

Testing & Monitoring Tools



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*





*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



*“The most advanced
load-testing platform
in the world.”*



*“A fancy
self-hosted monitoring tool”*





*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*





*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*





*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*





*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*



Access



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*

*paid service
website UI*



*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*

Access



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*

*paid service
website UI*



Artillery

*“The most advanced
load-testing platform
in the world.”*

*Open Source npm package
Terminal (CLI)*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*

Access



“Next-Generation Mobile Apps and Cross Browser Testing Cloud”

*paid service
website UI*



“The most advanced load-testing platform in the world.”

*Open Source npm package
Terminal (CLI)*



Uptime Kuma

“A fancy self-hosted monitoring tool”

*Open Source Docker image
website UI*

Claris / FM product: Tested



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*

Claris / FM product: Tested



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*

Claris / FM product: Tested



“Next-Generation Mobile Apps and Cross Browser Testing Cloud”

✓ *WebDirect*



“The most advanced load-testing platform in the world.”

✓ *Data API*

✓ *WebDirect*



Uptime Kuma

“A fancy self-hosted monitoring tool”

Claris / FM product: Tested



“Next-Generation Mobile Apps and Cross Browser Testing Cloud”

✓ *WebDirect*



“The most advanced load-testing platform in the world.”

✓ *Data API*
✓ *WebDirect*



“A fancy self-hosted monitoring tool”

✓ *ping responders*
✓ *Data API*
✓ *WebDirect*

Claris / FM product: Tested



“Next-Generation Mobile Apps and Cross Browser Testing Cloud”

✓ *WebDirect*



“The most advanced load-testing platform in the world.”

✓ *Data API*
✓ *WebDirect*



“A fancy self-hosted monitoring tool”

✓ *ping responders*
✓ *Data API*
✓ *WebDirect*

Presentation Time



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*

Presentation Time



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*

short



*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*

Presentation Time



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*

short



*“The most advanced
load-testing platform
in the world.”*

longer (coding!!!)



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*

Presentation Time



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*

short



*“The most advanced
load-testing platform
in the world.”*

longer (coding!!!)



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*

mercifully short

Presentation Time



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*

short



*“The most advanced
load-testing platform
in the world.”*

longer (coding!!!)



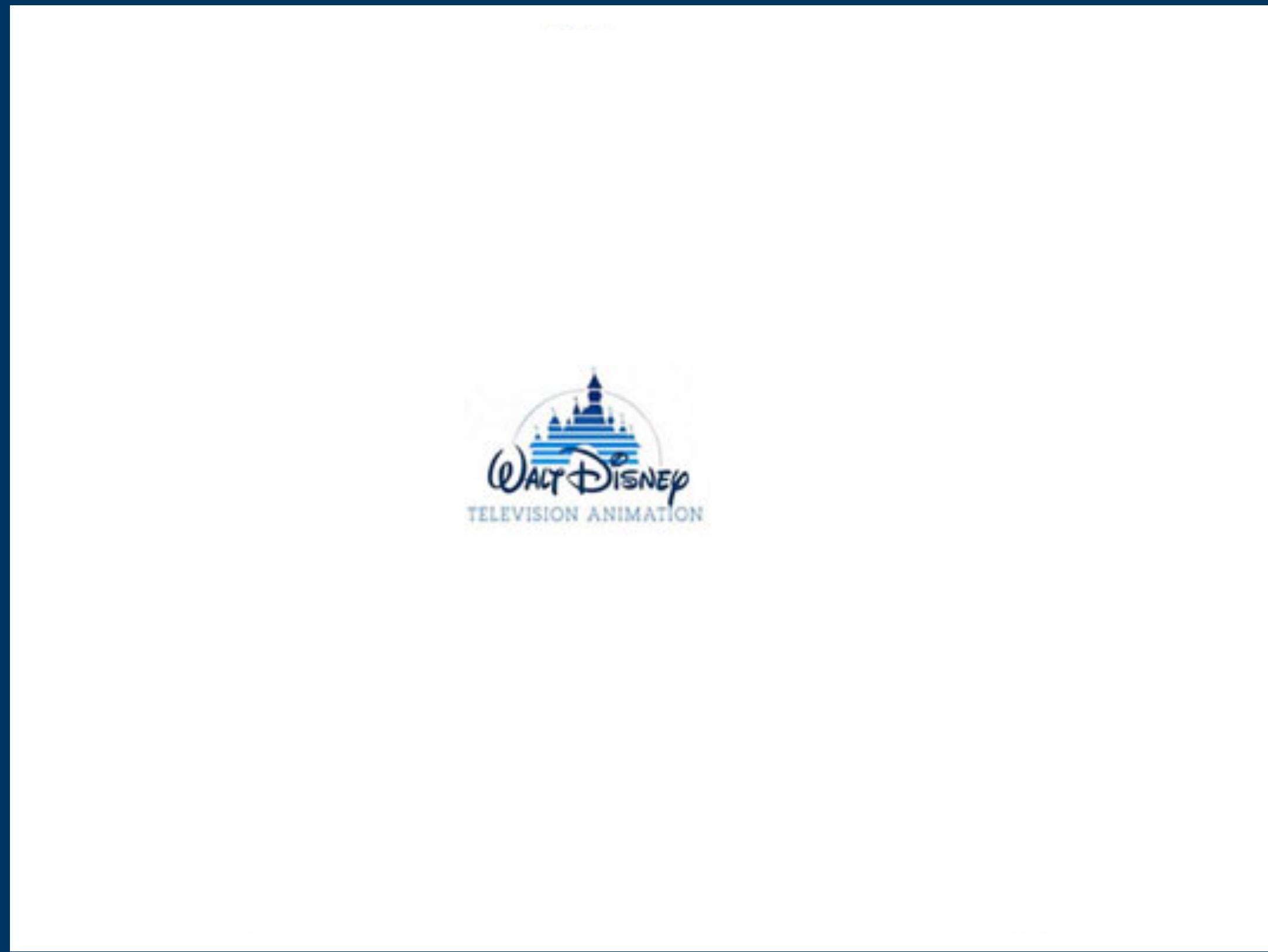
*“A fancy
self-hosted monitoring tool”*

mercifully short

Goal: introduce / demystify these tools

Patrick Reagan's Experience

Patrick Reagan's Experience



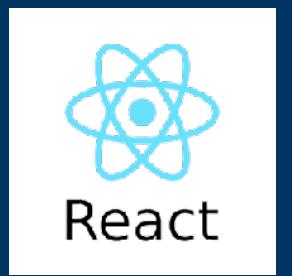
Patrick Reagan's Experience



Patrick Reagan's Experience



Patrick Reagan's Experience



Patrick Reagan's Experience

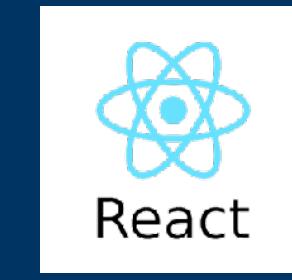
Discovery



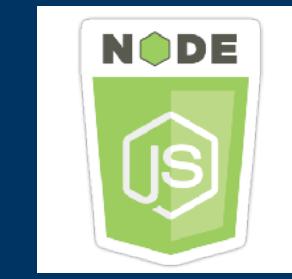
SketchBook



Figma



Front End



Back End



Support

Discovery

Front End

Back End

Support

LambdaTest

Artillery

Uptime Kuma

What they are...

LambdaTest

Artillery

Uptime Kuma

Discovery

Front End

Back End

Support

Discovery

What they are...

LambdaTest

Front End

Artillery

Back End

Uptime Kuma

Support

What they are...

LambdaTest

Artillery + **Playwright Plugin**

Uptime Kuma

Discovery

Front End

Back End

Support

Discovery

What they are...

LambdaTest

Front End

Artillery + Playwright Plugin

Back End

Uptime Kuma

Support

Discovery

What they are...

LambdaTest

Front End

Artillery + Playwright Plugin

Back End

Uptime Kuma

Support

What they are...

LambdaTest

Artillery + **Playwright Plugin**

Uptime Kuma

Discovery

Front End

Back End

Support

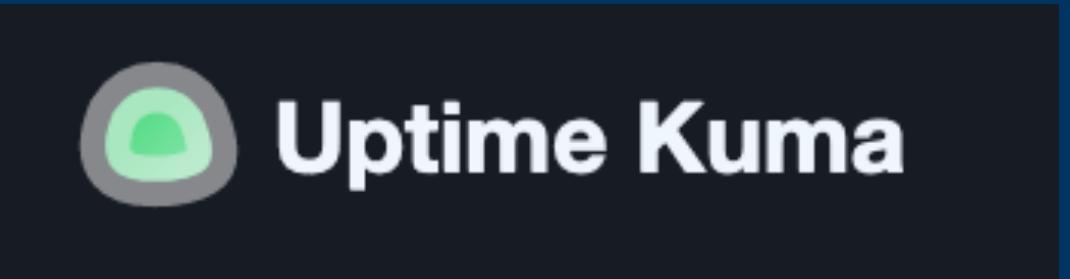
= good basic testing and monitoring



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



*“The most advanced
load-testing platform
in the world.”*



*“A fancy
self-hosted monitoring tool”*



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*

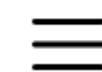


*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*

*paid service
website UI*

 *WebDirect*

<https://www.lambdatest.com/>



Next-Generation Mobile Apps and Cross Browser Testing Cloud

Deliver unparalleled digital experience with our next-gen AI-powered testing cloud platform. Ensure exceptional user experience across all devices and browsers.

[Get Started →](#)

[Learn More](#)

Trusted by 1M+ users globally



Microsoft



vimeo



NVIDIA



Telstra



rubrik

<https://www.lambdatest.com/pricing>

The screenshot shows the LambdaTest Pricing page with the following details:

Affordable and Scalable Pricing

Monthly Annual (Save up to 20%)

Free **\$0**

Live **\$15** PER MONTH BILLED ANNUALLY

Real Device **\$25** PER MONTH BILLED ANNUALLY

Enterprise

BEST VALUE

Real Device

Enterprise

CONTACT SALES

Includes access to:

- > Free live real time testing sessions
- > 10 Screenshot Tests per month
- > 10 Responsive Test per month
- > Unlimited access to LT Browser
- > 100 Minutes of free 2 parallel web+mobile automation testing
- > 3 sessions of Native App Testing on Emulators/Simulators
- > Free 2000 screenshots/month of Visual Regression Testing
- > Free 300 mins of testing on Hyperexecute, with upto 2 concurrent executions

Includes unlimited access to:

- > Desktop Browsers
- > Emulators / Simulators for mobile browsers
- > Emulators / Simulators for mobile app testing
- > Screenshot Testing
- > Responsive Testing
- > Geolocation Testing
- > Web and mobile debugging tools

Includes everything in Live plan plus unlimited access to:

- > Real Mobile devices
- > Test native, hybrid, and web apps
- > App Uploads
- > Natural gestures and interactions
- > Detailed device logs

Everything in Real Devices plan plus:

- > Premium Support options
- > Advanced access controls
- > Single Sign-on
- > IP whitelisting
- > Advanced data retention rules
- > Dedicated Device cloud
- > Advanced Local Testing
- > Usage analytics

Manual Testing
Live interactive exploratory testing

Web Automation Testing
Desktop and mobile browser automation

Native App Automation
Mobile App Automation on Real Devices

HyperExecute
Lightning fast test execution cloud

Smart UI
Automated visual UI testing cloud

GET STARTED

What are Parallel Tests?

GET STARTED

What are Parallel Tests?

GET STARTED

What are Parallel Tests?

GET STARTED

CONTACT SALES

CHAT

<https://www.lambdatest.com/dashboard>

The screenshot shows the LambdaTest dashboard interface. On the left, a vertical sidebar contains icons for different features: Home, Automation, Realtime, Test Logs, Test Types, Date Range, and Help.

REALTIME SESSIONS: Subscribed 1 Parallel Sessions. Total Realtime Tests : 29. A blue gauge meter shows 0 Parallel Sessions. A teal button labeled "View Logs" is visible.

AUTOMATION SESSIONS: Subscribed 2 Parallel Sessions. Free Minutes : 0 / 100. A pink gauge meter shows 0 Parallel Sessions. A red button labeled "Start Test" is visible.

INTEGRATIONS: Options to integrate with Jira, Slack, and Asana, each with an "INSTALL" button. A "See More" link is also present.

RECENT TESTS: A list of recent test runs. Each entry includes the URL, user (Live | patrickd3), browser (Chrome 111), and device (iPhone 10.14). The URLs listed are:

- URL : https://www.staging.leonj.d3.tools
- URL : https://www.staging.leonj.d3.tools
- URL : https://www.leonjoosen.com
- URL : https://www.staging.leonj.d3.tools
- URL : https://www.leonjoosen.com
- URL : https://d3tests.top/fmi/webd/Contacts

Bottom Navigation: Buttons for "All Live Tests" and "All Automation Tests".

<https://www.lambdatest.com/dashboard>

The screenshot shows the LambdaTest dashboard interface. On the left, there's a vertical sidebar with various icons and a 'More Tools' dropdown menu. A large yellow arrow points to the 'More Tools' button. The main dashboard area has three primary sections: 'REALTIME SESSIONS' (Subscribed 1 Parallel Sessions, Total Realtime Tests : 29), 'AUTOMATION SESSIONS' (Subscribed 2 Parallel Sessions, Free Minutes : 0 / 100), and 'INTEGRATIONS' (Jira, Slack, Asana). To the right, there's a sidebar for 'Recent Tests' and 'Recent Issues'.

REALTIME SESSIONS
Subscribed 1 Parallel Sessions
Total Realtime Tests : 29

AUTOMATION SESSIONS
Subscribed 2 Parallel Sessions
Free Minutes : 0 / 100

INTEGRATIONS

- Jira [INSTALL](#)
- Slack [INSTALL](#)
- Asana [INSTALL](#)

RECENT TESTS

- URL : <https://www.staging.leonj.d3.tools>
Live | patrickd3 111 10.14
- URL : <https://www.staging.leonj.d3.tools>
Live | patrickd3 111 10.14
- URL : <https://www.leonjoosen.com>
Live | patrickd3 111 12.0
- URL : <https://www.staging.leonj.d3.tools>
Live | patrickd3 111 10.14
- URL : <https://www.leonjoosen.com>
Live | patrickd3 111 12.0
- URL : <https://d3tests.top/fmi/webd/Contacts>
Live | patrickd3 111 13

More Tools

- Screenshot
- Responsive
- UI Comparison
- LT Browser
- Issue Tracker
- Test Logs
- Projects

<https://app.lambdatest.com/console/screenshot>

<https://app.lambdatest.com/console/screenshot>

The screenshot shows the LambdaTest console interface for capturing screenshots across multiple browsers and devices. The URL <https://app.lambdatest.com/console/screenshot> is displayed in the browser's address bar. The main area is titled "Capture screenshots for multiple browsers & devices". A sidebar on the left contains various icons for different features like tunneling, automation, and reporting.

Place your URL: <https://d3tests.top/fmi/webd/Contacts>

Optional Select Tunnel

CAPTURE

3/25

Windows 10	113	112	111	113	112	111	98	97	96	113	112	111	11
	110	109	108	110	109	108	95	94	93	110	109	108	
	107	106	105	107	106	105	92	91	90	107	106	105	
	67 more...	73 more...	64 more...	29 more...									

Windows 8.1	109	108	107	113	112	111	95	94	93	109	108	107	11
	106	105	104	110	109	108	92	91	90	106	105	104	
	103	102	101	107	106	105	89	88	87	103	102	101	
	73 more...	89 more...	63 more...	21 more...									

Windows 8	109	108	107	113	112	111	95	94	93	109	108	107	10
	106	105	104	110	109	108	92	91	90	106	105	104	
	103	102	101	107	106	105	89	88	87	103	102	101	
	73 more...	89 more...	63 more...	21 more...									

Windows 7	109	108	107	113	112	111	90	89	88	109	108	107	11	10	9
	106	105	104	110	109	108	87	85	84	106	105	104			8

Feedback icon:

<https://app.lambdatest.com/console/screenshot>

Enter URL



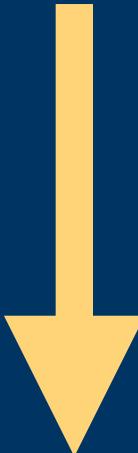
The screenshot shows the LambdaTest console interface for capturing screenshots across multiple browsers and devices. The URL <https://d3tests.top/fmi/webd/Contacts> is entered in the URL bar. The interface includes a sidebar with various icons for device selection, browser selection, and other tools. The main area displays a grid of browser and device combinations. The first row shows results for Windows 10: Chrome (113), Firefox (113), Opera (0), Edge (113), and Internet Explorer (113). The second row shows results for Windows 8.1: Chrome (109), Firefox (113), Opera (95), Edge (109), and Internet Explorer (11). The third row shows results for Windows 8: Chrome (109), Firefox (113), Opera (95), Edge (109), and Internet Explorer (10). The fourth row shows results for Windows 7: Chrome (109), Firefox (113), Opera (90), Edge (109), and Internet Explorer (11). Each row has a "more..." link at the bottom right. A large orange "CAPTURE" button is located in the top right corner of the main area.

<https://app.lambdatest.com/console/screenshot>

Enter URL

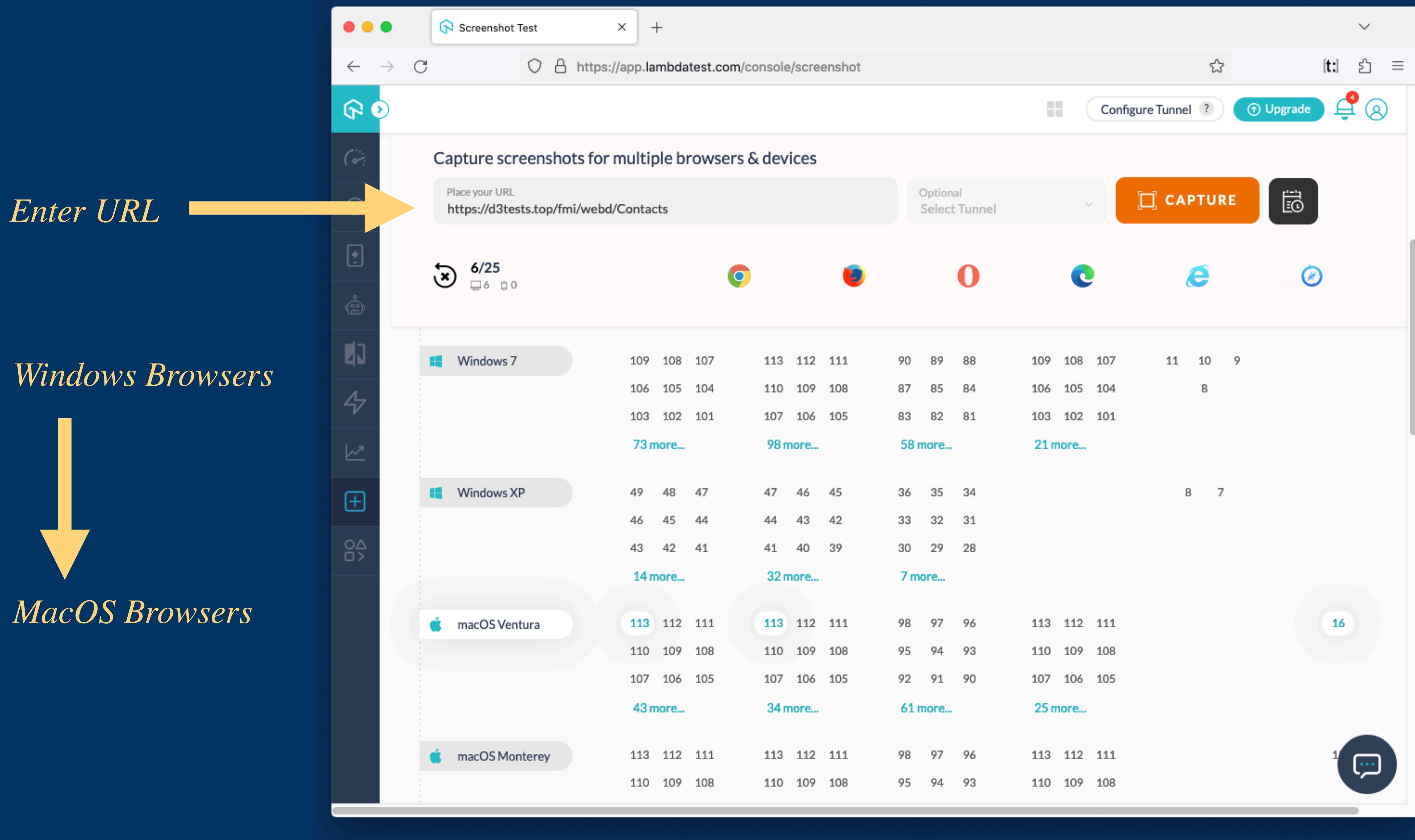


Windows Browsers

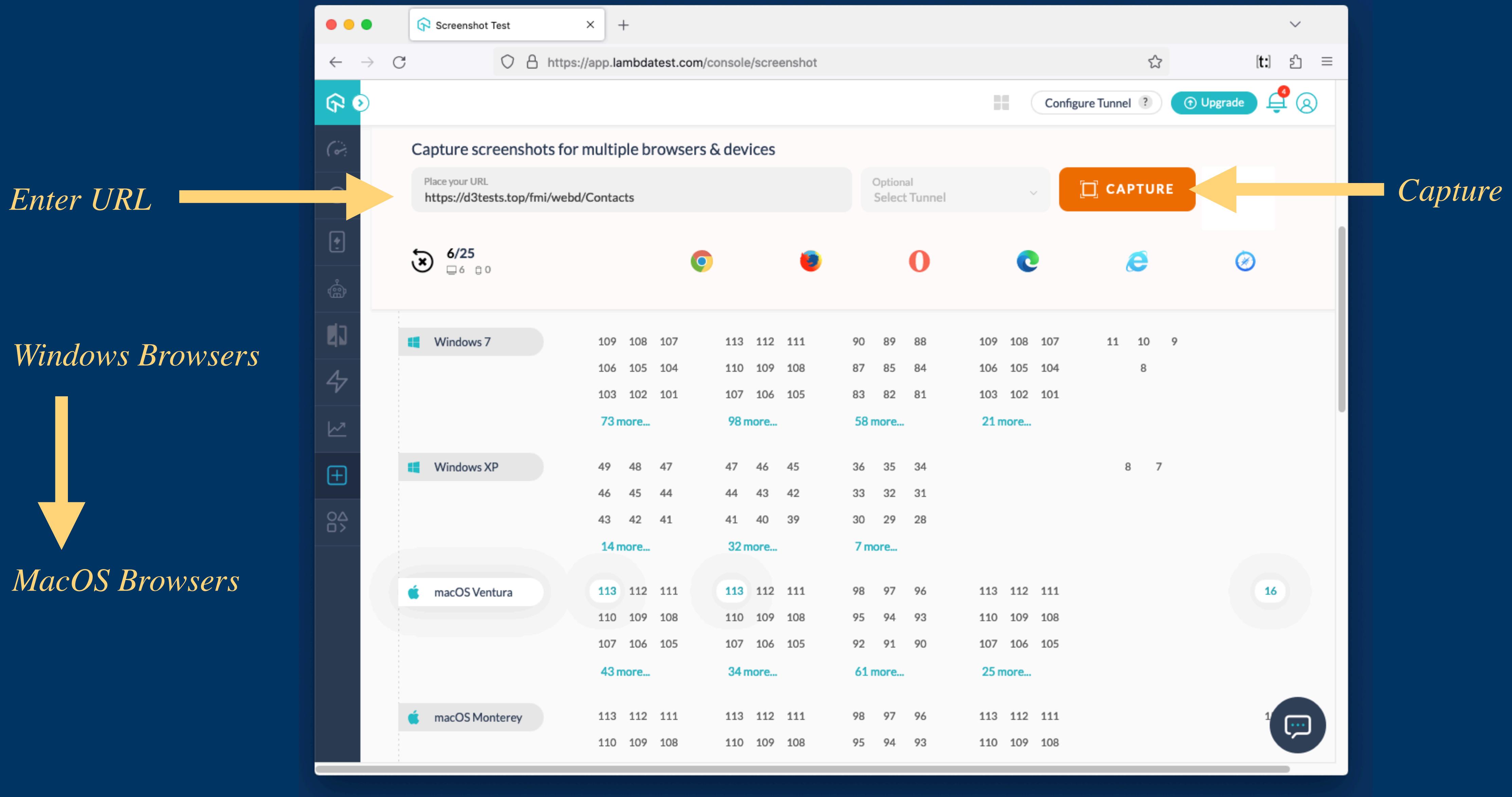


The screenshot shows the LambdaTest console interface for capturing screenshots across multiple browsers and devices. The main area displays a grid of browser icons (Chrome, Firefox, Opera, Edge, Internet Explorer) and OS versions (Windows 10, Windows 8.1, Windows 8, Windows 7). Each row represents a combination of browser and OS, with numerical values indicating screen counts or dimensions. A sidebar on the left contains various icons for different test types and configurations. At the top, there is a URL input field with the placeholder "Place your URL" and a "CAPTURE" button. The overall theme is dark with light-colored UI elements.

OS	Browser	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6	Value 7	Value 8	Value 9	Value 10	Value 11	
Windows 10	Chrome	113	112	111	113	112	111	98	97	96	113	112	111
	Firefox	110	109	108	110	109	108	95	94	93	110	109	108
	Opera	107	106	105	107	106	105	92	91	90	107	106	105
	Edge	67 more...	73 more...	64 more...	29 more...								
Windows 8.1	Chrome	109	108	107	113	112	111	95	94	93	109	108	107
	Firefox	106	105	104	110	109	108	92	91	90	106	105	104
	Opera	103	102	101	107	106	105	89	88	87	103	102	101
	Edge	73 more...	89 more...	63 more...	21 more...								
Windows 8	Chrome	109	108	107	113	112	111	95	94	93	109	108	107
	Firefox	106	105	104	110	109	108	92	91	90	106	105	104
	Opera	103	102	101	107	106	105	89	88	87	103	102	101
	Edge	73 more...	89 more...	63 more...	21 more...								
Windows 7	Chrome	109	108	107	113	112	111	90	89	88	109	108	107
	Firefox	106	105	104	110	109	108	87	85	84	106	105	104



<https://app.lambdatest.com/console/screenshot>



<https://app.lambdatest.com/console/screenshot>

Screenshot Test Results

https://app.lambdatest.com/console/screenshot/TES1016060781686185004477763

Testing URL: https://d3tests.top/fmi/webd/Contacts

Project: Untitled Version: 564

0/6 Screenshots Generated 6 Desktop 0 Mobile/Tablet

Stop Download all (PDF) Download all (Zip) Share Screenshots Back Recent History

Windows 10 Chrome 113

macOS Ventura Chrome 113

Windows 10 Firefox 113

macOS Ventura Firefox 113

e

Configure Tunnel Upgrade

Back Recent History

Message icon

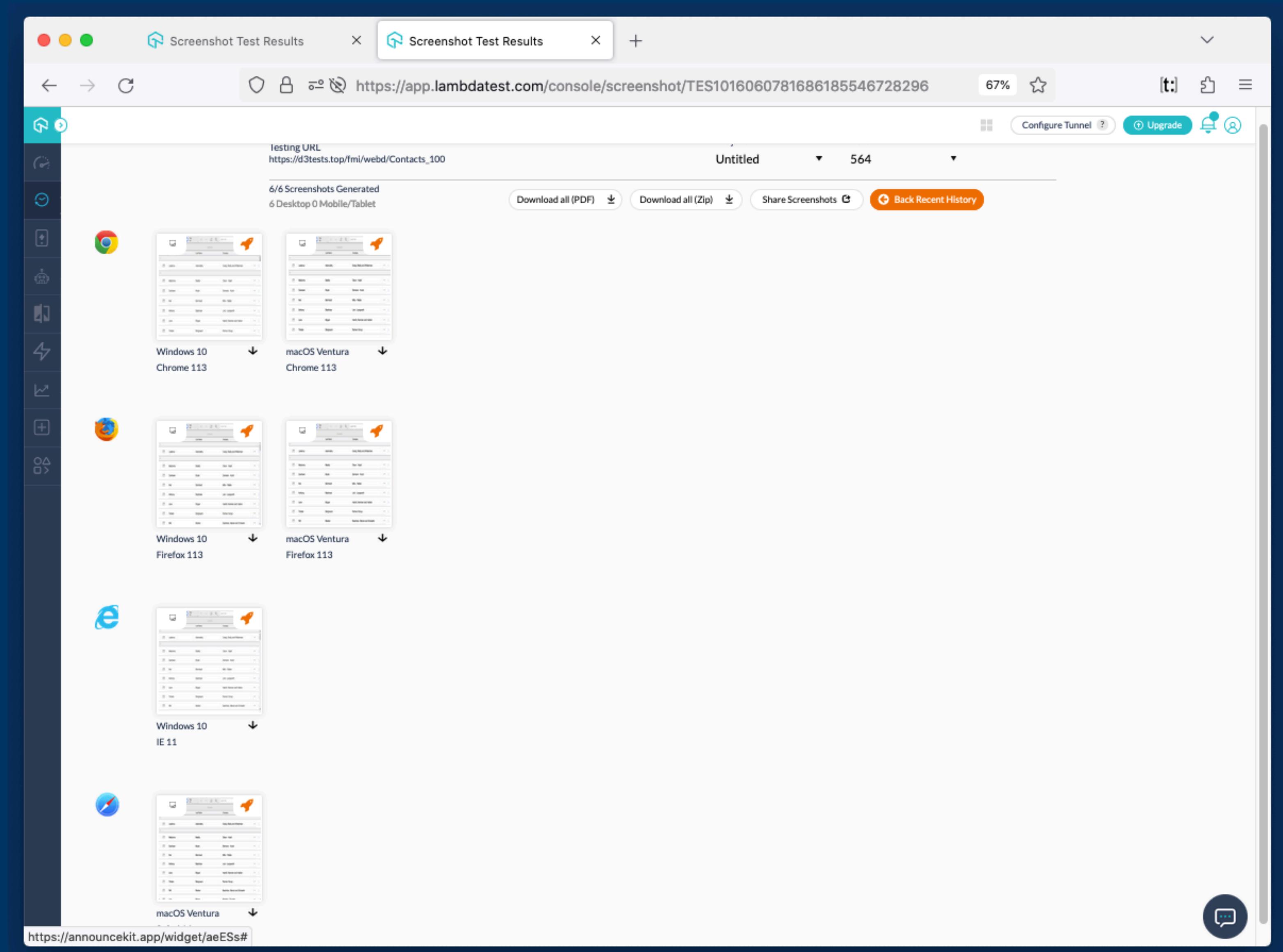
<https://app.lambdatest.com/console/screenshot/TES1016060781686185546728296>

The screenshot shows the LambdaTest console interface for a screenshot test. The URL being tested is https://d3tests.top/fmi/webd/Contacts_100. The project is Untitled and the version is 564. The interface displays six generated screenshots arranged in a 2x3 grid. Each screenshot is a comparison between two browser/OS pairs.

- Top Row:**
 - Left: Chrome 113 on Windows 10
 - Middle: Chrome 113 on macOS Ventura
 - Right: Firefox 113 on macOS Ventura
- Bottom Row:**
 - Left: Chrome 113 on Windows 10
 - Middle: Chrome 113 on macOS Ventura
 - Right: Firefox 113 on Windows 10

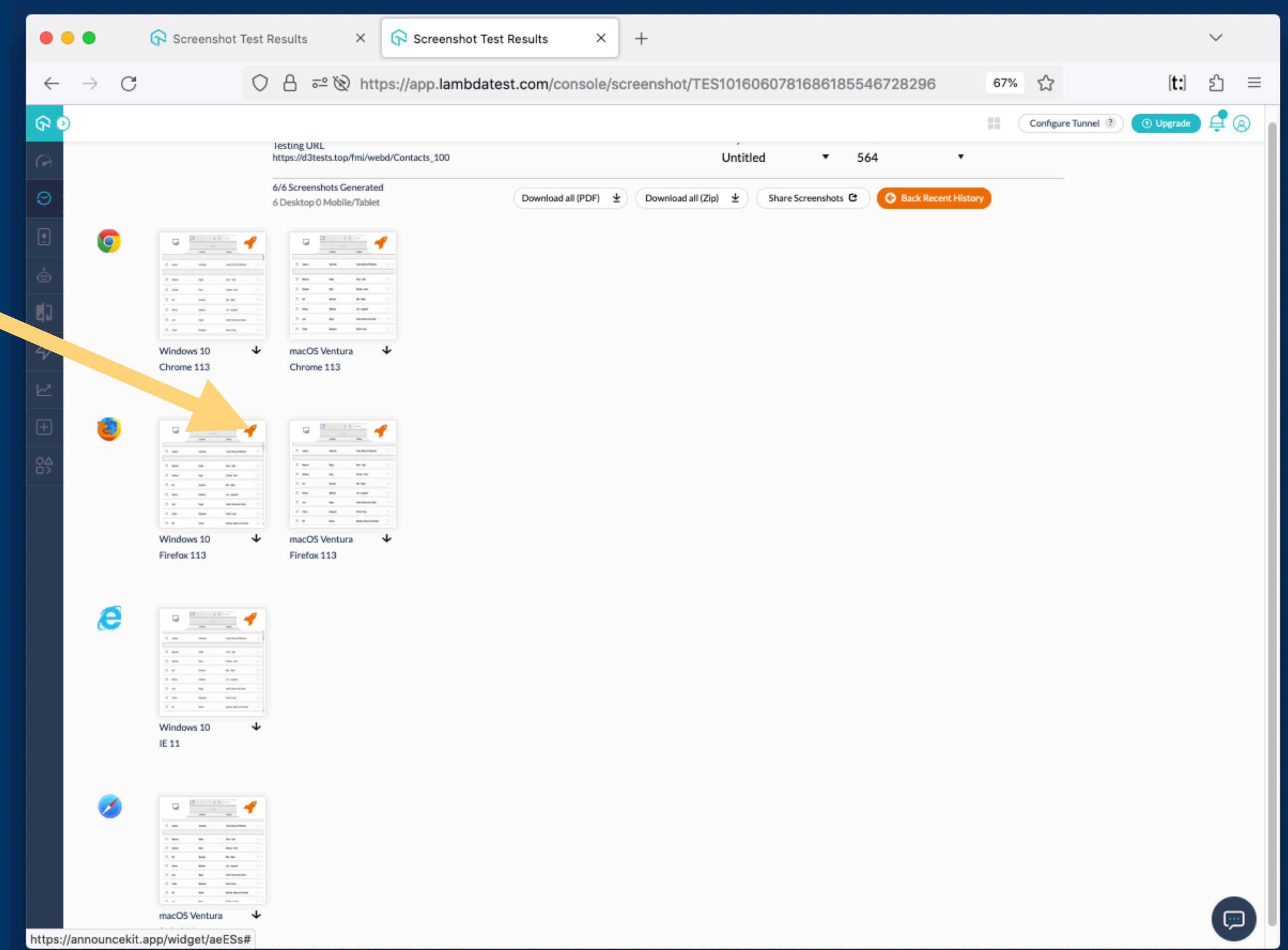
Each screenshot shows a table of contacts with columns for Name, Phone, and Email. The interface includes a sidebar with various icons for managing tests and a bottom navigation bar.

<https://app.lambdatest.com/console/screenshot/TES1016060781686185546728296>



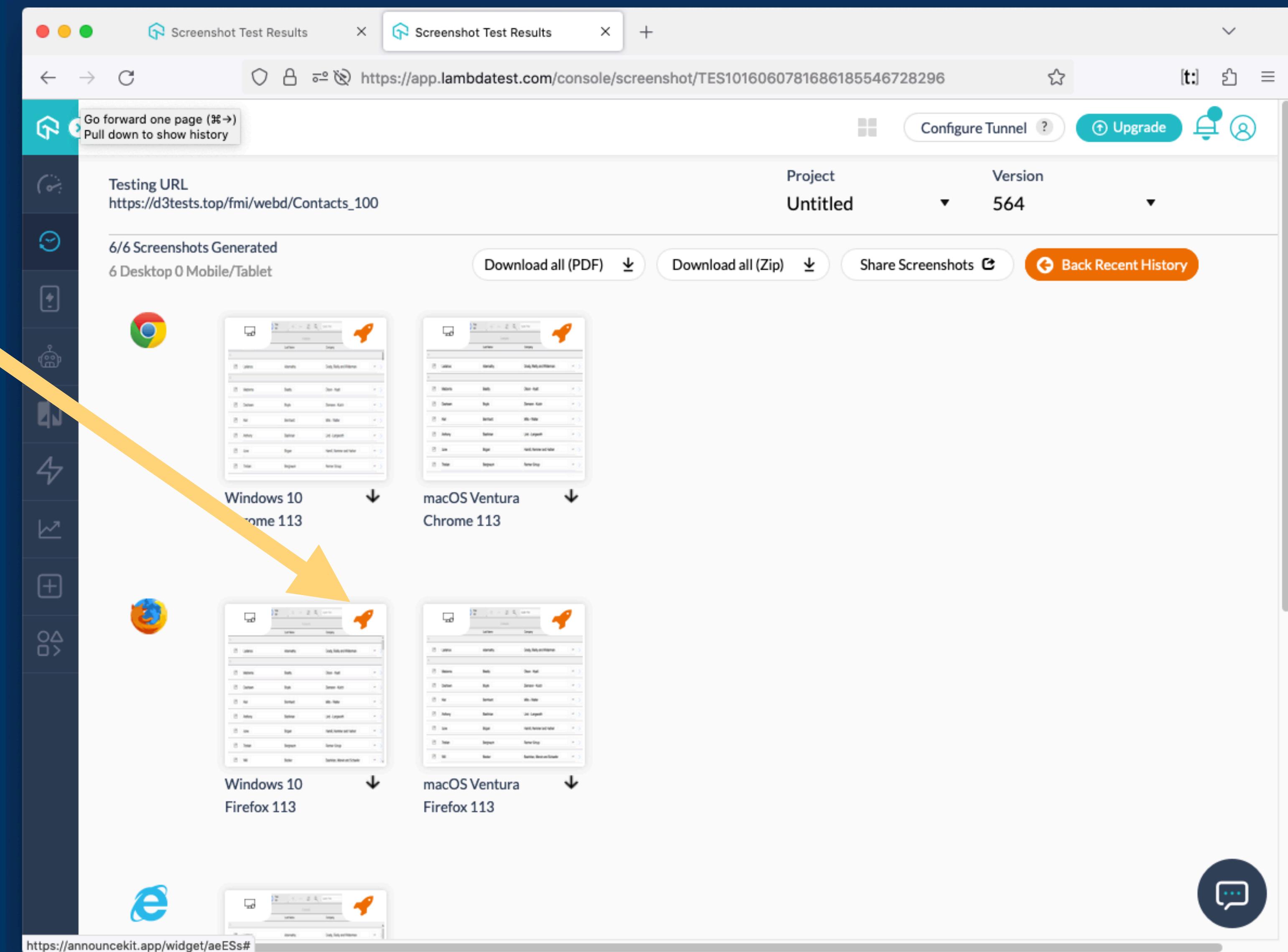
<https://app.lambdatest.com/console/screenshot/TES1016060781686185546728296>

*Click rocket icon
for
Virtual Machine*



<https://app.lambdatest.com/console/screenshot/TES1016060781686185546728296>

Click rocket icon
for
Virtual Machine



<https://app.lambdatest.com/viewer/>

Screenshot Test

Screenshot Test Results

Realtime Browser Viewer

https://app.lambdatest.com/viewer/TES1016060781686187371914663

67%

[t:]

Contacts_100 (DO-09)

https://d3tests.top/fmi/webd/Contacts_100

Total 100

First Name Last Name Company

Little	Klein	Jenkins - Breitenberg
Kevin	Kertzmann	Grady - Shields
Griffin	Kessler	Hills - Boehm
Bennett	Kshlerin	Lebsack - Boehm
Augustus	Krajcik	Hayes Inc
Krista	Kling	Reinger Group
Ralph	Larson	Ritchie, Hermiston and Hirthe
Salvatore	Lang	Veum - Johns
Waino	Lueilwitz	Larson and Sons

Activate Windows
Go to Settings to activate Windows.

9:23 PM
6/7/2023

<https://app.lambdatest.com/viewer/>

The screenshot shows a Firefox browser window with three tabs open:

- Screenshot Test
- Screenshot Test Results
- Realtime Browser Viewer

The main content area displays a contact record for "Kevin Kertzmann" from "Grady - Shields". The contact details are as follows:

First Name	Last Name	Company
Kevin	Kertzmann	Grady - Shields

Below the contact details, there is a message: "Unable to connect" followed by "An error occurred during a connection to https." A bulleted list provides troubleshooting steps:

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, [Activate Windows](#).

The browser interface includes a sidebar with icons for file operations like copy, paste, and delete, as well as a toolbar with navigation buttons and a search bar. The status bar at the bottom shows the time as 9:24 PM and the date as 6/7/2023.

<https://www.lambdatest.com/dashboard>

The screenshot shows the LambdaTest dashboard interface. On the left, there is a vertical sidebar with various icons. The main area is divided into several sections:

- REALTIME SESSIONS:** Subscribed 1 Parallel Sessions. Total Realtime Tests : 29. A blue gauge meter shows 0 Parallel Sessions. A blue "View Logs" button is present.
- AUTOMATION SESSIONS:** Subscribed 2 Parallel Sessions. Free Minutes : 0 / 100. A red gauge meter shows 0 Parallel Sessions. A red "Start Test" button is present.
- INTEGRATIONS:** Options to integrate with Jira, Slack, and Asana, each with an "INSTALL" button. There is also a "See More" link.
- RECENT TESTS:** A list of recent test runs. Each entry includes the URL, status (Live), user (patrickd3), browser (Chrome 111), and operating system (Apple 10.14).
 - URL : <https://www.staging.leonj.d3.tools>
Live | patrickd3 Chrome 111 Apple 10.14
 - URL : <https://www.staging.leonj.d3.tools>
Live | patrickd3 Chrome 111 Apple 10.14
 - URL : <https://www.leonjoosen.com>
Live | patrickd3 Chrome 111 Apple 12.0
 - URL : <https://www.staging.leonj.d3.tools>
Live | patrickd3 Chrome 111 Apple 10.14
 - URL : <https://www.leonjoosen.com>
Live | patrickd3 Chrome 111 Apple 12.0
 - URL : <https://d3tests.top/fmi/webd/Contacts>
Live | patrickd3 Chrome 111 Apple 13
- CHART:** A bar chart titled "No. Of Tests" vs "Date". It shows two bars: one at May 17 with a height of 3, and one at May 24 with a height of 7.
- NAVIGATION:** Filter dropdowns for "All Users", "Test Type", date range "May 08 2023 - Jun 07 2023", and "Day".
- RIGHT SIDE:** A sidebar with a profile picture for "Patrick Reagan", a message "Hello", and a "ADD TEAM" button. It also has "RECENT TESTS" and "RECENT ISSUES" sections.

<https://www.lambdatest.com/dashboard>

The screenshot shows the LambdaTest dashboard interface. On the left, there's a sidebar with various icons. A large yellow arrow points from the top of this sidebar towards the main content area. The main content is divided into several sections:

- REALTIME SESSIONS**: Shows 'Real Time Testing' with 29 sessions and 'Browser Testing' with 0 sessions. There's a 'View Logs' button.
- AUTOMATION SESSIONS**: Shows 'Subscribed 2 Parallel Sessions'. It has a progress bar for 'Free Minutes : 0 / 100' and another for 'Parallel Sessions' which is at 0. A red 'Start Test' button is present.
- INTEGRATIONS**: Lists Jira, Slack, and Asana with 'INSTALL' buttons. There's also a 'See More' link.
- RECENT TESTS** and **RECENT ISSUES**: A table showing test results. The first few rows are:

URL	User	Device	OS
https://www.staging.leonj.d3.tools	Live patrickd3	111	10.14
https://www.staging.leonj.d3.tools	Live patrickd3	111	10.14
https://www.leonjoosen.com	Live patrickd3	111	12.0
https://www.staging.leonj.d3.tools	Live patrickd3	111	10.14
https://www.leonjoosen.com	Live patrickd3	111	12.0
https://d3tests.top/fmi/webd/Contacts	Live patrickd3	111	13
- CHART**: A bar chart titled 'No. Of Tests' vs 'Date'. The x-axis shows dates from May 08 to Jun 07. The y-axis ranges from 0 to 7. Two bars are visible: one at May 17 (height 3) and one at May 24 (height 7).

At the bottom, there are links for 'All Live Tests' and 'All Automation Tests', along with a message icon.

<https://app.lambdatest.com/console/realtime/>

The screenshot shows the Lambdatest Realtime Browser Test interface. At the top, there are three tabs: "Screenshot Test", "Screenshot Test Results", and the active tab "Realtime Browser Test". Below the tabs is a browser header with back, forward, and refresh buttons, a URL field containing "https://app.lambdatest.com/console/realtime/browser", and a search bar.

The main area is divided into sections: "Browser Testing", "App Testing", and "VIRTUAL". The "Browser Testing" section is active. It features a "Place your URL" input field with the value "https://d3tests.top/fmi/webd/Contacts_100", an "Optional Select Tunnel" dropdown, and a large orange "START" button with a rocket icon.

