



# SYNTHETIC MONITORING AND E2E TESTING

## 2 SIDES OF THE SAME COIN

CARLY RICHMOND



# ABOUT ME

- Principal Developer Advocate & Manager @ Elastic
- Frontend Engineer, Speaker & Blogger

@CarlyLRichmond



carly-richmond



@carlyrichmond



carlyrichmond.com





# 2011

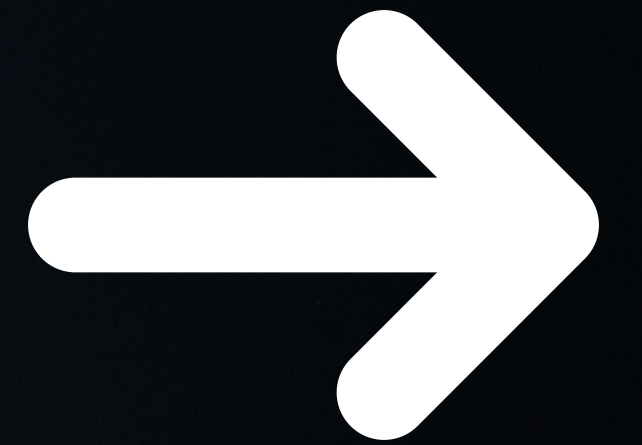








TESTERS



DEV

SUPPORT















DONT

^

WE HEAR YOU.

















E2E Testing

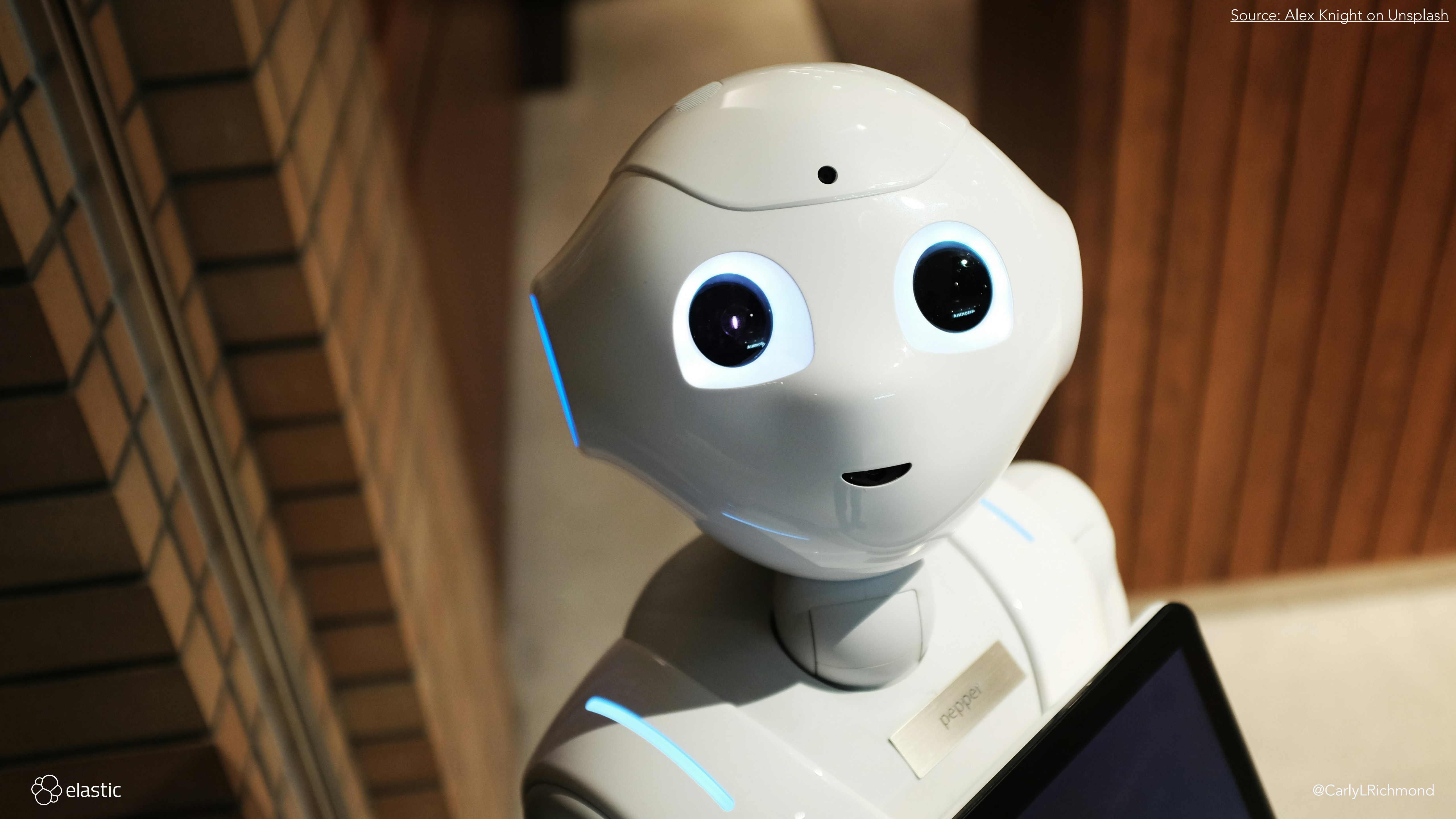


Synthetic Monitoring



# WASSER









## heartbeat.yml

```
1 heartbeat.monitors:  
2 - type: http  
3   name: Replicator HTTP ping  
4   id: synthetics-replicator-monitor-http  
5   enabled: true  
6   urls: "https://synthetics-replicator.netlify.app/"  
7   schedule: '@every 3m'  
8   timeout: 16s
```



