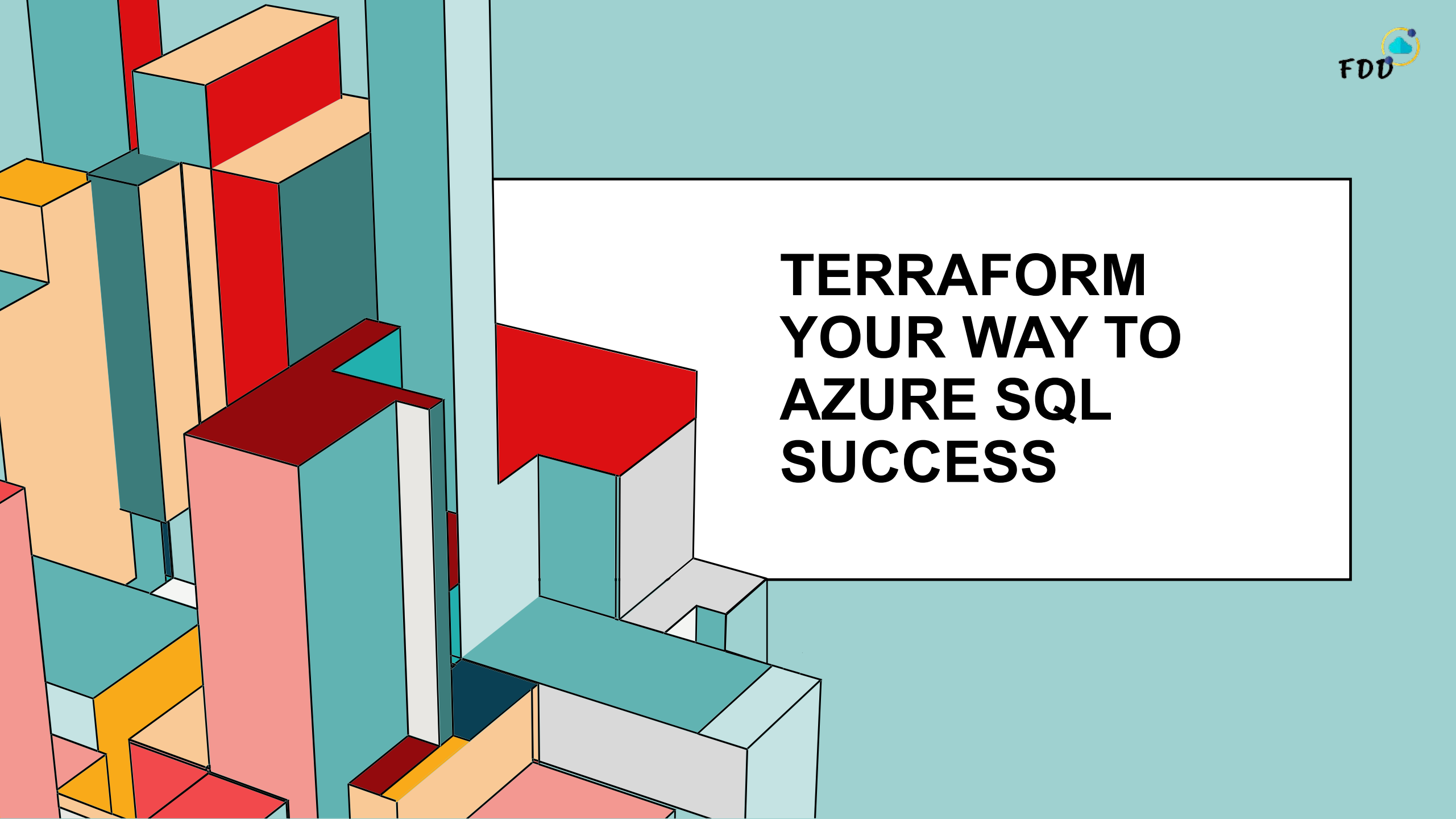

FDD
TERRAFORM
YOUR WAY TO
AZURE SQL
SUCCESS

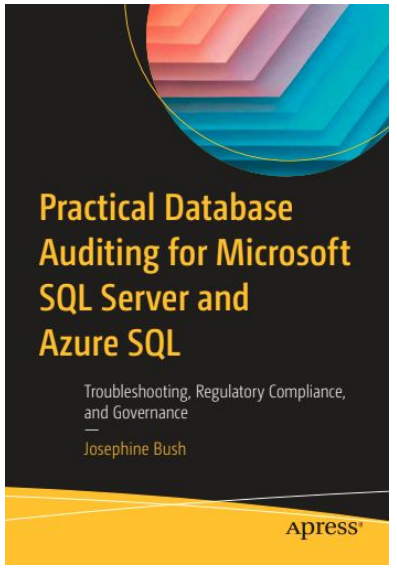
ABOUT ME

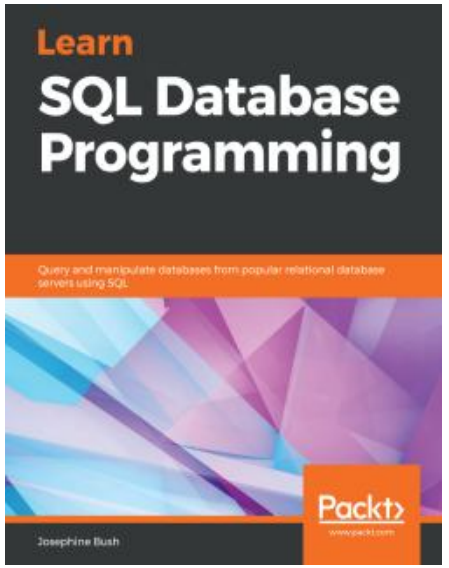
Josephine Bush

Microsoft® Most Valuable Professional

FEATURED IN THE redgate 100 MOST INFLUENTIAL IN THE DATABASE COMMUNITY 2022 redgate

@hellosqlkitty
sqlkitty.com





WHAT IS TERRAFORM?

CODE DRIVEN INFRASTRUCTURE

Terraform allows infrastructure provisioning through code.

RESOURCE LIFECYCLE AUTOMATION

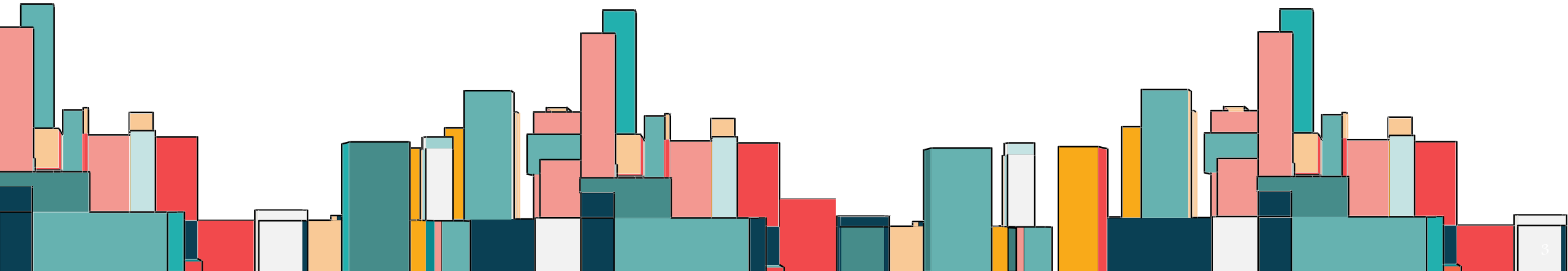
Terraform automates resource creation, updates, and destruction.

