Technical datasheet of terraform module for Dynamodb version 1.0.

This procedure only works to create Dynamodb tables through Terraform and comply with the security controls of PPG IT.

This documentation does not include items or aspects regarding the VPC gateway endpoints whose specific work must made through the cloud teams.

It starts with a local machine folder in this case it’s already synchronized with a personal git hub repository.

PS C:\Users\e283131\OneDrive - PPG Industries, Inc\!My Files\local\KRIVEGAA\Dynamodb\Tf\_dynamodb\_dev\Dynamodb>



Mode LastWriteTime Length Name

---- ------------- ------ ----

da---l 01/09/2023 03:26 p. m. .terraform

da---l 01/09/2023 04:54 p. m. **modules**

-a---l 01/09/2023 03:26 p. m. 1409 .terraform.lock.hcl

-a---l 10/08/2023 02:32 p. m. 2151 **main.tf**

-a---l 01/09/2023 04:58 p. m. 180 terraform.tfstate

-a---l 01/09/2023 04:58 p. m. 2937 terraform.tfstate.backup

From here the more important file is the above *main.tf* file that contains the creation of a principal table and the modules folder that also contains a couple of modules corresponding to autoscaling table options and the table’s accelerator dax.

1. Once we’re connected to vs code with either your appropriate Windows user and we already stay in the correct folder run the first main.tf file for the creation of dynamodb table as shown below.

Commands necessary of terraform to create the dynamodb table:

terraform init

terraform plan

terraform apply –auto-approve

*Ensure you already have switched to the appropriate Windows user with the grants and role for creating resources in AWS.*

Code of Dynamodb\main.tf:

|  |
| --- |
| terraform {  required\_providers {  aws = {  source = "hashicorp/aws"  version = "~> 5.11.0"  }  }  }  #configure the AWS Provider  provider "aws" {  region = "us-east-1"  profile = "PPG\_Developer\_ReadWrite-487517618172"  }  resource "aws\_dynamodb\_table" "basic-dynamodb-table" {  name = "GameScores\_Test\_iva"  billing\_mode = "PROVISIONED"  read\_capacity = 20  write\_capacity = 20  hash\_key = "UserId"  range\_key = "GameTitle"  deletion\_protection\_enabled = "true" #Enable deletion protection for DynamoDB tables, In case you need to destroy the table with this same script you must first deactivate the current deletion protection for this table.  attribute {  name = "UserId"  type = "S"  }  attribute {  name = "GameTitle"  type = "S"  }  attribute {  name = "TopScore"  type = "N"  }  ttl {  attribute\_name = "TimeToExist"  enabled = false  }  #Ensure DynamoDB Tables are encrypted using AWS managed keys (KMS)  server\_side\_encryption {  enabled = true  }  #Configure DynamoDB tables to have point-in-time recovery (PITR) enabled  point\_in\_time\_recovery {  enabled = true  }  lifecycle {  ignore\_changes = [ write\_capacity, read\_capacity ]  }  global\_secondary\_index {  name = "GameTitleIndex"  hash\_key = "GameTitle"  range\_key = "TopScore"  write\_capacity = 10  read\_capacity = 10  projection\_type = "INCLUDE"  non\_key\_attributes = ["UserId"]  }    tags = {  Name = "dynamodb-table-dev\_test\_ivega"  Environment = "DEV"  }  }  output "aws\_dynamodb\_table" {  value = aws\_dynamodb\_table.basic-dynamodb-table.id    }  #in case you need to use DAX you should have in advance an aws\_ dax\_subnet\_group  resource "aws\_dax\_subnet\_group" "sng-test-iva" {  name = "daxsng-test-iva"  subnet\_ids = [  "subnet-0bc4eeab5975373ef",  "subnet-0cf1b2a274a3a0f85" ]  }  output "aws\_dax\_subnet\_group" {  value = aws\_dax\_subnet\_group.sng-test-iva.id  } |

1. Execute terraform init, plan & apply –auto-approve. We are using autoapprove because we are executing from a normal domain account and these will always skip to our operator accounts that have the AWS role:

A screenshot of a computer program

Description automatically generated







A computer screen shot of a program code

Description automatically generated

1. After you already created the table and subnet group for dax, you should move to the folder “modules” where the last modules autoscaling and Dax are.

