Full Stack Self-Service Diagnostics with Dynatrace

dynatrace **Perform**

HOT Day sponsored by aws

Wayne Segar
Senior Architect

Matt Miller
Title

Chris Schwarzbauer
Title

Prerequisites

- Personal AWS Account:
  - https://aws.amazon.com/free/
- Postman API Client
  - https://www.getpostman.com/downloads/
- Download/Clone the Repository:
  - https://github.com/wmsegar/Dynatrace-Hotday-2019-SelfServiceDiagnostics

    or
  - https://bit.ly/2MkMkgl

Agenda

- Formalities
  - Introductions
  - What we're covering
  - Level of Expertise required
  - Individual EC2 Instances
  - EasyTravel – Book Exciting travel to far away destinations
- Setup and Launching our EC2 instance with EasyTravel
- Configuration API
- Use Case 1 – CPU Increase
- Use Case 2 – Booking Failure
- Use Case 3 – Slow Login
- Use Case 4 – Homepage Slowdown
- Tips and Tricks

What we're covering/not covering

- This is an exploration of common
- This is not a product tutorial
- This is not a test
- There is no need to be an AWS expert and everything we are covering can be applied to ANY environment (e.g. Azure, GCP, Kubernetes, On-Prem, etc…)
- The goal of this sessions is:
  - Help you become more comfortable using Dynatrace to diagnose application problems
  - Make the diagnosis easy for others in your organization
  - Not everybody needs to be a "Dynatrace Expert"

How we're covering

- We will walk through several Use Case Problem patterns
- Each Student will have their own private EC2 instance running a version of EasyTravel (created via CloudFormation)
- Each Student will have their own private Dynatrace SaaS tenant
    - Tenant ID and Credentials on your attached card
- During the course of this session we will be doing the following:
    - Creating our environment via CloudFormation
    - Performing Dynatrace configurations via the UI and the API
    - Triggering EasyTravel Problem Patterns via the EasyTravel Config UI

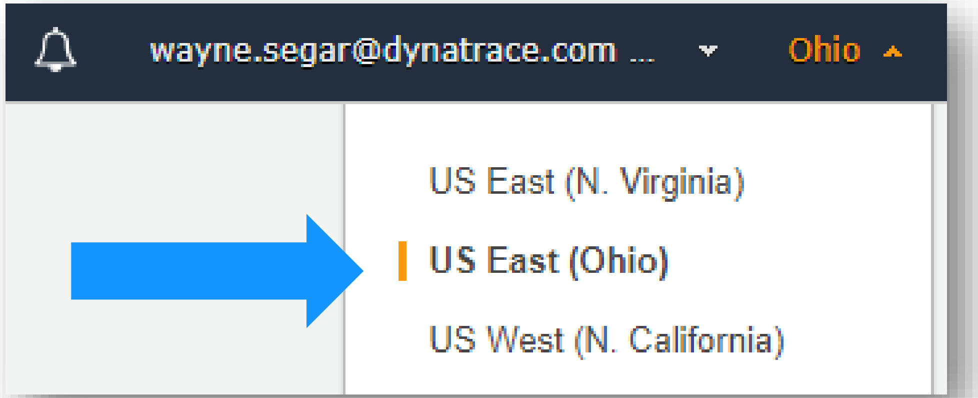
Level of Expertise

- Basic Dynatrace familiarity
- Expert certification not required
- REST API basics

Setup Work

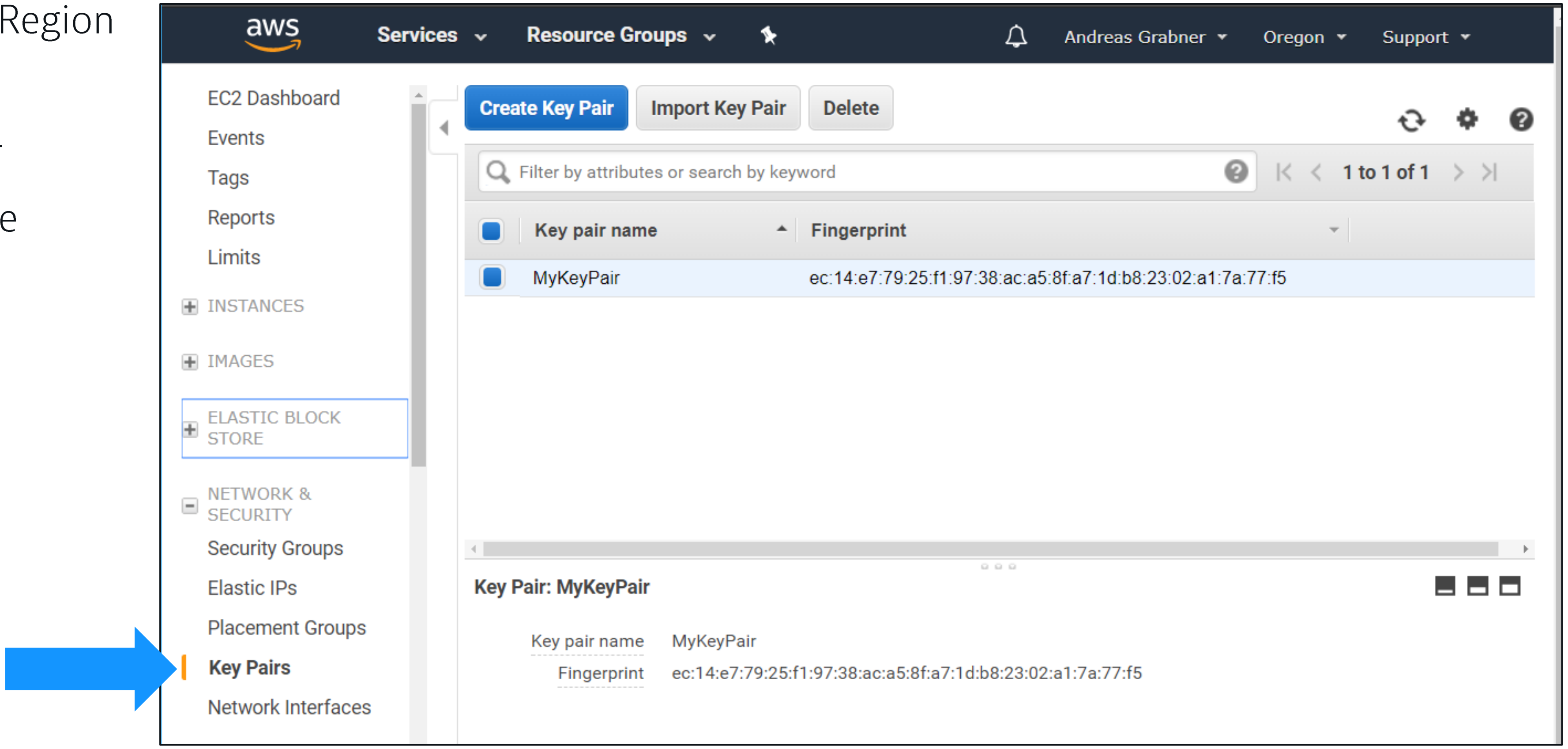
Setup AWS: Default Region

- Login to your AWS Account: https://aws.amazon.com
  - Validate and make note of your default region



Setup AWS: EC2 Key Pair

1. Create EC2 Key Pair
   1. Go to EC2 – Pick Region
   2. Click on Key Pairs
   3. Create a new Pair
   4. Store the .pem file



Key pair name	Fingerprint
MyKeyPair	ec:14:e7:79:25:f1:97:38:ac:a5:8f:a7:1d:b8:23:02:a1:7a:77:f5

Key Pair: MyKeyPair

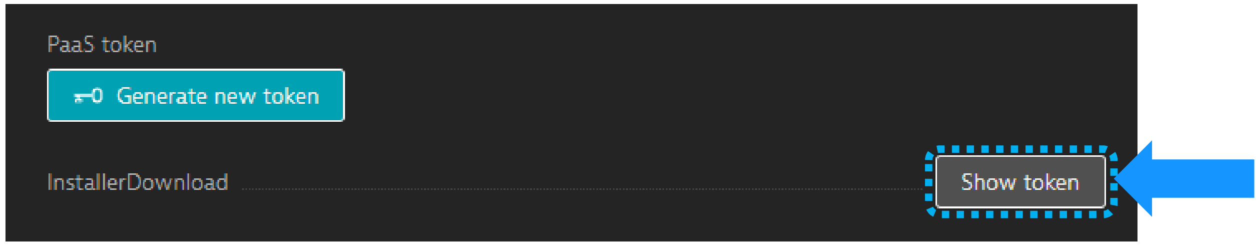
Key pair name	MyKeyPair
Fingerprint	ec:14:e7:79:25:f1:97:38:ac:a5:8f:a7:1d:b8:23:02:a1:7a:77:f5

Setup Dynatrace: Retrieve API Token

- Login to your Dynatrace SaaS Tenant provided to you:

https://<TENANT_ID>.sprint.dynatracelabs.com

- Navigate to:
  - Click “Deploy Dynatrace” from the left side menu
  - Click “Start Installation”
  - Click “Set up PaaS monitoring”
  - Click “Show token”



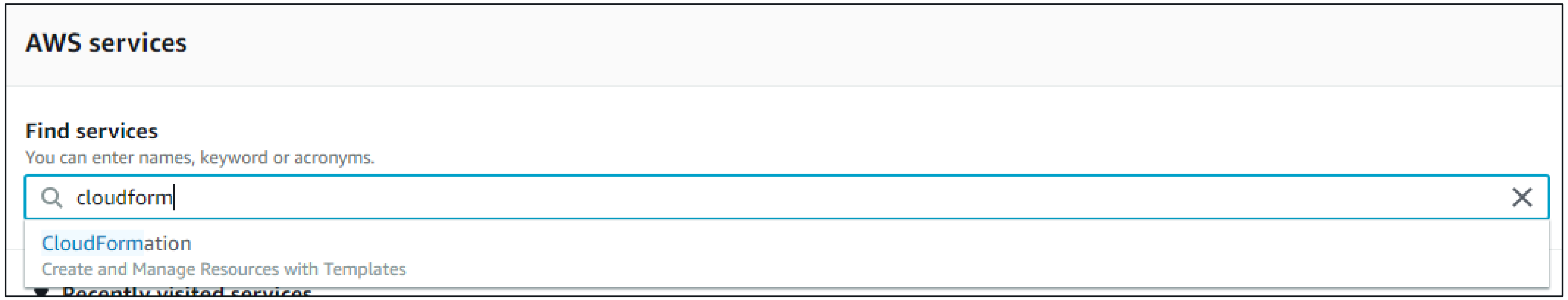
  - Copy your token to your clipboard

Launching our EC2 Instance

Our CloudFormation Template will create a Production Linux server with EasyTravel installed/running and deploy a OneAgent pointing to your Dynatrace SaaS Tenant

Launching the CloudFormation Stack

- What is CloudFormation?
  - A declarative way to define AWS Resources, e.g: EC2, CodeDeploy, CodePipeline, Lambda, Roles, ELBs, …
  - Learn more about CloudFormation: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide
- Our Steps
  - In your AWS Console browse to the CloudFormation Service

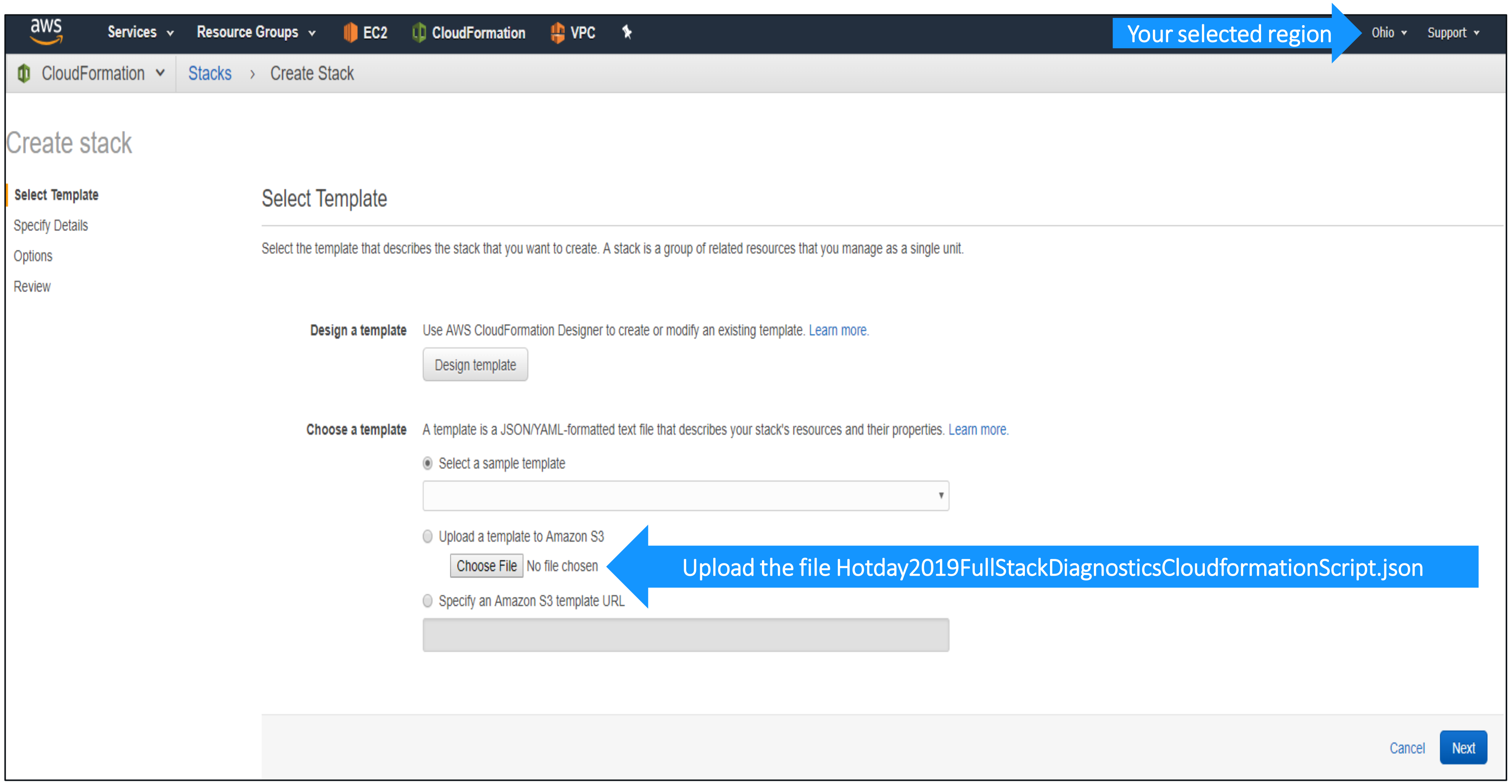

AWS services

Find services
You can enter names, keyword or acronyms.

cloudform

CloudFormation
Create and Manage Resources with Templates

Upload Template



Create stack

Select Template

Select the template that describes the stack that you want to create. A stack is a group of related resources that you manage as a single unit.