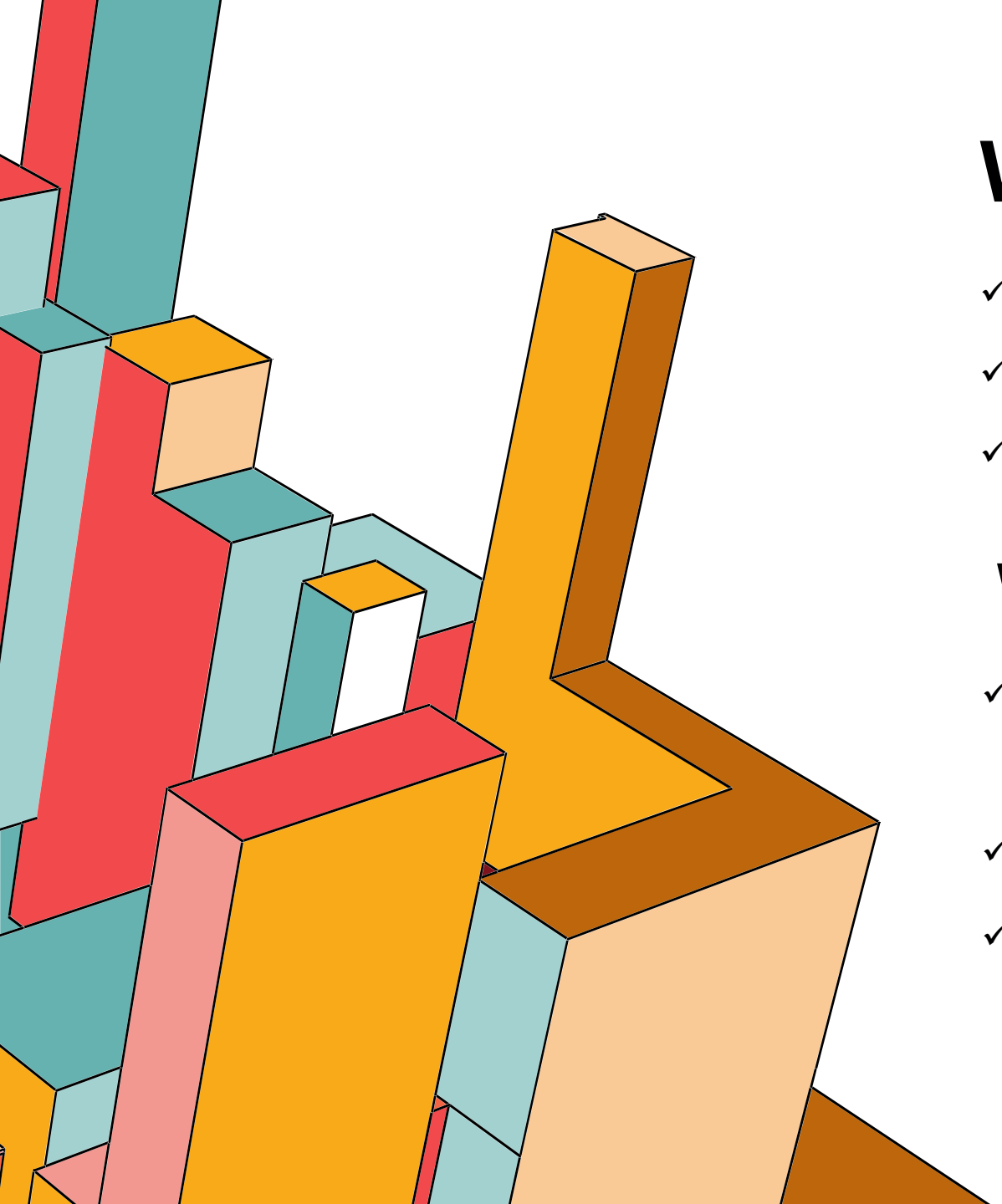
CLOUD AGNOSTICISM

It works with multiple cloud providers and on-premises systems.

EXTENSIVE ECOSYSTEM

It boasts a rich ecosystem of community-contributed modules and plugins for extending functionality and reuse.





WHAT YOU NEED

- ✓ TERRAFORM
- ✓ AZURE ACCOUNT
- ✓ AZURE CLI

WHAT YOU WANT

