



Observability as Code with Datadog and HashiCorp Terraform



Kerim Satirli

@ksatirli

Developer Advocate

HashiCorp

Daniel Maher

@phrawzty

Developer Advocate

Datadog

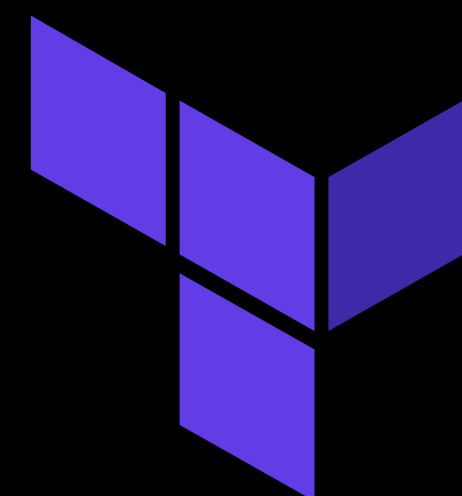


Agenda

- What is Datadog?
- What is Terraform?
- What is Observability as Code?
- How do Datadog and Terraform integrate?
- Q&A



DATADOG



HashiCorp

Terraform Cloud

app.terraform.io

hashicorp-team-demo / Workspaces / ecommerce-workshop / Runs / run-Q6xFgDaeT5vypeN2

ecommerce-workshop ? **Runs** States Variables Settings ☰ Queue plan ▾

✓ **PLANNED** **Queued manually in Terraform Cloud** CURRENT

hashicorp-demos triggered a run from Terraform Cloud UI a few seconds ago Run Details ▾

✓ **Plan finished** a few seconds ago ^

Started a few seconds ago

⬇ Download Sentinel mocks ? Sentinel mocks can be used for [testing your Sentinel policies](#)

📄 View raw log Follow log ↑ Top ↓ Bottom ⚙ Expand ⌕ Full screen

```
Terraform v0.13.2
Configuring remote state backend...
Initializing Terraform configuration...
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

data.google_compute_network.default: Refreshing state... [id=projects/hc-da-test/global/networks/default]
google_compute_firewall.ecommerce[0]: Refreshing state... [id=projects/hc-da-test/global/firewalls/allow-ecommerce]
google_compute_address.ecommerce: Refreshing state... [id=projects/hc-da-test/regions/us-east1/addresses/datadog-webinar-ecommerce]
google_compute_instance.ecommerce: Refreshing state... [id=projects/hc-da-test/zones/us-east1-b/instances/datadog-webinar-ecommerce]

-----

No changes. Infrastructure is up-to-date.

This means that Terraform did not detect any differences between your
configuration and real physical resources that exist. As a result, no
```





storedog.example



70514.2 ms x3



Login

All departments ▾

Search

Search

Home

Cart: (Empty)













SWEET SUNDAY!

Enter coupon code 'OFF' at checkout for a \$300.10 discount

Shop by Categories

- Bags
- Mugs
- Clothing

COOL HATS

 NO IMAGE AVAILABLE Datadog Tote \$15.99	 NO IMAGE AVAILABLE Datadog Bag \$22.99	 NO IMAGE AVAILABLE Datadog Baseball Jersey \$19.99	 NO IMAGE AVAILABLE Datadog Jr. Spaghetti \$19.99
 NO IMAGE AVAILABLE Datadog Ringer T-Shirt \$19.99	 NO IMAGE AVAILABLE Apache Baseball Jersey \$19.99	 NO IMAGE AVAILABLE Spree Baseball Jersey \$19.99	 NO IMAGE AVAILABLE Spree Jr. Spaghetti \$19.99
 NO IMAGE AVAILABLE Spree Ringer T-Shirt \$19.99	 NO IMAGE AVAILABLE Spree Tote \$15.99	 NO IMAGE AVAILABLE Spree Bag \$22.99	 NO IMAGE AVAILABLE Datadog Mug \$13.99



What is Observability as Code



How do Datadog and Terraform integrate?

Datadog Provider for Terraform



The screenshot shows the Terraform Registry page for the 'datadog' provider. The browser address bar displays 'registry.terraform.io/providers/datadog/datadog'. The page header includes the Terraform logo and a search bar. The breadcrumb trail is 'Providers / DataDog / datadog / Version 2.13.0', with a 'Latest Version' button. The provider name 'datadog' is shown with a verified badge and an 'Overview' link. The provider's logo, a dog's head, is displayed next to the name. A 'Verified' badge is present, along with the text 'by:DataDog'. A dark button labeled 'Logging & Monitoring' is visible. Below this, a table-like structure shows the version '2.13.0', published '20 hours ago', and '< 100' installs. A link to the source code is provided: 'DataDog/terraform-provider-datadog'.


registry.terraform.io/providers/datadog/datadog

Terraform Registry

Search Providers and Modules

Providers / DataDog / datadog / Version 2.13.0 Latest Version

datadog Overview

 **datadog**

Verified by:DataDog

Logging & Monitoring

VERSION	PUBLISHED	INSTALLS	SOURCE CODE
2.13.0	20 hours ago	< 100	DataDog/terraform-provider-datadog