# USER



# MONITORING AS CODE



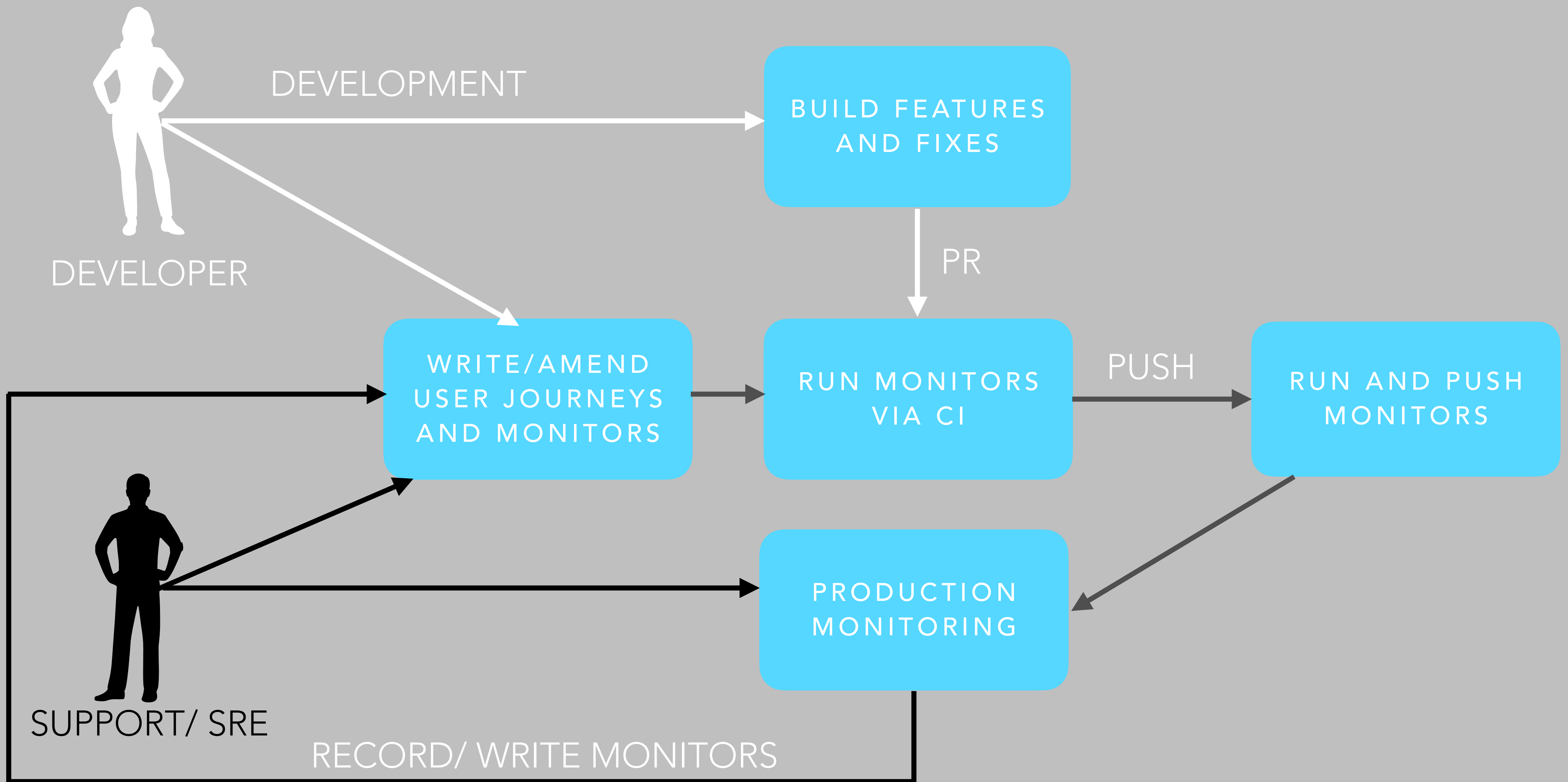




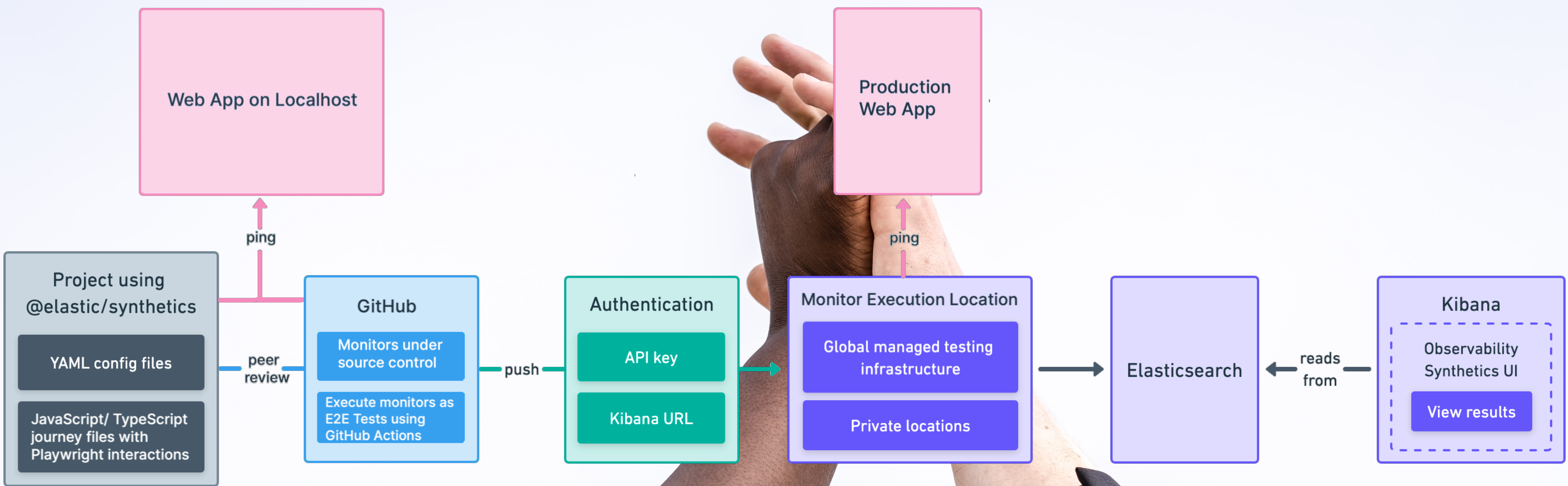
CLICK ME!

@CarlyLRichmond













Login

To seek out new food and new cuisines.  
To boldly eat what no one has eaten before!

Order Now!





```
1 import { test, expect } from '@playwright/test';
2
3 test('assert manual login successful', async ({ page }) => {
4   await page.goto('/login');
5
6   const loginForm = await page.getByTestId('login-form');
7   expect(loginForm).toBeDefined();
8
9   // Check submit button is disabled before entry
10  const submitButton = await page.getByTestId('login-button');
11  expect(submitButton).toBeDisabled();
12
13  // Add credentials
14  const username = 'me@me.com';
15  const usernameInput = page.getByTestId('username-input');
16  usernameInput.fill(username)
17  await expect(usernameInput).toHaveValue(username);
18
19  const password = 'password';
20  const passwordInput = page.getByTestId('password-input');
21  passwordInput.fill(password)
22  await expect(passwordInput).toHaveValue(password);
23
24  // Check login button is now enabled
25  expect(submitButton).toBeEnabled();
26  await submitButton.click();
27
28  // Wait for login
29  await page.waitForURL('**/order');
30
31  // Validate we have logged in successfully
32  const loginButton = await page.getByTestId('account-button')
33  expect(loginButton).toContainText('Welcome');
34 });
```







synthetics-replicator - -zsh

```
> npm install -g @elastic/synthetics  
> npx @elastic/synthetics -h  
> npx @elastic/synthetics init synthetics-replicator-  
test
```