Design a template Use AWS CloudFormation Designer to create or modify an existing template. Learn more.

Choose a template A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties. Learn more.

[✓ Select a sample template]

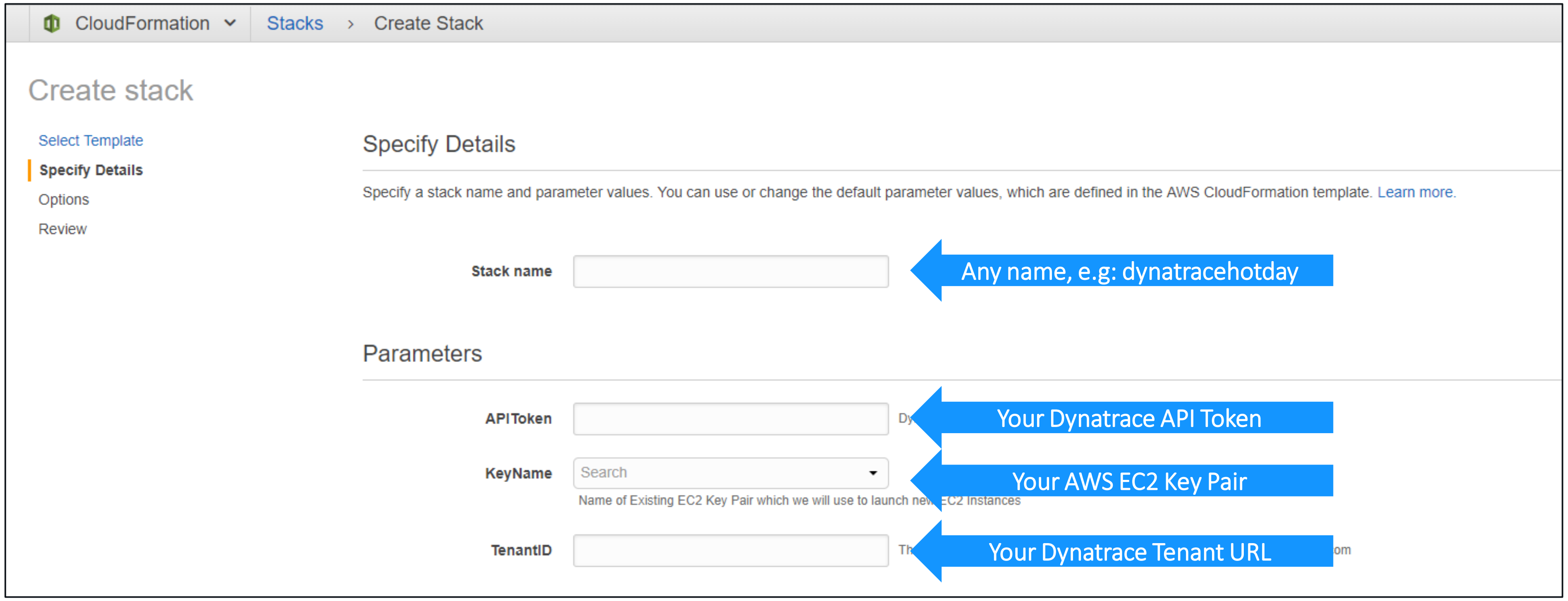
Upload a template to Amazon S3

Specify an Amazon S3 template URL

CloudFormation can be found in the GitHub repository

dynatrace **Perform**

Filling in the Details


CloudFormation › Stacks › Create Stack

Create stack

Select Template
Specify Details
Options
Review

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. Learn more.

Stack name — Any name, e.g: dynatracehotday

Parameters

APIToken — Your Dynatrace API Token

KeyName: Search — Your AWS EC2 Key Pair

Name of Existing EC2 Key Pair which we will use to launch new EC2 instances

TenantID — Your Dynatrace Tenant URL

Launching the Stack

Upload Template

Filling in the Details

Create and Confirm

Follow Events

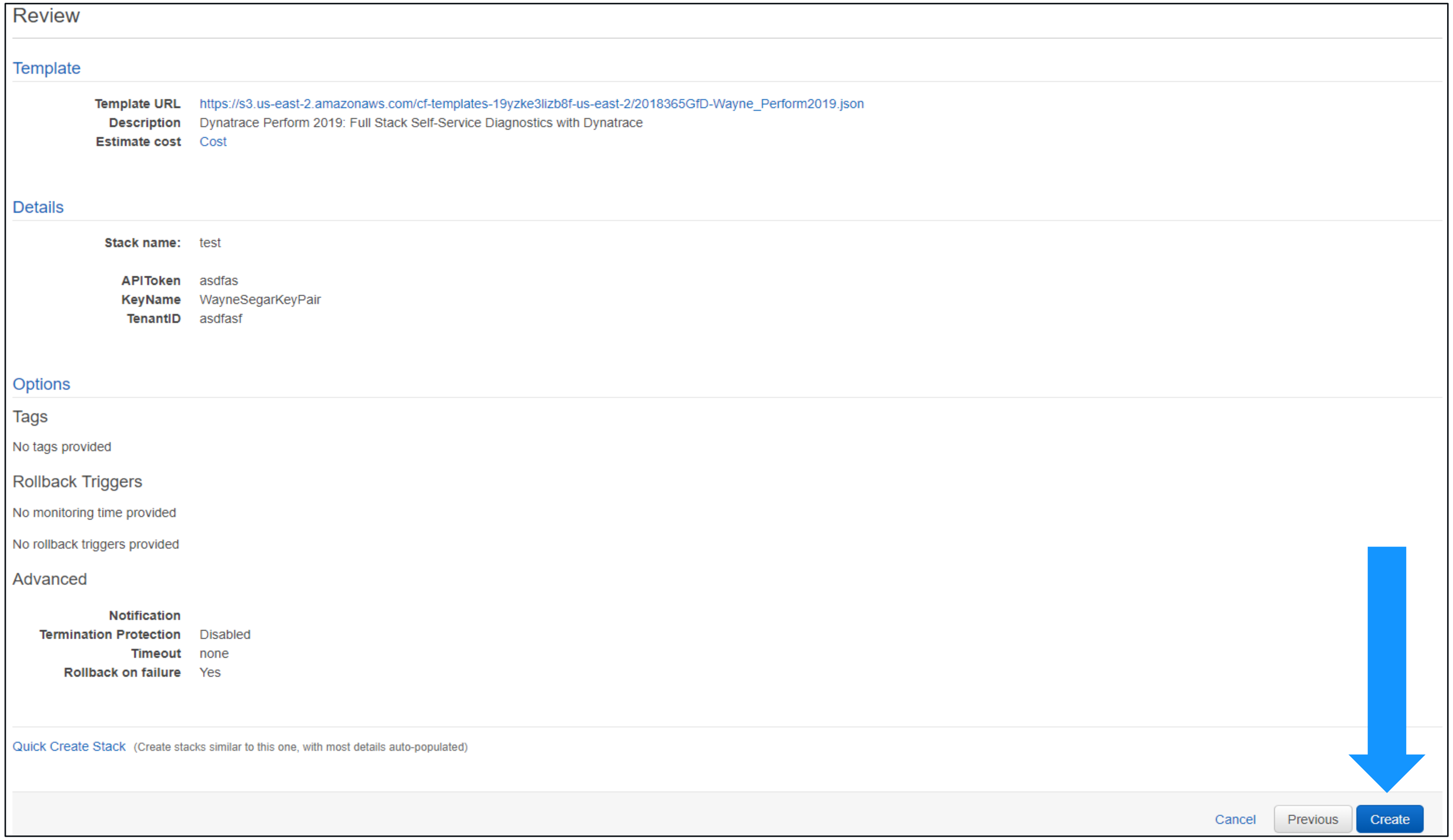
View Outputs

Validate Environment

Cleanup

dynatrace Perform

Create and Confirm



Review

Template

Template URL	https://s3.us-east-2.amazonaws.com/cf-templates-19yzke3lizb8f-us-east-2/2018365GfD-Wayne_Perform2019.json
Description	Dynatrace Perform 2019: Full Stack Self-Service Diagnostics with Dynatrace
Estimate cost	Cost

Details

Stack name:	test
APIToken	asdfas
KeyName	WayneSegarKeyPair
TenantID	asdfasf

Options

Tags

No tags provided

Rollback Triggers

No monitoring time provided

No rollback triggers provided

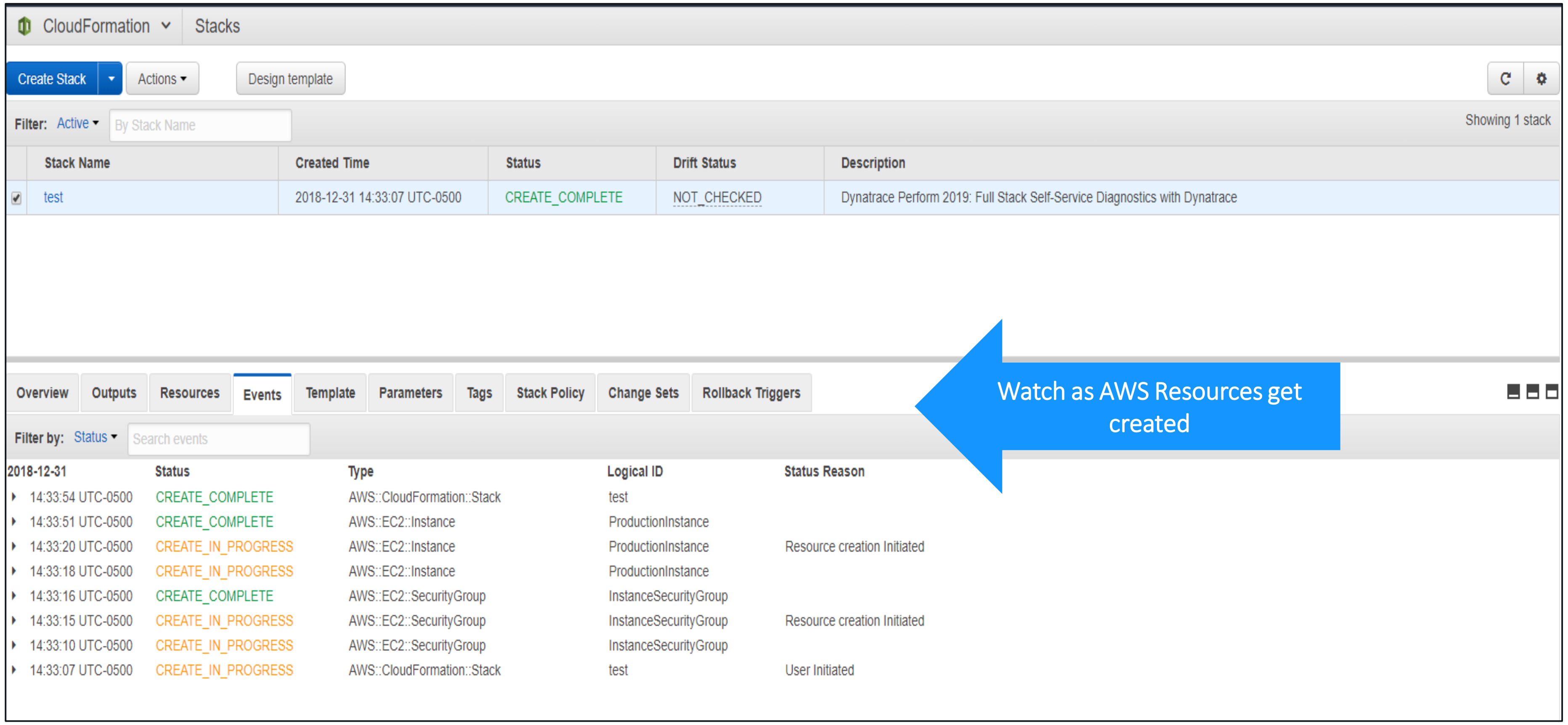
Advanced

Notification	
Termination Protection	Disabled
Timeout	none
Rollback on failure	Yes

