

# OpenPipeline, Segments, and IAM Policies

---

SE Training – January 2025



# Training Recorded

---

\*Warning\* This training is being recorded and will be available for future on-demand consumption. Please maintain professionalism at all times. Thank you.

# Agenda

---

- Goal: To introduce the paradigm shift from classic Management Zones to data permissions and filtering on Grail, we'll configure OpenPipeline, Segments, and IAM Policies to provide role-based access and analysis controls
- Segments 101
- Paradigm Shift – Management Zones vs Segments + IAM
  - Common Scenarios and Solutions
- Hands On: Segments
  - Platform Engineering, Internal Development Platform, Self-Service Onboarding
    - Analyzing Data without Segments
  - Basic Segments
    - Kubernetes Cluster and Namespace
  - OpenPipeline
  - Complex Segments
    - Project Team
- Segments Roadmap
  - Official Roadmap
  - Internal RFE (When is XYZ coming?)



# Segments 101

---

# Why Segments?

Settings &gt; Preferences &gt; Management zones

## Settings

Monitoring  
Setup and overviewCloud Automation  
Setup and configurationProcesses and containers  
Monitoring, detection and namingWeb and mobile monitoring  
Real user and synthetic monitoringCloud and virtualization  
Connect cloud and virtualization typesServer-side service monitoring  
Manage and customize service monitoringService Detection  
Define rules for services and spansLog Monitoring  
Set up management of logsAnomaly detection  
Configure detection sensitivityAlerting  
Configure alerting settingsDashboards  
Configure dashboard settings

Metrics

### Management zones settings

Management zones enable defining fine grained access rights to parts of an environment. A

Management zone consists of a set of entities like applications, hosts, process groups, or services.

[More...](#)

 Your user does not have the necessary write permissions.

### Page Unresponsive

You can wait for it to become responsive or exit the page.

 Management zones settings - Environment - Settings - Allianz ASI...

[Wait](#)[Exit page](#)

# Why Segments?

Smartscape topology > Services

Search Allianz ASIC: preprod...

mz-az-at 44

Applications 35

Services 3/2641

Processes 9583

Hosts 215

Data centers 12

Experience the full power of Grail and the Dynatrace Query Language

Where MZ?

Dynatrace

Logs and events

Logs and events Powered by Grail™

Explore your log data and Kubernetes events in simple mode. For deeper analysis, or to query business events, switch to advanced mode powered by DQL.

Advanced mode

Filter by

Run query

Search attributes...

Search results 1 Notification Execution time: 2 s Scanned data: 423 MB

Open with... Create processing rule Create metric Format table Actions

timestamp	status	content
2023-11-16 13:21:50.401	NONE	SNMP trap (CISCO-SMI::ciscoMgmt.41.2.0.1) reported from 10.69.0.18
2023-11-16 13:21:50.401	NONE	SNMP trap (CISCO-SMI::ciscoMgmt.41.2.0.1) reported from 10.69.0.19
2023-11-16 13:21:45.191	NONE	{"timestamp": "2023-11-16T12:21:45.1919765Z", "message": "Update loop too k 0.009809627s", "log_level": "info", "source": "cc.deployment_updater.sched"}

# Segments: Preconfigured Dynamic Filtering

The screenshot shows a dark-themed Kubernetes monitoring interface. At the top left is a sidebar with icons for Clusters, Nodes, Namespaces, Workloads (with sub-options like Deployments, DaemonSets, StatefulSets, ReplicaSets, Replication controllers, CronJobs, Jobs, Pods, Services, and Containers), and a gear icon for configuration.

The main header includes a cluster selector dropdown set to "Cluster: CL-Prod20" (highlighted with a red box), a search bar, and filter options. A prominent green banner on the right says "1 click to block out all the noise".

The central area displays a "Clusters" table with 12 records. The table has columns for Cluster, Problems, Nodes, Namespaces, Workloads, Pods, CPU Usage, CPU Requests, and CPL. Each row includes a small icon and a red diamond icon indicating specific status or alerts. The table also features horizontal and vertical scroll bars.

Cluster	Problems	Nodes	Namespaces	Workloads	Pods	CPU Usage	CPU Requests	CPL
code-matrix-cluster	2	12	8	789	1.2k	<div style="width: 120px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>
quantum-compute-pod	1	1 / 2	5	12	78	<div style="width: 78px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>
binary-nexus	-	30	12	13 / 568	234	<div style="width: 234px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>
devops-grid	-	1	5	456	48	<div style="width: 48px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>
cyberspace-core	-	2 / 500	23	1.3k	89	<div style="width: 89px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>
robo-mesh-cluster	-	4	3	8 / 90	5	<div style="width: 5px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>
nano-dene-forge	-	12	1	4 / 45	67	<div style="width: 67px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>
crypto-mesh-hub	-	1 / 45	24	456	124	<div style="width: 124px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>
data-pipeline-grid	-	5	4	545	78	<div style="width: 78px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>
quantum-logic-nest	-	1	3	3	3	<div style="width: 3px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>
technosphere-cluster	-	4	3	3 / 90	5	<div style="width: 5px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>	<div style="width: 10px; height: 10px;"></div>

# Segments Implementation

Trigger

## Trigger

We offer two predefined variants:

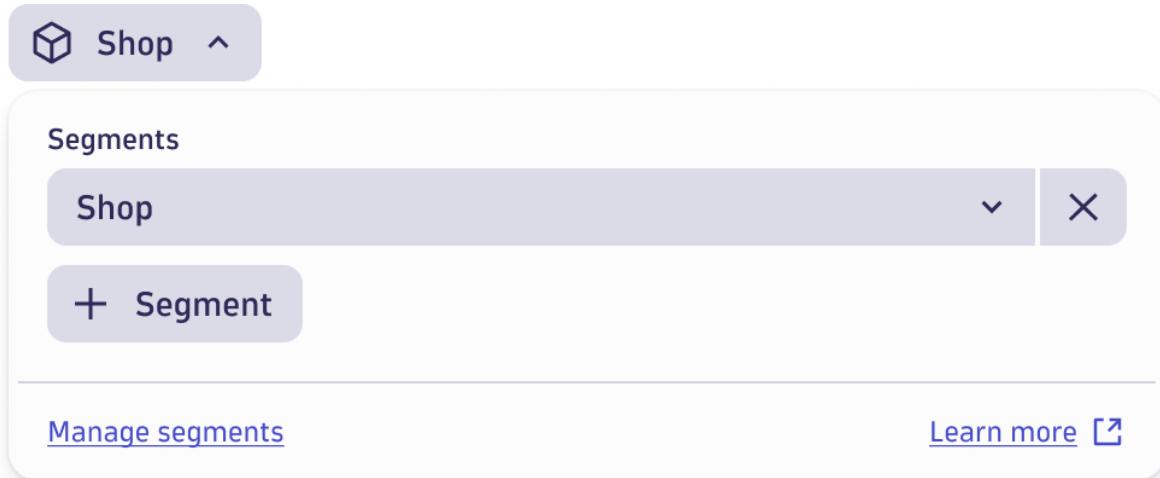
- Default, where we show the currently applied segment if one segment is selected. Multiple segments are grouped to make the width more predictable.
- Compact, where we only show the amount of selected segments.

Additionally, the trigger is customizable, as for all our overlays.

	No segment	Simple segment	Segment with variable (1 value)	Segment with variable (2+ values)	2+ segments
<b>Default trigger</b>	 ▾	 Shop ▾	 Region: eu-ireland ▾	 Region (2) ▾	 2 segments ▾
<b>Compact trigger</b>	 ▾	 1 ▾	 1 ▾	 1 ▾	 2 ▾

# Simple Segment

---



Show me entities, logs, events, etc. of my application

- Equivalent to single MZ

# Dynamic Segment

The screenshot shows a user interface for managing segments. At the top, there is a header with a cube icon and the text "Region: eu-ireland". Below this is a "Segments" section with a "Region" dropdown set to "eu-ireland, eu-austria". A search bar is present above a list of regions. The list includes "eu-ireland" and "eu-austria" with checked checkboxes, and other regions like "eu-france", "eu-germany", "eu-italy", "eu-spain", "us-east", and "us-west" with unchecked checkboxes. There is also a "+ Segment" button and a "Manage segments" link.

Region
eu-ireland
eu-austria
eu-france
eu-germany
eu-italy
eu-spain
us-east
us-west

Show me data of “eu-ireland” and “eu-austria” Regions

- Deduplication of MZs
- Dynamic conditions using variables
- Multi-select of Dynamic Segment values will result in OR logic when filtering
  - Filter region == “eu-Ireland” OR region == “eu-austria”

# Multiple Segments

2 segments ^

Segments

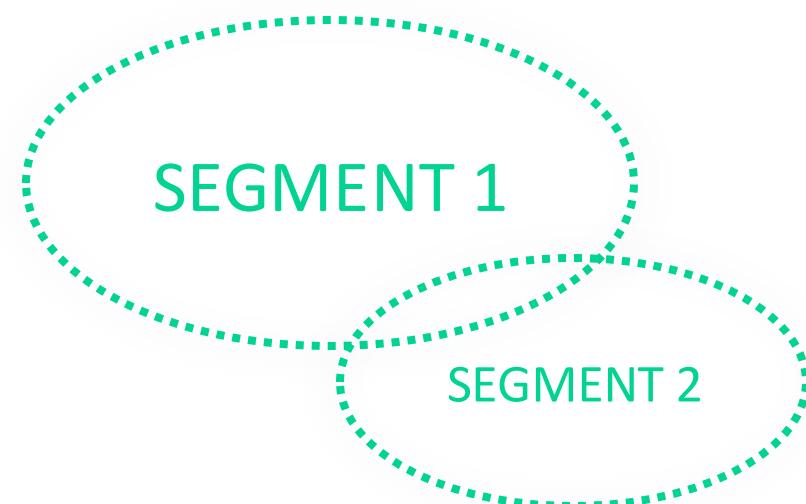
Region	eu-ireland, eu-austria	X
App-env	production	X

+ Segment      Clear all

[Manage segments](#)      [Learn more](#)

Show me data of my application from a specific infrastructure

- Intersection of multiple segments  
 $(\text{eu-ireland OR eu-austria}) \text{ AND production}$



# How it all works

2 segments ^

Segments

Region	eu-ireland, eu-austria	X
App-env	production	X

+ Segment Clear all

[Manage segments](#) [Learn more](#)



- 1) Segment IDs and values are passed to query API  
FiRmGMvWzPY(eu-ireland, eu-austria), HCkccp8aAIU(production)
- 2) Grail filters consumer queries with conditions defined in segments

```
fetch logs
| filter
// permission-based filters
... AND
// Segment-based filters
(
  (condition_a1 OR condition_a2 OR ...) AND
  (condition_b1 OR condition_b2 OR ...)
)
// consumer query continues
| summarize ...
```

Transparent filter conditions

# Building a Segment

< All segments

This segment has unsaved changes.

