Improved Methods for Using Selenium with Splunk to Monitor Web Apps

Justin Brown
IT Engineer | Pacific Northwest National Laboratory

splunk> .conf19

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

About Me

… and Pacific Northwest National Laboratory

.conf19
splunk>

About Pacific Northwest National Laboratory

https://energy.gov/about-national-labs

Idaho National Laboratory
Idaho Falls, Idaho

National Renewable Energy Laboratory
Golden, Colorado

Fermi National Accelerator Laboratory
Batavia, Illinois

Brookhaven National Laboratory
Upton, New York

Pacific Northwest National Laboratory
Richland, Washington

Argonne National Laboratory
Argonne, Illinois

National Energy Technology Laboratory
Morgantown, West Virginia
Pittsburgh, Pennsylvania

Ames Laboratory
Ames, Iowa

SLAC National Accelerator Laboratory
Menlo Park, California

Princeton Plasma Physics Laboratory
Princeton, New Jersey

Thomas Jefferson National Accelerator Facility
Newport News, Virginia

Sandia National Laboratory
Livermore, California
Albuquerque, New Mexico

Oak Ridge National Laboratory
Oak Ridge, Tennessee

Lawrence Berkeley National Laboratory
Berkeley, California

Lawrence Livermore National Laboratory
Livermore, California

Los Alamos National Laboratory
Los Alamos, New Mexico

Savannah River National Laboratory
Aiken, South Carolina

splunk> .conf19

About Pacific Northwest National Laboratory

The nation’s premier laboratory for scientific discovery in chemistry, earth sciences, data analytics, and for solutions to the nation’s toughest challenges in energy resiliency and national security. Founded in 1965, PNNL is operated by Battelle for the U.S. Department of Energy's Office of Science.

College Park, MD
Washington, DC

4,400+

STAFF

Agenda

- The Problem
- Why Selenium?
- Evolution at PNNL
- Current State Demo
- Implementation Details
- Monitoring Tips and Tricks

The Problem

An overview of the issues we were trying to solve

The Problem(s)

Hundreds of sites not being monitored

Almost zero root cause data from existing monitoring

Configuration for alerts was all manual

Desired Outcomes

We know first

Better data for investigating root cause

Automated testing

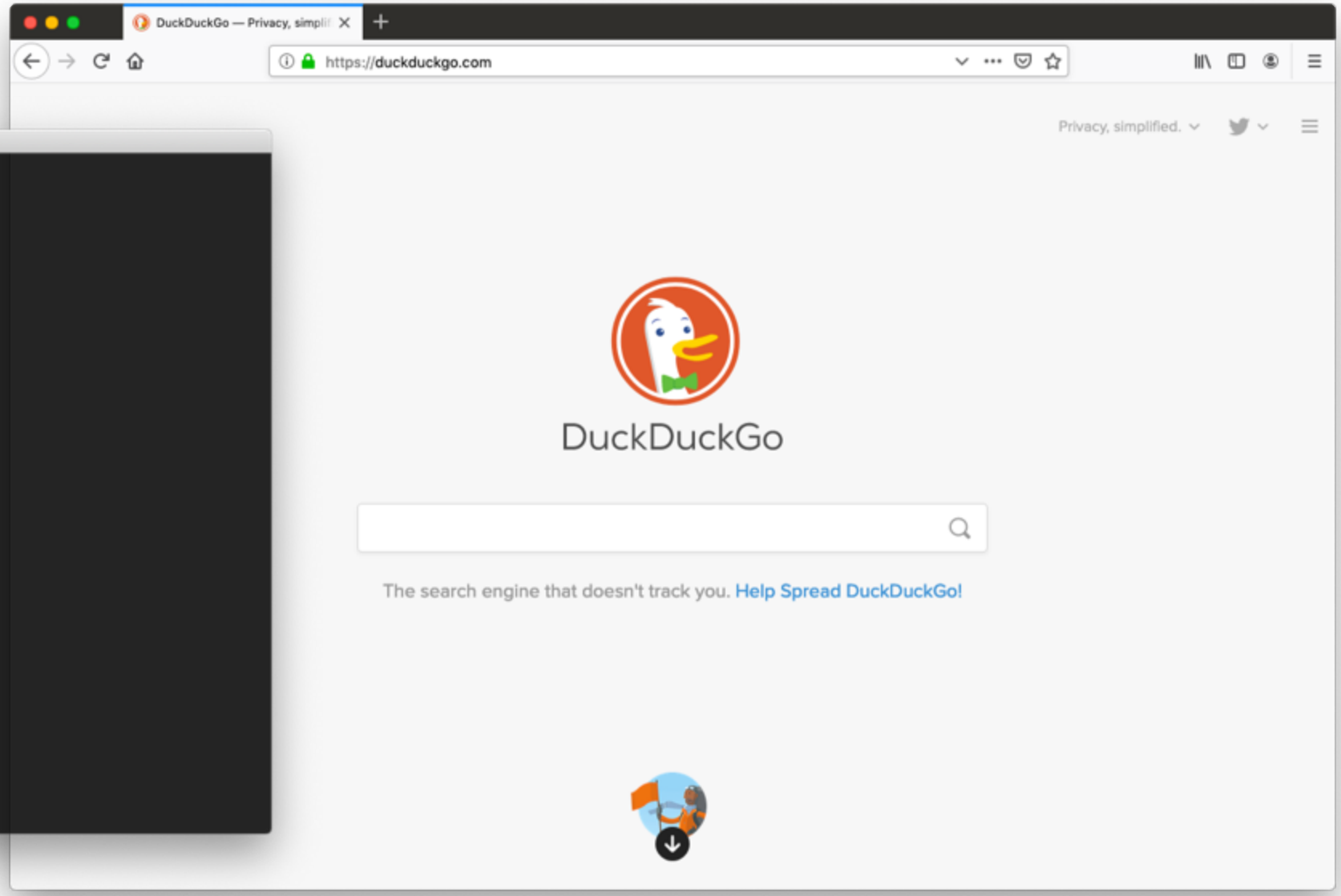
Why Selenium?