synthetics-replicator-tests

Name
└─ journeys
└─ advanced-example-helpers.ts
└─ advanced-example.journey.ts
└─ example.journey.ts
└─ lightweight
└─ heartbeat.yml
└─ node_modules
└─ package-lock.json
└─ package.json
└─ README.md
└─ synthetics.config.ts



```
synthetics.config.ts

1 import { SyntheticsConfig } from '@elastic/synthetics';
2 import { devices } from "playwright-chromium";
3
4 export default env => {
5   const config: SyntheticsConfig = {
6     params: {
7       url: 'http://localhost:5173',
8       username: process.env.REPLICATR_USERNAME,
9       password: process.env.REPLICATR_PASSWORD,
10      isLocal: env !== 'production'
11    },
12    playwrightOptions: {
13      ignoreHTTPSErrors: false,
14      ...devices['Desktop Chrome']
15    },
16    /**
17     * Configure global monitor settings
18     */
19    monitor: {
20      schedule: 10,
21      locations: ['united_kingdom'],
22      privateLocations: []
23    },
24    /**
25     * Project monitors settings
26     */
27    project: {
28      id: 'synthetics-replicator-tests',
29      url: 'https://my-elastic-deployment:443',
30      space: 'default',
31    },
32  };
33
34  if (env === 'production') {
35    config.params!.url = 'https://synthetics-replicator.netlify.app/';
36  }
37  return config;
38 };
```





```
1 import { journey, step, monitor, expect, before } from '@elastic/synthetics';
2
3 journey('Replicator Login Journey', ({ page, params }) => {
4
5   // Only relevant for the push command to create
6   // monitors in Kibana
7   monitor.use({
8     id: 'synthetics-replicator-monitor-login',
9     schedule: 20,
10  });
11
12  before(async () => {
13    await page.goto(params.url);
14  });
15
16  step('assert home page loads', async () => {
17    const header = await page.locator('h1');
18    expect(await header.textContent()).toBe('Replicatr');
19  });
20
21  step('assert go to login page', async () => {
22    const loginButton = await page.getByTestId('account-button');
23    expect(loginButton).toHaveText('Login');
24    await loginButton.click();
25
26    const url = page.url();
27    expect(url).toContain('/login');
28
29    const loginForm = await page.getByTestId('login-form');
30    expect(loginForm).toBeDefined();
31  });
32
33  step('assert manual login successful', async () => {
34    const loginForm = await page.getByTestId('login-form');
35    expect(loginForm).toBeDefined();
36
37    // Check submit button is disabled before entry
38    const submitButton = await page.getByTestId('login-button');
39    expect(submitButton).toBeDisabled();
40
41    // Add credentials
42    const username = params.username;
43    const usernameInput = page.getByTestId('username-input');
44    usernameInput.fill(username);
45    await expect(usernameInput).toHaveValue(username);
46
47    const password = params.password;
48    const passwordInput = page.getByTestId('password-input');
49    passwordInput.fill(password);
50    await expect(passwordInput).toHaveValue(password);
51
52    // Check login button is now enabled
53    expect(submitButton).toBeEnabled();
54    await submitButton.click();
55
56    // Wait for login
57    await page.waitForURL('**/order');
58
59    // Validate we have logged in successfully
60    const loginButton = await page.getByTestId('account-button');
61    expect(loginButton).toContainText('Welcome');
62
63    // Ensure we have navigated to menu
64    const menuTiles = await page.getByTestId('menu-item-card');
65    expect(await menuTiles.count()).toBeGreaterThan(0);
66  });
67 });
```







```
1 import { journey, step, monitor, expect, before } from '@elastic/synthetics';
2
3 journey('Replicator Login Journey', ({ page, params }) => {
4
5   // Only relevant for the push command to create
6   // monitors in Kibana
7   monitor.use({
8     id: 'synthetics-replicator-monitor-login',
9     schedule: 20,
10  });
11
12  before(async () => {
13    await page.goto(params.url);
14  });
15
16  step('assert home page loads', async () => {
17    const header = await page.locator('h1');
18    expect(await header.textContent()).toBe('Replicatr');
19  });
20
```



```
21 step('assert go to login page', async () => {
22     const loginButton = await page.getByTestId('account-button');
23     expect(loginButton).toHaveText('Login');
24     await loginButton.click();
25
26     const url = page.url();
27     expect(url).toContain('/login');
28
29     const loginForm = await page.getByTestId('login-form');
30     expect(loginForm).toBeDefined();
31 });
32
33 step('assert manual login successful', async () => {
34     const loginForm = await page.getByTestId('login-form');
35     expect(loginForm).toBeDefined();
36
37     // Check submit button is disabled before entry
38     const submitButton = await page.getByTestId('login-button');
39     expect(submitButton).toBeDisabled();
40
41     // Add credentials
42     const username = params.username;
```



```
41 // Add credentials
42 const username = params.username;
43 const usernameInput = page.getByTestId( 'username-input' );
44 usernameInput.fill(username);
45 await expect(usernameInput).toHaveValue(username);
46
47 const password = params.password;
48 const passwordInput = page.getByTestId( 'password-input' );
49 passwordInput.fill(password);
50 await expect(passwordInput).toHaveValue(password);
51
52 // Check login button is now enabled
53 expect(submitButton).toBeEnabled();
54 await submitButton.click();
55
56 // Wait for login
57 await page.waitForURL( '**/order' );
58
59 // Validate we have logged in successfully
60 const loginButton = await page.getByTestId( 'account-button' );
61 expect(loginButton).toContainText( 'Welcome' );
62
```



```
47     const password = params.password;
48     const passwordInput = page.getByTestId('password-input');
49     passwordInput.fill(password);
50     await expect(passwordInput).toHaveValue(password);
51
52     // Check login button is now enabled
53     expect(submitButton).toBeEnabled();
54     await submitButton.click();
55
56     // Wait for login
57     await page.waitForURL('**/order');
58
59     // Validate we have logged in successfully
60     const loginButton = await page.getByTestId('account-button');
61     expect(loginButton).toContainText('Welcome');
62
63     // Ensure we have navigated to menu
64     const menuTiles = await page.getByTestId('menu-item-card');
65     expect(await menuTiles.count()).toBeGreaterThan(0);
66   });
67 });
```



```
1 import { journey, step, monitor, expect, before } from '@elastic/synthetics';
2
3 journey('Replicator Login Journey', ({ page, params }) => {
4
5   // Only relevant for the push command to create
6   // monitors in Kibana
7   monitor.use({
8     id: 'synthetics-replicator-monitor-login',
9     schedule: 20,
10  });
11
12  before(async () => {
13    await page.goto(params.url);
14  });
15
16  step('assert home page loads', async () => {
17    const header = await page.locator('h1');
18    expect(await header.textContent()).toBe('Replicatr');
19  });
20
21  step('assert go to login page', async () => {
22    const loginButton = await page.getByTestId('account-button');
23    expect(loginButton).toHaveText('Login');
24    await loginButton.click();
25
26    const url = page.url();
27    expect(url).toContain('/login');
28
29    const loginForm = await page.getByTestId('login-form');
30    expect(loginForm).toBeDefined();
31  });
32
33  step('assert manual login successful', async () => {
34    const loginForm = await page.getByTestId('login-form');
35    expect(loginForm).toBeDefined();
36
37    // Check submit button is disabled before entry
38    const submitButton = await page.getByTestId('login-button');
39    expect(submitButton).toBeDisabled();
40
41    // Add credentials
42    const username = params.username;
43    const usernameInput = page.getByTestId('username-input');
44    usernameInput.fill(username);
45    await expect(usernameInput).toHaveValue(username);
46
47    const password = params.password;
48    const passwordInput = page.getByTestId('password-input');
49    passwordInput.fill(password);
50    await expect(passwordInput).toHaveValue(password);
51
52    // Check login button is now enabled
53    expect(submitButton).toBeEnabled();
54    await submitButton.click();
55
56    // Wait for login
57    await page.waitForURL('**/order');
58
59    // Validate we have logged in successfully
60    const loginButton = await page.getByTestId('account-button');
61    expect(loginButton).toContainText('Welcome');
62
63    // Ensure we have navigated to menu
64    const menuTiles = await page.getByTestId('menu-item-card');
65    expect(await menuTiles.count()).toBeGreaterThan(0);
66  });
67 });
```





```
1 import { journey, step, monitor, expect } from '@elastic/synthetics';
2
3 journey('My Example Token-based Authentication Journey', ({ page, params, request }) => {
4
5   // Only relevant for the push command to create
6   // monitors in Kibana
7   monitor.use({
8     id: 'example-token-based-monitor',
9     schedule: 60,
10    enabled: params.isLocal
11  });
12
13  // We will populate these variable from example HTTP API responses
14  let apiKey;
15
16  // Lets simulate getting an authentication token back from an auth service
17  const apiBaseUrl = params.example_oauth2_endpoint;
18  const clientId = params.example_client_id;
19
20  // Let's say we expect the auth API to return a token
21  const exampleAccessTokenKey = 'access_token';
22  const exampleAccessToken = params.example_access_token;
23
24  // Get headers for use by the page
25  step('Get token from auth service', async() => {
26    const resp = await request.get(`${apiBaseUrl}?client_id=${clientId}&${exampleAccessTokenKey}=${exampleAccessToken}`);
27    apiKey = resp.headers()[exampleAccessTokenKey];
28    expect(apiKey).toEqual(exampleAccessToken);
29  });
30
31  // Use headers in other page access
32  step('Use API Key in page request', async() => {
33    await page.setExtraHTTPHeaders({ [exampleAccessTokenKey]: apiKey });
34
35    await page.goto(`${params.redirectUrl} ${exampleAccessTokenKey}=${apiKey}`);
36    await page.waitForSelector(`text=${apiKey}`);
37  });
38 });
```





```
synthetics-replicator — -zsh — 129x37
[carly.richmond@MacBookPro synthetics-replicator % npm run test

> synthetics-replicator@0.0.0 test
> cd apps/synthetics-replicator-tests && npm run test

> synthetics-replicator-tests@1.0.0 test
> npx @elastic/synthetics journeys

Journey: Local Replicator Order Journey
  ✓ Step: 'assert home page loads' succeeded (65 ms)
  ✓ Step: 'assert move to order page' succeeded (268 ms)
  ✓ Step: 'assert adding to order' succeeded (232 ms)

Journey: Recorded Order journey
  ✓ Step: 'Go to order items page' succeeded (383 ms)
  ✓ Step: 'Add item to cart successfully' succeeded (83 ms)
  ✓ Step: 'Add 2nd item to cart successfully' succeeded (66 ms)
  ✓ Step: 'Add 3rd item to cart successfully' succeeded (67 ms)

Journey: Replicator Order Journey
  ✓ Step: 'assert home page loads' succeeded (48 ms)
  ✓ Step: 'assert move to order page' succeeded (580 ms)
  ✓ Step: 'assert adding to order' succeeded (597 ms)

Journey: Replicator Login Journey
  ✓ Step: 'assert home page loads' succeeded (71 ms)
  ✓ Step: 'assert go to login page' succeeded (141 ms)
  ✓ Step: 'assert manual login successful' succeeded (633 ms)
  ✓ Step: 'assert unable to go to login page when signed in' succeeded (197 ms)

14 passed (5853 ms)

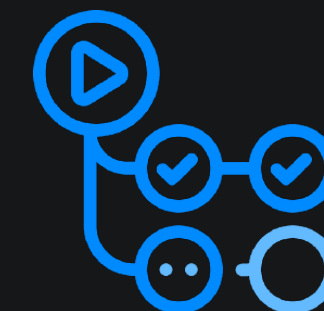
carly.richmond@MacBookPro synthetics-replicator %
```





```
push-workflow.yml

1 name: Push Workflow- Elastic Synthetics With Svelte Replicator
2 on:
3   push:
4     branches: [main, master]
5 jobs:
6   test:
7     env:
8     NODE_ENV: development
9     REPLICATR_USERNAME: ${ secrets.REPLICATR_USERNAME }
10    REPLICATR_PASSWORD: ${ secrets.REPLICATR_PASSWORD }
11    SYNTHETICS_JUNIT_FILE: junit-synthetics.xml
12 runs-on: ubuntu-latest
13 steps:
14   - uses: actions/checkout@v3
15   - uses: actions/setup-node@v3
16     with:
17       node-version: 18
18   - run: npm ci
19   - run: npm start &
20   - run: "npm install @elastic/synthetics && npx @elastic/synthetics . --reporter=junit"
21     working-directory: ./apps/synthetics-replicator-tests/journeys
22   - name: Publish Unit Test Results
23     uses: EnricoMi/publish-unit-test-result-action@v2
24     if: always()
25     with:
26       junit_files: '**/junit-*.xml'
27       check_name: Elastic Synthetics Tests
28 push:
29   env:
30     NODE_ENV: production
31     REPLICATR_USERNAME: ${ secrets.REPLICATR_USERNAME }
32     REPLICATR_PASSWORD: ${ secrets.REPLICATR_PASSWORD }
33     SYNTHETICS_API_KEY: ${ secrets.SYNTHETICS_API_KEY }
34 needs: test
35 defaults:
36   run:
37     working-directory: ./apps/synthetics-replicator-tests
38 runs-on: ubuntu-latest
39 steps:
40   - uses: actions/checkout@v3
41   - uses: actions/setup-node@v3
42     with:
43       node-version: 18
44   - run: npm ci
45   - run: npm run push
```