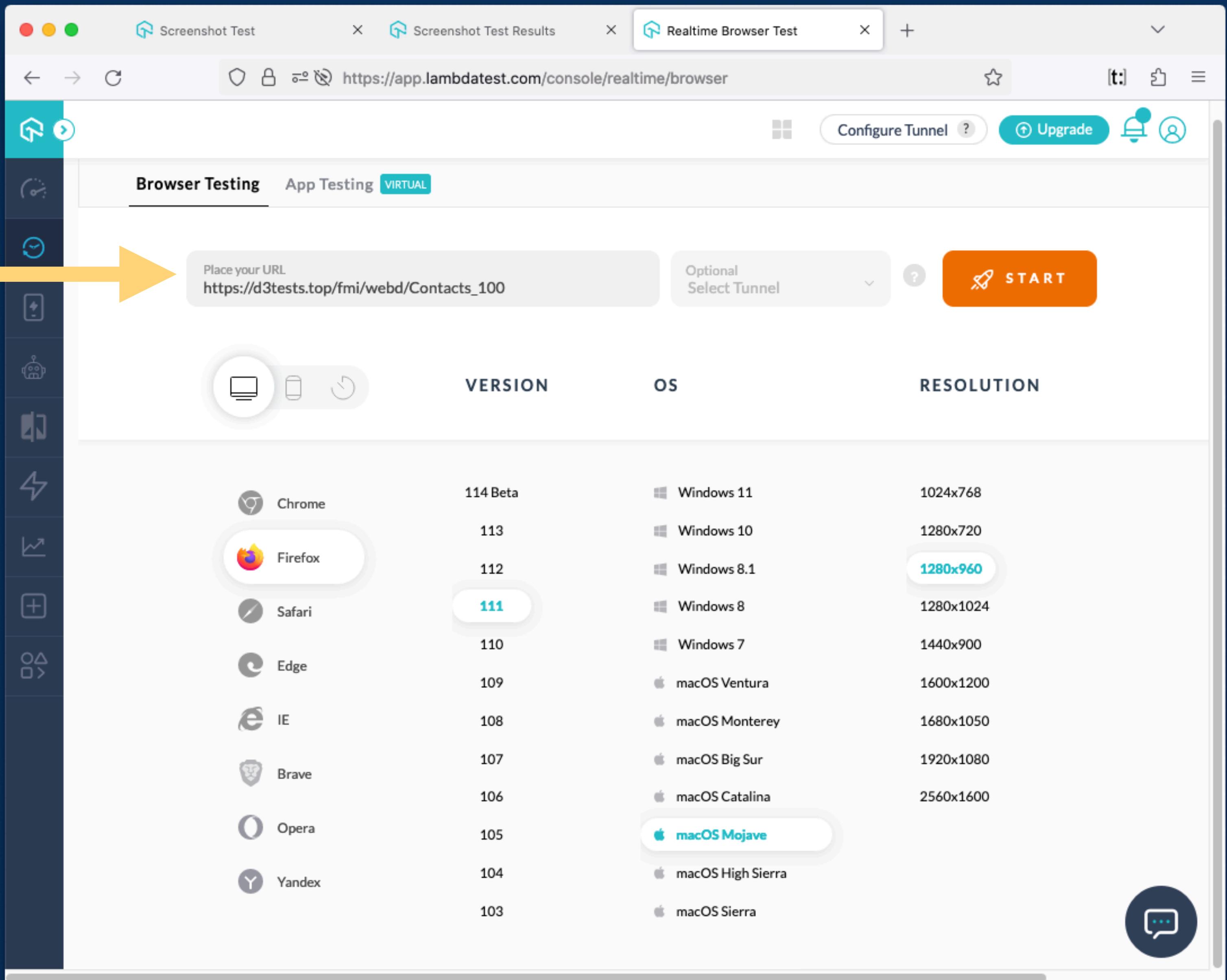
Below these controls is a table for selecting browser versions. The columns are "VERSION", "OS", and "RESOLUTION". The rows list various browsers and their versions:

VERSION	OS	RESOLUTION
Chrome 114 Beta	Windows 11	1024x768
Firefox 113	Windows 10	1280x720
Firefox 112	Windows 8.1	1280x960
Safari 111	Windows 8	1280x1024
Edge 110	Windows 7	1440x900
IE 109	macOS Ventura	1600x1200
Brave 107	macOS Monterey	1680x1050
Brave 106	macOS Big Sur	1920x1080
Opera 105	macOS Catalina	2560x1600
Yandex 104	macOS Mojave	
Yandex 103	macOS High Sierra	
Yandex 103	macOS Sierra	

A vertical sidebar on the left contains icons for Screenshot Test, Screenshot Test Results, Realtime Browser Test, and other testing tools like App Testing, Virtual, and Device Testing.

<https://app.lambdatest.com/console/realtime/>

Enter URL 



The screenshot shows the Lambdatest Realtime Browser Test interface. On the left, there's a sidebar with various icons for different testing features. The main area has tabs for "Browser Testing" (selected), "App Testing", and "VIRTUAL". A central input field says "Place your URL" with the value "https://d3tests.top/fmi/webd/Contacts_100". To the right is a "Optional Select Tunnel" dropdown and a large orange "START" button. Below these are sections for "VERSION", "OS", and "RESOLUTION". A table lists browser versions, operating systems, and resolutions. The "Firefox" row is highlighted with a rounded rectangle, and its resolution "1280x960" is also highlighted. The "macOS Mojave" row under "macOS" is also highlighted.

	VERSION	OS	RESOLUTION
Chrome	114 Beta	Windows 11	1024x768
Firefox	113	Windows 10	1280x720
Safari	112	Windows 8.1	1280x960
Edge	111	Windows 8	1280x1024
IE	110	Windows 7	1440x900
Brave	109	macOS Ventura	1600x1200
Opera	108	macOS Monterey	1680x1050
Yandex	107	macOS Big Sur	1920x1080
	106	macOS Catalina	2560x1600
		macOS Mojave	
		macOS High Sierra	
		macOS Sierra	

<https://app.lambdatest.com/console/realtime/>

Enter URL 

Choose 

Screenshot Test Screenshot Test Results Realtime Browser Test

https://app.lambdatest.com/console/realtime/browser

Configure Tunnel ? Upgrade 

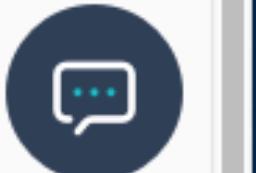
Browser Testing App Testing VIRTUAL

Place your URL
https://d3tests.top/fmi/webd/Contacts_100

Optional Select Tunnel ?

START

	VERSION	OS	RESOLUTION
Chrome	114 Beta	Windows 11	1024x768
Firefox	113	Windows 10	1280x720
Safari	112	Windows 8.1	1280x960
Edge	111	Windows 8	1280x1024
IE	110	Windows 7	1440x900
Brave	109	macOS Ventura	1600x1200
Opera	108	macOS Monterey	1680x1050
Yandex	107	macOS Big Sur	1920x1080
	106	macOS Catalina	2560x1600
		macOS Mojave	
		macOS High Sierra	
		macOS Sierra	



<https://app.lambdatest.com/console/realtime/>

The screenshot shows the Lambdatest Realtime Browser Test interface. On the left, a sidebar menu lists various testing options like Screenshot Test, Screenshot Results, and Realtime Browser Test. The main area is titled "Realtime Browser Test". It features a toolbar with "Configure Tunnel", "Upgrade", and user icons. Below the toolbar, there's a "Browser Testing" tab, an "App Testing" tab (marked as "VIRTUAL"), a URL input field containing "https://d3tests.top/fmi/webd/Contacts_100", an optional tunnel selection dropdown, and a prominent orange "START" button.

Enter URL → URL Input Field: https://d3tests.top/fmi/webd/Contacts_100

Start ← "START" Button

Choose → Browser Selection Table:

	VERSION	OS	RESOLUTION
Chrome	114 Beta	Windows 11	1024x768
Firefox	113	Windows 10	1280x720
Safari	112	Windows 8.1	1280x960
Edge	111	Windows 8	1280x1024
IE	110	Windows 7	1440x900
Brave	109	macOS Ventura	1600x1200
Opera	108	macOS Monterey	1680x1050
Yandex	107	macOS Big Sur	1920x1080
	106	macOS Catalina	2560x1600
	105	macOS Mojave	
	104	macOS High Sierra	
	103	macOS Sierra	

<https://app.lambdatest.com/viewer/>

Screenshot Test

Screenshot Test Results

Realtime Browser Viewer

https://app.lambdatest.com/viewer/TES1016060781686187371914663

67%

[t:]

Contacts_100 (DO-09)

https://d3tests.top/fmi/webd/Contacts_100

Total 100

First Name Last Name Company

Little	Klein	Jenkins - Breitenberg
Kevin	Kertzmann	Grady - Shields
Griffin	Kessler	Hills - Boehm
Bennett	Kshlerin	Lebsack - Boehm
Augustus	Krajcik	Hayes Inc
Krista	Kling	Reinger Group
Ralph	Larson	Ritchie, Hermiston and Hirthe
Salvatore	Lang	Veum - Johns
Waino	Lueilwitz	Larson and Sons

Activate Windows
Go to Settings to activate Windows.

9:23 PM
6/7/2023

<https://app.lambdatest.com/console/realtime/>

The screenshot shows the Lambdatest Realtime Browser Test interface. At the top, there are three tabs: "Screenshot Test", "Screenshot Test Results", and the active tab "Realtime Browser Test". Below the tabs is a browser header with navigation icons, a URL bar containing "https://app.lambdatest.com/console/realtime/browser", and a toolbar with "Configure Tunnel", "Upgrade", and user profile icons.

The main area is divided into sections: "Browser Testing" (selected), "App Testing", and "VIRTUAL". A central input field displays the URL "https://d3tests.top/fmi/webd/Contacts_100" and an optional tunnel selection dropdown. To the right is a large orange "START" button with a rocket icon.

A sidebar on the left contains various icons for different testing features: Screenshot, Screenshot Results, Realtime, App Testing, Virtual, Tunnel, and Help.

The main content area lists browser configurations in a table:

	VERSION	OS	RESOLUTION
Chrome	114 Beta	Windows 11	1024x768
Firefox	113	Windows 10	1280x720
Safari	111	Windows 8.1	1280x960
Edge	110	Windows 8	1280x1024
IE	108	Windows 7	1440x900
Brave	107	macOS Ventura	1600x1200
Opera	106	macOS Monterey	1680x1050
Yandex	105	macOS Big Sur	1920x1080
		macOS Catalina	2560x1600
		macOS Mojave	
		macOS High Sierra	
		macOS Sierra	

A blue callout box highlights the "1280x960" resolution entry for Safari. Another blue callout box highlights the "macOS Mojave" entry for Opera. A blue callout box also highlights the "macOS Mojave" entry for Yandex.

<https://app.lambdatest.com/console/realtime/>

History

The screenshot shows the Lambdatest Realtime Browser Test interface. On the left, there is a vertical sidebar with various icons representing different testing features. The main area is titled "Browser Testing" and includes tabs for "App Testing" and "VIRTUAL". A search bar at the top has the URL "https://d3tests.top/fmi/webd/Contacts_100" entered. Below the search bar are buttons for "Optional Select Tunnel" and a large orange "START" button with a rocket icon. The main content area displays a table of browser and OS combinations:

	VERSION	OS	RESOLUTION
Chrome	114 Beta	Windows 11	1024x768
Firefox	113	Windows 10	1280x720
Safari	112	Windows 8.1	1280x960
Edge	111	Windows 8	1280x1024
IE	110	Windows 7	1440x900
Brave	109	macOS Ventura	1600x1200
Opera	108	macOS Monterey	1680x1050
Yandex	107	macOS Big Sur	1920x1080
	106	macOS Catalina	2560x1600
		macOS Mojave	
		macOS High Sierra	
		macOS Sierra	

A yellow arrow points from the "History" section on the left towards the "Safari" row in the table.

<https://app.lambdatest.com/console/realtime/>

The screenshot shows the Lambdatest Realtime Browser Test interface. At the top, there are three tabs: "Screenshot Test", "Screenshot Test Results", and "Realtime Browser Test". The "Realtime Browser Test" tab is active. The URL in the address bar is <https://app.lambdatest.com/console/realtime/browser>. On the left, a sidebar contains various icons for different testing features. The main area has tabs for "Browser Testing", "App Testing", and "VIRTUAL". It includes fields for "Place your URL" (set to https://d3tests.top/fmi/webd/Contacts_100) and "Optional Select Tunnel". A large orange "START" button is prominent. Below these are sections for "RECENT" and "FAVOURITE" configurations. The "RECENT" section displays six recent configurations in a grid:

Date	Dimensions	OS	Browser	Count
2023-06-08T01:22:52.000Z	1024 x 768	Windows 10	Firefox	113
2023-06-08T00:49:19.000Z	1024 x 768	Windows 10	Firefox	113
2023-05-24T20:16:23.000Z	1280 x 960	Mac Mojave	Firefox	111
2023-05-24T16:48:33.000Z	1280 x 960	Mac Mojave	Firefox	111
2023-05-24T16:48:10.000Z	1440 x 900	Mac Monterey	Chrome	111

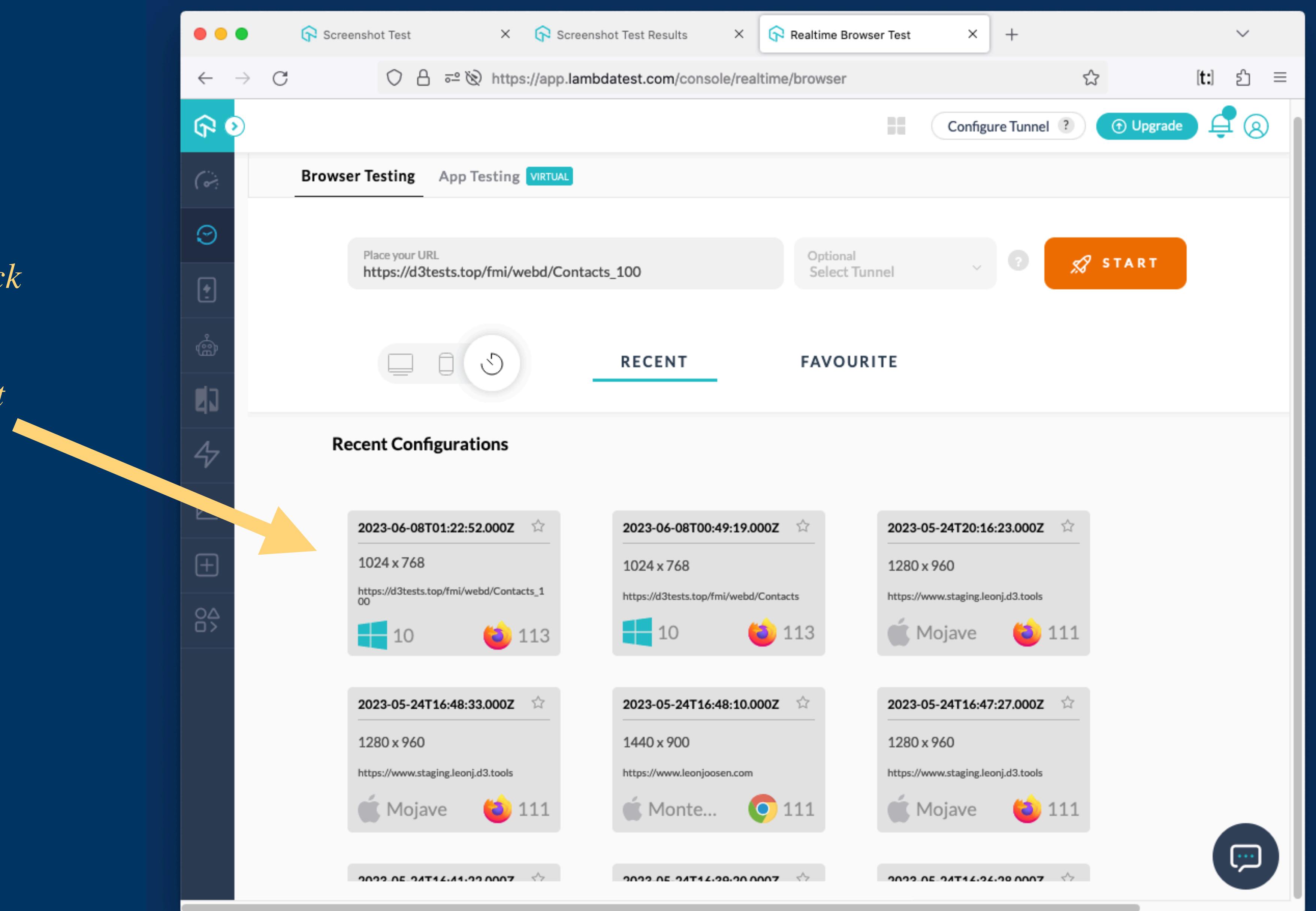
A "Recent Configurations" section at the bottom lists three more configurations:

Date	Dimensions	OS	Browser	Count
2023-05-24T16:47:27.000Z	1280 x 960	Mac Mojave	Firefox	111
2023-05-24T16:44:22.000Z	1280 x 960	Mac Mojave	Firefox	111
2023-05-24T16:28:20.000Z	1280 x 960	Mac Mojave	Firefox	111

A blue message icon is located in the bottom right corner.

<https://app.lambdatest.com/console/realtime/>

*Double click
to
repeat test*



<https://www.lambdatest.com>



*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



Artillery

*“The most advanced
load-testing platform
in the world.”*



*“The most advanced
load-testing platform
in the world.”*

*Open Source npm package
Terminal (CLI)*

- ✓ *Data API*
- ✓ *WebDirect*

<https://www.artillery.io/cloud>

The website for Artillery, a load testing platform, features a dark background with a faint, light-colored world map. At the top left is the Artillery logo with a stylized icon. The top navigation bar includes links for Documentation, Cloud, Changelog, Blog, and Careers, along with a GitHub badge showing 6.6k stars. A prominent call-to-action button at the top center reads "New! See how Auth0 optimizes cost and performance with Artillery >". The main headline, "The most advanced load-testing platform in the world", is displayed in large, bold, white font. Below it, a subtext explains the product's nature: "Artillery is cloud-native, open source, and integrates with your favorite monitoring and CI/CD stack. Load test anything, at any scale." A terminal-style code box shows the command "\$ npm install -g artillery", and a badge indicates "336k downloads / month • 100 contributors • last updated Wed, 24 May 23".

.documentation

Cloud

Changelog

Blog

Careers

New! See how Auth0 optimizes cost and performance with Artillery >

The most advanced load-testing platform in the world

Artillery is cloud-native, open source, and integrates with your favorite monitoring and CI/CD stack. Load test anything, at any scale.

```
$ npm install -g artillery
```

336k downloads / month • 100 contributors • last updated Wed, 24 May 23

Features...

- an npm application installed on your computer
- emulate user behavior with scenarios
- can have multiple steps
- run scenarios as load tests or end-to-end tests
- detailed performance metrics
- extension APIs for custom scripting in JS, and plugins
- runs on Windows, MacOS, and Linux systems

Important Concept...

- VU = Virtual User

Features...

- an npm application installed on your computer
- emulate user behavior with scenarios
- can have multiple steps
- run scenarios as load tests or end-to-end tests
- detailed performance metrics
- extension APIs for custom scripting in JS, and plugins
- runs on Windows, MacOS, and Linux systems



Important Concept...

- VU = Virtual User

Features...

- an npm application **installed on your computer**
- emulate user behavior with scenarios
- can have multiple steps
- run scenarios as load tests or end-to-end tests
- detailed performance metrics
- extension APIs for custom scripting in JS, and plugins
- runs on Windows, MacOS, and Linux systems



Important Concept...

- VU = Virtual User

Features...

- an npm application **installed on your computer**
- emulate user behavior with scenarios
- can have multiple steps
- run scenarios as load tests or end-to-end tests
- detailed performance metrics
- extension APIs for custom scripting in JS, and plugins
- runs on Windows, MacOS, and Linux systems



Important Concept...

- VU = Virtual User

- an npm application

- an npm application

npm = Node Package Manager

- an npm application
- requires Node.js

*npm = Node Package Manager
Node.js think: JavaScript for servers*

- an npm application
 - requires Node.js
 - must be most recent LTS version

*npm = Node Package Manager
Node.js think: JavaScript for servers
LTS = Long Term Support*

- an npm application
- requires Node.js
- must be most recent LTS version
- check what you have, install or upgrade

```
→ ~ node -v  
v18.15.0
```

npm = Node Package Manager
Node.js think: JavaScript for servers
LTS = Long Term Support

- an npm application
- requires Node.js
- must be most recent LTS version
- check what you have, install or upgrade

```
→ ~ node -v  
v18.15.0
```

npm = Node Package Manager
Node.js think: JavaScript for servers
LTS = Long Term Support



Downloads

Latest LTS Version: 18.16.0 (includes npm 9.5.1)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS	Current
Recommended For Most Users	Latest Features
 Windows Installer node-v18.16.0-x86.msi	 macOS Installer node-v18.16.0.pkg
 Source Code node-v18.16.0.tar.gz	

<https://nodejs.org/en/download>

- an npm application
 - requires Node.js
 - must be most recent LTS version
 - check what you have, install or upgrade
 - install Artillery, then check

```
→ ~ npm install -g artillery@latest  
...
```

npm = Node Package Manager
Node.js think: JavaScript for servers
LTS = Long Term Support

-g = global install option

- an npm application
- requires Node.js
- must be most recent LTS version
- check what you have, install or upgrade
- install Artillery, then check

```
→ ~ npm install -g artillery@latest
...
→ ~ npm artillery -v
... Artillery: 2.0.0-31 ...
```

*npm = Node Package Manager
Node.js think: JavaScript for servers
LTS = Long Term Support*

*-g = global install option
-v = show version*

- an npm application
- emulate user behavior with scenarios
- run scenarios as load tests

- an npm application
- emulate user behavior with scenarios
- run scenarios as load tests
- use .yml file to configure

.yml aka: YAML file

- commonly used for config files
- uses key: value pairs
- indentation is important

- an npm application
- emulate user behavior with scenarios
- run scenarios as load tests

- use .yml file to configure

```
→ ~ touch ~/test_1.yml
```

.yml aka: YAML file

- commonly used for config files
- uses key: value pairs
- indentation is important

touch creates empty file

Data API tests

config:

config = how to run the scenario

`config:`

config = how to run the scenario

`scenarios:`

scenarios = scenario (test steps)

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'
```

```
scenarios:
```

config = how to run the scenario
target = base url

scenarios = scenario (test steps)

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 5  
      arrivalRate: 2  
scenarios:
```

config = how to run the scenario

target = base url

phases = how often to test

scenarios = scenario (test steps)

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 5  
      arrivalRate: 2  
scenarios:  
  - flow:  
    - get:  
      url: '/records'
```

config = how to run the scenario

target = base url

phases = how often to test

scenarios = scenario (test steps)

url = specific route (required)

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 5  
      arrivalRate: 2  
scenarios:  
  - flow:  
    - get:  
      url: '/records'  
      headers:  
        content-type: 'application/json'  
        authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

config = how to run the scenario

target = base url

phases = how often to test

scenarios = scenario (test steps)

url = specific route (required)

headers: as required by Data API

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 5  
      arrivalRate: 2  
scenarios:  
  - flow:  
    - get:  
      url: '/records'  
      headers:  
        content-type: 'application/json'  
        authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

config = how to run the scenario

target = base url

phases = how often to test

scenarios = scenario (test steps)

url = specific route (required)

headers: as required by Data API

authorization: token

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 5  
      arrivalRate: 2  
scenarios:  
  - flow:  
    - get:  
      url: '/records'  
      headers:  
        content-type: 'application/json'  
        authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

config = how to run the scenario

target = base url

phases = how often to test

scenarios = scenario (test steps)

url = specific route (required)

headers: as required by Data API

authorization: token

1) FM script

Base64EncodeRFC (3548 ; "user:password")

// xxxxyyyyyyzzzz=

2) Terminal

curl -H 'Content-type: application/json' -H

'Authorization: Basic xxxxyyyyyyzzzz=' -X POST -d

'{}' https://d3tests.top/fmi/data/v1/databases/

Contacts_100/sessions

// {"response":{"token":"b3641b...b0d"}}

1) FM script

Base64EncodeRFC (3548 ; "user:password")

config = how to run the scenario

target = base url

phases = how often to test

scenarios = scenario (test steps)

url = specific route (required)

headers: as required by Data API

authorization: token

1) FM script

```
Base64EncodeRFC ( 3548 ; "user:password" )  
// xxxx yyyyzzzz=
```

config = how to run the scenario

target = base url

phases = how often to test

scenarios = scenario (test steps)

url = specific route (required)

headers: as required by Data API

authorization: token

1) FM script

```
Base64EncodeRFC ( 3548 ; "user:password" )  
    // xxxxyyyyyyzzzz=
```

2) Terminal

```
curl  
-H 'Content-type: application/json'  
-H 'Authorization: Basic xxxxyyyyyyzzzz='  
-X POST  
-d '{}'
```

https://d3tests.top/fmi/data/v1/databases/Contacts_100/sessions

config = how to run the scenario

target = base url

phases = how often to test

scenarios = scenario (test steps)

url = specific route (required)

headers: as required by Data API

authorization: token

curl = a tool to transfer data from or to a server

1) FM script

```
Base64EncodeRFC ( 3548 ; "user:password" )  
// xxxxyyyyyyzzzz=
```

2) Terminal

```
curl  
-H 'Content-type: application/json'  
-H 'Authorization: Basic xxxxyyyyyyzzzz='  
-X POST  
-d '{}'
```

https://d3tests.top/fmi/data/v1/databases/Contacts_100/sessions

```
// {"response": {"token": "b3641b...b0d"}}
```

config = how to run the scenario

target = base url

phases = how often to test

scenarios = scenario (test steps)

url = specific route (required)

headers: as required by Data API

authorization: token

curl = a tool to transfer data from or to a server

1) FM script

```
Base64EncodeRFC ( 3548 ; "user:password" )  
    // xxxxyyyyyyzzzz=
```

2) Terminal

```
curl  
-H 'Content-type: application/json'  
-H 'Authorization: Basic xxxxyyyyyyzzzz='  
-X POST  
-d '{}'
```

https://d3tests.top/fmi/data/v1/databases/Contacts_100/sessions

```
// {"response": {"token": "b3641b...b0d"}}
```

Normal 15 minute timeout holds true

config = how to run the scenario

target = base url

phases = how often to test

scenarios = scenario (test steps)

url = specific route (required)

headers: as required by Data API

authorization: token

curl = a tool to transfer data from or to a server

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 5  
      arrivalRate: 2  
scenarios:  
  - flow:  
    - get:  
      url: '/records'  
      headers:  
        content-type: 'application/json'  
        authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

config = how to run scenario

target = base url

phases = how often to test

scenarios = scenario (test steps)

headers as required by Data API

1) FM script

```
Base64EncodeRFC ( 3548 ; "user:password" )  
// xxxxyyyyyyzzzz=
```

2) Terminal

```
curl -H 'Content-type: application/json' -H  
'Authorization: Basic xxxxyyyyyyzzzz=' -X POST -d '{}'  
https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/sessions
```

```
// {"response":{"token":"b3641b...b0d"}}
```

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 5  
      arrivalRate: 2  
scenarios:  
  - flow:  
    - get:  
      url: '/records'  
      headers:  
        content-type: 'application/json'  
        authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
→ ~ artillery run artillery-FMDiSC_test_1.yml
```

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 5  
      arrivalRate: 2  
scenarios:  
  - flow:  
    - get:  
      url: '/records'  
      headers:  
        content-type: 'application/json'  
        authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
→ ~ artillery run artillery-FMDiSC_test_1.yml  
Test run id: tz5z8_6nz8yptmhwgdgnwdyk5hy7rfqjpd3m_4qaj  
Phase started: unnamed (index: 0, duration: 5s)  
14:34:09(-0700)  
· · ·
```

```
config:
  target: 'https://d3tests.top/fmi/data/v1/databases/
Contacts_100/layouts/contact_list_api'
  phases:
    - duration: 5
      arrivalRate: 2
scenarios:
  - flow:
    - get:
      url: '/records'
      headers:
        content-type: 'application/json'
        authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
→ ~ artillery run artillery-FMDiSC_test_1.yml
Test run id: tz5z8_6nz8yptmhwgdgnwdyk5hy7rfqjpd3m_4qaj
Phase started: unnamed (index: 0, duration: 5s)
14:34:09(-0700)
. . .

-----
Metrics for period to: 14:34:20(-0700) (width:
4.135s)
-----
http.codes.200: ..... 10
http.request_rate: ..... 2/sec
http.requests: ..... 10
. . .
```

```
config:
  target: 'https://d3tests.top/fmi/data/v1/databases/
Contacts_100/layouts/contact_list_api'
  phases:
    - duration: 5
      arrivalRate: 2
scenarios:
  - flow:
    - get:
      url: '/records'
      headers:
        content-type: 'application/json'
        authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
→ ~ artillery run artillery-FMDiSC_test_1.yml
Test run id: tz5z8_6nz8yptmhwgdgnwdyk5hy7rfqjpd3m_4qaj
Phase started: unnamed (index: 0, duration: 5s)
14:34:09(-0700)
...
-----
```

```
Metrics for period to: 14:34:20(-0700) (width:
4.135s)
```

```
http.codes.200: ..... 10
http.request_rate: ..... 2/sec
http.requests: ..... 10
...
-----
```

```
All VUs finished. Total time: 6 seconds
```

```
Summary report @ 14:34:17(-0700)
-----
```

```
http.codes.200: ..... 10
http.request_rate: ..... 2/sec
http.requests: ..... 10
http.response_time:
  min: ..... 25
  max: ..... 44
  median: ..... 26.8
  p95: ..... 39.3
  p99: ..... 39.3
...
-----
```

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 5  
      arrivalRate: 2  
scenarios:  
  - flow:  
    - get:  
      url: '/records'  
      headers:  
        content-type: 'application/json'  
        authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 120  
      arrivalRate: 20  
scenarios:  
  - flow:  
    - get:  
      url: '/records'  
      headers:  
        content-type: 'application/json'  
        authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 120  
      arrivalRate: 20  
      name: 20 per second  
scenarios:  
  - flow:  
    - get:  
      url: '/records'  
      headers:  
        content-type: 'application/json'  
        authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

name = identify each phase for reporting

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 120  
      arrivalRate: 20  
      name: 20 per second  
    - duration: 120  
      arrivalRate: 40  
      name: 40 per second  
  scenarios:  
    - flow:  
      - get:  
        url: '/records'  
        headers:  
          content-type: 'application/json'  
          authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

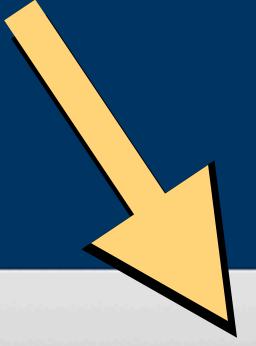
name = identify each phase for reporting

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 120  
      arrivalRate: 20  
      name: 20 per second  
    - duration: 120  
      arrivalRate: 40  
      name: 40 per second  
    . . .  
  scenarios:  
    - flow:  
      - get:  
        url: '/records'  
        headers:  
          content-type: 'application/json'  
          authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

name = identify each phase for reporting

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 120  
      arrivalRate: 20  
      name: 20 per second  
    - duration: 120  
      arrivalRate: 40  
      name: 40 per second  
    . . .  
    - duration: 120  
      arrivalRate: 200  
      name: 200 per second  
  scenarios:  
    - flow:  
      - get:  
        url: '/records'  
        headers:  
          content-type: 'application/json'  
          authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

name = identify each phase for reporting



Contacts_100

100 / 100 Found (Sorted)

Layout: contact_list_api | View As: | Preview | Aa | Edit Layout

Ladarius	Abernathy	Grady, Reilly and Wilderman
Madonna	Beatty	Olson - Hyatt
Dashawn	Boyle	Ziemann - Kutch
Kiel	Bernhard	Mills - Walter
Anthony	Bashirian	Lind - Langworth
June	Bogan	Hamill, Kemmer and Harber
Tristian	Bergnaum	Renner Group
Will	Becker	Bashirian, Marvin and Schaefer
Lina	Brown	Stracke - Douglas
Carmela	Bartell	Hand - Macejkovic

Claris FileMaker 2023 Technical Specifications

FileMaker Server on Linux - Primary Machine

Minimum	Better	Best
CPU: 4 Core	CPU: 8 Core	CPU: 16 Core
RAM: 8 GB or more	RAM: 16 GB or more	RAM: 32 GB or more
Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive space.	Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive space.	Hard Drive: 1 TB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive space.
Network: Gigabit Ethernet	Network: Gigabit Ethernet	Network: Gigabit Ethernet
Number of WebDirect connections: 50	Number of WebDirect connections: 100	Number of WebDirect connections: 120



Claris FileMaker 2023 Technical Specifications

FileMaker Server on Linux - Primary Machine

Minimum	Better	Best
CPU: 4 Core	CPU: 8 Core	CPU: 16 Core
RAM: 8 GB or more	RAM: 16 GB or more	RAM: 32 GB or more
Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive space.	Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive space.	Hard Drive: 1 TB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive space.
Network: Gigabit Ethernet	Network: Gigabit Ethernet	Network: Gigabit Ethernet
Number of WebDirect connections: 50	Number of WebDirect connections: 100	Number of WebDirect connections: 120



DO-09-v6

in ppreagan / 4 GB Memory / 80 GB Disk / SFO3 - Ubuntu DO-09 v6 2CPU, 4GB, FMS...

ON

ipv4: 143.198.128.107

ipv6: [Enable now](#)

Private IP: 10.124.0.2

Reserved IP: [Enable now](#)

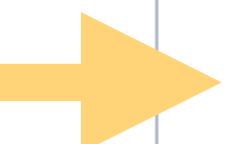
Console:

Graphs
Access
Power
Volumes
Resize
Backups
Snapshots

Resize Droplet

This Droplet is on a Basic plan. You must [turn off your Droplet](#) to resize it.

Currently using: Basic / 4 GB / 2 vCPUs



Claris FileMaker 2023 Technical Specifications

FileMaker Server on Linux - Primary Machine

Minimum	Better	Best
CPU: 4 Core	CPU: 8 Core	CPU: 16 Core
RAM: 8 GB or more	RAM: 16 GB or more	RAM: 32 GB or more
Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive space.	Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive space.	Hard Drive: 1 TB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive space.
Network: Gigabit Ethernet	Network: Gigabit Ethernet	Network: Gigabit Ethernet
Number of WebDirect connections: 50	Number of WebDirect connections: 100	Number of WebDirect connections: 120



DO-09-v6

in ppreagan / 4 GB Memory / 80 GB Disk / SFO3 - Ubuntu DO-09 v6 2CPU, 4GB, FMS...

ON

ipv4: 143.198.128.107

ipv6: [Enable now](#)

Private IP: 10.124.0.2

Reserved IP: [Enable now](#)

Console: ?

Graphs

Access

Power

Volumes

Resize

Backups

Snapshots

Resize Droplet

This Droplet is on a Basic plan. You must [turn off your Droplet](#) to resize it.

Currently using: Basic / 4 GB / 2 vCPUs

\$24/month vs \$48/month



Claris FileMaker 2023 Technical Specifications

FileMaker Server on Linux - Primary Machine

Minimum

CPU: 4 Core

RAM: 8 GB or more

Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive

Better

CPU: 8 Core

RAM: 16 GB or more

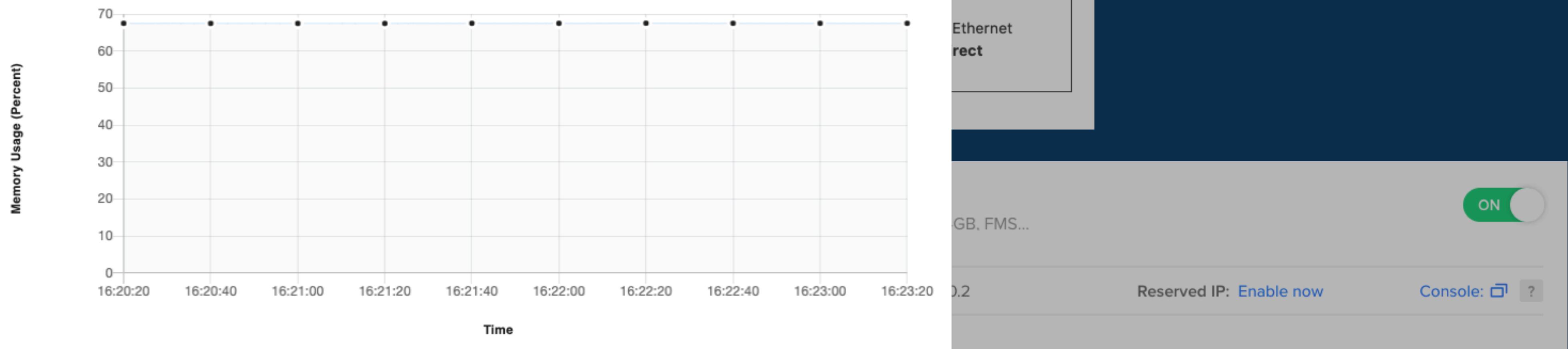
Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive

Best

CPU: 16 Core

RAM: 32 GB or more

Hard Drive: 1 TB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive



Graphs
Access
Power
Volumes
Resize

Backups
Snapshots

Resize Droplet

This Droplet is on a Basic plan. You must [turn off your Droplet](#) to resize it.

Currently using: Basic / 4 GB / 2 vCPUs

\$24/month vs \$48/month

Claris FileMaker 2023 Technical Specifications

FileMaker Server on Linux - Primary Machine

Minimum

CPU: 4 Core

RAM: 8 GB or more

Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive

Better

CPU: 8 Core

RAM: 16 GB or more

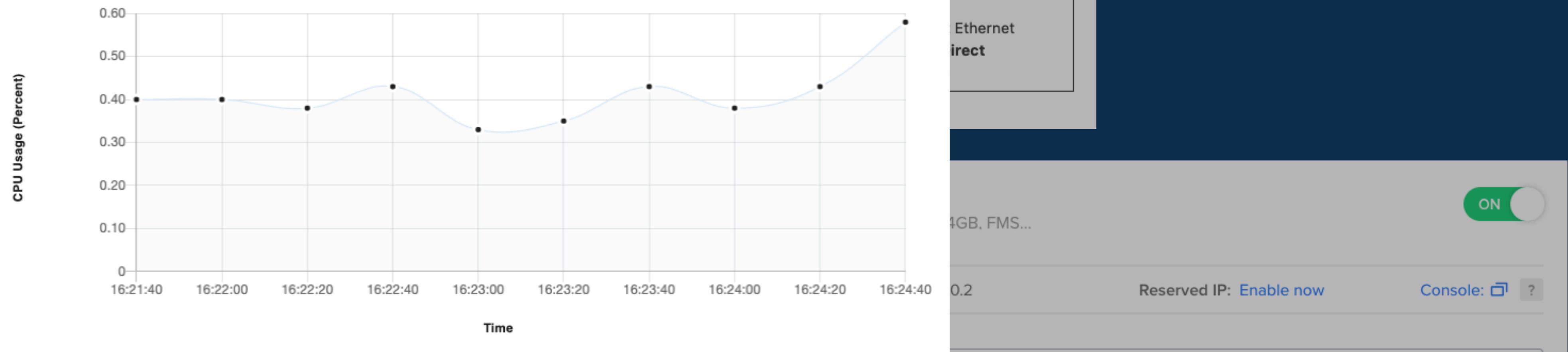
Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive

Best

CPU: 16 Core

RAM: 32 GB or more

Hard Drive: 1 TB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive



Graphs
Access
Power
Volumes
Resize

Backups
Snapshots

Resize Droplet

This Droplet is on a Basic plan. You must [turn off your Droplet](#) to resize it.

Currently using: Basic / 4 GB / 2 vCPUs

\$24/month vs \$48/month

Claris FileMaker 2023 Technical Specifications

FileMaker Server on Linux - Primary Machine

Minimum

CPU: 4 Core
RAM: 8 GB or more
Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive

Better

CPU: 8 Core
RAM: 16 GB or more
Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive

Best

CPU: 16 Core
RAM: 32 GB or more
Hard Drive: 1 TB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive



```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 120  
      arrivalRate: 20  
      name: 20 per second  
    - duration: 120  
      arrivalRate: 40  
      name: 40 per second  
    . . .  
    - duration: 120  
      arrivalRate: 200  
      name: 200 per second  
  
scenarios:  
  - flow:  
    - get:  
      url: '/records'  
      headers:  
        content-type: 'application/json'  
        authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

→ ~

```
config:
  target: 'https://d3tests.top/fmi/data/v1/databases/
Contacts_100/layouts/contact_list_api'
phases:
  - duration: 120
    arrivalRate: 20
    name: 20 per second
  - duration: 120
    arrivalRate: 40
    name: 40 per second
  . . .
  - duration: 120
    arrivalRate: 200
    name: 200 per second

scenarios:
  - flow:
    - get:
      url: '/records'
      headers:
        content-type: 'application/json'
        authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
→ ~ artillery run artillery-FMDiSC_test_2.yml
```

```
config:
  target: 'https://d3tests.top/fmi/data/v1/databases/
Contacts_100/layouts/contact_list_api'
  phases:
    - duration: 120
      arrivalRate: 20
      name: 20 per second
    - duration: 120
      arrivalRate: 40
      name: 40 per second
    . . .
    - duration: 120
      arrivalRate: 200
      name: 200 per second

scenarios:
  - flow:
    - get:
        url: '/records'
        headers:
          content-type: 'application/json'
          authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

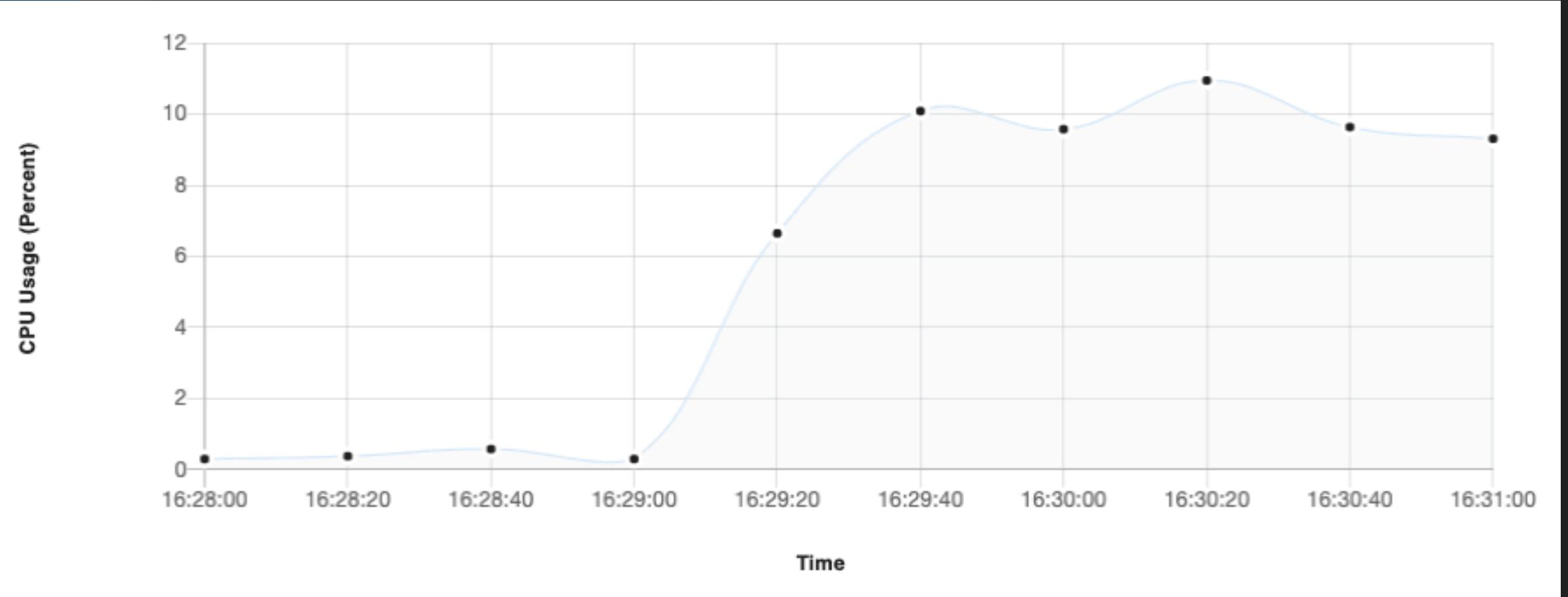
```
→ ~ artillery run artillery-FMDiSC_test_2.yml
Test run id: twnzy_xqwat8b3kkjxykgg3enac5wmyz769_qw3z
Phase started: 20 per second (index: 0, duration:
120s) 16:29:06(-0700)
. . .
```

```
config:
  target: 'https://d3tests.top/fmi/data/v1/databases/
Contacts_100/layouts/contact_list_api'
  phases:
    - duration: 120
      arrivalRate: 20
      name: 20 per second
    - duration: 120
      arrivalRate: 40
      name: 40 per second
    ...
    - duration: 120
      arrivalRate: 200
      name: 200 per second

scenarios:
  - flow:
    - get:
        url: '/records'
        headers:
          content-type: 'application/json'
          authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
→ ~ artillery run artillery-FMDiSC_test_2.yml
Test run id: twnzy_xqwat8b3kkjxykgg3enac5wmyz769_qw3z
Phase started: 20 per second (index: 0, duration:
120s) 16:29:06(-0700)
...
-----
Metrics for period to: 16:29:10(-0700) (width: 2.72s)
-----
http.codes.200: ..... 53
http.request_rate: ..... 21/sec
http.requests: ..... 55
```

```
config:
  target: 'https://d3tests.top/fmi/data/v1/databases/
Contacts_100/layouts/contact_list_api'
  phases:
    - duration: 120
      arrivalRate: 20
      name: 20 per second
    - duration: 120
```



```
headers:
  content-type: 'application/json'
  authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
→ ~ artillery run artillery-FMDiSC_test_2.yml
Test run id: twnzy_xqwat8b3kkjxykgg3enac5wmyz769_qw3z
Phase started: 20 per second (index: 0, duration:
120s) 16:29:06(-0700)
...
```

```
-----  
Metrics for period to: 16:29:10(-0700) (width: 2.72s)  
-----
```

```
http.codes.200: ..... 53
http.request_rate: ..... 21/sec
http.requests: ..... 55
```

```
config:
  target: 'https://d3tests.top/fmi/data/v1/databases/
Contacts_100/layouts/contact_list_api'
  phases:
    - duration: 120
      arrivalRate: 20
      name: 20 per second
    - duration: 120
      arrivalRate: 40
      name: 40 per second
    ...
    - duration: 120
      arrivalRate: 200
      name: 200 per second

scenarios:
  - flow:
    - get:
        url: '/records'
        headers:
          content-type: 'application/json'
          authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
→ ~ artillery run artillery-FMDiSC_test_2.yml
Test run id: twnzy_xqwat8b3kkjxykgg3enac5wmyz769_qw3z
Phase started: 20 per second (index: 0, duration:
120s) 16:29:06(-0700)
. . .
. . .

Phase started: 40 per second (index: 1, duration:
120s) 16:31:06(-0700)

-----
Metrics for period to: 16:31:10(-0700) (width:
9.982s)
-----
errors.ETIMEDOUT: ..... 1
http.codes.200: ..... 256
http.request_rate: ..... 27/sec
http.requests: ..... 259
```

```
config:
  target: 'https://d3tests.top/fmi/data/v1/databases/
Contacts_100/layouts/contact_list_api'
  phases:
    - duration: 120
      arrivalRate: 20
      name: 20 per second
    - duration: 120
      arrivalRate: 40
      name: 40 per second
    ...
    - duration: 120
      arrivalRate: 200
      name: 200 per second

  scenarios:
    - flow:
        - get:
            url: '/records'
            headers:
              content-type: 'application/json'
              authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
→ ~ artillery run artillery-FMDiSC_test_2.yml
Test run id: twnzy_xqwat8b3kkjxykgg3enac5wmyz769_qw3z
Phase started: 20 per second (index: 0, duration:
120s) 16:29:06(-0700)
. . .
. . .

Phase started: 40 per second (index: 1, duration:
120s) 16:31:06(-0700)

-----
Metrics for period to: 16:31:10(-0700) (width:
9.982s)
-----
errors.ETIMEDOUT: ..... 1
http.codes.200: ..... 256
http.request_rate: ..... 27/sec
http.requests: ..... 259
```

```
config:
  target: 'https://d3tests.top/fmi/data/v1/databases/
Contacts_100/layouts/contact_list_api'
  phases:
    - duration: 120
      arrivalRate: 20
      name: 20 per second
    - duration: 120
      arrivalRate: 40
      name: 40 per second
    . . .
    - duration: 120
      arrivalRate: 200
      name: 200 per second

  scenarios:
    - flow:
        - get:
            url: '/records'
            headers:
              content-type: 'application/json'
              authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
→ ~ artillery run artillery-FMDiSC_test_2.yml
Test run id: twnzy_xqwat8b3kkjxykgg3enac5wmyz769_qw3z
Phase started: 20 per second (index: 0, duration:
120s) 16:29:06(-0700)
. . .
. . .
. . .

