



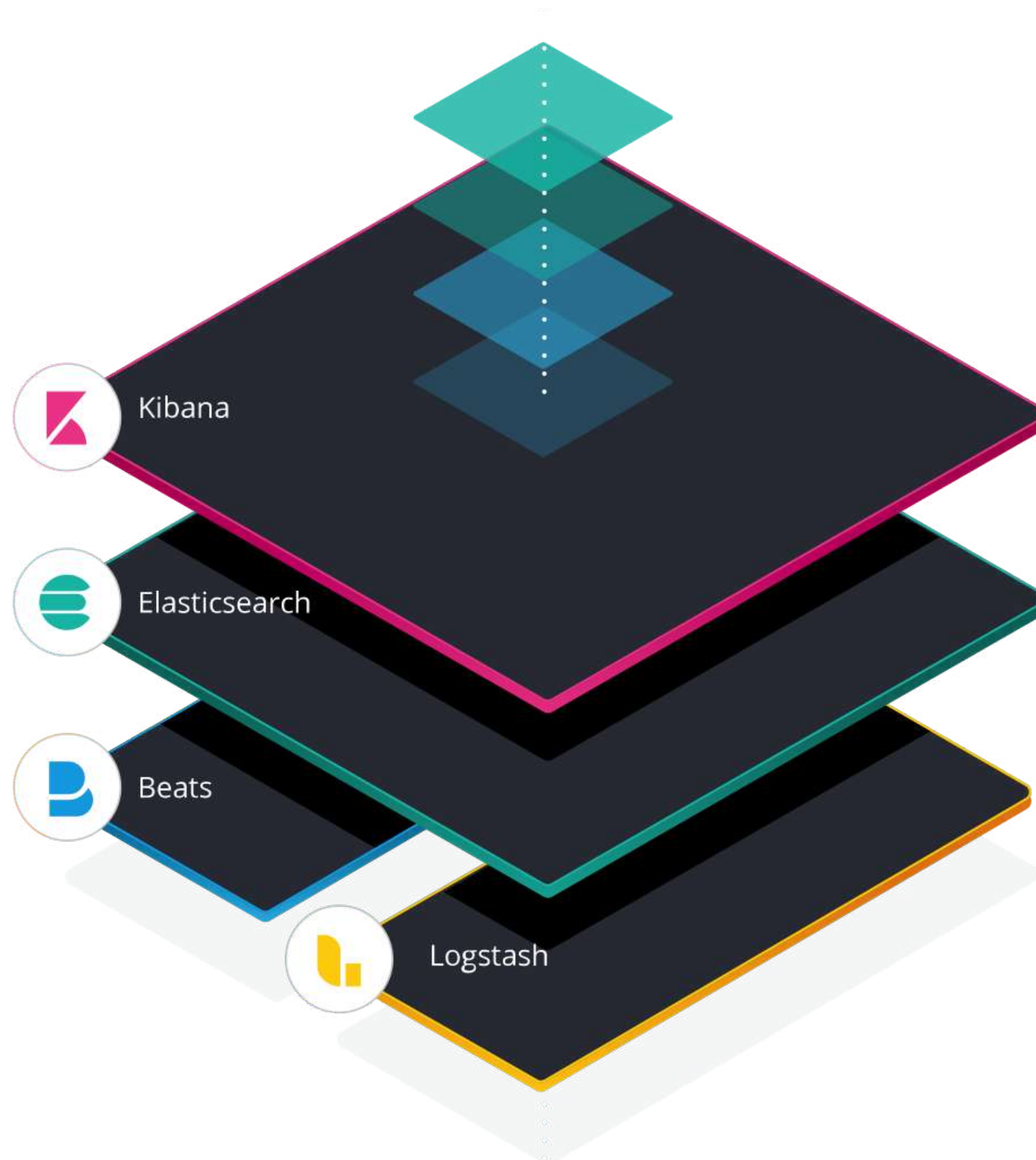
elastic



Terraform

Philipp Krenn

@xeraa



Installation Surface

DEB, RPM, ZIP, TAR.GZ, ~~MSI~~

Windows / Linux / MacOS, x86_64 / aarch64

Docker, ~~Docker Compose~~, Helm Charts*, Operator

Chef*, Puppet**, ~~Ansible~~

Cloud

* best effort

** community run



Why Terraform?

