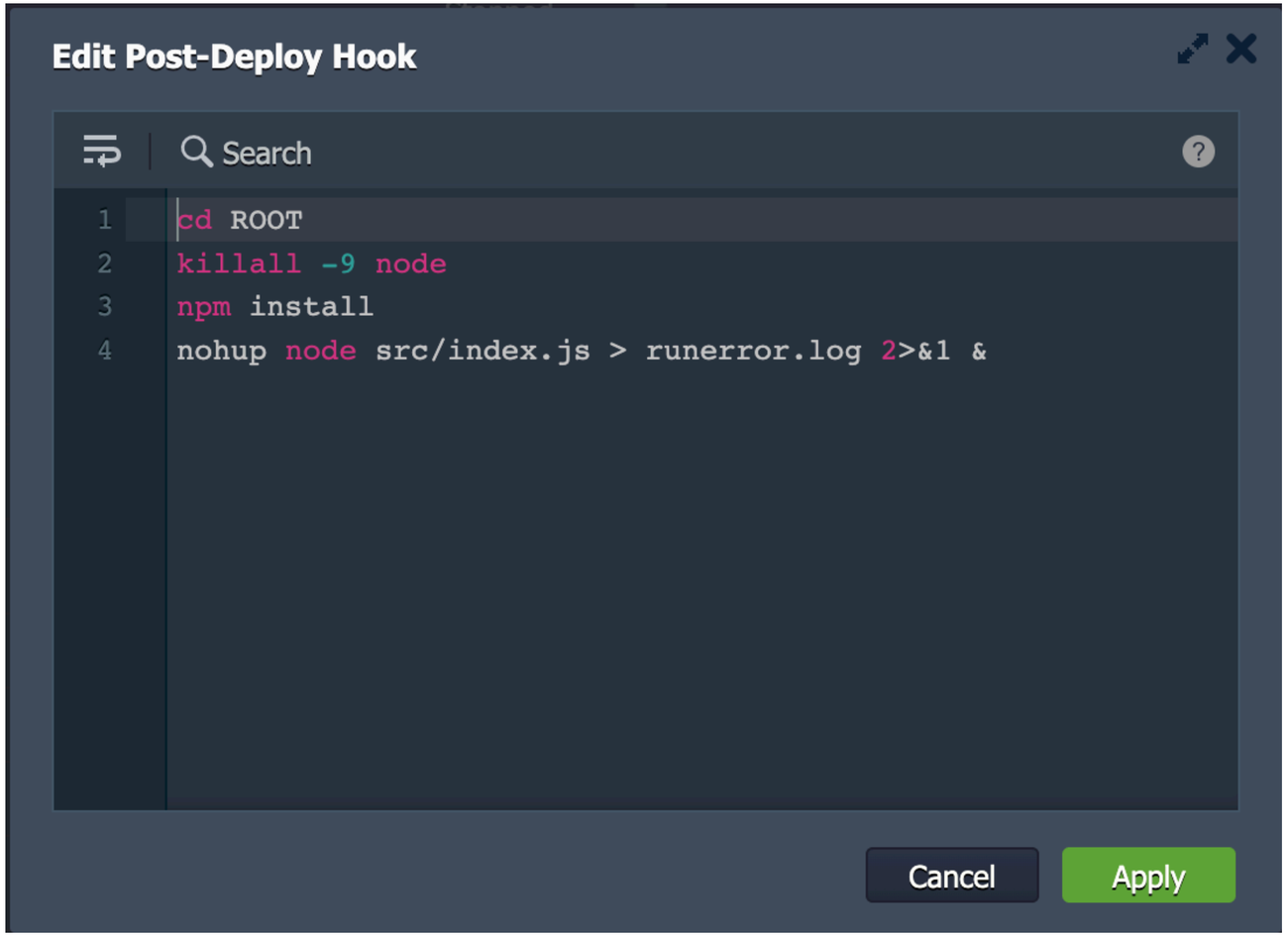

Add Repository
Name:
Git
SVN
URL:
https://... git://... ftp://... example.c
Branch:
master
Use Authentication
Login:
Access Type:
Token
SSH Key
Token:
Cancel
Add