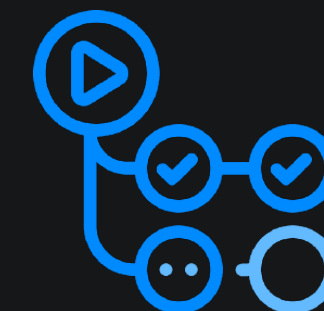


```
5 jobs:
6   test:
7     env:
8       NODE_ENV: development
9       REPLICATR_USERNAME: ${ secrets.REPLICATR_USERNAME }
10      REPLICATR_PASSWORD: ${ secrets.REPLICATR_PASSWORD }
11      SYNTHETICS_JUNIT_FILE: junit-synthetics.xml
12 runs-on: ubuntu-latest
13 steps:
14   - uses: actions/checkout@v3
15   - uses: actions/setup-node@v3
16     with:
17       node-version: 18
18   - run: npm ci
19   - run: npm start &
20   - run: "npm install @elastic/synthetics && npx @elastic/synthetics . --reporter=junit"
21     working-directory: ./apps/synthetics-replicator-tests/journeys
22   - name: Publish Unit Test Results
23     uses: EnricoMi/publish-unit-test-result-action@v2
24     if: always()
25     with:
26       junit_files: '**/junit-*.xml'
27       check_name: Elastic Synthetics Tests
28 push:
29   env:
30     NODE_ENV: production
```



```
push-workflow.yml

1 name: Push Workflow- Elastic Synthetics With Svelte Replicator
2 on:
3   push:
4     branches: [main, master]
5 jobs:
6   test:
7     env:
8     NODE_ENV: development
9     REPLICATR_USERNAME: ${ secrets.REPLICATR_USERNAME }
10    REPLICATR_PASSWORD: ${ secrets.REPLICATR_PASSWORD }
11    SYNTHETICS_JUNIT_FILE: junit-synthetics.xml
12 runs-on: ubuntu-latest
13 steps:
14   - uses: actions/checkout@v3
15   - uses: actions/setup-node@v3
16     with:
17       node-version: 18
18   - run: npm ci
19   - run: npm start &
20   - run: "npm install @elastic/synthetics && npx @elastic/synthetics . --reporter=junit"
21     working-directory: ./apps/synthetics-replicator-tests/journeys
22   - name: Publish Unit Test Results
23     uses: EnricoMi/publish-unit-test-result-action@v2
24     if: always()
25     with:
26       junit_files: '**/junit-*.xml'
27       check_name: Elastic Synthetics Tests
28 push:
29   env:
30   NODE_ENV: production
31   REPLICATR_USERNAME: ${ secrets.REPLICATR_USERNAME }
32   REPLICATR_PASSWORD: ${ secrets.REPLICATR_PASSWORD }
33   SYNTHETICS_API_KEY: ${ secrets.SYNTHETICS_API_KEY }
34 needs: test
35 defaults:
36   run:
37     working-directory: ./apps/synthetics-replicator-tests
38 runs-on: ubuntu-latest
39 steps:
40   - uses: actions/checkout@v3
41   - uses: actions/setup-node@v3
42     with:
43       node-version: 18
44   - run: npm ci
45   - run: npm run push
```