Quick Create Stack (Create stacks similar to this one, with most details auto-populated)

dynatrace **Perform**

Follow Events

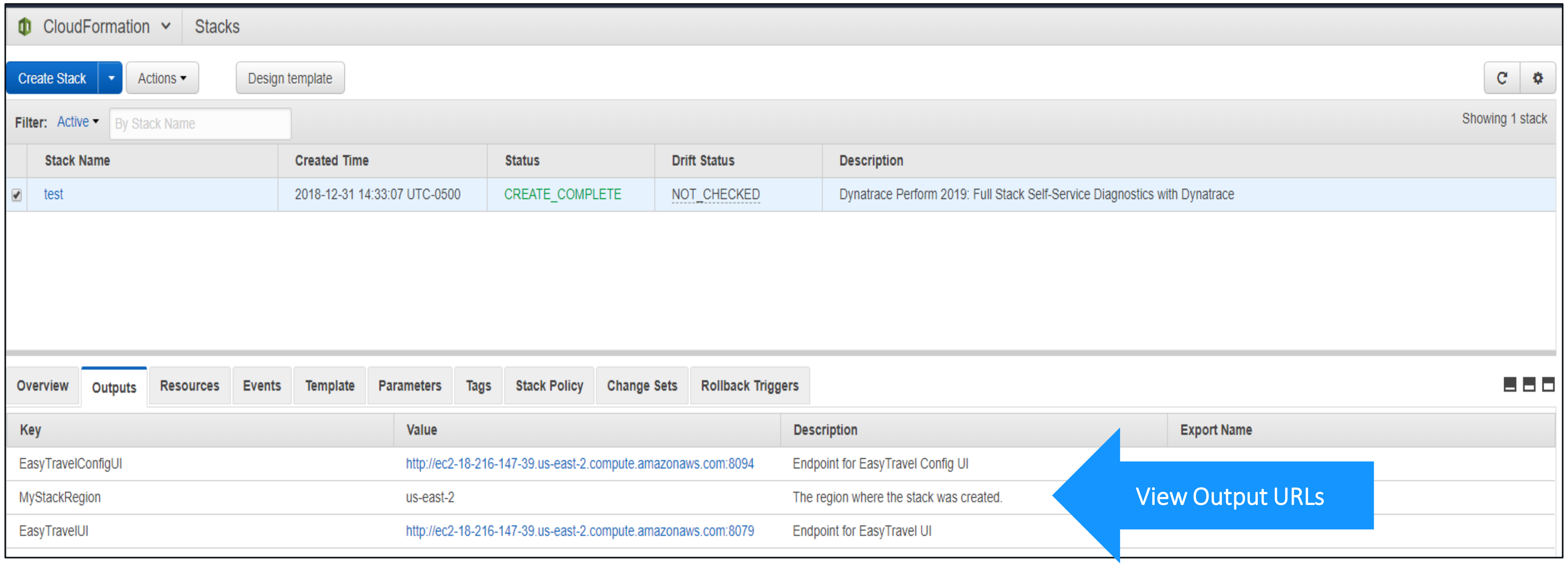


Stack Name	Created Time	Status	Drift Status	Description
test	2018-12-31 14:33:07 UTC-0500	CREATE_COMPLETE	NOT_CHECKED	Dynatrace Perform 2019: Full Stack Self-Service Diagnostics with Dynatrace

2018-12-31	Status	Type	Logical ID	Status Reason
14:33:54 UTC-0500	CREATE_COMPLETE	AWS::CloudFormation::Stack	test	
14:33:51 UTC-0500	CREATE_COMPLETE	AWS::EC2::Instance	ProductionInstance	
14:33:20 UTC-0500	CREATE_IN_PROGRESS	AWS::EC2::Instance	ProductionInstance	Resource creation Initiated
14:33:18 UTC-0500	CREATE_IN_PROGRESS	AWS::EC2::Instance	ProductionInstance	
14:33:16 UTC-0500	CREATE_COMPLETE	AWS::EC2::SecurityGroup	InstanceSecurityGroup	
14:33:15 UTC-0500	CREATE_IN_PROGRESS	AWS::EC2::SecurityGroup	InstanceSecurityGroup	Resource creation Initiated
14:33:10 UTC-0500	CREATE_IN_PROGRESS	AWS::EC2::SecurityGroup	InstanceSecurityGroup	
14:33:07 UTC-0500	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	test	User Initiated

dynatrace **Perform**

View Outputs



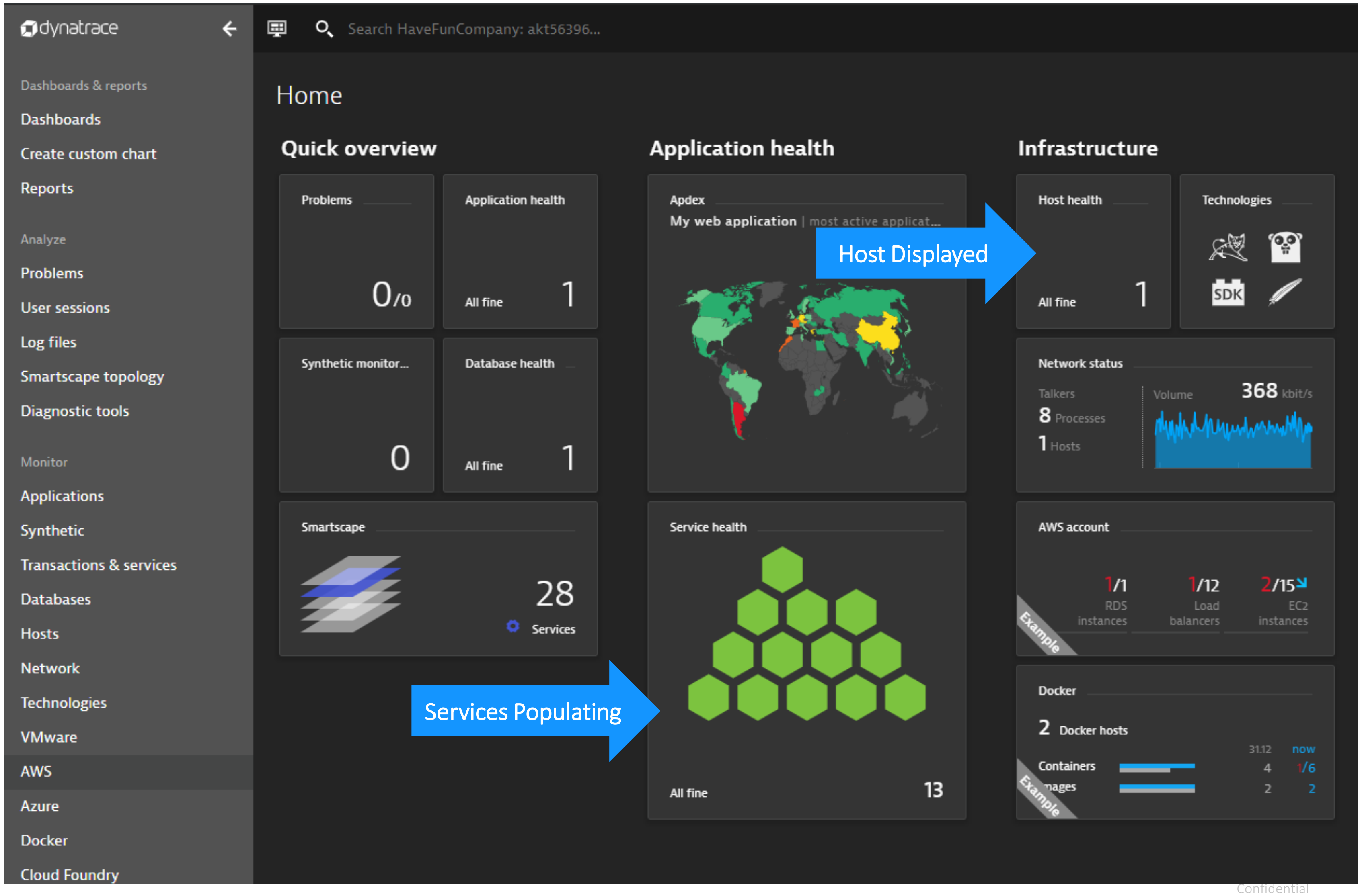
Stack Name	Created Time	Status	Drift Status	Description
test	2018-12-31 14:33:07 UTC-0500	CREATE_COMPLETE	NOT_CHECKED	Dynatrace Perform 2019: Full Stack Self-Service Diagnostics with Dynatrace

Key	Value	Description	Export Name
EasyTravelConfigUI	http://ec2-18-216-147-39.us-east-2.compute.amazonaws.com:8094	Endpoint for EasyTravel Config UI	
MyStackRegion	us-east-2	The region where the stack was created.	
EasyTravelUI	http://ec2-18-216-147-39.us-east-2.compute.amazonaws.com:8079	Endpoint for EasyTravel UI	

Note: It will take roughly 10 minutes for EasyTravel to fully come up

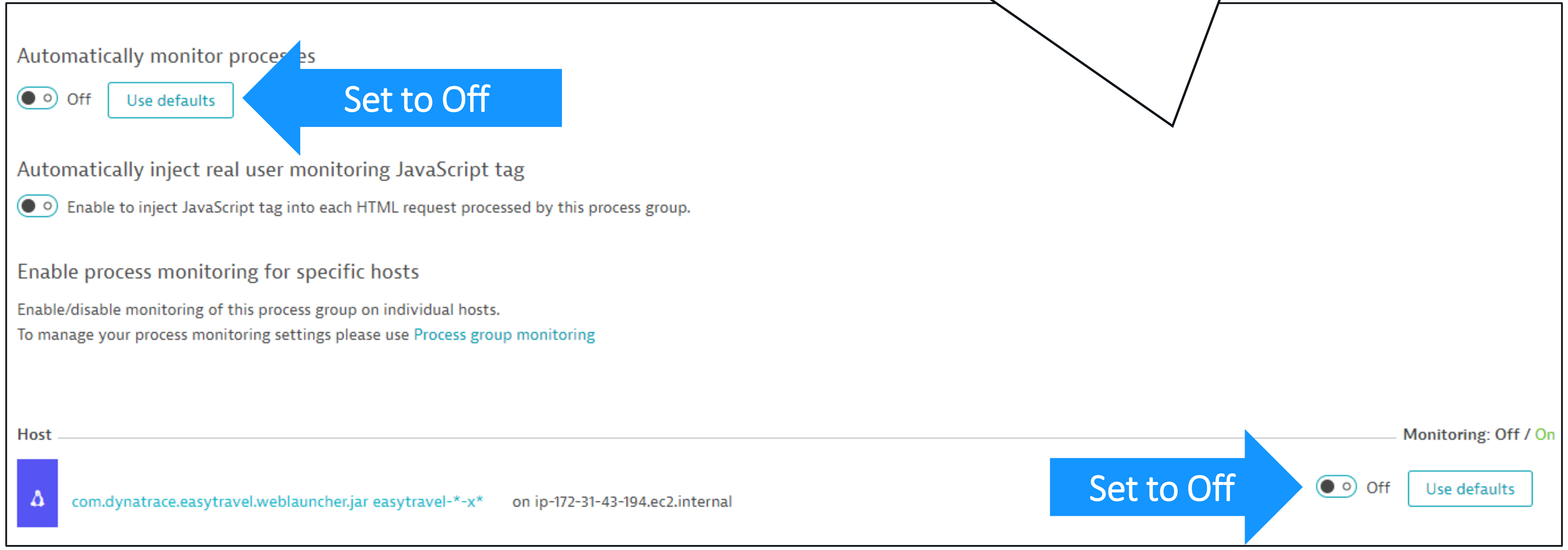
dynatrace
Perform

Validate Environment



Would it be nice if this was automated via the ConfigAPI?
Vote for RFE: https://bit.ly/2RGytHF

Technologies > Select "weblauncher.jar" p ess group
details > Edit > Disable for the host.



Automatically monitor processes
Off Use defaults

Automatically inject real user monitoring JavaScript tag
Enable to inject JavaScript tag into each HTML request processed by this process group.