-----
Metrics for period to: 16:44:40(-0700) (width:
9.991s)
-----
errors.ENOTFOUND: ..... 6
http.codes.200: ..... 1597
http.request_rate: ..... 160/sec
http.requests: ..... 1600
```

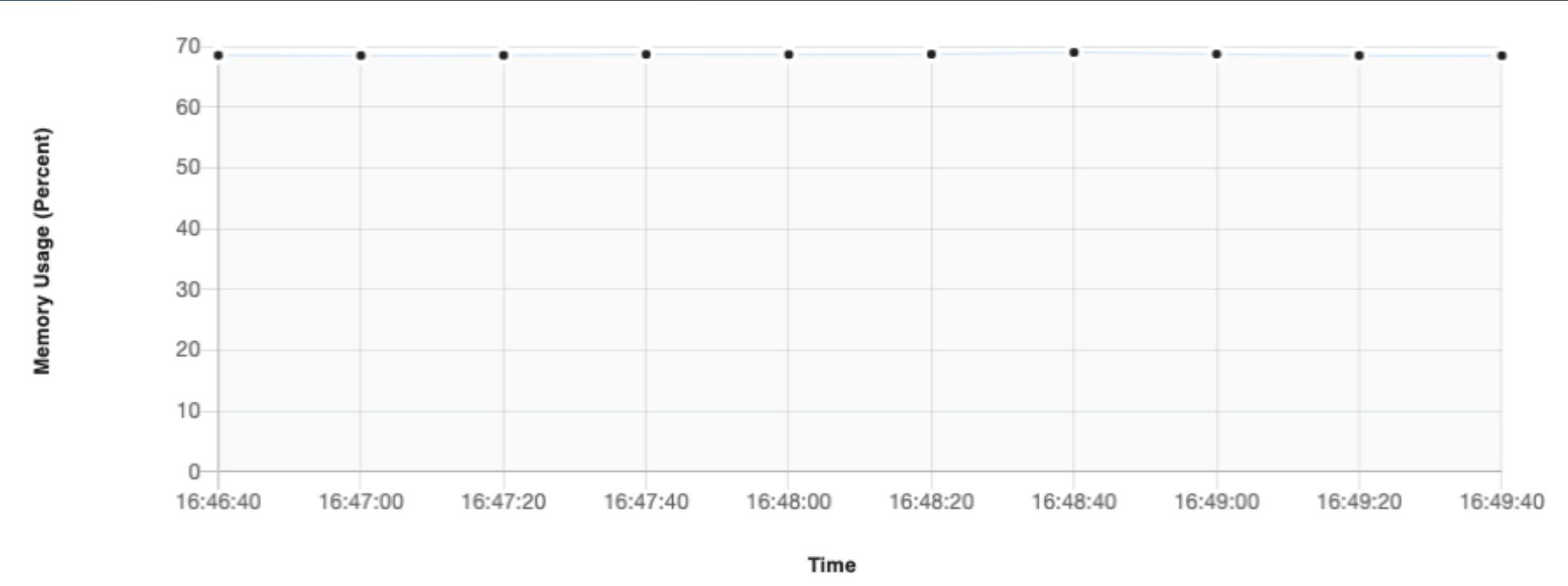
```
config:
  target: 'https://d3tests.top/fmi/data/v1/databases/
Contacts_100/layouts/contact_list_api'
  phases:
    - duration: 120
      arrivalRate: 20
      name: 20 per second
    - duration: 120
      arrivalRate: 40
      name: 40 per second
    . . .
    - duration: 120
      arrivalRate: 200
      name: 200 per second

scenarios:
  - flow:
    - get:
        url: '/records'
        headers:
          content-type: 'application/json'
          authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
→ ~ artillery run artillery-FMDiSC_test_2.yml
Test run id: twnzy_xqwat8b3kkjxykgg3enac5wmyz769_qw3z
Phase started: 20 per second (index: 0, duration:
120s) 16:29:06(-0700)
. . .
. . .
. . .
. . .

All VUs finished. Total time: 20 minutes, 1 second
-----
Summary report @ 16:49:08(-0700)
-----
errors.ENOTFOUND: ..... 93
errors.ETIMEDOUT: ..... 16
http.codes.200: ..... 131891
http.request_rate: ..... 110/sec
http.requests: ..... 132000
```

```
config:
  target: 'https://d3tests.top/fmi/data/v1/databases/
Contacts_100/layouts/contact_list_api'
  phases:
    - duration: 120
      arrivalRate: 20
      name: 20 per second
    - duration: 120
```



```
headers:
  content-type: 'application/json'
  authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
→ ~ artillery run artillery-FMDiSC_test_2.yml
Test run id: twnzy_xqwat8b3kkjxykgg3enac5wmyz769_qw3z
Phase started: 20 per second (index: 0, duration:
120s) 16:29:06(-0700)
```

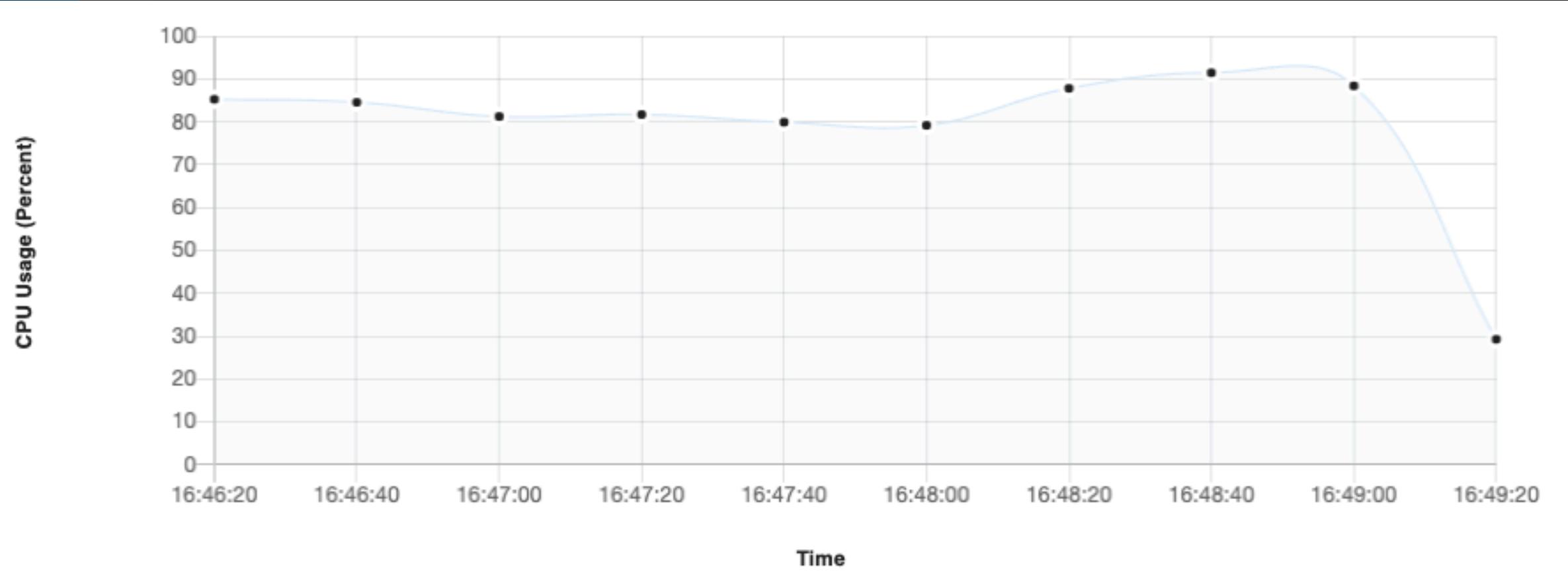
```
• • •  
• • •  
• • •  
• • •
```

All VUs finished. Total time: 20 minutes, 1 second

Summary report @ 16:49:08(-0700)

errors.ENOTFOUND:	93
errors.ETIMEDOUT:	16
http.codes.200:	131891
http.request_rate:	110/sec
http.requests:	132000

```
config:
  target: 'https://d3tests.top/fmi/data/v1/databases/
Contacts_100/layouts/contact_list_api'
  phases:
    - duration: 120
      arrivalRate: 20
      name: 20 per second
    - duration: 120
```



```
headers:
  content-type: 'application/json'
  authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
→ ~ artillery run artillery-FMDiSC_test_2.yml
Test run id: twnzy_xqwat8b3kkjxykgg3enac5wmyz769_qw3z
Phase started: 20 per second (index: 0, duration:
120s) 16:29:06(-0700)
```

```
• • •  
• • •  
• • •  
• • •
```

```
All VUs finished. Total time: 20 minutes, 1 second
```

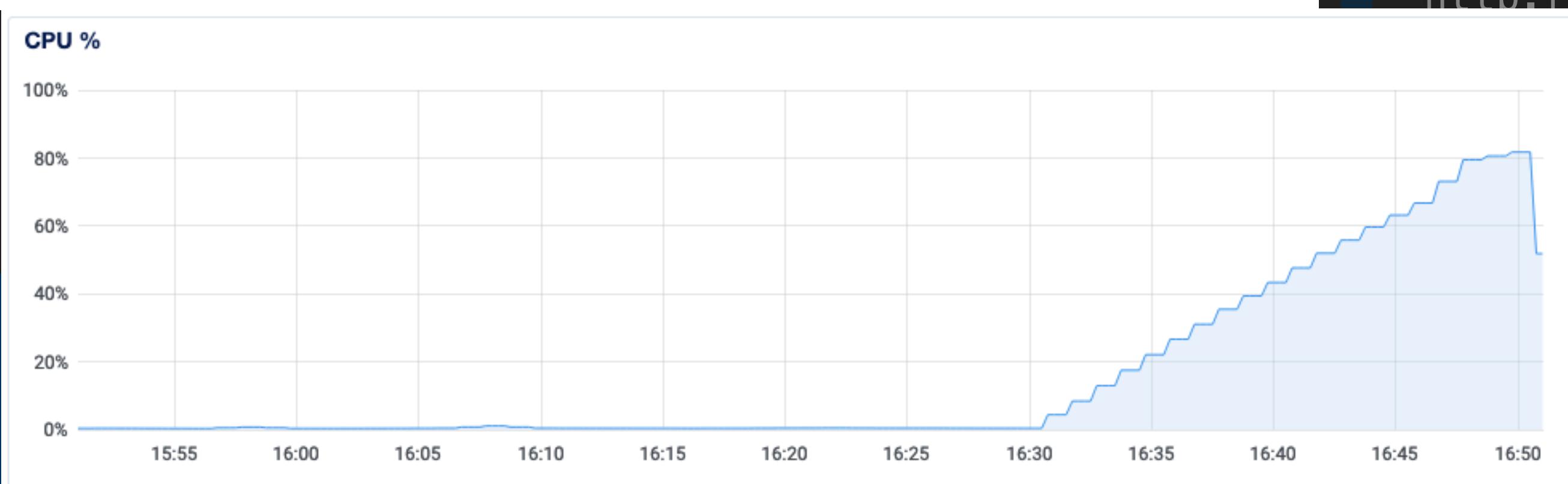
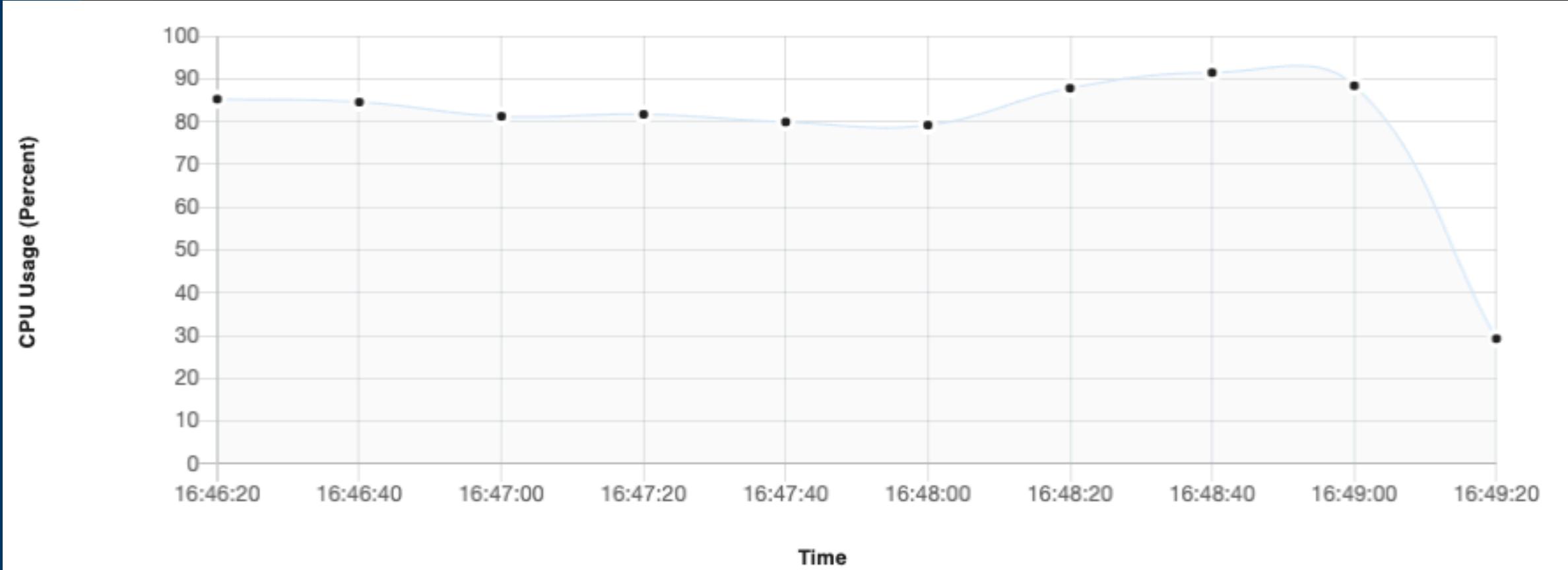
```
Summary report @ 16:49:08(-0700)
```

errors.ENOTFOUND:	93
errors.ETIMEDOUT:	16
http.codes.200:	131891
http.request_rate:	110/sec
http.requests:	132000

```

config:
  target: 'https://d3tests.top/fmi/data/v1/databases/
Contacts_100/layouts/contact_list_api'
  phases:
    - duration: 120
      arrivalRate: 20
      name: 20 per second
    - duration: 120

```



→ ~ artillery run artillery-FMDiSC_test_2.yml
Test run id: twnzy_xqwat8b3kkjxykkg3enac5wmyz769_qw3z
Phase started: 20 per second (index: 0, duration:
120s) 16:29:06(-0700)

• • •
• • •
• • •
• • •

All VUs finished. Total time: 20 minutes, 1 second

Summary report @ 16:49:08(-0700)

errors.ENOTFOUND:	93
errors.ETIMEDOUT:	16
http.codes.200:	131891
http.request_rate:	110/sec
quests:	132000

Load Testing

“...generally refers to the practice of modeling the expected usage of a software program by simulating multiple users accessing the program concurrently.”

https://en.wikipedia.org/wiki/Load_testing

Smoke Testing

Load Testing

“...generally refers to the practice of modeling the expected usage of a software program by simulating multiple users accessing the program concurrently.”

https://en.wikipedia.org/wiki/Load_testing

“How will things work under expected load?”



Smoke Testing

Load Testing

“...generally refers to the practice of modeling the expected usage of a software program by simulating multiple users accessing the program concurrently.”

https://en.wikipedia.org/wiki/Load_testing

“How will things work under expected load?”



Smoke Testing

“..preliminary testing to reveal simple failures severe enough to, for example, reject a prospective software release”

[https://en.wikipedia.org/wiki/Smoke_testing_\(software\)](https://en.wikipedia.org/wiki/Smoke_testing_(software))

Load Testing

“...generally refers to the practice of modeling the expected usage of a software program by simulating multiple users accessing the program concurrently.”

https://en.wikipedia.org/wiki/Load_testing

“How will things work under expected load?”



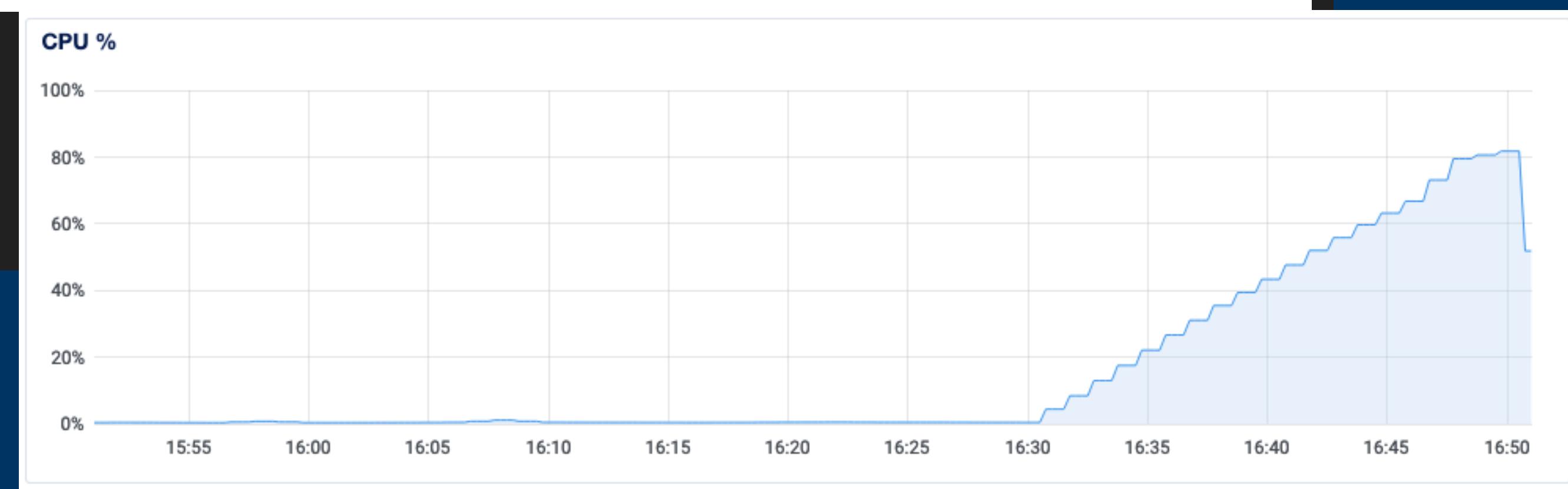
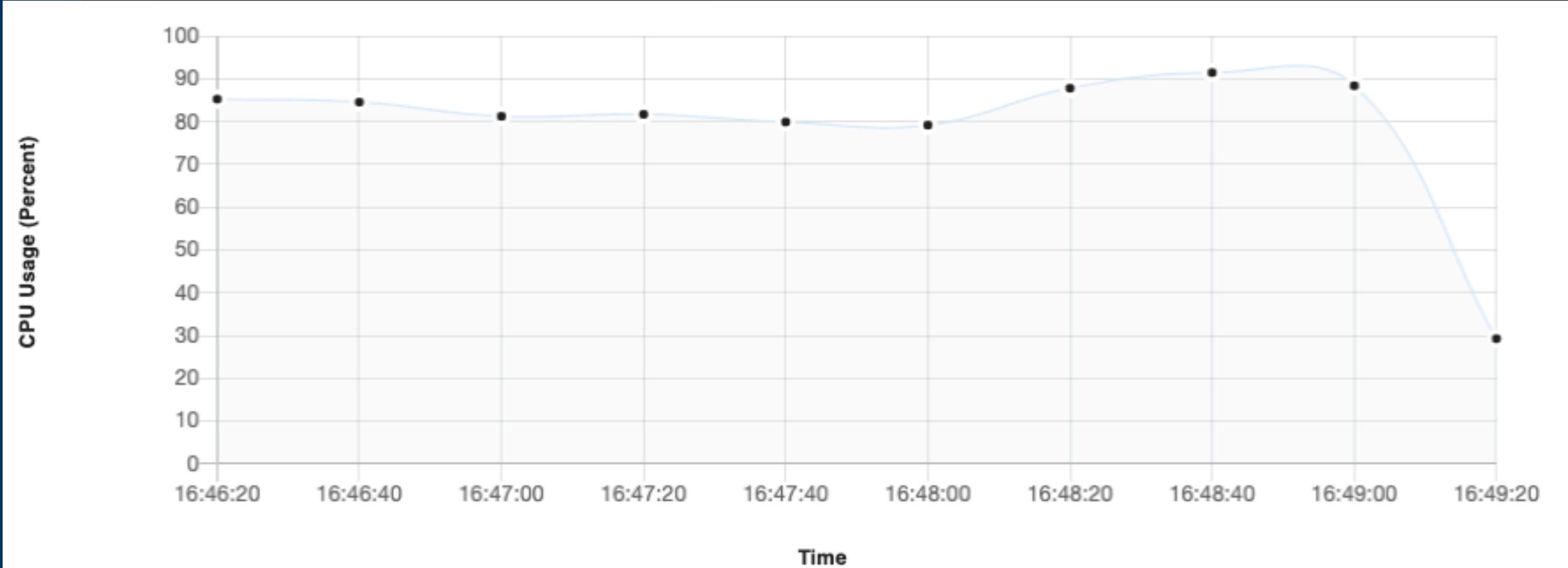
Smoke Testing

“..preliminary testing to reveal simple failures severe enough to, for example, reject a prospective software release”

[https://en.wikipedia.org/wiki/Smoke_testing_\(software\)](https://en.wikipedia.org/wiki/Smoke_testing_(software))

“How much load will break it?”

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 120  
      arrivalRate: 20  
      name: 20 per second  
    - duration: 120
```

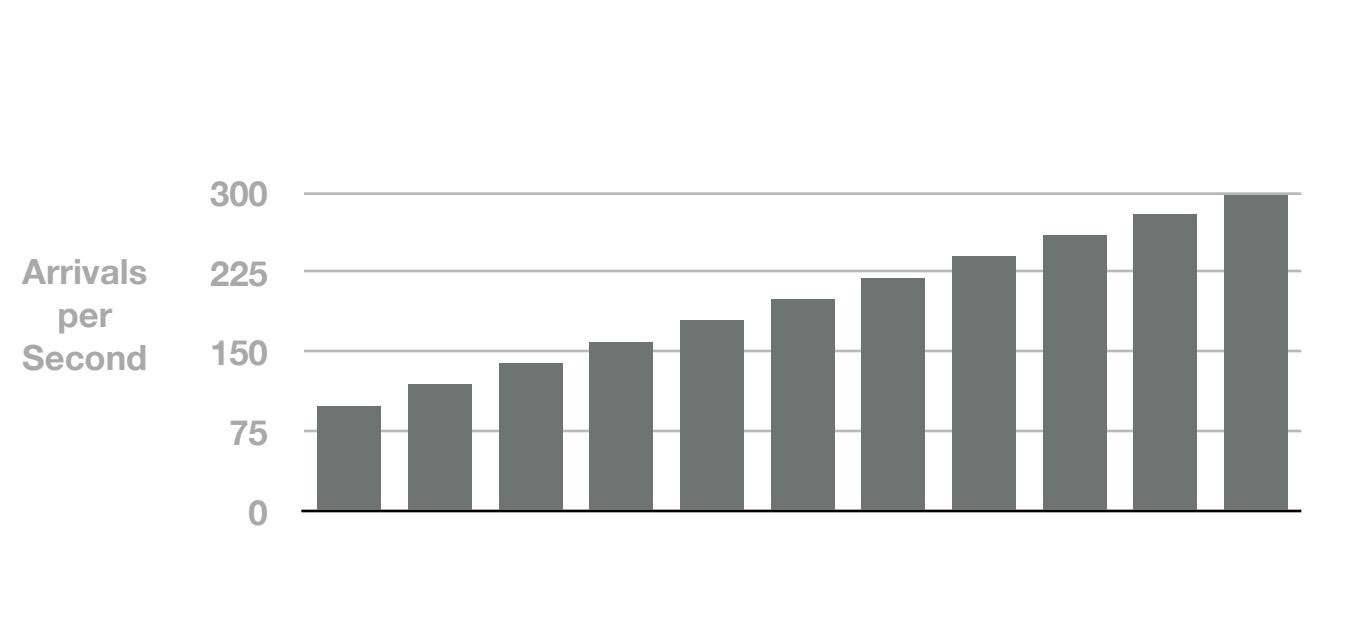


Load or Smoke?

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 120  
      arrivalRate: 20  
      name: 20 per second  
    - duration: 120  
      arrivalRate: 40  
      name: 40 per second  
    . . .  
    - duration: 120  
      arrivalRate: 200  
      name: 200 per second  
  
scenarios:  
  - flow:  
    - get:  
      url: '/records'  
      headers:  
        content-type: 'application/json'  
        authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

```
config:  
  target: 'https://d3tests.top/fmi/data/v1/databases/  
Contacts_100/layouts/contact_list_api'  
  phases:  
    - duration: 20  
      arrivalRate: 100  
      name: 100 per second  
    - duration: 20  
      arrivalRate: 120  
      name: 120 per second  
    . . .  
    - duration: 300  
      arrivalRate: 300  
      name: 300 per second  
  
scenarios:  
  - flow:  
    - get:  
      url: '/records'  
      headers:  
        content-type: 'application/json'  
        authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

→ ~

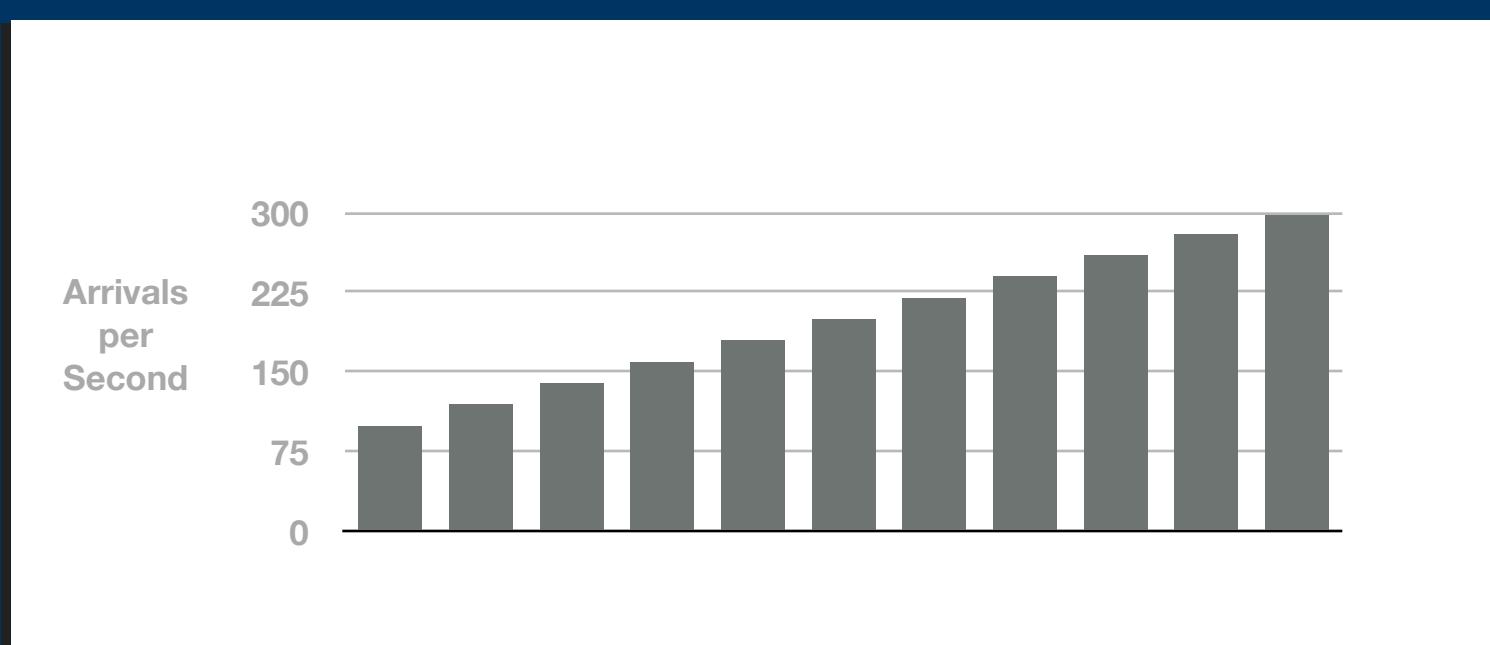


'data/v1/databases/
api'

→ ~

```
- duration: 20
  arrivalRate: 120
  name: 120 per second
  .
  .
  -
  duration: 300
  arrivalRate: 300
  name: 300 per second
```

```
scenarios:
- flow:
  - get:
    url: '/records'
    headers:
      content-type: 'application/json'
      authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```



'data/v1/databases/
api'

```
- duration: 20
  arrivalRate: 120
  name: 120 per second
  .
  .
  - duration: 300
    arrivalRate: 300
    name: 300 per second

scenarios:
- flow:
  - get:
    url: '/records'
    headers:
      content-type: 'application/json'
      authorization: 'Bearer
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```

→ ~ artillery run artillery-FMDiSC_test_2b_api.yml



```
→ ~ artillery run artillery-FMDiSC_test_2b_api.yml
.
.
.
Phase started: 160 per second (index: 3, duration: 20s) 08:41:36(-0700)
.
.
.
errors.ENOTFOUND: ..... 3
http.codes.200: ..... 1456
http.request_rate: ..... 147/sec
```



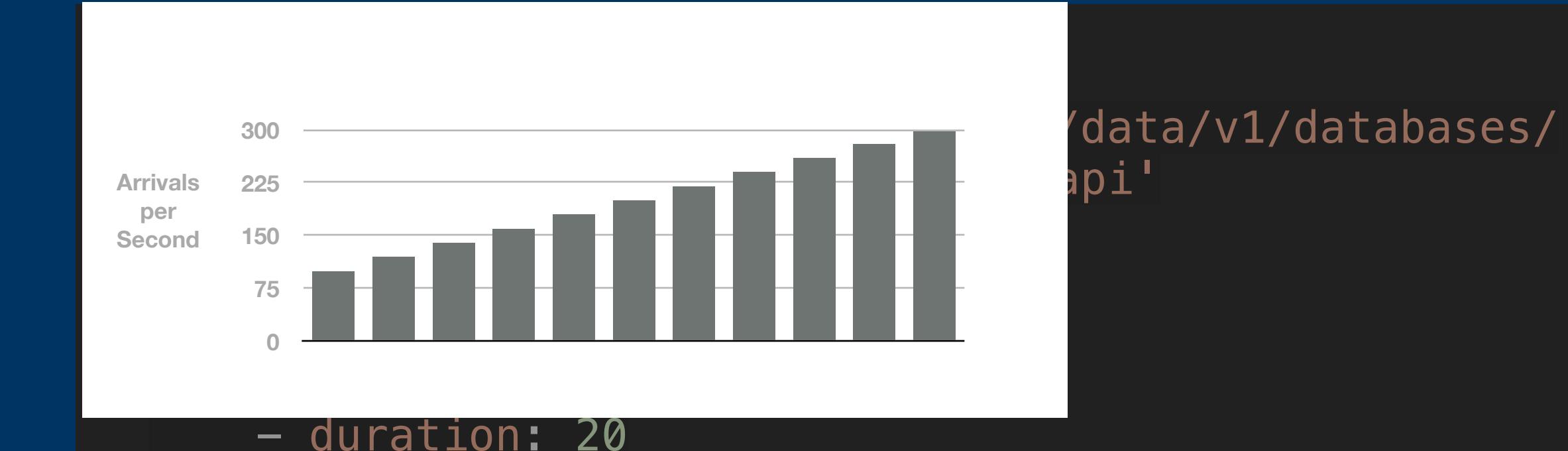
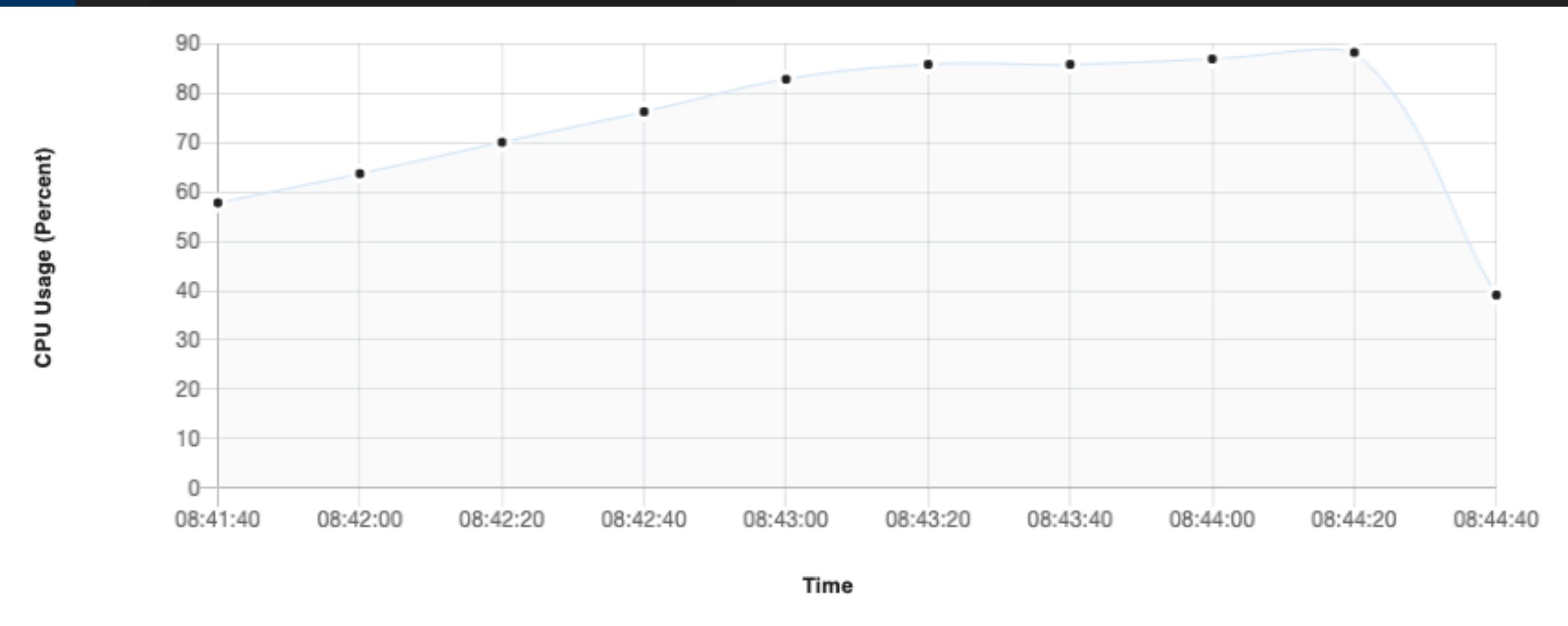
```
→ ~ artillery run artillery-FMDiSC_test_2b_api.yml
...
...
Phase started: 280 per second (index: 9, duration: 20s) 08:43:36(-0700)
errors.ENOTFOUND: ..... 9
http.codes.200: ..... 2140
http.codes.500: ..... 12
http.request_rate: ..... 254/sec
```



```
→ ~ artillery run artillery-FMDiSC_test_2b_api.yml
...
.
.
.

Phase started: 280 per second (index: 9, duration: 20s) 08:43:36(-0700)

errors.ENOTFOUND: ..... 9
http.codes.200: ..... 2140
http.codes.500: ..... 12
http.request_rate: ..... 254/sec
```



```
→ ~ artillery run artillery-FMDiSC_test_2b_api.yml
```

```
...
```

```
...
```

```
Phase started: 280 per second (index: 9, duration: 20s) 08:43:36(-0700)
```

```
errors.ENOTFOUND: .....
```

```
http.codes.200: .....
```

```
http.codes.500: .....
```

```
http.request_rate: .....
```

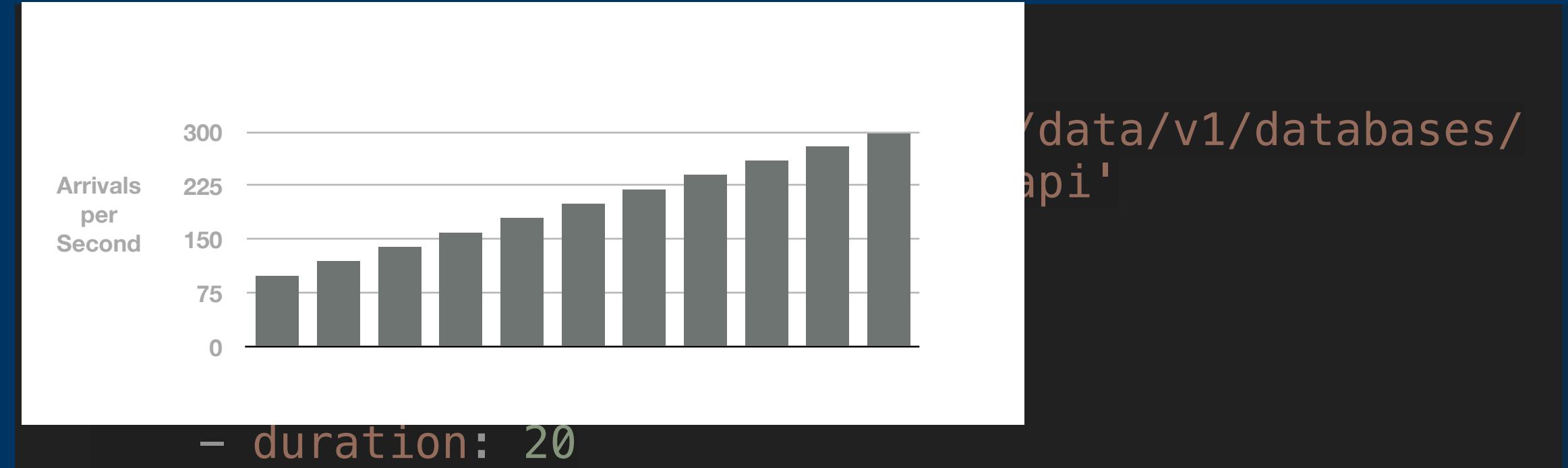
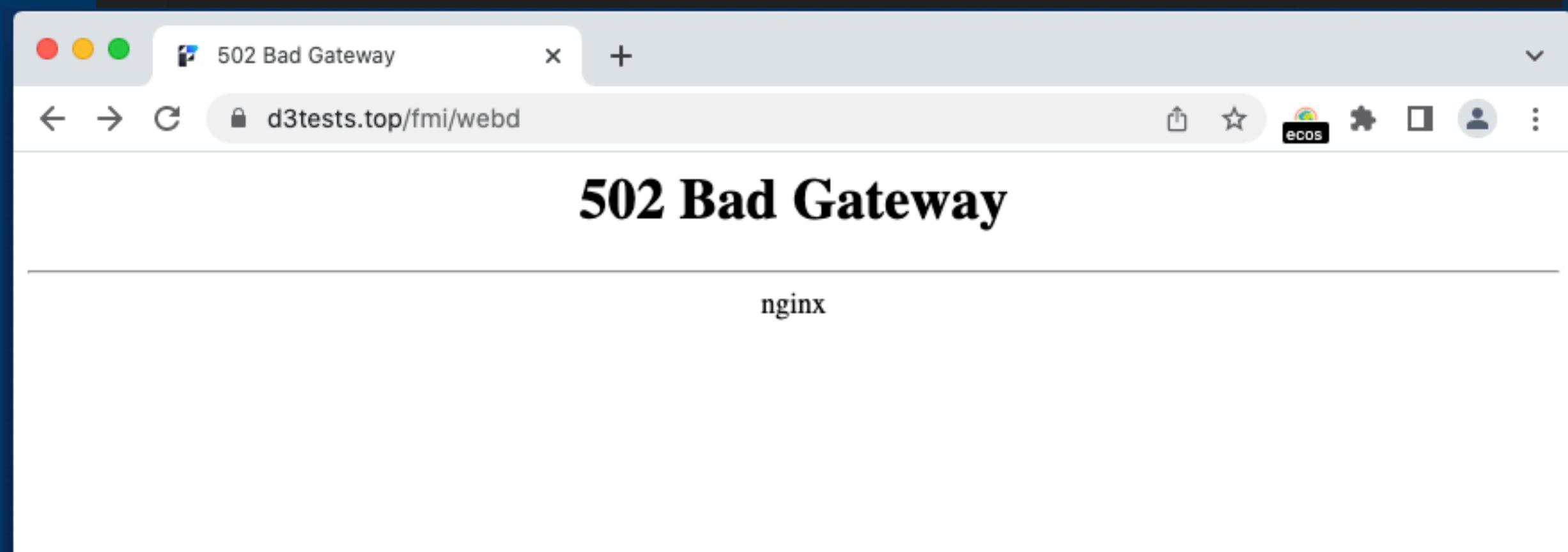
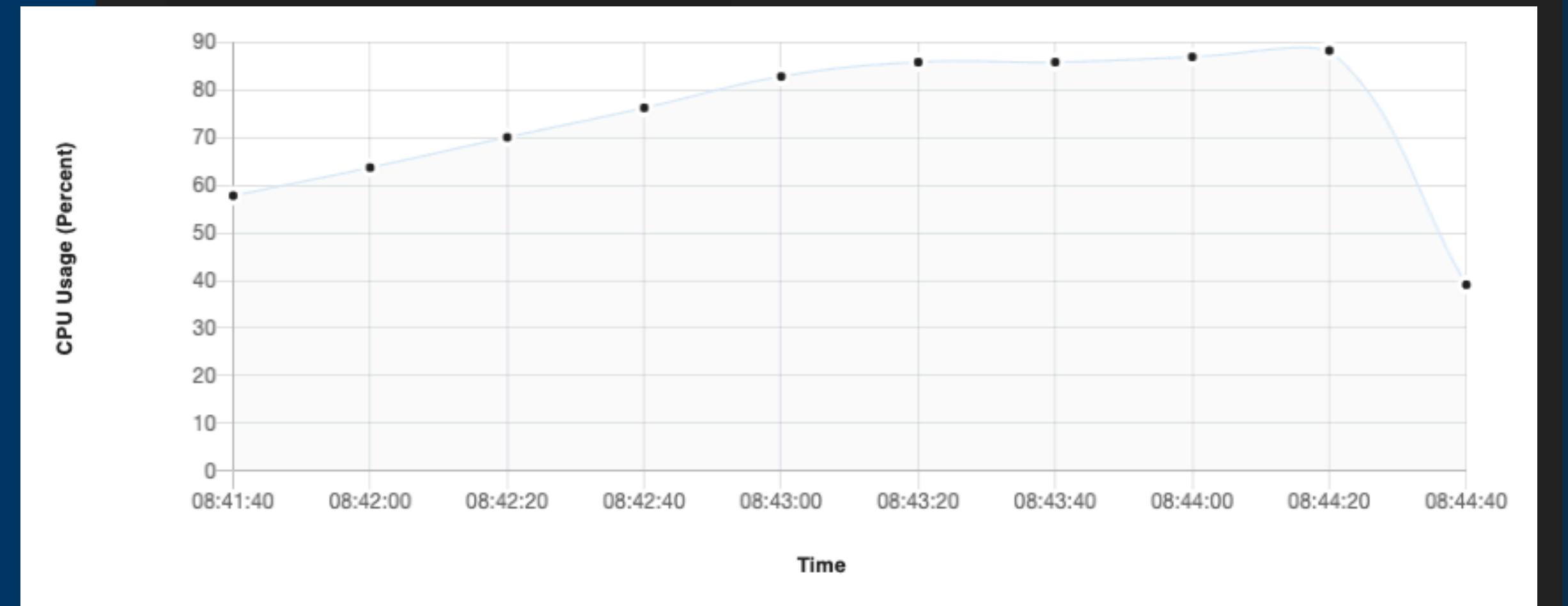
```
9
```

```
2140
```

```
12
```

```
254/sec
```

```
content-type: 'application/json'  
authorization: 'Bearer  
b3641b1aa6b5264b8b3140da574122a1ffd1571638672ec48b0d'
```



'data/v1/databases/
api'

```
→ ~ artillery run artillery-FMDiSC_test_2b_api.yml
```

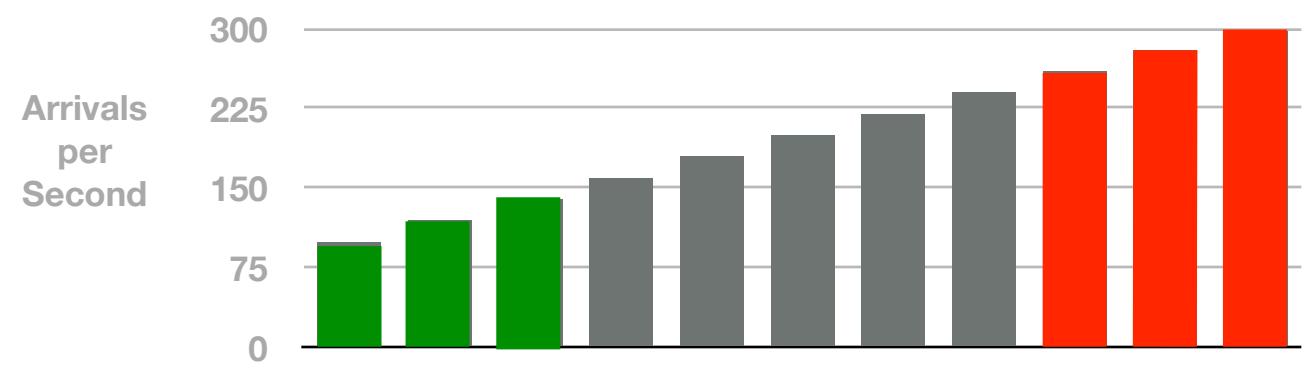
```
...
```