# Revert test command

main 73d0aff

Re-run all jobs

✓ Push Workflow- Elastic Syntheti... 1  
on: push

✓ test

✓ push

✓ Elastic Synthetics Tests

## GitHub Actions / Elastic Synthetics Tests

succeeded 2 minutes ago in 0s

### All 14 tests pass in 2s

14 tests	+14	14	+14	2s	+2s
4 suites	+ 4	0	± 0		
1 files	+ 1	0	± 0		

Results for commit [73d0aff](#) . ± Comparison against earlier commit [03d2e72](#) .

#### ANNOTATIONS

Check notice on line 0 in .github

**github-actions** / Elastic Synthetics Tests

#### 14 tests found

There are 14 tests, see "Raw output" for the full list of tests.

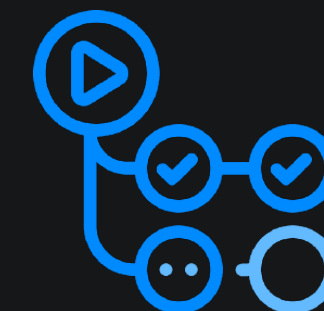
Raw output

```
Local Replicator Order Journey assert adding to order - assert adding to order
Local Replicator Order Journey assert home page loads - assert home page loads
Local Replicator Order Journey assert move to order page - assert move to order page
Recorded Order journey Add 2nd item to cart successfully - Add 2nd item to cart successfully
Recorded Order journey Add 3rd item to cart successfully - Add 3rd item to cart successfully
Recorded Order journey Add item to cart successfully - Add item to cart successfully
Recorded Order journey Go to order items page - Go to order items page
Replicator Login Journey assert go to login page - assert go to login page
Replicator Login Journey assert home page loads - assert home page loads
Replicator Login Journey assert manual login successful - assert manual login successful
Replicator Login Journey assert unable to go to login page when signed in - assert unable to go to login page when signed in
Replicator Order Journey assert adding to order - assert adding to order
Replicator Order Journey assert home page loads - assert home page loads
Replicator Order Journey assert move to order page - assert move to order page
```



```
push-workflow.yml

1 name: Push Workflow- Elastic Synthetics With Svelte Replicator
2 on:
3   push:
4     branches: [main, master]
5 jobs:
6   test:
7     env:
8     NODE_ENV: development
9     REPLICATR_USERNAME: ${ secrets.REPLICATR_USERNAME }
10    REPLICATR_PASSWORD: ${ secrets.REPLICATR_PASSWORD }
11    SYNTHETICS_JUNIT_FILE: junit-synthetics.xml
12 runs-on: ubuntu-latest
13 steps:
14   - uses: actions/checkout@v3
15   - uses: actions/setup-node@v3
16     with:
17       node-version: 18
18   - run: npm ci
19   - run: npm start &
20   - run: "npm install @elastic/synthetics && npx @elastic/synthetics . --reporter=junit"
21     working-directory: ./apps/synthetics-replicator-tests/journeys
22   - name: Publish Unit Test Results
23     uses: EnricoMi/publish-unit-test-result-action@v2
24     if: always()
25     with:
26       junit_files: '**/junit-*.xml'
27       check_name: Elastic Synthetics Tests
28 push:
29   env:
30     NODE_ENV: production
31     REPLICATR_USERNAME: ${ secrets.REPLICATR_USERNAME }
32     REPLICATR_PASSWORD: ${ secrets.REPLICATR_PASSWORD }
33     SYNTHETICS_API_KEY: ${ secrets.SYNTHETICS_API_KEY }
34 needs: test
35 defaults:
36   run:
37     working-directory: ./apps/synthetics-replicator-tests
38 runs-on: ubuntu-latest
39 steps:
40   - uses: actions/checkout@v3
41   - uses: actions/setup-node@v3
42     with:
43       node-version: 18
44   - run: npm ci
45   - run: npm run push
```