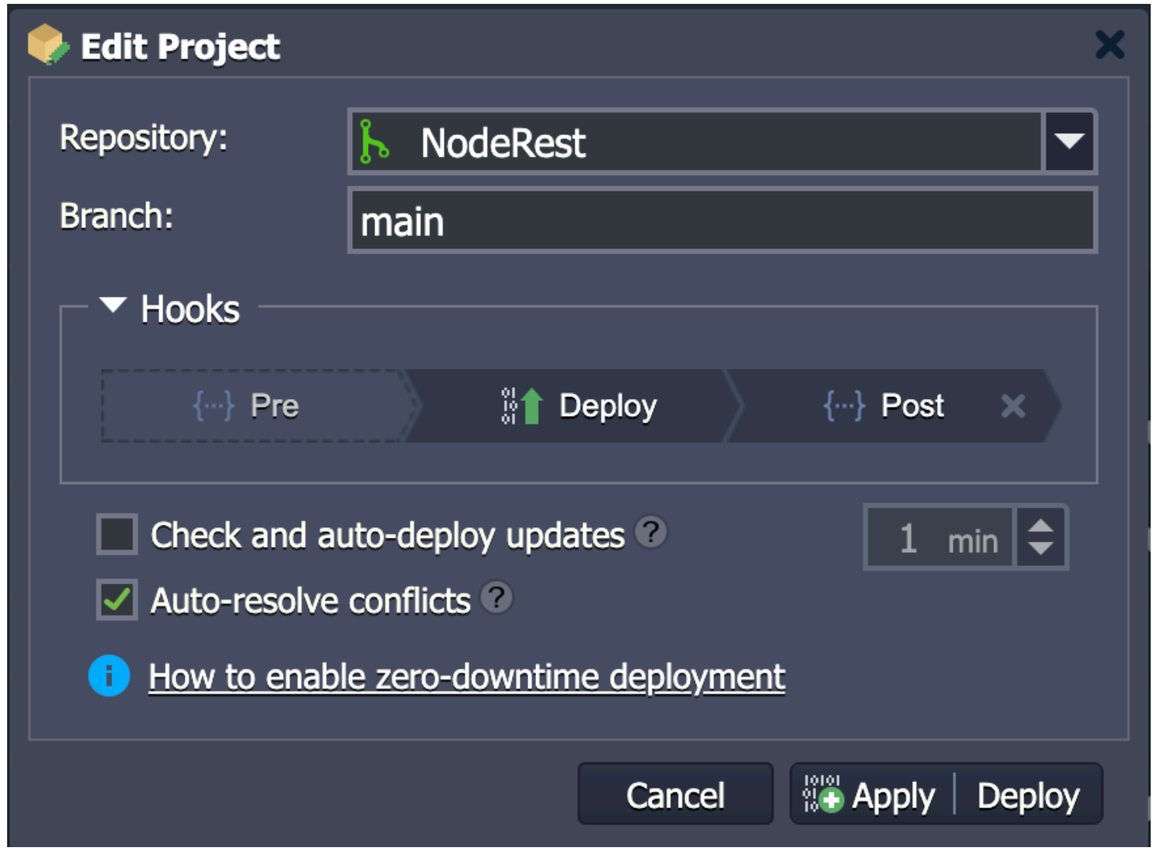


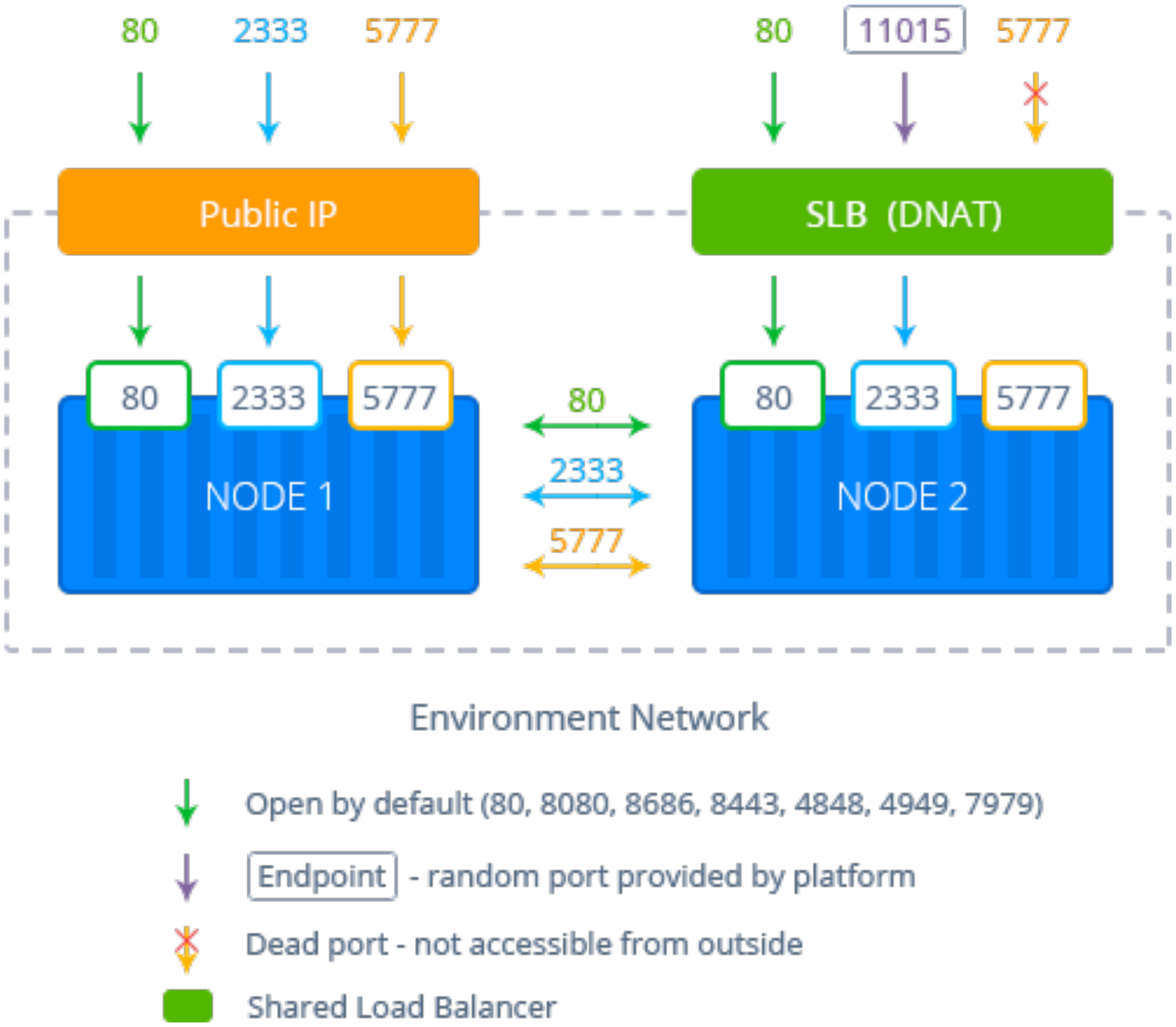


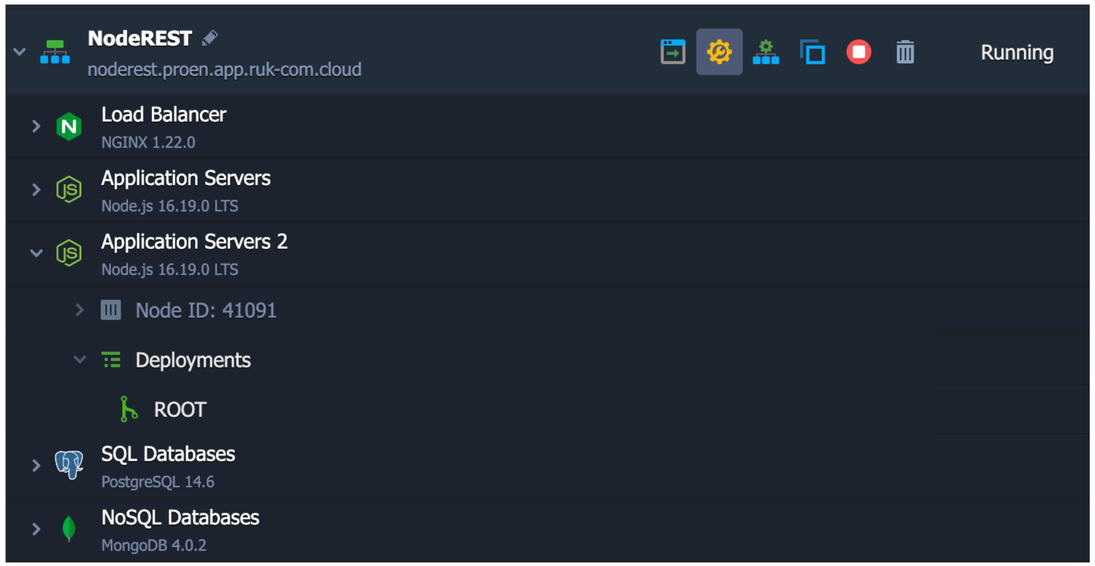
https://github.com/settings/tokens

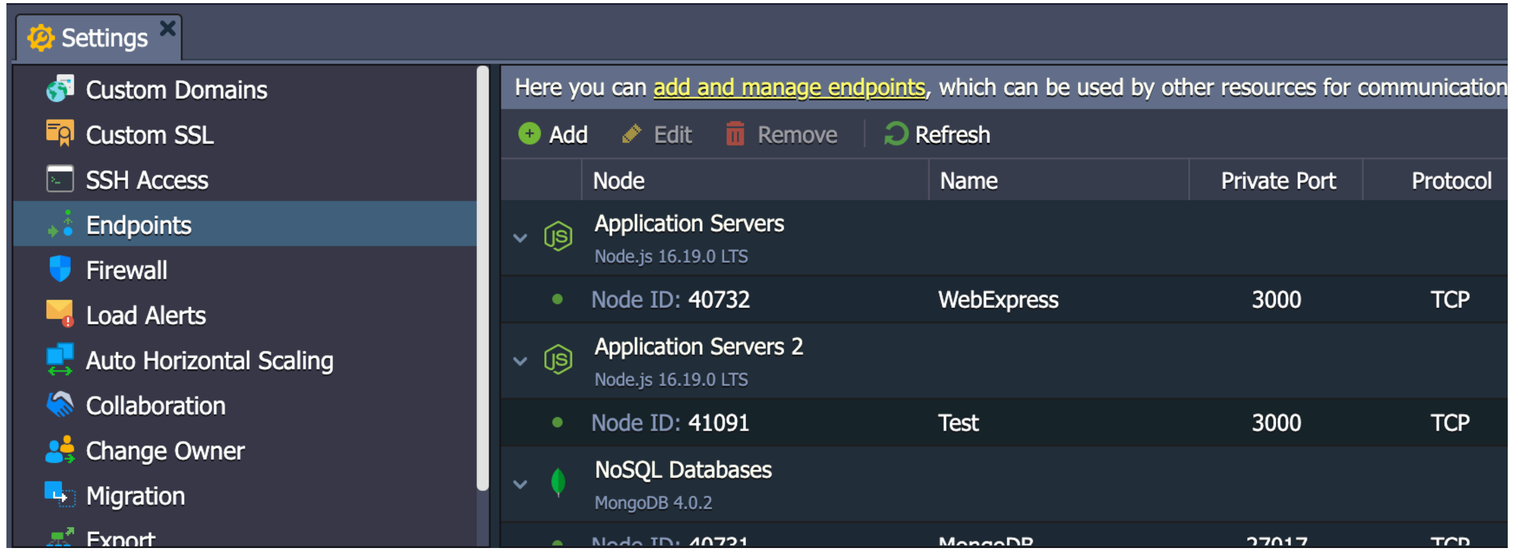

Edit Project
Repository:
NodeRest
Branch:
main
Hooks
Pre
Deploy
Post