Datadog Provider for Terraform



A screenshot of a web browser showing the Terraform Registry page for the Datadog provider. The browser's address bar displays 'registry.terraform.io/providers/datadog/datadog'. The page features a blue header with the Terraform logo and a search bar. Below the header, there are navigation links for 'Providers', 'DataDog', 'datadog', and 'Version 2.13.0', along with a 'Latest Version' button. The main content area is titled 'datadog' and includes a 'Documentation' link. A sidebar on the left contains a search filter and a list of links: 'datadog provider', 'Guides', 'Resources', and 'Data Sources'. The main text area is titled 'Datadog Provider' and contains a paragraph explaining the provider's purpose and a section for 'Example Usage' which begins with a code snippet: '# Configure the Datadog provider'.

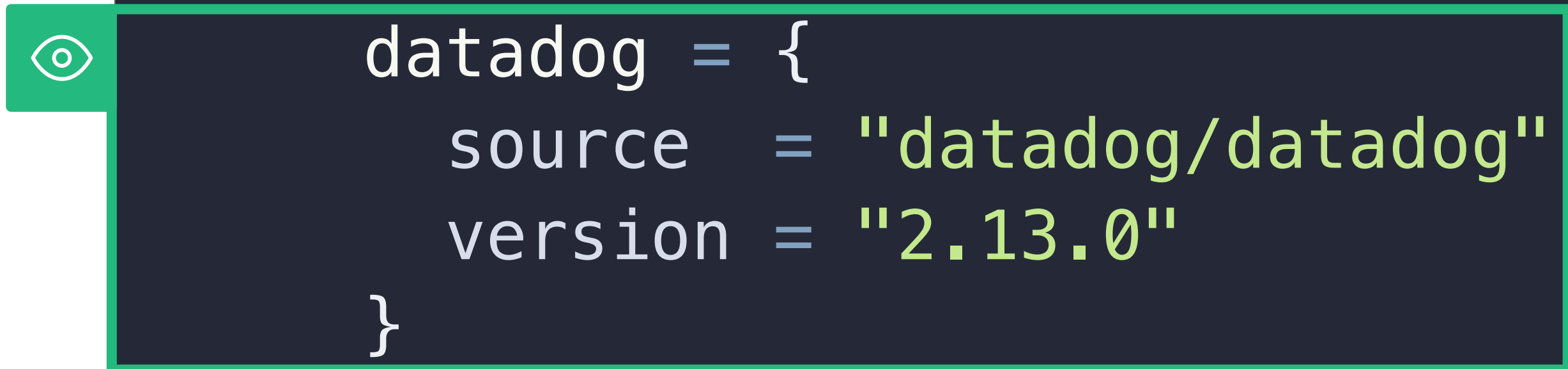
Datadog Provider for Terraform



```
CODE EDITOR

terraform {
  required_providers {
    datadog = {
      source = "datadog/datadog"
      version = "2.13.0"
    }
  }

  required_version = "~> 0.13"
}
```



Datadog Provider for Terraform



```
CODE EDITOR

provider "datadog" {
  # Datadog credentials
}
```

Datadog Provider for Terraform



```
provider "datadog" {  
  api_url = "https://api.datadoghq.com/"  
}
```

CODE EDITOR

Datadog Provider for Terraform



```
CODE EDITOR

provider "datadog" {
  api_url = "https://api.datadoghq.eu/"
}
```

Datadog Monitors Interface



The screenshot shows the Datadog Monitors interface in a browser window. The URL is `app.datadoghq.com/monitors/manage`. The interface includes a sidebar with navigation options like 'New Stuff!', 'Watchdog', 'Events', 'Dashboards', 'Infrastructure', 'Monitors', 'Metrics', 'Integrations', 'APM', 'Notebooks', 'Logs', 'Security', and 'UX Monitoring'. The main content area is titled 'Monitors' and has tabs for 'Manage Monitors', 'Triggered Monitors', and 'Manage Downtime'. A search bar is present with the text 'Filter monitors'. Below the search bar, there are filters for 'Status' (Triggered, Alert, Warn, No Data, OK) and 'Muted' (false, true). A table lists 11 monitors with columns for 'STATUS', 'NAME', 'TAGS', and a notification bell icon. The table shows various monitors for 'ruby-shop' with different services and metrics, mostly in an 'OK' state, and one 'NO DATA' state.

