Devops Engineer Code Challenge

Code challenge for Devops Engineer in Telefonica Citadel Project. We'd like you to design and develop a playable demo to create and deploy a helm chart.

Challenges

Challenge 1

Modify the Ping Helm Chart to deploy the application on the following restrictions:

- Isolate specific node groups forbidding the pods scheduling in this node groups.
- Ensure that a pod will not be scheduled on a node that already has a pod of the same type.
- Pods are deployed across different availability zones.

Challenge 2

We have a private registry based on Azure Container Registry where we publish all our Helm charts. Let's call this registry *reference.azurecr.io*.

When we create an AKS cluster, we also create another Azure Container Registry where we need to copy the Helm charts we are going to install in that AKS from the reference registry. Let's call this registry *instance.azurecr.io* and assume it resides in an Azure subscription with ID *c9e7611c-d508-4fbf-aede-0bedfabc1560*.

As we work with Terraform to install our charts in our AKS cluster, we've thought that it would be quite helpful to have a reusable module that allows us to **import** a set of charts from the reference registry to the instance registry using a local provisioner and install them on our AKS cluster.

We will call our reusable module in the following way:

```
module "chart" {
  . . .

  acr_server              = "instance.azurecr.io"
  acr_server_subscription = "c9e7611c-d508-4fbf-aede-0bedfabc1560"

  source_acr_client_id     = "1b2f651e-b99c-4720-9ff1-ede324b8ae30"
  source_acr_client_secret = "Zrrr8~5~F2Xiaaaa7eS.S85SXXAAfTYizZEF1cRp"
  source_acr_server        = "reference.azurecr.io"

  charts = [
    {
      chart_name       = <chart_name>
      chart_namespace  = <chart_namespace>
      chart_repository = <chart_repository>
      chart_version    = <chart_version>

      values = [
        {
          name  = <name>
          value = <value>
        },
        {
          name  = <name>
          value = <value>
        }
      ]

      sensitive_values = [
        {
          name  = <name>
          value = <value>
        },
        {
          name  = <name>
          value = <value>
        }
      ]
    },
    {
      . . .
    }
  ]
}
```

You need to implement the reusable module. It should pass validations provided by the *terraform fmt* and *terraform validate* commands.

You can assume the caller will be authenticated in Azure with enough permissions to import Helm charts into the instance registry and will provide the module a configured helm provider.

Challenge 3

Create a Github workflow to allow installing helm chart from Challenge #1 using module from Challenge #2, into an AKS cluster (considering a preexisting resource group and cluster name).

How do we evaluate the test?

1. How clear your code is.
2. How clear your explanation is.
3. Easiness to run the code.