Check and auto-deploy updates
1 min
Auto-resolve conflicts
How to enable zero-downtime deployment
Cancel
Apply
Deploy

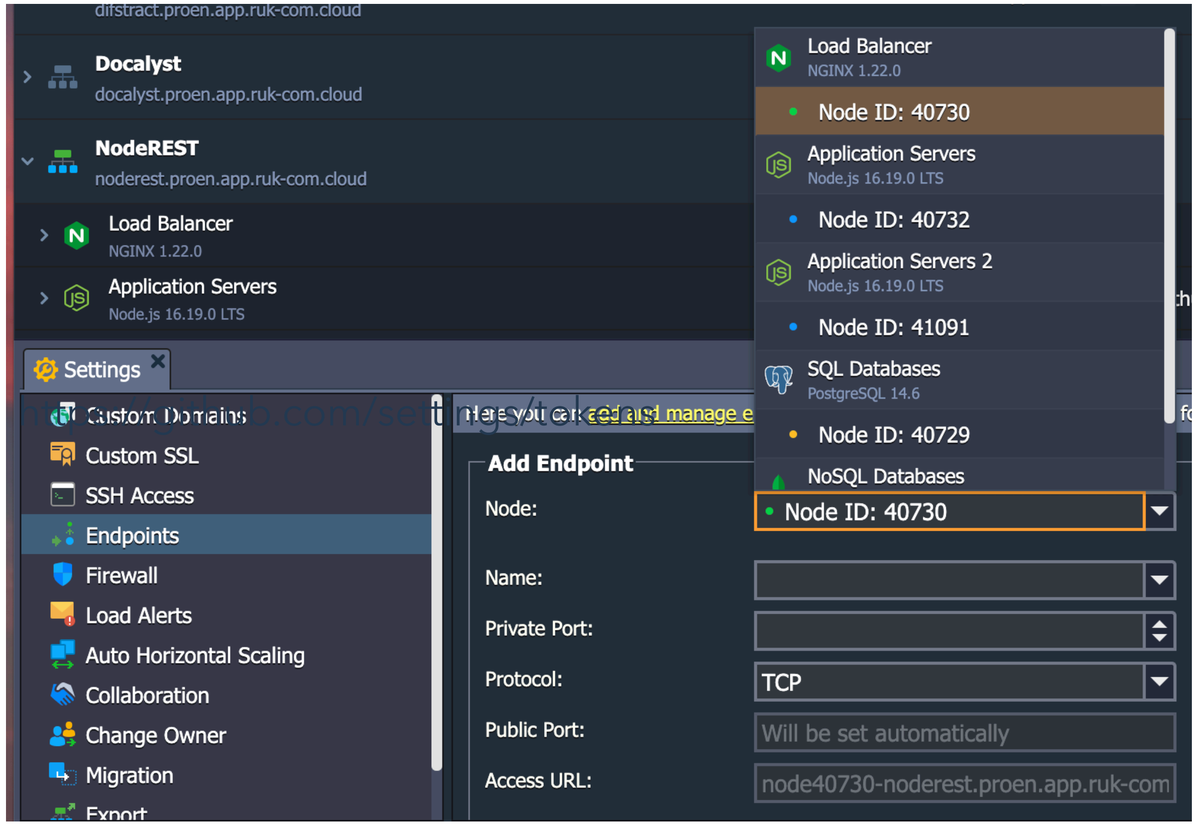

Edit Post-Deploy Hook
Search
1 cd ROOT
2 killall -9 node
3 npm install
4 nohup node src/index.js > runerror.log 2>&1 &
Cancel
Apply