- ✓ VS CODE
  - ✓ With Terraform extension
- ✓ POWERSHELL
- ✓ GITHUB REPO

TERRAFORM FILES

MAIN.TF

Often used to define the primary configuration for your infrastructure resources

PROVIDERS.TF

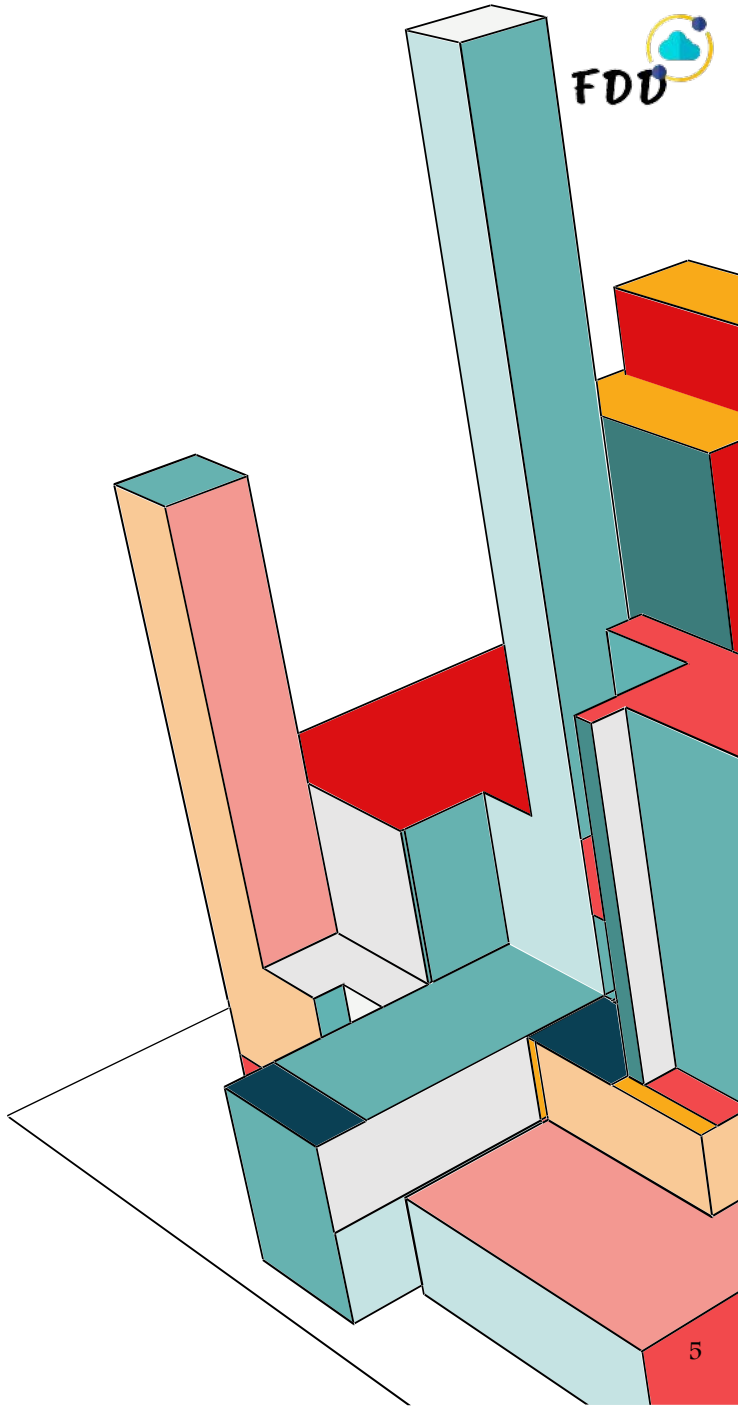
Separate providers.tf file can help keep your project more organized and maintainable.