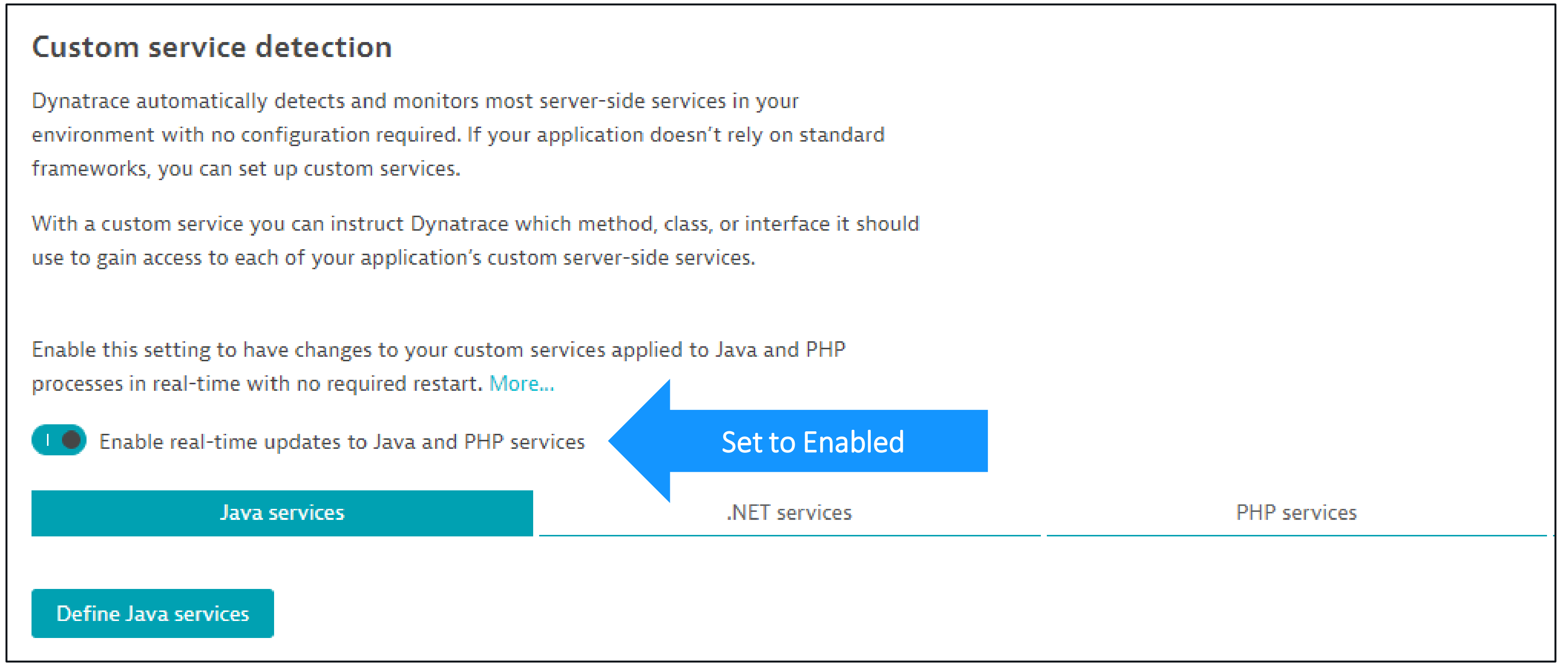
Enable process monitoring for specific hosts
Enable/disable monitoring of this process group on individual hosts.
To manage your process monitoring settings please use Process group monitoring

Host — Monitoring: Off / On
com.dynatrace.easytravel.weblauncher.jar easytravel-*-x* on ip-172-31-43-194.ec2.internal — Off Use defaults

dynatrace
Perform

Cleanup

- Enable Java Hotsensor Placement
  - We will be creating Custom Services in the next section so we will need this in order to avoid a required restart of the Java Processes
  - Settings > Server-side > service monitoring > Custom service detection > Enable real-time updates to Java and PHP services



Custom service detection

Dynatrace automatically detects and monitors most server-side services in your environment with no configuration required. If your application doesn't rely on standard frameworks, you can set up custom services.

With a custom service you can instruct Dynatrace which method, class, or interface it should use to gain access to each of your application's custom server-side services.

Enable this setting to have changes to your custom services applied to Java and PHP processes in real-time with no required restart. More...

Enable real-time updates to Java and PHP services: enabled

Configuration API

dynatrace
Perform

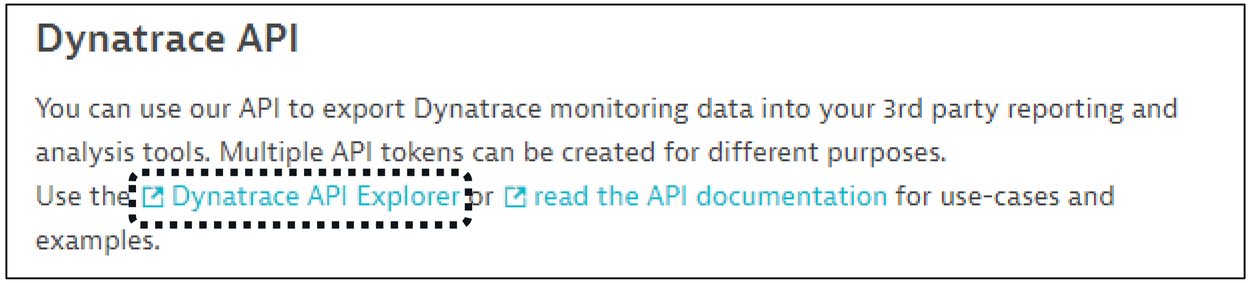
Dynatrace Configuration API

- The Dynatrace Configuration API allows API consumers to globally read and write the configuration settings of any Dynatrace environment

- The primary use cases for the Configuration API are:
  - Read and copy an existing environment configuration over to a new environment.
  - Create or change individual configuration settings via the API when setting up a new environment.
  - Read and store configurations, along with their histories, in a version management system (for example, Git).
  - Provide a mechanism to automate Monitoring/Performance-as-a-Service
    - Provide easy access to meaningful data to Application Owners with creation of Management Zones
    - Simplify complex environment with auto tagging rules
    - Extract specific business context information with request attributes

View the Configuration API Explorer

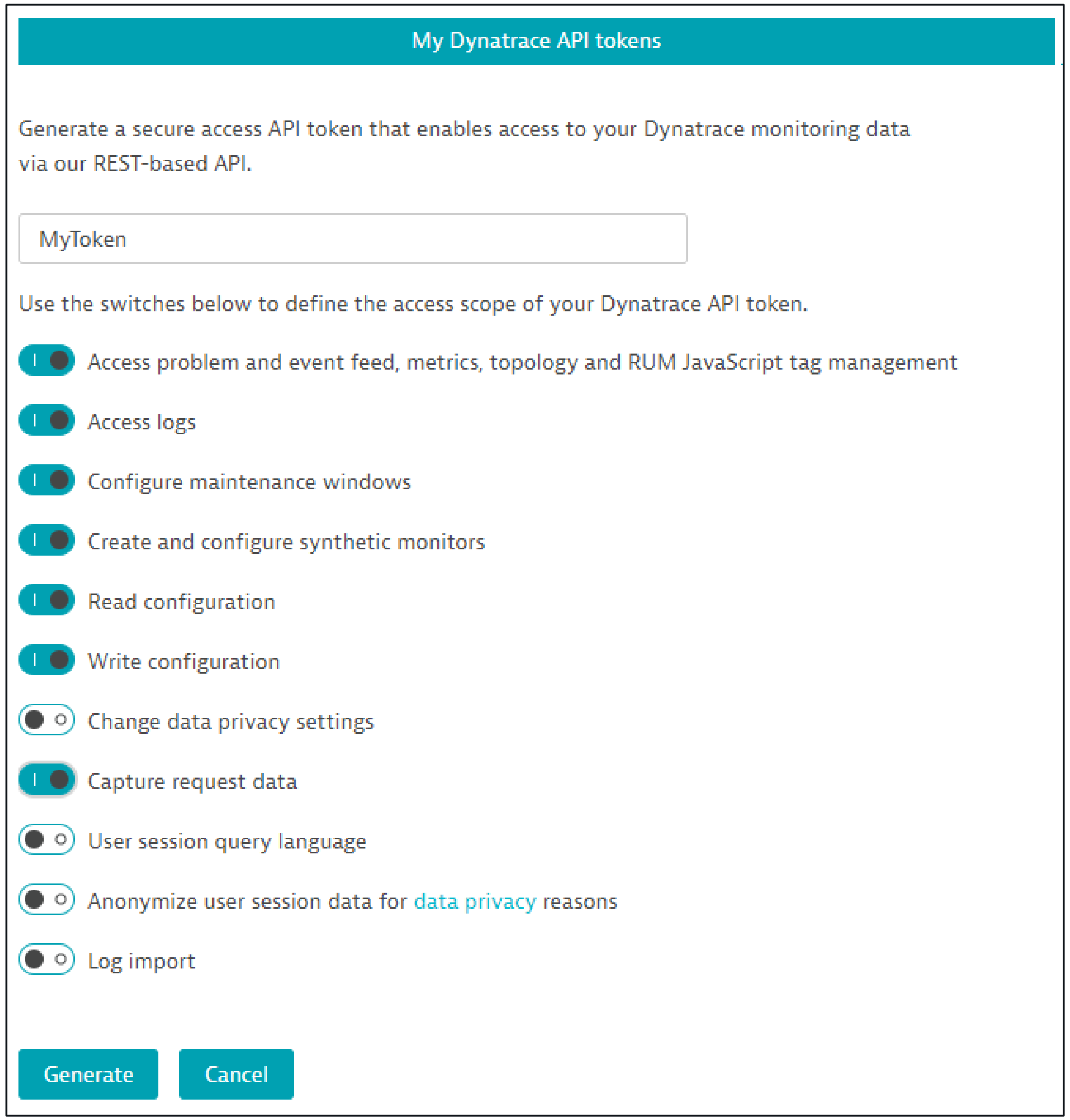
- In your Dynatrace tenant navigate to:
  - Settings > Integration > Dynatrace API
  - Click "Dynatrace API Explorer"


Dynatrace API

You can use our API to export Dynatrace monitoring data into your 3rd party reporting and analysis tools. Multiple API tokens can be created for different purposes.
Use the Dynatrace API Explorer or read the API documentation for use-cases and examples.

Create API Token to use the Config API

- In the Dynatrace API screen, click “Generate token”
- Enter a name for your token (ex. MyToken)
- Make sure to enable the following switches
- Click the “Generate” button



My Dynatrace API tokens

Generate a secure access API token that enables access to your Dynatrace monitoring data via our REST-based API.

MyToken

Use the switches below to define the access scope of your Dynatrace API token.

- [✓] Access problem and event feed, metrics, topology and RUM JavaScript tag management
- [✓] Access logs
- [✓] Configure maintenance windows
- [✓] Create and configure synthetic monitors
- [✓] Read configuration
- [✓] Write configuration
- [ ] Change data privacy settings
- [✓] Capture request data
- [ ] User session query language
- [ ] Anonymize user session data for data privacy reasons
- [ ] Log import

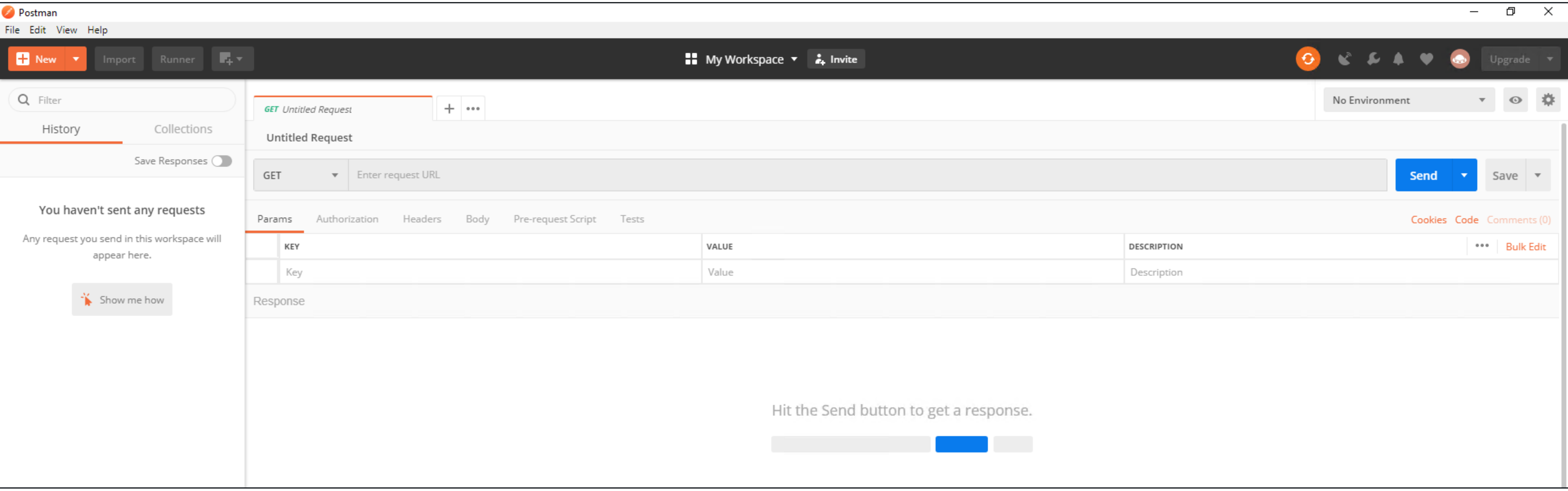
dynatrace
Perform

Using the ConfigAPI via Postman

- We will be using Postman to run a series of API requests to the ConfigAPI in order to do the following:
  - Create Auto Tagging rules
  - Create Management Zones
  - Application setup (URL Grouping and Framework support)
  - Create Custom Services
  - Create Request Attributes
  - Create Synthetics Monitors
- You can use any API client to interact with any of the Dynatrace APIs, however, for this course we are using Postman because it is simple to share a collection of API requests.

Setup Postman

- If this is the first time you are using Postman, after you download and install, you will need to create a free account.
  - You can press “setup later”/”skip” on the Preferences and Team screens



You haven't sent any requests

Any request you send in this workspace will appear here.