```
...
```

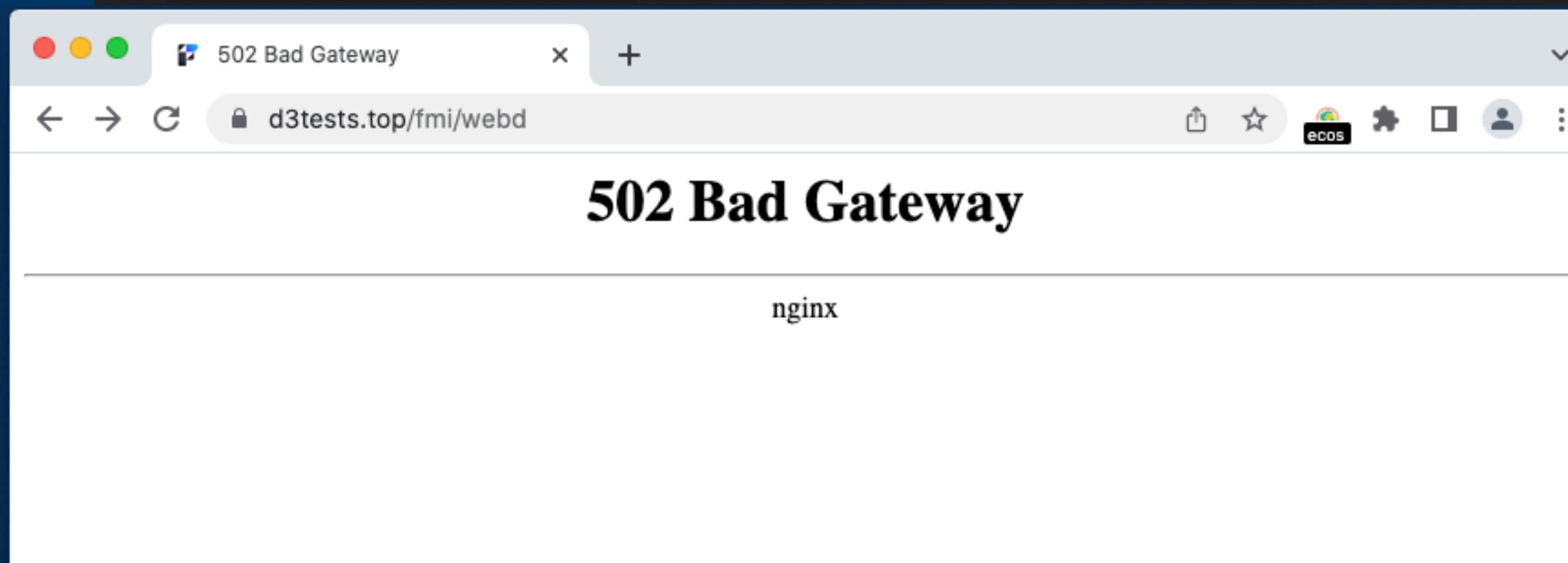
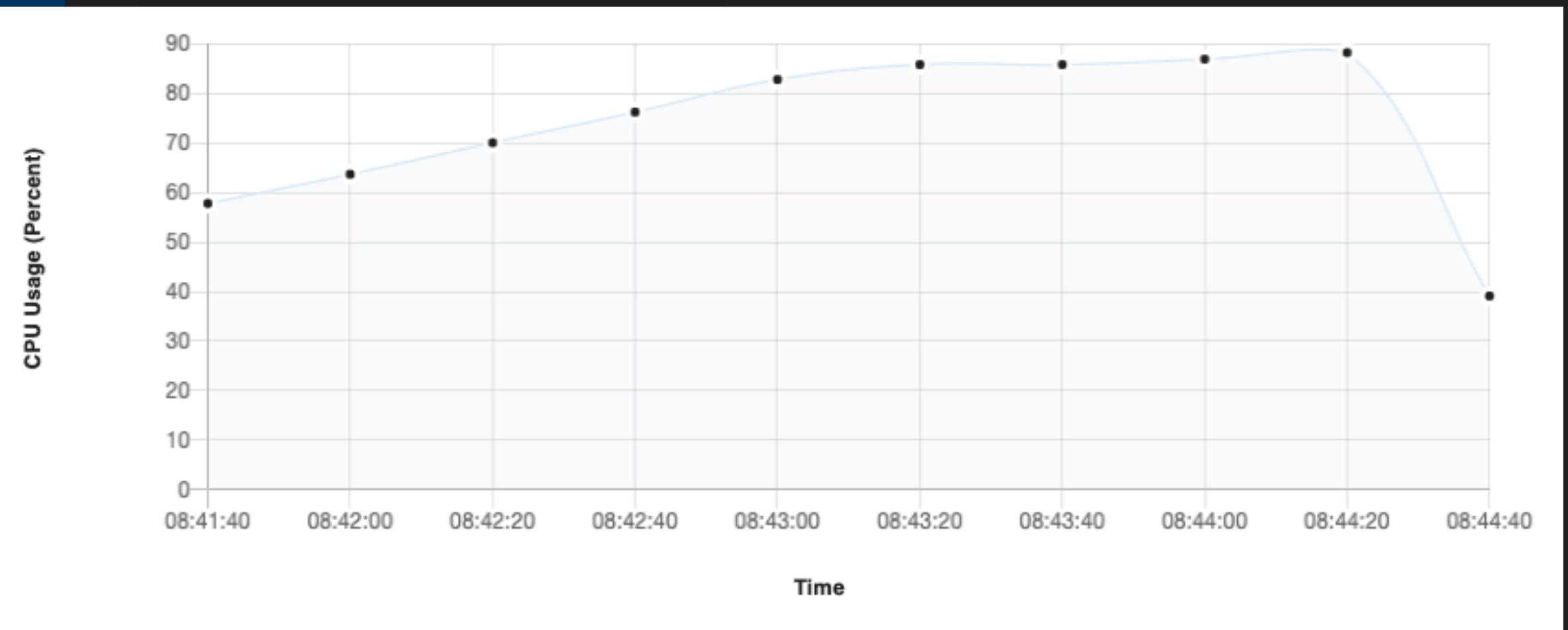
```
Phase started: 280 per second (index: 9, duration:  
20s) 08:43:36(-0700)
```

```
errors.ENOTFOUND: ..... 9  
http.codes.200: ..... 2140  
http.codes.500: ..... 12  
http.request_rate: ..... 254/sec
```

100 Records | **120/sec** | **254/sec**



'data/v1/databases/
api'



→ ~ artillery run artillery-FMDiSC_test_2b_api.yml

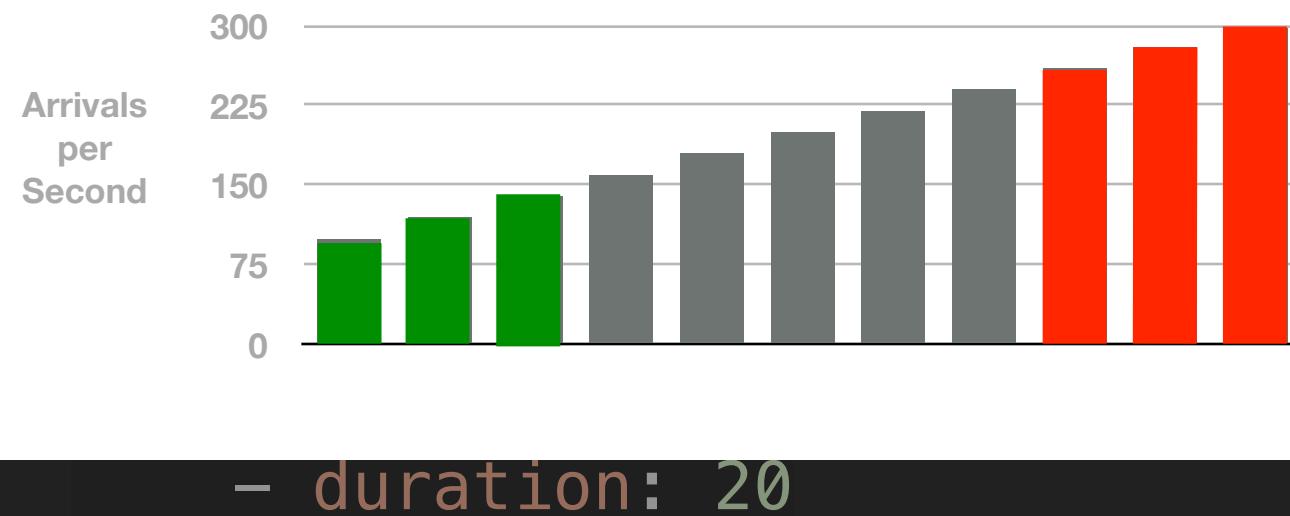
...

...

Phase started: 280 per second (index: 9, duration: 20s) 08:43:36(-0700)

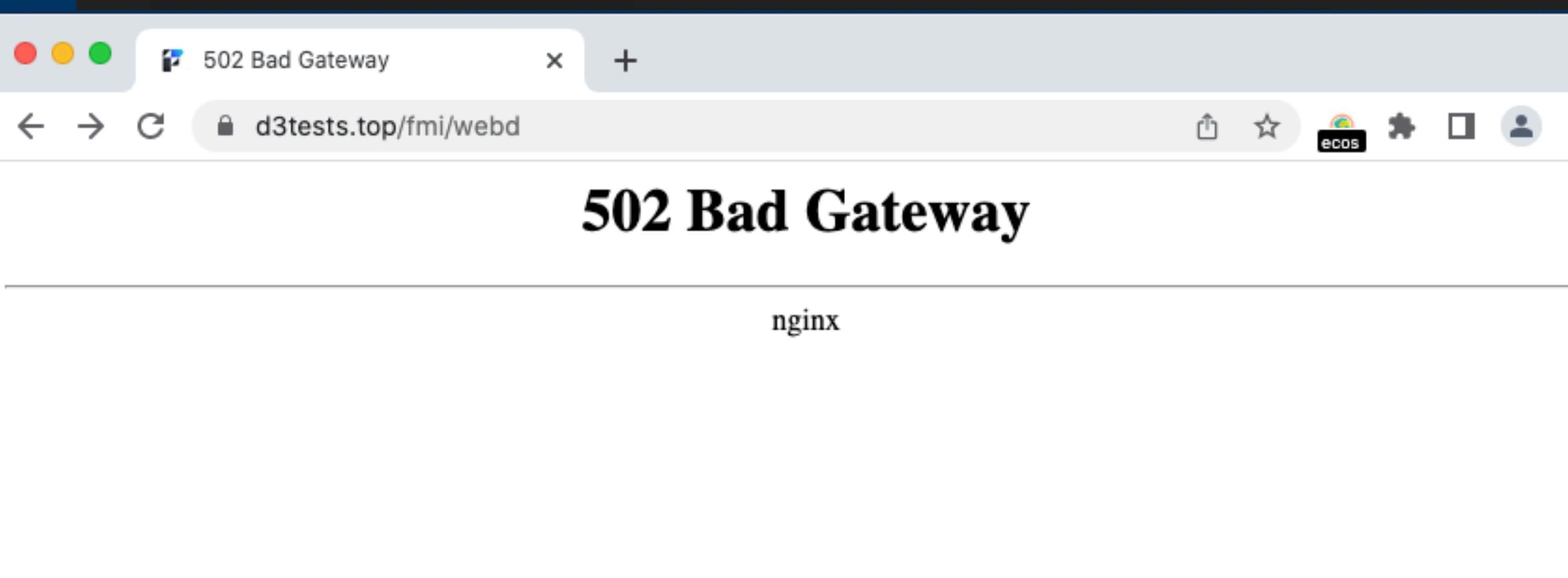
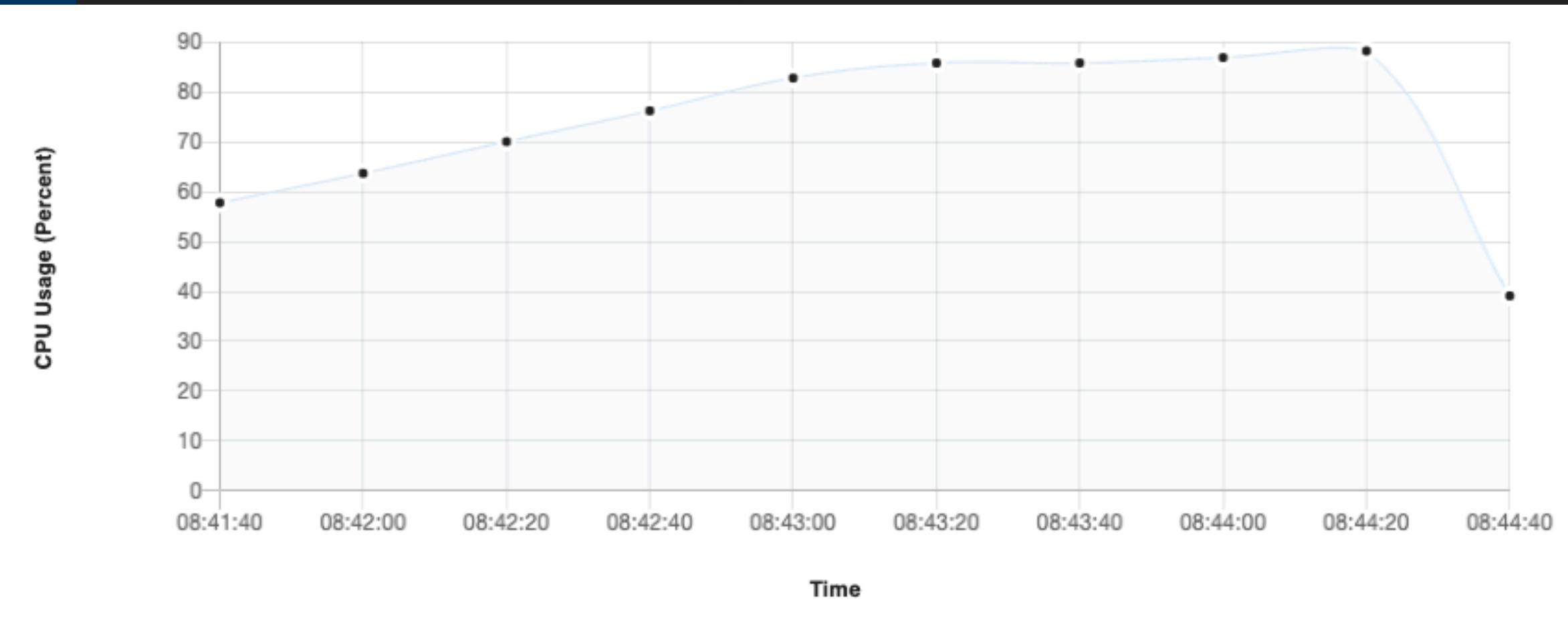
errors.ENOTFOUND: 9
http.codes.200: 2140
http.codes.500: 12
http.request_rate: 254/sec

100 Records | **120/sec** | **254/sec**



'data/v1/databases/
api'

- duration: 20



```
→ ~ artillery run artillery-FMDiSC_test_2b_api.yml
```

```
...
```

```
...
```

```
Phase started: 280 per second (index: 9, duration:  
20s) 08:43:36(-0700)
```

```
errors.ENOTFOUND: ..... 9  
http.codes.200: ..... 2140  
http.codes.500: ..... 12  
http.request_rate: ..... 254/sec
```

```
→ ~ sudo service fmshelper stop
```

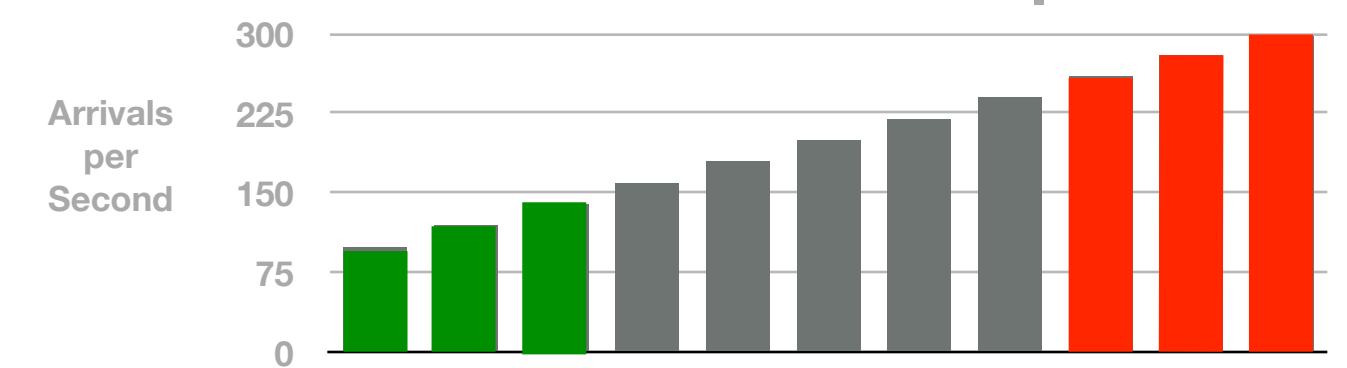
```
→ ~ sudo service fmshelper start
```

```
→ ~ ps -A | grep fm
```

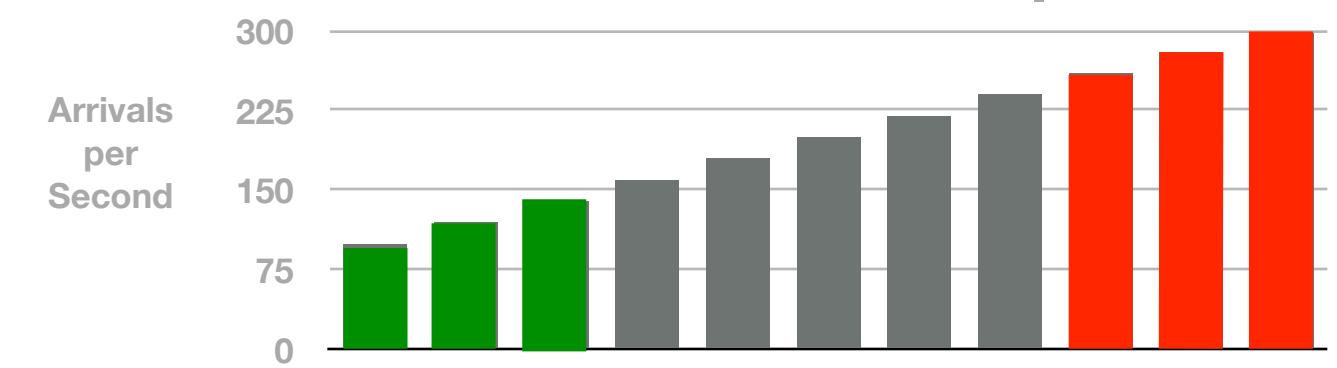
606462	?	00:00:00	fmshelper
606493	?	00:00:00	fmsib
606495	?	00:00:00	fmserverd
606538	?	00:00:00	fmsased
606628	?	00:00:00	fmscwpc
606850	?	00:00:00	fmwipd
606854	?	00:00:00	fmsadmin

All 7 processes needed

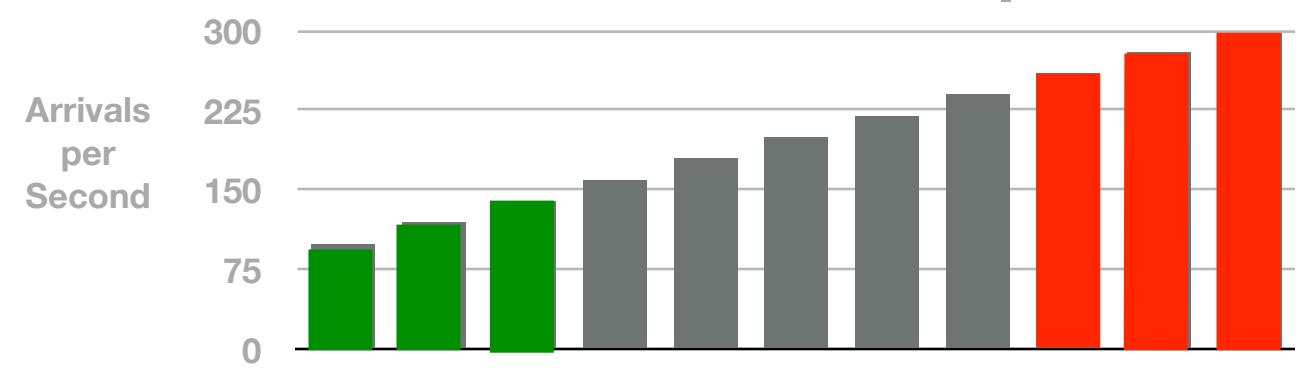
100 Records | **120/sec** | **254/sec**



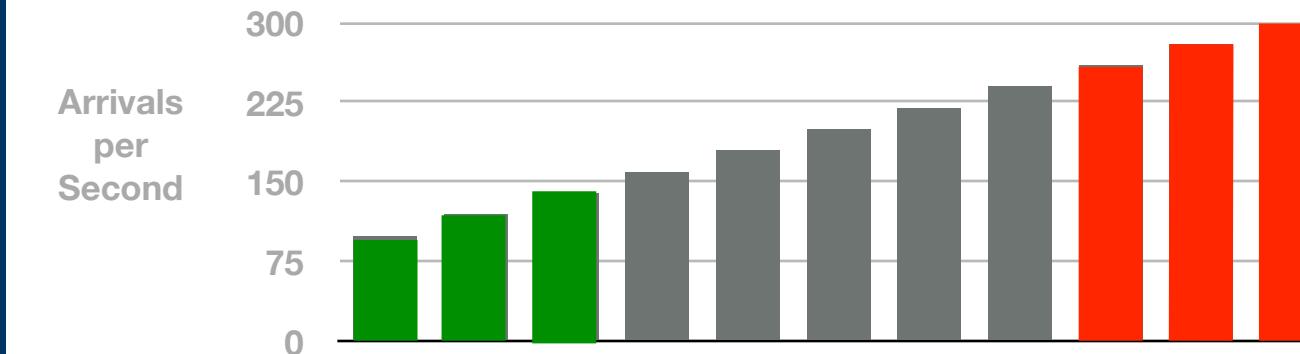
100 Records | **120/sec** | **254/sec**



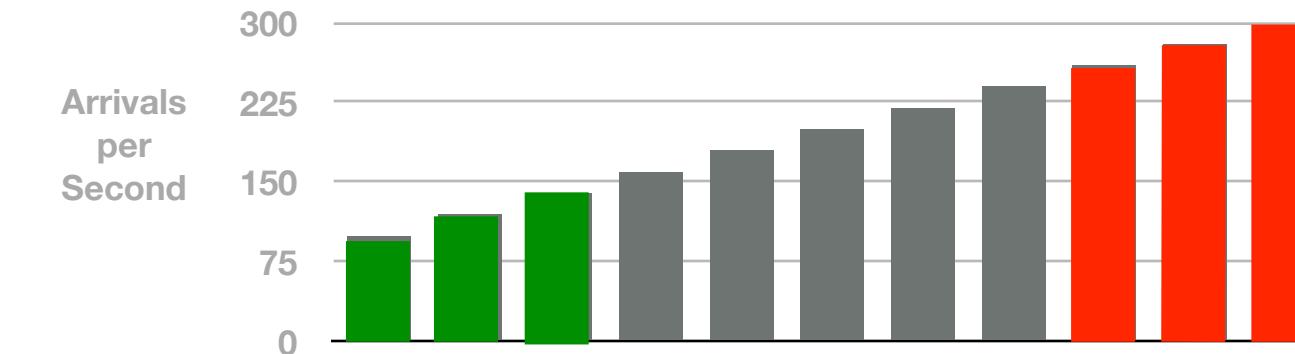
10K Records | **120/sec** | **260/sec**



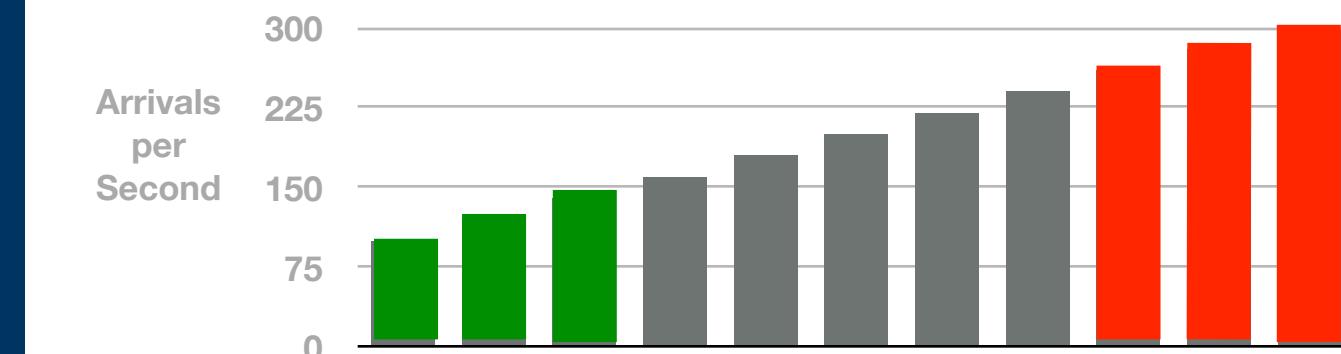
100 Records | **120/sec** | **254/sec**



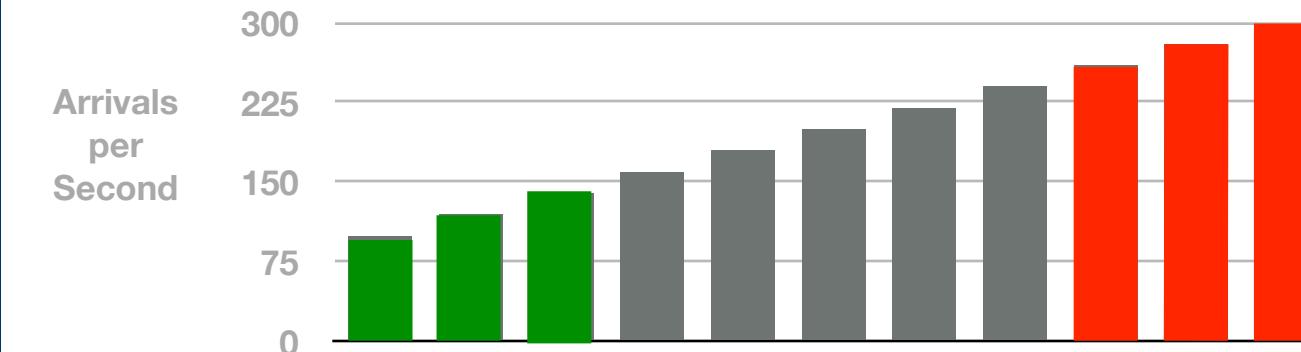
10K Records | **120/sec** | **260/sec**



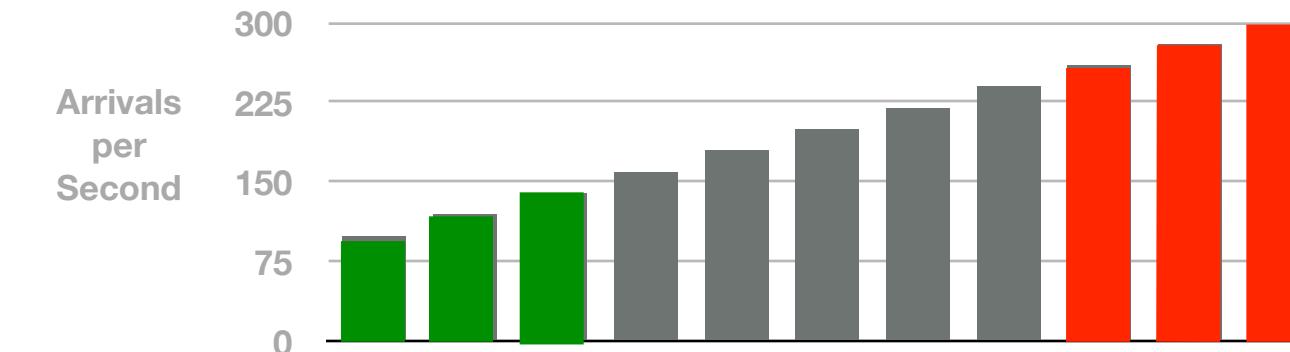
1M Records | **120/sec** | **280/sec**



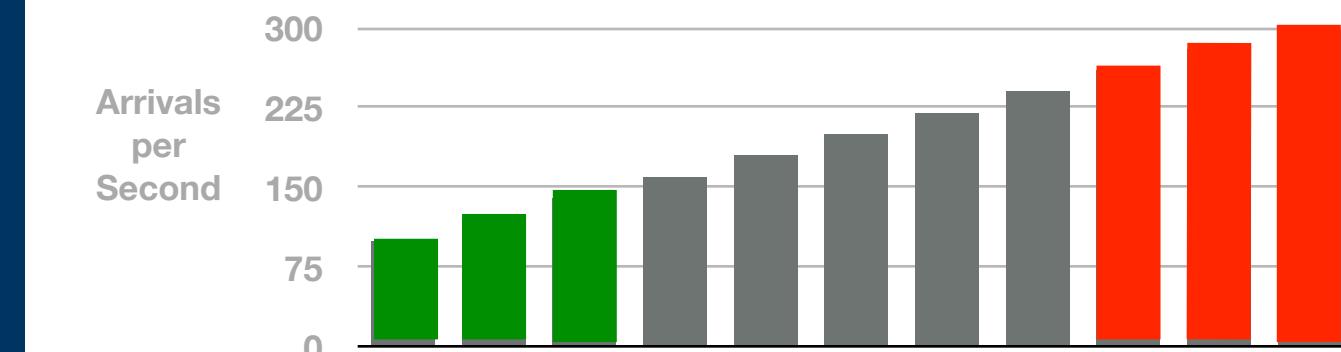
100 Records | **120/sec** | **254/sec**



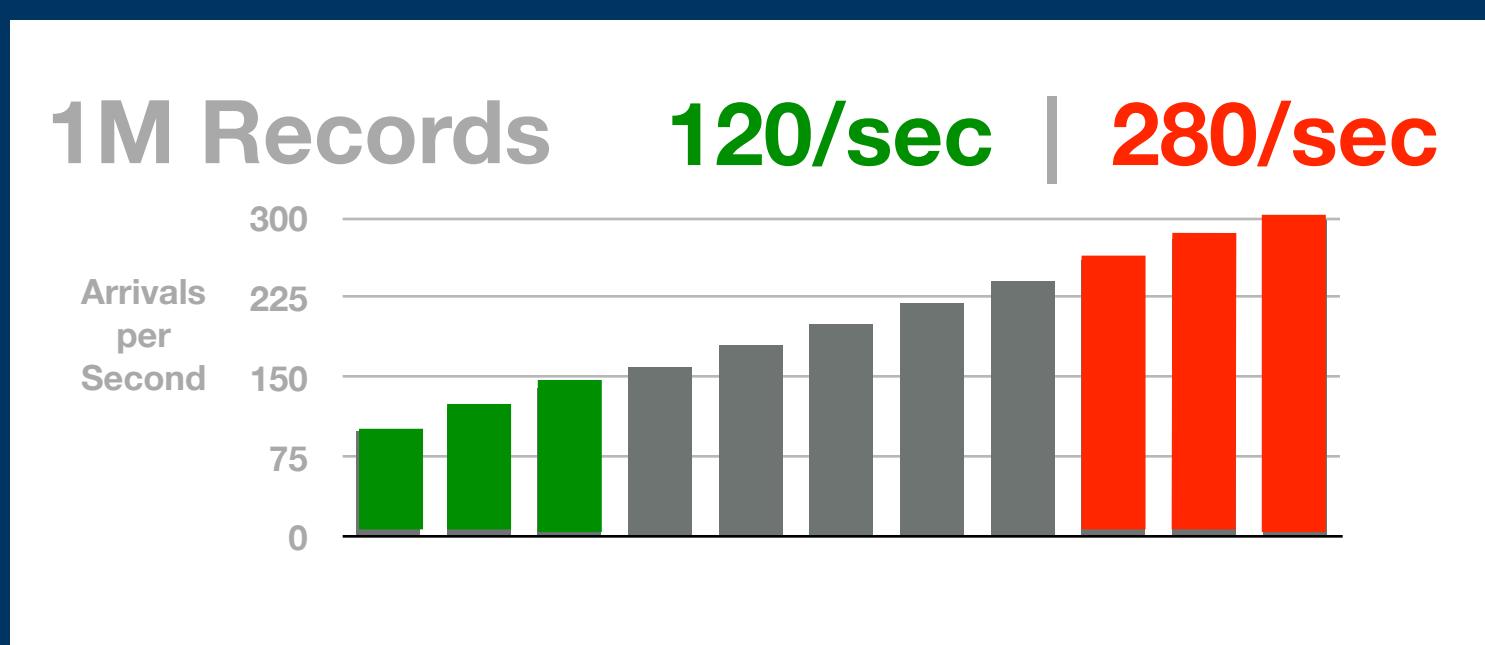
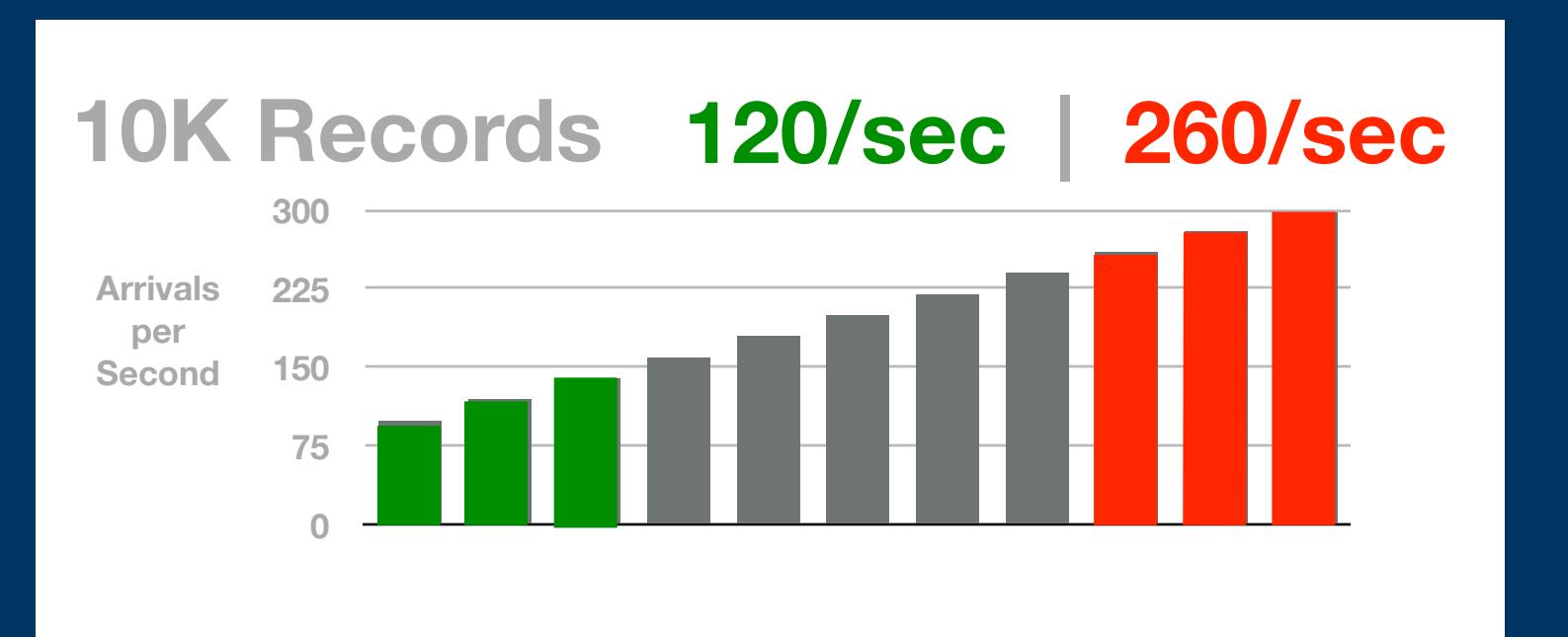
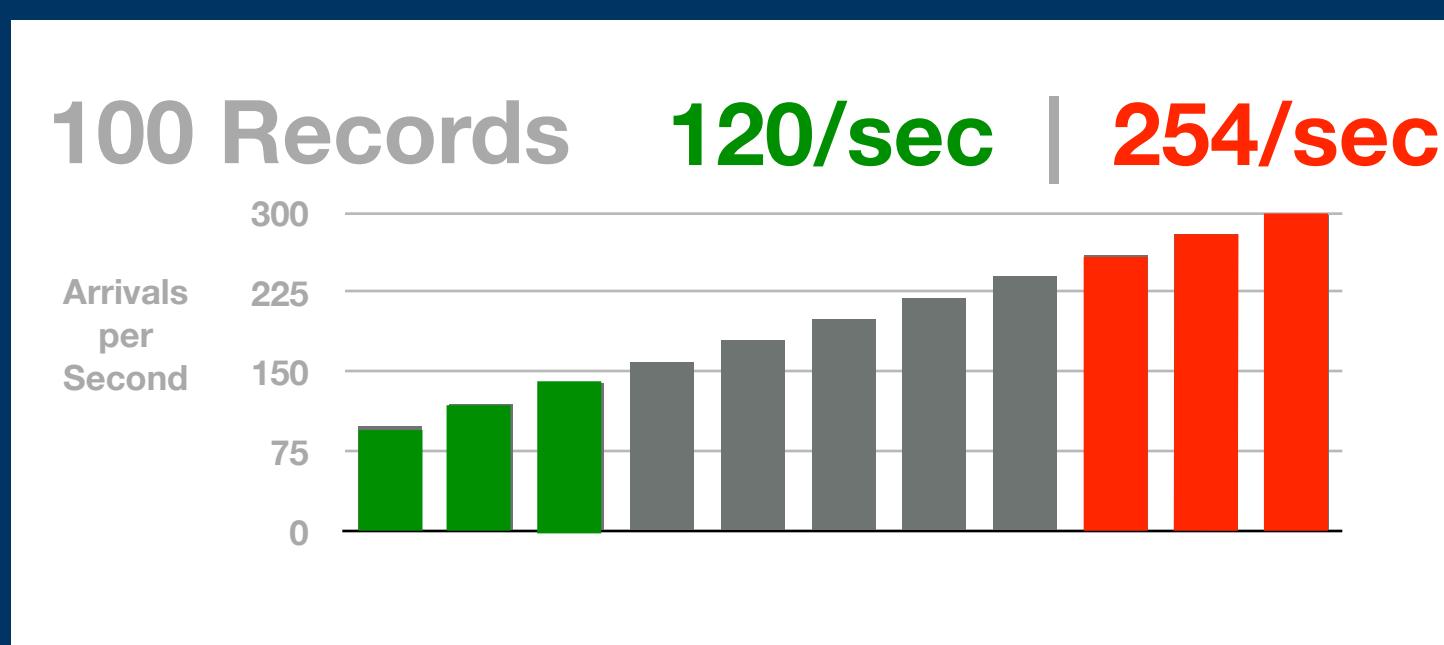
10K Records | **120/sec** | **260/sec**



1M Records | **120/sec** | **280/sec**



Conclusions?



Conclusions?

Break point unaffected by file

WebDirect tests

Artillery Features...

- an npm application **installed on your computer**
- **emulate user behavior with scenarios**
- **can have multiple steps**
- **run scenarios as load tests or end-to-end tests**
- **detailed performance metrics**
- **extension APIs for custom scripting in JS, and plugins**
- **runs on Windows, MacOS, and Linux systems**

Artillery Features...

- an npm application installed on your computer**
- emulate user behavior with scenarios**
- can have multiple steps**
- run scenarios as load tests or end-to-end tests**
- detailed performance metrics**
- extension APIs for custom scripting in JS, and plugins**



Artillery Features...

- an npm application installed on your computer
- emulate user behavior with scenarios
- can have multiple steps
- run scenarios as load tests or end-to-end tests
- detailed performance metrics
- extension APIs for custom scripting in JS, and plugins

Playwright plugin needed for accurate test results

Artillery alone only counts preflight responses?

Artillery Features...

- an npm application installed on your computer
- emulate user behavior with scenarios
- can have multiple steps
- run scenarios as load tests or end-to-end tests
- detailed performance metrics
- extension APIs for custom scripting in JS, and plugins

Playwright plugin needed for accurate test results

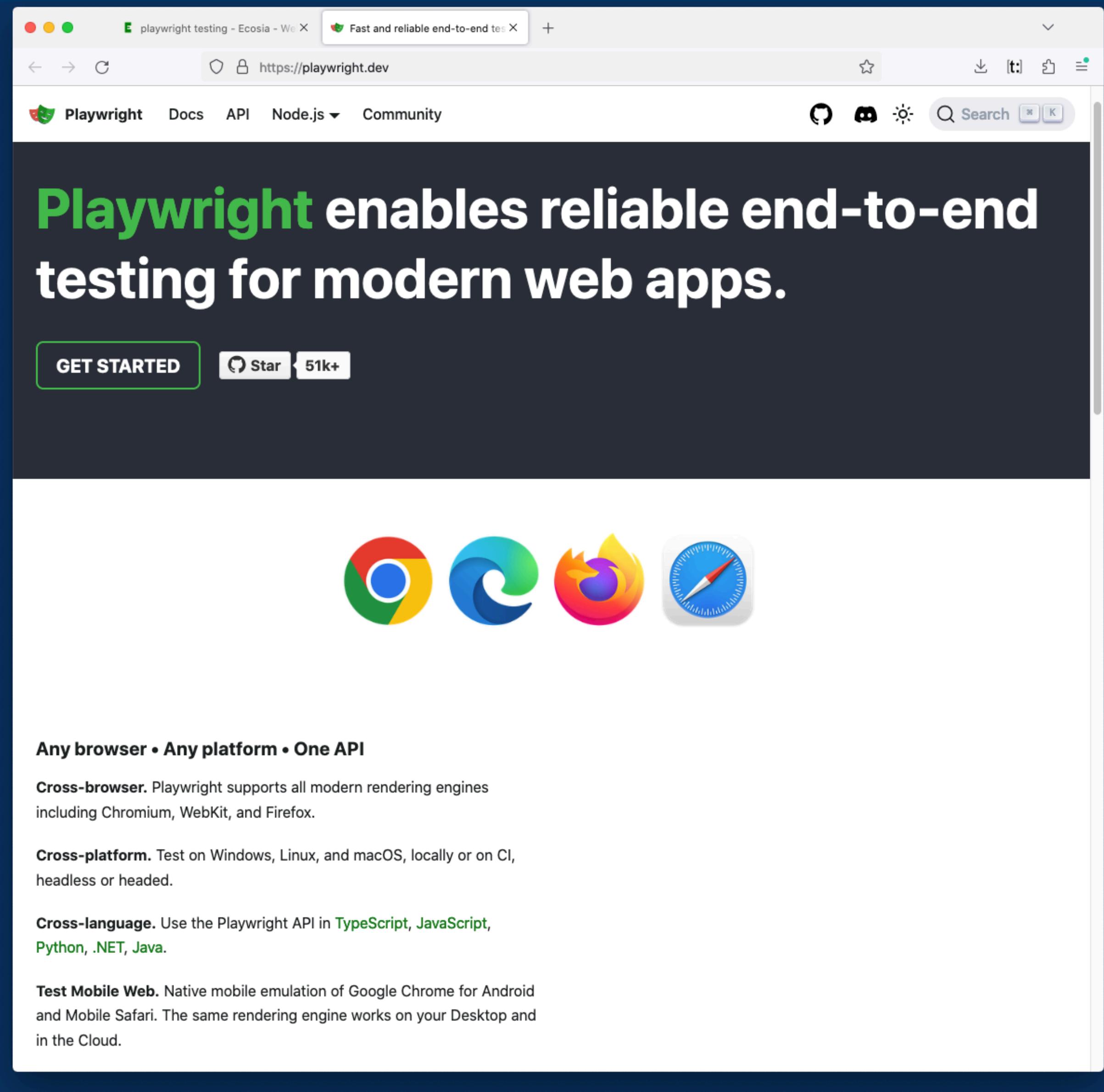
Artillery alone only counts preflight responses?

Playwright Virtual Users are Chromium browser instances

Very resource intensive

Recommended: dedicate a virtual server for testing

<https://playwright.dev/>



standalone cross-browser automation tool

mimics a single user's experience

for End-To-End testing

not meant to load test

but does load test as an Artillery plugin

Artillery Features...

- an npm application installed on your computer
- emulate user behavior with scenarios
- can have multiple steps
- run scenarios as load tests or end-to-end tests
- detailed performance metrics
- extension APIs for custom scripting in JS, and plugins
 - also install plugin

```
→ ~ npm install -g artillery-engine-playwright@latest
```


1) the .yml file for Artillery

```
config:  
  target: https://d3tests.top/fmi/webd/Contacts_100  
  engines:  
    playwright: {}  
  processor: './artillery-FMDiSC_test_3_WD.js'  
phases:  
  - duration: 5  
    arrivalRate: 2  
    name: 2 per second  
  
scenarios:  
  -  
    flow: []
```

engines: reference the playwright plugin
processor: the .js file for Playwright

2) a .js file for Playwright

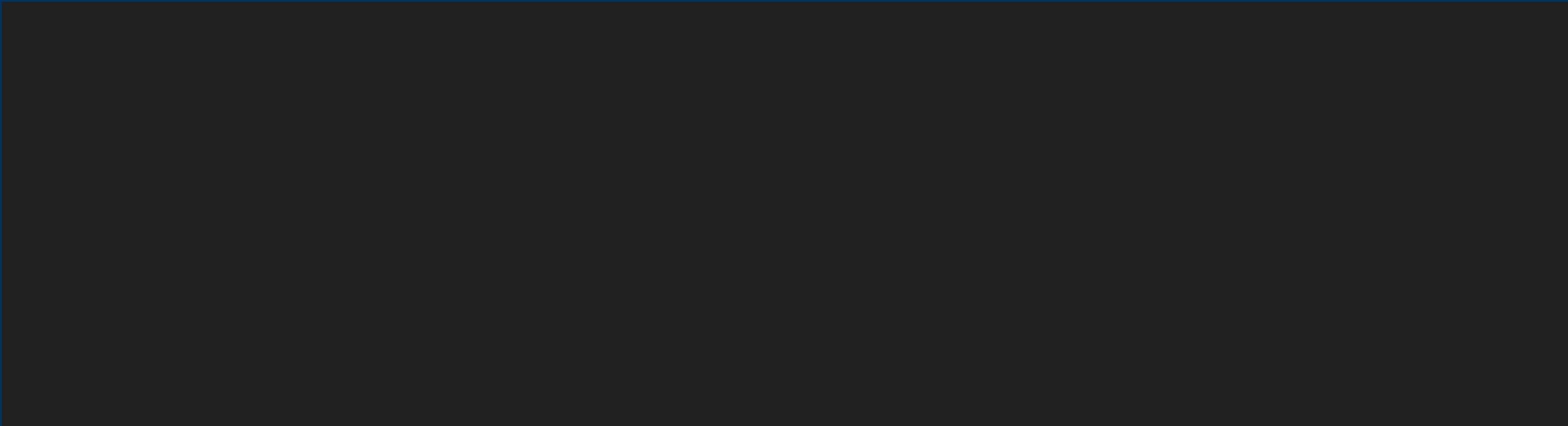
1) the .yml file for Artillery

```
config:  
  target: https://d3tests.top/fmi/webd/Contacts_100  
  engines:  
    playwright: {}  
  processor: './artillery-FMDiSC_test_3_WD.js'  
phases:  
  - duration: 5  
    arrivalRate: 2  
    name: 2 per second  
  
scenarios:  
  - engine: playwright  
    flowFunction: 'wdtest'  
    flow: []
```

engines: reference the playwright plugin

processor: the .js file for Playwright

2) a .js file for Playwright



1) the .yml file for Artillery

```
config:  
  target: https://d3tests.top/fmi/webd/Contacts_100  
  engines:  
    playwright: {}  
  processor: './artillery-FMDiSC_test_3_WD.js'  
phases:  
  - duration: 5  
    arrivalRate: 2  
    name: 2 per second  
  
scenarios:  
  - engine: playwright  
    flowFunction: 'wdtest'  
    flow: []
```

engines: reference the playwright plugin

processor: the .js file for Playwright

2) a .js file for Playwright

```
async function wdtest(page) {  
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');  
}  
  
module.exports = { wdtest };
```

function that calls the WD app

then gets exported

1) the .yml file for Artillery

```
config:  
  target: https://d3tests.top/fmi/webd/Contacts_100  
  engines:  
    playwright: {}  
  processor: './artillery-FMDiSC_test_3_WD.js'  
phases:  
  - duration: 5  
    arrivalRate: 2  
    name: 2 per second  
  
scenarios:  
  - engine: playwright  
    flowFunction: 'wdtest'  
    flow: []
```

engines: reference the playwright plugin

processor: the .js file for Playwright

2) a .js file for Playwright

```
async function wdtest(page) {  
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');  
}  
  
module.exports = { wdtest };
```

function that calls the WD app

then gets used in .yml file

then gets exported



CONTACTS		
First Name	Last Name	Company
A		
Ladarius	Abernathy	Grady, Reilly and Wilderman
B		
Madonna	Beatty	Olson - Hyatt
Dashawn	Boyle	Ziemann - Kutch
Kiel	Bernhard	Mills - Walter
Anthony	Bashirian	Lind - Langworth
June	Bogan	Hamill, Kemmer and Harber
Tristian	Bergnaum	Renner Group
Will	Becker	Bashirian, Marvin and Schaefer
Lina	Brown	Stracke - Douglas
Carmela	Bartell	Hand - Macejkovic

same file, different layout

100 records

opens to list view as Guest

no password

Claris FileMaker 2023 Technical Specifications

FileMaker Server on Linux - Primary Machine

Minimum	Better	Best
CPU: 4 Core	CPU: 8 Core	CPU: 16 Core
RAM: 8 GB or more	RAM: 16 GB or more	RAM: 32 GB or more
Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive space.	Hard Drive: 512 GB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive space.	Hard Drive: 1 TB Solid State Drive (SSD) Storage or more, with a minimum of 20% of free drive space.
Network: Gigabit Ethernet	Network: Gigabit Ethernet	Network: Gigabit Ethernet
Number of WebDirect connections: 50	Number of WebDirect connections: 100	Number of WebDirect connections: 120

same server FM server machine



DO-09-v6

in ppreagan / 4 GB Memory / 80 GB Disk / SFO3 - Ubuntu DO-09 v6 2CPU, 4GB, FMS...

ON

ipv4: 143.198.128.107

ipv6: [Enable now](#)

Private IP: 10.124.0.2

Reserved IP: [Enable now](#)

Console: ?

Graphs

Access

Power

Volumes

Resize

Backups

Snapshots

Resize Droplet

This Droplet is on a Basic plan. You must [turn off your Droplet](#) to resize it.

Currently using: Basic / 4 GB / 2 vCPUs

1) the .yml file for Artillery

```
config:  
  target: https://d3tests.top/fmi/webd/Contacts_100  
  engines:  
    playwright: {}  
  processor: './artillery-FMDiSC_test_3_WD.js'  
phases:  
  - duration: 5  
    arrivalRate: 2  
    name: 2 per second  
  
scenarios:  
  - engine: playwright  
    flowFunction: 'wdtest'  
    flow: []
```

```
→ ~ artillery run test_3_WD.yml
```

2) a .js file for Playwright

```
async function wdtest(page) {  
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');  
}  
  
module.exports = { wdtest };
```

1) the .yml file for Artillery

```
config:
  target: https://d3tests.top/fmi/webd/Contacts_100
  engines:
    playwright: {}
  processor: './artillery-FMDiSC_test_3_WD.js'
phases:
  - duration: 5
    arrivalRate: 2
    name: 2 per second

scenarios:
  - engine: playwright
    flowFunction: 'wdtest'
    flow: []
```

2) a .js file for Playwright

```
async function wdtest(page) {
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');
}

module.exports = { wdtest };
```

```
→ ~ artillery run test_3_WD.yml
Test run id: tjtgx_wp4rmtzk5wq333maz_n78q
Phase started: 2 per second
. . .