[Discard changes](#) [Save](#)



**ACE DT**

(Full-stack) Observability data for ACE DT team.

## 2 Variables for dynamic segments

**{\$} Variables** ⓘ

Create variables to apply in your Segment data filters

\$namespace.name      \$namespace.id

dynatrace



Owner

R Roman Windischhofer

Visibility

Anyone in the environment

[Discard changes](#) [Save](#)

1 Visible to anyone or unlisted

## 3 Include data directly

Logs

3 Include data directly

k8s.namespace.name = \$namespace.name

[Edit](#) [Delete](#) [Up](#) [Down](#) [More](#)

[Preview](#)

Metrics

4 Include data of entities

k8s.namespace.name = \$namespace.name

[Edit](#) [Delete](#) [Up](#) [Down](#) [More](#)

[Preview](#)

Hosts

4 Include data of entities

tags = team:ACE

Include  Problems  Vulnerabilities  Metrics  Logs  Events  BizEvents  Spans

[Edit](#) [Delete](#) [Up](#) [Down](#) [More](#)

[Preview](#)

Services

5 Include data of related entities

runs\_on = \$this.hosts

Include  Problems  Vulnerabilities  Metrics  Logs  Events  BizEvents  Spans

[Edit](#) [Delete](#) [Up](#) [Down](#) [More](#)

[Preview](#)

+ All types + Metrics + Logs + Events + BizEvents + Spans + Entities



20 include blocks per segment

1 filter for data of entities (logs, events, spans)

No limit for problems, vulnerabilities, metrics

1 topology traversal step only

# Segments and in-app filters

2 Additional filter for further problem analysis

Region (2) Type to filter Update

Filter by segments  
Region eu-ireland, eu-austria X  
+ Segment

1 Segments selector to set context Manage segments Learn more

24 Nodes 3 / 3 Namespaces 12 Workloads 3.5k / 12.7k

Probabilities Topology 12 columns hidden

Node	Problems	Pods	Workloads	Services	Usage	Requests	alertmanager	apiserver	key	key	key
full-observability...	-	123	3	45	123	123	main	main	✓	-	-
full-observability...	-	12	3	345	12	12	longvaluetextexample...	longvaluetextexample...	-	-	-
GKE CP KLU	-	3	12	3	-	3	main	main	✓	-	-
OCP 3.11	-	12	456	1	12	12	longvaluetextexample...	longvaluetextexample...	✓	-	-
ocp4-12	-	467	24	12	467	467	-	-	longvaluetextexample...	-	-
ocp4-9-in-cluster	-	4	89	3	4	4	longvaluetext	longvaluetextexample...	-	-	-
ocp4-9-kubernetes	-	4	45	54	4	45	main	longvaluetext	-	-	-
rauter-cri-test	-	45	456	34	45	456	✓	✓	longvaluetextexample... main	-	-
trauter-e2e	-	5	545	1,234	5	545	-	-	✓	longvaluetext example...	longvaluetext e
full-observability...	-	123	3	45	123	123	main	main	✓	-	-
full-observability...	-	12	3	345	12	12	longvaluetextexample...	longvaluetextexample...	✓	-	-
GKE CP KLU	-	3	12	3	-	3	main	main	✓	-	-
OCP 3.11	-	12	456	1	12	456	longvaluetextexample...	longvaluetextexample...	✓	-	-
ocp4-12	-	467	24	12	467	467	-	-	longvaluetextexample...	-	-
ocp4-9-in-cluster	-	4	89	3	4	89	longvaluetext	longvaluetextexample...	-	-	-
ocp4-9-kubernetes	-	4	45	54	4	45	main	longvaluetext	-	-	-
rauter-cri-test	-	45	456	34	45	456	✓	✓	longvaluetextexample... main	-	-
trauter-e2e	-	5	545	1,234	5	545	-	-	✓	longvaluetext example...	longvaluetext e
ocp4-12	-	467	24	12	467	467	-	-	longvaluetextexample...	-	-
rauter-e2e	-	1	3	3	1	3	-	-	✓	longvaluetextexample...	-

20 rows per page Page 1 of 2 < >



**1 Additional filter for further problem analysis**

Region (2) Type to filter Last 24 hours Update

Filter by segments Region eu-ireland, eu-austria + Segment Manage segments Learn more

1 Segments selector to set context

2 selected Open 2 columns hidden

ID	Name	Status	Labels	Category	Affected	Root cause	Started	Duration
P-230980	Service slowdown	Active	Ack Warning service09	Slowdown	150	loginService	21/09/2023, 6:10	30 min
P-230981	SLO Burnrate	Active	Ack	Availability	23	easyService	21/09/2023, 5:42	2 min
P-230982	Failure rate increase	Active	Warning	Monitoring unavailable	4	SSO-Service	21/09/2023, 6:22	42 min
P-230983	Support sliding window size	Active	Warning	Custom	12	loginService	21/09/2023, 6:36	56 min
P-230984	Security - Critical vulnerability detected	Closed	Ack	Warning	7	loginService	21/09/2023, 7:33	1 hr 23 min
P-230985	Error rate increment	Closed	Ack	Error	1	easyService	21/09/2023, 6:34	min
P-230986	CPU saturation	Closed	Ack	Resources shortage	4	paymentService	21/09/2023, 6:32	1 hr
P-230987	Multiple infrastructure problems	Closed	Ack	Custom	2	loginService	21/09/2023, 6:32	1 hr
P-230988	State Service is in undesirable state	Closed	Ack	Availability	1	paymentService	21/09/2023, 6:32	1 hr
P-230989	Windows Search is in undesirable	Closed	Ack	Availability	1	easyService	21/09/2023, 6:32	1 hr
P-230990	zyrdy	Closed	Ack	Custom	1	easyService	21/09/2023, 6:32	1 hr
P-230991	!ChPe_test_2	Closed	Ack	Slowdown	1	easyService	21/09/2023, 6:32	1 hr
P-230992	Embedded Mode is in undesirable state	Closed	Ack	Availability	1	easyService	21/09/2023, 6:32	1 hr
P-230993	Security - Critical vulnerability detected	Closed	Ack	Error	1	paymentService	21/09/2023, 6:32	1 hr
P-230994	Simple default transformation	Closed	Ack	Resources shortage	1	easyService	21/09/2023, 6:32	1 hr
P-230995	Browser monitor global outage	Closed	Ack	Availability	1	paymentService	21/09/2023, 6:32	1 hr

Category: All, Slowdown, Error, Monitoring unavailable, Custom, Available, Resources shortage  
Root cause: paymentService, easyService, easyTravel, Windows

# Management Zones to Segments and IAM

---

Common Scenarios and Solutions

## Application with Host Group

---

- Application deployed on dedicated infrastructure
- Monitored with OneAgent using Host Group configuration
  - Host group identifier based on application name, infrastructure layer, and environment
- Application team should only have access to data in Dynatrace related to their monitored application

## easyTravel AWS Linux

## Management zone name

easyTravel AWS Linux



Be careful when renaming - if there are policies that are referencing this Management zone, they will need to be adapted to the new name!

## Description

Add a description

## Rules

Add a new rule

[Enabled](#) [Summary](#) **Browser monitors (Synthetic monitor)** on **Browser monitors** where Browser monitor tags tag key equals 'easytravel-aws-linux' **HTTP monitors (Synthetic monitor)** on **HTTP monitors** where HTTP monitor tags tag key equals 'easytravel-aws-linux' **Process groups on Hosts** where Host group name equals 'aws-easytravel-lnx' **Web applications (Application Entity)** on **Web applications** where Web application name equals 'easyTravel Linux'

Display entities

**Process groups on Hosts** where Host group name equals 'aws-easytravel-lnx'

Rule type: Monitored entity

Rule applies to: Process groups

Conditions:

Add condition

Include all process groups, hosts, and services that belong to our host group

Summary: Host group name equals 'aws-easytravel-lnx'

Apply to underlying hosts of matching process groups

Apply to all services provided by the process groups

 Dynatrace Groups

## View group easyTravel AWS Linux Group

View and edit group permissions.