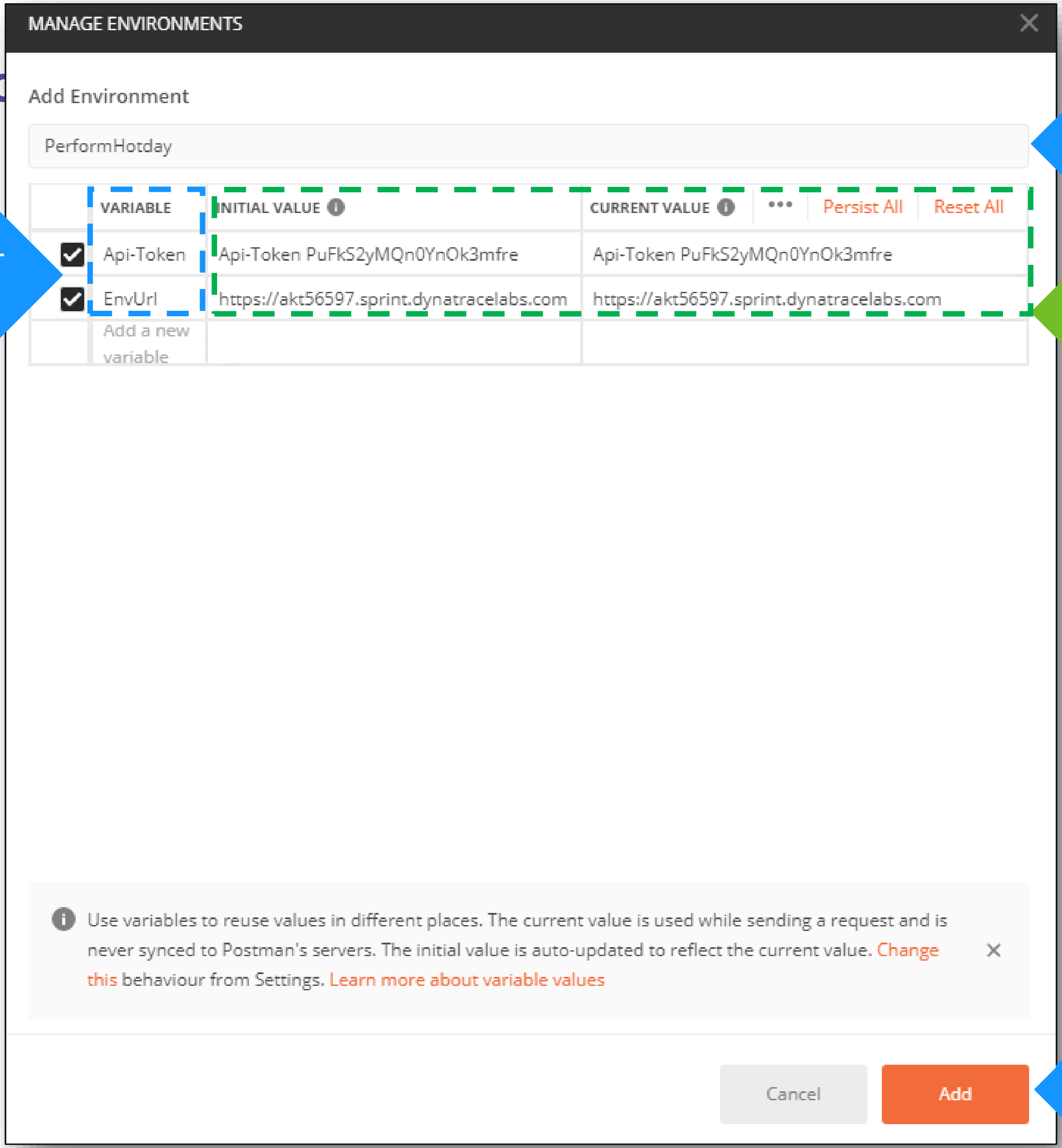
Hit the Send button to get a response.

dynatrace
Perform

Setup Postman (co

Create an Environm
• Click the icon

Create 2 Variables EXACTLY as they appear here



MANAGE ENVIRONMENTS

Add Environment

PerformHotday

	VARIABLE	INITIAL VALUE	CURRENT VALUE
✓	Api-Token	Api-Token PuFkS2yMQn0YnOk3mfre	Api-Token PuFkS2yMQn0YnOk3mfre
✓	EnvUrl	https://akt56597.sprint.dynatracelabs.com	https://akt56597.sprint.dynatracelabs.com
	Add a new variable		

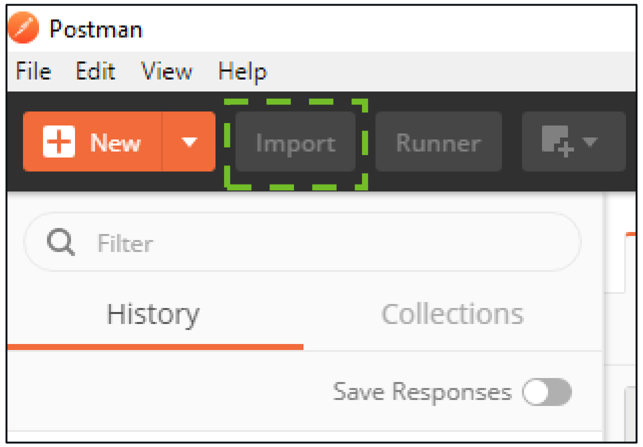
Use variables to reuse values in different places. The current value is used while sending a request and is never synced to Postman's servers. The initial value is auto-updated to reflect the current value. Change this behaviour from Settings. Learn more about variable values

Enter a name for your Environment

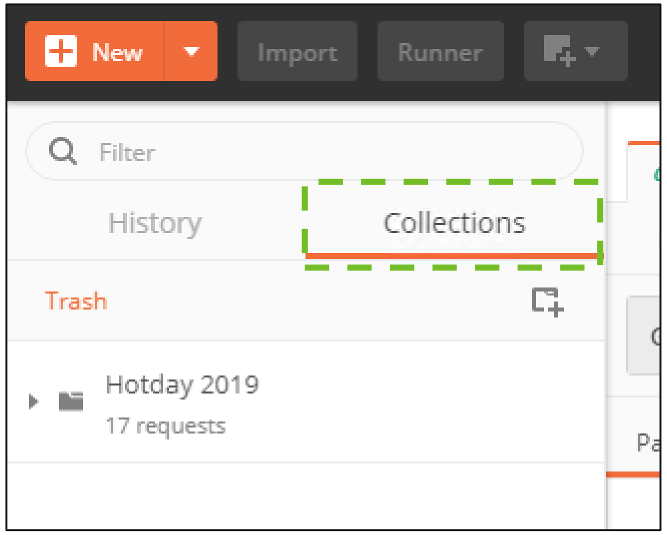
Populate the values with your Full Tenant URL and API-Token

Click "Add" to create the Variables

Importing the Config API Collection

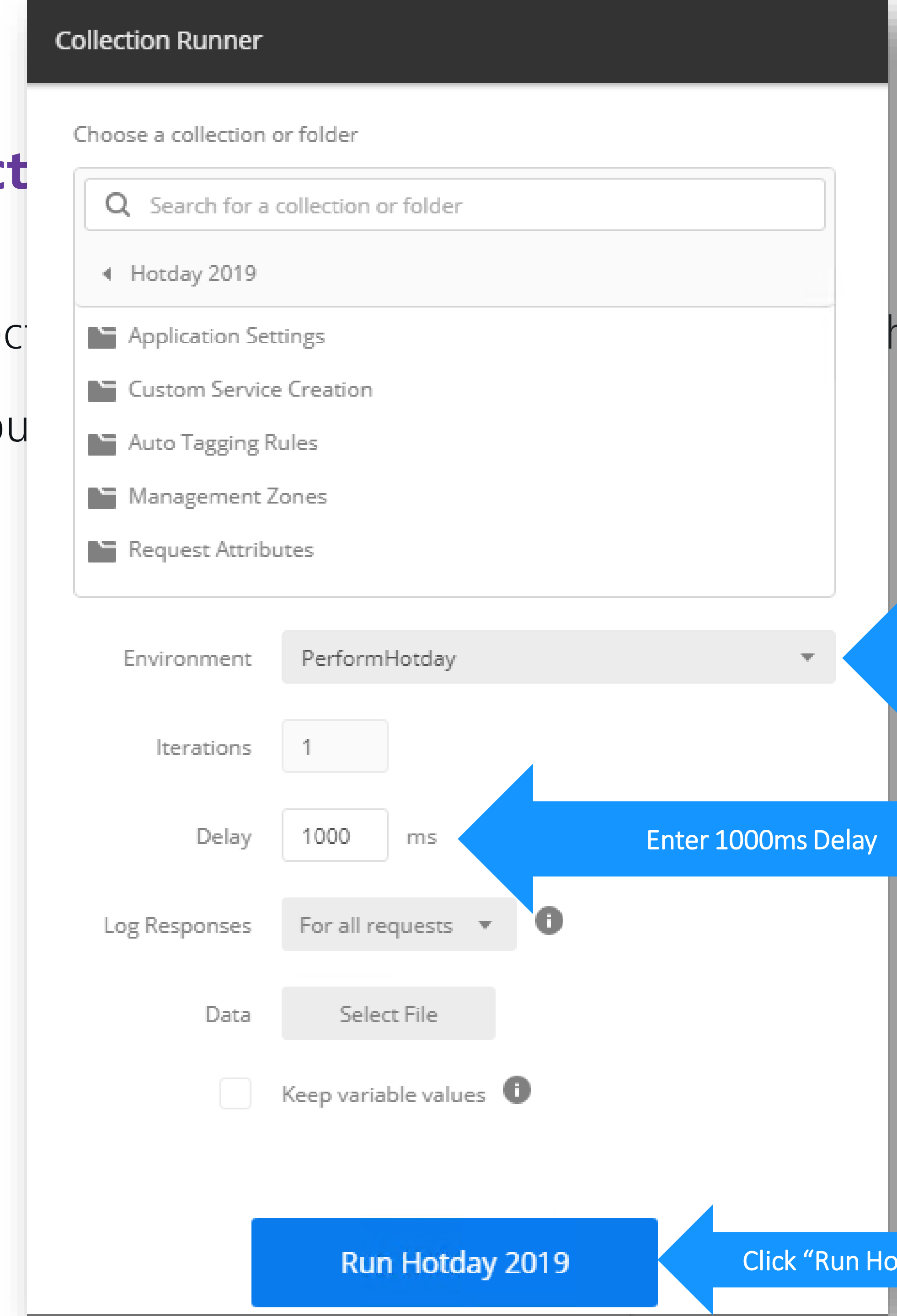


- Click the "Import" button in the upper left corner
- In the Import dialog box, click "Choose Files"
- Browse to the location where you downloaded the Github repository, and select the file: "Hotday 2019.postman_collection.json"
- After the file is uploaded, select "Collections" in the upper left corner and you should now see the collect that was just imported titled "Hotday 2019"



Running the API Collect

- In the “Hotday 2019” collec he popout menu
- The Collection Runner shou



Collection Runner

Choose a collection or folder

Search for a collection or folder

Hotday 2019

- Application Settings
- Custom Service Creation
- Auto Tagging Rules
- Management Zones
- Request Attributes

Environment	PerformHotday
Iterations	1
Delay	1000 ms
Log Responses	For all requests
Data	Select File
	Keep variable values

Run Hotday 2019

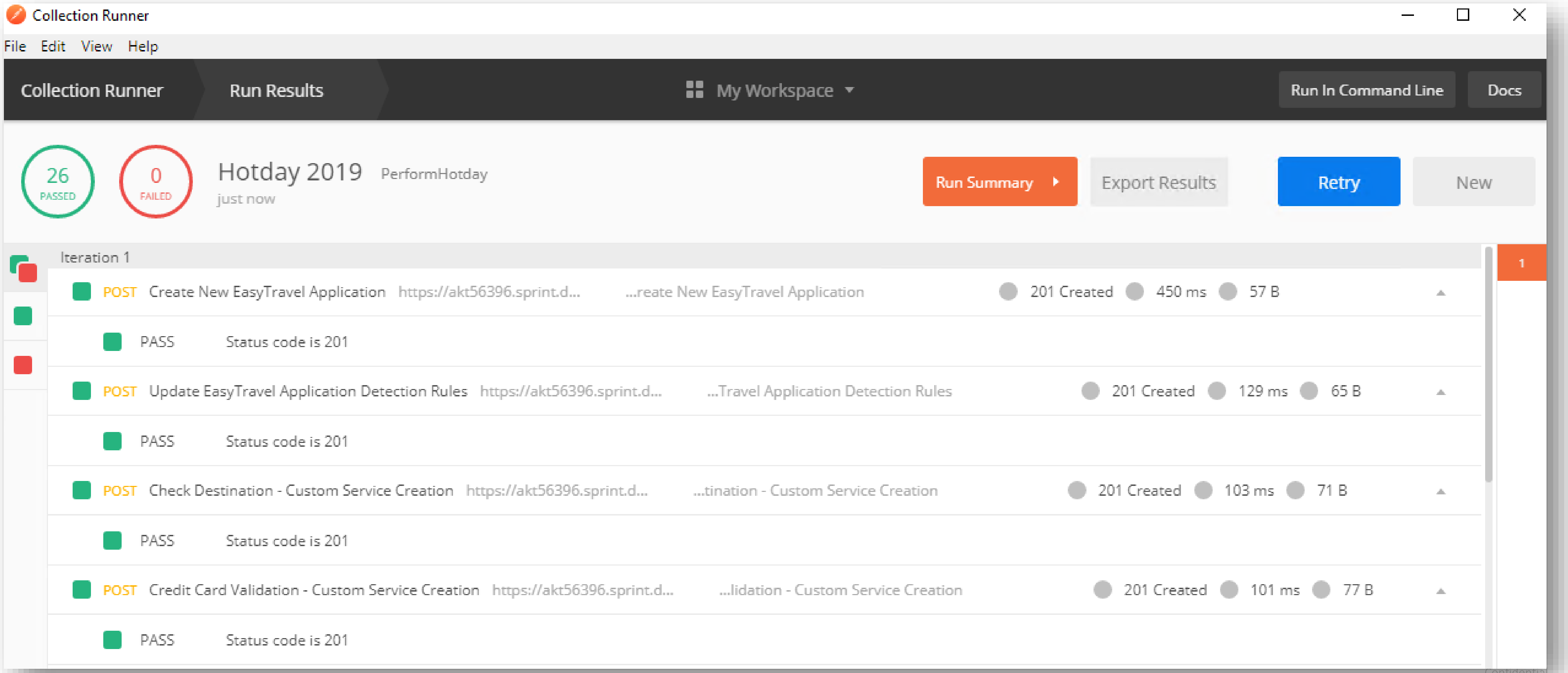
Select the Environment you created from the dropdown