All VUs finished. Total time: 7 seconds
-----
Summary report @ 13:13:21(-0700)
-----
. . .

vusers.completed: ..... 15
vusers.created:..... 15
vusers.created_by_name.0:..... 15
vusers.failed: ..... 0
. . .
```


1) the .yml file for Artillery

```
config:  
  target: https://d3tests.top/fmi/webd/Contacts_100  
  engines:  
    playwright: {}  
  processor: './artillery-FMDiSC_test_3_WD.js'  
phases:  
  - duration: 120  
    arrivalRate: 1  
    name: 1 per second  
  
scenarios:  
  - engine: playwright  
    flowFunction: 'wdtest'  
    flow: []
```

2) a .js file for Playwright

```
async function wdtest(page) {  
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');  
}  
  
module.exports = { wdtest };
```

→ ~

1) the .yml file for Artillery

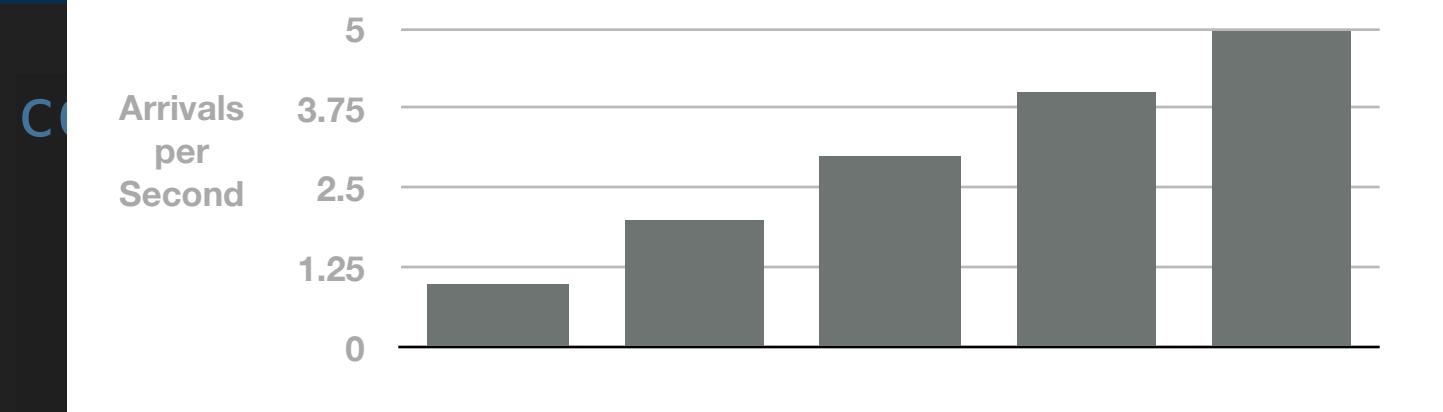
```
config:  
  target: https://d3tests.top/fmi/webd/Contacts_100  
  engines:  
    playwright: {}  
  processor: './artillery-FMDiSC_test_3_WD.js'  
  phases:  
    - duration: 120  
      arrivalRate: 1  
      name: 1 per second  
    ...  
  
scenarios:  
  - engine: playwright  
    flowFunction: 'wdtest'  
    flow: []
```

2) a .js file for Playwright

```
async function wdtest(page) {  
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');  
}  
  
module.exports = { wdtest };
```

→ ~

100 Records



```
phases:  
  - duration: 120  
    arrivalRate: 1  
    name: 1 per second
```

```
...  
...
```

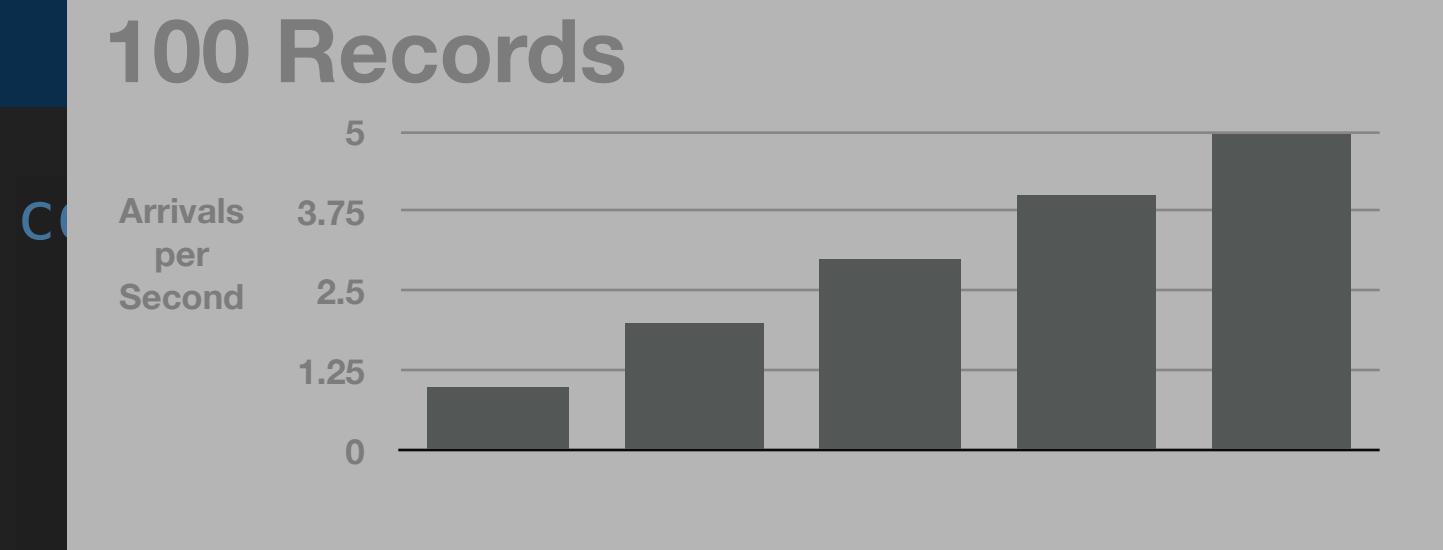
```
scenarios:  
  - engine: playwright  
    flowFunction: 'wdtest'  
    flow: []
```

/Contacts_100

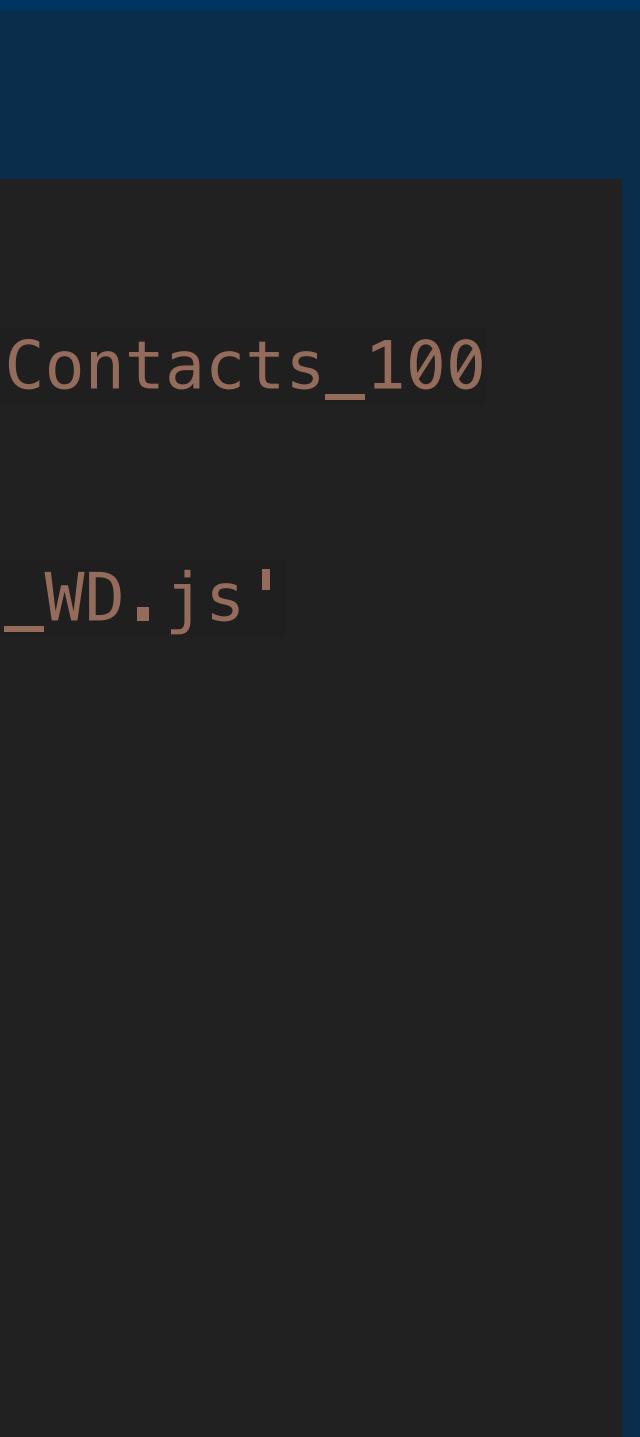
β_WD.js'

2) a .js file for Playwright

```
async function wdtest(page) {  
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');  
}  
  
module.exports = { wdtest };
```

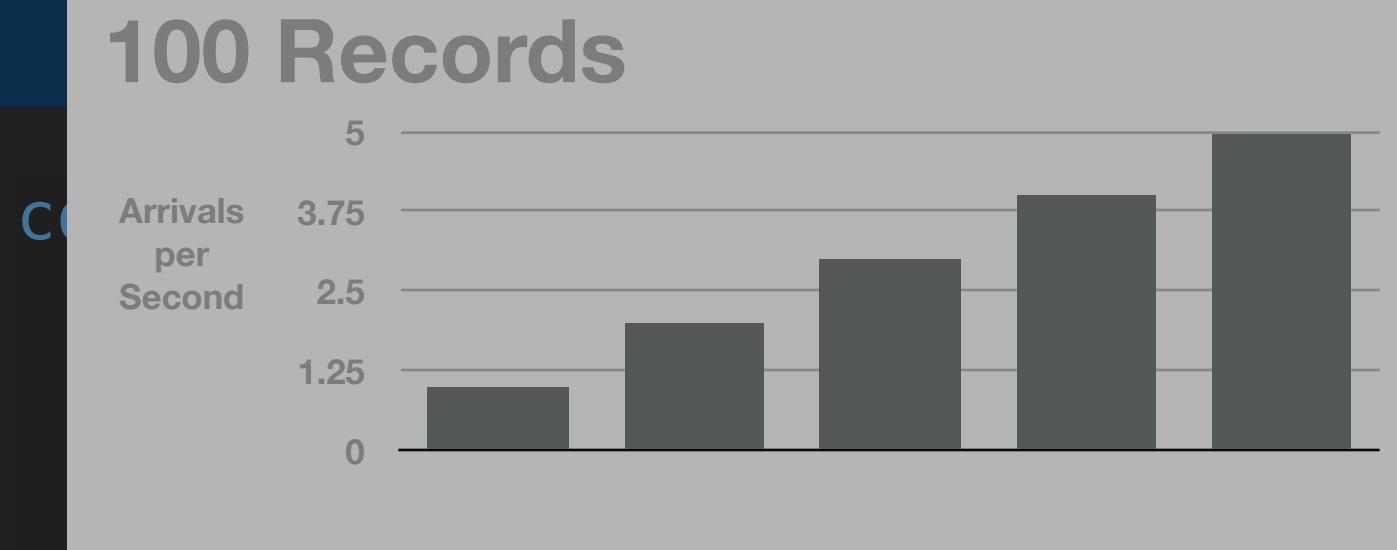


```
phases:  
  - duration: 120  
    arrivalRate: 1  
    name: 1 per second  
    ...  
  
scenarios:  
  - engine: playwright  
    flowFunction: 'wdtest'  
    flow: []
```



2) a .js file for Playwright

```
async function wdtest(page) {  
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');  
}  
  
module.exports = { wdtest };
```



```

phases:
  - duration: 120
    arrivalRate: 1
    name: 1 per second
    ...
  ...
scenarios:
  - engine: playwright
    flowFunction: 'wdtest'
    flow: []

```

/Contacts_100

β_WD.js'

→ ~

2) a .js file for Playwright

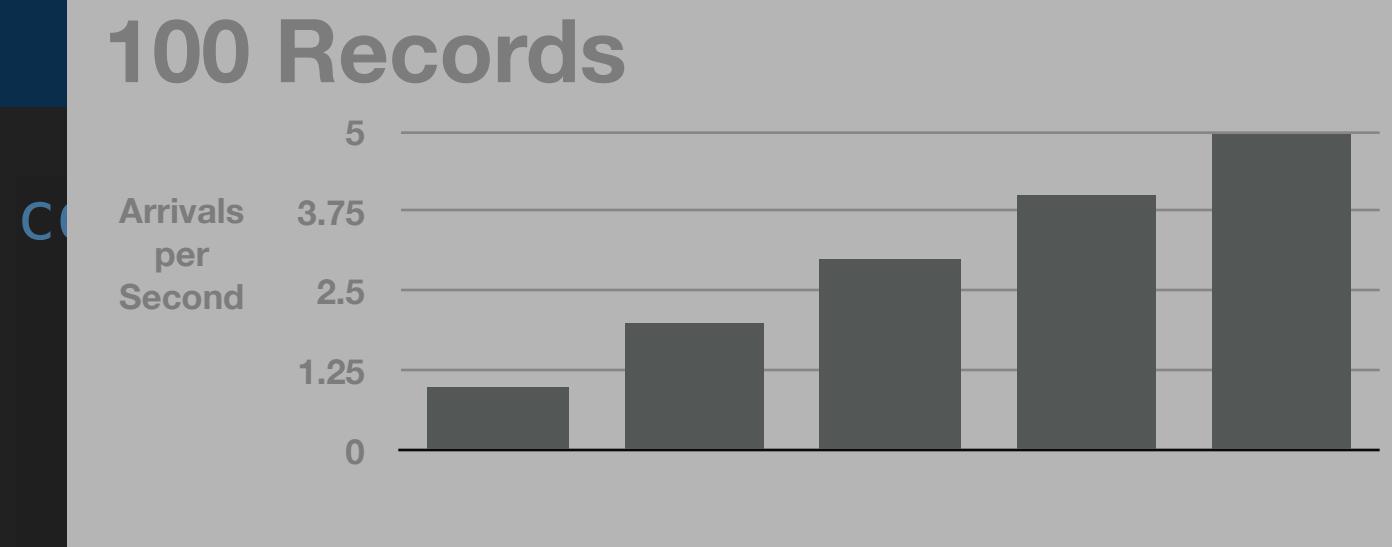
```

async function wdtest(page) {
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');
}

module.exports = { wdtest };

```

*WebDirect uses sessions
this allows test sessions to accumulate*



```

phases:
  - duration: 120
    arrivalRate: 1
    name: 1 per second
    ...
scenarios:
  - engine: playwright
    flowFunction: 'wdtest'
    flow: []

```

/Contacts_100

β_WD.js'

2) a .js file for Playwright

```

async function wdtest(page) {
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');

  await page.close();
}

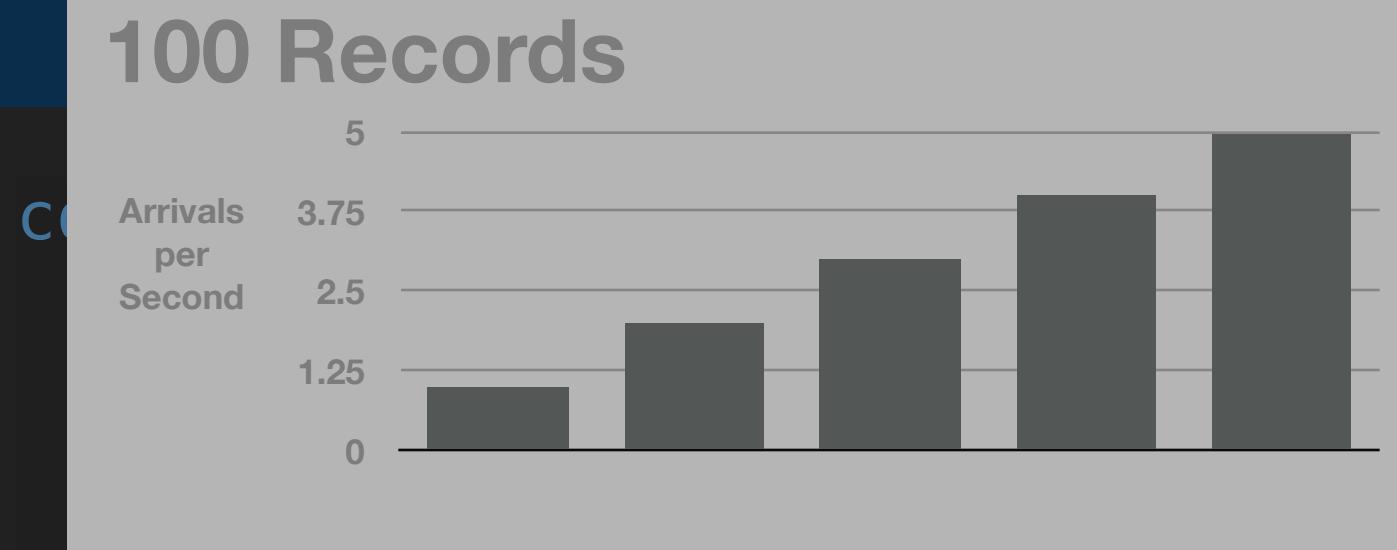
module.exports = { wdtest };

```

→ ~

WebDirect uses sessions

will only close the Chromium VU



```
contactcenter.js
phases:
  - duration: 120
    arrivalRate: 1
    name: 1 per second
    ...
scenarios:
  - engine: playwright
    flowFunction: 'wdtest'
    flow: []
```

/Contacts_100

β_WD.js'

→ ~

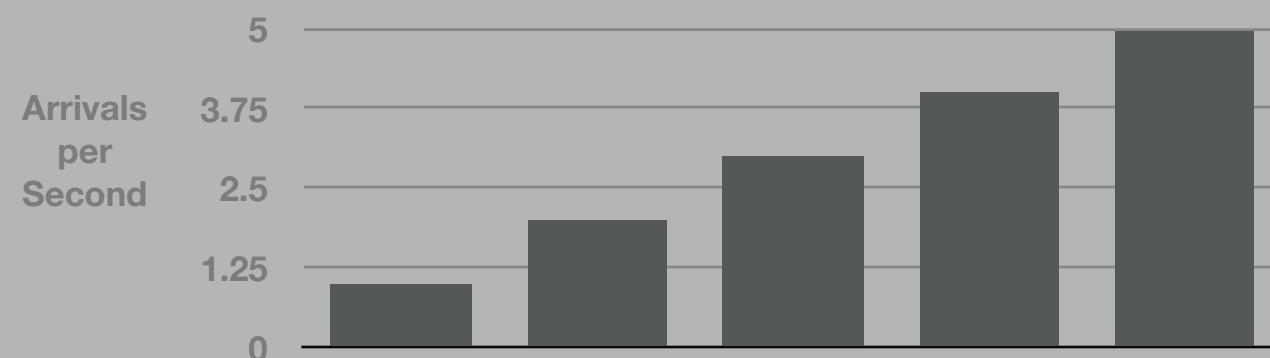
2) a .js file for Playwright

```
→
async function wdtest(page) {
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');
  await page.locator('#t0mainmenubar div').nth(2).click();
  await page.getByText('Close File').click();
  await page.close();

module.exports = { wdtest };
```

*WebDirect uses sessions
tell Playwright to mimic user closing file*

100 Records



```
phases:  
  - duration: 120  
    arrivalRate: 1  
    name: 1 per second
```

```
  . . .
```

```
scenarios:  
  - engine: playwright  
    flowFunction: 'wdtest'  
    flow: []
```

2) a .js file for Playwright

```
async function wdtest(page) {  
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');  
  await page.locator('#t0mainmenubar div').nth(2).click();  
  await page.getByText('Close File').click();  
  await page.close();  
  
module.exports = { wdtest };
```

/Contacts_100

β_WD.js'

→ ~

Playwright

Docs

API

Node.js ▾

Community



Getting Started

Installation

Writing tests

Running tests

Test generator

Trace viewer

UI Mode

CI GitHub Actions

Getting started - VS Code

Release notes

Canary releases

Playwright Test

Test configuration

Guides

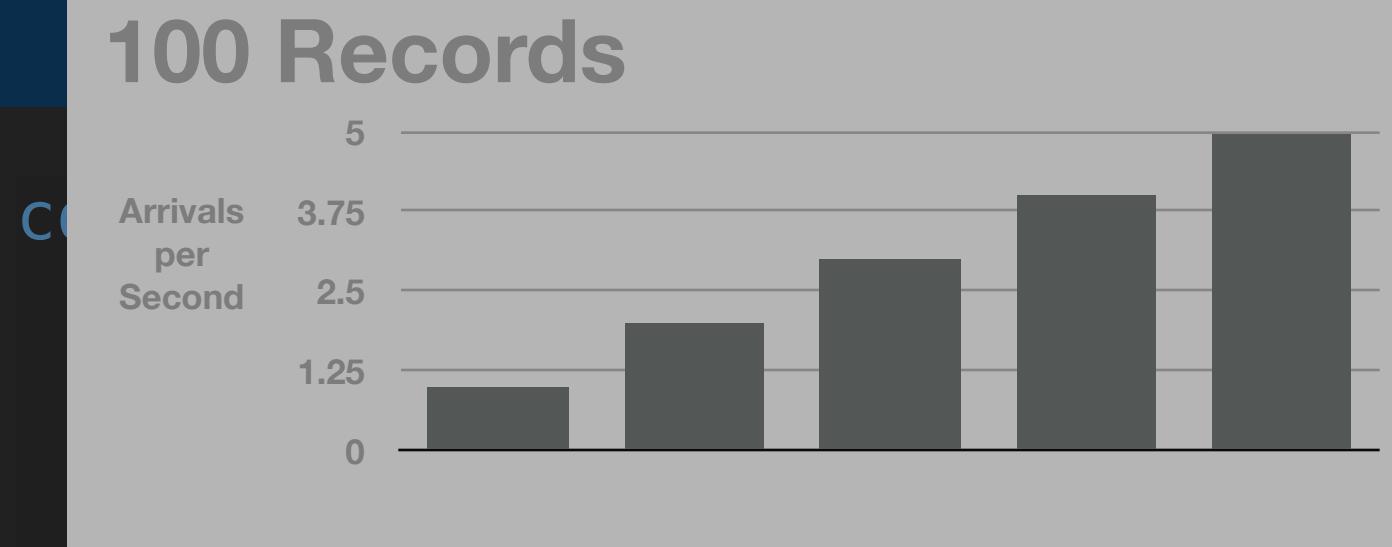
Test generator

Test generator

Playwright comes with the ability to generate tests for you as you perform actions in the browser and is a great way to quickly get started with testing. Playwright will look at your page and figure out the best locator, prioritizing **role, text and test id locators**. If the generator finds multiple elements matching the locator, it will improve the locator to make it resilient that uniquely identify the target element.

Generate tests in VS Code

Install the VS Code extension and generate tests directly from VS Code. The extension is available on the [VS Code Marketplace](#). Check out our guide on [getting started with VS Code](#).



```
phases:
  - duration: 120
    arrivalRate: 1
    name: 1 per second
```

```
  . . .
```

```
scenarios:
  - engine: playwright
    flowFunction: 'wdtest'
    flow: []
```

/Contacts_100

β_WD.js'

2) a .js file for Playwright

```
async function wdtest(page) {
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');
  await page.locator('#t0mainmenubar div').nth(2).click();
  await page.getByText('Close File').click();
  await page.close();}

module.exports = { wdtest };
```

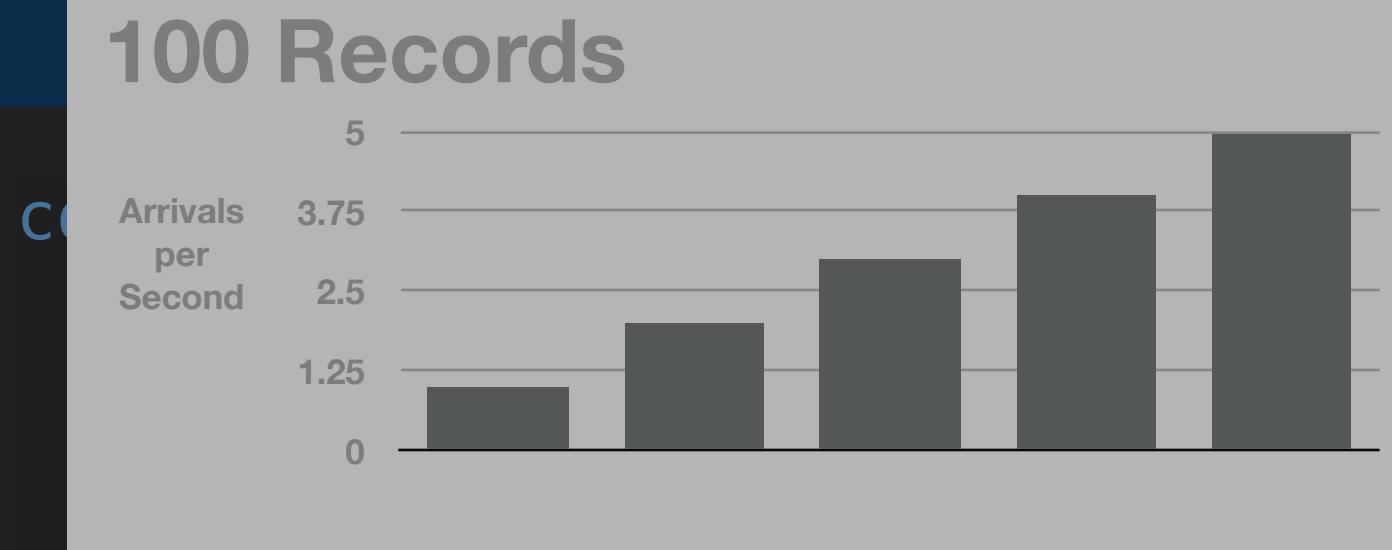
```
→ ~ artillery run test_4_WD.yml
Test run id: tjtgx_wp4rmtzk5wq333maz_n78q
Phase started: 1 per second
. . .
```

All VUs finished. Total time: 7 seconds

Summary report @ 14:33:21(-0700)

```
. . .

vusers.completed: ..... 15
vusers.created:..... 15
vusers.created_by_name.0:..... 15
vusers.failed: ..... 0
. . .
```



```

phases:
  - duration: 120
    arrivalRate: 1
    name: 1 per second
  ...
scenarios:
  - engine: playwright
    flowFunction: 'wdtest'
    flow: []

```

/Contacts_100

β_WD.js'

2) a .js file for Playwright

```

async function wdtest(page) {
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');
  await page.locator('#t0mainmenubar div').nth(2).click();
  await page.getByText('Close File').click();
  await page.close();
}

module.exports = { wdtest };

```

```

→ ~ artillery run test_4_WD.yml
Test run id: tjtgx_wp4rmtzk5wq333maz_n78q
Phase started: 1 per second
...

```

All VUs finished. Total time: 7 seconds

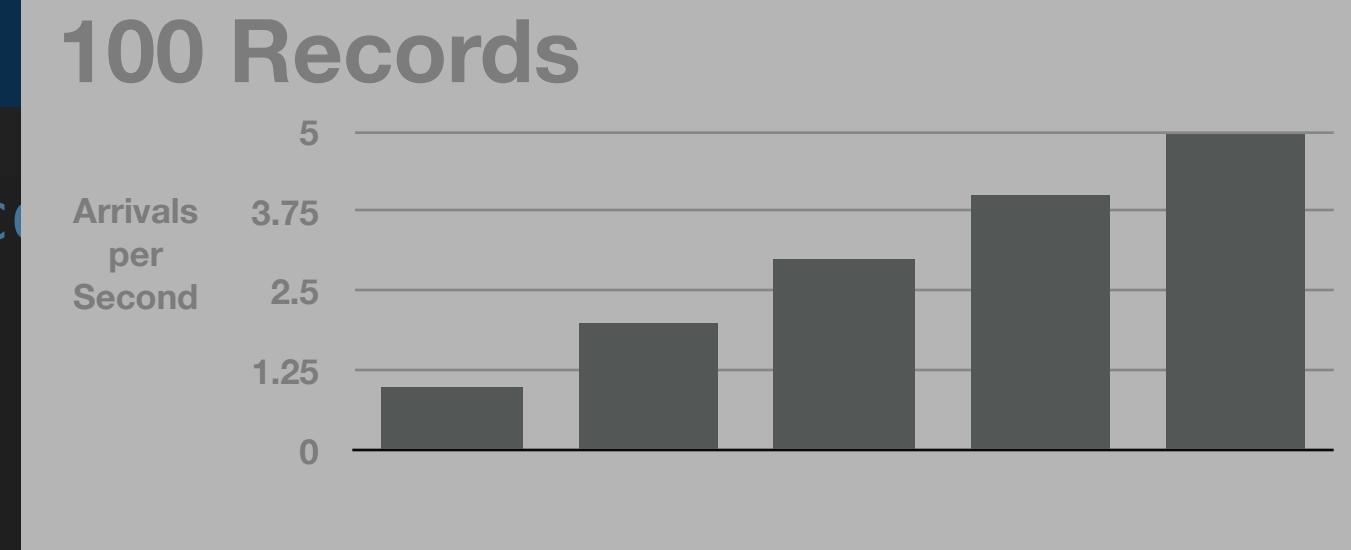
Summary report @ 14:33:21(-0700)

```

vusers.completed: ..... 15
vusers.created:..... 15
vusers.created_by_name.0:..... 15
vusers.failed: ..... 0
...

```

Only if Virtual User cannot be established



```

phases:
  - duration: 120
    arrivalRate: 1
    name: 1 per second
    ...
scenarios:
  - engine: playwright
    flowFunction: 'wdtest'
    flow: []

```

/Contacts_100

β_WD.js'

```

→ ~ artillery run test_4_WD.yml
Test run id: tjtgx_wp4rmtzk5wq333maz_n78q
Phase started: 1 per second
...

```

All VUs finished. Total time: 7 seconds

Summary report @ 14:33:21(-0700)

...

...instead watch for repeating metrics...

Metrics for period to: 00:29:30(+0000)
(width: 9.632s)

2) a .js file for Playwright

```

async function wdtest(page) {
  await page.goto('https://d3tests.top/fmi/webd/Contacts_100');
  await page.locator('#t0mainmenubar div').nth(2).click();
  await page.getByText('Close File').click();
  await page.close();
}

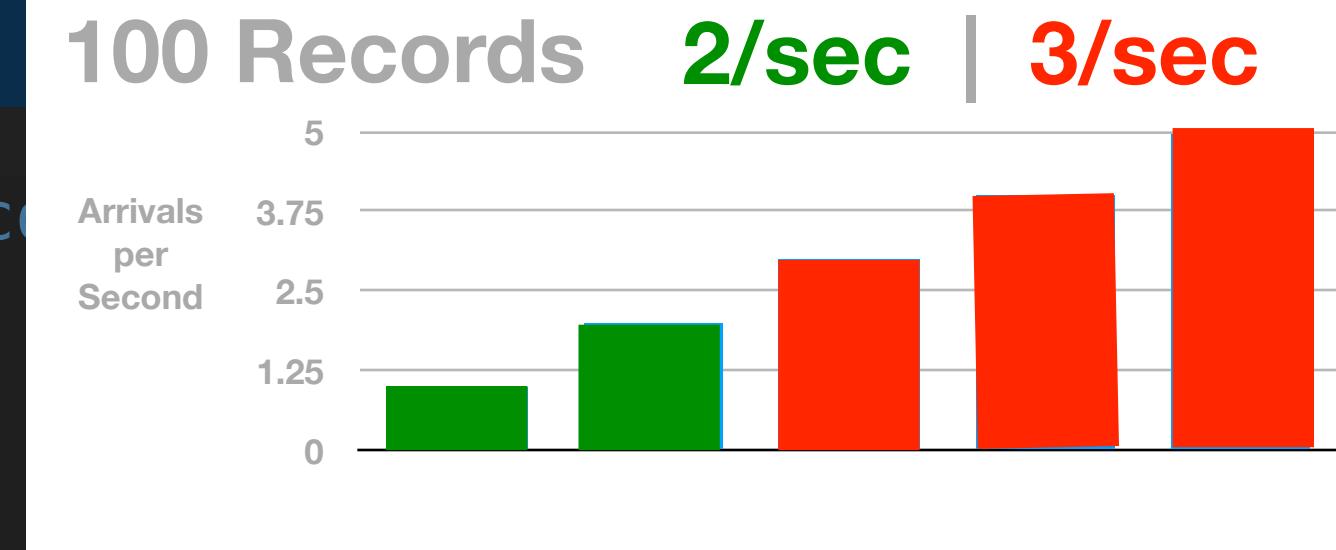
module.exports = { wdtest };

```

...and timeouts

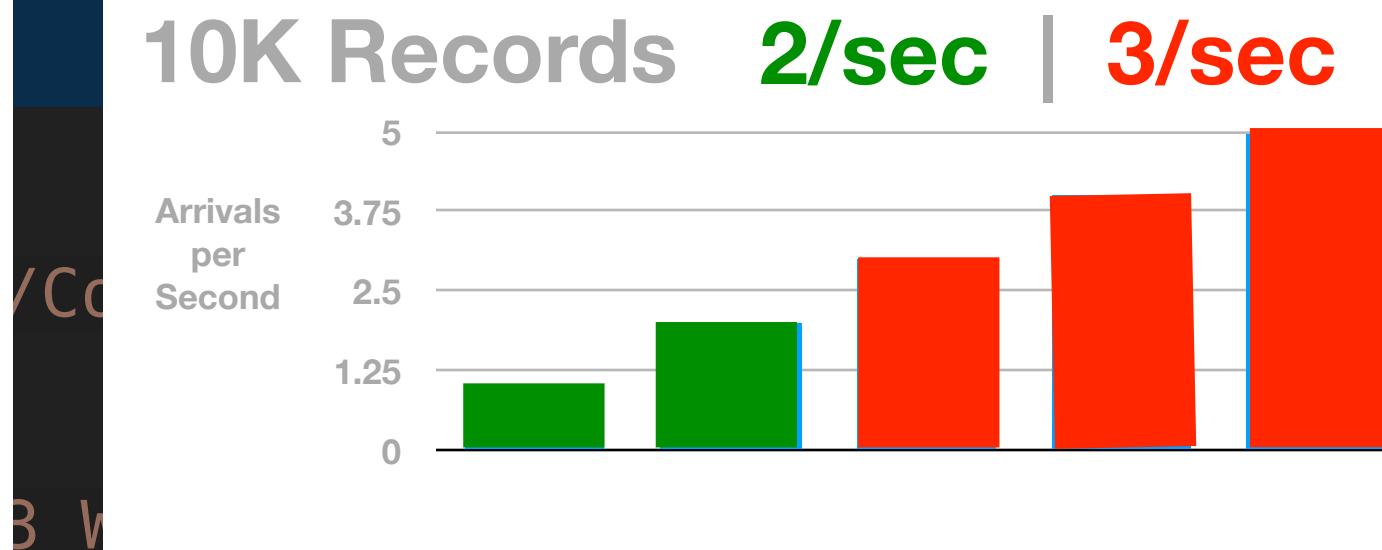
```

:: locator.click: Timeout 30000ms exceeded.
===== logs =====
waiting for locator('#t0mainmenubar
div').nth(2)
=====
```



```
phases:  
- duration: 120  
  arrivalRate: 1  
  name: 1 per second
```

```
scenarios:  
- engine: playwright  
  flowFunction: 'wdtest'  
  flow: []
```



```
rtillery run test_4_WD.yml
n id: tjtgx_wp4rmtzk5wq333maz_n78q
tarted: 1 per second
```

finished. Total time: 7 seconds

Summary report @ 14:33:21(-0700)

10 of 10

...instead watch for repeating metrics...

Metrics for period to: 00:29:30(+0000)
(width: 9.632s)

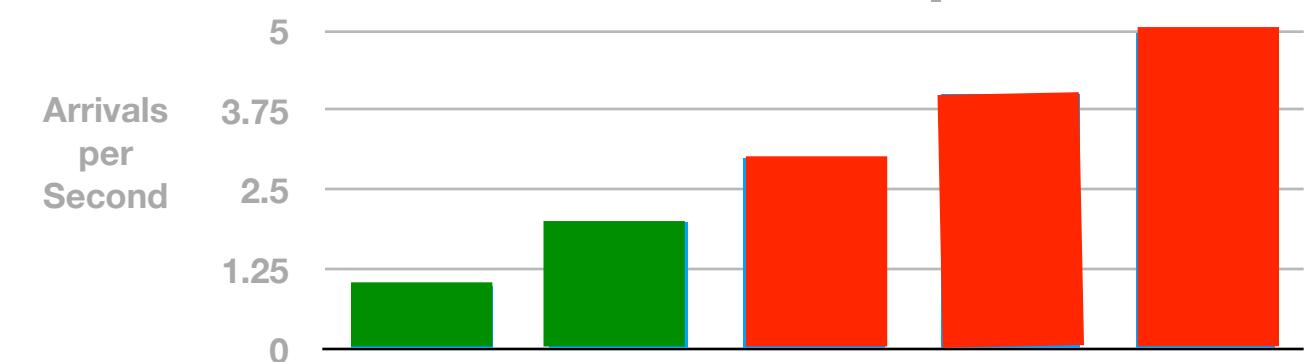
2) a *.js* file for *Playwright*

```
async function wdtest(page) {  
    await page.goto('https://d3tests.top/fmi/webd/Contacts_100');  
    await page.locator('#t0mainmenubar div').nth(2).click();  
    await page.getByText('Close File').click();  
    await page.close();}  
  
module.exports = { wdtest };
```

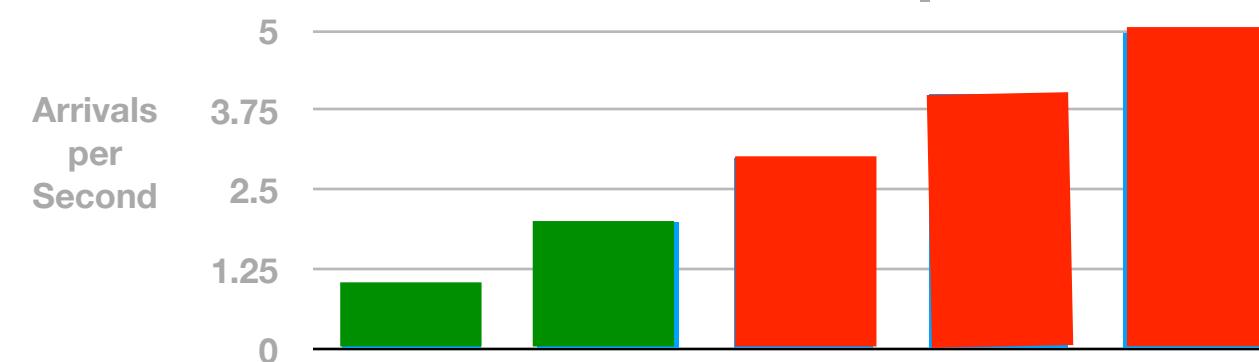
...and timeouts