What is Selenium?

- Selenium – Web Browser Automation
  - http://www.seleniumhq.org
- Python Bindings for Selenium WebDriver
  - http://selenium-python.readthedocs.io

Demo

```
bash$    python
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
>>>   from selenium import webdriver
>>>   ff = webdriver.Firefox()
>>>   ff.get("https://www.duckduckgo.com")
>>>
```



There's was an app for that…

- **Splunk App for Synthetic Transaction Monitoring**
  - http://apps.splunk.com/app/1880
  - By: Elias Haddad

Splunk App for Synthetic Transaction Monitoring

Time	Event
10/24/19 10:16:12 AM	2019-10-24 10:16:12 app_name="External Sites" transaction_name=”www.google.com" event_type="end" transaction_end="2019-10-24 10:16:12" transaction_end_epoch="1439831772.32" execution_id="a30fd38f-4503-11e5-a699-0050568542ad" transaction_duration=”1.56299996376" browser="Firefox" browser_version="40" os="Windows 7" os_version="" ip="173.194.123.72”
10/24/19 10:16:02 AM	2019-10-24 10:16:02 app_name="External Sites" transaction_name=”www.google.com" event_type="start" transaction_start="2019-10-24 10:16:02" transaction_start_epoch="1439831762.76" execution_id="a30fd38f-4503-11e5-a699-0050568542ad" browser="Firefox" browser_version="40" os="Windows 7" os_version="" ip="173.194.123.72”

Application Performance

Edit | More Info

Time Range: Today

Application Name: External Sites

Current Status

Performance KPI <1m ago

100.0

▲ +0%

Availibility KPI <1m ago

100.0

— no change

Performance SLA

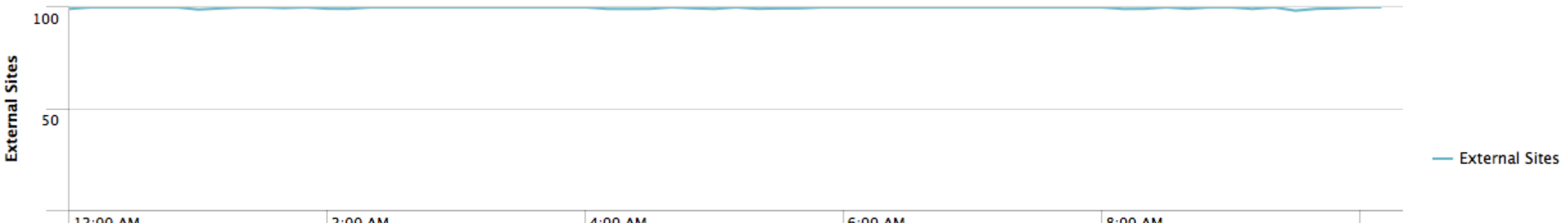
<1m ago

Availibility SLA

<1m ago

Application Availibility

<1m ago



Evolution at PNNL

Because science

Filling the Gaps

- **Two events per test**
  - Needed to use a TRANSACTION command
  - Status had to be interpreted
  - False positives
- **Missing data**
  - No server info
  - No context for failures
- **Building tests was manual**
  - Slow to build
  - Difficult to maintain

Round One Goals

- **Single event per test**
- **Capture error information**
  - Add Passed / Warning / Failed status
  - Capture error messages
- **Collect server information**

- **Build tests dynamically given an array of info**
  - URL
  - App Name
  - Transaction Name
  - Expected title
  - Server

Workflow of v1

- Report of top sites
- Windows VM with UF
  - Scripted input that reads from saved search
  - Sends results to indexers
- Dashboards
- Alerts