### Details

 Edit**Group name**

easyTravel AWS Linux Group

**Group description****Group type**

LOCAL

### Permissions

Permission	Permission Type	Scope	Action
View environment	ROLE	Environment (hfn13693:easyTravel AWS Linux, tpc-demo)	
Manage monitoring settings	ROLE	Environment (hfn13693:easyTravel AWS Linux, )	

10 ▾ rows per page

Page 1 ▾ of 1 &lt; &gt;

**Permission name**

Manage monitoring settings

**Permission description**

Manage monitoring settings

**Permission type**

role

Classic role-based access to  
only view and/or manage  
entities/data within  
management zone

**Scope**

Environments

Select environments

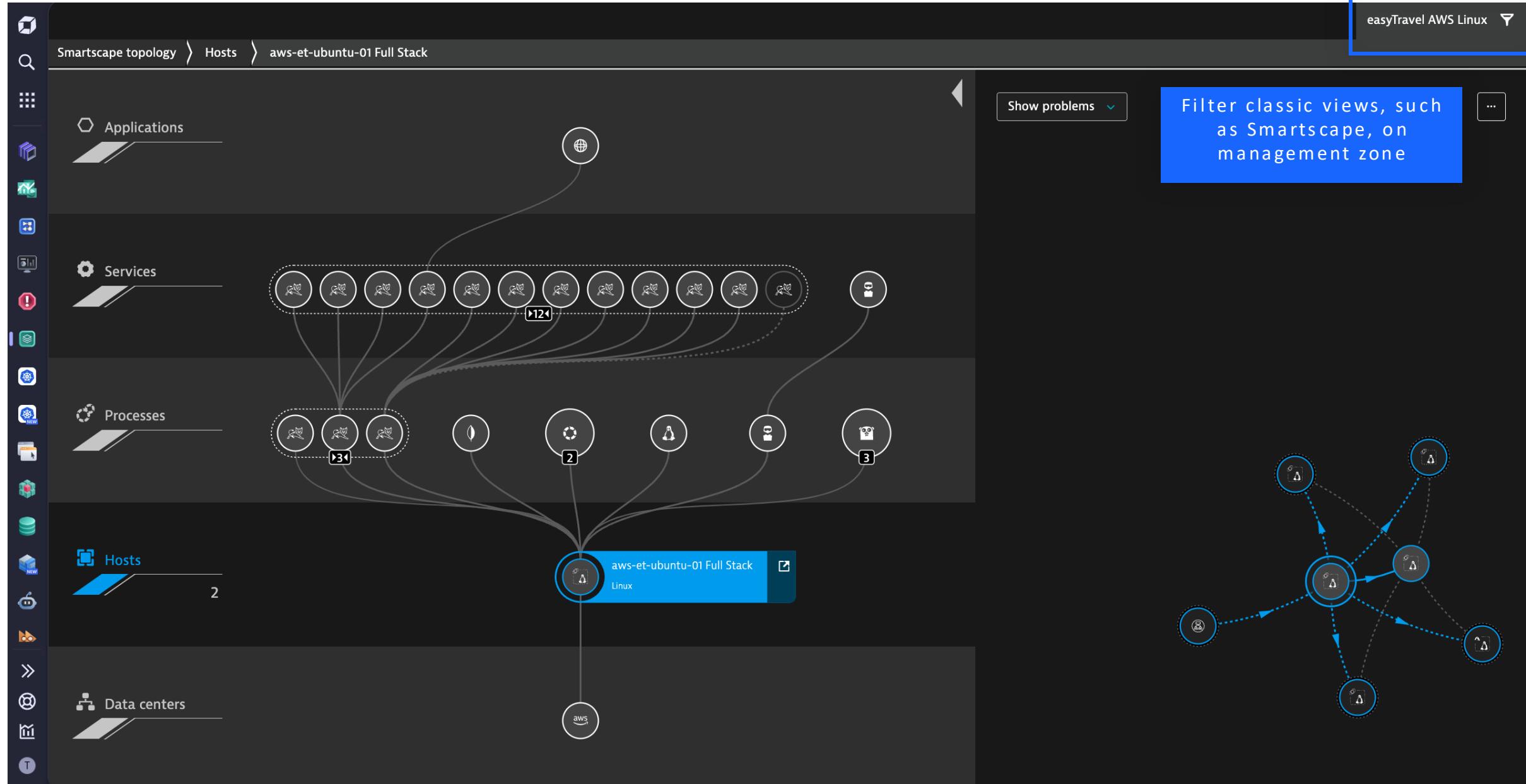
## Management zones

easyTravel AWS Linux

**Summary**

Role is scoped for 1 management zone: hfn13693:easyTravel AWS Linux

 Delete group Edit Permission



## View group easyTravel AWS Linux Group

View and edit group permissions.

### Details

**Group name**

easyTravel AWS Linux Group

**Group description****Group type**

LOCAL

### Permissions

**Permission**Read entities based on  
dt.security\_context valueRead data based on  
dt.host\_group.id value**Permission Type**

View environment

ROLE

Manage monitoring settings

ROLE

Grail permissions based on  
IAM Policy statements**Policy name\***

Data Access - AWS easyTravel Linux

**Policy description**

Read access to data in Grail for AWS easyTravel Linux

**Policy statement\***

```
1 ALLOW storage:buckets:read;
2 ALLOW storage:system:read;
3 ALLOW storage:entities:read WHERE
   storage:dt.security_context IN ('aws-easytravel-
   lnx', 'easyTravel AWS Linux');
4 ALLOW storage:events:read WHERE
   storage:dt.host_group.id = 'aws-easytravel-lnx';
5 ALLOW storage:logs:read WHERE
   storage:dt.host_group.id = 'aws-easytravel-lnx';
6 ALLOW storage:metrics:read WHERE
   storage:dt.host_group.id = 'aws-easytravel-lnx';
7 ALLOW storage:bizevents:read WHERE
```

[Learn how to manage policy statements](#)

10 rows per page

Page 1 of 1 &lt; &gt;



# IAM Policy Statements – Storage (Grail)

- <https://docs.dynatrace.com/docs/manage/identity-access-management/permission-management/manage-user-permissions-policies/advanced/iam-policystatements#storage>

## storage:entities:read

Grants permission to read records from entities

### conditions:

- `storage:entity.type` - The type of the entity.  
operators: `=`, `IN`, `startsWith`
- `storage:dt.security_context` - Custom field for security context.  
operators: `=`, `IN`, `startsWith`

Read entities based on  
`dt.security_context` value only until  
entity model actually in Grail

## storage:logs:read

Grants permission to read records from the logs-table

### conditions:

- `storage:bucket.name` - This condition reduces the effect of the record-level permission to a defined list of buckets.  
operators: `=`, `!=`, `IN`, `NOT IN`, `startsWith`, `NOT startsWith`
- `storage:k8s.namespace.name` - The name of the namespace that the pod is running in.  
operators: `=`, `IN`, `startsWith`
- `storage:k8s.cluster.name` - The name of the cluster that the pod is running in.  
operators: `=`, `IN`, `startsWith`
- `storage:host.name` - Name of the host.  
operators: `=`, `IN`, `startsWith`
- `storage:dt.host_group.id` - Id of the host group.  
operators: `=`, `IN`, `startsWith`
- `storage:log.source` - The location where the log comes from.  
operators: `=`, `IN`, `startsWith`
- `storage:dt.security_context` - Custom field for security context.  
operators: `=`, `IN`, `startsWith`
- `storage:gcp.project.id` - Google Cloud Platform Project ID.  
operators: `=`, `IN`, `startsWith`
- `storage:aws.account.id` - Amazon Web Services Account ID.  
operators: `=`, `IN`, `startsWith`
- `storage:azure.subscription` - Azure subscription.  
operators: `=`, `IN`, `startsWith`
- `storage:azure.resource.group` - Azure resource group.  
operators: `=`, `IN`, `startsWith`

Read data based on set of record  
attributes supported by 'record-level  
permissions on Grail'





Notebooks

+ Notebook

Upload



AWS easyTravel Linux



Share

History

Rerun sections



## Entity - Services

```
1 fetch dt.entity.service  
2 | fields id, entity.name, dt.security_context
```

Entity query returns entities  
with matching  
dt.security\_context value(s)

14 records

Executed at: 12/2/2024, 14:37:07, Timeframe: 12/1/2024, 14:37:07 - 12/2/2024, 14:37:07



id	entity.name	dt.security_context
SERVICE-C3FB2347EC52E636	aws-easytravel-lnx easyTravel JourneyService	easyTravel AWS Linux, easyTravel AWS
SERVICE-F3562EB1A6006F51	aws-easytravel-lnx - easyTravel-Business	easyTravel AWS Linux, easyTravel AWS
SERVICE-B0B6F60715DCB0D6	aws-easytravel-lnx - easyTravel Customer Frontend	easyTravel AWS Linux, easyTravel AWS
SERVICE-8D582233ADD12F3A	aws-easytravel-lnx - Customer Login	easyTravel AWS Linux, easyTravel AWS
SERVICE-1E17BCE54F9789F2	aws-easytravel-lnx easyTravel AuthenticationService	easyTravel AWS Linux, easyTravel AWS
SERVICE-6F109B7AE2845299	aws-easytravel-lnx - Journey Search Service	easyTravel AWS Linux, easyTravel AWS
SERVICE-72DC04145D4F5FD8	aws-easytravel-lnx - CheckDestination	easyTravel AWS Linux, easyTravel AWS
SERVICE-B6A42181E0B7B48D	aws-easytravel-lnx easyTravel VerificationService	easyTravel AWS Linux, easyTravel AWS
SERVICE-E6C1F26F8B8A8CCA	aws-easytravel-lnx - MongoDB Client Service	easyTravel AWS Linux, easyTravel AWS
SERVICE-C76957A8A73A293C	aws-easytravel-lnx - Requests executed in background threads of easytravel.customer.front..	easyTravel AWS Linux, easyTravel AWS
SERVICE-9DD2CFC955F487AC	aws-easytravel-lnx easyTravel ConfigurationService	easyTravel AWS Linux, easyTravel AWS
SERVICE-62C83A2E5651A225	aws-easytravel-lnx - CreditCardValidation	easyTravel AWS Linux, easyTravel AWS
SERVICE-7980CA47F2ACC8FC	aws-easytravel-lnx easyTravel BookingService	easyTravel AWS Linux, easyTravel AWS
SERVICE-9FFD4D061A82C88C	aws-easytravel-lnx - easyTravel Business Backend	easyTravel AWS Linux, easyTravel AWS