Enter 1000ms Delay

Click “Run Hotday 2019”

Run Validation

- After running the collection, a Run Results window should popup with all requests successful



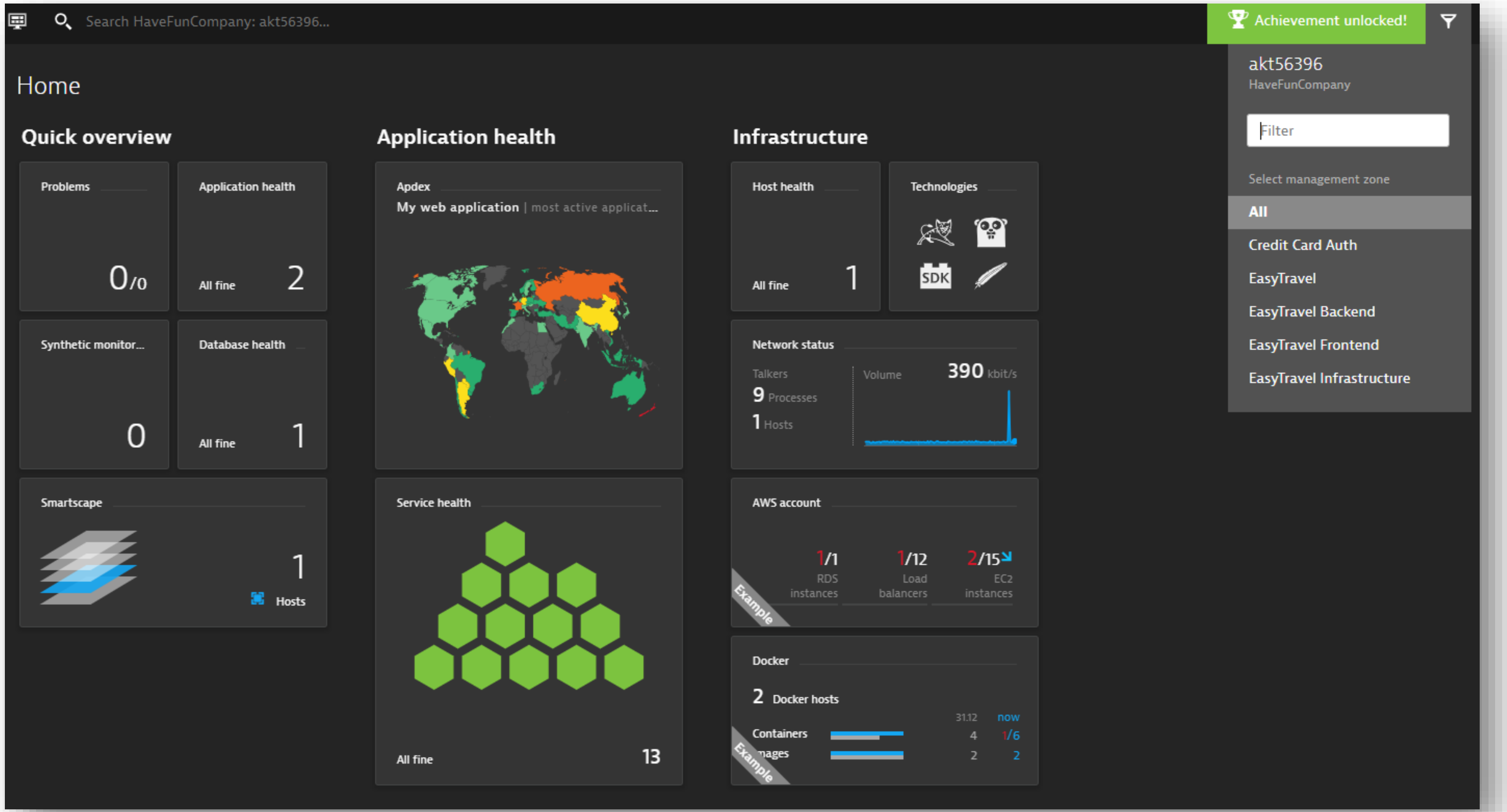
Hotday 2019 PerformHotday
just now

Iteration 1

```
POST  Create New EasyTravel Application  https://akt56396.sprint.d...  ...reate New EasyTravel Application  201 Created  450 ms  57 B
      PASS   Status code is 201
POST  Update EasyTravel Application Detection Rules  https://akt56396.sprint.d...  ...Travel Application Detection Rules  201 Created  129 ms  65 B
      PASS   Status code is 201
POST  Check Destination - Custom Service Creation  https://akt56396.sprint.d...  ...tination - Custom Service Creation  201 Created  103 ms  71 B
      PASS   Status code is 201
POST  Credit Card Validation - Custom Service Creation  https://akt56396.sprint.d...  ...lidation - Custom Service Creation  201 Created  101 ms  77 B
      PASS   Status code is 201
```

Verify the Results

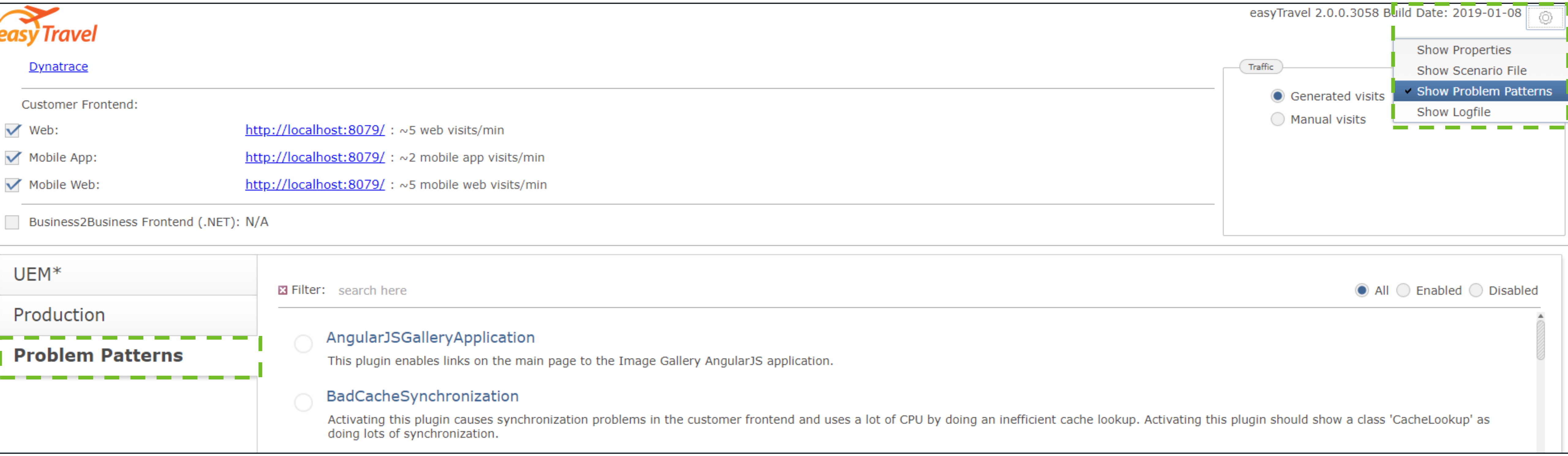
- Back in your Dynatrace tenant, you should now see the results of the Config API (ex. Management Zones, Tags, etc…)



Use Case #1 – CPU Increase

Problem Pattern #1 – CPU Increase

- How to trigger Problem:
  1. Navigate to the EasyTravelConfigUI – This URL was Output as part of the CloudFormation script
  2. Once in the EasyTravelConfigUI, enable “Show Problem Patterns”



easyTravel 2.0.0.3058 Build Date: 2019-01-08

Customer Frontend:

✓ Web:	http://localhost:8079/	: ~5 web visits/min
✓ Mobile App:	http://localhost:8079/	: ~2 mobile app visits/min
✓ Mobile Web:	http://localhost:8079/	: ~5 mobile web visits/min

☐ Business2Business Frontend (.NET): N/A

[● Generated visits] ○ Manual visits

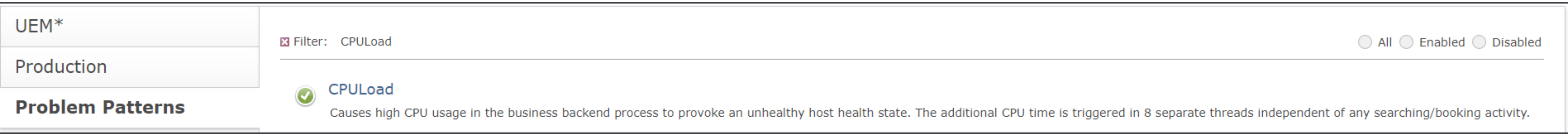
Filter: search here [● All] ○ Enabled ○ Disabled

AngularJSGalleryApplication
This plugin enables links on the main page to the Image Gallery AngularJS application.

BadCacheSynchronization
Activating this plugin causes synchronization problems in the customer frontend and uses a lot of CPU by doing an inefficient cache lookup. Activating this plugin should show a class 'CacheLookup' as doing lots of synchronization.

Problem Pattern #1 – CPU Increase (cont.)

- How to trigger Problem:
  1. In the Filter bar search for the Problem Pattern “CPULoad”
  2. Click the “CPULoad” Problem Pattern to trigger the problem



CPULoad
Causes high CPU usage in the business backend process to provoke an unhealthy host health state. The additional CPU time is triggered in 8 separate threads independent of any searching/booking activity.

Use Case #1 – CPU Increase

Background

EasyTravel is the main revenue generating application for booking travel. After a recent deployment to one of the backend components, your team notices that CPU usage is much higher than usual. This of course could impact users attempting to book travel on our site.

1. Investigate and figure out why this is happening
2. How would you instruct the Dev team to go about finding the problem and triaging it?

Use Case #1 – CPU Increase (cont.)

How to investigate the problem:

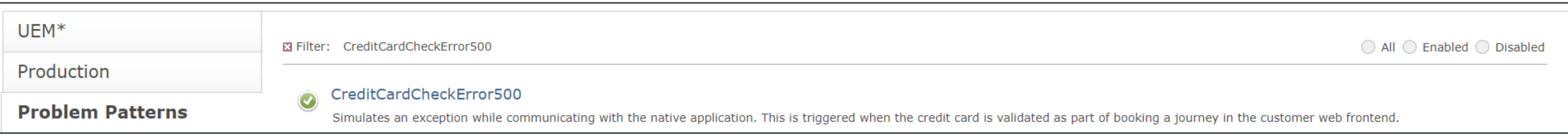
- Navigate to Diagnostic Tools and select the CPU Profiler (filter by Management Zone or Tag)
- View the Service with the Highest CPU consumption
- Navigate to the code level breakdown to determine what piece of code is causing increase in CPU consumption

Use Case #2 – Booking Failure

dynatrace
Perform

Problem Pattern #2 – Booking Failure

- How to trigger Problem:
  1. In the Filter bar search for the Problem Pattern “CreditCardCheckError500”
  2. Click the “CreditCardCheckError500” Problem Pattern to trigger the problem



Filter: CreditCardCheckError500

CreditCardCheckError500

Simulates an exception while communicating with the native application. This is triggered when the credit card is validated as part of booking a journey in the customer web frontend.

dynatrace
Perform

Use Case #2 – Booking Failure

Background

The Marketing Department just called in a panic that bookings dropped to near zero over the past few minutes! Obviously, this is a pretty big issue because the core business is being impacted.

1. Investigate and figure out why this is happening
2. How would you instruct the Dev team to go about finding the problem and triaging it?

Use Case #2 – Booking Failure (cont.)

How to investigate the problem:

- Navigate to Diagnostic Tools and select the Exception Analysis (filter by Management Zone or Tag)

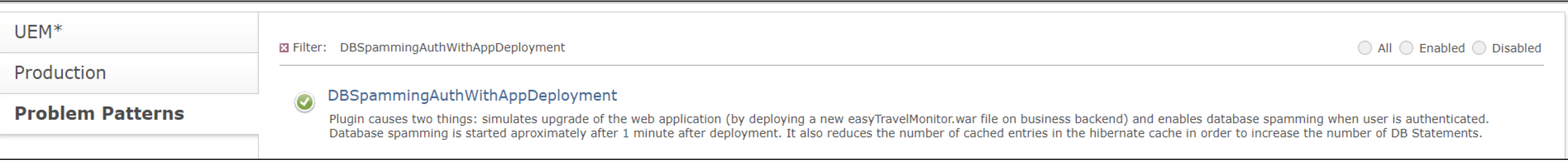
- Determine the exception message that is “most likely” the culprit

- Determine which Service is causing this Exception and what should be done

Use Case #3 – Slow User Login

Problem Pattern #3 – Slow User Login

- How to trigger Problem:
  1. In the Filter bar search for the Problem Pattern "DBSpammingAuthWithAppDeployment"
  2. Click the "DBSpammingAuthWithAppDeployment" Problem Pattern to trigger the problem



DBSpammingAuthWithAppDeployment

DBSpammingAuthWithAppDeployment

Plugin causes two things: simulates upgrade of the web application (by deploying a new easyTravelMonitor.war file on business backend) and enables database spamming when user is authenticated. Database spamming is started aproximately after 1 minute after deployment. It also reduces the number of cached entries in the hibernate cache in order to increase the number of DB Statements.