.conf2015

Using Web Logs in Splunk to Dynamically Create Synthetic Transaction Tests

Justin Brown

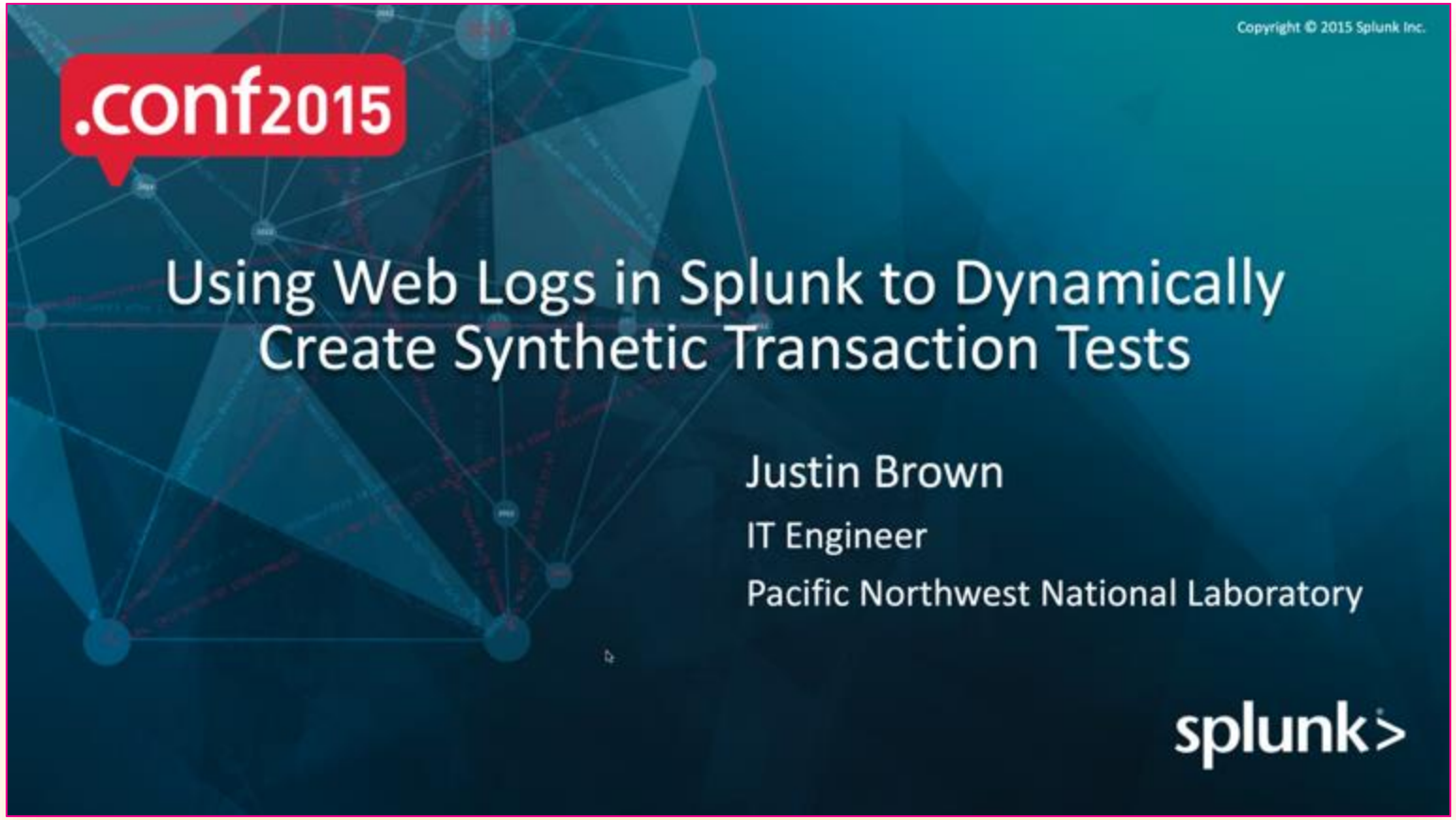
IT Engineer

Pacific Northwest National Laboratory

splunk>

Future Plans

- User Interface for adding monitoring
- Better root cause analysis
- Screenshots!
- More browsers supported
- More testing options
- Dependency checking