Notebooks + Notebook Upload

AWS easyTravel Linux Share Logs

Logs query returns log records with matching dt.host\_group.id value(s)

History Rerun sections

Logs

```
1 fetch logs
2 | summarize logs = count(), by: {dt.security_context, dt.host_group.id, dt.source_entity}
3 | sort logs desc
```

9 records Executed at: 12/2/2024, 14:19:00, Timeframe: 12/1/2024, 14:19:00 - 12/2/2024, 14:19:00, Scanned bytes: 2 GB

dt.security_context	dt.host_group.id	dt.source_entity	logs
aws-easytravel-lnx	aws-easytravel-lnx	PROCESS_GROUP_INSTANCE-1F08E3AE069C398F	32,378
aws-easytravel-lnx	aws-easytravel-lnx	PROCESS_GROUP_INSTANCE-6141FDAE2A84512F	13,911
aws-easytravel-lnx	aws-easytravel-lnx	PROCESS_GROUP_INSTANCE-2AC82E1D3DD3BCD7	4,991
aws-easytravel-lnx	aws-easytravel-lnx	PROCESS_GROUP_INSTANCE-90E0C324D701A6A7	1,745
aws-easytravel-lnx	aws-easytravel-lnx	PROCESS_GROUP_INSTANCE-5D12B96EA912FD13	925
aws-easytravel-lnx	aws-easytravel-lnx	PROCESS_GROUP_INSTANCE-06343C12BD54100A	159
aws-easytravel-lnx	aws-easytravel-lnx	PROCESS_GROUP_INSTANCE-FDCC6E8541194024	159
aws-easytravel-lnx	aws-easytravel-lnx	HOST-AB3EB8C0807B7DDA	71
aws-easytravel-lnx	aws-easytravel-lnx	HOST-50C6EA577C362DF4	53

[Notebooks](#)[+ Notebook](#)[Upload](#)

AWS easyTravel Linux

[Share](#)

Spans query returns log records  
with matching dt.host\_group.id  
value(s)

[History](#)[Rerun sections](#)**Spans**

```
1 fetch spans, samplingRatio: 10
2 | fieldsAdd service = entityName(dt.entity.service)
3 | summarize spans = count(), by: {dt.security_context, dt.host_group.id, service, dt.entity.service}
4 | sort spans desc
```

12 records

Executed at: 12/2/2024, 14:31:37, Timeframe: 12/1/2024, 14:31:36 - 12/2/2024, 14:31:36, Scanned bytes: 194 MB



dt.security_context	dt.host_group.id	service	dt.entity.service	spans
null	aws-easytravel-lnx	aws-easytravel-lnx - easyTravel Customer Frontend	SERVICE-B0B6F60715DCB0D6	23,319
null	aws-easytravel-lnx	aws-easytravel-lnx - MongoDB Client Service	SERVICE-E6C1F26F8B8A8CCA	8,387
null	aws-easytravel-lnx	aws-easytravel-lnx easyTravel JourneyService	SERVICE-C3FB2347EC52E636	6,679
null	aws-easytravel-lnx	aws-easytravel-lnx easyTravel ConfigurationService	SERVICE-9DD2CFC955F487AC	2,688
null	aws-easytravel-lnx	aws-easytravel-lnx - CheckDestination	SERVICE-72DC04145D4F5FD8	2,128
null	aws-easytravel-lnx	aws-easytravel-lnx - Requests executed in background threads of easytravel.custo...	SERVICE-C76957A8A73A293C	830
null	aws-easytravel-lnx	aws-easytravel-lnx easyTravel AuthenticationService	SERVICE-1E17BCE54F9789F2	591
null	aws-easytravel-lnx	aws-easytravel-lnx - Journey Search Service	SERVICE-6F109B7AE2845299	551
null	aws-easytravel-lnx	aws-easytravel-lnx - Customer Login	SERVICE-8D582233ADD12F3A	511
null	aws-easytravel-lnx	aws-easytravel-lnx easyTravel BookingService	SERVICE-7980CA47F2ACC8FC	161
null	aws-easytravel-lnx	aws-easytravel-lnx easyTravel VerificationService	SERVICE-B6A42181E0B7B48D	108
null	aws-easytravel-lnx	aws-easytravel-lnx - CreditCardValidation	SERVICE-62C83A2E5651A225	70





Notebooks

+ Notebook

Upload

?

AWS easyTravel Linux



Share

Metrics - Host CPU

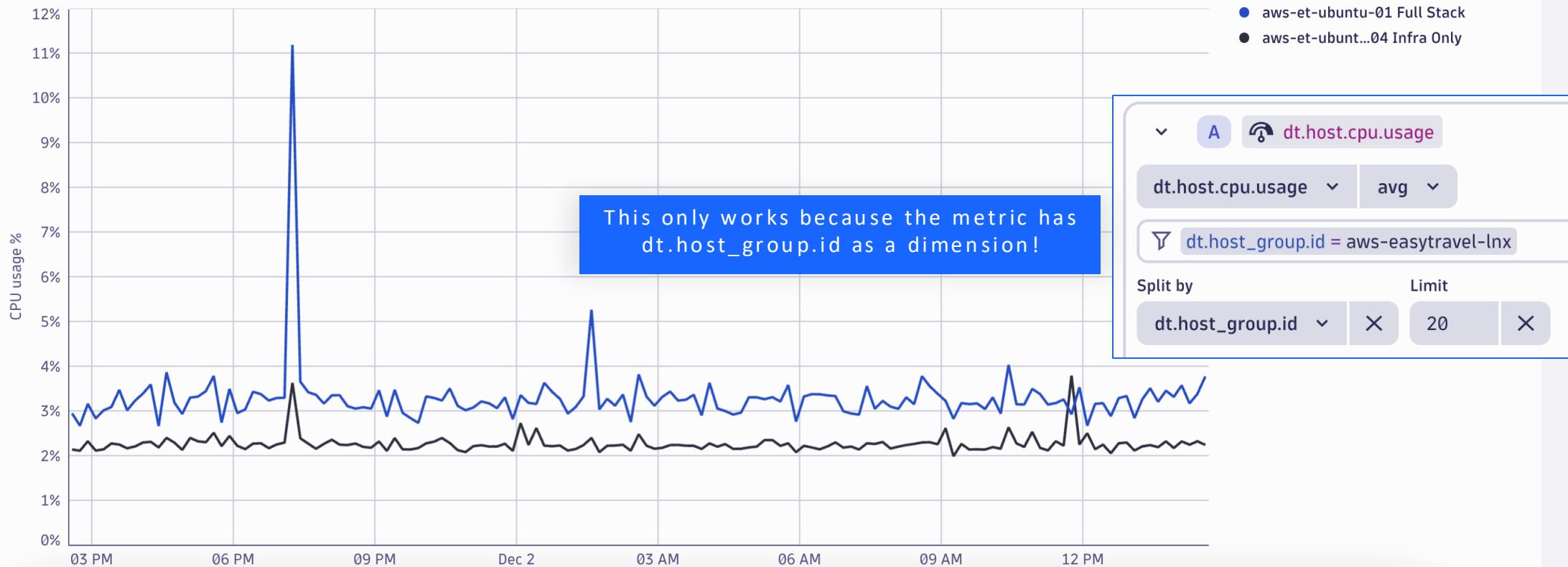
Metrics query returns log records with matching dt.host\_group.id value(s)

History

Rerun sections

```
1 timeseries cpu = avg(dt.host.cpu.usage), by: {dt.entity.host}  
2 | fieldsAdd host = entityName(dt.entity.host)
```

2 records Executed at: 12/2/2024, 14:33:55, Timeframe: 12/1/2024, 14:30:00 - 12/2/2024, 14:40:00 ⓘ



## View group Ansible Group

View and edit group permissions.

### Details

**Group name**

Ansible Group

**Group description**

No description

**Group type**

LOCAL

### Permissions

Permission	Permission Type	Actions
View environment	ROLE	Environment (hfn13693:Ansible) ⋮
Manage monitoring settings	ROLE	Environment (hfn13693:Ansible) ⋮
AppEngine - All Apps	POLICY	Environment (tpc-demo) ⋮
Data Access - Ansible	POLICY	Environment (tpc-demo) ⋮
AppEngine - Notebooks - Restricted User	POLICY	Environment (tpc-demo) ⋮

10 rows per page

Page 1 of 1 ⏪ ⏴

## Grail permissions based on IAM Policy statements

**Policy name\***

Data Access - Ansible

**Policy description**

Read access to data in Grail for Ansible

**Policy statement\***

```
1 ALLOW storage:buckets:read;
2 ALLOW storage:system:read;
3 ALLOW storage:entities:read WHERE
storage:dt.security_context IN ('aws-
ansible', 'Ansible');
4 ALLOW storage:events:read WHERE
storage:dt.host_group.id = 'aws-ansible';
5 ALLOW storage:logs:read WHERE
storage:dt.host_group.id = 'aws-ansible';
6 ALLOW storage:metrics:read WHERE
storage:dt.host_group.id = 'aws-ansible';
7 ALLOW storage:bizevents:read WHRFR
```

[Learn how to manage policy statements](#)



Notebooks

+ Notebook

Upload

?

