**✅ Terraform is detecting changes in 11 resources, including:**

* 5 google\_monitoring\_alert\_policy resources (for different alert types)
* 1 google\_monitoring\_notification\_channel (email for hsbc.devopsteam@hsbc.com)
* Several time\_sleep and null\_resource blocks (used for dependency handling)

**✅ Notification channels exist and are referenced correctly, specifically:**

* xMatters webhook
* Email (hsbc.devopsteam@hsbc.com)

**✅ Alert policies are configured to include both xMatters and email notification channels**

For example, google\_monitoring\_alert\_policy.vm\_utilization includes:

hcl

CopyEdit

notification\_channels = [

"projects/hsbc-12609073-peakplat-dev/notificationChannels/10709580538556470038", # xMatters

"projects/hsbc-12609073-peakplat-dev/notificationChannels/11696354860105901907" # Email

]

**⚠️ Important Notes for Next Steps:**

1. **xMatters Notification Channel** is being created in this run:
   * It may cause a delay in propagation, which is why null\_resource.wait\_for\_xmatters\_channel and time\_sleep.wait\_for\_xmatters\_channel are used.
   * This sleep is important—make sure it's not removed unless you confirm propagation is complete before referencing it.
2. **Apply with Caution**:
   * If you're running this in a non-production environment, go ahead and terraform apply.
   * In prod, review all metric filters carefully to ensure they won’t cause excessive alerts (especially log-based metrics).
3. **Validation in GCP Console**:  
   After applying, confirm in GCP Monitoring (under Alerting) that:
   * Each alert policy is created.
   * Notification channels are listed and active (green tick).
   * Test an alert trigger (e.g. CPU spike or sample log) to verify xMatters and email delivery.

Let me know if you'd like help automating test alerts or writing documentation for what’s been deployed.

4o

**You said:**

I need to demo this code in client so can you please explain complete code with each point

**ChatGPT said:**

Absolutely — here’s a **complete and clear explanation** of your Terraform code for setting up GCP alerting with xMatters and email notifications, broken down **point by point**, so you're fully prepared to present this in a client demo:

**✅ Overview**

You're managing **GCP alerting infrastructure** using Terraform for the project hsbc-12609073-peakplat-dev. The alerting setup includes:

* **5 alerting policies** (for VMs, GKE, Cloud SQL, Flink, XDS)
* **Notification channels**:
  + **xMatters** (via webhook)
  + **Email** (hsbc.devopsteam@hsbc.com)
* **Automation logic** to ensure notification channels are fully created before alerts are associated.

**🔧 Detailed Breakdown**

**1. 📦 google\_monitoring\_notification\_channel — xMatters**

hcl

CopyEdit

resource "google\_monitoring\_notification\_channel" "xmatters\_webhook" {

display\_name = "xMatters Webhook"

type = "webhook\_tokenauth"

labels = {

url = var.xmatters\_webhook\_url

}

sensitive\_labels {

auth\_token = var.xmatters\_auth\_token

}

}

* **Purpose**: Sets up a GCP Monitoring notification channel that sends alerts to **xMatters** using a webhook URL and auth token.
* **Key Fields**:
  + url: xMatters integration endpoint
  + auth\_token: Secured using sensitive\_labels

**2. ⏱ time\_sleep & null\_resource — Wait for xMatters channel to be ready**

hcl

CopyEdit

resource "time\_sleep" "wait\_for\_xmatters\_channel" {

depends\_on = [google\_monitoring\_notification\_channel.xmatters\_webhook]

create\_duration = "60s"

}

hcl

CopyEdit

resource "null\_resource" "wait\_for\_xmatters\_channel" {

depends\_on = [time\_sleep.wait\_for\_xmatters\_channel]

}

* **Why Needed**: GCP may take several seconds to propagate the newly created xMatters channel.
* **Purpose**: Adds a **60-second wait** to ensure the channel is available before attaching it to alert policies.

**3. 📧 google\_monitoring\_notification\_channel — Email**

hcl

CopyEdit

resource "google\_monitoring\_notification\_channel" "email" {

display\_name = "DevOps Email"

type = "email"

labels = {

email\_address = "hsbc.devopsteam@hsbc.com"

}

}

* **Purpose**: Sends alerts to the **DevOps team via email**
* **Note**: This is a static channel and doesn’t require wait logic like xMatters.

**4. 📊 google\_monitoring\_alert\_policy Resources — Alert definitions**

Each alert policy monitors a specific component:

**a. VM Utilization Alert**

hcl

CopyEdit

resource "google\_monitoring\_alert\_policy" "vm\_utilization" {

display\_name = "VM Utilization Alert"

combiner = "OR"

conditions {

display\_name = "High CPU Utilization"

condition\_threshold {

filter = "metric.type=\"compute.googleapis.com/instance/cpu/utilization\" AND resource.type=\"gce\_instance\""

comparison = "COMPARISON\_GT"

threshold\_value = 0.8

duration = "60s"

aggregations {

alignment\_period = "60s"

per\_series\_aligner = "ALIGN\_MEAN"