Current State Demo

Website Monitoring Configuration

Choose an existing site

Create New

Add New

Item ID	URL	Browser	Enabled
DuckDuckGo	www.duckduckgo.com	Chrome	☑

To:	CC:	BCC:	Alert Threshold
justin@pnnl.gov			2

Testing Steps

1	Go to URL	https://www.duckduckgo.com		☑ ✕
2	Verify Title	Contains	Privacy	☑ ✕

Add Step

Run Test

Save

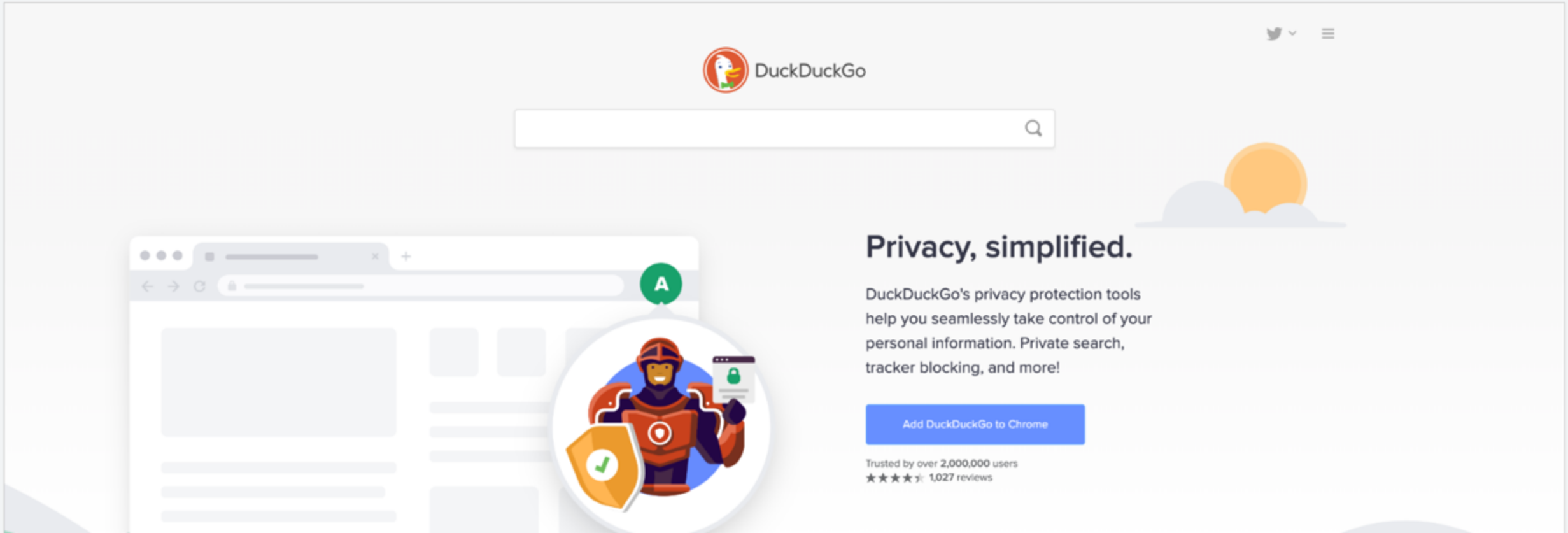
Testing Steps

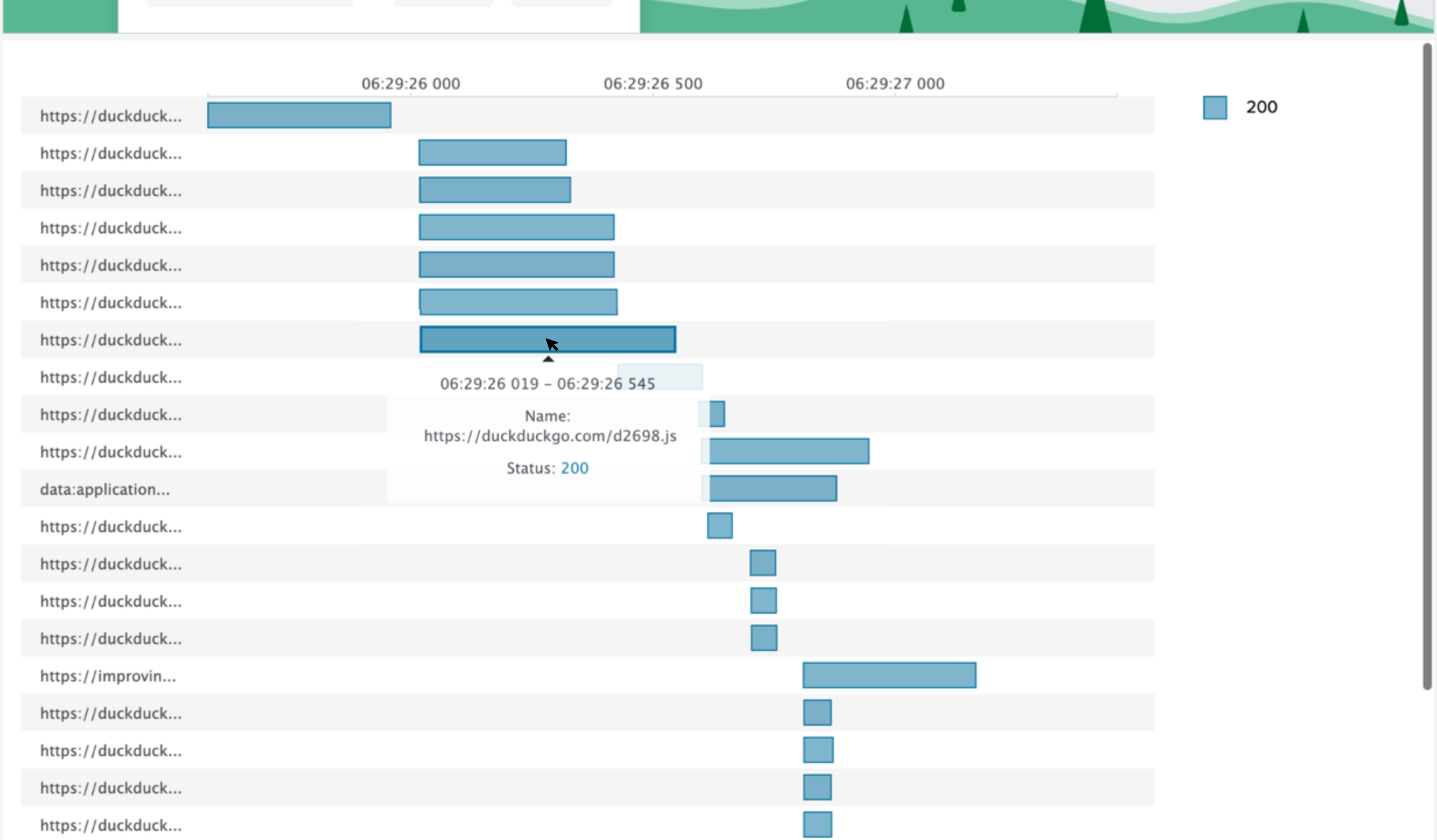
1	Go to URL	https://www.duckduckgo.com		PASSED: 1.24 SECONDS	☑ ✕
2	Verify Title	Contains	Privacy	PASSED: 0.01 SECONDS	☑ ✕