OUTPUT.TF

Used to define output values that provide information about the infrastructure resources created by your Terraform configuration

VARIABLES.TF

It's a best practice to use this file to define input variables that allow you to parameterize your Terraform configurations.



MAIN.TF

main.tf > ...

```
1   #creates resource group
2   resource "random_pet" "rg_name" {
3     prefix = var.resource_group_name_prefix
4   }
5
6   resource "azurerm_resource_group" "rg" {
7     location = var.resource_group_location
8     name     = random_pet.rg_name.id
9   }
```

PROVIDERS.TF

providers.tf > ...

```
terraform {
  required_version = ">=0.12"

  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = ">= 3.52.0"
    }

    random = {
      source  = "hashicorp/random"
      version = "~>3.0"
    }
  }
}

provider "azurerm" {
  features {
  }
}
```

VARIABLES.TF

variables.tf > ...

```
1   variable "resource_group_location" {
2     default     = "eastus2"
3     description = "Location of the resource group."
4   }
5
6   variable "resource_group_name_prefix" {
7     default     = "rg"
8     description = "Prefix of the resource group name that's combined with a random ID so name is unique in your Azure subs
9   }
10
```

OUTPUT.TF

output.tf > ...

```
output "resource_group_name" {
  value = azurerm_resource_group.rg.name
}

output "sql_server_fqdn" {
  value = azurerm_mssql_server.example.fully_qualified_domain_name
}

output "database_name" {
  value = azurerm_mssql_database.example.name
}
```

