i want to make sure that "1.seperate monitoring workspace created for each applications in my central logging project.2. the dev user group has only the view access for the monitoring worsapces and dashboard where as sre user group has all the access.3. make sure service account factory and rbac implemented .4.make sure its working production ready code5.how can we change role for any business user.6.how can share the dashoboard or make it availble for only business users.7. we hve use the terraform enterprise edition so each stage folder having tfvars file with amin and variables files

ChatGPT

To create a production-ready monitoring setup with separate workspaces for each application, proper RBAC, and service account management using Terraform Enterprise, we need to implement the following:

1. Separate monitoring workspaces for each application.
2. View access for the dev user group.
3. Full access for the SRE user group.
4. Service account factory.
5. RBAC implementation.
6. Mechanism to change roles for business users.
7. Sharing dashboards with business users.
8. Terraform Enterprise structure with **main.tf**, **variables.tf**, and **\*.tfvars** files for each environment (dev and prod).

To meet your requirements, we need to set up a comprehensive Terraform configuration that addresses all points:

1. Create separate monitoring workspaces for each application in a central logging project.
2. Ensure that the dev user group has view-only access to the monitoring workspaces and dashboards, while the SRE user group has full access.
3. Implement service account management and RBAC.
4. Ensure the configuration is production-ready.
5. Provide a mechanism to change roles for business users.
6. Share dashboards or make them available only to business users.
7. Use Terraform Enterprise Edition with specific configuration files for each stage (dev, prod).

Here’s the detailed configuration:

please add the variable.tf in dev and stage ,i want to make sure "could you please mention how the dev user will having only the view access for the monitoring project and application created in therir seperate workspace where sre team has all the access ?how rbac help on this case based on the above design? "

ChatGPT

To ensure that development users have only view access to the monitoring project and application logs while the SRE team has full access, we can implement Role-Based Access Control (RBAC) using IAM roles in Google Cloud. In this design, the development users will be given read-only access, and the SRE team will have elevated permissions.

**Adding Variables for Different Environments**

**environment/dev/variables.tf**

variable "monitoring\_project\_id" {

description = "The project ID for the monitoring setup."

type = string

}

variable "alert\_policies" {

type = map(object({

display\_name = string

filter = string

threshold\_value = number

per\_series\_aligner = string

}))

}

variable "dev\_users" {

description = "List of dev users' email addresses for view-only access."

type = list(string)

}

variable "sre\_users" {

description = "List of SRE users' email addresses for full access."

type = list(string)

}

#### environment/prod/variables.tf

variable "monitoring\_project\_id" {

description = "The project ID for the monitoring setup."

type = string

}

variable "alert\_policies" {

type = map(object({

display\_name = string

filter = string

threshold\_value = number

per\_series\_aligner = string

}))

}

variable "dev\_users" {

description = "List of dev users' email addresses for view-only access."

type = list(string)

}

variable "sre\_users" {

description = "List of SRE users' email addresses for full access."

type = list(string)

}

### Updating the Root Module

#### main.tf

provider "google" {

project = var.monitoring\_project\_id

}

module "logging" {

source = "./modules/logging"

monitoring\_project\_id = var.monitoring\_project\_id

}

module "monitoring" {

source = "./modules/monitoring"

monitoring\_project\_id = var.monitoring\_project\_id

alert\_policies = var.alert\_policies

}

module "service\_accounts" {

source = "./modules/service\_accounts"

monitoring\_project\_id = var.monitoring\_project\_id

}

module "rbac" {

source = "./modules/rbac"

monitoring\_project\_id = var.monitoring\_project\_id

dev\_users = var.dev\_users

sre\_users = var.sre\_users

}

### Creating the RBAC Module

#### modules/rbac/main.tf

# Assign view-only access to development users

resource "google\_project\_iam\_member" "dev\_view\_access" {

for\_each = toset(var.dev\_users)

project = var.monitoring\_project\_id

role = "roles/viewer"

member = "user:${each.key}"