```
:: locator.click: Timeout 30000ms exceeded.  
===== logs =====  
waiting for locator('#t0mainmenubar  
div').nth(2)
```

100 Records **2/sec | 3/sec**



10K Records **2/sec | 3/sec**



System Overview

85

Total Connections

5 / 5

Hosted Databases

FileMaker Pro 0 Connections

FileMaker Go 0 Connections

FileMaker WebDirect 85 Connections

Additional 0 Connections

Server Name

DO-09

Server IP Addresses

143.198.128.107 10.48.0.5 10.124.0.2

Server Version

20.1.1.38

FileMaker Server License Expiration

06/25/2023, 23:59

SSL Certificate

Expires on 2023-08-14 21:56:08 UTC

Details

FileMaker Data API/OData Annual Limit

4.3 GB / Unrestricted

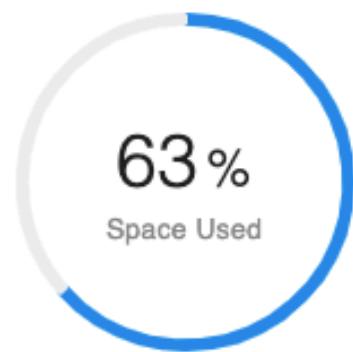
ODBC and JDBC Connections

0 / Unrestricted

FileMaker Script Engine Connections

0

Volume Status



Total Space

24 GB

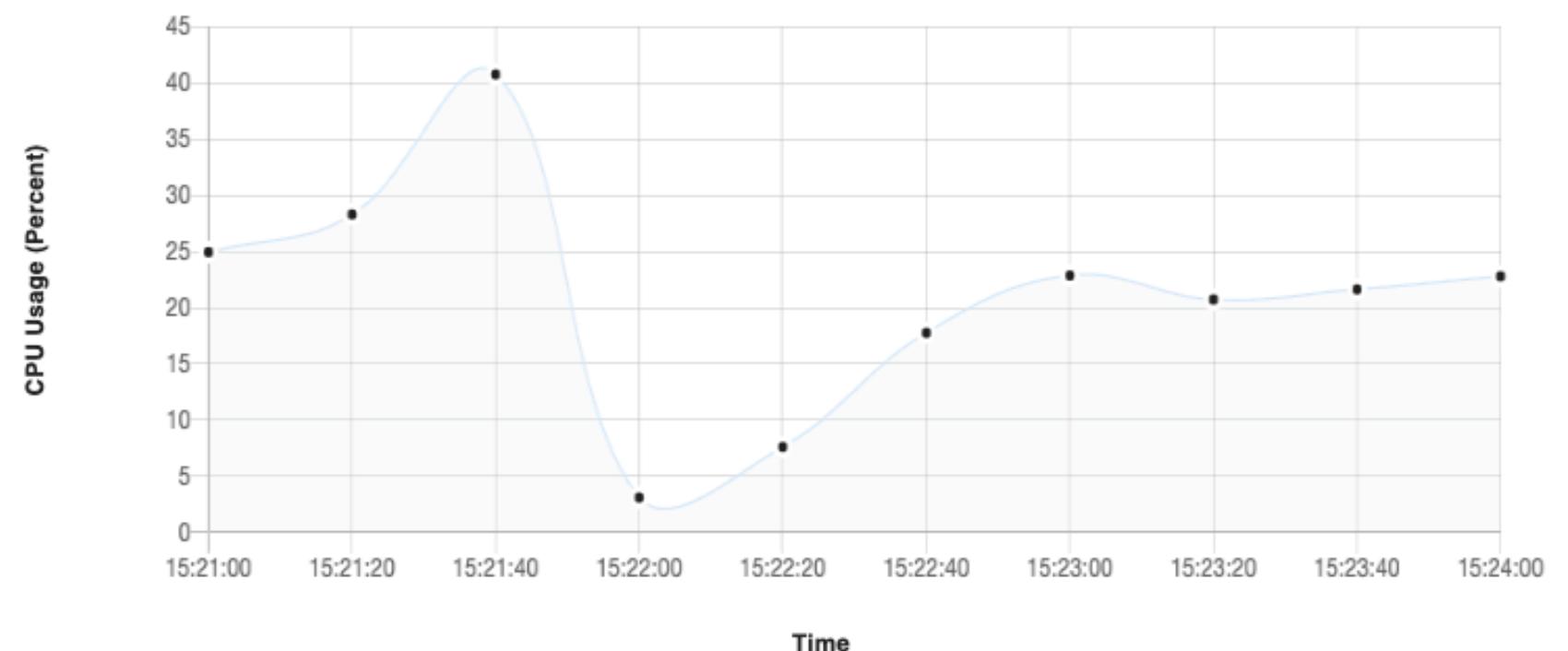
Space Used

15.3 GB

Space Remaining

8.8 GB

System Statistics



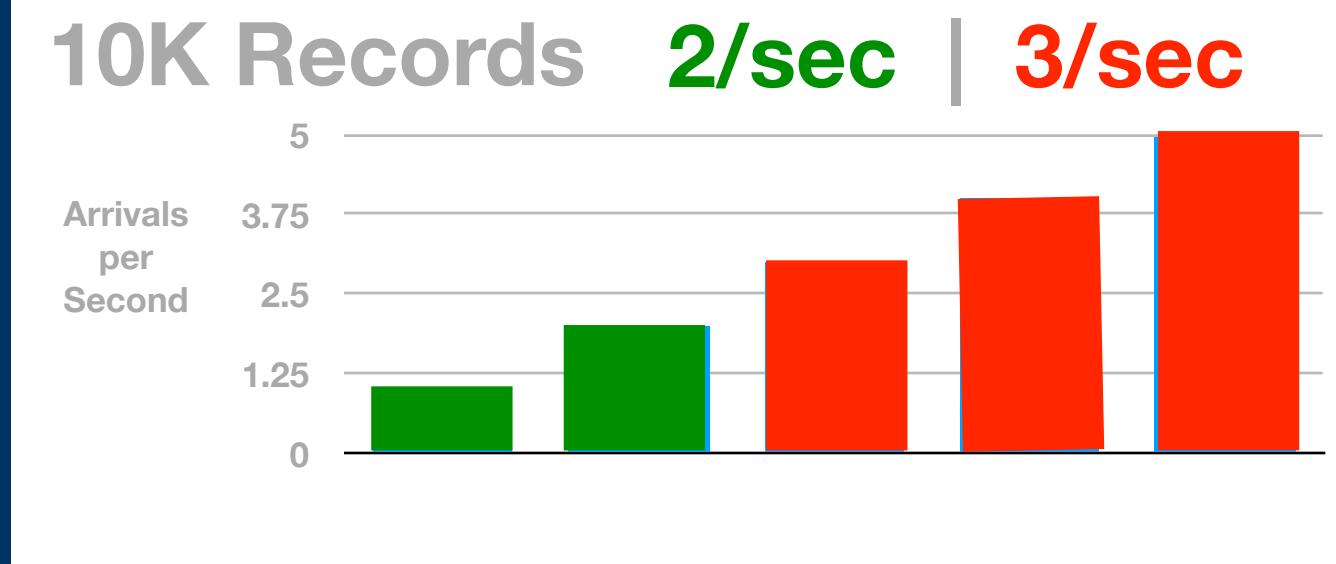
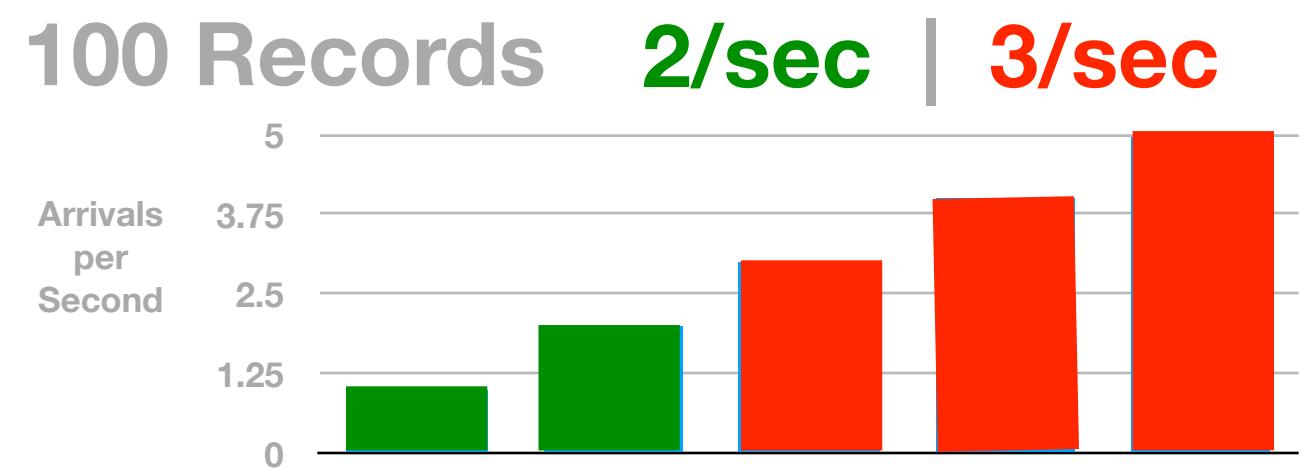
Session Retention over 120 sec

1/sec 0

2/sec ~35 sessions, cleared

3/sec ~85 sessions, timeouts

4/sec ~120 sessions, crashed



Conclusions?

More testing needed



Artillery

*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*



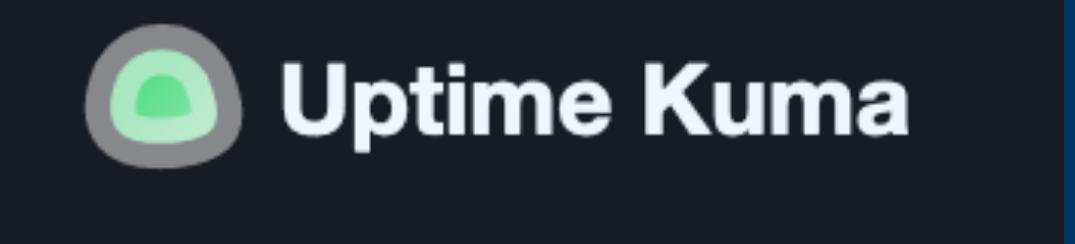
Uptime Kuma

*“A fancy
self-hosted monitoring tool”*

*Open Source Docker image
website UI*

- ✓ *ping responders*
- ✓ *Data API*
- ✓ *WebDirect*

The Layers...



The Layers...

Docker Compose



Uptime Kuma

The Layers...

Docker

Docker Compose



Uptime Kuma

The Layers...

Your OS

Docker

Docker Compose



Uptime Kuma

Setup...

- *check for Docker*

```
→ ~ docker version
```

Setup...

- check for Docker

```
→ ~ docker version  
Client: Docker Engine - Community  
Version: 24.0.2  
...
```

Setup...

- *check for Docker*
- *install if necessary*
- *download install script*

```
→ ~ curl -fsSL https://get.docker.com -o  
get-docker.sh  
...  
→ ~
```

Setup...

- *check for Docker*
- *install if necessary*
- *download install script*
- *run install script*

```
→ ~ curl -fsSL https://get.docker.com -o  
get-docker.sh  
...  
→ ~ sudo sh get-docker.sh  
# Executing docker install script  
...
```

Setup...

- check for Docker

```
→ ~ docker version  
Client: Docker Engine - Community  
Version: 24.0.2  
...
```

Setup...

- *check for Docker*
- *check for Docker Compose*
 - *install if necessary*

(Ubuntu: installed along with Docker)

```
→ ~ docker compose version
```

```
Docker Compose version v2.18.1
```

Setup...

- *check for Docker*
- *check for Docker Compose*
- *prepare directories*



Setup...

- *check for Docker*
- *check for Docker Compose*
- *prepare directories*
 - *navigate to your home directory*

```
→ ~ cd ~  
→ ~
```

Setup...

- *check for Docker*
- *check for Docker Compose*
- *prepare directories*
 - *navigate to your home directory*
 - *make docker_volumes directory*

```
→ ~ cd ~  
→ ~ mkdir docker_volumes  
→ ~
```

Setup...

- *check for Docker*
- *check for Docker Compose*
- *prepare directories*
 - *navigate to your home directory*
 - *make docker_volumes directory*
 - *make directory for UK*

```
→ ~ cd ~  
→ ~ mkdir docker_volumes  
→ ~ mkdir uptime-kum  
→ ~
```

Setup...

- *check for Docker*
- *check for Docker Compose*
- *prepare directories*
 - *navigate to your home directory*
 - *make docker_volumes directory*
 - *make directory for UK*
 - *navigate there*

```
→ ~ cd ~  
→ ~ mkdir docker_volumes  
→ ~ mkdir uptime-kum  
→ ~ cd uptime-kuma  
→ ~
```

Setup...

- *check for Docker*
- *check for Docker Compose*
- *prepare directories*
 - *navigate to your home directory*
 - *make docker_volumes directory*
 - *make directory for UK*
 - *navigate there*
 - *make directory for UK data*

```
→ ~ cd ~  
→ ~ mkdir docker_volumes  
→ ~ mkdir uptime-kum  
→ ~ cd uptime-kuma  
→ ~ mkdir data
```

Setup...

- *check for Docker*
- *check for Docker Compose*
- *prepare directories*
- *make Compose .yml file*

```
→ ~ touch docker-compose.yml
```

Setup...

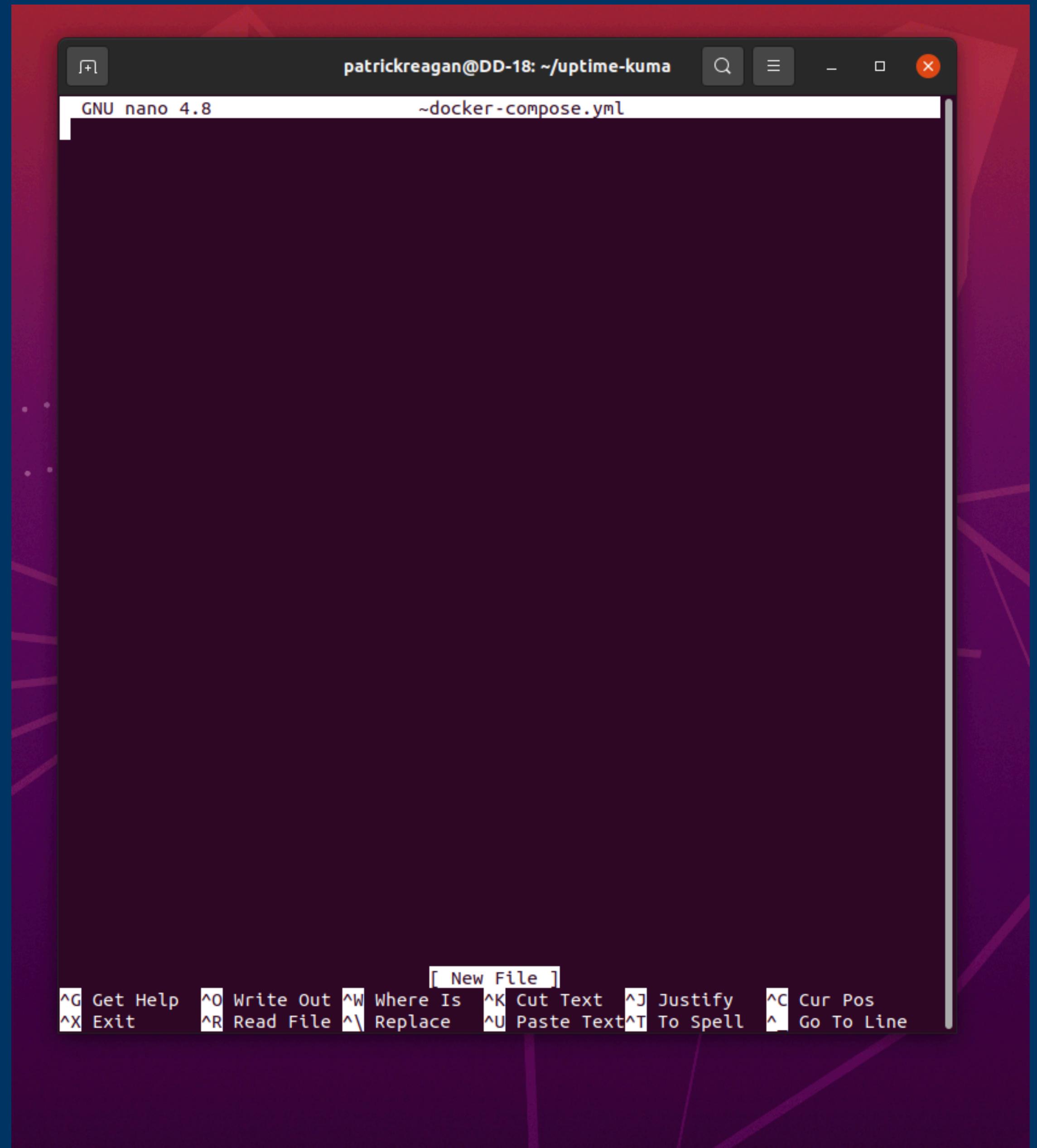
- *check for Docker*
- *check for Docker Compose*
- *prepare directories*
- *make Compose .yml file*
- *open in nano text editor*

```
→ ~ touch docker-compose.yml
```

```
→ ~ nano docker-compose.yml
```

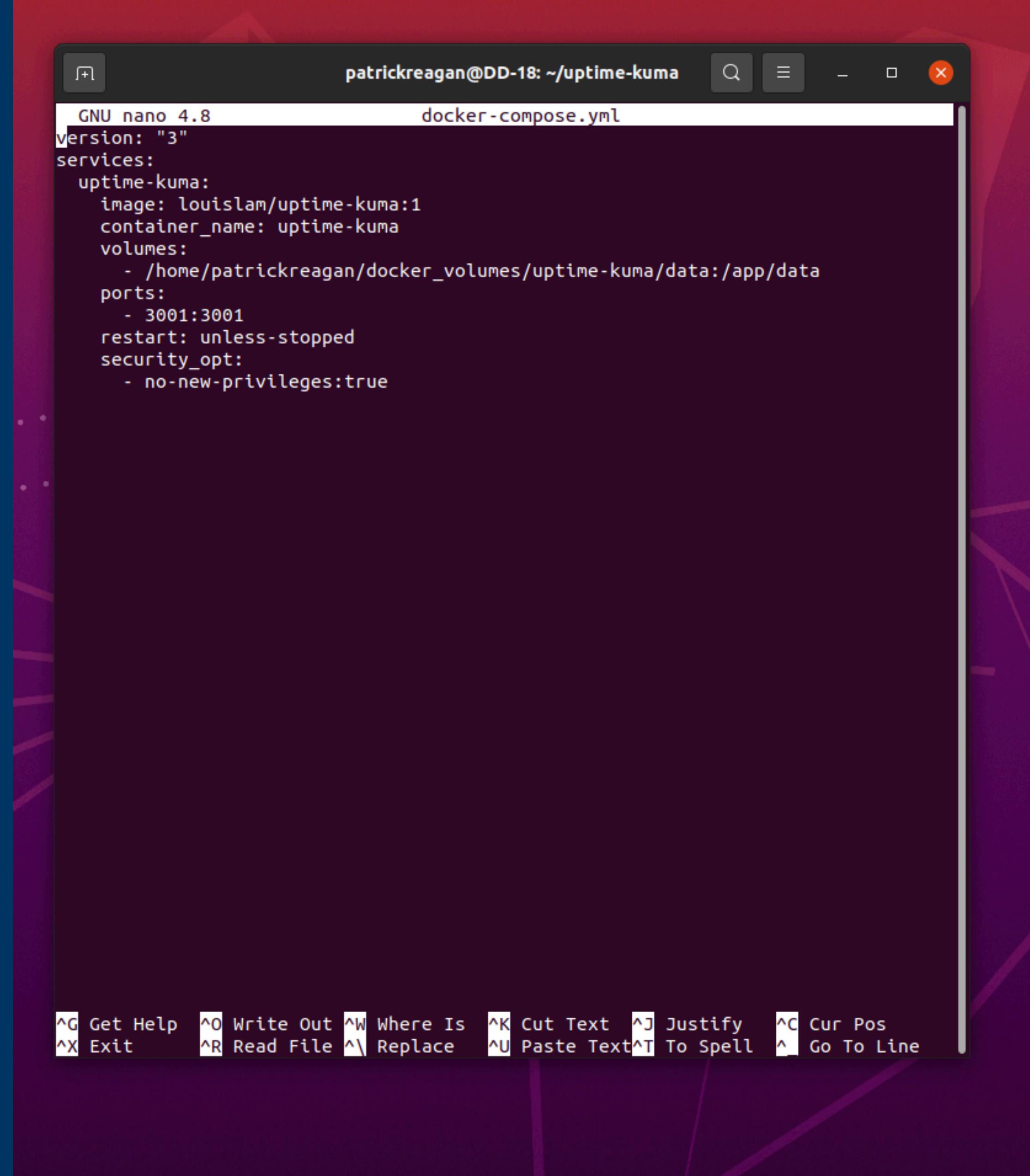
Setup...

- *check for Docker*
- *check for Docker Compose*
- *prepare directories*
- *make Compose .yml file*
- *open in nano text editor*



Setup...

- *check for Docker*
- *check for Docker Compose*
- *prepare directories*
- *make Compose .yml file*
- *open in nano text editor*



The screenshot shows a terminal window titled "patrickreagan@DD-18: ~/uptime-kuma" running the "GNU nano 4.8" text editor. The file being edited is named "docker-compose.yml". The content of the file is as follows:

```
version: "3"
services:
  uptime-kuma:
    image: louislam/uptime-kuma:1
    container_name: uptime-kuma
    volumes:
      - /home/patrickreagan/docker_volumes/uptime-kuma/data:/app/data
    ports:
      - 3001:3001
    restart: unless-stopped
    security_opt:
      - no-new-privileges:true
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts for navigating and editing the text.

Setup...

- *check for Docker*
- *check for Docker Compose*
- *prepare directories*
- *make Compose .yml file*
- *open in nano text editor*
- *run the .yml file*

```
→ ~ sudo docker compose up -d --force-recreate
```

Setup...

- *check for Docker*
- *check for Docker Compose*
- *prepare directories*
- *make Compose .yml file*
- *open in nano text editor*
- *run the .yml file*

```
➔ ~ sudo docker compose up -d --force-recreate
...
...
✓ Container uptime-kuma          Started
```

Setup...

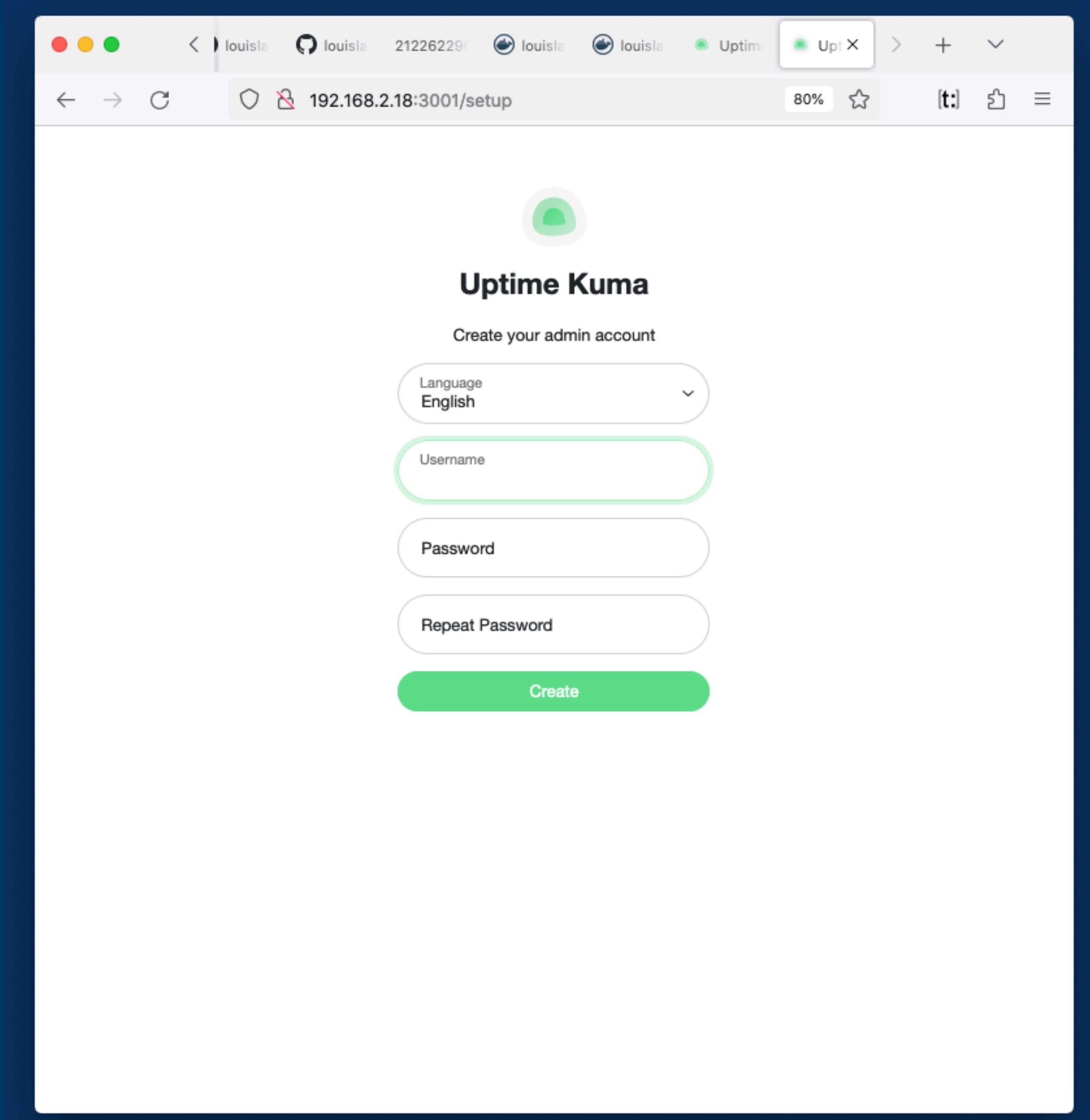
- *check for Docker*
- *check for Docker Compose*
- *prepare directories*
- *make Compose .yml file*
- *open in nano text editor*
- *run the .yml file*
- *open the web interface using browser*

http://<your machine>:3001

http://192.168.2.18:3001

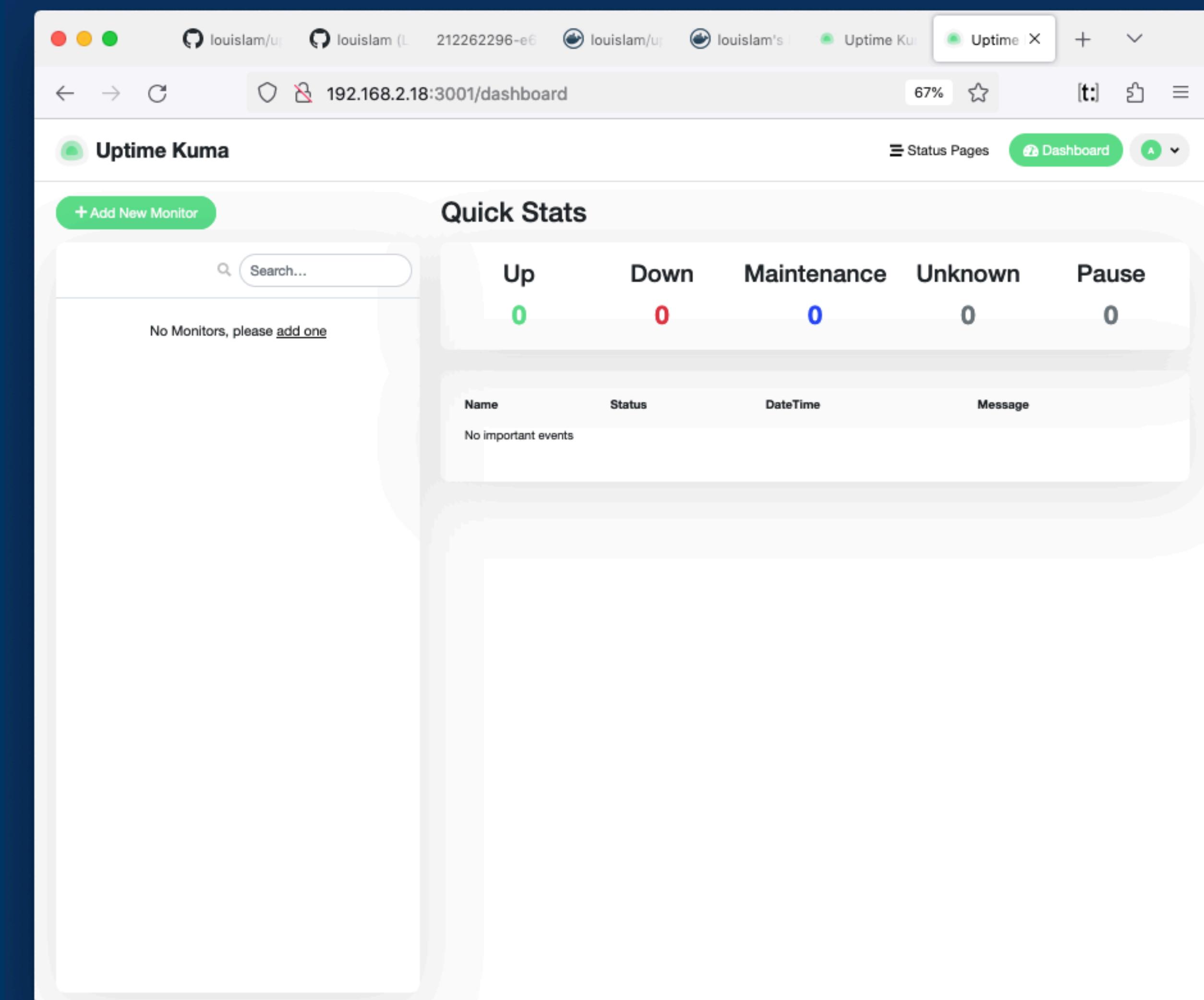
Setup...

- *check for Docker*
- *check for Docker Compose*
- *prepare directories*
- *make Compose .yml file*
- *open in nano text editor*
- *run the .yml file*
- *open the web interface using browser*
- *establish your credentials*



Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode



Settings

Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode

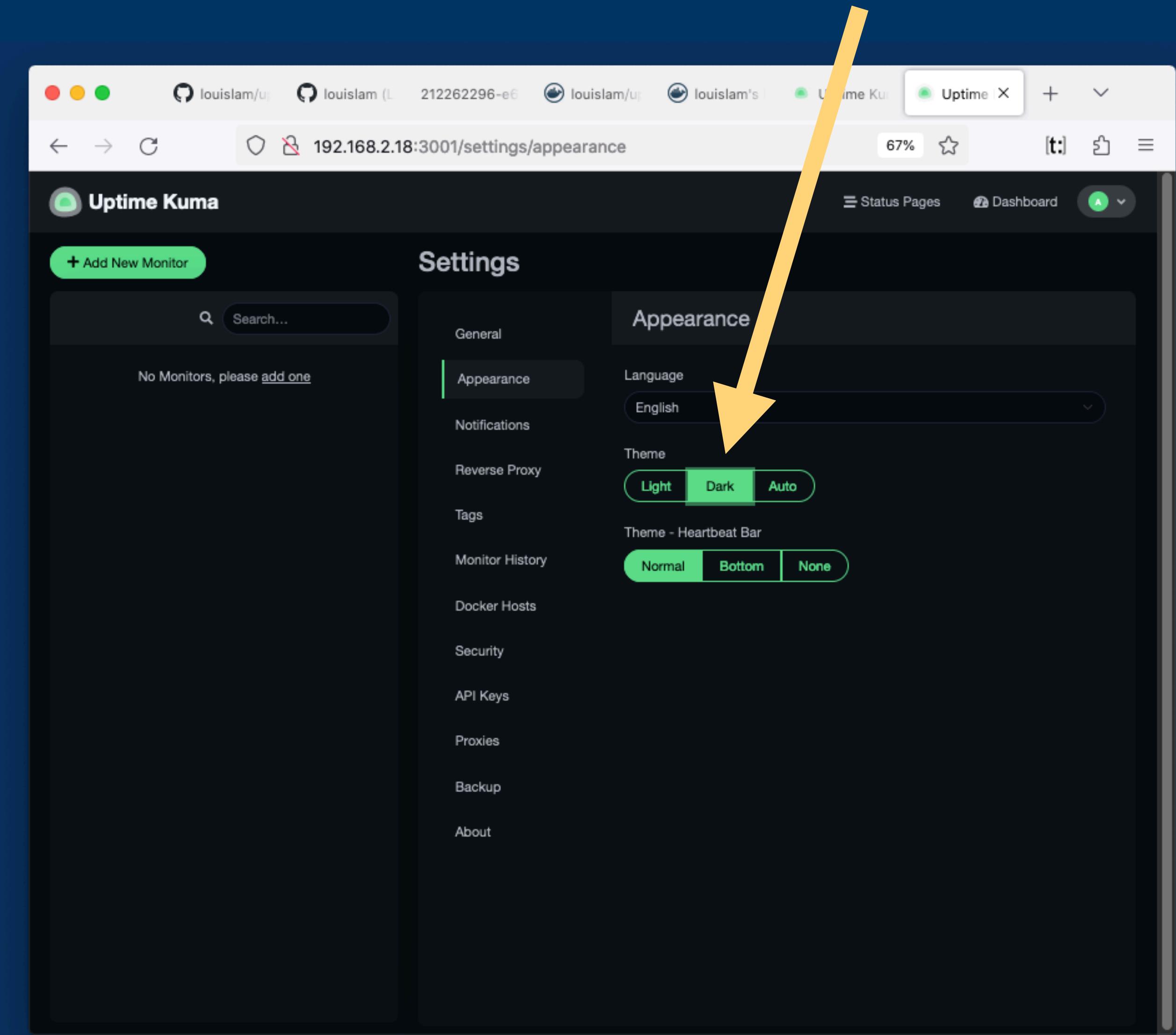
The screenshot shows the Uptime Kuma web interface running in a browser window. The title bar indicates the URL is 192.168.2.18:3001/dashboard. The main area is titled "Uptime Kuma" and features a "Quick Stats" section with a table showing 0 Up, 0 Down, 0 Maintenance, 0 Unknown, and 0 Pause monitors. Below this is a table with columns: Name, Status, DateTime, and Message, which currently displays "No important events". A yellow arrow points to the green "Dashboard" button in the top right corner of the interface.

Up	Down	Maintenance	Unknown	Pause
0	0	0	0	0

Name	Status	DateTime	Message
No important events			

Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode



Setup...

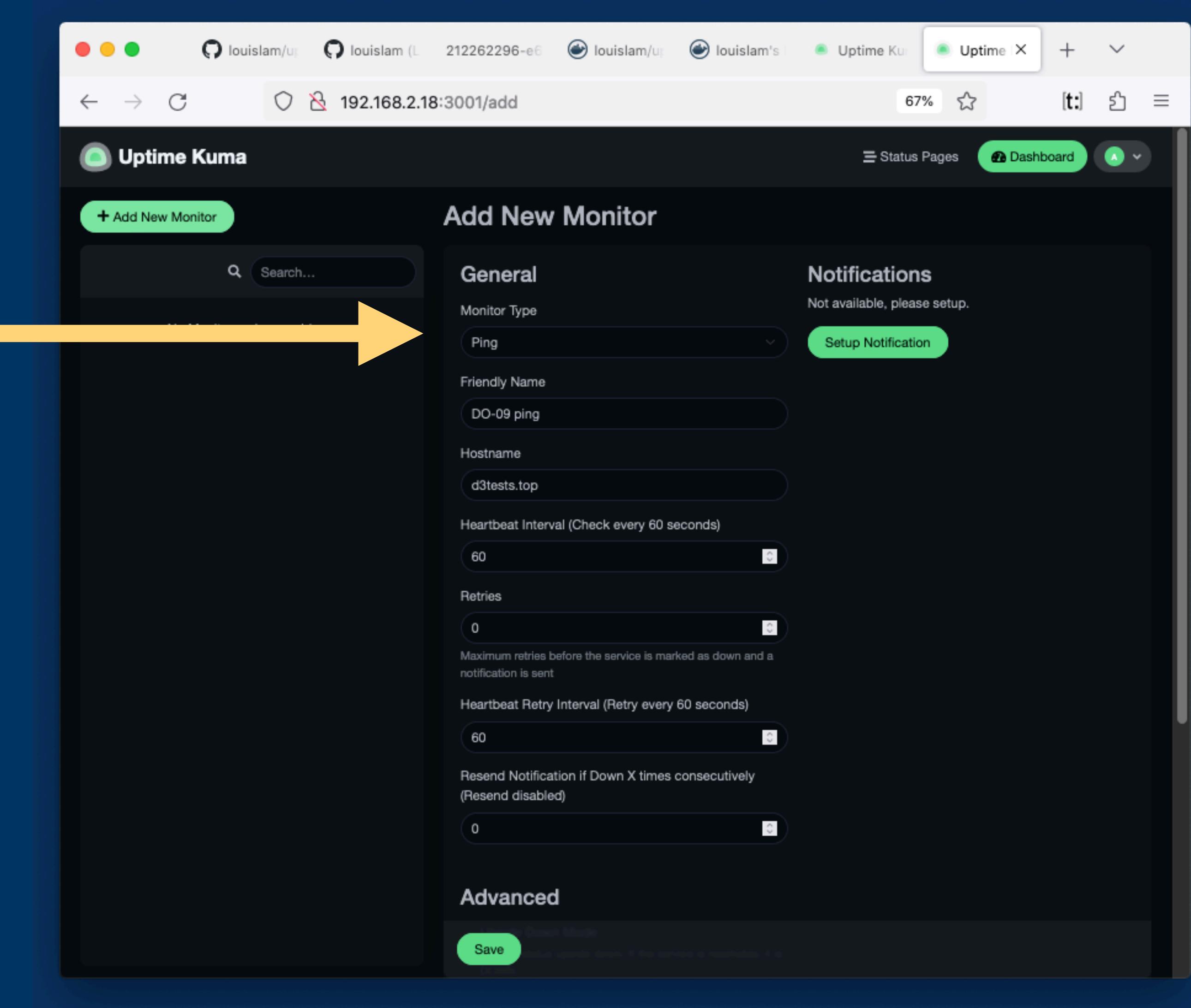
- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping



The screenshot shows the 'Uptime Kuma' web interface with a dark theme. The top navigation bar includes links for Status Pages, Dashboard, and a user icon. The main content area is titled 'Settings' and contains a 'General' section with a 'No Monitors, please [add one](#)' message, a search bar, and a green '+ Add New Monitor' button. To the right is the 'Appearance' section, which includes settings for Language (English), Theme (Dark selected), and Theme - Heartbeat Bar (Bottom selected). Other sections visible include Reverse Proxy, Tags, Monitor History, Docker Hosts, Security, API Keys, Proxies, Backup, and About.

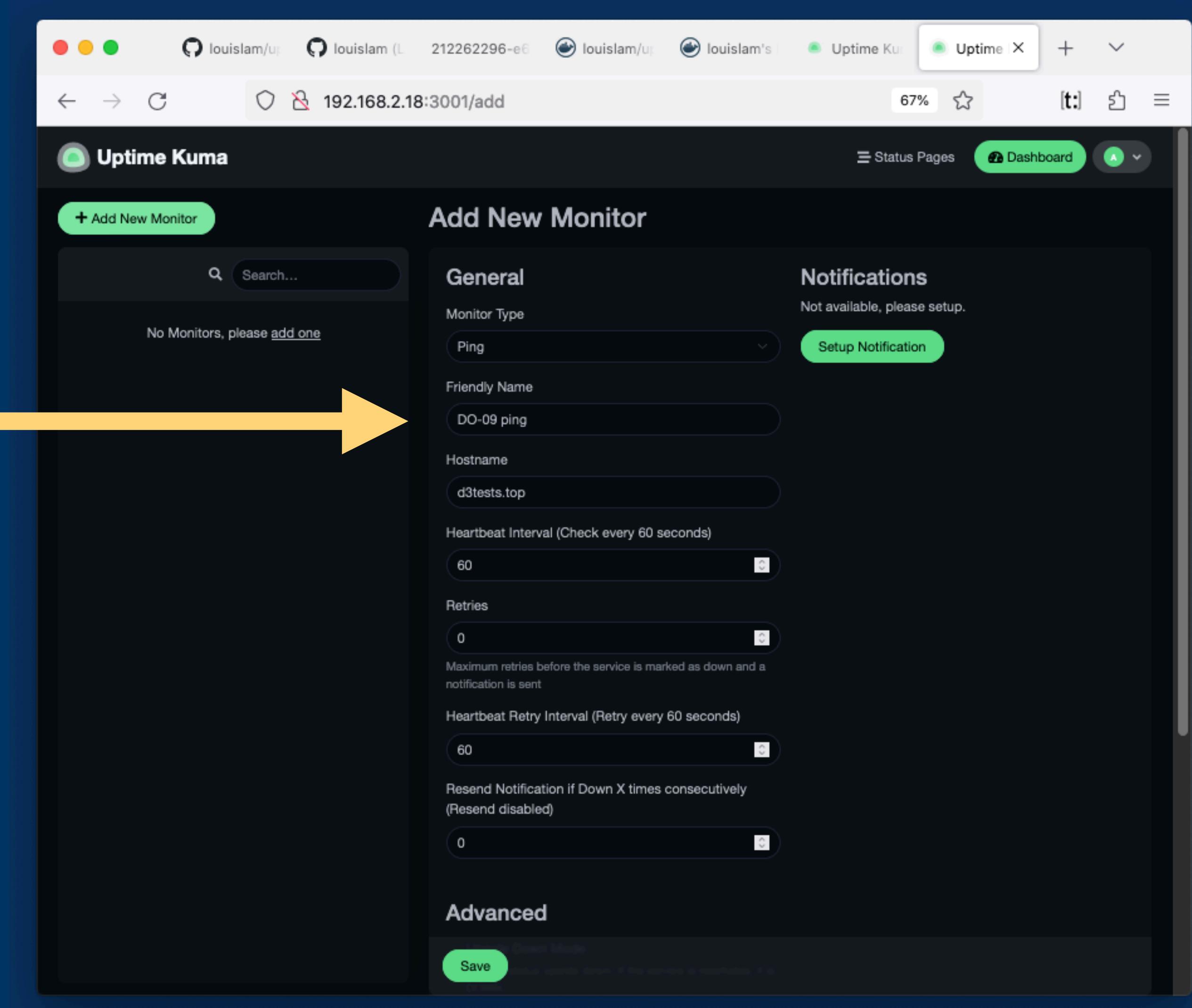
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping



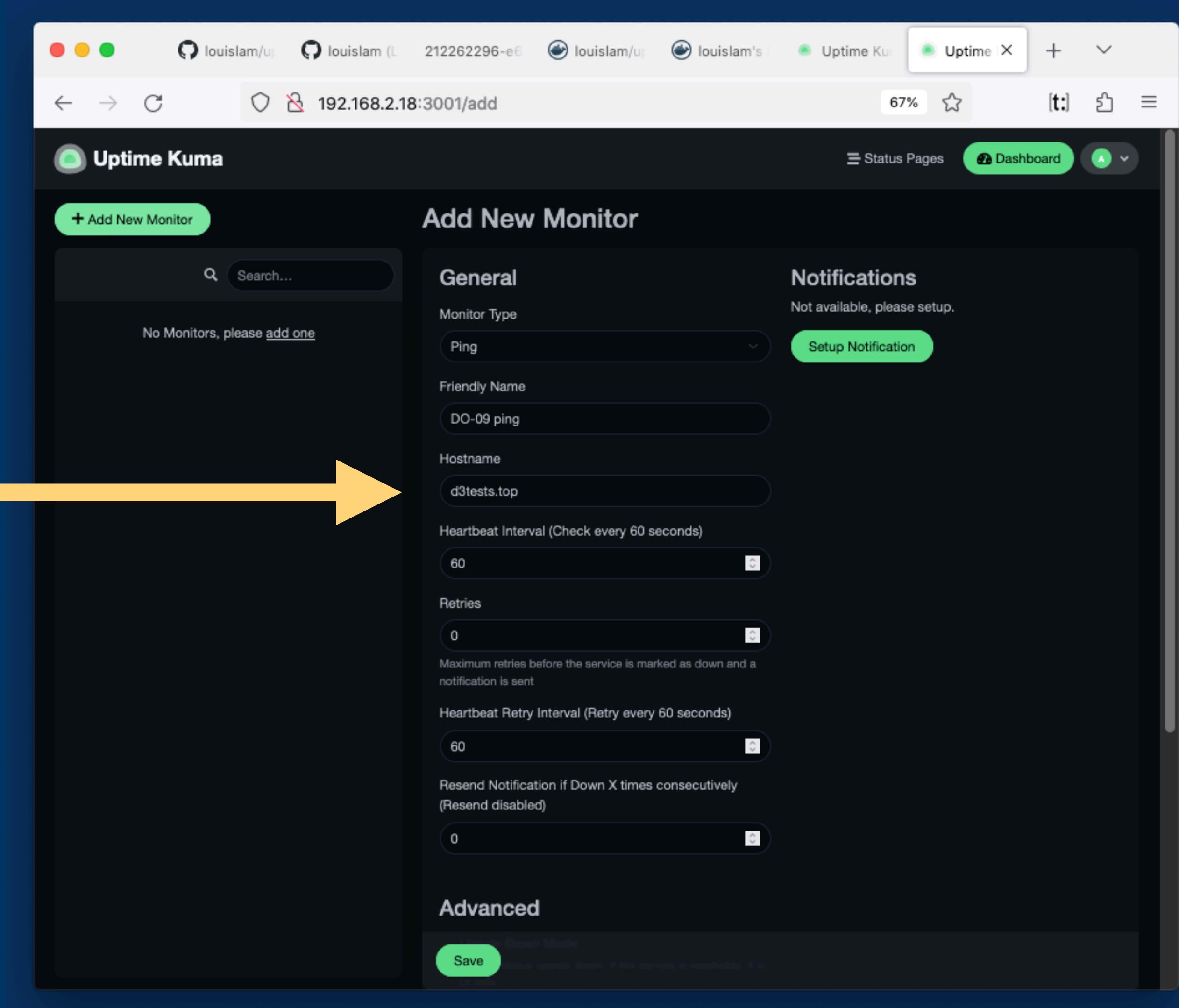
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping



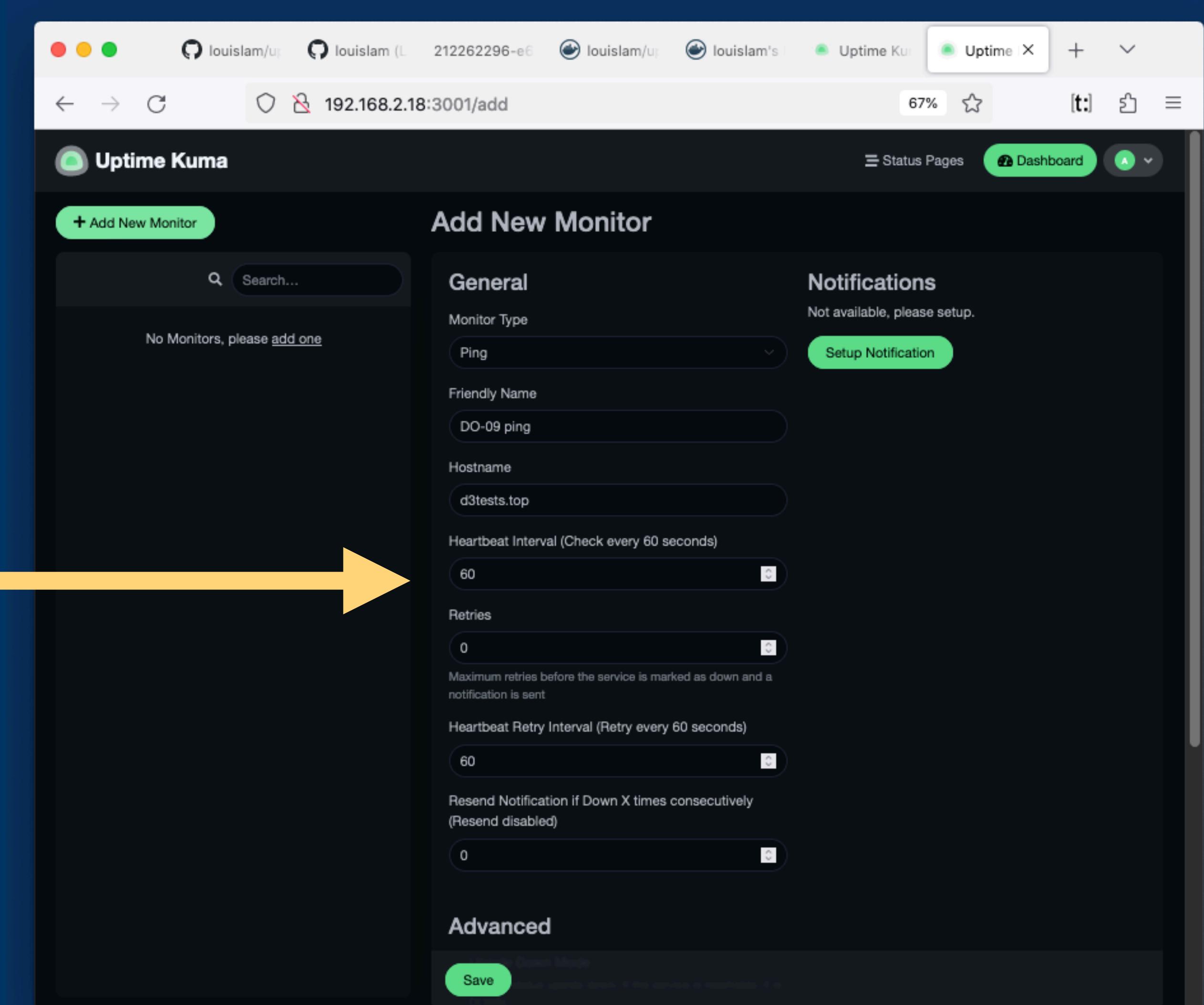
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping



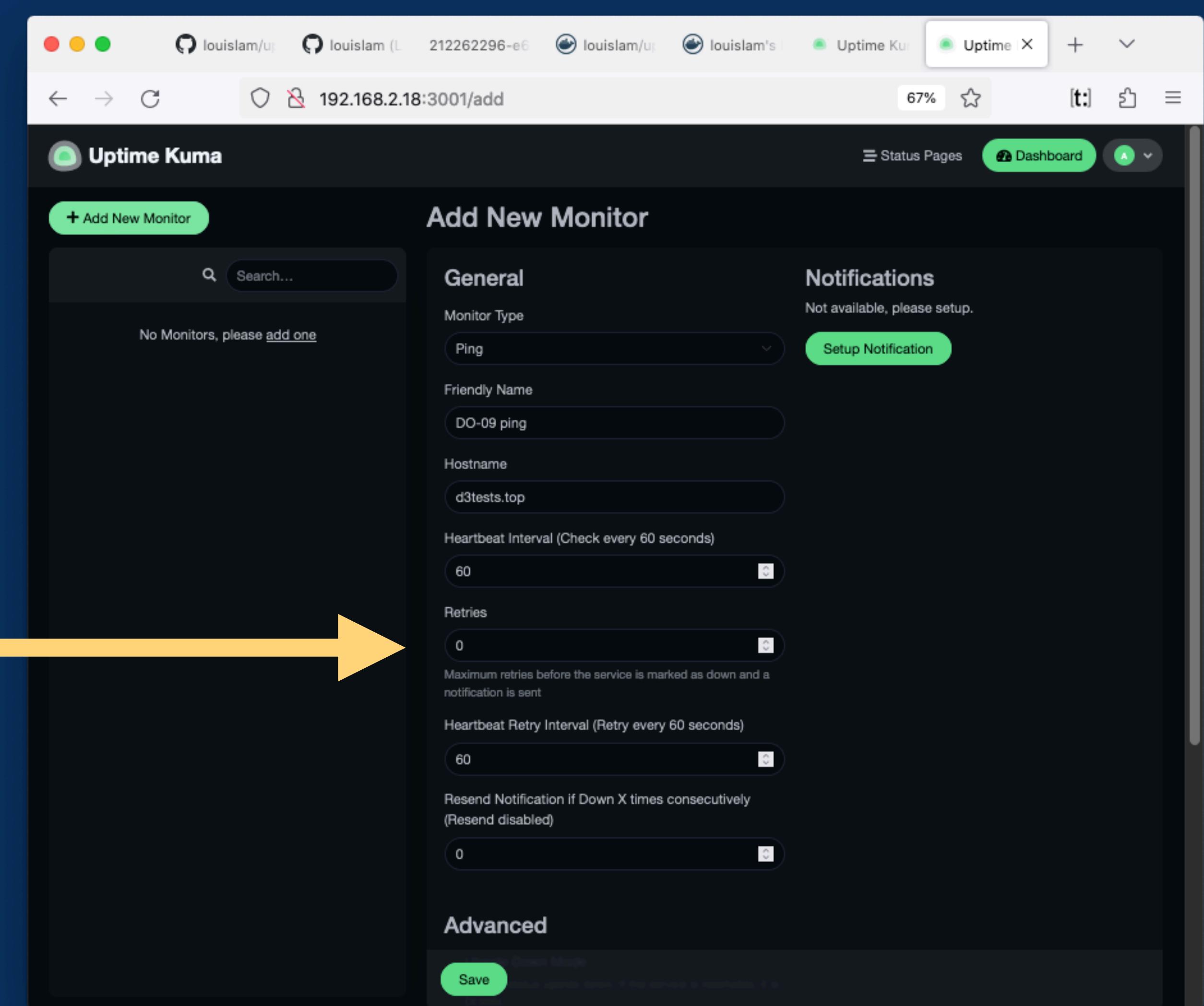
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping