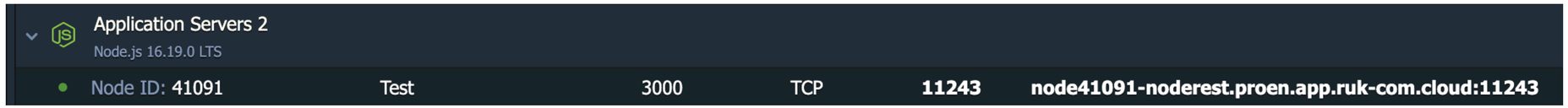

Edit Project
Repository:
NodeRest
Branch:
main
Hooks
Pre
Deploy
Post
Check and auto-deploy updates
1 min
Auto-resolve conflicts
How to enable zero-downtime deployment
Cancel
Apply
Deploy

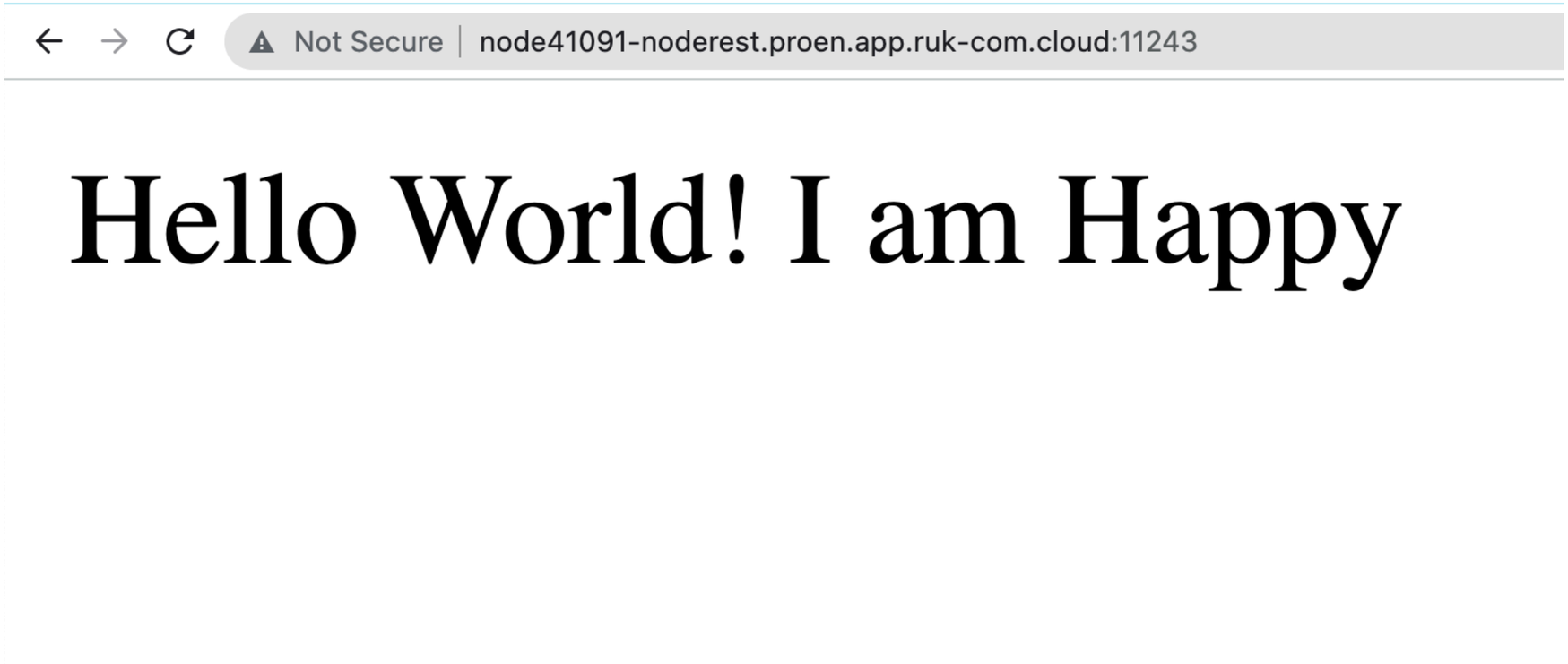

80
2333
5777
80
11015
5777
Public IP
SLB (DNAT)
80
2333
5777
80
2333
5777
NODE 1
80
2333
5777
NODE 2
Environment Network
Open by default (80, 8080, 8686, 8443, 4848, 4949, 7979)
Endpoint - random port provided by platform
Dead port - not accessible from outside
Shared Load Balancer


NodeREST
noderest.proen.app.ruk-com.cloud
Running
Load Balancer
NGINX 1.22.0
Application Servers
Node.js 16.19.0 LTS
Application Servers 2
Node.js 16.19.0 LTS
Node ID: 41091
Deployments
ROOT
SQL Databases
PostgreSQL 14.6
NoSQL Databases
MongoDB 4.0.2


Settings
Custom Domains
Custom SSL
SSH Access
Endpoints
Firewall
Load Alerts
Auto Horizontal Scaling
Collaboration
Change Owner
Migration
Export
Here you can add and manage endpoints, which can be used by other resources for communication
Add
Edit
Remove
Refresh
Node
Name
Private Port
Protocol
Application Servers
Node.js 16.19.0 LTS
Node ID: 40732
WebExpress
3000
TCP
Application Servers 2
Node.js 16.19.0 LTS
Node ID: 41091
Test
3000
TCP
NoSQL Databases
MongoDB 4.0.2