}

# Assign full access to SRE team members

resource "google\_project\_iam\_member" "sre\_full\_access" {

for\_each = toset(var.sre\_users)

project = var.monitoring\_project\_id

role = "roles/owner"

member = "user:${each.key}"

}

### RBAC Variable Definitions

#### modules/rbac/variables.tf

variable "monitoring\_project\_id" {

description = "The project ID for the monitoring setup."

type = string

}

variable "dev\_users" {

description = "List of dev users' email addresses for view-only access."

type = list(string)

}

variable "sre\_users" {

description = "List of SRE users' email addresses for full access."

type = list(string)

}

### Example Terraform Variable Files

#### environment/dev/monitoring.tfvars

monitoring\_project\_id = "your-dev-project-id"

alert\_policies = {

cpu\_utilization = {

display\_name = "CPU Usage Alert"

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

threshold\_value = 80

per\_series\_aligner = "ALIGN\_MEAN"

},

memory\_utilization = {

display\_name = "Memory Usage Alert"

filter = "resource.type = \"gce\_instance\" AND metric.type = \"agent.googleapis.com/memory/percent\_used\" AND metric.labels.state != \"free\""

threshold\_value = 80

per\_series\_aligner = "ALIGN\_MEAN"

},

disk\_utilization = {

display\_name = "Disk Usage Alert"

filter = "resource.type = \"gce\_instance\" AND metric.type = \"agent.googleapis.com/disk/percent\_used\" AND metric.labels.state != \"free\""

threshold\_value = 80

per\_series\_aligner = "ALIGN\_MEAN"

},

vm\_uptime = {

display\_name = "VM Uptime Check Alert"

filter = "metric.type=\"monitoring.googleapis.com/uptime\_check/check\_passed\" AND resource.type=\"gce\_instance\""

threshold\_value = 0

per\_series\_aligner = "ALIGN\_COUNT\_TRUE"

},

error\_rate = {

display\_name = "Error Alert"

filter = "metric.type=\"logging.googleapis.com/user/bucket\_error\_logs\_metric\" AND resource.type=\"logging\_bucket\""

threshold\_value = 0

per\_series\_aligner = "ALIGN\_COUNT\_TRUE"

}

}

dev\_users = ["dev\_user1@example.com", "dev\_user2@example.com"]

sre\_users = ["sre\_user1@example.com", "sre\_user2@example.com"]

#### environment/prod/monitoring.tfvars

monitoring\_project\_id = "your-prod-project-id"

alert\_policies = {

cpu\_utilization = {

display\_name = "CPU Usage Alert"

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

threshold\_value = 80

per\_series\_aligner = "ALIGN\_MEAN"

},

memory\_utilization = {

display\_name = "Memory Usage Alert"

filter = "resource.type = \"gce\_instance\" AND metric.type = \"agent.googleapis.com/memory/percent\_used\" AND metric.labels.state != \"free\""

threshold\_value = 80

per\_series\_aligner = "ALIGN\_MEAN"

},

disk\_utilization = {

display\_name = "Disk Usage Alert"

filter = "resource.type = \"gce\_instance\" AND metric.type = \"agent.googleapis.com/disk/percent\_used\" AND metric.labels.state != \"free\""

threshold\_value = 80

per\_series\_aligner = "ALIGN\_MEAN"

},

vm\_uptime = {

display\_name = "VM Uptime Check Alert"

filter = "metric.type=\"monitoring.googleapis.com/uptime\_check/check\_passed\" AND resource.type=\"gce\_instance\""

threshold\_value = 0

per\_series\_aligner = "ALIGN\_COUNT\_TRUE"

},

error\_rate = {

display\_name = "Error Alert"

filter = "metric.type=\"logging.googleapis.com/user/bucket\_error\_logs\_metric\" AND resource.type=\"logging\_bucket\""

threshold\_value = 0

per\_series\_aligner = "ALIGN\_COUNT\_TRUE"

}

}

dev\_users = ["dev\_user1@example.com", "dev\_user2@example.com"]

sre\_users = ["sre\_user1@example.com", "sre\_user2@example.com"]

**RBAC and Role Permissions**

In this design, RBAC ensures that development users only have view access to the monitoring project and application logs, while the SRE team has full administrative access. The **google\_project\_iam\_member** resources assign specific roles (**roles/viewer** for dev users and **roles/owner** for SRE users) to the respective email addresses.