Ansible and AWS easyTravel Linux

Share

History

Rerun sections

User now has IAM access to both Ansible and AWS easyTravel Linux

### Entity - Services

```
1   fetch dt.entity.service  
2   | fields id, entity.name, dt.security_context
```

18 records

Executed at: 12/2/2024, 14:45:32, Timeframe: 12/1/2024, 14:45:32 - 12/2/2024, 14:45:32

Entity query returns entities with matching dt.security\_context value(s)

id	entity.name
SERVICE-C3FB2347EC52E636	aws-easytravel-lnx easyTravel JourneyService
SERVICE-F3562EB1A6006F51	aws-easytravel-lnx - easyTravel-Business
SERVICE-B0B6F60715DCB0D6	aws-easytravel-lnx - easyTravel Customer Frontend
SERVICE-8D582233ADD12F3A	aws-easytravel-lnx - Customer Login
SERVICE-B11B493347B992E1	aws-ansible - host.containers.internal:443
SERVICE-1E17BCE54F9789F2	aws-easytravel-lnx easyTravel AuthenticationService
SERVICE-300550AAB9828818	aws-ansible - ip-172-31-38-80.ec2.internal:443
SERVICE-6F109B7AE2845299	aws-easytravel-lnx - Journey Search Service
SERVICE-27A2DF5F0E01A6CA	aws-ansible - _:80,443
SERVICE-72DC04145D4F5FD8	aws-easytravel-lnx - CheckDestination
SERVICE-B6A42181E0B7B48D	aws-easytravel-lnx easyTravel VerificationService
SERVICE-57CFF2B055896D2A	aws-ansible - uwsgi via domain socket /var/run/tower/uwsgi.sock
SERVICE-E6C1F26F8B8A8CCA	aws-easytravel-lnx - MongoDB Client Service
SERVICE-C76957A8A73A293C	aws-easytravel-lnx - Requests executed in background threads of easytravel.customer.front...
SERVICE-9DD2CFC955F487AC	aws-easytravel-lnx easyTravel ConfigurationService

dt.security_context
easyTravel AWS Linux, easyTravel AWS
Ansible
easyTravel AWS Linux, easyTravel AWS
Ansible
easyTravel AWS Linux, easyTravel AWS
Ansible
easyTravel AWS Linux, easyTravel AWS
easyTravel AWS Linux, easyTravel AWS
Ansible
easyTravel AWS Linux, easyTravel AWS
easyTravel AWS Linux, easyTravel AWS
Ansible
easyTravel AWS Linux, easyTravel AWS
easyTravel AWS Linux, easyTravel AWS
easyTravel AWS Linux, easyTravel AWS



 Notebooks + Notebook Upload

Ansible and AWS easyTravel Linux



Share

History

Rerun sections

## Logs

```
1 fetch logs
2 | summarize logs = count(), by: {dt.security_context, dt.host_group.id, dt.source_entity}
3 | sort logs desc
```

22 records

Executed at: 12/2/2024, 14:45:32, Timeframe: 12/1/2024, 14:45:32 - 12/2/2024, 14:45:32, Scanned bytes: 2 GB 

dt.security_context	dt.host_group.id	dt.source_entity	logs
aws-easytravel-lnx	aws-easytravel-lnx	PROCESS_GROUP_INSTANCE-1F08E3AE069C398F	32,421
aws-easytravel-lnx	aws-easytravel-lnx	PROCESS_GROUP_INSTANCE-6141FDAE2A84512F	13,914
aws-ansible	aws-ansible	PROCESS_GROUP_INSTANCE-D7BE6010DCCA71BE	8,810
aws-ansible	aws-ansible	PROCESS_GROUP_INSTANCE-71D9C5723081A753	6,958
aws-ansible	aws-ansible	PROCE	5,401
aws-easytravel-lnx	aws-easytravel-lnx	PROCE	4,992
aws-ansible	aws-ansible	PROCE	4,202
aws-ansible	aws-ansible	PROCESS_GROUP_INSTANCE-C97CACD84F92549A	3,653
aws-ansible	aws-ansible	PROCESS_GROUP_INSTANCE-69B8E6EAE96D5D9D	2,422
aws-easytravel-lnx	aws-easytravel-lnx	PROCESS_GROUP_INSTANCE-90E0C324D701A6A7	1,746
aws-ansible	aws-ansible	PROCESS_GROUP_INSTANCE-B63D4F5D67CA9A86	1,415
aws-ansible	aws-ansible	PROCESS_GROUP_INSTANCE-3807403B1AEC53C7, PROCESS_GROUP_INSTANCE-71D9C57...	1,032
aws-easytravel-lnx	aws-easytravel-lnx	PROCESS_GROUP_INSTANCE-5D12B96EA912FD13	910
aws-ansible	aws-ansible	PROCESS_GROUP_INSTANCE-14EEBE83506CE865, PROCESS_GROUP_INSTANCE-D191362...	552
aws-ansible	aws-ansible	PROCESS_GROUP_INSTANCE-14EEBE83506CE865	226
aws-easytravel-lnx	aws-easytravel-lnx	PROCESS_GROUP_INSTANCE-06343C12BD54100A	159

Logs query returns log records  
with matching dt.host\_group.id  
value(s)

# IAM Policy Statements - Segments

- <https://docs.dynatrace.com/docs/manage/identity-access-management/permission-management/manage-user-permissions-policies/advanced/iam-policystatements#storage-filter-segments-read>

Policy name\*

Policy description

Grants access to manage Segments with Admin permission.

Policy statement\*

1	ALLOW storage:filter-segments:read;
2	ALLOW storage:filter-segments:write;
3	ALLOW storage:filter-segments:delete;
4	ALLOW storage:filter-segments:admin;

[Learn how to manage policy statements ↗](#)

Policy name\*

Policy description

Grants access to manage Segments without Admin permissions.

Policy statement\*

1	ALLOW storage:filter-segments:read;
2	ALLOW storage:filter-segments:write;
3	ALLOW storage:filter-segments:delete;

[Learn how to manage policy statements ↗](#)

Policy name\*

Policy description

Provides read access to use Segments.

Policy statement\*

1	ALLOW storage:filter-segments:read;
---	-------------------------------------

[Learn how to manage policy statements ↗](#)



# IAM Policy Statements – Default Policies

## Effective Policies

Search users to find policy access.

### Find user permissions

User\* environments Find Permissions Clear  
anthony.pope-cruz@dynatrac... All Find permissions Clear inputs

### Results for 'anthony.pope-cruz@dynatrace.com'

Filter Effect Service Policies Groups reset  
segments All All AppEngine - Admin... All X Reset filters

The default (out of the box) policies for Admin and User include the policy statements for managing Segments

Effect	Service	Permission	Has conditions	Policy name	Binding level	Group name
> ALLOW	Storage	filter-segments:admin	No	AppEngine - Admin	Environment	Monitoring.admin
> ALLOW	Storage	filter-segments:admin	No	AppEngine - Admin	Environment	Account_manager
> ALLOW	Storage	filter-segments:admin	No	AppEngine - Admin	Environment	Monitoring.poweruser
> ALLOW	Storage	filter-segments:delete	No	AppEngine - Admin	Environment	Monitoring.admin
> ALLOW	Storage	filter-segments:delete	No	AppEngine - Admin	Environment	Account_manager
> ALLOW	Storage	filter-segments:delete	No	AppEngine - Admin	Environment	Monitoring.poweruser
> ALLOW	Storage	filter-segments:read	No	AppEngine - User	Environment	Dynatrace Peers - Gen3 Power User
> ALLOW	Storage	filter-segments:read	No	AppEngine - User	Environment	Dynatrace Peers
> ALLOW	Storage	filter-segments:read	No	AppEngine - User	Environment	team-wwse-txmv
> ALLOW	Storage	filter-segments:read	No	AppEngine - Admin	Environment	Monitoring.admin

10 rows per page

Page 1 of 2 < >



Segments

AWS easyTravel Linux

+ Description

Owner: T Pope-Cruz

Visibility: Unlisted

Variables

Create variables to apply in your Segment data filters

Segment data

Include all data that should be accessible when applying the Segment

**Host + Related entity**

hostGroupName = "aws-easytravel-lnx"

Entity filter to include Hosts with relevant hostGroupName and related entities

**Service instance runs\_on Host**

Type to filter

**Process belongs\_to Host**

Type to filter

This configuration alone only includes the entities, not the data! (logs, metrics, traces)

**Service runs\_on Host**

Type to filter

**Process Group runs\_on Host**

Run query

Run query

Run query

Run query





## AWS easyTravel Linux

[+ Description](#)

Owner



T Pope-Cruz

Visibility



Unlisted

### Variables

[+ Variable](#)

Data type filters to include data with matching attributes, such as `dt.security_context` and `dt.host_group.id`

Create variables to apply in your Segment data filters

### Segment data

Include all data that should be accessible when applying the Segment

#### All data types

`dt.security_context = "aws-easytravel-lnx" OR dt.host_group.id = "aws-easytravel-lnx"`[X](#)[Run query](#)[Preview](#)

50 records



Executed at: December 2, 2024, 3:39:05 PM Timeframe: 1:39:05 PM – 3:39:05 PM

[X](#)