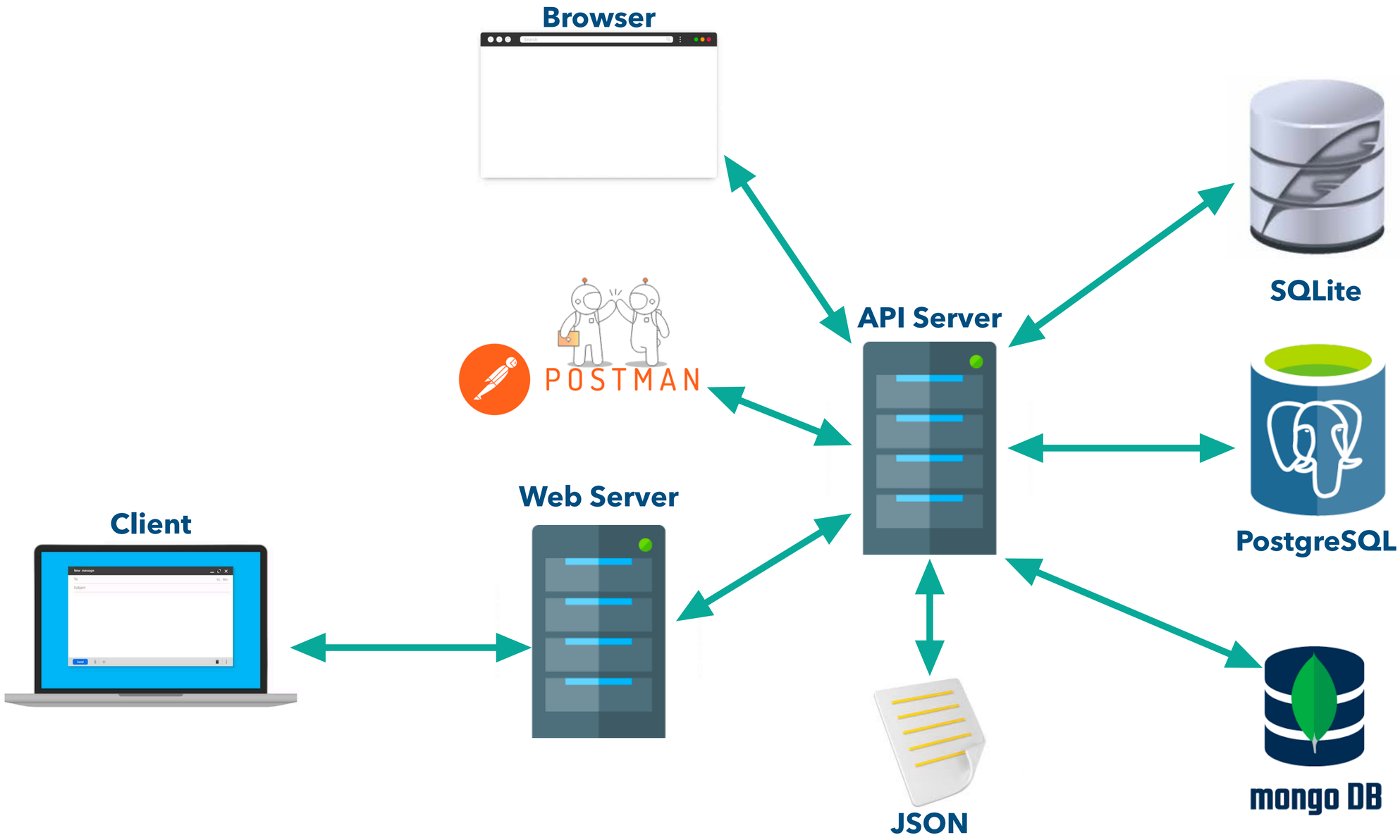

difstract.proen.app.ruk-com.cloud
Docalyst
docalyst.proen.app.ruk-com.cloud
NodeREST
noderest.proen.app.ruk-com.cloud
Load Balancer
NGINX 1.22.0
Application Servers
Node.js 16.19.0 LTS
Settings
Custom Domains
Custom SSL
SSH Access
Endpoints
Firewall
Load Alerts
Auto Horizontal Scaling
Collaboration
Change Owner
Migration
Export
Here you can add and manage e
Add Endpoint
Node:
Node ID: 40730
Name:
Private Port:
Protocol:
TCP
Public Port:
Will be set automatically
Access URL:
node40730-noderest.proen.app.ruk-com
Load Balancer
NGINX 1.22.0
Node ID: 40730
Application Servers
Node.js 16.19.0 LTS
Node ID: 40732
Application Servers 2
Node.js 16.19.0 LTS
Node ID: 41091
SQL Databases
PostgreSQL 14.6
Node ID: 40729
NoSQL Databases


Application Servers 2
Node.js 16.19.0 LTS
Node ID: 41091
Test
3000
TCP
11243
node41091-noderest.proen.app.ruk-com.cloud:11243


Not Secure
node41091-noderest.proen.app.ruk-com.cloud:11243
Hello World! I am Happy