TERRAFORM AZURE SQL FILES

SQLDB.TF

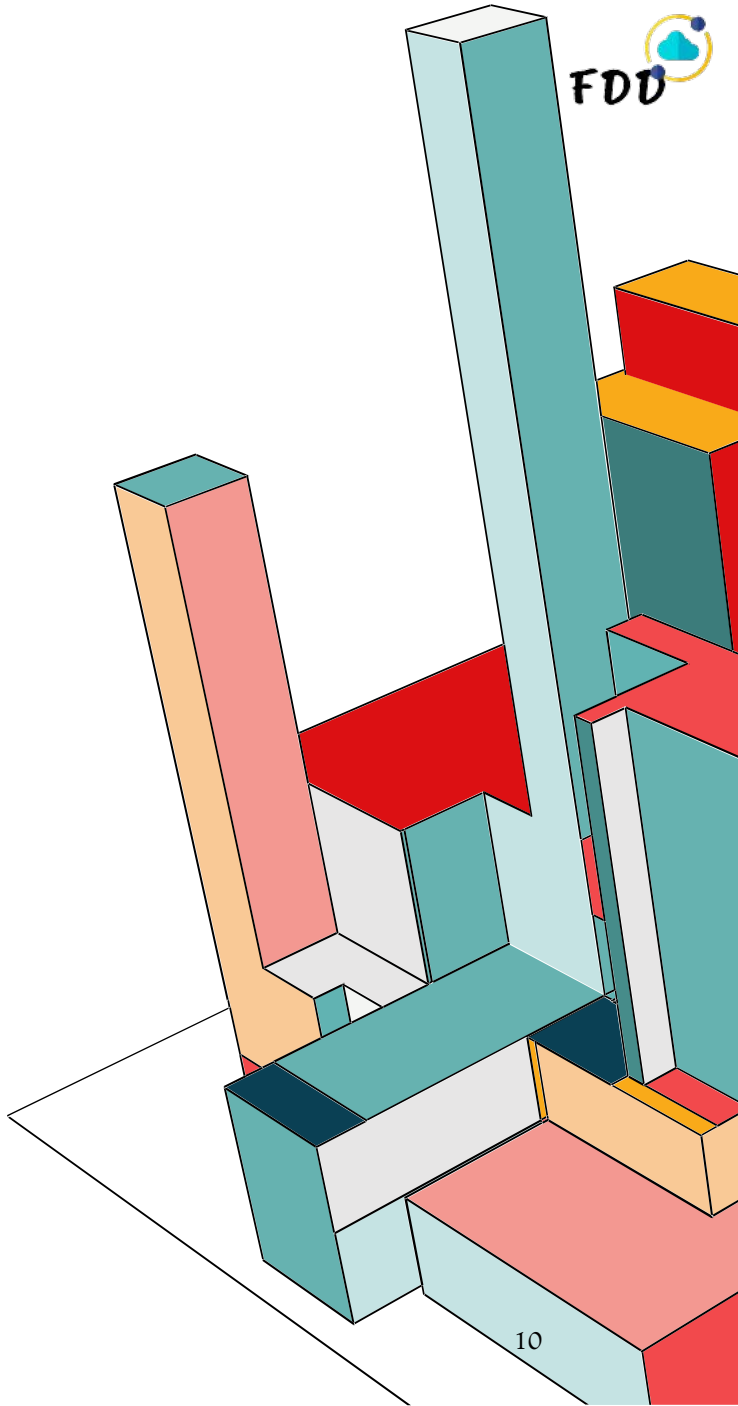
For creating my Azure SQL Server, its dbs, and firewall rules

ALERTS.TF

For creating the alerts on my Azure SQL dbs

AUDIT.TF

For creating the audits associated with my SQL dbs



SQLDB.TF CREATE SQL SERVER/DB

sqldb.tf > ...

```
# creates the azure sql server
resource "azurerm_mssql_server" "example" {
  name                         = "sql-${azurerm_resource_group.rg.name}"
  resource_group_name          = random_pet.rg_name.id
  location                     = var.resource_group_location
  version                      = "12.0"
  administrator_login          = "sqladmin"
  administrator_login_password = "password@123!"

  azuread_administrator {
    login_username = "jb_msn.com#EXT#@hellosqlkittygmail.onmicrosoft.com"
    object_id      = "edd56623-e123-43ef-b847-71443ac454a0"
  }

  depends_on = [
    azurerm_resource_group.rg
  ]
}
```

```
# creates the azure sql db
resource "azurerm_mssql_database" "example" {
  name                         = "db-${azurerm_resource_group.rg.name}"
  server_id                    = azurerm_mssql_server.example.id
  create_mode                  = "Default"
  sku_name                     = "Basic"
  collation                    = "SQL_Latin1_General_CP1_CI_AS"
  depends_on = [
    azurerm_mssql_server.example
  ]
}
```

```
# enables access to your db.  change out the IP to match your IP.
resource "azurerm_mssql_firewall_rule" "example" {
  name                = "my-ip"
  server_id          = azurerm_mssql_server.example.id
  start_ip_address    = "11.11.11.11"
  end_ip_address      = "11.11.11.11"
  depends_on = [
    azurerm_mssql_server.example
  ]
}

resource "azurerm_mssql_firewall_rule" "azure-services" {
  name                = "allow-azure-services"
  server_id          = azurerm_mssql_server.example.id
  start_ip_address    = "0.0.0.0"
  end_ip_address      = "0.0.0.0"
  depends_on = [
    azurerm_mssql_server.example
  ]
}
```

ALERTS.TF CREATE SQL ALERTS

```
# creates alert action group
resource "azurerm_monitor_action_group" "ag" {
  name                = "dbactiongroup"
  resource_group_name = random_pet.rg_name.id
  short_name          = "dbactgrp"

  email_receiver {
    name                    = "sendtome"
    email_address           = "hellosqlkitty@gmail.com"
    use_common_alert_schema = true
  }
  depends_on = [
     azurerm_mssql_database.example
   ]

}
```

```
# creates alert for max dtu 80%
resource "azurerm_monitor_metric_alert" "alertdtu80" {
  name                = "db-DTUalertMax80"
  resource_group_name = random_pet.rg_name.id
  scopes              = ["/subscriptions/244eb28e-a9b8"]
  description         = "db DTU alert greater than 80%"
  target_resource_type = "Microsoft.Sql/servers/databases"
  target_resource_location = var.resource_group_location
  severity            = 2

  criteria {
    metric_namespace = "Microsoft.Sql/servers/databases"
    metric_name      = "dtu_consumption_percent"
    aggregation      = "Maximum"
    operator         = "GreaterThan"
    threshold        = 80
  }

  action {
    action_group_id = azurerm_monitor_action_group.ag.id
  }
  depends_on = [
     azurerm_mssql_database.example
   ]
}
```