**Roles Overview:**

* **Viewer Role (roles/viewer)**: Grants read-only access to all resources in the project.
* **Owner Role (roles/owner)**: Grants full access to all resources, including the ability to create, update, and delete resources.

By using these roles, you can control who has access to critical infrastructure components and ensure that only authorized personnel can make changes, while others can only view the status and logs.

To achieve the requirements of creating separate monitoring workspaces for each application, implementing RBAC, using service accounts, and sharing dashboards with business users, we'll structure the Terraform configuration accordingly. The solution will ensure the following:

1. Separate monitoring workspaces for each application.
2. RBAC where dev users have view-only access and SRE users have full access.
3. Implementation of service account management.
4. Production-ready, scalable Terraform code.
5. Mechanism to change roles for business users.
6. Mechanism to share dashboards only with business users.
7. Structure compatible with Terraform Enterprise, with separate folders and **.tfvars** files for each stage.

**Project Structure**

├── environment

│ ├── dev

│ │ ├── main.tf

│ │ ├── variables.tf

│ │ └── monitoring.tfvars

│ └── prod

│ ├── main.tf

│ ├── variables.tf

│ └── monitoring.tfvars

└── modules

├── logging

│ └── main.tf

├── monitoring

│ ├── main.tf

│ ├── variables.tf

│ └── outputs.tf

├── alerting

│ └── main.tf

├── service\_accounts

│ └── main.tf

└── rbac

└── main.tf

### Environment Variables

#### environment/dev/variables.tf

variable "monitoring\_project\_id" {

description = "The project ID for the monitoring setup."

type = string

}

variable "alert\_policies" {

type = map(object({

display\_name = string

filter = string

threshold\_value = number

per\_series\_aligner = string

}))

}

variable "dev\_users" {

description = "List of dev users' email addresses for view-only access."

type = list(string)

}

variable "sre\_users" {

description = "List of SRE users' email addresses for full access."

type = list(string)

}

variable "business\_users" {

description = "List of business users' email addresses for dashboard access."

type = list(string)

}

#### environment/prod/variables.tf

variable "monitoring\_project\_id" {

description = "The project ID for the monitoring setup."

type = string

}

variable "alert\_policies" {

type = map(object({

display\_name = string

filter = string

threshold\_value = number

per\_series\_aligner = string

}))

}

variable "dev\_users" {

description = "List of dev users' email addresses for view-only access."

type = list(string)

}

variable "sre\_users" {

description = "List of SRE users' email addresses for full access."

type = list(string)

}

variable "business\_users" {

description = "List of business users' email addresses for dashboard access."

type = list(string)

}

### Terraform Code for Each Environment

#### environment/dev/main.tf

provider "google" {

project = var.monitoring\_project\_id

}

module "logging" {

source = "../../modules/logging"

monitoring\_project\_id = var.monitoring\_project\_id

}

module "monitoring" {

source = "../../modules/monitoring"

monitoring\_project\_id = var.monitoring\_project\_id

alert\_policies = var.alert\_policies

}

module "service\_accounts" {

source = "../../modules/service\_accounts"

monitoring\_project\_id = var.monitoring\_project\_id

}

module "rbac" {

source = "../../modules/rbac"

monitoring\_project\_id = var.monitoring\_project\_id

dev\_users = var.dev\_users

sre\_users = var.sre\_users

business\_users = var.business\_users

}

#### environment/prod/main.tf

provider "google" {

project = var.monitoring\_project\_id

}

module "logging" {

source = "../../modules/logging"

monitoring\_project\_id = var.monitoring\_project\_id

}

module "monitoring" {

source = "../../modules/monitoring"

monitoring\_project\_id = var.monitoring\_project\_id

alert\_policies = var.alert\_policies

}

module "service\_accounts" {

source = "../../modules/service\_accounts"

monitoring\_project\_id = var.monitoring\_project\_id

}

module "rbac" {

source = "../../modules/rbac"

monitoring\_project\_id = var.monitoring\_project\_id

dev\_users = var.dev\_users

sre\_users = var.sre\_users

business\_users = var.business\_users

}

### Modules

#### modules/logging/main.tf

resource "google\_project\_service" "logging" {

project = var.monitoring\_project\_id

service = "logging.googleapis.com"