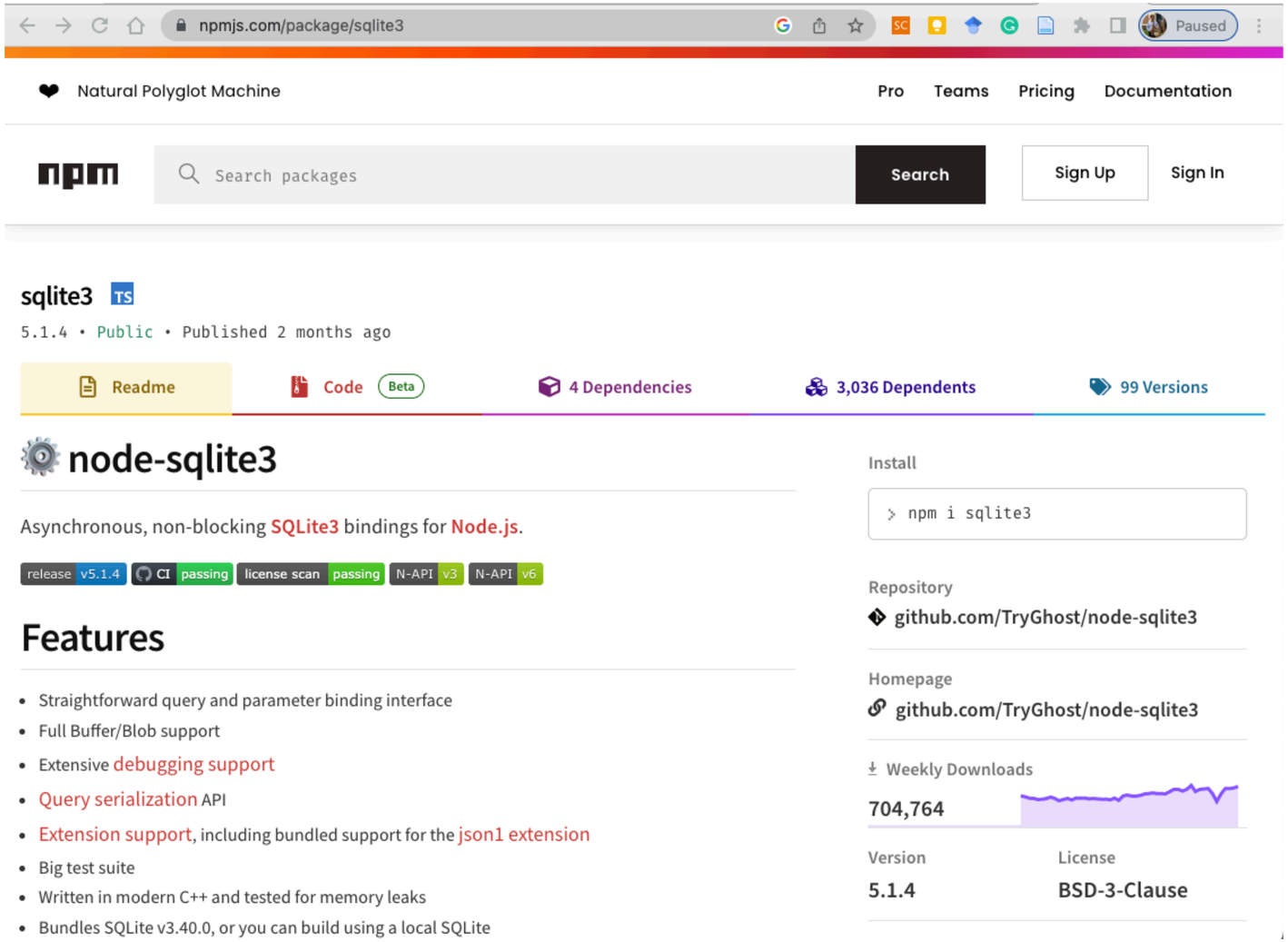
LOCAL DATABASE

What we will learn?



SQLITE

SQLITE



npm i sqlite3

SQLITE VIEWER

Extension: SQLite Viewer

SQLite Viewer v0.1.5

Florian Klampfer | 307,191 | ★★★★★ (23)

SQLite Viewer for VSCode

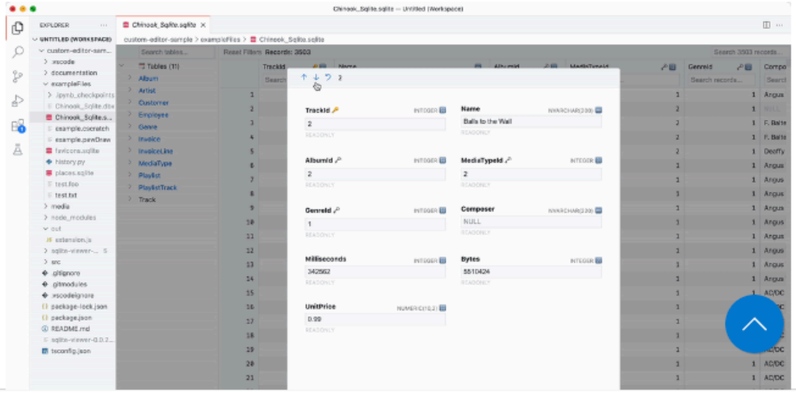
Disable | Uninstall | Switch to Pre-Release Version

This extension is enabled globally.

DETAILS FEATURE CONTRIBUTIONS CHANGELOG RUNTIME STATUS

SQLite Viewer for VSCode

A quick and easy SQLite viewer for VSCode, inspired by DB Browser for SQLite and Airtable.