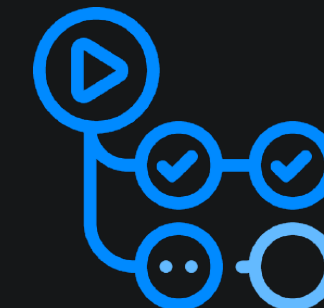


```
26     junit_files: '**/junit-*.xml'
27     check_name: Elastic Synthetics Tests
28   push:
29     env:
30       NODE_ENV: production
31       REPLICATR_USERNAME: ${ secrets.REPLICATR_USERNAME }
32       REPLICATR_PASSWORD: ${ secrets.REPLICATR_PASSWORD }
33       SYNTHETICS_API_KEY: ${ secrets.SYNTHETICS_API_KEY }
34     needs: test
35     defaults:
36       run:
37         working-directory: ./apps/synthetics-replicator-tests
38     runs-on: ubuntu-latest
39     steps:
40       - uses: actions/checkout@v3
41       - uses: actions/setup-node@v3
42         with:
43           node-version: 18
44       - run: npm ci
45       - run: npm run push
```



```
push-workflow.yml

1 name: Push Workflow- Elastic Synthetics With Svelte Replicator
2 on:
3   push:
4     branches: [main, master]
5 jobs:
6   test:
7     env:
8     NODE_ENV: development
9     REPLICATR_USERNAME: ${ secrets.REPLICATR_USERNAME }
10    REPLICATR_PASSWORD: ${ secrets.REPLICATR_PASSWORD }
11    SYNTHETICS_JUNIT_FILE: junit-synthetics.xml
12 runs-on: ubuntu-latest
13 steps:
14   - uses: actions/checkout@v3
15   - uses: actions/setup-node@v3
16     with:
17       node-version: 18
18   - run: npm ci
19   - run: npm start &
20   - run: "npm install @elastic/synthetics && npx @elastic/synthetics . --reporter=junit"
21     working-directory: ./apps/synthetics-replicator-tests/journeys
22   - name: Publish Unit Test Results
23     uses: EnricoMi/publish-unit-test-result-action@v2
24     if: always()
25     with:
26       junit_files: '**/junit-*.xml'
27       check_name: Elastic Synthetics Tests
28 push:
29   env:
30   NODE_ENV: production
31   REPLICATR_USERNAME: ${ secrets.REPLICATR_USERNAME }
32   REPLICATR_PASSWORD: ${ secrets.REPLICATR_PASSWORD }
33   SYNTHETICS_API_KEY: ${ secrets.SYNTHETICS_API_KEY }
34 needs: test
35 defaults:
36   run:
37     working-directory: ./apps/synthetics-replicator-tests
38 runs-on: ubuntu-latest
39 steps:
40   - uses: actions/checkout@v3
41   - uses: actions/setup-node@v3
42     with:
43       node-version: 18
44   - run: npm ci
45   - run: npm run push
```





**Observability**

- Overview
- Alerts
- SLOs
- Cases
- AI Assistant

- Logs
- Explorer **BETA**
- Stream
- Anomalies
- Categories

- Infrastructure**
- Inventory
- Metrics Explorer
- Hosts **BETA**
- APM**
- Services
- Traces
- Dependencies

**Synthetics**

# Monitors

[+ Create Monitor](#) [Refresh](#)

Overview Management

🔍 Search by name, URL, host, tag, project or l

Up Down Disabled

Type 2 ⌵ Location 1 ⌵ Tags 0 ⌵ Frequency 5 ⌵ Project 2 ⌵

**Current status**

**6** Up   **2** Down   **1** Disabled



Showing **9 Monitors** Sort by Status ⌵ Group by Project ⌵

synthetics-replicator-tests 5 Monitors

**Local Replicator Order Journey**  
Europe - United Kingdom

Duration ? **3 s**

**Recorded Order journey**  
Europe - United Kingdom

Duration ? **2 s**

**Replicator HTTP ping**  
Europe - United Kingdom

Duration ? **64 ms**

**Replicator Login Journey** ⚠️  
Europe - United Kingdom

Duration ? **2 s**

**Replicator Order Journey**  
Europe - United Kingdom



- Observability
- Overview
- Alerts
- SLOs
- Cases
- AI Assistant
- Logs
- Explorer BETA
- Stream
- Anomalies
- Categories

- Infrastructure
- Inventory
- Metrics Explorer
- Hosts BETA
- APM
- Services
- Traces
- Dependencies

- Synthetics
- Monitors**
- TLS Certificates
- User Experience

# Monitors

Overview Management

Search by name, URL, host, tag, project or l Up Down Disabled

**Current status**

**6** Up   **2** Down   **1** Disabled

**Last 6 hours**

**3** Errors

Showing 9 Monitors

synthetics-replicator-tests

**Local Replicator Order Journey**

Europe - United Kingdom

Duration ? **3 s**

**Recorded Order journey**

Europe - United Kingdom

Duration ? **2 s**

**Replicator Order Journey**

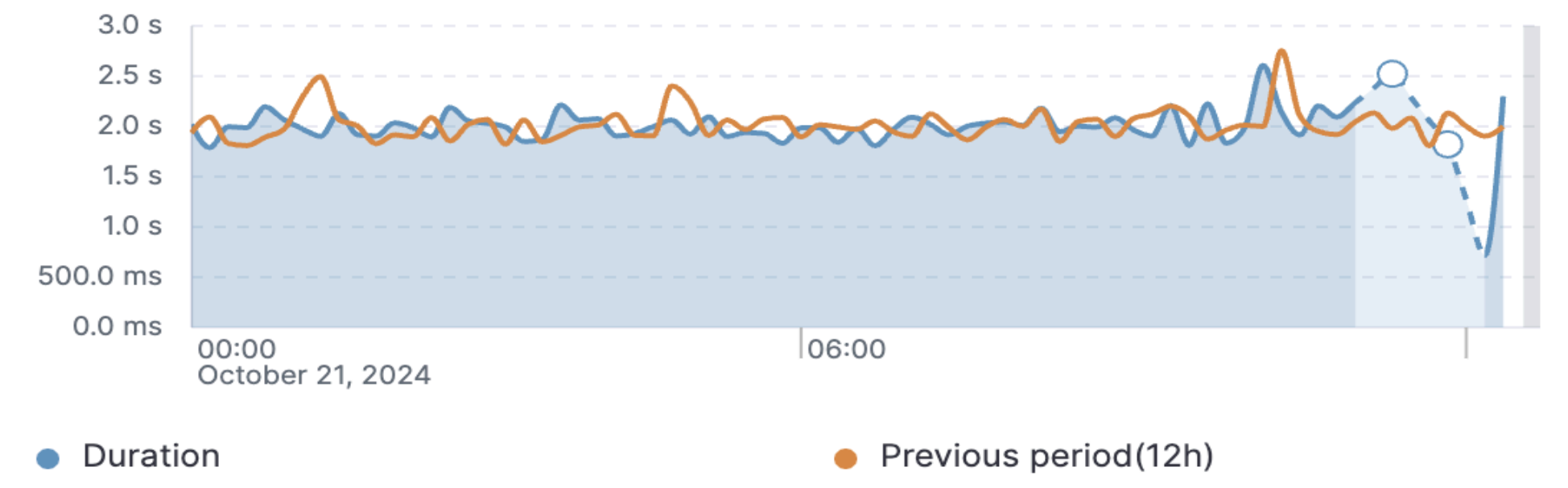
Europe - United Kingdom

Duration ? **2 s**

## Replicator Login Journey

**Status** Succeeded   **Location** Europe - United Kingdom   **Enabled (all locations)**

### Duration



### Monitor details

**URL** <https://synthetics-replicator.netlify.app/>

**Last run** Oct 21, 2024 @ 1:09:24 PM

**Last modified** Oct 21, 2024 @ 1:08:11 PM

**Project ID** synthetics-replicator-tests

**Monitor ID** synthetics-replicator-monitor-login-synthetics-replicator-tests-default

**Monitor type** Journey

**Frequency** Every 30 minutes

**Locations** Europe - United Kingdom

**Tags** --

Close

Go to monitor →



### Observability

- Overview
- Alerts
- SLOs
- Cases
- AI Assistant

### Logs

- Explorer **BETA**
  - Stream
  - Anomalies
  - Categories
- ### Infrastructure
- Inventory
  - Metrics Explorer
  - Hosts **BETA**