[localhost]: PS C:\Users\e283131\OneDrive - PPG Industries, Inc\!My Files\local\KRIVEGAA\Dynamodb\Tf\_dynamodb\_dev\Dynamodb\modules>



Mode LastWriteTime Length Name

---- ------------- ------ ----

da---l 01/09/2023 03:26 p. m. .terraform

da---l 03/08/2023 04:06 p. m. **autoscaling**

da---l 01/09/2023 03:27 p. m. **dax**

-a---l 01/09/2023 03:26 p. m. 1409 .terraform.lock.hcl

-a---l 09/08/2023 06:53 p. m. 1000 **main.tf**

-a---l 01/09/2023 04:54 p. m. 181 terraform.tfstate

-a---l 01/09/2023 04:54 p. m. 956 terraform.tfstate.backup

The above folder has its own *main.tf* file which controls and runs Dax and the autoscaling module, from here you must run terraform init, plan, and apply again so we can create these.

A screenshot of a computer program

Description automatically generated

Code of Dynamodb\modules\main.tf.

|  |
| --- |
| terraform {  required\_providers {  aws = {  source = "hashicorp/aws"  version = "~> 5.11.0"  }  }  }  #configure the AWS Provider  provider "aws" {  region = "us-east-1"  profile = "PPG\_Developer\_ReadWrite-487517618172"  }  module "autoscaling-control" {  source = "./autoscaling"  autoscaling-control = "Mod-Autoscaling"  }  module "dax-control" {  source = "./dax"  dax-control = "Mod-DAX"  }  output "autoscaling-control-1" {  value = module.autoscaling-control.module\_output\_1  }  output "autoscaling-control-2" {  value = module.autoscaling-control.module\_output\_2  }  output "aws\_dax\_parameter\_group-3" {  value = module.dax-control.module\_output\_3  }  output "aws\_dax\_cluster-4" {  value = module.dax-control.module\_output\_4  }  output "autoscaling-control-5" {  value = module.autoscaling-control.module\_output\_5  }  output "autoscaling-control-6" {  value = module.autoscaling-control.module\_output\_6  } |

Code of module set-auto-scaling.

|  |
| --- |
| #Configure DynamoDB tables to scale capacity automatically with demand  variable "autoscaling-control" {  type = string  }  resource "aws\_appautoscaling\_target" "dynamodb\_table\_read\_target" {  max\_capacity = 100  min\_capacity = 20  resource\_id = "table/GameScores\_Test\_iva"  scalable\_dimension = "dynamodb:table:ReadCapacityUnits"  service\_namespace = "dynamodb"  }  resource "aws\_appautoscaling\_policy" "dynamodb\_table\_read\_policy" {  name = "DynamoDBReadCapacityUtilization:${aws\_appautoscaling\_target.dynamodb\_table\_read\_target.resource\_id}"  policy\_type = "TargetTrackingScaling"  resource\_id = aws\_appautoscaling\_target.dynamodb\_table\_read\_target.resource\_id  scalable\_dimension = aws\_appautoscaling\_target.dynamodb\_table\_read\_target.scalable\_dimension  service\_namespace = aws\_appautoscaling\_target.dynamodb\_table\_read\_target.service\_namespace  target\_tracking\_scaling\_policy\_configuration {  predefined\_metric\_specification {  predefined\_metric\_type = "DynamoDBReadCapacityUtilization"  }  target\_value = 70  }  }  resource "aws\_appautoscaling\_target" "dynamodb\_table\_write\_target" {  max\_capacity = 200  min\_capacity = 5  resource\_id = "table/GameScores\_Test\_iva"  scalable\_dimension = "dynamodb:table:WriteCapacityUnits"  service\_namespace = "dynamodb"  }  resource "aws\_appautoscaling\_policy" "dynamodb\_table\_write\_policy" {  name = "DynamoDBWriteCapacityUtilization:${aws\_appautoscaling\_target.dynamodb\_table\_write\_target.resource\_id}"  policy\_type = "TargetTrackingScaling"  resource\_id = aws\_appautoscaling\_target.dynamodb\_table\_write\_target.resource\_id  scalable\_dimension = aws\_appautoscaling\_target.dynamodb\_table\_write\_target.scalable\_dimension  service\_namespace = aws\_appautoscaling\_target.dynamodb\_table\_write\_target.service\_namespace  target\_tracking\_scaling\_policy\_configuration {  predefined\_metric\_specification {  predefined\_metric\_type = "DynamoDBWriteCapacityUtilization"  }  target\_value = 70  }  }  resource "aws\_appautoscaling\_target" "dynamodb\_index\_read\_target" {  max\_capacity = 100  min\_capacity = 5  resource\_id = "table/GameScores\_Test\_iva/index/GameTitleIndex"  scalable\_dimension = "dynamodb:index:ReadCapacityUnits"  service\_namespace = "dynamodb"  }  resource "aws\_appautoscaling\_target" "dynamodb\_index\_write\_target" {  max\_capacity = 100  min\_capacity = 5  resource\_id = "table/GameScores\_Test\_iva/index/GameTitleIndex"  scalable\_dimension = "dynamodb:index:WriteCapacityUnits"  service\_namespace = "dynamodb"  }  output "module\_output\_1" {  value = aws\_appautoscaling\_policy.dynamodb\_table\_read\_policy.id  }  output "module\_output\_2" {  value = aws\_appautoscaling\_policy.dynamodb\_table\_write\_policy.id  }  output "module\_output\_5" {  value = aws\_appautoscaling\_target.dynamodb\_index\_read\_target.scalable\_dimension  }  output "module\_output\_6" {  value = aws\_appautoscaling\_target.dynamodb\_index\_write\_target.scalable\_dimension  } |