Use Case #3 – Slow User Login

Background

After a recent deployment to a component of EasyTravel customers are beginning to call the helpdesk and complain that logging in to view/book travel is “slow”.

1. Investigate and figure out why this is happening
2. How would you instruct the Dev team to go about finding the problem and triaging it?

Use Case #3 – Slow User Login (cont.)

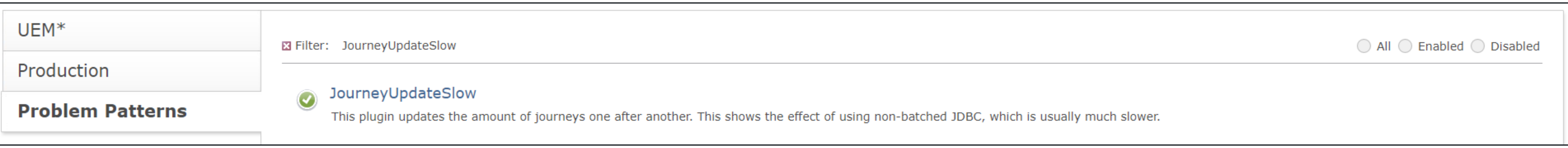
How to investigate the problem:

- Navigate to Diagnostic Tools and Select Top Database Statements
- Determine if there are any slow running queries are impacts from the recent change
- Determine what Service is responsible for this
- Is this a Database Slowdown issue?

Use Case #4 – Homepage Slowdown

Problem Pattern #4 – Homepage Slowdown

- How to trigger Problem:
  1. In the Filter bar search for the Problem Pattern "JourneyUpdateSlow"
  2. Click the "JourneyUpdateSlow" Problem Pattern to trigger the problem



Filter: JourneyUpdateSlow

JourneyUpdateSlow
This plugin updates the amount of journeys one after another. This shows the effect of using non-batched JDBC, which is usually much slower.

Use Case #4 – Homepage Slowdown

Background

The Operations team is claiming that the ‘Special-Offers.jsp’ and ‘CalculateRecommendations’ pages are experiencing a significant increased response time. VP of Operations is not happy. Your job is to find out what’s going on with these pages.   Some other pages may also be affected.

1. Investigate and figure out why this is happening
2. How would you instruct the Dev team to go about finding the problem and triaging it?

Use Case #4 – Homepage Slowdown (cont.)

How to investigate the problem:

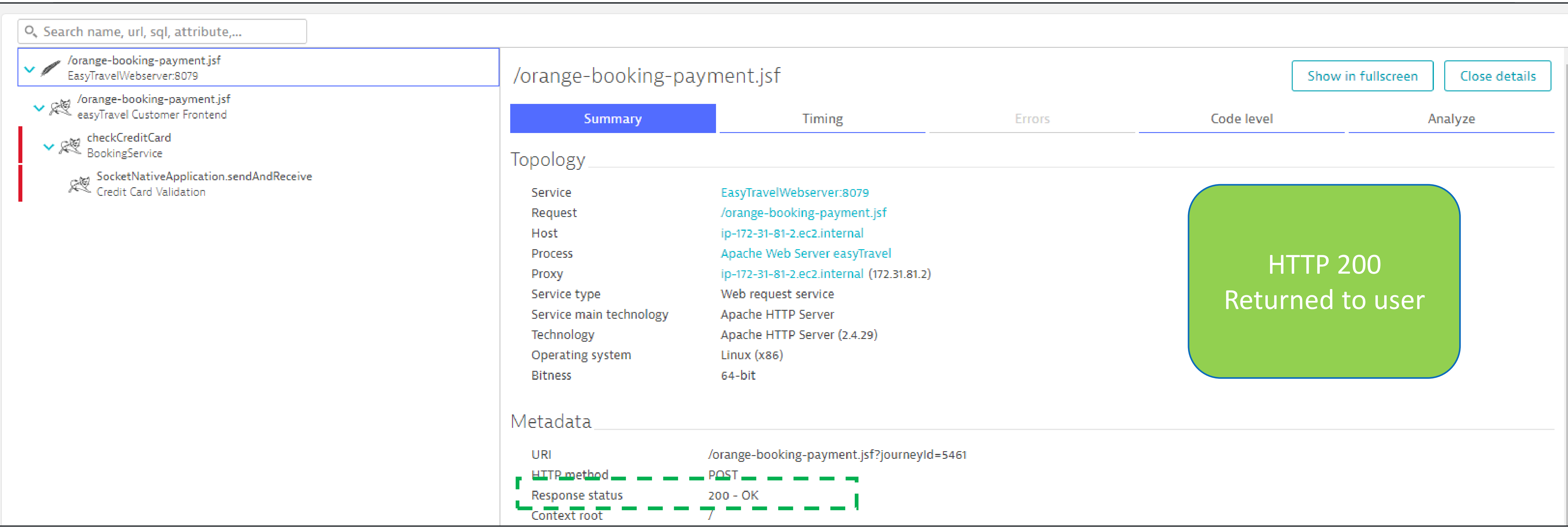
- In Diagnostic Tools, view the Top Web Requests and search for Special-Offers.jsp and CalculateRecommendations Web Requests
- View the Service Flow and look for the highest % of contribution
- Determine if this is a Database issue or an Application issue or both

Tips & Tricks

dynatrace **Perform**

Returning HTTP 200 on Failure

- Try to avoid returning a HTTP 200 to the user, when the transaction actually fails:



Search name, url, sql, attribute,...

/orange-booking-payment.jsf
EasyTravelWebserver:8079

/orange-booking-payment.jsf
easyTravel Customer Frontend

checkCreditCard
BookingService

SocketNativeApplication.sendAndReceive
Credit Card Validation

/orange-booking-payment.jsf

Show in fullscreen | Close details

Summary | Timing | Errors | Code level | Analyze

Topology

Service	EasyTravelWebserver:8079
Request	/orange-booking-payment.jsf
Host	ip-172-31-81-2.ec2.internal
Process	Apache Web Server easyTravel
Proxy	ip-172-31-81-2.ec2.internal (172.31.81.2)
Service type	Web request service
Service main technology	Apache HTTP Server
Technology	Apache HTTP Server (2.4.29)
Operating system	Linux (x86)
Bitness	64-bit

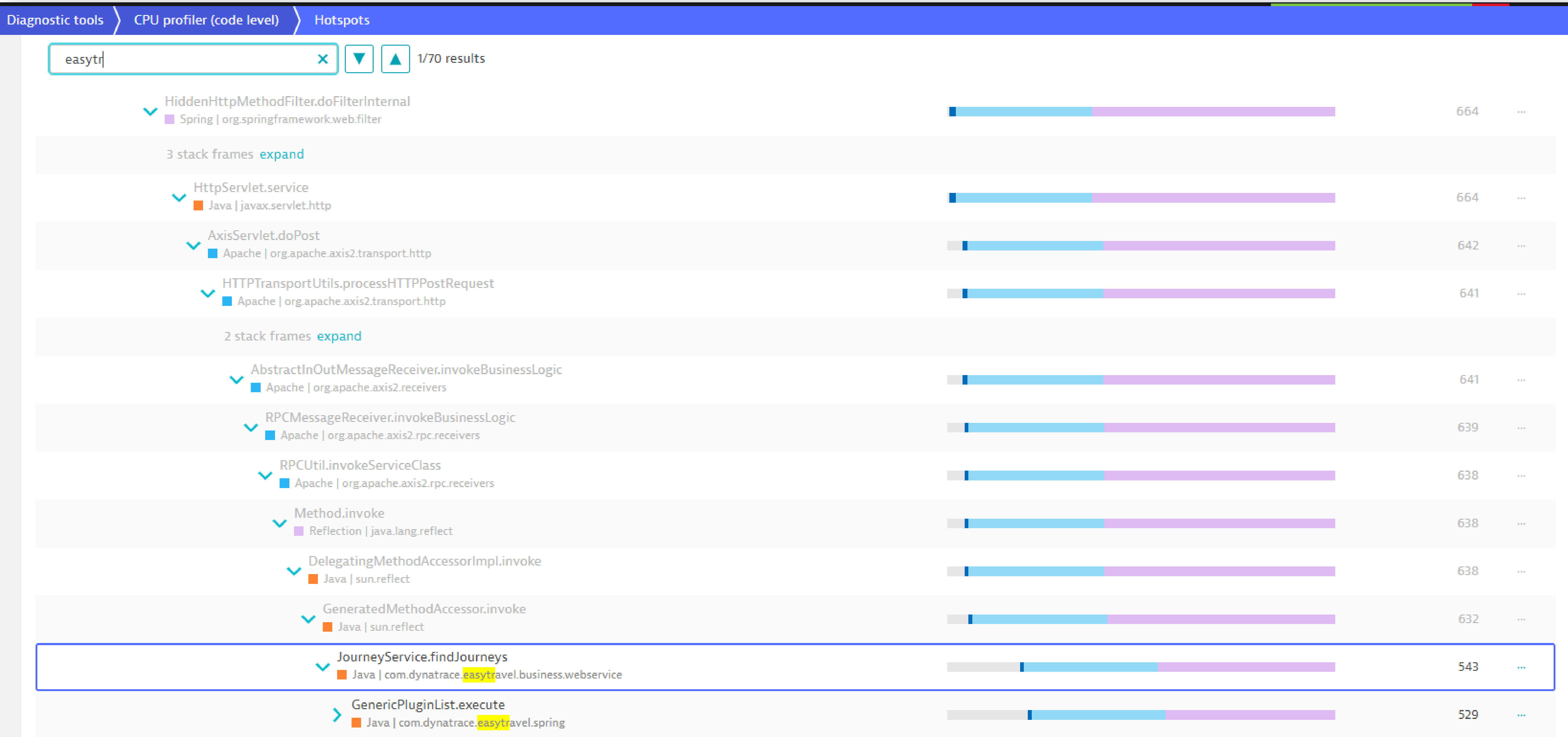
HTTP 200
Returned to user

Metadata

URI	/orange-booking-payment.jsf?journeyId=5461
HTTP method	POST
Response status	200 - OK
Context root	/

