Creating new service within an application

Overview

A service is a collection of a set of Terraform components which upon executed provisions all the infra required. This article explains the procedure to create a new service within an application and explains the terraform components template to add a new service.

Content

In a hub and spoke architecture, creating service involves the creation of common components in the hub and specific components in the spoke.

- Hub components: creating a GitLab repository and ECR repository.
- Spoke components: Involves creating new ECS Fargate service.

Hub components:

Terraform components highlighted below creates a new service named `ui` . It creates an ECR repository and Gitlab repository. The ECR repository name is constructed using the variable application name and passed service name. Predefined tags are assigned to for easy reference. The principals which can push and pull from the repository are assigned using ARNs of `iam` roles. This module makes use of custom Terraform module `app.terraform.io/min-au-infra/ecr/aws`. The module is codebase can be found in here. The details about the module including inputs, outputs can be found here.

The second component, which is related to the Gitlab project component, creates Gitlab repo. It makes use of the module listed here. The source of the module is located in this Gitlab repository. The module takes three required inputs and produces 4 outputs related to the repository.

```
module "service_ui_ecr" {
  source  = "app.terraform.io/min-au-infra/ecr/aws"
  version = "v3.1.0"

  repository_name = "${lower(var.application_name)}-${local.
service_ui_name}"
  custom_tags     = var.custom_tags

  push_access_principal_arns = ["arn:aws:iam::${data.
aws_caller_identity.devops.account_id}:role/${var.application_name}-
gitlab-runner-role"]

  # We know how the ECS task execution role naming will be defined so
we can just build it here
  pull_access_principal_arns = formatlist(
    "arn:aws:iam::%s:role/%s-%s-ecs-task-execution",
    [for acct in var.spoke_aws_accounts : acct.account_id],
    lower(var.application_name),
    lower(local.service_ui_name),
  )

  providers = {
    aws = aws.devops
  }
}

module "service_ui_gitlab_project" {
  source  = "app.terraform.io/min-au-infra/gitlab-project/aws"
  version = "v2.0.0"

  name        = "${lower(var.application_name)}-${local.
service_ui_name}"
  description = "Project for service '${lower(var.application_name)}
-${local.service_ui_name}'"
  group_id    = module.gitlab_group.id
}
```

Spoke components:

Spoke components initialize the ECS services with the help of fargate-service module. Below Terraform module takes required inputs such as `service_name`, `custom_tags` from the variables module. This module also depends on `ecs_cluster` which creates cluster for fargate service. Default secrets are created with the service.

```
module "service_ui" {
  source  = "app.terraform.io/min-au-infra/fargate-service/aws"
  version = "v5.1.0"

  service_name = "${var.application_name}-${local.service_ui_name}"
  charge_code  = var.charge_code
  custom_tags  = var.custom_tags

  cpu    = var.ui_service_info.cpu_allocation
  memory = var.ui_service_info.mem_allocation

  service_count = var.ui_service_info.num_containers
  service_port  = local.service_ui_port
  service_image = var.ui_ecr_repository_url
  envvars       = []
  email_ids     = var.monitoring_email_ids

  attach_lb        = true
  alb_path         = "*"
  alb_priority     = 200
  alb_listener_arn = module.ecs_cluster.alb_listener_http_arn

  ecs_cluster_name = module.ecs_cluster.ecs_cluster_name
  vpc_id           = var.vpc_id
  subnets          = var.private_subnet_ids
  security_groups  = module.ecs_cluster.ecs_task_security_group_ids

  service_discovery_namespace = module.ecs_cluster.
service_discovery_namespace

  ecs_task_role_policy_arns_count = 0
  ecs_task_role_policy_arns       = []

  secrets_count = 1

  secrets = [{
    name  = "default"
    value = "${jsonencode(var.ui_secret_value)}"
  }]

  account_alias      = var.spoke_name
  log_aggregation_s3 = var.log_aggregation_s3
  providers = {
    aws.service = aws.spoke
    aws.logging = aws.logging
  }
}
```

The corresponding variables file is below which declares variables required for the fargate service component. Before running the spoke components in the Terraform cloud, the below variables need to updated in the terraform application workspace.

```
variable "ui_ecr_repository_url" {
  description = "URL to the Docker repository in ECR"
}

variable "ui_service_info" {
  description = "ECS service runtime parameters"
  type = object({
    # number of CPU unit to reserve for the containers
    cpu_allocation = number
    # max memory (Mb) used by the containers
    mem_allocation = number
    # number of containers to launch
    num_containers = number
  })
  default = {
    cpu_allocation = 256
    mem_allocation = 512
    num_containers = 1
  }
}

variable "ui_secret_value" {
  description = "A map which is used to populate the SecretsManager
secret. See README for additional information"
  type        = map(string)
  default     = {}
}
```

Relevant materials

- ECR module file https://app.terraform.io/app/min-au-infra/modules/view/ecr/aws/3.1.0
- Gitlab module file https://app.terraform.io/app/min-au-infra/modules/view/gitlab-project/aws/2.0.0
- Fargate documentation https://aws.amazon.com/fargate/