Show results found for

[Metrics](#)[Logs](#)[Events](#)[Business events](#)[Spans](#)

timestamp	content	aws.region	aws.resource.id	dt.entity
12/2/2024, 2:17:01 PM	Dec 2 20:17:01 ip-172-31-63-127 CRON[1040374]: pam_unix(cron:session): session open...	us-east-1	i-0a1127473b5144f54	HOST-AE
12/2/2024, 2:17:01 PM	Dec 2 20:17:01 ip-172-31-63-127 CRON[1040374]: pam_unix(cron:session): session clos...	us-east-1	i-0a1127473b5144f54	HOST-AE
12/2/2024, 3:17:01 PM	Dec 2 21:17:01 ip-172-31-63-127 CRON[1043704]: pam_unix(cron:session): session open...	us-east-1	i-0a1127473b5144f54	HOST-AE
12/2/2024, 3:17:01 PM	Dec 2 21:17:01 ip-172-31-63-127 CRON[1043704]: pam_unix(cron:session): session clos...	us-east-1	i-0a1127473b5144f54	HOST-AE
12/2/2024, 2:17:01 PM	Dec 2 20:17:01 ip-172-31-63-48 CRON[361177]: pam_unix(cron:session): session opene...	us-east-1	i-06487a8b95e6993bc	HOST-50
12/2/2024, 2:17:01 PM	Dec 2 20:17:01 ip-172-31-63-48 CRON[361177]: pam_unix(cron:session): session closed...	us-east-1	i-06487a8b95e6993bc	HOST-50
12/2/2024, 3:17:01 PM	Dec 2 21:17:01 ip-172-31-63-48 CRON[366292]: pam_unix(cron:session): session opene...	us-east-1	i-06487a8b95e6993bc	HOST-50
12/2/2024, 3:17:01 PM	Dec 2 21:17:01 ip-172-31-63-48 CRON[366292]: pam_unix(cron:session): session closed...	us-east-1	i-06487a8b95e6993bc	HOST-50



## Segments - Ansible and AWS easyTravel Linux

Share

History

Rerun sections

## Entity - Services

```
1 fetch dt.entity.service  
2 | fields id, entity.name, dt.sec
```

13 records Executed at: 12/2/2024, 15:41:33, Timeframe:

Filter by segments

AWS easyTravel Linux

+ Segment

Applying the Segment filter will pre-filter the entity query on entities matching the Segment definition

Apply

id	entity.
SERVICE-C3FB2347EC52E636	aws-ea...
SERVICE-F3562EB1A6006F51	aws-easytravel-lnx - easyTravel-Business
SERVICE-B0B6F60715DCB0D6	aws-easytravel-lnx - easyTravel Customer Frontend
SERVICE-8D582233ADD12F3A	aws-easytravel-lnx - Customer Login
SERVICE-1E17BCE54F9789F2	aws-easytravel-lnx easyTravel AuthenticationService
SERVICE-6F109B7AE2845299	aws-easytravel-lnx - Journey Search Service
SERVICE-72DC04145D4F5FD8	aws-easytravel-lnx - CheckDestination
SERVICE-B6A42181E0B7B48D	aws-easytravel-lnx easyTravel VerificationService
SERVICE-E6C1F26F8B8A8CCA	aws-easytravel-lnx - MongoDB Client Service
SERVICE-C76957A8A73A293C	aws-easytravel-lnx - Requests executed in background threads of easytravel.customer.front...
SERVICE-9DD2CFC955F487AC	aws-easytravel-lnx easyTravel ConfigurationService
SERVICE-62C83A2E5651A225	aws-easytravel-lnx - CreditCardValidation
SERVICE-7980CA47F2ACC8FC	aws-easytravel-lnx easyTravel BookingService

dt.security_context
easyTravel AWS Linux, easyTravel AWS

Segments ?

< All segments

**Ansible** Owner T Pope-Cruz ▼ Visibility Unlisted ▼

+ Description

**Variables**

Create variables to apply in your Segment data filters

**Segment data**

Include all data that should be accessible when applying the Segment

Repeat the process when building the additional Segment for the other application stack

**All data types**

dt.security\_context = "aws-ansible" OR dt.host\_group.id = "aws-ansible" X Run query

**Host** + Related entity X Run query

hostGroupName = "aws-ansible" X Run query

**Service instance runs\_on Host** X Run query

Type to filter X Run query

**Service runs\_on Host** X Run query

Type to filter X Run query





Notebooks + Notebook Upload ?

Segments - Ansible and AWS easyTravel Linux ⚙️ Share History Rerun sections

Logs

1 fetch logs  
2 | summarize logs = count(), by: {dt.se...  
3 | sort logs desc

13 records Executed at: 12/2/2024, 15:46:08, Timeframe: 12/1/2024

Filter by segments

Ansible X

+ Segment

dt.security\_context dt.host\_group.id dt.source logs

aws-ansible aws-ansible PROCESS\_GROUP\_INSTANCE-B7D8C1D8CA7A1DE 8,816

aws-ansible aws-ansible PR... Applying the Segment filter will pre-filter the data (logs) query on records matching the Segment definition 6,923

aws-ansible aws-ansible PR... 5,403

aws-ansible aws-ansible PR... 4,204

aws-ansible aws-ansible PR... 3,660

aws-ansible aws-ansible PR... 2,423

aws-ansible aws-ansible PROCESS\_GROUP\_INSTANCE-B63D4F5D67CA9A86 1,408

aws-ansible aws-ansible PROCESS\_GROUP\_INSTANCE-3807403B1AEC53C7, PROCESS\_GROUP\_INSTANCE-71D9C57... 1,166

aws-ansible aws-ansible PROCESS\_GROUP\_INSTANCE-14EEBE83506CE865, PROCESS\_GROUP\_INSTANCE-D191362... 541

aws-ansible aws-ansible PROCESS\_GROUP\_INSTANCE-14EEBE83506CE865 237

aws-ansible aws-ansible PROCESS\_GROUP\_INSTANCE-AA0323D92FA41615 154

aws-ansible aws-ansible PROCESS\_GROUP\_INSTANCE-430F8DF97F408760 24

aws-ansible aws-ansible PROCESS\_GROUP\_INSTANCE-728623FC62373B4D 3

Manage segments Apply

Notebooks

+ Notebook

Upload



Segments - Ansible and AWS easyTravel Linux

Share

History

Rerun sections

Spans

```
1 fetch spans, samplingRatio: 10
2 | fieldsAdd service = entityName(dt..)
3 | summarize spans = count(), by: {dt..}
4 | sort spans desc
```

0 records

Executed at: 12/2/2024, 15:47:06, Timeframe: 12/1/20

Filter by segments

Ansible	X
AWS easyTravel Linux	X

+ Segment

[Manage segments](#) [Clear all](#) [Apply](#)

Applying multiple Segment filters will pre-filter the data (spans) query on records that match BOTH Segment definitions

No data returned!  
There are zero records where  
dt.host\_group.id == "aws-easytravel-  
lnx" AND dt.host\_group.id == "aws-  
ansible"

AND logic, not OR logic!!!

There are no records  
Try changing the timeframe or another parameter.

Possible solution?  
Dynamic Segment 'Host Group' with  
variable for dt.host\_group.id and  
dt.security\_context values!



## Kubernetes Namespace

---

- Application deployed in Kubernetes namespace on shared cluster

k8s-easytravel-aks

Management zone name

k8s-easytravel-aks

Be careful when renaming - if there are policies that are referencing this Management zone, they will need to be adapted to the new name!

Description

Add a description

Entity selector rules to include  
all Kubernetes entities related to  
the namespace

Rules

Add a new rule

Filter by

Enabled	Summary	Delete	Details
	Entities matching type("APPLICATION"),fromRelationshipSCALLS(type("SERVICE"),toRelationship.isNamespaceOfService(type("CLOUD_APPLICATION_NAMESPACE"),entityName.equals("easytravel-aks"))))		
	Entities matching type("CLOUD_APPLICATION_INSTANCE"),toRelationship.isNamespaceOfCai(type("CLOUD_APPLICATION_NAMESPACE"),entityName.equals("easytravel-aks")))		
	Entities matching type("CLOUD_APPLICATION_NAMESPACE"),entityName.equals("easytravel-aks")		
	Entities matching type("CLOUD_APPLICATION"),toRelationship.isNamespaceOfCa(type("CLOUD_APPLICATION_NAMESPACE"),entityName.equals("easytravel-aks")))		
	Entities matching type("CONTAINER_GROUP_INSTANCE"),fromRelationship.isCgiOfNamespace(type("CLOUD_APPLICATION_NAMESPACE"),entityName.equals("easytravel-aks")))		
	Entities matching type("CONTAINER_GROUP"),toRelationship.isPgOfCg(type("PROCESS_GROUP"),toRelationship.isNamespaceOfPg(type("CLOUD_APPLICATION_NAMESPACE"),entityName.equals("easytravel-aks"))))		
	Entities matching type("HTTP_CHECK"),fromRelationship.MONITORS(type("APPLICATION"),fromRelationshipSCALLS(type("SERVICE"),toRelationship.isNamespaceOfService(type("CLOUD_APPLICATION_NAMESPACE"),entityName.equals("easytravel-aks")))))		
	Entities matching type("KUBERNETES_CLUSTER"),fromRelationship.isClusterOfNamespace(type("CLOUD_APPLICATION_NAMESPACE"),entityName.equals("easytravel-aks")))		
	Entities matching type("KUBERNETES_NODE"),toRelationship.isClusterOfNode(type("KUBERNETES_CLUSTER"),fromRelationship.isClusterOfNamespace(type("CLOUD_APPLICATION_NAMESPACE"),entityName.equals("easytravel-aks"))))		
	Entities matching type("KUBERNETES_SERVICE"),toRelationship.isNamespaceOfKubernetesSv(type("CLOUD_APPLICATION_NAMESPACE"),entityName.equals("easytravel-aks")))		
	Entities matching type("PROCESS_GROUP_INSTANCE"),fromRelationship.isInstanceOf(type("PROCESS_GROUP"),toRelationship.isNamespaceOfPg(type("CLOUD_APPLICATION_NAMESPACE"),entityName.equals("easytravel-aks"))))		
	Entities matching type("PROCESS_GROUP"),toRelationship.isNamespaceOfPg(type("CLOUD_APPLICATION_NAMESPACE"),entityName.equals("easytravel-aks")))		
	Entities matching type("SERVICE"),tag(Tier:Database),toRelationshipSCALLS(type("SERVICE"),toRelationship.isNamespaceOfService(type("CLOUD_APPLICATION_NAMESPACE"),entityName.equals("easytravel-aks"))))		



## View group k8s-easytravel-aks Group

View and edit group permissions.