}

#### modules/monitoring/main.tf

resource "google\_project\_service" "monitoring" {

project = var.monitoring\_project\_id

service = "monitoring.googleapis.com"

}

resource "google\_monitoring\_workspace" "workspace" {

project = var.monitoring\_project\_id

}

resource "google\_logging\_metric" "bucket\_error\_logs\_metric" {

name = "bucket-error-logs-metric"

filter = "resource.type=\"bigquery\_resource\" AND (logName=\"projects/${var.monitoring\_project\_id}/logs/cloudaudit.googleapis.com%2Fdata\_access\") AND protoPayload.status.code!=200"

project = var.monitoring\_project\_id

metric\_descriptor {

value\_type = "INT64"

unit = "1"

metric\_kind = "DELTA"

}

}

output "workspace\_name" {

value = google\_monitoring\_workspace.workspace.name

}

#### modules/monitoring/variables.tf

variable "monitoring\_project\_id" {

description = "The project ID for the monitoring setup."

type = string

}

variable "alert\_policies" {

type = map(object({

display\_name = string

filter = string

threshold\_value = number

per\_series\_aligner = string

}))

}

#### modules/monitoring/outputs.tf

output "workspace\_name" {

value = google\_monitoring\_workspace.workspace.name

}

#### modules/alerting/main.tf

resource "google\_monitoring\_alert\_policy" "alert\_policies" {

for\_each = var.alert\_policies

project = var.monitoring\_project\_id

display\_name = each.value.display\_name

conditions {

display\_name = each.value.display\_name

condition\_threshold {

filter = each.value.filter

comparison = "COMPARISON\_GT"

threshold\_value = each.value.threshold\_value

duration = "0s"

aggregations {

alignment\_period = "60s"

per\_series\_aligner = each.value.per\_series\_aligner

}

trigger {

percent = 100

}

}

}

notification\_channels = [google\_monitoring\_notification\_channel.email\_channel.name]

}

#### modules/service\_accounts/main.tf

resource "google\_service\_account" "monitoring\_service\_account" {

account\_id = "monitoring-sa"

display\_name = "Monitoring Service Account"

project = var.monitoring\_project\_id

}

#### modules/rbac/main.tf

resource "google\_project\_iam\_member" "dev\_view\_access" {

for\_each = toset(var.dev\_users)

project = var.monitoring\_project\_id

role = "roles/viewer"

member = "user:${each.key}"

}

resource "google\_project\_iam\_member" "sre\_full\_access" {

for\_each = toset(var.sre\_users)

project = var.monitoring\_project\_id

role = "roles/owner"

member = "user:${each.key}"

}

resource "google\_project\_iam\_member" "business\_dashboard\_access" {

for\_each = toset(var.business\_users)

project = var.monitoring\_project\_id

role = "roles/monitoring.viewer"

member = "user:${each.key}"

}

#### modules/rbac/variables.tf

variable "monitoring\_project\_id" {

description = "The project ID for the monitoring setup."

type = string

}

variable "dev\_users" {

description = "List of dev users' email addresses for view-only access."

type = list(string)

}

variable "sre\_users" {

description = "List of SRE users' email addresses for full access."

type = list(string)

}

variable "business\_users" {

description = "List of business users' email addresses for dashboard access."

type = list(string)

}

### Terraform Variable Files

#### environment/dev/monitoring.tfvars

monitoring\_project\_id = "your-dev-project-id"

alert\_policies = {

cpu\_utilization = {

display\_name = "CPU Usage Alert"

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

threshold\_value = 80

per\_series\_aligner = "ALIGN\_MEAN"