Categories

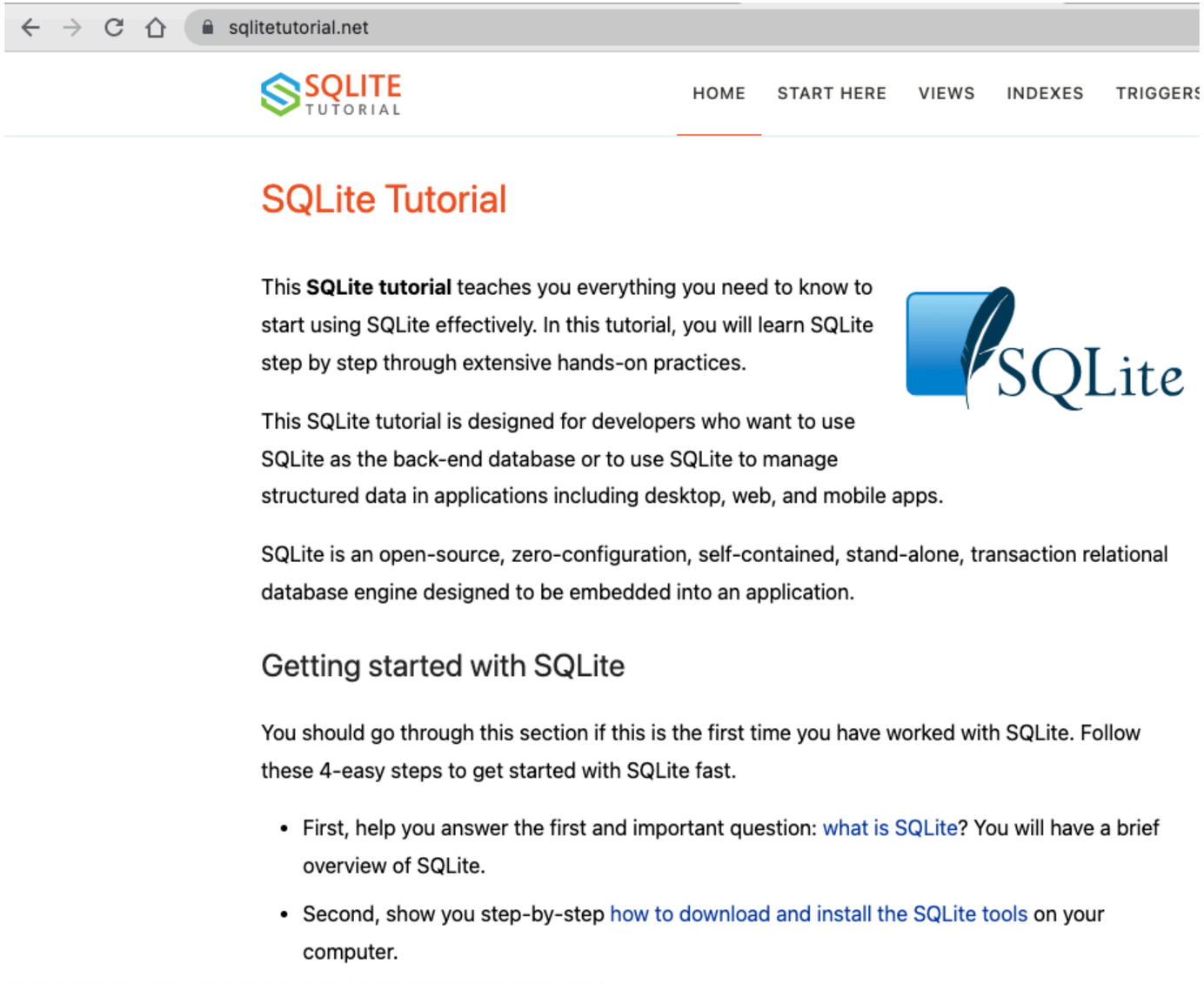
Other

Extension Resources

Marketplace
Repository
License
Florian Klampfer

More Info

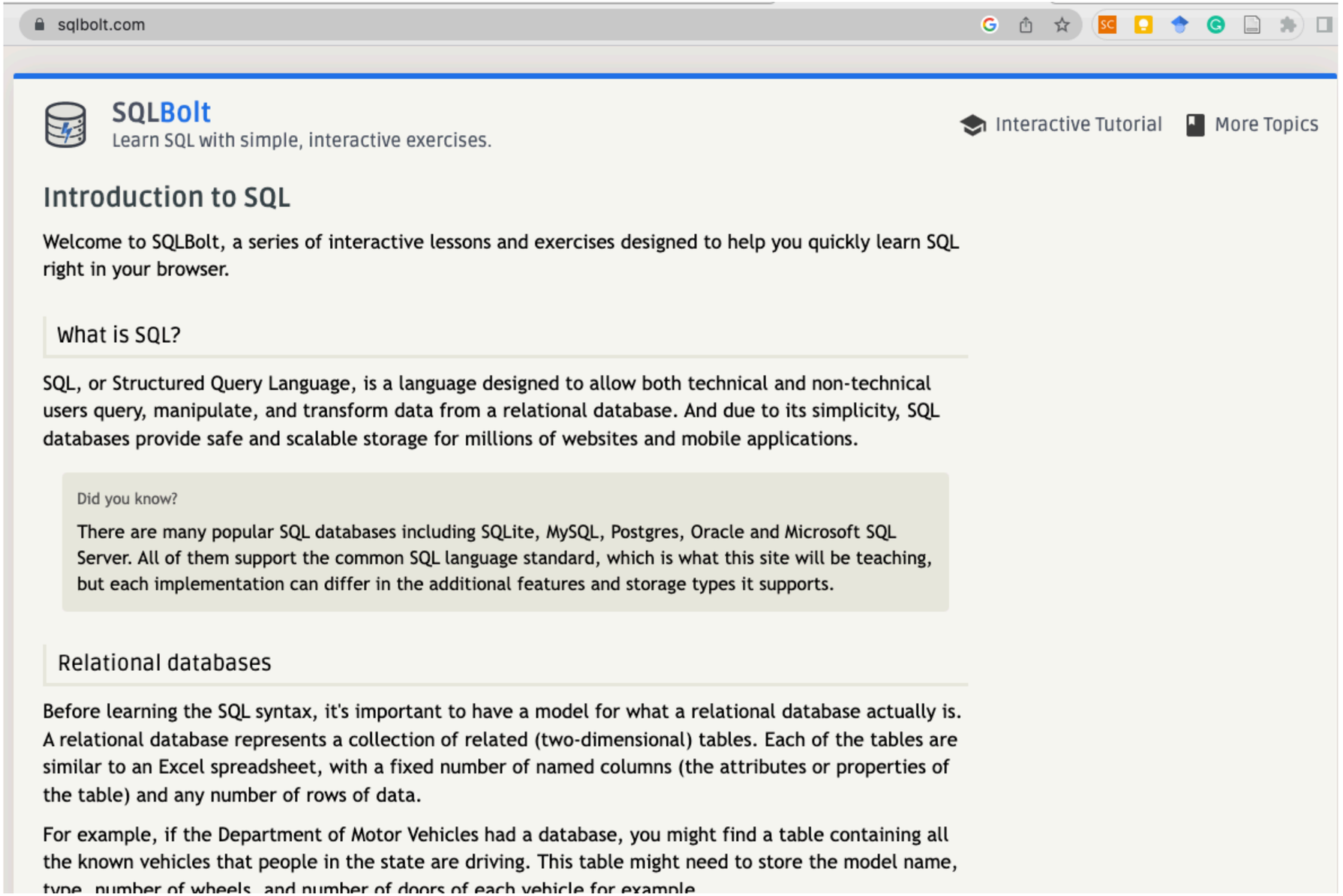
Published	10/1/2021, 15:04:30
Last released	3/21/2022, 11:57:39
Last updated	10/1/2022, 17:34:16
Identifier	qwtel.sqlite-



https://www.sqlitetutorial.net/

SQL COMMAND

```
// create books table if it doesn't exist
db.run(`CREATE TABLE IF NOT EXISTS books (
    id INTEGER PRIMARY KEY,
    title TEXT,
    author TEXT
)`);
```



https://sqlbolt.com/

API-SQL-SQLITE

CRUDBookSQLite.js U

src > CRUDBookSQLite.js > ...

```
// SQLite3 CRUD operations
// npm install sqlite3
// Create a Bood.sqlite file in Database folder
// Run this file with node CRUDBookSQLite.js
// Test with Postman

const express = require('express');
const sqlite3 = require('sqlite3');
const app = express();

// connect to database
const db = new sqlite3.Database('./Database/Book.sqlite');

// parse incoming requests
app.use(express.json());

// create books table if it doesn't exist
db.run(`CREATE TABLE IF NOT EXISTS books (
  id INTEGER PRIMARY KEY,
  title TEXT,
  author TEXT
)`);

```

CRUDBookSQLite.js U

src > CRUDBookSQLite.js > ...

```
23
24  // route to get all books
25  app.get('/books', (req, res) => {
26    db.all('SELECT * FROM books', (err, rows) => {
27      if (err) {
28        res.status(500).send(err);
29      } else {
30        res.json(rows);
31      }
32    });
33  });
34
35  // route to get a book by id
36  app.get('/books/:id', (req, res) => {
37    db.get('SELECT * FROM books WHERE id = ?', req.params.id, (err, row) => {
38      if (err) {
39        res.status(500).send(err);
40      } else {
41        if (!row) {
42          res.status(404).send('Book not found');
43        } else {
44          res.json(row);
45        }
46      }
47    });
48  });
49
```

CRUDBookSQLite.js U

src > CRUDBookSQLite.js > ...