### Details

 Edit

**Group name**

k8s-easytravel-aks Group

**Group description**

No description

**Group type**

LOCAL

### Permissions

Permission	Permission Type	Scope	Actions
View environment	ROLE	Environment (hfn13693:k8s-easytravel-aks)	
Manage monitoring settings	ROLE	Environment (hfn13693:k8s-easytravel-aks)	

10 ▾ rows per page

Page 1 ▾ of 1 < >

k8s-easytravel-aks ⚙

Smartscape topology > Services > easytravel-nginx [easytravel-aks]

Applications

Services

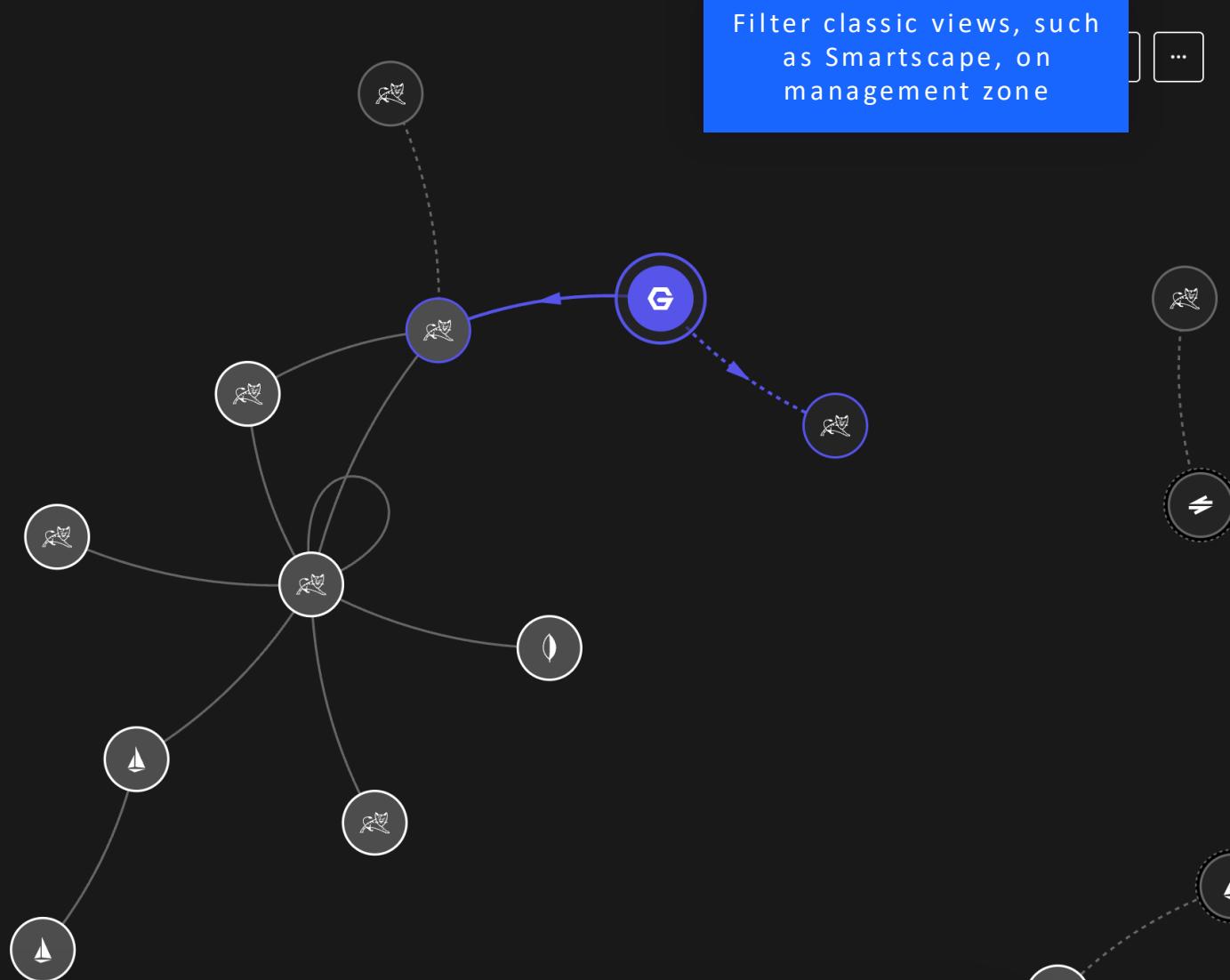
Processes

Hosts

Data centers



easytravel-nginx [ea  
sytravel-aks]  
Web request service



Filter classic views, such  
as Smartscape, on  
management zone

<https://docs.dynatrace.com/docs/manage/identity-access-management/permission-management/manage-user-permissions-policies/iam-policy-boundaries>

The screenshot shows the Dynatrace IAM Policy boundaries page. At the top, there's a navigation bar with Account Management, Home, Subscription, Identity & access management (selected), Lens, and Settings. Below the navigation is a secondary navigation bar with Dynatrace and Policies. Under Policies, the 'Boundaries' tab is selected. A large section title 'Policy boundaries' is followed by the sub-instruction 'Define policy boundaries that clearly specify whether an action in Dynatrace is allowed.' A link 'Learn how to manage policy boundaries' is provided. A central callout box states: 'Policy boundaries make IAM policies modular, separating the permission statement from the where condition statement'. The main content area displays a table of boundaries:

Name	Source	Actions
Kubernetes Namespace - easytravel-aks	Custom	⋮

At the bottom, there are filters for 'rows per page' (set to 10) and 'Page' (set to 1 of 1). A blue button '+ Boundary' is located on the right side.

Boundary name\*

Kubernetes Namespace - easytravel-aks

Boundary query\*

1 storage:k8s.namespace.name = "easytravel-aks";

Boundary query acts as WHERE clause condition



Policy name\*

Data Access - Kubernetes Namespace with Boundary

Policy description

Read access to data in Grail for Kubernetes Namespace with dynamic Boundary

Policy statement\*

```
1 ALLOW storage:buckets:read;  
2 ALLOW storage:system:read;  
3 ALLOW storage:events:read;  
4 ALLOW storage:logs:read;  
5 ALLOW storage:metrics:read;  
6 ALLOW storage:bizevents:read;  
7 ALLOW storage:spans:read;
```

Generic policy granting  
access to all data types in  
Grail without conditions

[Learn how to manage policy statements](#)

Permission name

Data Access - Kubernetes Namespace with Boundary

Permission description

Read access to data in Grail for Kubernetes Namespace with dynamic Boundary

Permission type

policy

Scope

Account (all environments)

hfn13693 - tpc-demo

Policy boundary applies  
conditions to populate WHERE  
clause behind the scenes

Boundaries

You can view/edit policy boundaries in [Policy management](#)

Kubernetes Namespace - easytravel-aks

Summary

Policy is scoped for 1 environment: hfn13693

Policy is bounded with 1 boundary: Kubernetes Namespace - easytravel-aks



## View group k8s-easytravel-aks Group

View and edit group permissions.

### Details

**Group name**

k8s-easytravel-aks Group

**Group description**

No description

**Group type**

LOCAL

[Edit](#)

### Permissions

Permission	Permission Type
View environment	ROLE
Manage monitoring settings	ROLE
Data Access - Kubernetes Namespace easytravel-aks	POLICY
AppEngine - Notebooks - Restricted User	POLICY
AppEngine - All Apps	POLICY
Data Access - Kubernetes Namespace with Boundary	POLICY

**Policy name\***

Data Access - Kubernetes Namespace easytravel-aks

**Policy description**

Read access to data in Grail for Kubernetes Namespace easytravel-aks

**Policy statement\***

```
1 ALLOW storage:entities:read WHERE  
storage:dt.security_context = 'k8s-easytravel-aks';|
```

Entity permissions do not support policy boundaries, until entities move to Grail

[Learn how to manage policy statements ↗](#)

Environment (minio95:k8s-easytravel-aks)

[Delete group](#)[Edit](#)[+ Permission](#)[Actions](#)



## Segments



&lt; All segments

**k8s-namespace**

Kubernetes Namespace

**Variables**

Create variables to apply in your Segment data filters

**Segment data**

Include all data that should be accessible when applying the Segment



Use Variables in Segments to create Dynamic Segments

Owner



T Pope-Cruz

Visibility



Unlisted



\$name

\$id

easytravel-aks

CLOUD\_APPLICATION\_NAM...



## Variables configuration

Values

```
1 fetch dt.entity.cloud_application_namespace
2 | filterOut in(entity.name, "default", "kube-public", "kube-node-lease", "kube-system", "kube-public")
3 | fields name = entity.name, id
```

Run query

Open in Notebooks

## Variables based on DQL query

1 record

⚠ Executed at: December 2, 2024, 4:39:33 PM Timeframe: 2:39:34 PM - 4:39:34 PM

Preview

name

id

easytravel-aks CLOUD\_APPLICATION\_NAMESPACE-7564A140BA2430F2

The following variables will be created

\$name

\$id

easytravel-aks

CLOUD\_APPLICATION\_NAM...

Cancel

Done

ts + Spans + Entities





## Segments



## k8s-namespace

Kubernetes Namespace

Owner



T Pope-Cruz

Visibility



Unlisted



### Variables

Create variables to apply in your Segment data filters

\$name

\$id

easytravel-aks

CLOUD\_APPLICATION\_NAM...