AUDIT.TF CREATE SQL SERVER AUDIT

```hcl
/* LAW to hold auditing data*/
resource "azurerm_log_analytics_workspace" "example" {
  name                = "law-${azurerm_resource_group.rg.name}"
  location            = var.resource_group_location
  resource_group_name = random_pet.rg_name.id
  sku                 = "PerGB2018"
  retention_in_days   = 30
}
```

```hcl
/* auditing setup */
resource "azurerm_monitor_diagnostic_setting" "example" {
  name                       = "ds-${azurerm_resource_group.rg.name}"
  target_resource_id         = "${azurerm_mssql_server.example.id}/databases/master"
  log_analytics_workspace_id = azurerm_log_analytics_workspace.example.id

  enabled_log {
    category = "SQLSecurityAuditEvents"
    # enabled  = true

    retention_policy {
      enabled = false
    }
  }

  metric {
    category = "AllMetrics"

    retention_policy {
      enabled = false
    }
  }

  lifecycle {
    ignore_changes = [log, metric]
  }
}
```

```hcl
resource "azurerm_mssql_database_extended_auditing_policy" "example" {
  database_id            = "${azurerm_mssql_server.example.id}/databases/master"
  log_monitoring_enabled = true
  depends_on = [
    azurerm_mssql_database.example
  ]
}

resource "azurerm_mssql_server_extended_auditing_policy" "example" {
  server_id              = azurerm_mssql_server.example.id
  log_monitoring_enabled = true
}
```

TERRAFORM LIFECYCLE

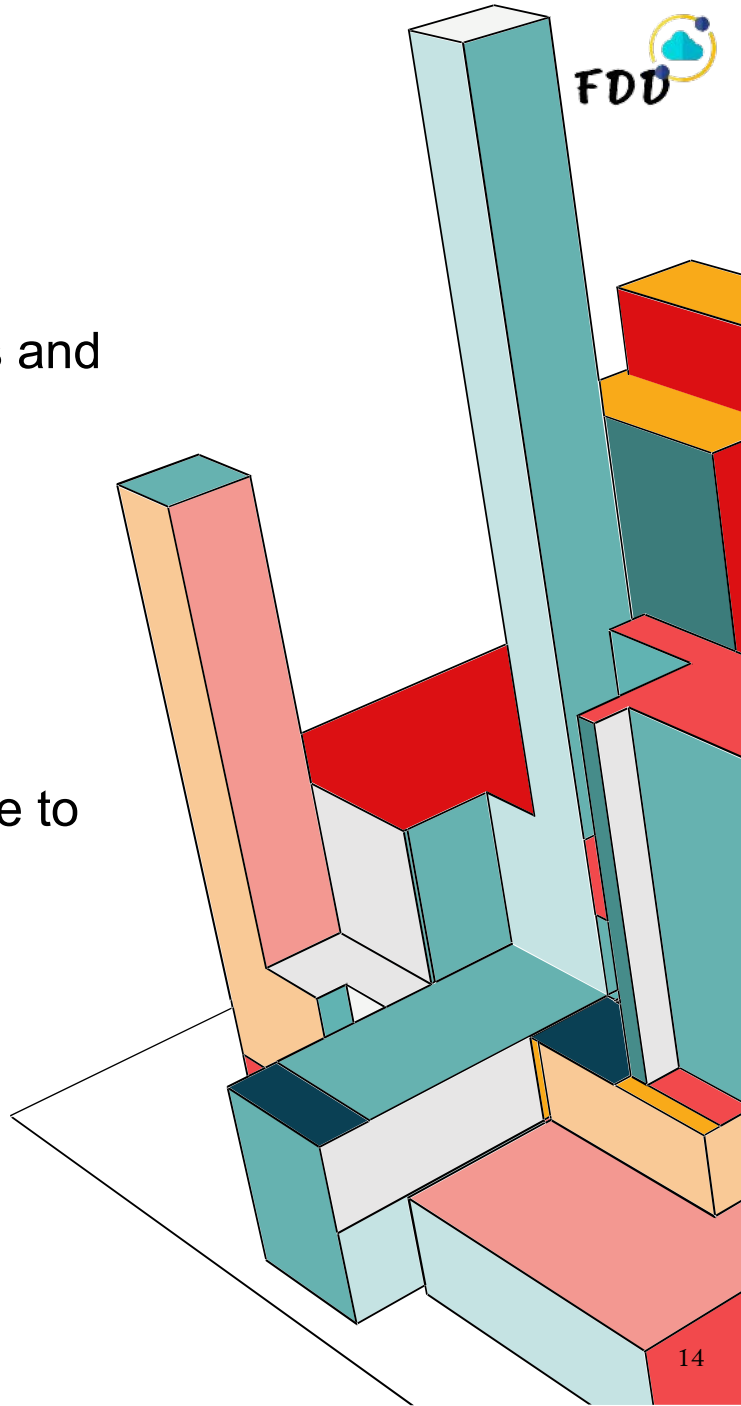
- **Init (terraform init)**
  Initializes a Terraform working directory by downloading provider plugins and setting up local state like terraform.tfstate