Code of module set-dax.

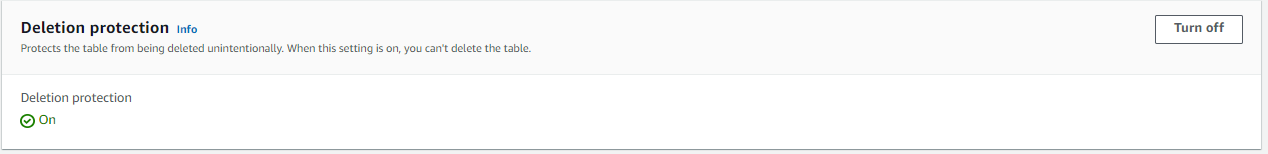
|  |
| --- |
| variable "dax-control" {  type = string  }  resource "aws\_dax\_parameter\_group" "pg-test-iva" {  name = "daxpg-test-iva"  parameters {  name = "query-ttl-millis"  value = "10000"  }  parameters {  name = "record-ttl-millis"  value = "10000"  }  }  resource "aws\_dax\_cluster" "cl-test-iva" {  cluster\_name = "dax-cl-test-iva"  iam\_role\_arn = "arn:aws:iam::487517618172:role/ppg-role-dax-dba"  node\_type = "dax.t2.small"  replication\_factor = 3  subnet\_group\_name = "daxsng-test-iva"  cluster\_endpoint\_encryption\_type = "TLS"  security\_group\_ids = ["sg-0e218bf4f343724b0"]  #Configure DynamoDB Accelerator (DAX) clusters to encrypt data at rest AND in transit  server\_side\_encryption {  enabled = true  }  }  output "module\_output\_3" {  value = aws\_dax\_parameter\_group.pg-test-iva.id  }  output "module\_output\_4" {  value = aws\_dax\_cluster.cl-test-iva.id  } |

A screenshot of a computer

Description automatically generated

Considerations to destroy resources with the same terraform script:

* You must unset deletion protection in the table.



* You must unlink the parameter group that we created previously from the set-dax.tf of this Dax resource.

A screenshot of a computer

Description automatically generated

Please follow the following order in case you need to destroy these resources.

[localhost]: PS C:\Users\e283131\OneDrive - PPG Industries, Inc\!My Files\local\KRIVEGAA\Dynamodb\Tf\_dynamodb\_dev\Dynamodb\**modules>** terraform destroy --auto-approve

[localhost]: PS C:\Users\e283131\OneDrive - PPG Industries, Inc\!My Files\local\KRIVEGAA\Dynamodb\Tf\_dynamodb\_dev\**Dynamodb>** terraform destroy --auto-approve