### Segment data

Include all data that should be accessible when applying the Segment

Kubernetes namespace

+ Related entity

entity.name = "\$name"

Entity filter in Segment based on selected variable value



Run query

Container group instance belongs\_to Kubernetes namespace



Type to filter

Run query

Service belongs\_to Kubernetes namespace



Type to filter

Run query

Process Group belongs\_to Kubernetes namespace



Type to filter

Run query

Kubernetes pod belongs\_to Kubernetes namespace



Segments

**k8s-namespace**  
Kubernetes Namespace

Owner: T Pope-Cruz | Visibility: Unlisted

**Variables**

\$name: easytravel-aks | \$id: CLOUD\_APPLICATION\_NAM...

Create variables to apply in your Segment data filters

**Segment data**

All data types

Data filter in Segment based on selected variable values

Filter: k8s.namespace.name = "\$name" OR dt.entity.cloud\_application\_namespace = "\$id"

Run query

Preview: 50 records | Executed with: \$name: easytravel-aks | \$id: CLOUD\_APPLICATION\_NAM... | At: December 2, 2024, 4:43:34 PM | Timeframe: 2:43:34 PM - 4:43:34 PM

Show results found for Metrics | Logs | Events | Business events | Spans

timestamp	content	azure.location	dt.entity.cloud_application	dt.entity.cloud_application_instance	dt.entity.cloud_application_namespace	dt.entity.container
12/2/2024	2024...	eastus	CLOUD_APPLICATION-93...	CLOUD_APPLICATION_INSTANCE-...	CLOUD_APPLICATION_NAMESPACE-7...	CONTAINER_G
12/2/2024	2024...	eastus	CLOUD_APPLICATION-93...	CLOUD_APPLICATION_INSTANCE-...	CLOUD_APPLICATION_NAMESPACE-7...	CONTAINER_G
12/2/2024	2024...	eastus	CLOUD_APPLICATION-93...	CLOUD_APPLICATION_INSTANCE-...	CLOUD_APPLICATION_NAMESPACE-7...	CONTAINER_G
12/2/2024	2024...	eastus	CLOUD_APPLICATION-93...	CLOUD_APPLICATION_INSTANCE-...	CLOUD_APPLICATION_NAMESPACE-7...	CONTAINER_G
12/2/2024	2024...	eastus	CLOUD_APPLICATION-93...	CLOUD_APPLICATION_INSTANCE-...	CLOUD_APPLICATION_NAMESPACE-7...	CONTAINER_G
12/2/2024	2024...	eastus	CLOUD_APPLICATION-93...	CLOUD_APPLICATION_INSTANCE-...	CLOUD_APPLICATION_NAMESPACE-7...	CONTAINER_G
12/2/2024	2024...	eastus	CLOUD_APPLICATION-93...	CLOUD_APPLICATION_INSTANCE-...	CLOUD_APPLICATION_NAMESPACE-7...	CONTAINER_G
12/2/2024	2024...	eastus	CLOUD_APPLICATION-93...	CLOUD_APPLICATION_INSTANCE-...	CLOUD_APPLICATION_NAMESPACE-7...	CONTAINER_G



CONFIDENTIAL

Notebooks + Notebook Upload ?

Kubernetes - k8s-easytravel-aks ⋮ Share History Rerun sections

+ Entity - Services

Filter by segments

k8s-namespace easytravel-aks X

+ Segment

10 records Executed at: 12/2/2024, 16:44:58, Timeframe Last 24 hours Hide input Options ⋮

Applying the Segment filter will pre-filter the entity query on entities matching the Segment definition

id	entity	tags
SERVICE-1ADB907C6A750A54	easytravel-aks	k8s-easytravel-aks, Kubernetes, OpenTelemetry
SERVICE-3174F290B0B9F855	tpc-aks-demo - easytravel-angular-frontend	k8s-easytravel-aks, Kubernetes, k8s-easytravel, easyTravel EKS, Azure
SERVICE-89069B5ED17725F5	CheckDestinationService [easytravel-aks]	k8s-easytravel-aks, Kubernetes, k8s-easytravel, easyTravel EKS, Azure
SERVICE-991D984498EC15F9	easytravel-nginx [easytravel-aks]	k8s-easytravel-aks, Kubernetes, k8s-easytravel, easyTravel EKS, Azure
SERVICE-D712F1BFE46B2007	tpc-aks-demo - easytravel-nginx	k8s-easytravel-aks, Kubernetes, OpenTelemetry
SERVICE-F0644A49D8AAA318	aks easytravel-aks Requests executed in background t...	k8s-easytravel-aks, Kubernetes, k8s-easytravel, easyTravel EKS, Azure
SERVICE-78F01A3E67A5DDF5	tpc-aks-demo - easytravel-backend	k8s-easytravel-aks, Kubernetes, OpenTelemetry
SERVICE-14DF912141204861	easytravel-angular-frontend [easytravel-aks]	k8s-easytravel-aks, Kubernetes, k8s-easytravel, easyTravel EKS, Azure
SERVICE-91394FE9C528092A	CustomerLogin [easytravel-aks]	k8s-easytravel-aks, Kubernetes, k8s-easytravel, easyTravel EKS, Azure
SERVICE-D715056066D2E9C2	easytravel-backend [easytravel-aks]	k8s-easytravel-aks, Kubernetes, k8s-easytravel, easyTravel EKS, Azure

Admin - Kubernetes k8s-easytravel-aks

```
1 fetch dt.entity.service
2 | fields id, entity.name, dt.security_context
3 | sort dt.security_context desc
```

48 records Executed at: 12/2/2024, 16:52:30, Timeframe: 12/1/2024, 16:52:30 - 12/2/2024, 16:52:30

id	entity.name
SERVICE-991D984498EC15F9	easytravel-nginx [easytravel-aks]
SERVICE-F0644A49D8AAA318	aks easytravel-aks Requests executed in background threads of tomcat ...
SERVICE-14DF912141204861	easytravel-angular-frontend [easytravel-aks]
SERVICE-89069B5ED17225F5	CheckDestinationService [easytravel-aks]
SERVICE-D715056066D2E9C2	easytravel-backend [easytravel-aks]
SERVICE-91394FE9C528092A	CustomerLogin [easytravel-aks]
SERVICE-1ADB907C6A750A54	easyTravel-Business DB [easytravel-aks]
SERVICE-D712F1BF46B2007	tpc-aks-demo - easytravel-nginx
SERVICE-78F01A3E67A5DDF5	tpc-aks-demo - easytravel-backend
SERVICE-3174F290B0B9F855	tpc-aks-demo - easytravel-angular-frontend
SERVICE-A9885C924BEC893C	eks easytravel-prod Java Frontend
SERVICE-AD085A6CA5A0595F	eks easytravel-prod AuthenticationService
SERVICE-DCF70038864DB041	eks easytravel-prod easyTravel Business Backend
SERVICE-5AAC3159E253E5A2	eks easytravel-prod EasytravelService
SERVICE-F7F61BC150373D1C	eks easytravel-prod BookingService
SERVICE-719FDD47599045E6	eks easytravel-prod tomcat easytravel-frontend-* on port 8080
SERVICE-EE35311C7179AE8B	eks easytravel-prod DB

Filter by segments

k8s-namespace easytravel-nonprod, easytr... + Segment Search

- easytravel-nonprod
- easytravel-prod
- easytravel-aks
- easytrade-nonprod
- argocd
- dynatrace
- prometheus
- opentelemetry-operator-system
- app-routing-system
- istio-system
- astronomy-shop
- springio
- cert-manager
- easytrade-prod

Apply

CONFIDENTIAL

Applying multiple values of Dynamic Segment filter uses OR logic, includes any entity (or data) that matches at least one value selected from list

Admin - Kubernetes k8s-easytravel-aks

```
1 fetch dt.entity.service
2 | fields id, entity.name, dt.security_context
3 | sort dt.security_context desc
```

19 records Executed at: 12/2/2024, 16:54:32, Timeframe: 12/1/2024, 16:54:31 - 12/2/2024, 16:54:31

id	entity.name
SERVICE-A9885C924BEC893C	eks easytravel-prod Java Frontend
SERVICE-AD085A6CA5A0595F	eks easytravel-prod AuthenticationService
SERVICE-DCF70038864DB041	eks easytravel-prod easyTravel Business Backend
SERVICE-5AAC3159E253E5A2	eks easytravel-prod EasytravelService

Filter by segments

k8s-namespace easytravel-nonprod, easytr... + Segment

owner easytrade-devops, easytr... + Segment

Manage segments

Clear all Apply

CONFIDENTIAL



# Hands On

---

# Hands On

---

- **Environment Set Up**

- Dynatrace Setup
- GitHub Repository & Codespaces Setup
- IDP Setup & Validation

- **Self-Service Onboarding**

- Onboard New Apps
- Explore New Apps
- Analyze without Segments
  - Kubernetes App, Dashboards App, Notebooks App

- **Basic Segments**

- Kubernetes
  - Cluster, Namespace
- Analyze with Segments

- **OpenPipeline**

- Logs Pipeline
- Metrics Pipeline

- **Complex Segments**

- IDP Project Team
- Analyze with Segments



# Wrap Up

---

# Key Takeaways

---

- Management Zones are still supported for Gen2 use cases
  - Customers can continue to manage access and filter visualizations to Classic Dynatrace using MZ
- Managing access and filtering visualizations with Grail uses IAM + Segments
  - Manage access to data in Grail with IAM Policies, including record-level permission statements
  - Filter visualizations with Grail using Segments
- Customers should strategize their IAM Policies for data access management first before developing Segment configurations
  - Segments are quick and reusable DQL query filters, not like management zones!
- Segment feature improvements are coming
  - Many of today's limitations are due to the entity model not being in Grail
- Best practice implementation sequence:
  - Ingest data into Dynatrace, Configure OpenPipeline, Configure IAM, Configure Segments





CLOUD DONE RIGHT