**We are going to create three resources in our Terraform template. The first will be an account level billing alarm just in case you don’t already have one. Then we will create two budget resources for our EC2 and RDS forecasted spend.**

**Our account level billing alarm:**

provider "aws" {

region = "us-east-1"

}

resource "aws\_cloudwatch\_metric\_alarm" "account-billing-alarm" {

alarm\_name = "account-billing-alarm"

comparison\_operator = "GreaterThanOrEqualToThreshold"

evaluation\_periods = "1"

metric\_name = "EstimatedCharges"

namespace = "AWS/Billing"

period = "21600"

statistic = "Average"

threshold = "100"

alarm\_description = "Billing alarm by account"

alarm\_actions = ["<your-sns-topic-arn-for-notification>"]

dimensions {

Currency = "USD"

LinkedAccount = "<your-aws-account-id>"

}

}

**Once our template is initialized we can apply our template to our AWS account.**

**$ terraform apply**

An execution plan has been generated and is shown below.

Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

+ aws\_cloudwatch\_metric\_alarm.account-billing-alarm

....

....

aws\_cloudwatch\_metric\_alarm.account-billing-alarm: Creating...

actions\_enabled: "" => "true"

alarm\_actions.#: "" => "1"

alarm\_actions.321893454: "" => "<your-sns-topic-arn-for-notification>"

alarm\_description: "" => "Billing alarm by account"

alarm\_name: "" => "account-billing-alarm"

arn: "" => "<computed>"

comparison\_operator: "" => "GreaterThanOrEqualToThreshold"

dimensions.%: "" => "2"

dimensions.Currency: "" => "USD"

dimensions.LinkedAccount: "" => "<your-aws-account-id>"

evaluate\_low\_sample\_count\_percentiles: "" => "<computed>"

evaluation\_periods: "" => "1"

metric\_name: "" => "EstimatedCharges"

namespace: "" => "AWS/Billing"

period: "" => "21600"

statistic: "" => "Average"

threshold: "" => "100"

treat\_missing\_data: "" => "missing"

aws\_cloudwatch\_metric\_alarm.account-billing-alarm: Creation complete after 1s (ID: account-billing-alarm)

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

**We now have our account level billing alarm to notify us when our estimated bill is going to be greater than $100 USD. Let’s create our budgets so we can alarm when our forecasted spend on EC2 or RDS exceeds our expectation.**

**Add the following resources to your Terraform template.**

resource "aws\_budgets\_budget" "ec2-forecast-alarm" {

name = "budget-ec2-monthly"

budget\_type = "COST"

limit\_amount = "50"

limit\_unit = "USD"

time\_period\_start = "2020-01-01\_00:00"

time\_unit = "MONTHLY"

cost\_filters {

service = "Amazon Elastic Compute Cloud - Compute"

}

}

resource "aws\_budgets\_budget" "rds-forecast-alarm" {

name = "budget-rds-monthly"

budget\_type = "COST"

limit\_amount = "40"

limit\_unit = "USD"

time\_period\_start = "2020-01-01\_00:00"

time\_unit = "MONTHLY"

cost\_filters {

service = "Amazon Relational Database Service"

}

}

**Here we have our budgets for EC2 and RDS.**

**The cost\_filters are using the fully qualified names for each service, this is what AWS budgets via Terraform expect. We fire “apply” command again to create our new budgets.**

$ terraform apply

An execution plan has been generated and is shown below.

Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

+ aws\_budgets\_budget.ec2-forecast-alarm

id: <computed>

account\_id: <computed>

budget\_type: "COST"

cost\_filters.%: "1"

cost\_filters.Service: "ec2"

cost\_types.#: <computed>

limit\_amount: "50"

limit\_unit: "USD"

name: "budget-ec2-monthly"

name\_prefix: <computed>

time\_period\_end: "2089-06-15\_00:00"

time\_period\_start: "2020-01-01\_00:00"

time\_unit: "MONTHLY"

+ aws\_budgets\_budget.rds-forecast-alarm

id: <computed>

account\_id: <computed>

budget\_type: "COST"

cost\_filters.%: "1"

cost\_filters.Service: "rds"

cost\_types.#: <computed>

limit\_amount: "40"

limit\_unit: "USD"

name: "budget-rds-monthly"

name\_prefix: <computed>

time\_period\_end: "2089-06-15\_00:00"

time\_period\_start: "2020-01-01\_00:00"

time\_unit: "MONTHLY"

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

**We now have two budgets configured in our AWS account. The first tracks our spend on EC2 and the latter tracks our spend on RDS. However, we do not have notifications set up for these yet.**

**Adding notifications to our budgets:**

Because of the limitation of Terraform as of today, we need to add the notifications to our budgets outside of our template. We can quickly add notifications to each of our budgets by using the AWS CLI.

$ aws budgets create-notification --account-id <your-aws-account-id> --budget-name budget-rds-monthly --notification NotificationType=FORECASTED,ComparisonOperator=GREATER\_THAN,Threshold=100,ThresholdType=PERCENTAGE --subscribers SubscriptionType=EMAIL,Address=<your-email-address>

$ aws budgets create-notification --account-id <your-aws-account-id> --budget-name budget-ec2-monthly --notification NotificationType=FORECASTED,ComparisonOperator=GREATER\_THAN,Threshold=100,ThresholdType=PERCENTAGE --subscribers SubscriptionType=EMAIL,Address=<your-email-address>