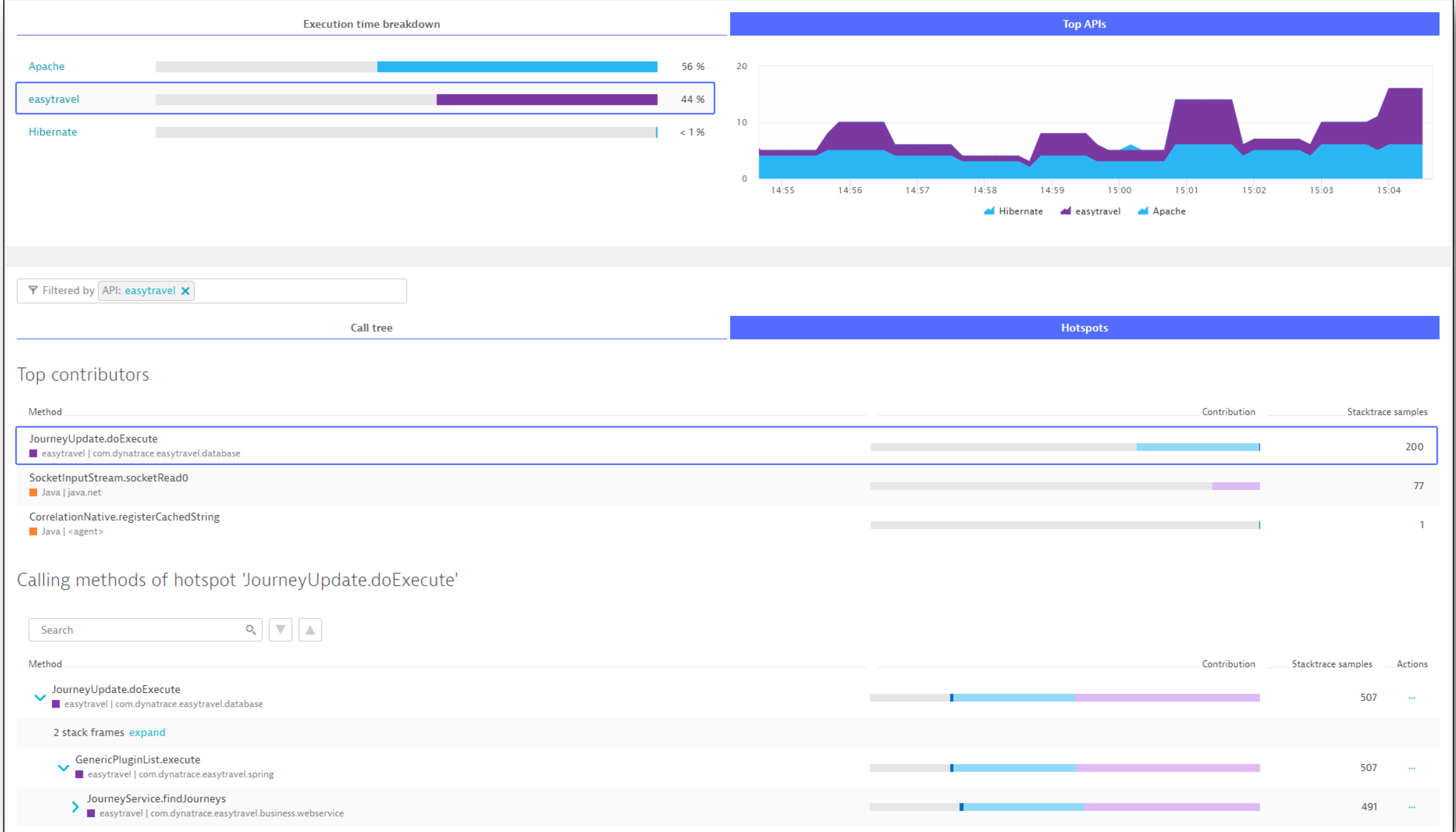
dynatrace Perform

Finding your code in the CPU Profiler


Diagnostic tools 〉 CPU profiler (code level) 〉 Hotspots

easytr ✕ ▼ ▲ 1/70 results

Method	Details	Value
HiddenHttpMethodFilter.doFilterInternal	Spring | org.springframework.web.filter	664
3 stack frames expand		
HttpServlet.service	Java | javax.servlet.http	664
AxisServlet.doPost	Apache | org.apache.axis2.transport.http	642
HTTPTransportUtils.processHTTPPostRequest	Apache | org.apache.axis2.transport.http	641
2 stack frames expand		
AbstractInOutMessageReceiver.invokeBusinessLogic	Apache | org.apache.axis2.receivers	641
RPCMessageReceiver.invokeBusinessLogic	Apache | org.apache.axis2.rpc.receivers	639
RPCUtil.invokeServiceClass	Apache | org.apache.axis2.rpc.receivers	638
Method.invoke	Reflection | java.lang.reflect	638
DelegatingMethodAccessorImpl.invoke	Java | sun.reflect	638
GeneratedMethodAccessor.invoke	Java | sun.reflect	632
JourneyService.findJourneys	Java | com.dynatrace.easytravel.business.webservice	543
GenericPluginList.execute	Java | com.dynatrace.easytravel.spring	529



Execution time breakdown | **Top APIs**

Apache		56 %
easytravel		44 %
Hibernate		< 1 %

Filtered by API: easytravel

Call tree | **Hotspots**

Top contributors

Method	Contribution	Stacktrace samples
JourneyUpdate.doExecute easytravel | com.dynatrace.easytravel.database		200
SocketInputStream.socketRead0 Java | java.net		77
CorrelationNative.registerCachedString Java | <agent>		1

Calling methods of hotspot 'JourneyUpdate.doExecute'

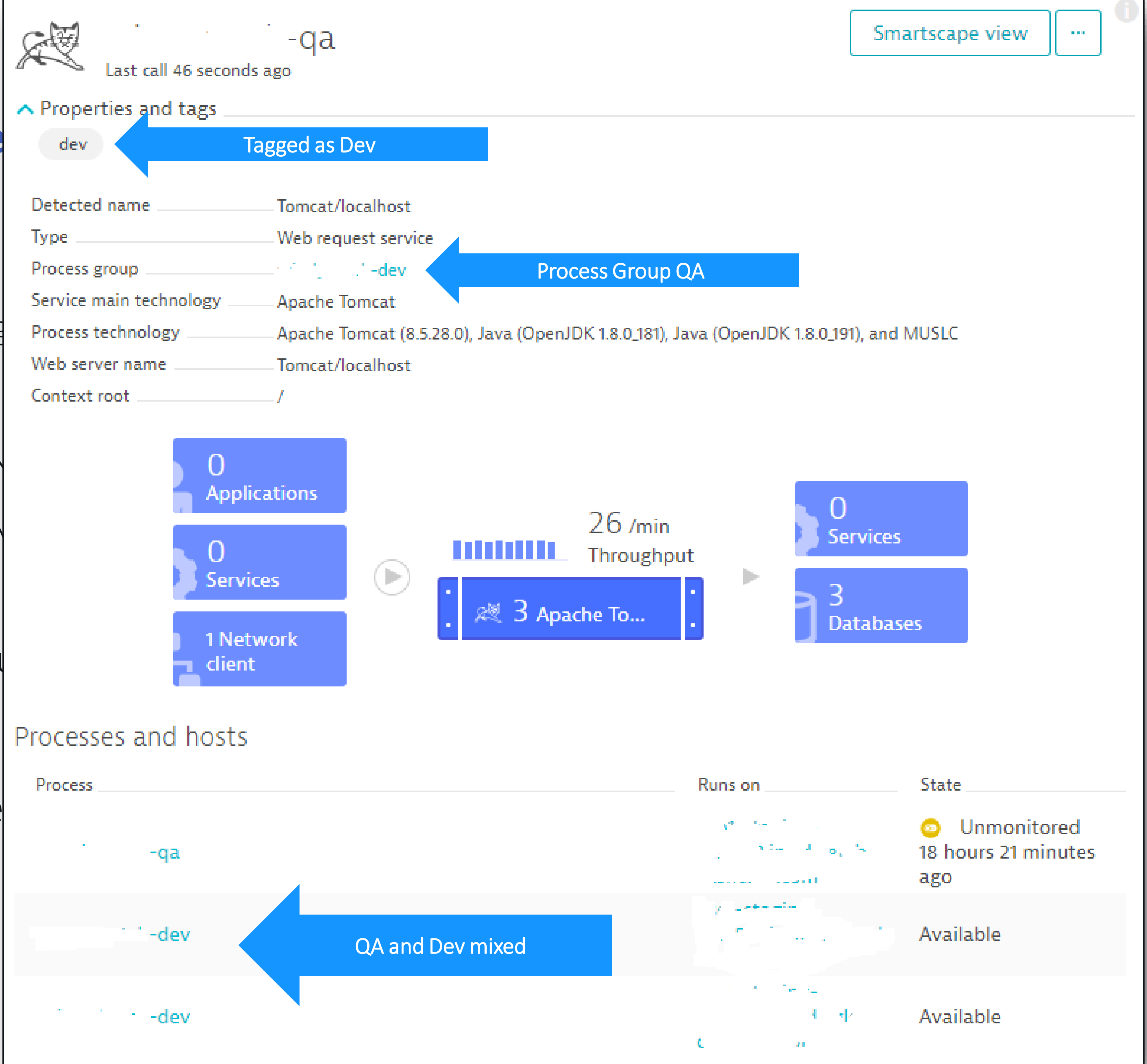
Search

Method	Contribution	Stacktrace samples	Actions
JourneyUpdate.doExecute easytravel | com.dynatrace.easytravel.database		507	···
2 stack frames expand			
GenericPluginList.execute easytravel | com.dynatrace.easytravel.spring		507	···
JourneyService.findJourneys easytravel | com.dynatrace.easytravel.business.webservice		491	···

dynatrace
Perform

Properly Se

- Get Process
  - Including
- Services CAI
  - Service N
- Process Grou
- Tagging rule



-qa
Last call 46 seconds ago

Smartscape view | ···

Properties and tags

dev ← Tagged as Dev

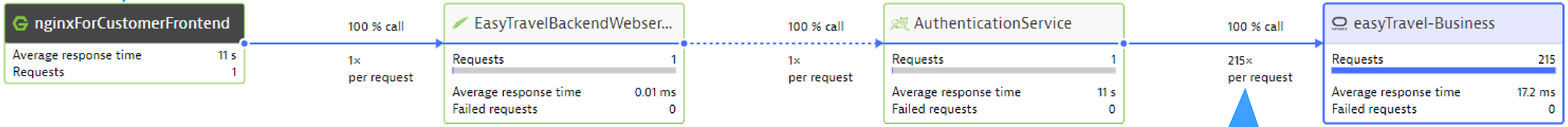
Detected name	Tomcat/localhost
Type	Web request service
Process group	-dev ← Process Group QA
Service main technology	Apache Tomcat
Process technology	Apache Tomcat (8.5.28.0), Java (OpenJDK 1.8.0_181), Java (OpenJDK 1.8.0_191), and MUSLC
Web server name	Tomcat/localhost
Context root	/

Processes and hosts

Process	Runs on	State
-qa		Unmonitored 18 hours 21 minutes ago
-dev ← QA and Dev mixed		Available
-dev		Available

DB Spamming

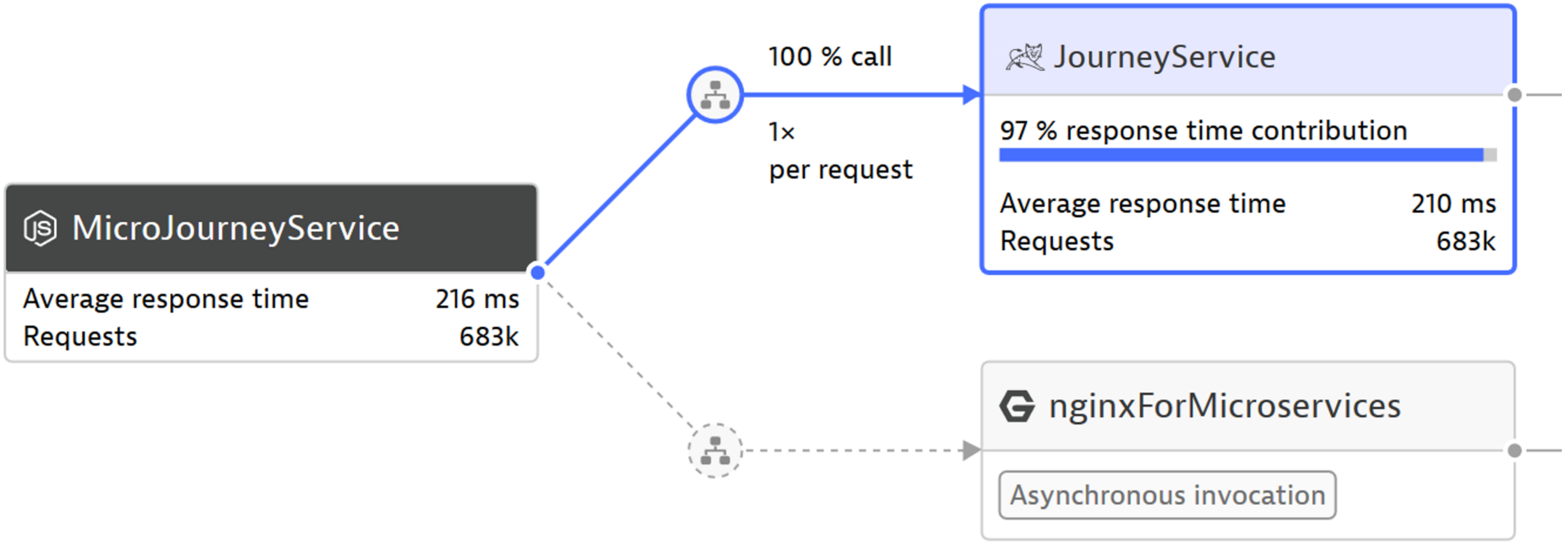
Total Transaction took 11s, for a single request



Single request spawns 215 DB queries!

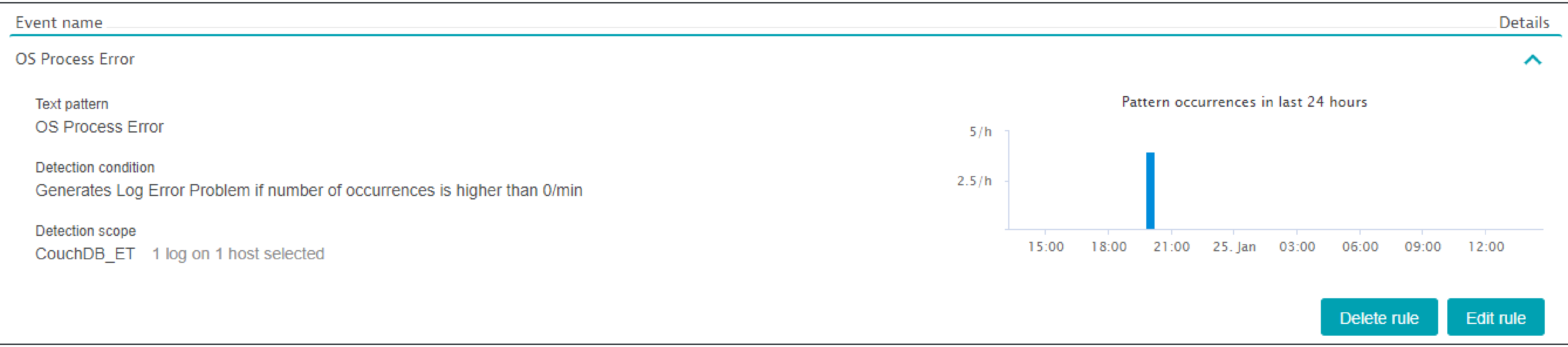
dynatrace Perform

Microservices – Considerations for tightly coupled services



Don’t forget about the Logs

- Using Log Event Detection can help find application errors
  - Common Use Cases:
    - Detect Application Errors in Native Processes
    - Detect Errors with the Application Infrastructure (syslog, Windows Event Logs, etc…)
    - Detect Errors in CloudTrail logs


Event name — Details

OS Process Error

Text pattern
OS Process Error

Detection condition
Generates Log Error Problem if number of occurrences is higher than 0/min

Detection scope
CouchDB_ET 1 log on 1 host selected

Remember to Delete your AWS Resources!!!

Q & A

Thank you

dynatrace Perform

HOT Day sponsored by aws