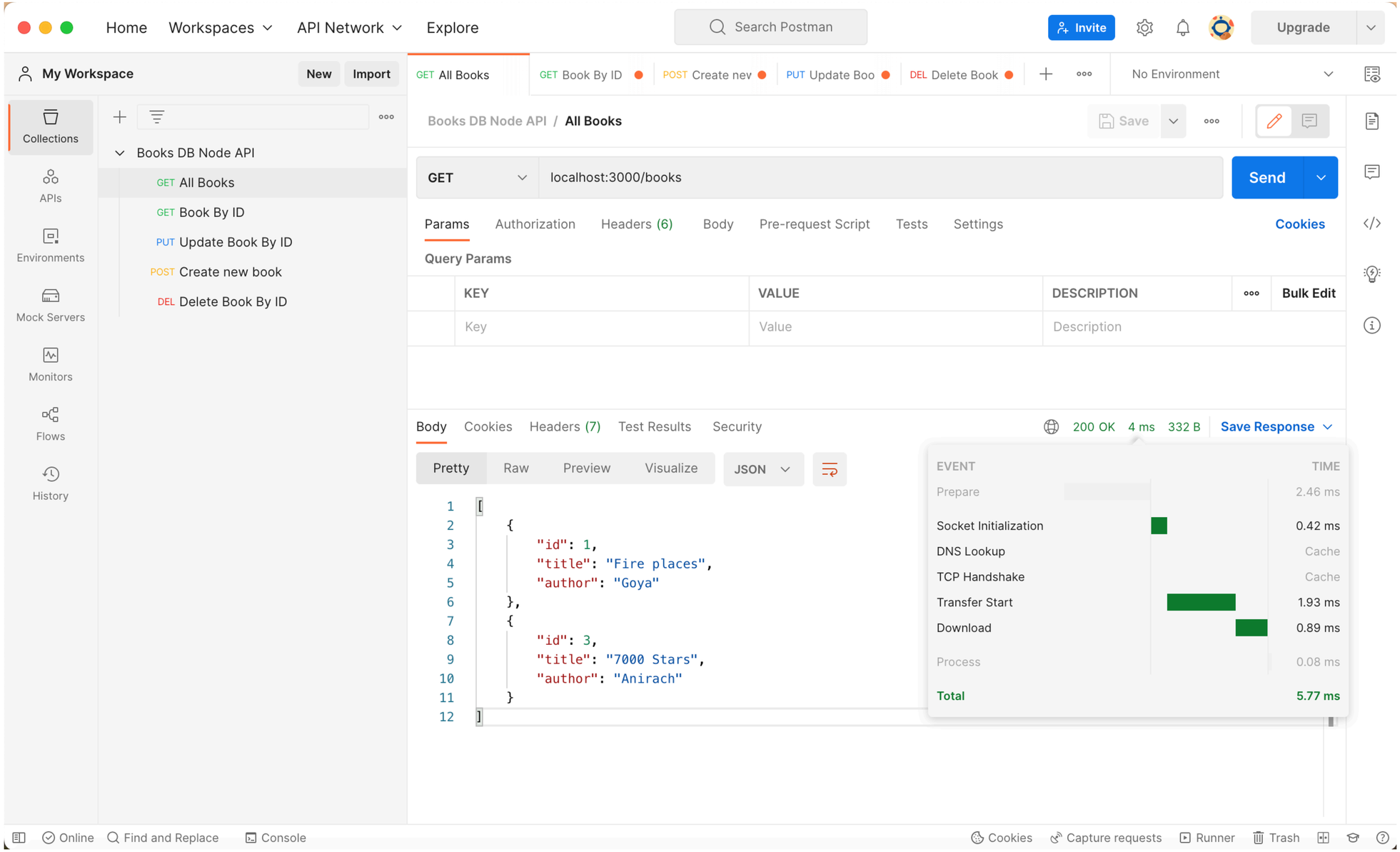
```
49
50  // route to create a new book
51  app.post('/books', (req, res) => {
52    const book = req.body;
53    db.run('INSERT INTO books (title, author) VALUES (?, ?)', book.title, book.author, function(err) {
54      if (err) {
55        res.status(500).send(err);
56      } else {
57        book.id = this.lastID;
58        res.send(book);
59      }
60    });
61  });
62
63  // route to update a book
64  app.put('/books/:id', (req, res) => {
65    const book = req.body;
66    db.run('UPDATE books SET title = ?, author = ? WHERE id = ?', book.title, book.author, req.params.id, function(err) {
67      if (err) {
68        res.status(500).send(err);
69      } else {
70        res.send(book);
71      }
72    });
73  });
74
```

CRUDBookSQLite.js U

src > CRUDBookSQLite.js > ...

```
74
75  // route to delete a book
76  app.delete('/books/:id', (req, res) => {
77    db.run('DELETE FROM books WHERE id = ?', req.params.id, function(err) {
78      if (err) {
79        res.status(500).send(err);
80      } else {
81        res.send({});
82      }
83    });
84  });
85
86  const port = process.env.PORT || 3000;
87  app.listen(port, () => console.log(`Listening on port ${port}...`));
88
```

TEST WITH POSTMAN


Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
GET All Books
GET Book By ID
POST Create nev
PUT Update Boo
DEL Delete Book
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Books DB Node API
GET All Books
GET Book By ID
PUT Update Book By ID
POST Create new book
DEL Delete Book By ID
Books DB Node API / All Books
Save
GET
localhost:3000/books
Send
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
Key
Value
Description
Body
Cookies
Headers (7)
Test Results
Security
200 OK
4 ms
332 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": 1,
"title": "Fire places",
"author": "Goya"
},
{
"id": 3,
"title": "7000 Stars",
"author": "Anirach"
}
]
EVENT
TIME
Prepare
2.46 ms
Socket Initialization
0.42 ms
DNS Lookup
Cache
TCP Handshake
Cache
Transfer Start
1.93 ms
Download
0.89 ms
Process
0.08 ms
Total
5.77 ms
Online
Find and Replace
Console
Cookies
Capture requests
Runner
Trash

THANK YOU

- PaaS
- Cloud Setting
- Local Database
- SQLite
- SQL command
- API with SQLite DB
- Postman Test