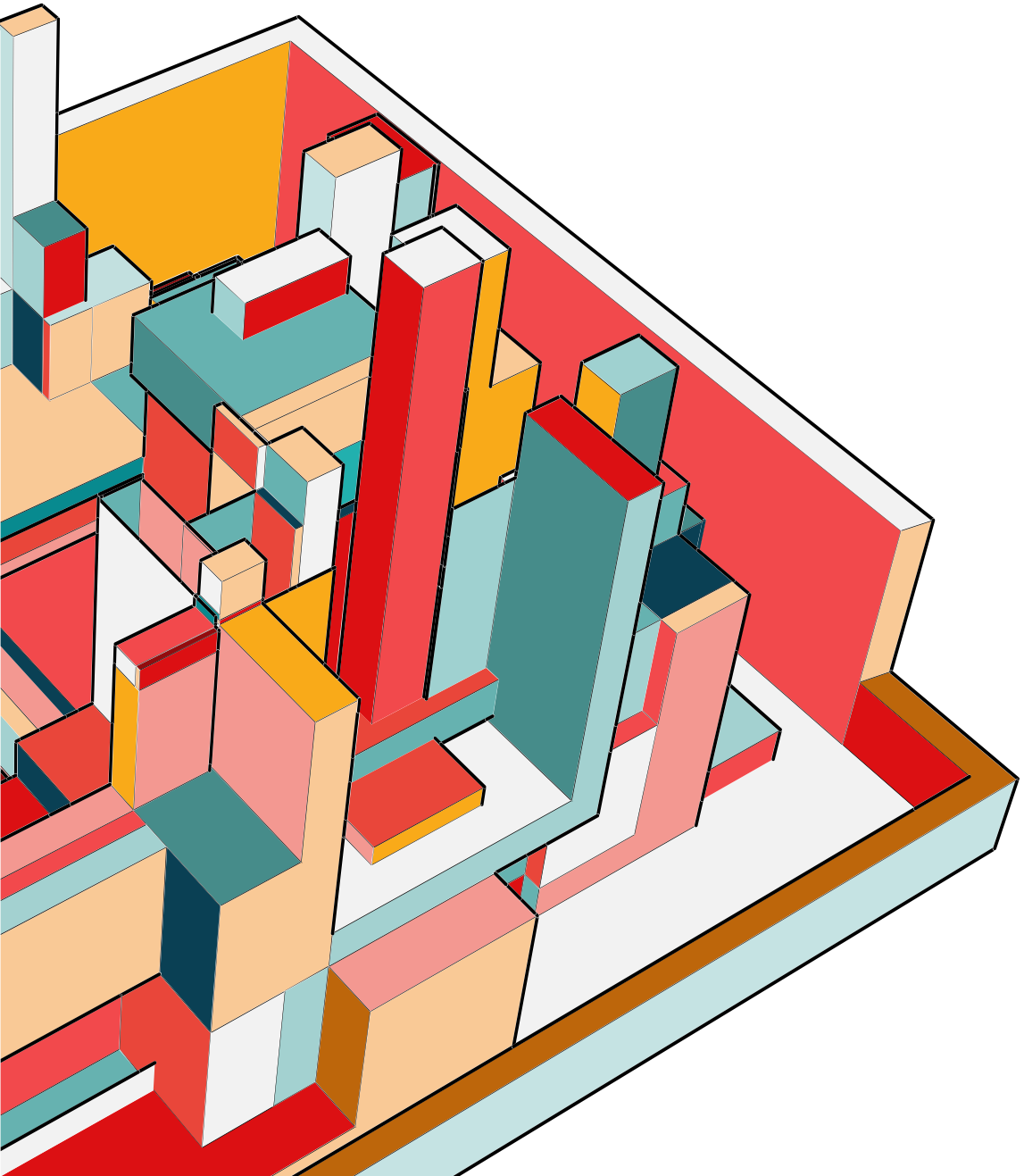
- **Validate (terraform validate)**
  Checks the syntax and validity of your Terraform configuration files.