### APM

- Services
- Traces
- Dependencies

### Synthetics

- Monitors
- TLS Certificates

### User Experience

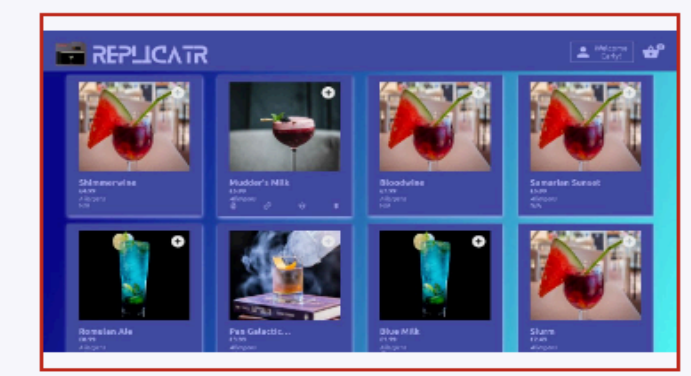
< Replicator Login Journey

# Test run details

**Location** Europe - United Kingdom **Status** Failed **Date** Oct 21, 2024 @ 1:09:24 PM

⚠ **Error running test**  
 error executing step: expect(received).toBeGreaterThan(expected)

**Step 3 of 4** < Previous 3 / 4 Next >



**Step name**  
 assert manual login successful  
Failed After 1.1 sec  
⚠ [View error details](#) [View performance breakdown](#)

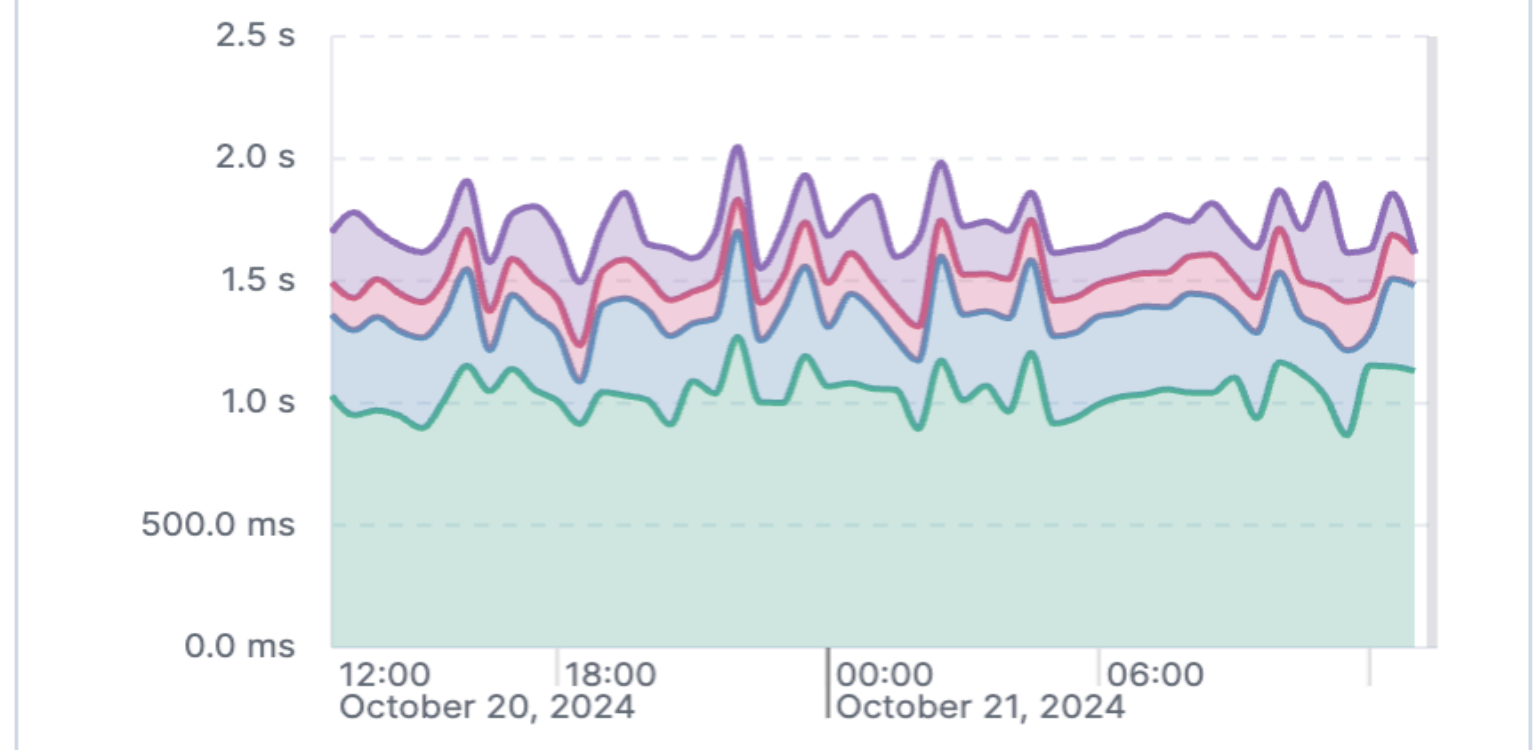
**Stacktrace** Code executed Console

```

Error: expect(received).toBeGreaterThan(expected)

Expected: > 100
Received: 23
  