Given for (Cloud) provisioning and momentum

One set of tools for Cloud + Stack

Golang

Terraform for Cloud

<https://github.com/elastic/terraform-provider-ec>

```
resource "ec_deployment" "cluster_1" {  
  name           = "cluster_1"  
  region         = "gcp-europe-west3"  
  version        = "8.2.3"  
  deployment_template_id = "gcp-memory-optimized-v2"  
  
  elasticsearch {}  
  
  kibana {}  
}
```

More Terraform for Cloud

```
resource "ec_deployment" "cluster_2" {
  name                = "cluster_2"
  region              = "gcp-europe-west3"
  version              = "8.2.3"
  deployment_template_id = "gcp-memory-optimized-v2"

  elasticsearch {
    remote_cluster {
      deployment_id = ec_deployment.cluster_1.id
      alias          = ec_deployment.cluster_1.name
      ref_id         = ec_deployment.cluster_1.elasticsearch.0.ref_id
    }
  }
}

kibana {}
}
```

♥ Command-Line

```
output=$(terraform output -json)
endpoint=$(echo $output | jq -r ".elasticsearch_endpoint.value")
username=$(echo $output | jq -r ".elasticsearch_username.value")
password=$(echo $output | jq -r ".elasticsearch_password.value")
curl $endpoint -u $username:$password
```


Terraform for Stack Setup

<https://github.com/elastic/terraform-provider-elasticstack>

```
provider "elasticstack" {  
  elasticsearch {  
    username = ec_deployment.custom-deployment-id.elasticsearch_username  
    password = ec_deployment.custom-deployment-id.elasticsearch_password  
    endpoints = [ec_deployment.custom-deployment-id.elasticsearch[0].https_endpoint]  
  }  
}
```

Terraform for Stack

```
data "elasticstack_elasticsearch_ingest_processor_set" "set_field_terraform" {
  field = "pipeline-source"
  value = "terraform"
}

data "elasticstack_elasticsearch_ingest_processor_grok" "grok_the_log" {
  field      = "message"
  patterns = ["%%{TIMESTAMP_ISO8601:@timestamp} %%{LOGLEVEL:level} %%{GREEDYDATA:message}"]
}

resource "elasticstack_elasticsearch_ingest_pipeline" "ingest" {
  name = "my-ingest-pipeline"
  processors = [
    data.elasticstack_elasticsearch_ingest_processor_set.set_field_terraform.json,
    data.elasticstack_elasticsearch_ingest_processor_grok.grok_the_log.json
  ]
}
```

Terraform Stack in Operator?

Reuse but what is native (enough)?

Hacks (no Kibana in Stack yet)

```
data "local_file" "dashboard" {
  filename = "${path.module}/dashboard.ndjson"
}

resource "null_resource" "store_local_dashboard" {
  provisioner "local-exec" {
    command = "curl -X POST -u ${ec_deployment.custom-deployment-id.elasticsearch_username}:\
    ${ec_deployment.custom-deployment-id.elasticsearch_password} \
    ${ec_deployment.custom-deployment-id.kibana[0].https_endpoint}/api/saved_objects/_import?overwrite=true \
    -H \"kbn-xsrf: true\" --form file=@dashboard.ndjson"
  }

  depends_on = [ec_deployment.custom-deployment-id]
  triggers = {
    dashboard_shal = "${sha1(file("dashboard.ndjson"))}"
  }
}
```

PS: Azure Package

Independent provider – more Terraform ❤️

[https://registry.terraform.io/providers/hashicorp/azurearm/latest/docs/
resources/elastic_cloud_elasticsearch](https://registry.terraform.io/providers/hashicorp/azurearm/latest/docs/resources/elastic_cloud_elasticsearch)



elastic



Terraform

Philipp Krenn

@xeraa