Add Step | Run Test | Save

Status: Passed

All tests passed!




06:29:26 000
06:29:26 500
06:29:27 000
200
https://duckduck...
https://duckduck...
https://duckduck...
https://duckduck...
https://duckduck...
https://duckduck...
https://duckduck...
https://duckduck...
06:29:26 019 – 06:29:26 545
Name:
https://duckduckgo.com/d2698.js
Status: 200
https://duckduck...
https://duckduck...
data:application...
https://duckduck...
https://duckduck...
https://duckduck...
https://duckduck...
https://improvin...
https://duckduck...
https://duckduck...
https://duckduck...
https://duckduck...

Root Cause Analysis

splunk>enterprise App: Website Monitoring ▾ Justin ▾ Messages ▾ Settings ▾ Activity ▾ Help ▾ Find

Website Monitoring Overview | Website Performance Details | Monitoring Configuration | More ▾ | Website Monitoring

Website Performance Details

Edit | Export ▾ | ...

Time Range: Sep 4, 2019

Application Name: PeopleSoft Enterpri... ✕

Hide Filters

Page Availibility

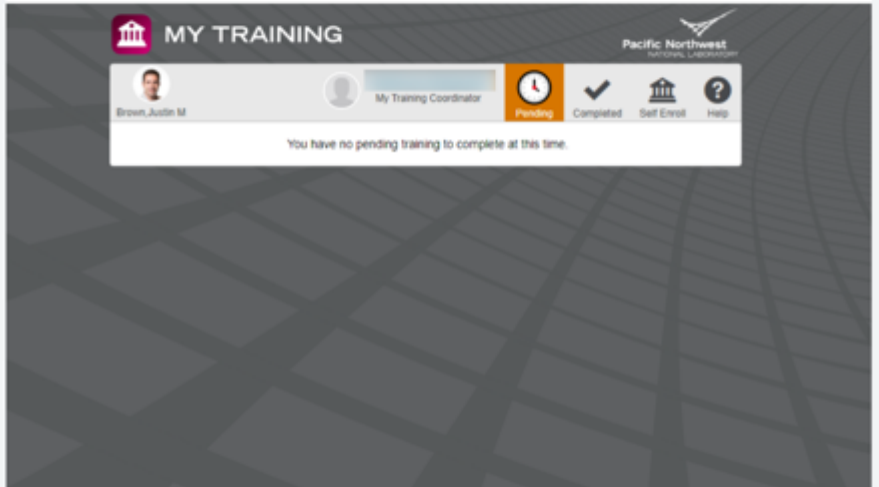
Test Steps

	Test	Duration	Status
1	Go to url https://	1.87	Passed
2	Verify title contains "My Homepage"	0.02	Passed
3	Go to url https://	30.06	Failed

Test Details

Details	Results
Test OS	Windows 10
Test Server	
Test Browser	Chrome 76.0.3809
Status	Failed ✕
Error	Timeout: Page did not load within 30 seconds
Duration	31.95 ✕
Successful Tests	2 of 4