```

### Duration by step Last 24 hours



- assert manual login successful
- assert home page loads

**Steps executed** Total duration: 1.6 sec



Observability

- Overview
- Alerts
- SLOs
- Cases
- AI Assistant

Logs

- Explorer **BETA**
- Stream
- Anomalies
- Categories

Infrastructure

- Inventory
- Metrics Explorer
- Hosts **BETA**



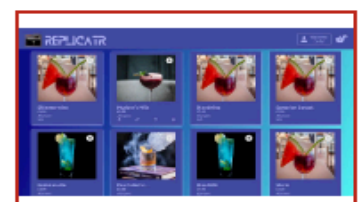
APM

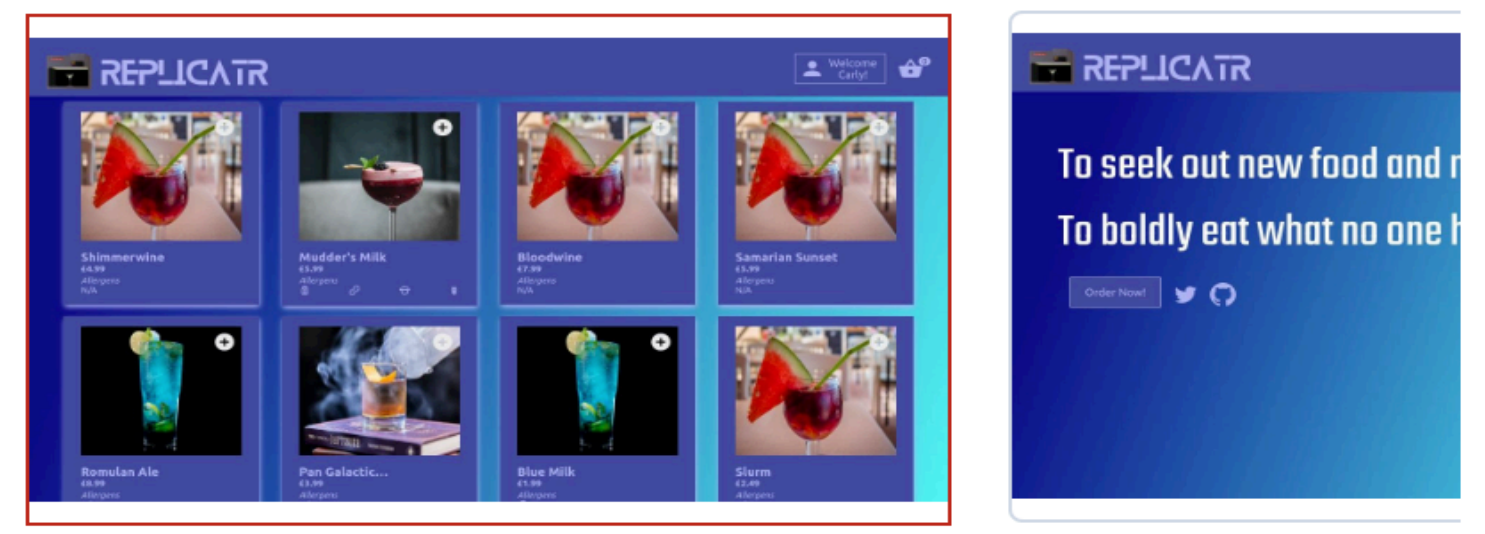
- Services
- Traces
- Dependencies

Synthetics

- Monitors
- TLS Certificates

User Experience

>	1		assert home page loads	<b>Complete</b> after 349 ms	359 ms
>	2		assert go to login page	<b>Complete</b> after 130 ms	179 ms
∨	3		assert manual login successful	<b>Failed</b> after 1.13 s	1.15 s



Blocked	<b>32 ms</b> ↑	Blocked	<b>33 ms</b>
DNS	-- ↑	DNS	--
Connect	-- ↑	Connect	--
TLS	-- —	TLS	--
Wait	<b>1.10 s</b> ↓	Wait	<b>2.04 s</b>
Receive	<b>1.95 s</b> ↑	Receive	<b>587 ms</b>
Send	<b>0 ms</b> —	Send	<b>0 ms</b>
Step duration	<b>1.1 s</b> ↑	Step duration	<b>1.2</b>
LCP	<b>0 ms</b> —	LCP	<b>0 m</b>
FCP	<b>0 ms</b> —	FCP	<b>0 m</b>
CLS	<b>0</b> —	CLS	

Monitor details

<b>URL</b>	<a href="https://synthetics-replicator.netlify.app/">https://synthetics-replicator.netlify.app/</a>
<b>Last run</b>	Oct 21, 2024 @ 1:09:24 PM
<b>Last modified</b>	Oct 21, 2024 @ 1:08:11 PM
<b>Project ID</b>	synthetics-replicator-tests
<b>Monitor ID</b>	synthetics-replicator-monitor-login-synthetics-replicator-tests-default
<b>Monitor type</b>	Journey
<b>Frequency</b>	Every 30 minutes
<b>Tags</b>	--



Observability

- Overview
- Alerts
- SLOs
- Cases
- AI Assistant

Logs

- Explorer BETA
- Stream
- Anomalies
- Categories

Infrastructure

- Inventory
- Metrics Explorer
- Hosts BETA

APM

- Services
- Traces
- Dependencies

# Rules

Rules Logs

● Succeeded: 4 ● Failed: 0 ● Warning: 0

4 rules

<input type="checkbox"/>	Name ↑	Last run	Noti
<input type="checkbox"/>	<a href="#">monitors-alive-check</a> Uptime monitor status	Oct 18, 2024 16:09:06pm 4 minutes ago	
<input type="checkbox"/>	<a href="#">Replicator Order Journey(Simple status alert)</a> Uptime monitor status	Oct 18, 2024 16:12:18pm 29 seconds ago	
<input type="checkbox"/>	<a href="#">Synthetics internal alert</a> Synthetics monitor status	Oct 18, 2024 16:12:18pm 29 seconds ago	
<input type="checkbox"/>	<a href="#">Synthetics internal TLS alert</a> Synthetics TLS certificate	Oct 18, 2024 16:12:18pm 29 seconds ago	

Rows per page: 10

## Create rule

Name

Tags Optional

### Synthetics monitor status

Alert when a monitor is down. [Learn more](#)

Manage synthetics monitor status rule actions.

Check every  minutes

Advanced options

Alert after  consecutive matches

Cancel





HELP!



START  
HERE



**COME  
TOGE  
THER**



**CLICK ME!**



# RESOURCES

- <https://ela.st/synthetics-replicator>
- <https://ela.st/synthetics-replicator-blog>
- <https://ela.st/synthetics-auth-advent>
- <https://semaphoreci.com/blog/synthetic-monitoring>

**COME  
TOGE  
THER**



**CLICK ME!**

@CarlyLRichmond



DANKE!

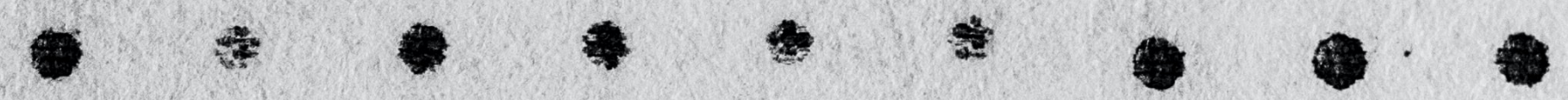
THANK YOU!

MERCI!

GRAZIE!

GRACIAS!

DANK JE WEL!



CLICK ME!

@CarlyLRichmond