STATUS	NAME ↑	TAGS	🔔
<input type="checkbox"/> OK	Service advertisements-service has a high average latency on ruby-shop	service:advertisements-service env:ruby-shop	<input type="checkbox"/>
<input type="checkbox"/> OK	Service advertisements-service has a high error rate on ruby-shop	service:advertisements-service env:ruby-shop	<input type="checkbox"/>
<input type="checkbox"/> OK	Service advertisements-service has a high p90 latency on ruby-shop	service:advertisements-service env:ruby-shop	<input type="checkbox"/>
<input type="checkbox"/> OK	Service discounts-service has a high average latency on ruby-shop	service:discounts-service env:ruby-shop	<input type="checkbox"/>
<input type="checkbox"/> OK	Service discounts-service has a high error rate on ruby-shop	service:discounts-service env:ruby-shop	<input type="checkbox"/>
<input type="checkbox"/> OK	Service discounts-service has a high p90 latency on ruby-shop	service:discounts-service env:ruby-shop	<input type="checkbox"/>
<input type="checkbox"/> NO DATA	Service store-frontend has a high average latency on ruby-shop	service:store-frontend env:ruby-shop	<input type="checkbox"/>
<input type="checkbox"/> OK	Service store-frontend has a high error rate on ruby-shop	service:store-frontend env:ruby-shop	<input type="checkbox"/>
<input type="checkbox"/> OK	Service store-frontend has a high p90 latency on ruby-shop	service:store-frontend env:ruby-shop	<input type="checkbox"/>
<input type="checkbox"/> OK	Service store-frontend has an anomalous p90 latency on ruby-shop	service:store-frontend env:ruby-shop	<input type="checkbox"/>
<input type="checkbox"/> OK	[Auto] Clock in sync with NTP		

Datadog Monitor



```
resource "datadog_monitor" "high_error_rate" {  
  for_each = var.services  
  name     = "HIGH: ${each.key} (${each.value.env})"  
  type     = "query alert"  
  message  = "${each.key} has a high error rate!"  
  
  ...  
}
```

CODE EDITOR

Datadog Monitor



```
CODE EDITOR

...

query = "avg(last_10m):(sum:trace.$
{each.value.framework}.request.errors{env:$
{each.value.environment},service:${each.key} } / sum:trace.$
{each.value.framework}.request.hits{env:$
{each.value.environment},service:${each.key} }) > $
{each.value.high_error_rate_critical}"
```

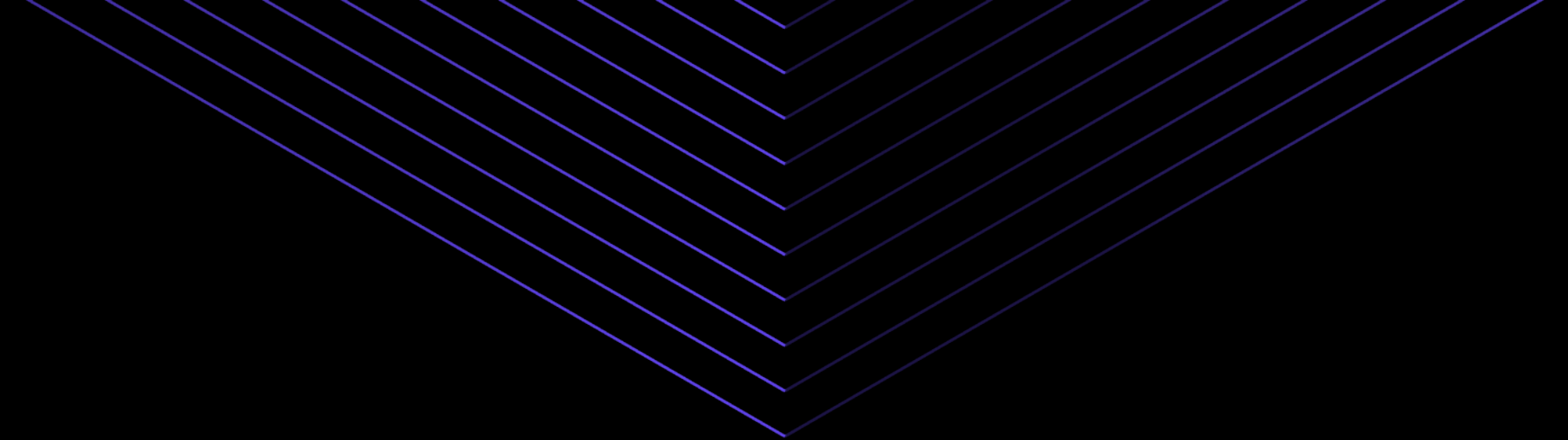


Demonstration

Resources



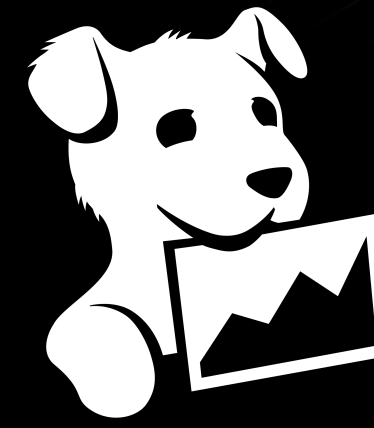
- Datadog: datadoghq.com/free-datadog-trial
- Terraform Cloud: app.terraform.io/signup
- Terraform provider: registry.terraform.io/providers/datadog/datadog
- webshop code: github.com/datadog/ecommerce-workshop
- workshop code: github.com/hashicorp/observability-as-code



Q & A

ask us anything

(related to today's session)



Thank You

kerim@hashicorp.com

daniel.maher@datadoghq.com