Screenshot

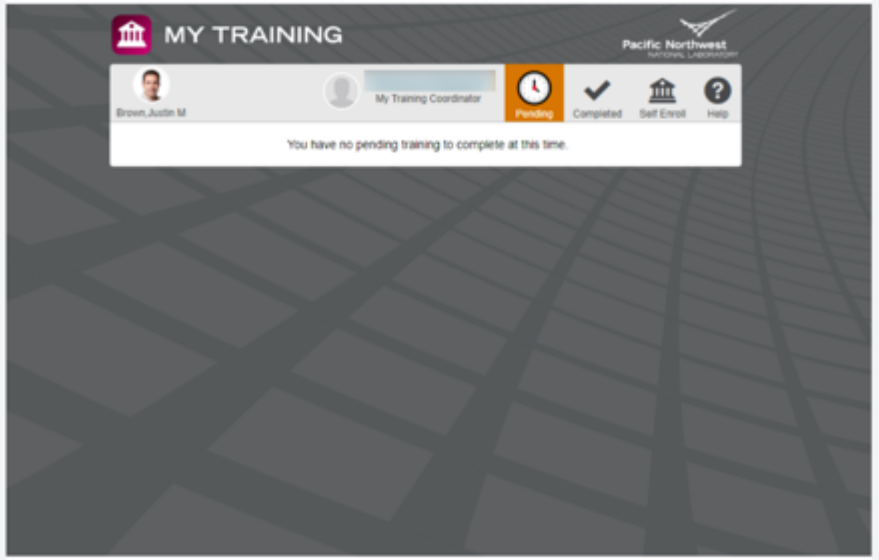


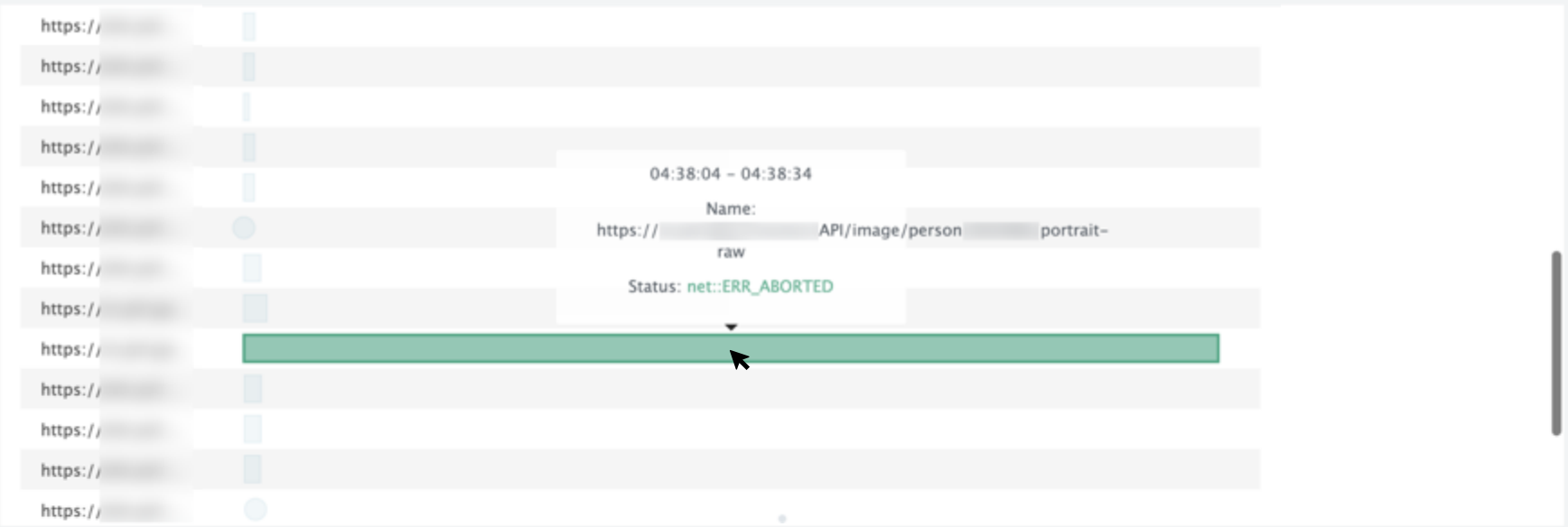
Root Cause Analysis

Test Details

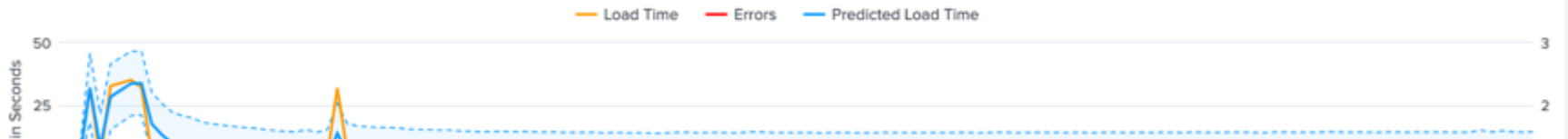
Details	Results
Test OS	Windows 10
Test Server	[redacted]
Test Browser	Chrome 76.0.3809
Status	Failed ✗
Error	Timeout: Page did not load within 30 seconds
Duration	31.95 ✗
Successful Tests	2 of 4