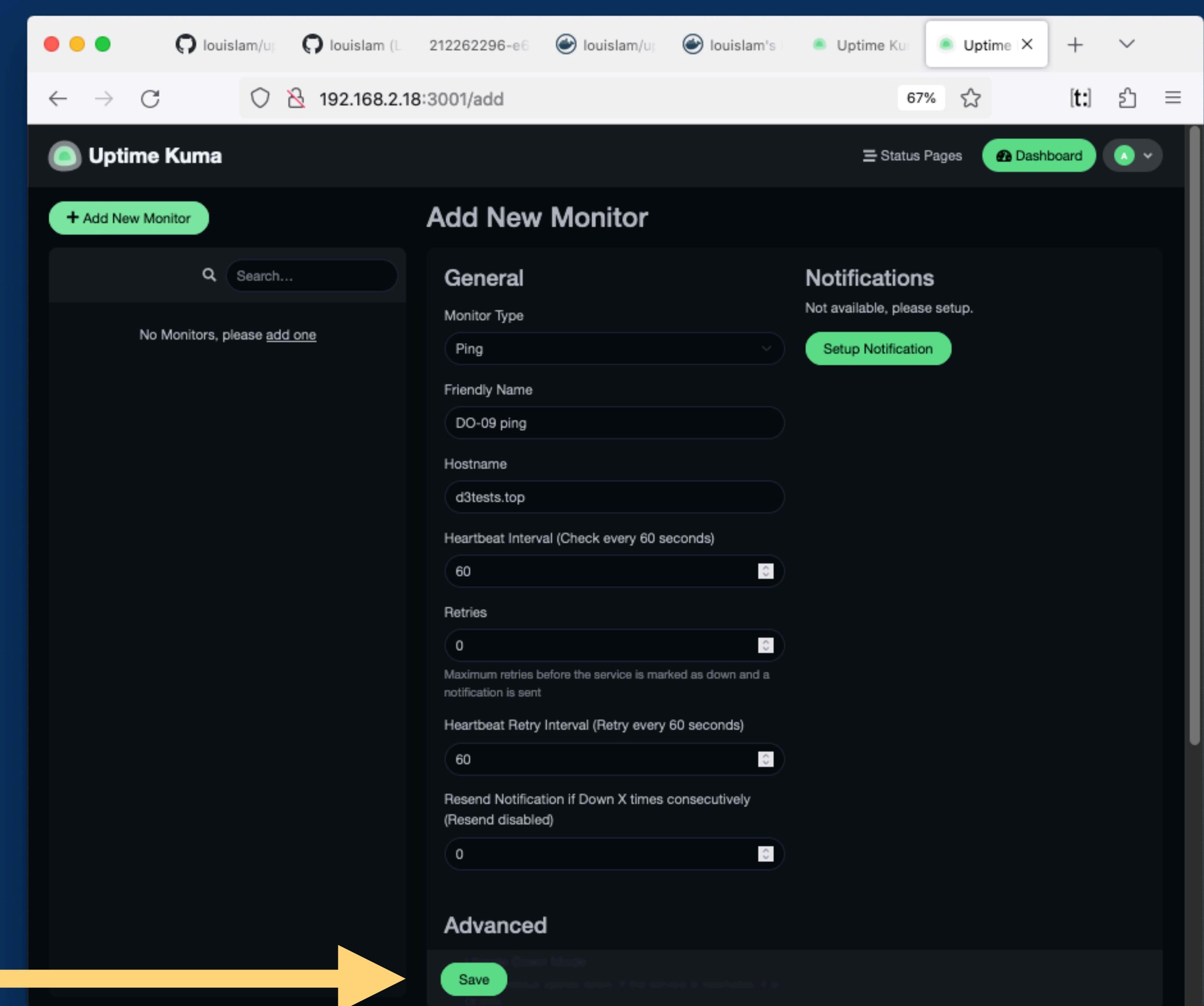
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping



Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping



Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping

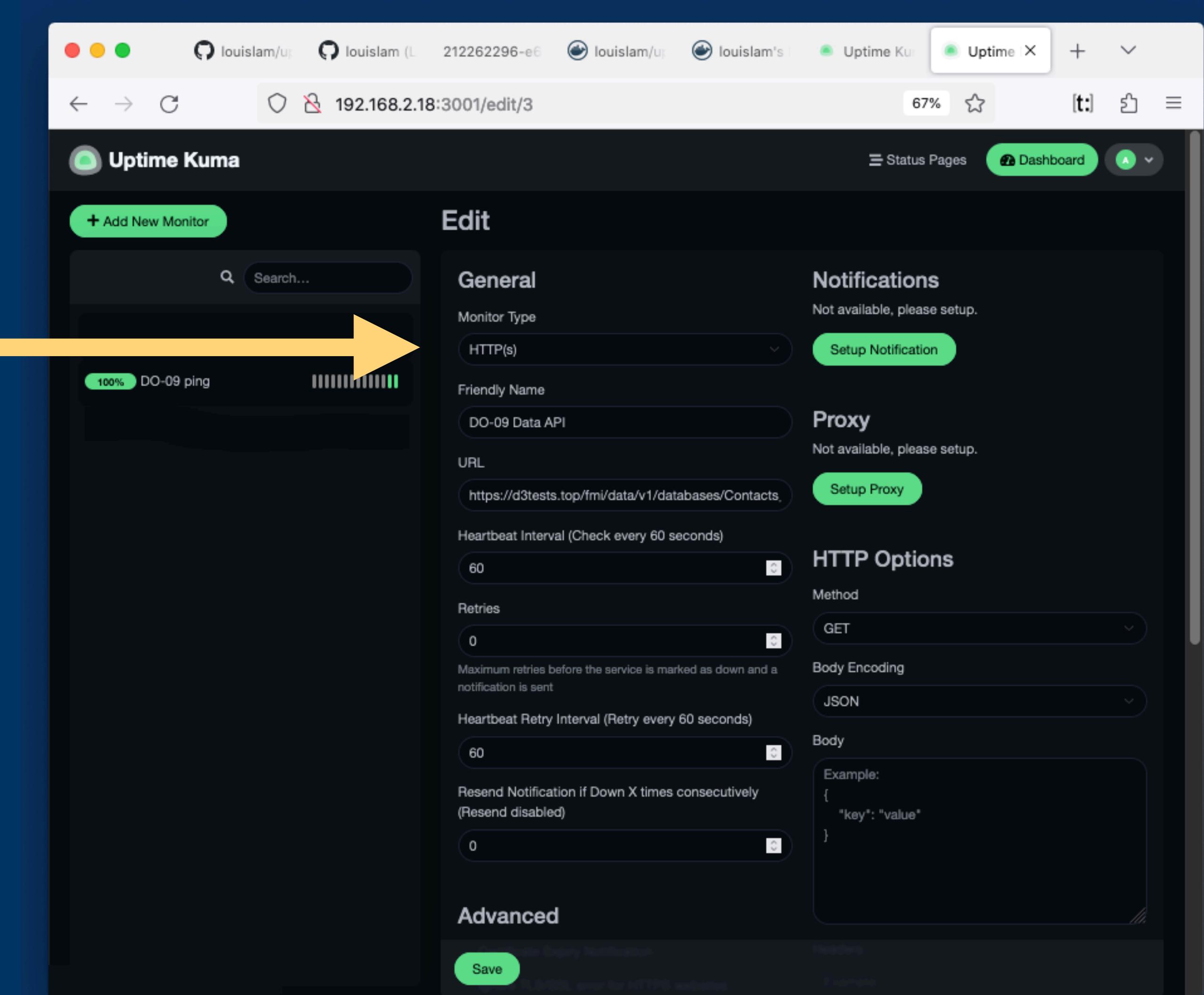


The screenshot shows the 'Edit' screen for a new monitor in the Uptime Kuma web interface. The monitor type is set to 'HTTP(s)' and is named 'DO-09 Data API'. The URL is set to <https://d3tests.top/fmi/data/v1/databases/Contacts>. The heartbeat interval is set to 60 seconds, and the retries are set to 0. The advanced section includes an example JSON body:

```
{  
  "key": "value"  
}
```

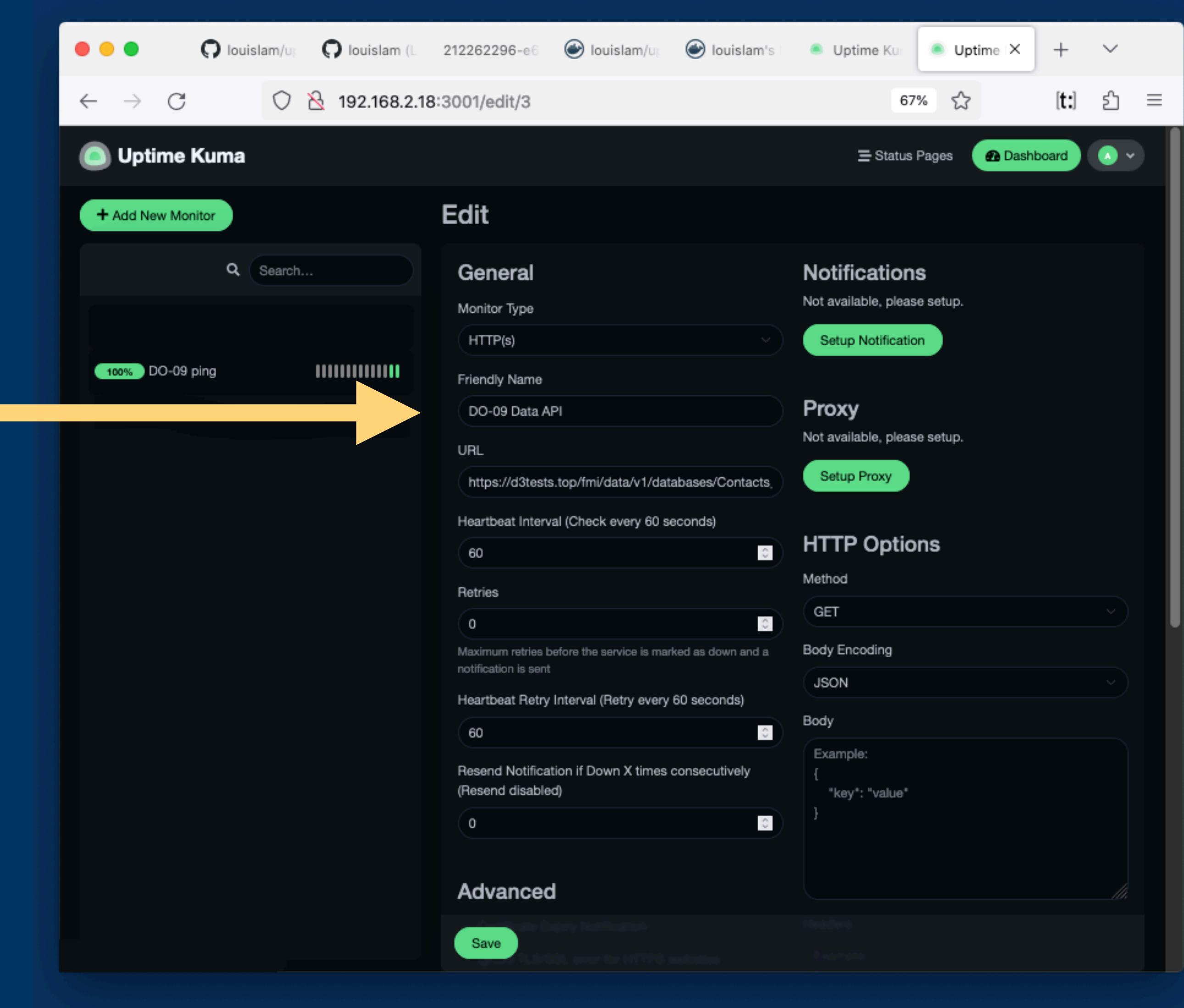
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping
 - API



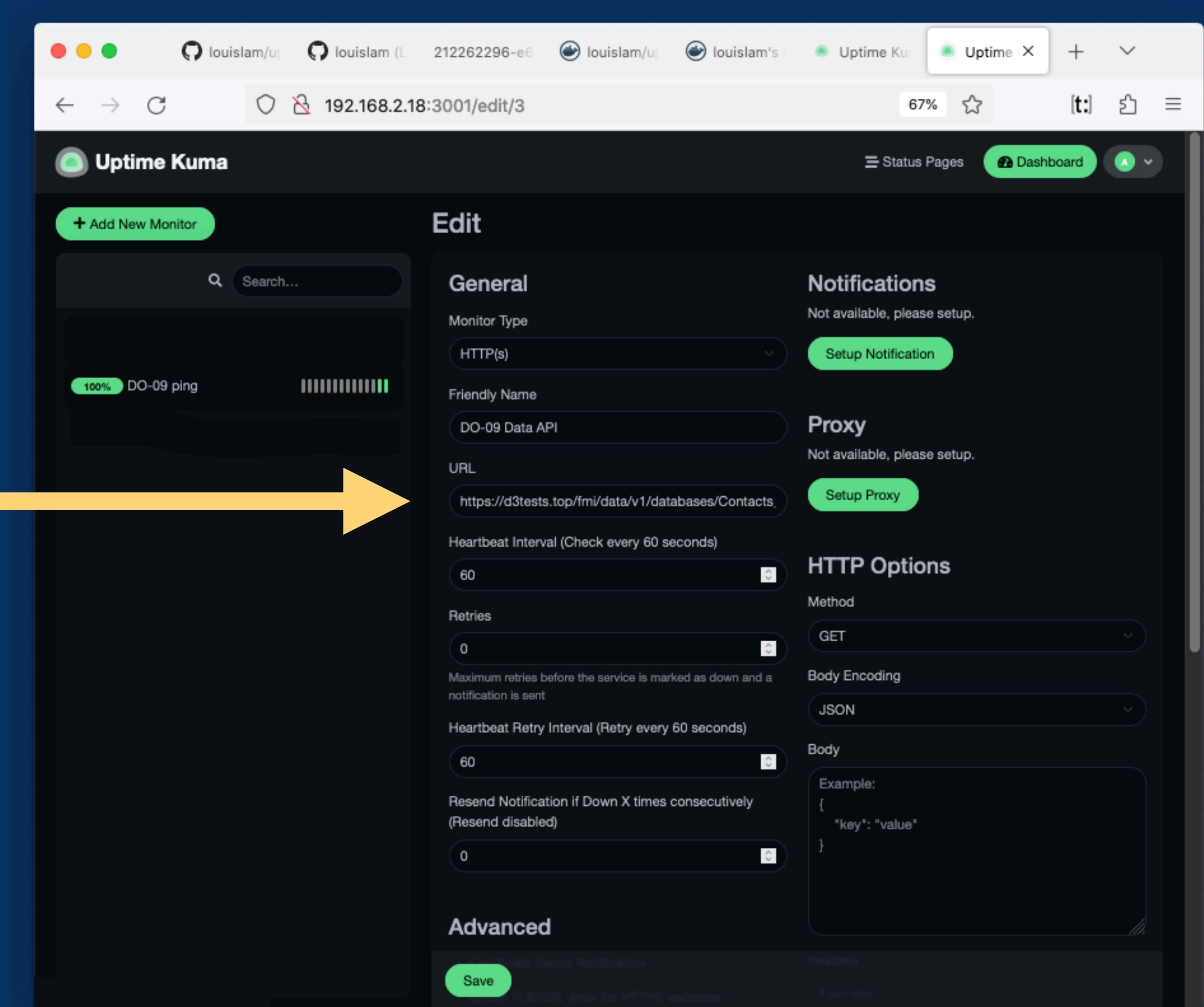
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping
 - API



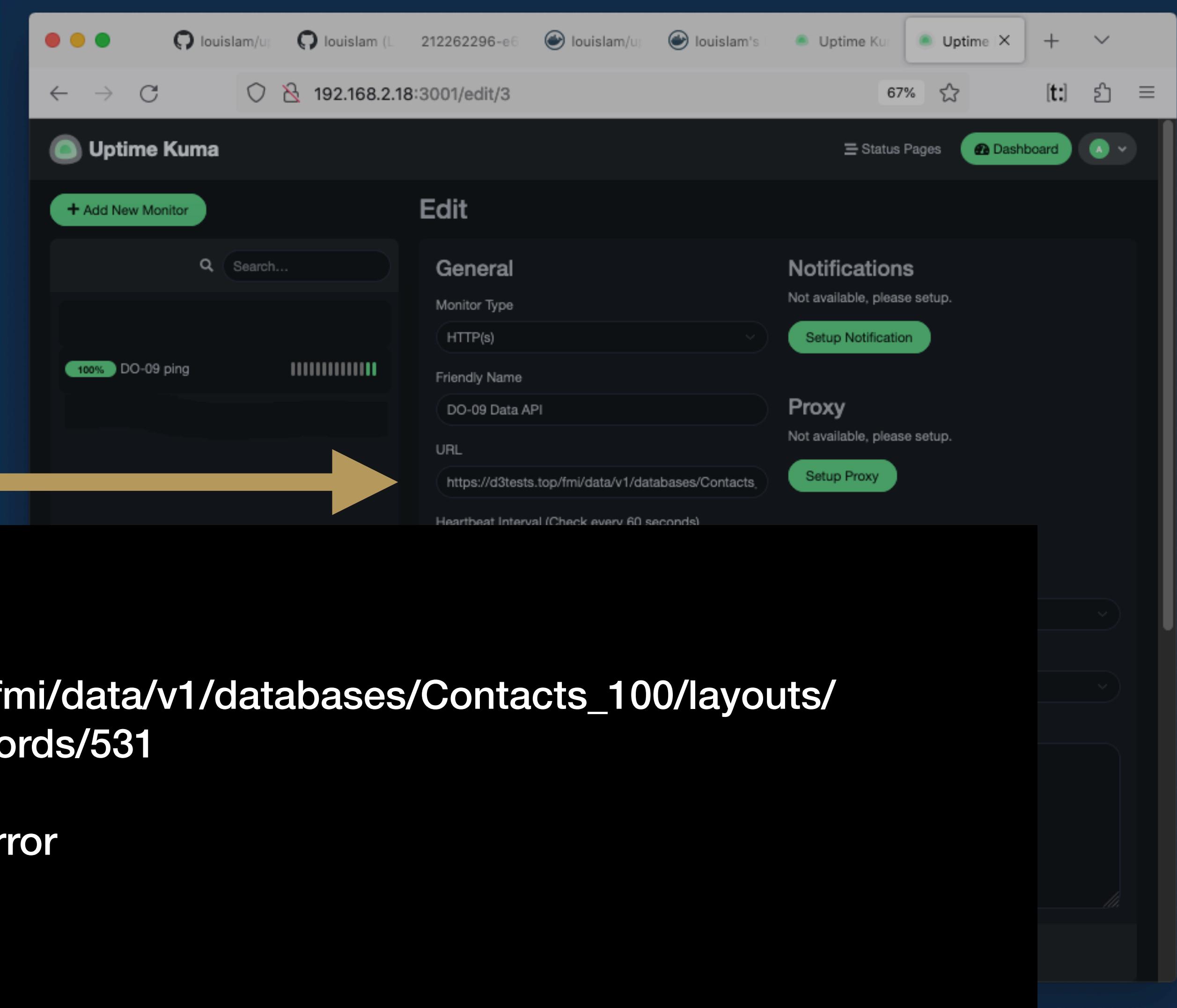
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping
 - API



Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface
- establish your connection
- change to night mode
- add new monitor
- ping
- API

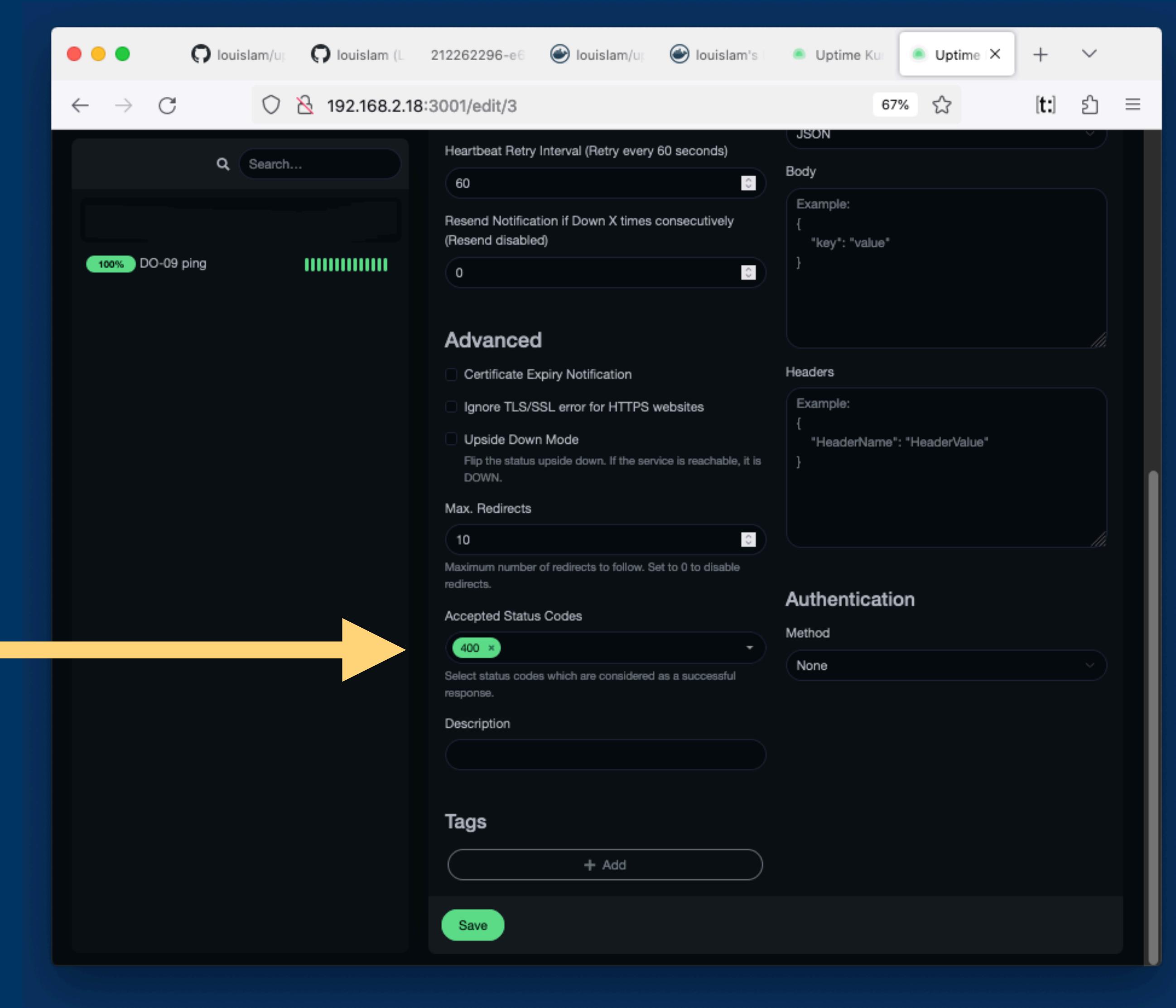


`https://d3tests.top/fmi/data/v1/databases/Contacts_100/layouts/contact_list_api/records/531`

No token => 400 error

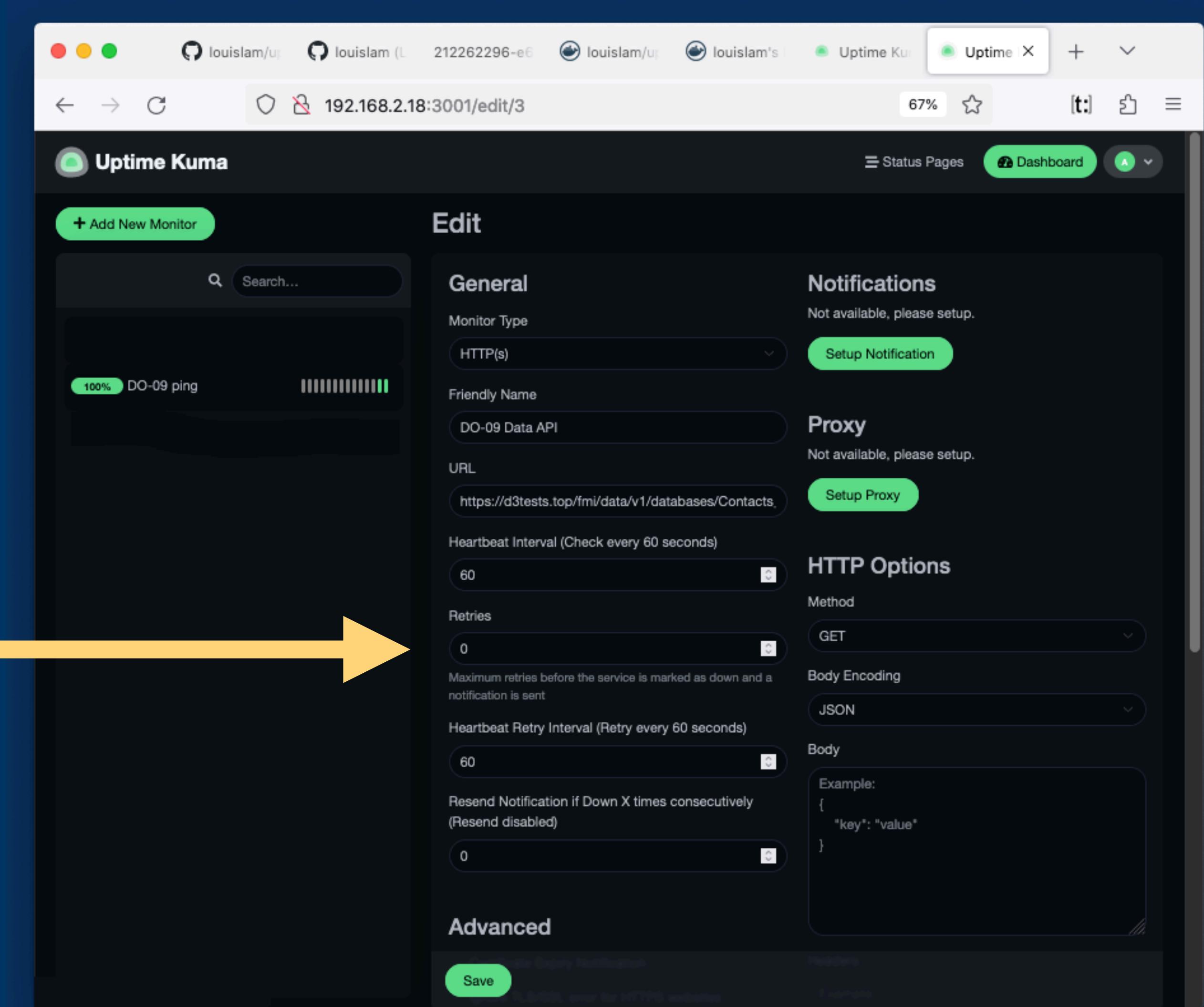
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping
 - API



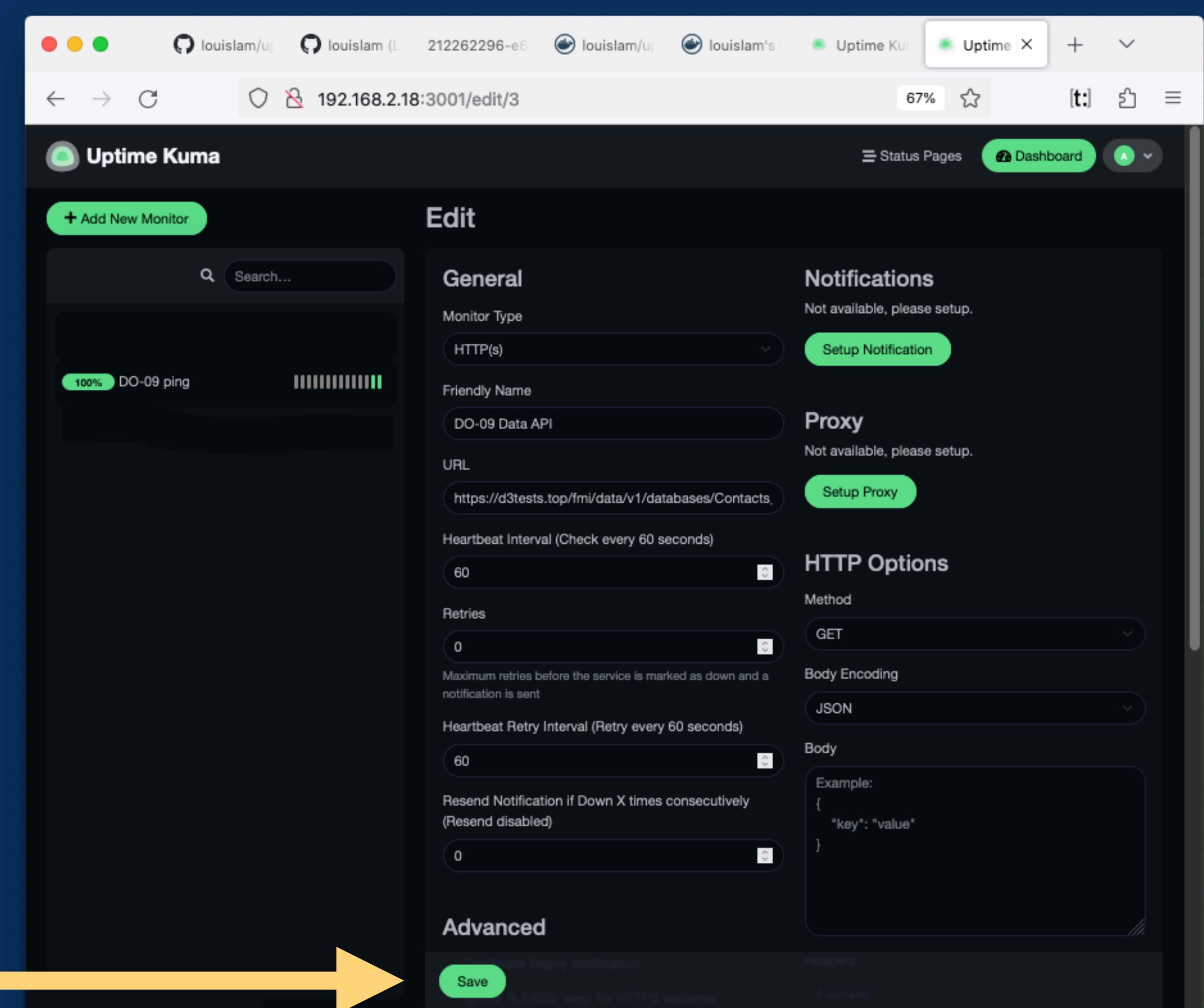
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping
 - API



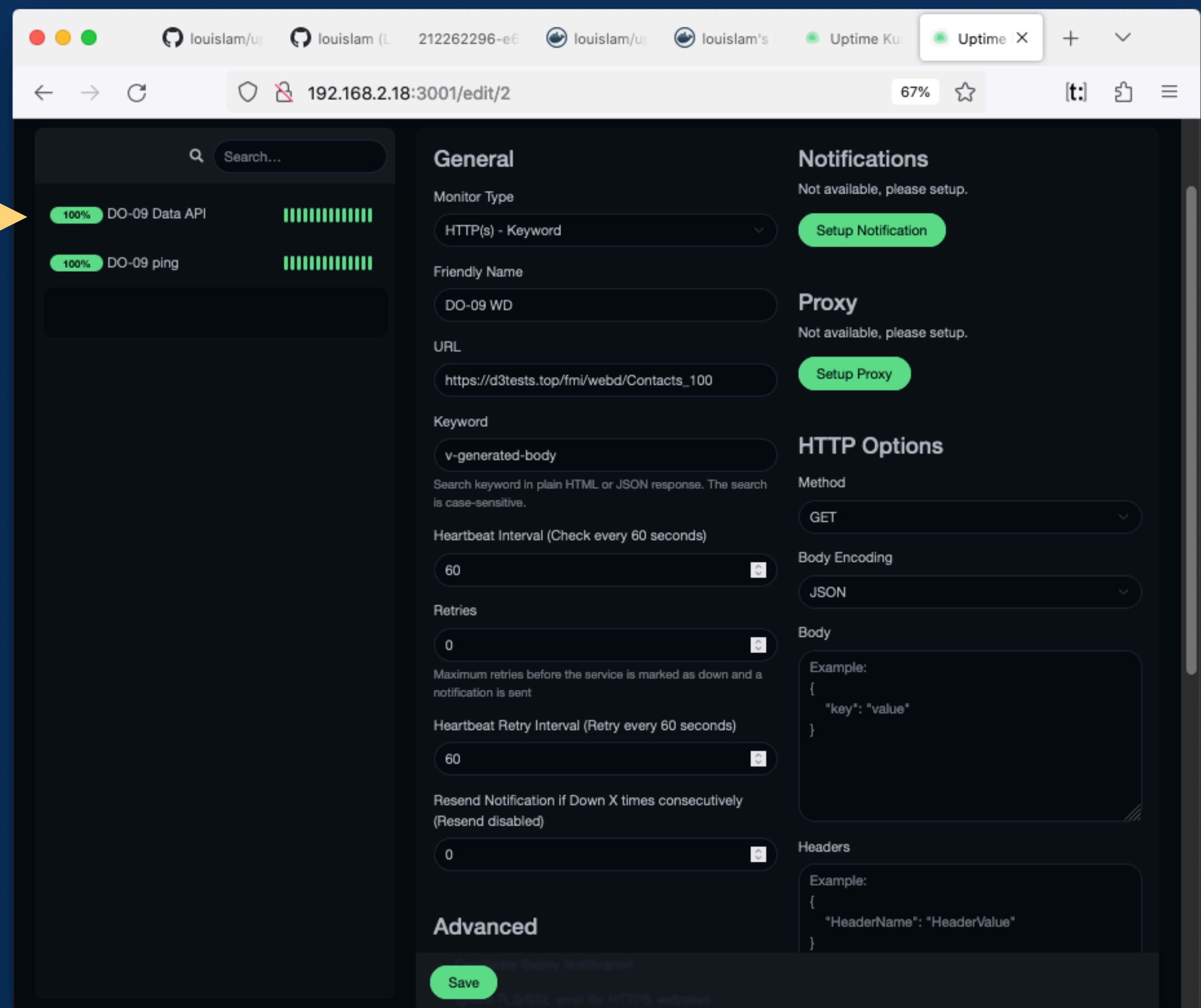
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping
 - API



Setup...

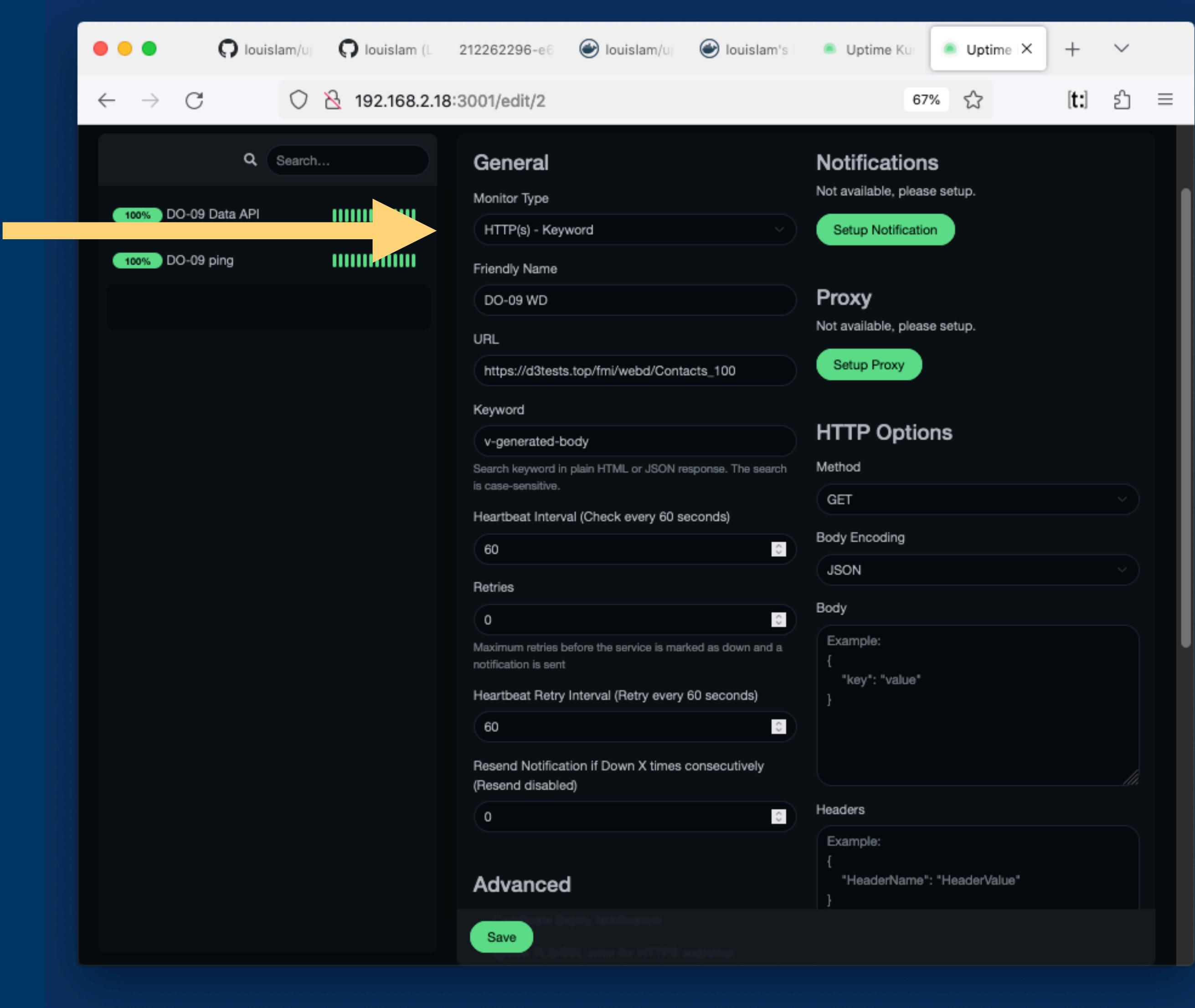
- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping
 - API
 - WebDirect



The screenshot shows the Uptime Kuma configuration interface for a monitor named "DO-09 WD". The monitor type is set to "HTTP(s) - Keyword". The URL is "https://d3tests.top/fmi/webd/Contacts_100" and the keyword is "v-generated-body". The method is set to "GET". The heartbeat interval is 60 seconds, and the retries are 0. The advanced section includes an example for Headers: { "HeaderName": "HeaderValue" }.

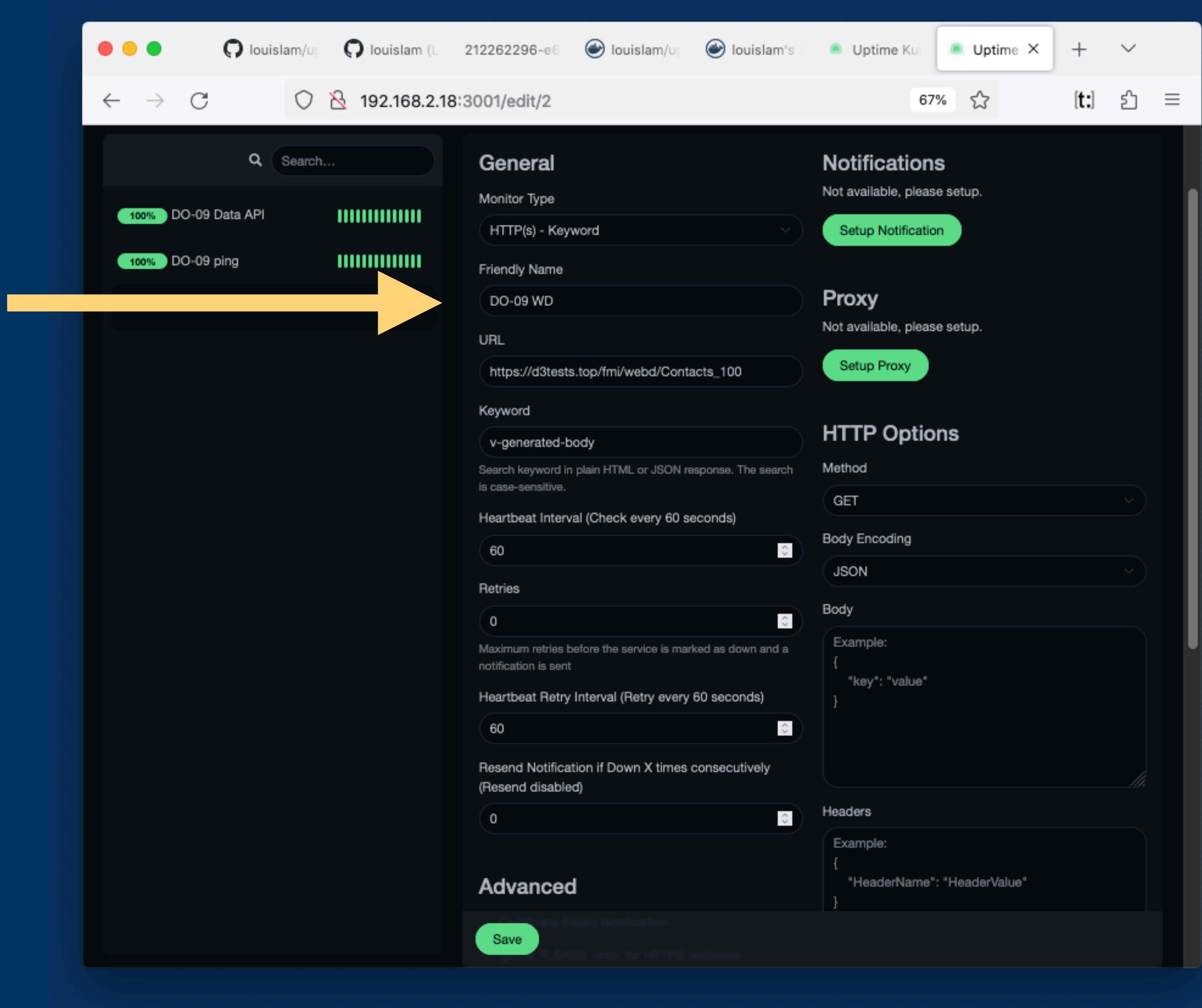
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping
 - API
 - WebDirect



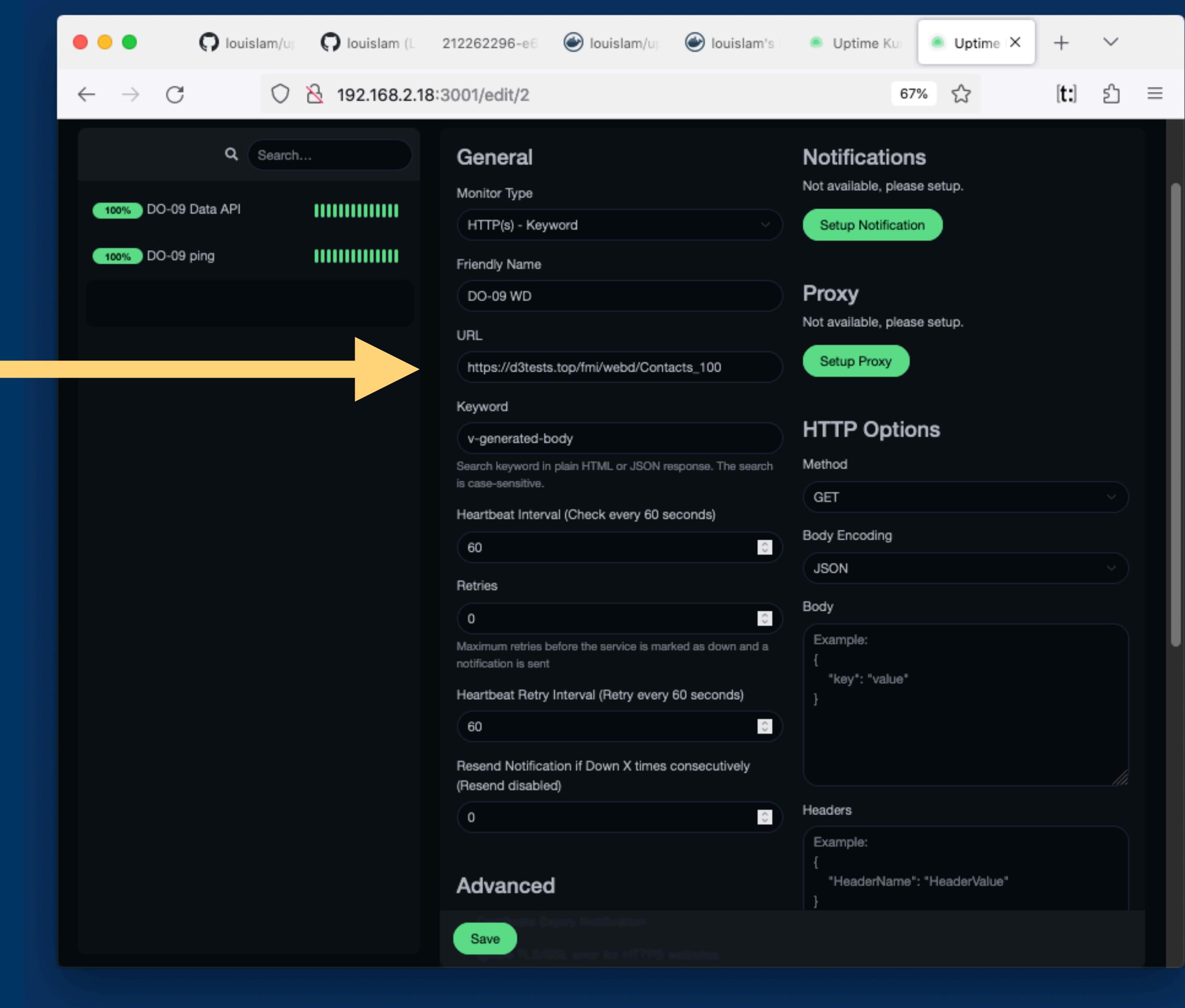
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping
 - API
 - WebDirect



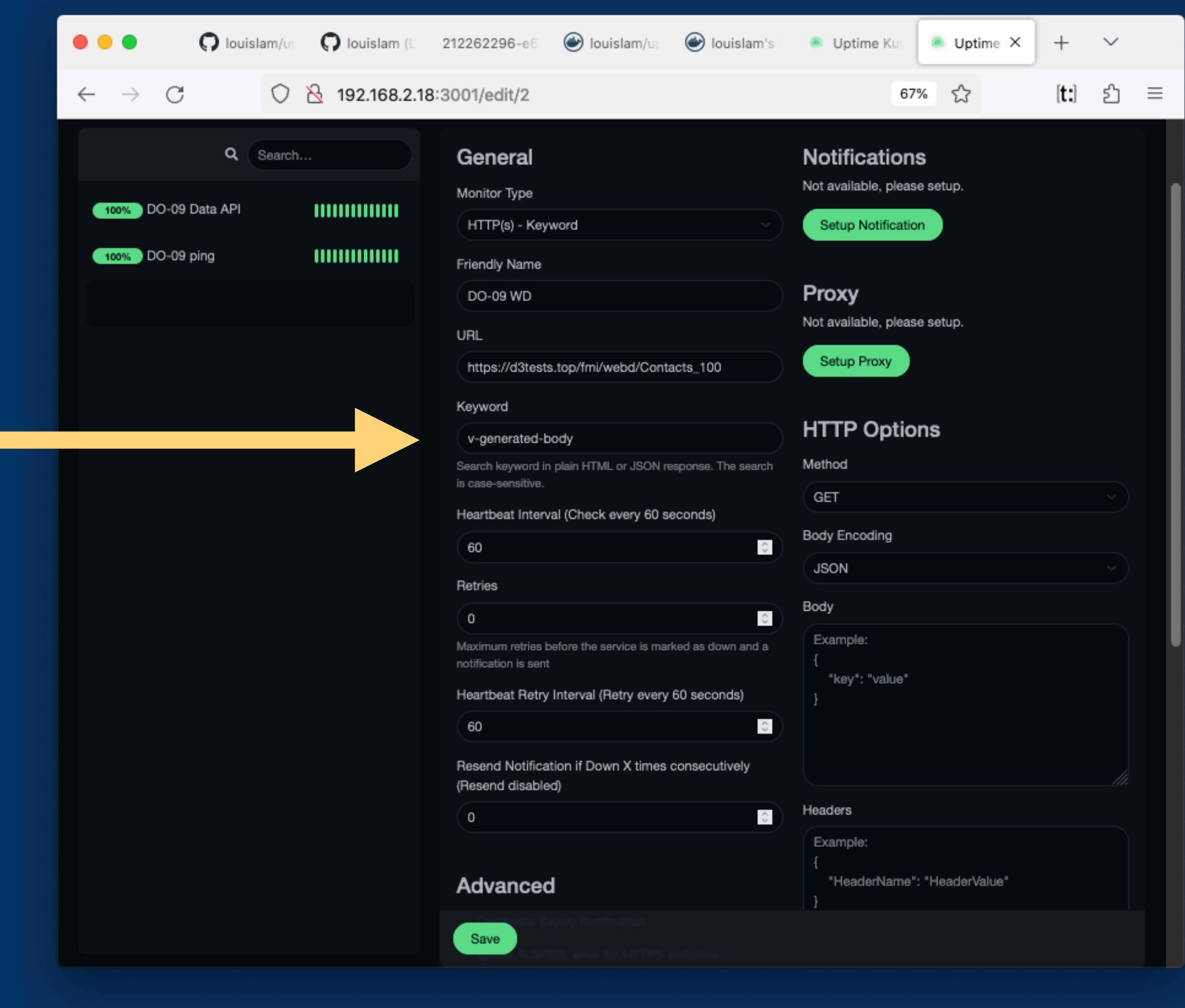
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping
 - API
 - WebDirect



Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping
 - API
 - WebDirect



Setup...