},

memory\_utilization = {

display\_name = "Memory Usage Alert"

filter = "resource.type = \"gce\_instance\" AND metric.type = \"agent.googleapis.com/memory/percent\_used\" AND metric.labels.state != \"free\""

threshold\_value = 80

per\_series\_aligner = "ALIGN\_MEAN"

},

disk\_utilization = {

display\_name = "Disk Usage Alert"

filter = "resource.type = \"gce\_instance\" AND metric.type = \"agent.googleapis.com/disk/percent\_used\" AND metric.labels.state != \"free\""

threshold\_value = 80

per\_series\_aligner = "ALIGN\_MEAN"

},

vm\_uptime = {

display\_name = "VM Uptime Check Alert"

filter = "metric.type=\"monitoring.googleapis.com/uptime\_check/check\_passed\" AND resource.type=\"gce\_instance\""

threshold\_value = 0

per\_series\_aligner = "ALIGN\_COUNT\_TRUE"

},

error\_rate = {

display\_name = "Error Alert"

filter = "metric.type=\"logging.googleapis.com/user/bucket\_error\_logs\_metric\" AND resource.type=\"logging\_bucket\""

threshold\_value = 0

per\_series\_aligner = "ALIGN\_COUNT\_TRUE"

}

}

dev\_users = ["dev-user1@example.com", "dev-user2@example.com"]

sre\_users = ["sre-user1@example.com", "sre-user2@example.com"]

business\_users = ["business-user1@example.com", "business-user2@example.com"]

#### environment/prod/monitoring.tfvars

monitoring\_project\_id = "your-prod-project-id"

alert\_policies = {

cpu\_utilization = {

display\_name = "CPU Usage Alert"

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

threshold\_value = 80

per\_series\_aligner = "ALIGN\_MEAN"

},

memory\_utilization = {

display\_name = "Memory Usage Alert"

filter = "resource.type = \"gce\_instance\" AND metric.type = \"agent.googleapis.com/memory/percent\_used\" AND metric.labels.state != \"free\""

threshold\_value = 80

per\_series\_aligner = "ALIGN\_MEAN"

},

disk\_utilization = {

display\_name = "Disk Usage Alert"

filter = "resource.type = \"gce\_instance\" AND metric.type = \"agent.googleapis.com/disk/percent\_used\" AND metric.labels.state != \"free\""

threshold\_value = 80

per\_series\_aligner = "ALIGN\_MEAN"

},

vm\_uptime = {

display\_name = "VM Uptime Check Alert"

filter = "metric.type=\"monitoring.googleapis.com/uptime\_check/check\_passed\" AND resource.type=\"gce\_instance\""

threshold\_value = 0

per\_series\_aligner = "ALIGN\_COUNT\_TRUE"

},

error\_rate = {

display\_name = "Error Alert"

filter = "metric.type=\"logging.googleapis.com/user/bucket\_error\_logs\_metric\" AND resource.type=\"logging\_bucket\""

threshold\_value = 0

per\_series\_aligner = "ALIGN\_COUNT\_TRUE"

}

}

dev\_users = ["dev-user1@example.com", "dev-user2@example.com"]

sre\_users = ["sre-user1@example.com", "sre-user2@example.com"]

business\_users = ["business-user1@example.com", "business-user2@example.com"]

**Changing Roles for Business Users**

To change the roles for any business user, you can modify the **business\_users** list in the respective **monitoring.tfvars** file and apply the changes using Terraform.

**Sharing Dashboards**

To share dashboards only with business users, you need to ensure that only those users have the **roles/monitoring.viewer** role assigned, which we've already done in the RBAC module.

**Final Notes**

* Ensure that the service accounts and permissions are correctly set up and that the Terraform code is executed in a secure environment.
* Regularly audit and review IAM policies and roles to maintain security and compliance.
* Use Terraform Enterprise features such as workspaces and variables to manage different environments and stages effectively.

This setup should provide a robust and production-ready centralized logging and monitoring system with proper RBAC and service account management.