Screenshot



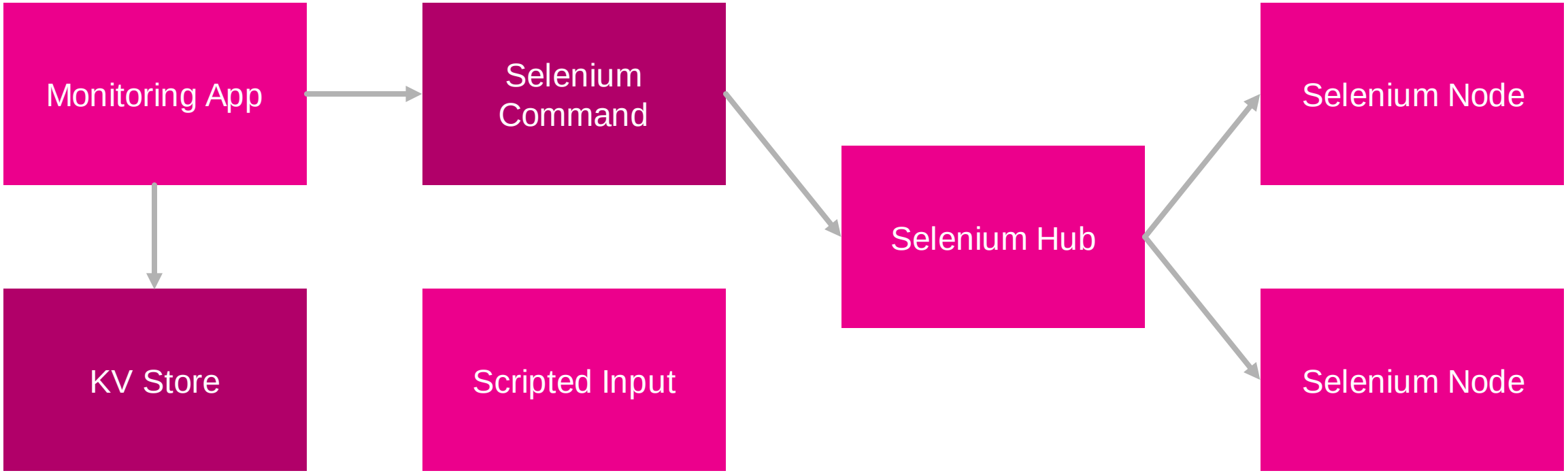


Average Page Load Duration Over Time

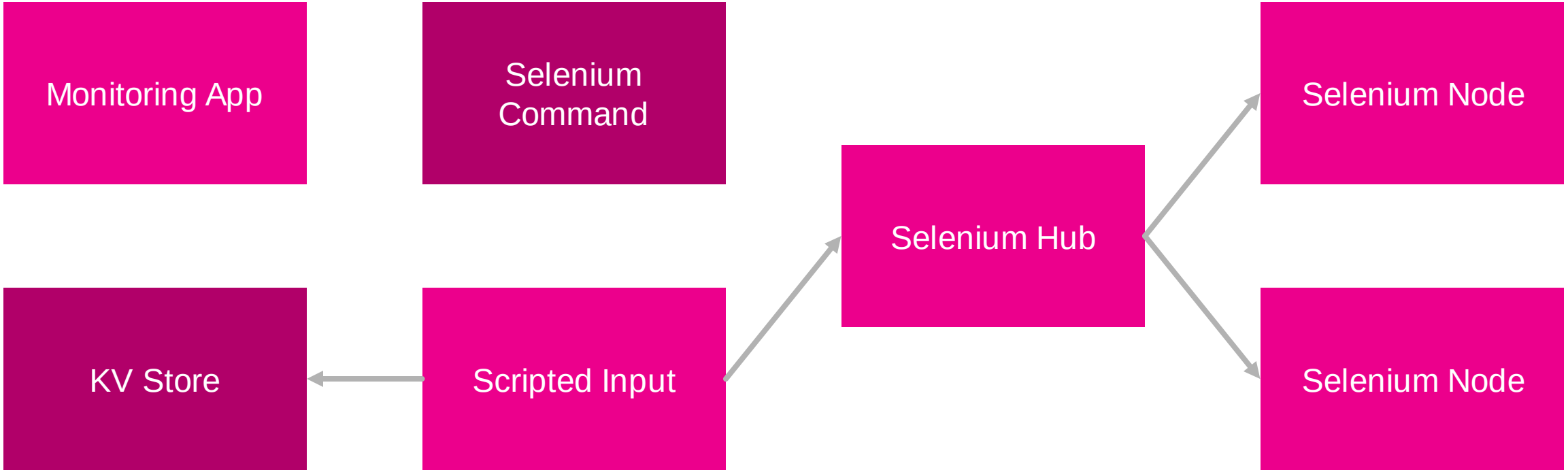


Implementation Details

Components



Components



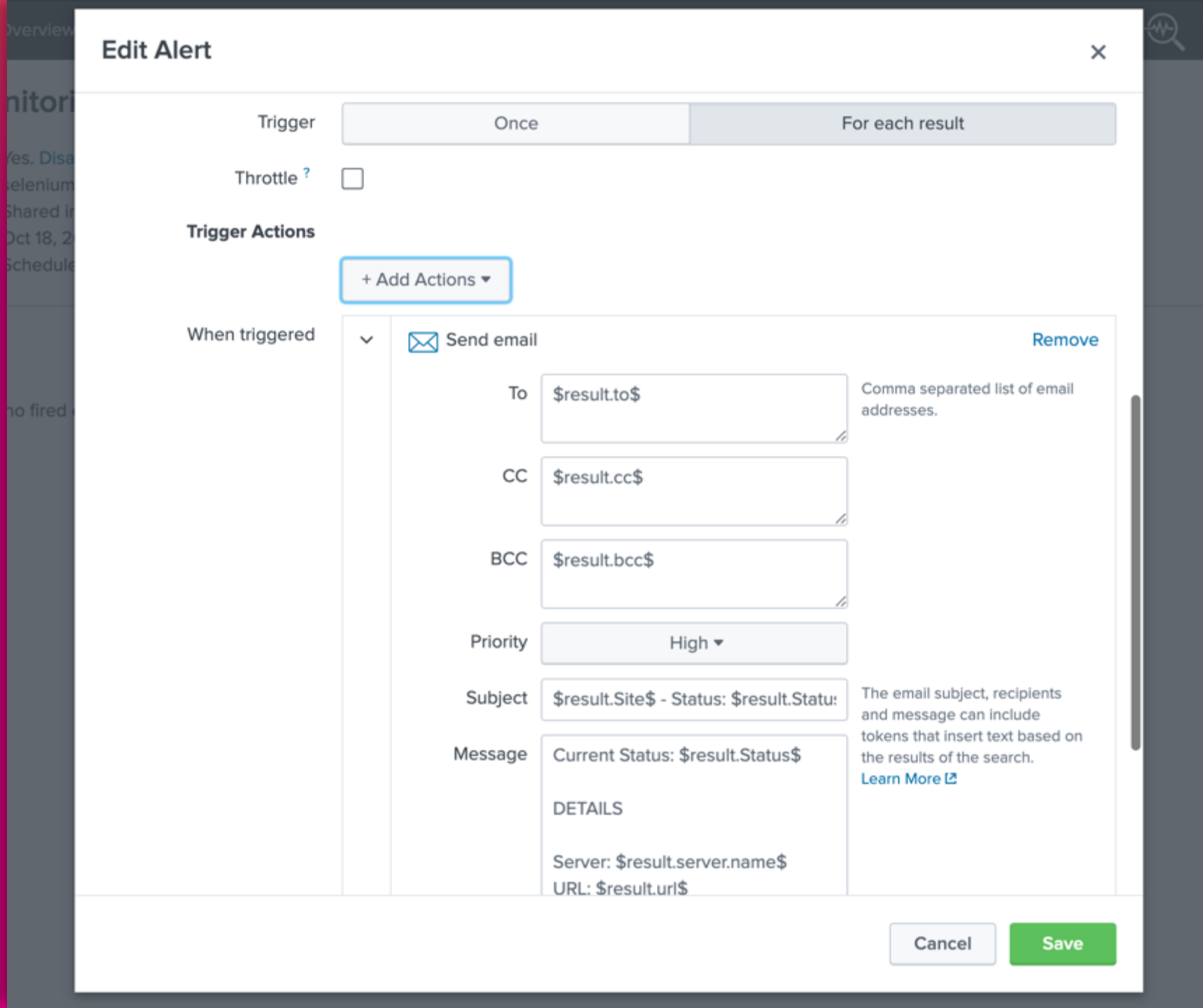
Monitoring Tips & Tricks

Best Practices

1. Create “Health Check” pages for dependency checking
2. Create ties to your CMDB
3. Simplify alerts

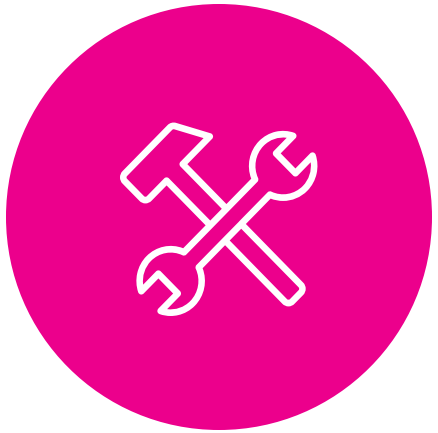
Simplify Alerts

- Single alert for all
- Trigger “For each result”
- Use lookup for email recipients



Get the bits…

Basic web monitoring app



Custom Splunk selenium command

Sample scripted input

Selenium helper scripts & docs

https://github.com/justinatpnnl

Questions?

Thank You!