- check for Docker

- check for Docker

- prepare directory

- make Compose

- open in nano text

- run the .yml file

- open the web interface

- establish your connection

- change to night mode

- add new monitor

- ping

- API

- WebDirect

First Name	Last Name	Company
Ladarius	Abernathy	Grady, Reilly and Wilderman
Madonna	Beatty	Olson - Hyatt
Dashawn	Boyle	Ziemann - Kutch
Kiel	Bernhard	Mills - Walter
Anthony	Bashirian	Lind - Langworth

Setup...

- check for Docker

- check for Docker

- prepare directory

- make Compose

- open in nano terminal

- run the .yml file

- open the web interface

- establish your connection

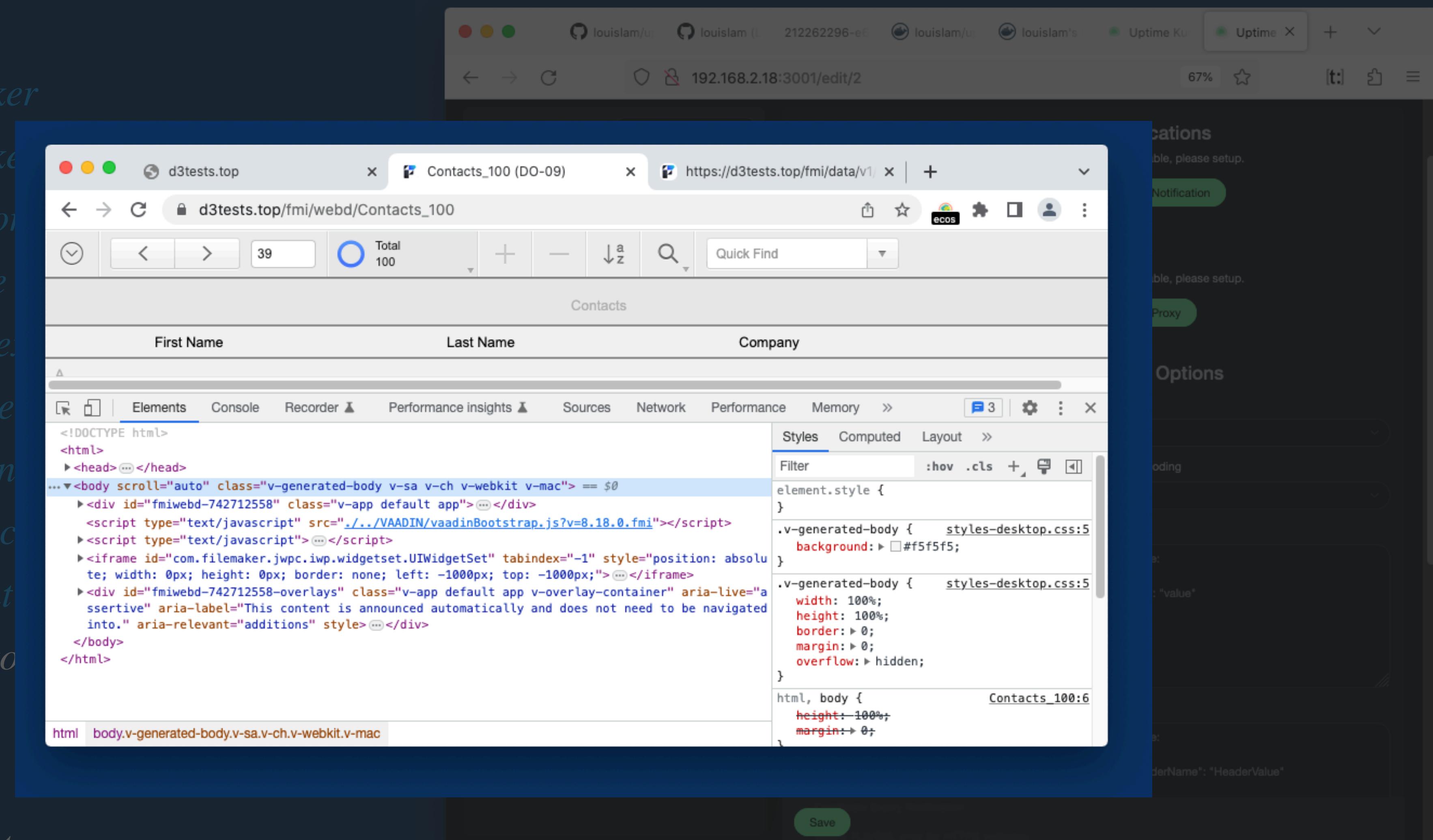
- change to night mode

- add new monitor

- ping

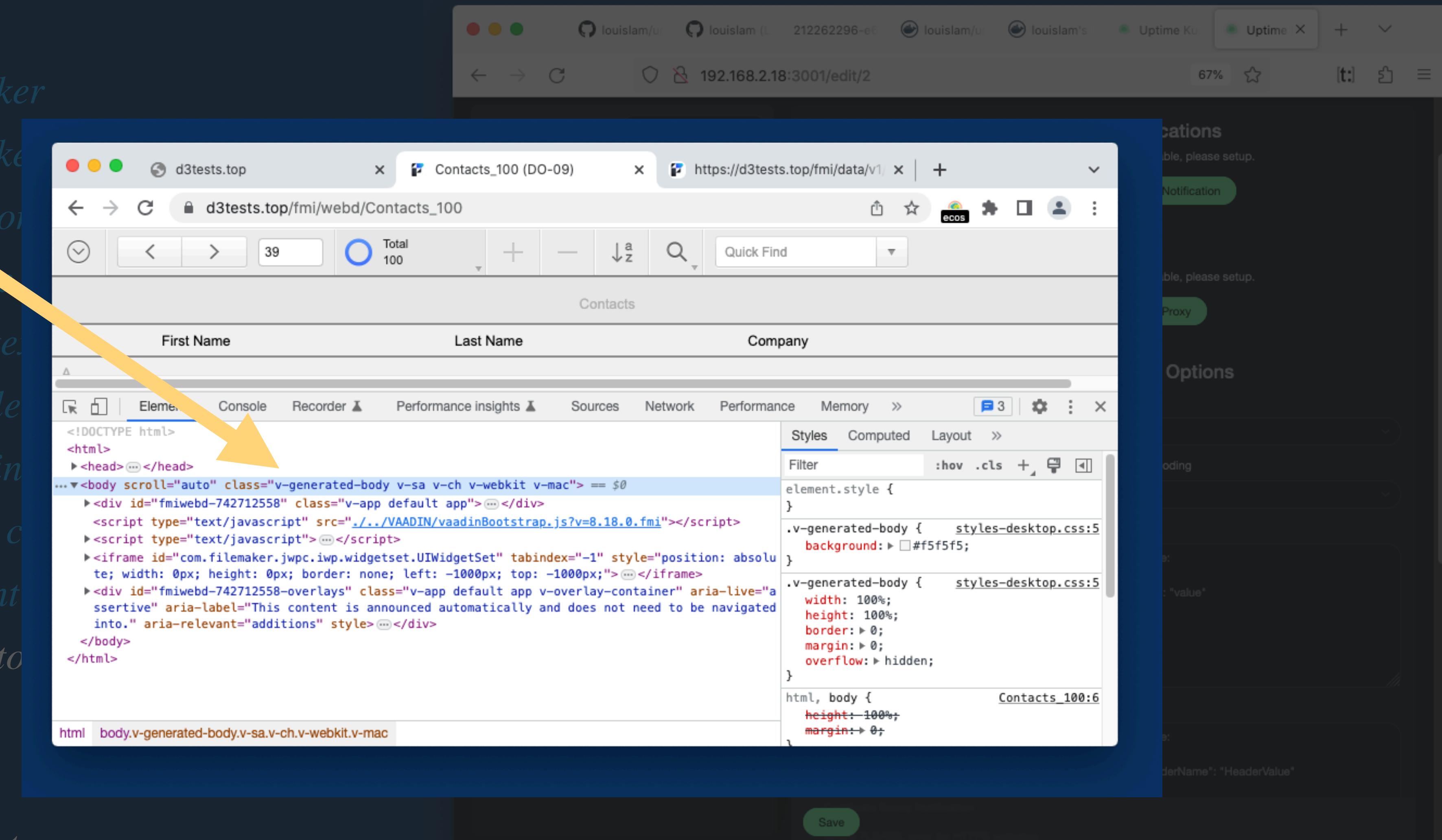
- API

- WebDirect



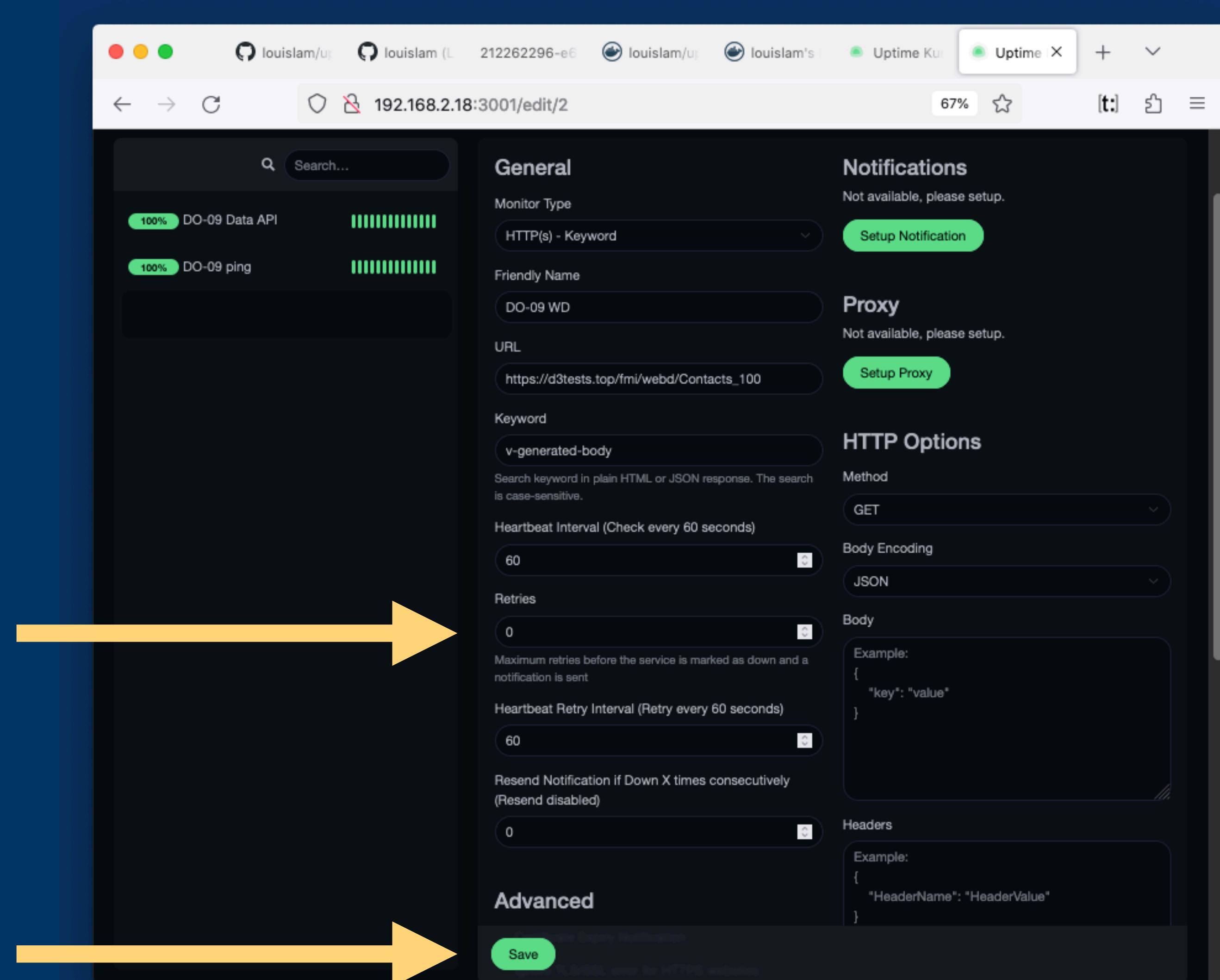
Setup...

- check for Docker
- check for Docker
- prepare directory
- make Compose
- open in nano terminal
- run the .yml file
- open the web interface
- establish your connection
- change to night mode
- add new monitor
- ping
- API
- WebDirect



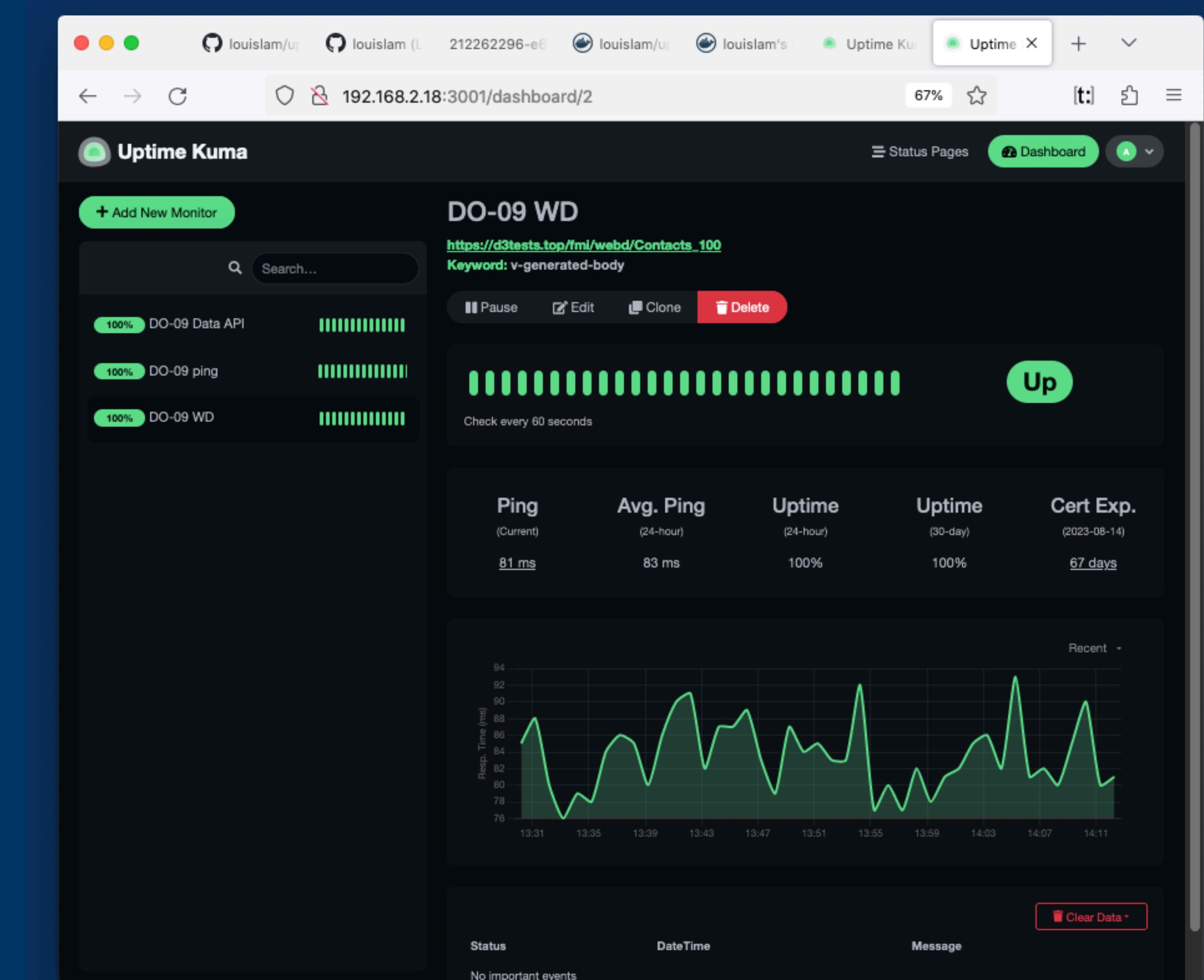
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
 - ping
 - API
 - WebDirect



Setup...

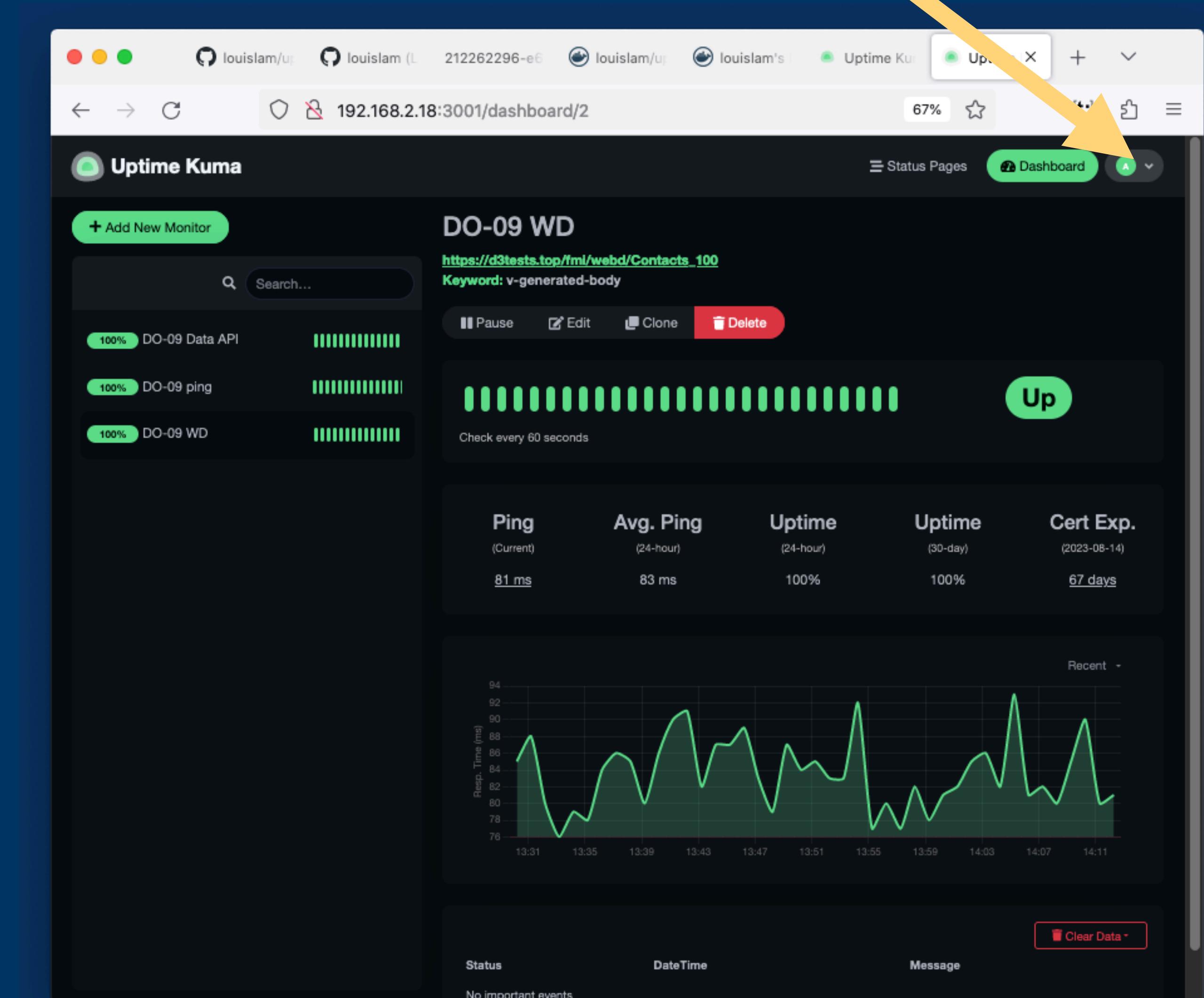
- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors



Settings

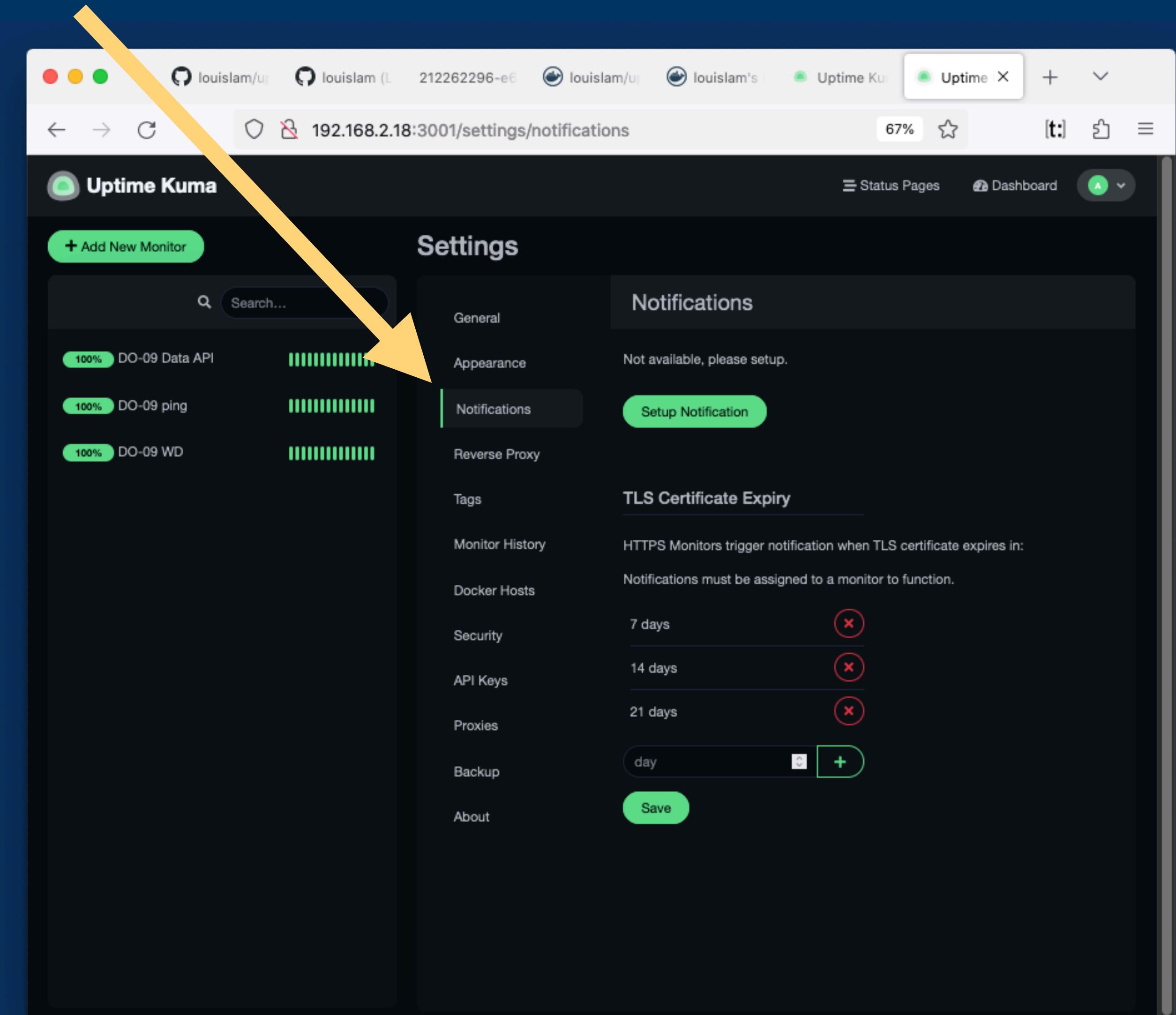
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
- set up notifications



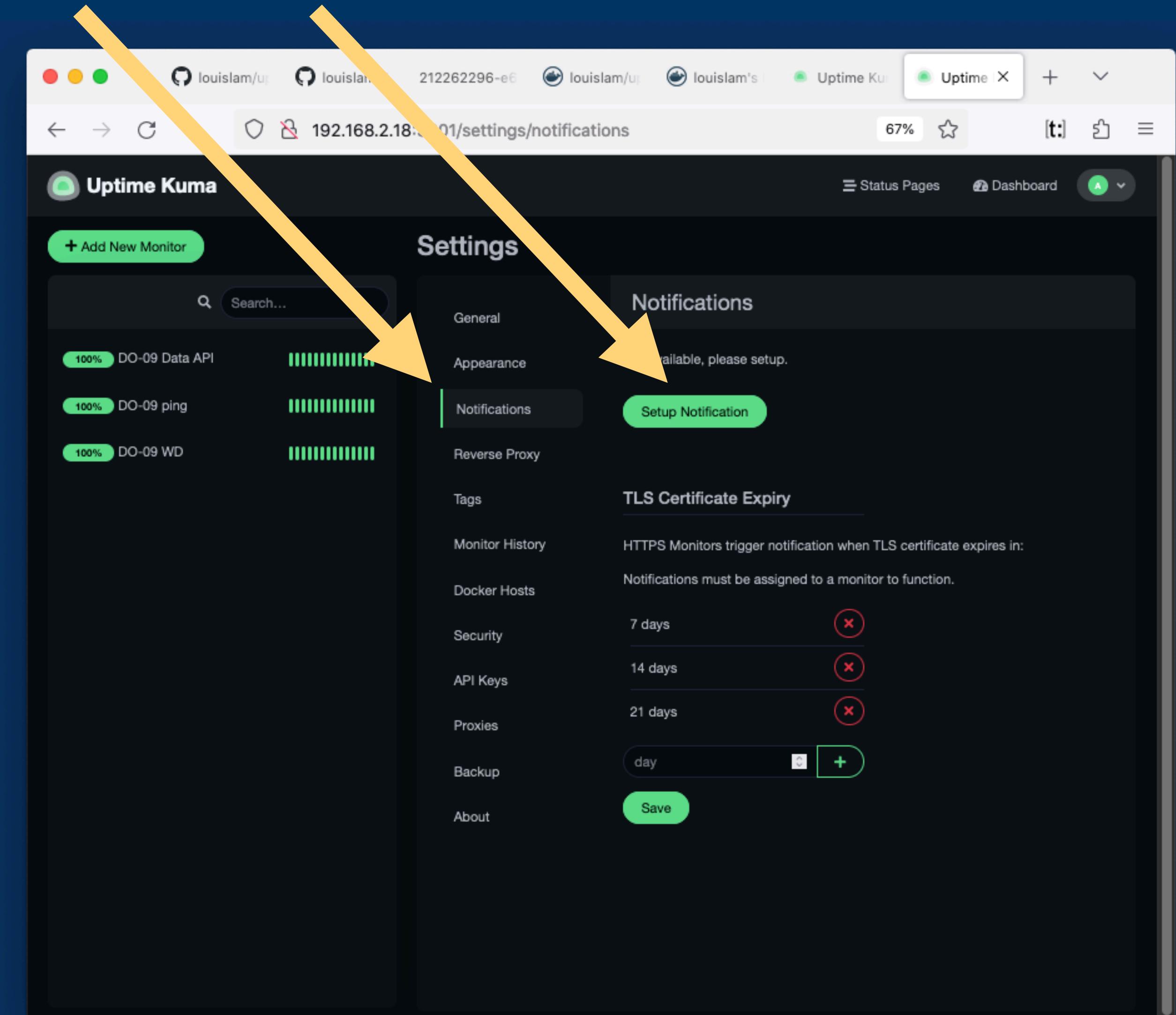
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
- set up notifications



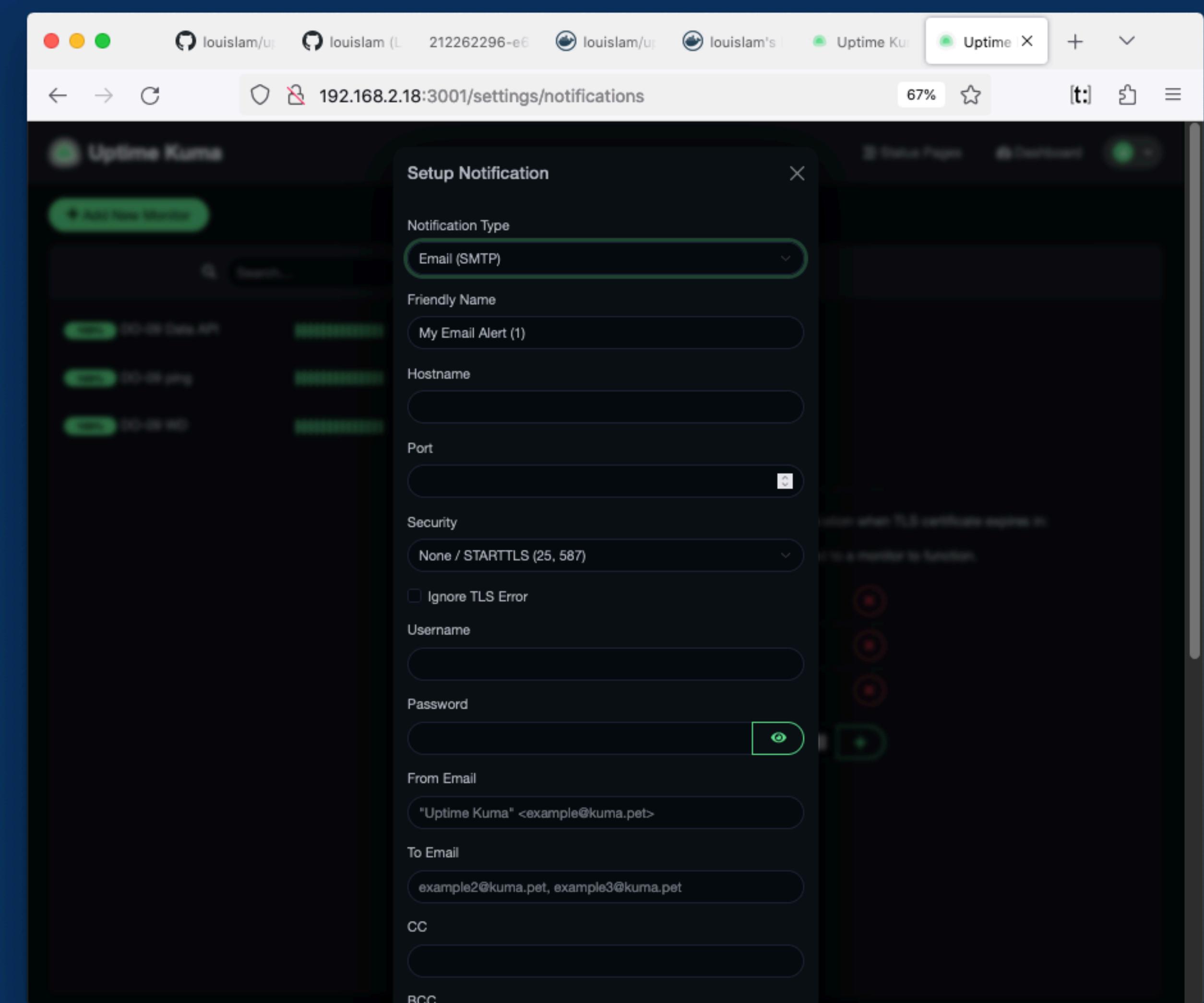
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
- set up notifications



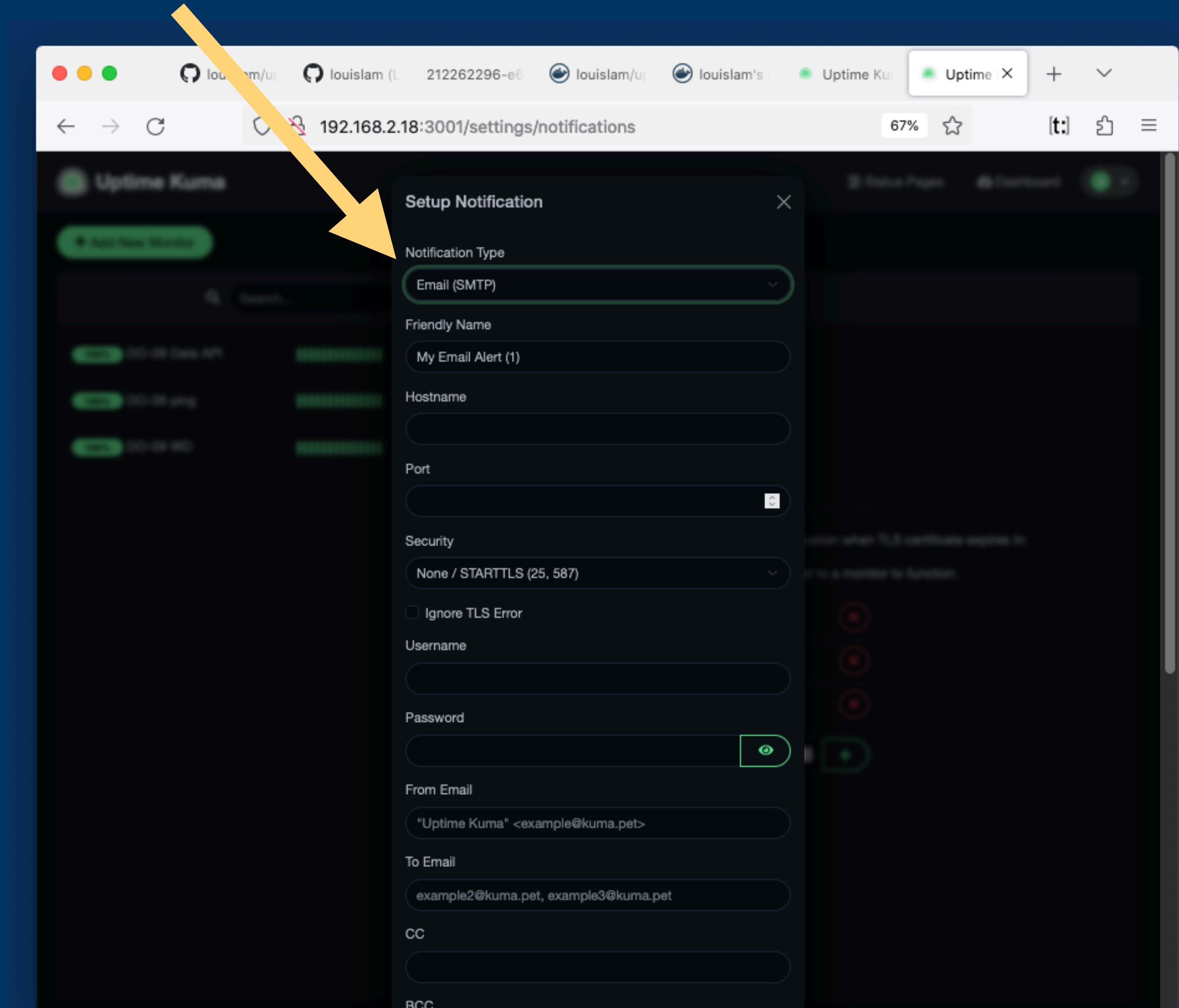
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
- set up notifications



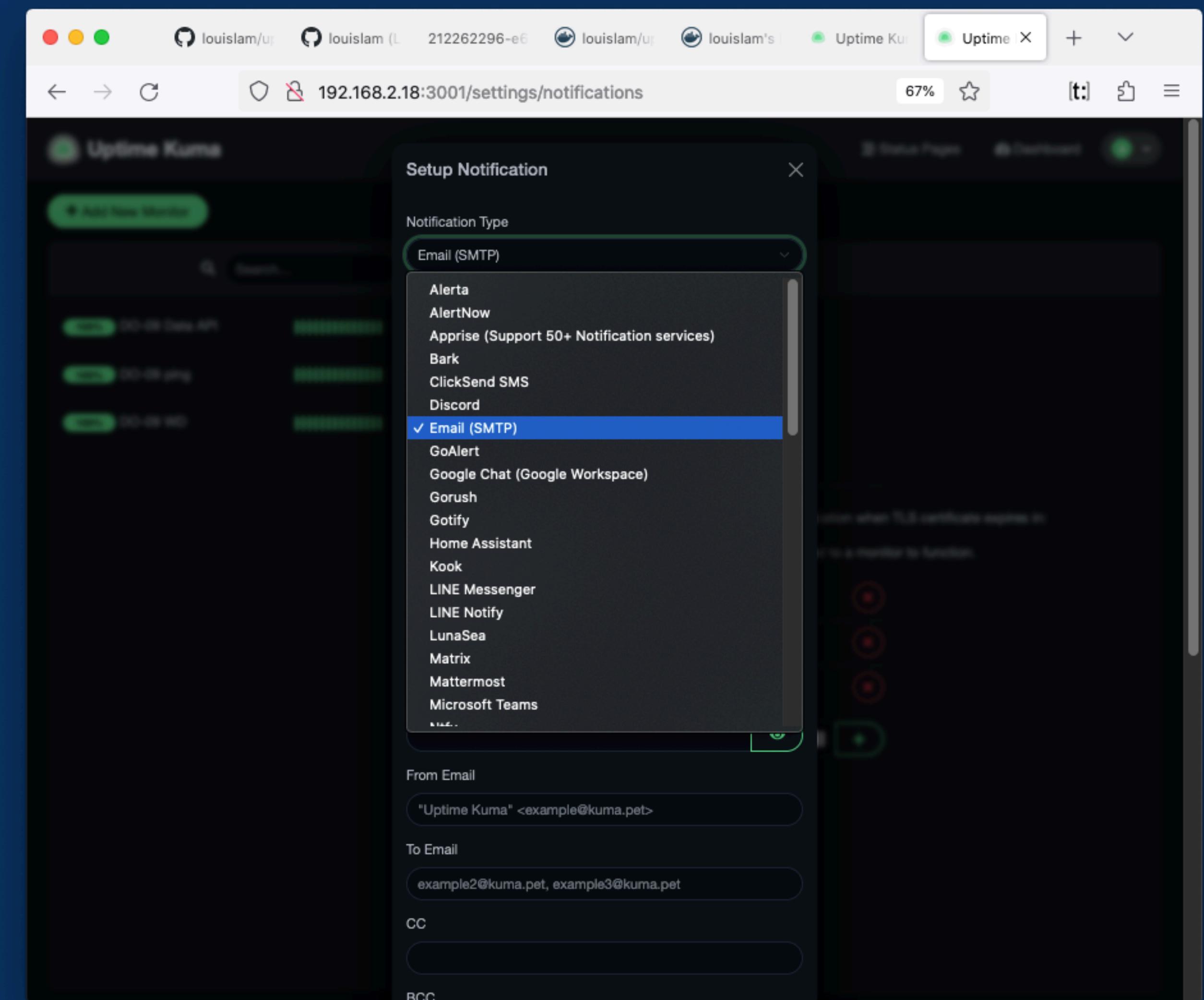
Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
- set up notifications



Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
- set up notifications



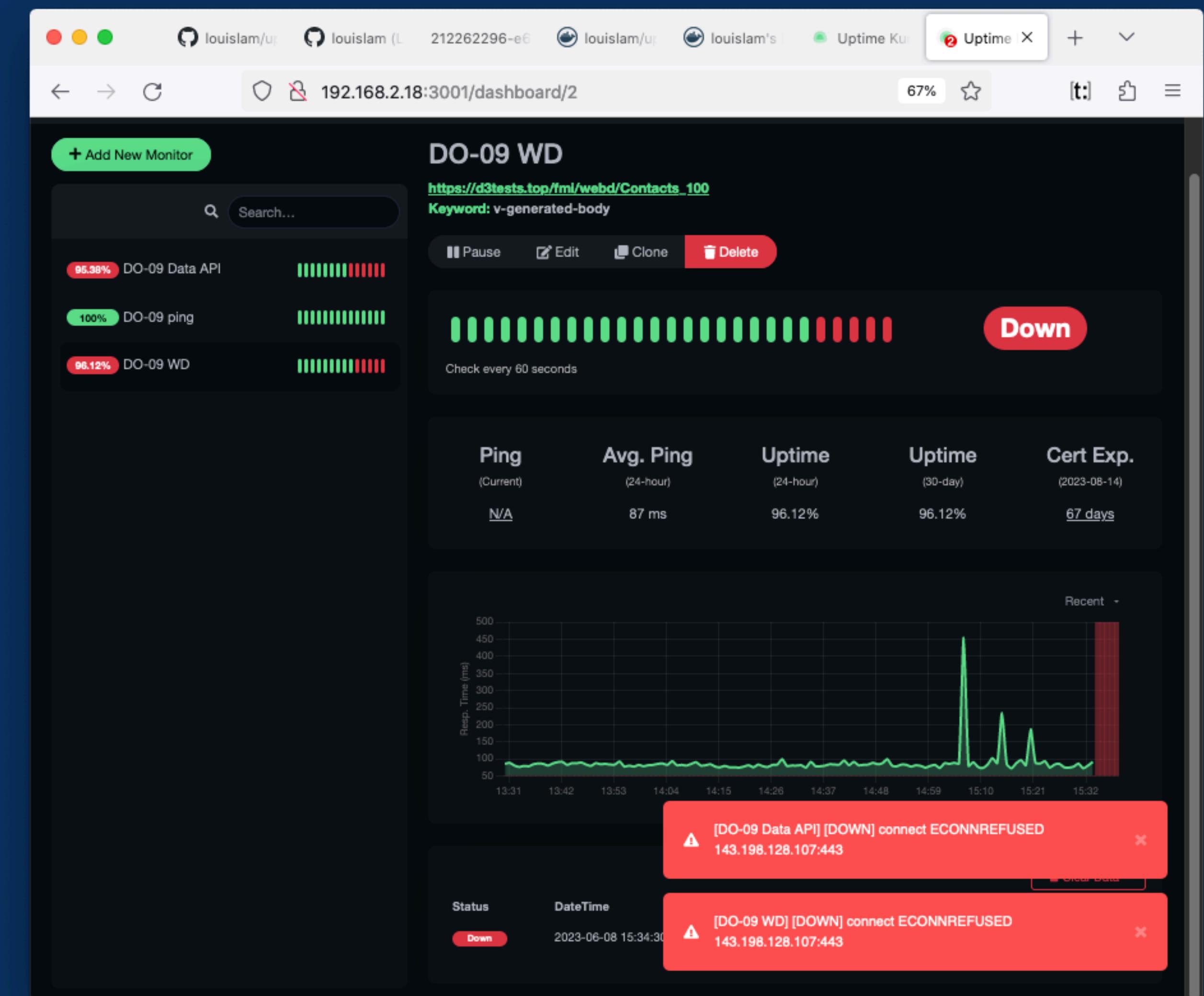

```
→ ~ sudo service fmshelper stop
```

Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
- set up notifications
- test

Setup...

- check for Docker
- check for Docker Compose
- prepare directories
- make Compose .yml file
- open in nano text editor
- run the .yml file
- open the web interface using browser
- establish your credentials
- change to night mode
- add new monitors
- set up notifications
- test





*“Next-Generation Mobile Apps and
Cross Browser Testing Cloud”*



*“The most advanced
load-testing platform
in the world.”*



Uptime Kuma

*“A fancy
self-hosted monitoring tool”*



Copy of this presentation and notes at my github...



d3.tools/gh



Copy of this presentation and notes at my github...

d3.tools/gh