- **Plan (terraform plan)**
- Generates an execution plan to preview the changes Terraform will make to your infrastructure.

- **Apply (terraform apply)**
  Executes the changes defined in your Terraform configuration to create, update, or delete resources.

- **Destroy (terraform destroy)**
  Destroys all the resources managed by Terraform in your configuration.

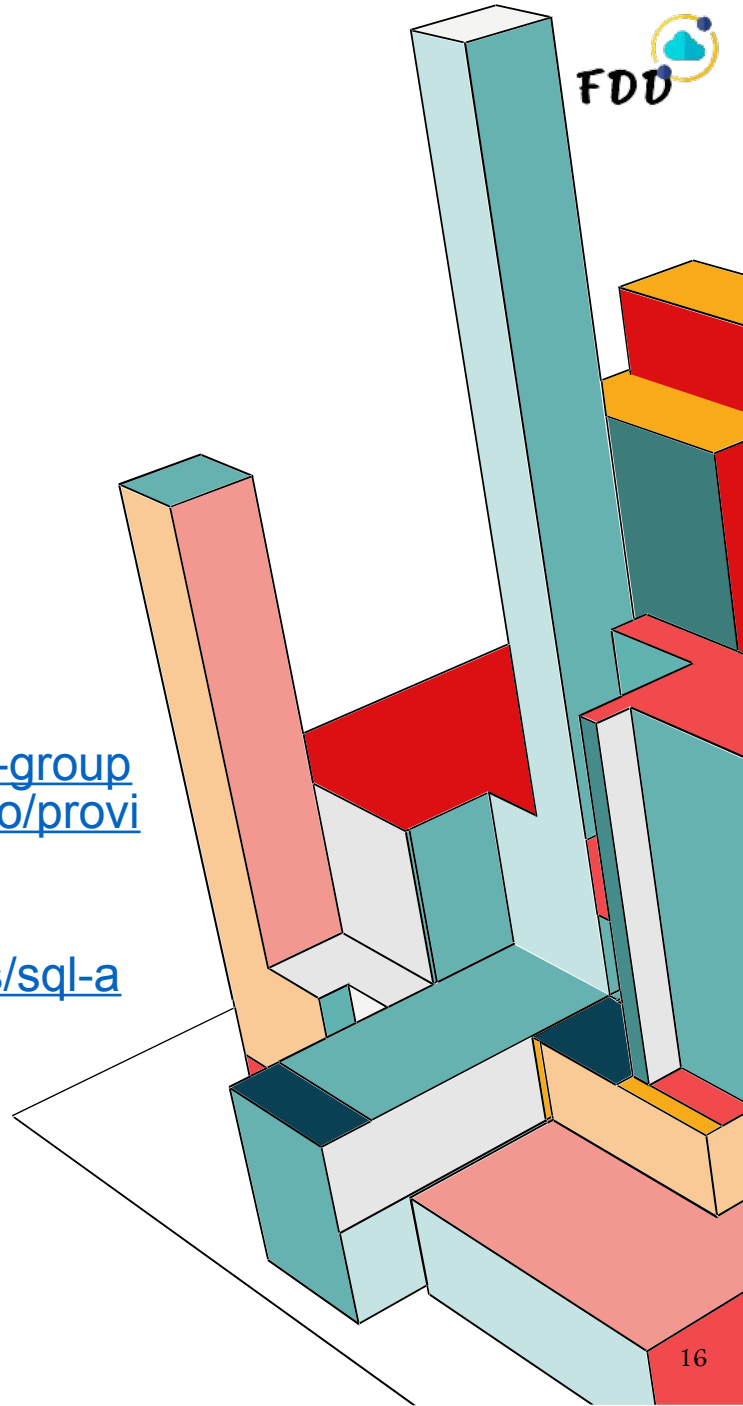




DEMO

RESOURCES

- https://sqlkitty.com/new-job-week-3-creating-alerts/
- https://sqlkitty.com/new-job-week-4-set-up-auditing/
- https://github.com/sqlkitty/terraform
- https://developer.hashicorp.com/terraform/tutorials/azure-get-started
- https://learn.microsoft.com/en-us/azure/developer/terraform/create-resource-group?tabs=azure-cli#implement-the-terraform-code%20https://registry.terraform.io/providers/hashicorp/random/latest/docs/resources/pet
- https://github.com/hashicorp/terraform-provider-azurerm/tree/main/examples/sql-azure



THANK YOU

Contact me
@hellosqlkitty or
visit me at
sqlkitty.com