}

}

}

notification\_channels = [

google\_monitoring\_notification\_channel.xmatters\_webhook.id,

google\_monitoring\_notification\_channel.email.id

]

depends\_on = [null\_resource.wait\_for\_xmatters\_channel]

}

* **Trigger**: CPU utilization > 80% for 1 minute
* **Scope**: GCE VMs (resource.type = "gce\_instance")
* **Notifies**: xMatters and email

**b. GKE Utilization Alert**

hcl

CopyEdit

resource "google\_monitoring\_alert\_policy" "gke\_utilization" {

display\_name = "GKE Utilization Alert"

combiner = "OR"

conditions {

display\_name = "High GKE CPU Utilization"

condition\_threshold {

filter = "metric.type=\"kubernetes.io/container/cpu/usage\_time\" AND resource.type=\"k8s\_container\""

comparison = "COMPARISON\_GT"

threshold\_value = 0.8

duration = "60s"

aggregations {

alignment\_period = "60s"

per\_series\_aligner = "ALIGN\_MEAN"

}

}

}

notification\_channels = [

google\_monitoring\_notification\_channel.xmatters\_webhook.id,

google\_monitoring\_notification\_channel.email.id

]

depends\_on = [null\_resource.wait\_for\_xmatters\_channel]

}

* **Trigger**: GKE CPU usage > 80%
* **Scope**: Kubernetes containers
* **Notifies**: xMatters and email

**c. Cloud SQL Utilization Alert**

h

CopyEdit

resource "google\_monitoring\_alert\_policy" "cloudsql\_utilization" {

display\_name = "Cloud SQL Utilization Alert"

combiner = "OR"

conditions {

display\_name = "High CPU Utilization"

condition\_threshold {

filter = "metric.type=\"cloudsql.googleapis.com/database/cpu/utilization\" AND resource.type=\"cloudsql\_database\""

comparison = "COMPARISON\_GT"

threshold\_value = 0.8

duration = "60s"

aggregations {

alignment\_period = "60s"

per\_series\_aligner = "ALIGN\_MEAN"

}

}

}

notification\_channels = [

google\_monitoring\_notification\_channel.xmatters\_webhook.id,

google\_monitoring\_notification\_channel.email.id

]

depends\_on = [null\_resource.wait\_for\_xmatters\_channel]

}

* **Trigger**: Cloud SQL CPU usage > 80%
* **Scope**: cloudsql\_database resources
* **Notifies**: xMatters and email

**d. Flink Logs Alert (Log-based)**

hcl

CopyEdit

resource "google\_monitoring\_alert\_policy" "flink\_logs" {

display\_name = "Flink Logs Alert"

combiner = "OR"

conditions {

display\_name = "Flink Error Logs"

condition\_threshold {

filter = "metric.type=\"logging.googleapis.com/user/flink\_error\_count\""

comparison = "COMPARISON\_GT"

threshold\_value = 0

duration = "60s"

aggregations {

alignment\_period = "60s"

per\_series\_aligner = "ALIGN\_SUM"

}

}

}

notification\_channels = [

google\_monitoring\_notification\_channel.xmatters\_webhook.id,

google\_monitoring\_notification\_channel.email.id

]

depends\_on = [null\_resource.wait\_for\_xmatters\_channel]

}

* **Trigger**: More than 0 error logs (from Flink)
* **Scope**: Log-based metric flink\_error\_count
* **Notifies**: xMatters and email

**e. XDS Logs Alert (Log-based)**

hcl

CopyEdit

resource "google\_monitoring\_alert\_policy" "xds\_logs" {

display\_name = "XDS Logs Alert"

combiner = "OR"

conditions {

display\_name = "XDS Error Logs"

condition\_threshold {

filter = "metric.type=\"logging.googleapis.com/user/xds\_error\_count\""

comparison = "COMPARISON\_GT"

threshold\_value = 0

duration = "60s"

aggregations {

alignment\_period = "60s"

per\_series\_aligner = "ALIGN\_SUM"

}

}

}

notification\_channels = [

google\_monitoring\_notification\_channel.xmatters\_webhook.id,

google\_monitoring\_notification\_channel.email.id

]

depends\_on = [null\_resource.wait\_for\_xmatters\_channel]

}

* **Trigger**: Any error in XDS logs
* **Notifies**: xMatters and email

**✅ Demo Walkthrough Tips**

Here’s how you can demo this to the client:

**💬 Start With:**

"This Terraform code sets up a robust alerting system in GCP for monitoring VMs, Kubernetes, Cloud SQL, and two critical logs—Flink and XDS. It integrates both xMatters and email to ensure our on-call team is notified immediately."

**📍 Key Points to Emphasize:**

* **Automation**: Everything is declaratively defined — repeatable and auditable.
* **xMatters Integration**: Critical for managing incident workflows.
* **Propagation Handling**: We use a sleep/wait mechanism to ensure dependencies are respected.
* **Flexibility**: Thresholds, resources, and notification targets can be adapted easily.

**✅ After Apply:**

Show the GCP Monitoring console to verify:

* Alerting policies appear
* Notification channels are attached
* Optionally simulate high CPU load or push error logs to show an alert firing.

Would you like a **slide summary deck** or **demo script** prepared for client presentation too?

4o