1.Projects in GCP – 3 tier applications on Google Cloud

As a Google Cloud Architect, your understanding of [GCP Projects](https://googlecloudarchitect.us/single-gcp-project-or-multiple-small-gcp-projects/) and [Service Accounts](https://googlecloudarchitect.us/best-practices-around-service-accounts/) to access resources within that project, is key to gaining operational competency.

This post discusses how GCP projects can serve as an IAM boundary, a billing boundary as well as an application boundary (app tier boundary).

### The Project Boundary is PRIMARILY an IAM Boundary

Say your infrastructure consists of a large number of cloud resources, that need to be shared in specific ways across your organization.

* Development teams only need development network resources (VPN tunnels for example) whereas
* Production teams don’t care about your VPN. They have their own direct connect that they use to get onto their jump boxes to get to their production hosted assets.

How do you isolate these networking resources between different IAM groups (prod users and dev users)?

In both AWS and Azure, you would start isolating these at the network level — i.e. you may define a development VPC (or a dev-VNET in Azure) for development resources and a production VPC (for production resources).

In other words, you have to lean on the cloud network itself to provide you with the isolation levels of the resources you need.

There should be an easier way to target certain IAM groups for certain resources and others for other resources.

### Enter the GCP Project

Now, enter GCP and the Project Level Construct. The same problem can be solved using the trust boundary provided by Projects.

In GCP, you would solve this problem by having two separate projects (a DEV project, a PROD project etc.). Each project comes with it’s own Default VPC (and you can add as many custom VPCs as you like).

Projects are holders of resources, akin to Accounts in AWS.

   While AWS accounts are certainly more than simple resource containers, this is still the best way to visualize the correspondence between AWS accounts and GCP projects. They are also both the highest level ‘billing containers’.

### One or multiple- GCP Projects as Application Tier Boundaries

Since the project boundary is mainly invisible (and only provides advantages), GCP’s recommendation is to have as MANY projects — with each project containing only a limited number of resources.

Take the example of a built in service account that is used to access compute engine resources. This SA (service account)  actually has Project Editor (which means, it can do just about everything to EVERY resource in the project, except DELETE the project itself).

This is not the ‘least privilege’ approach.

Hence, what you need it to split up your SINGLE large project, into smaller projects. That way, access control is simplified.

Another example is that of splitting up your app into tiered project. A three tier app would exist on GCP as three projects – a middle tier project, a web tier project and a data tier project.

### Cross project access to resources?

  First of all, you should ALMOST NEVER need to cross a project boundary. A project should contain all the resources it needs. Should you need a resource from another project, there is a specific construct in GCP (called a Shared Project), that enables you to do just that.

Still, if the need arises (say you have a centralized logging bucket in a separate ‘logging project’), you can access that resource from another project.

If you think AWS accounts, how does a resource in Account A, access a resource in Account B? The answer is cross account roles.

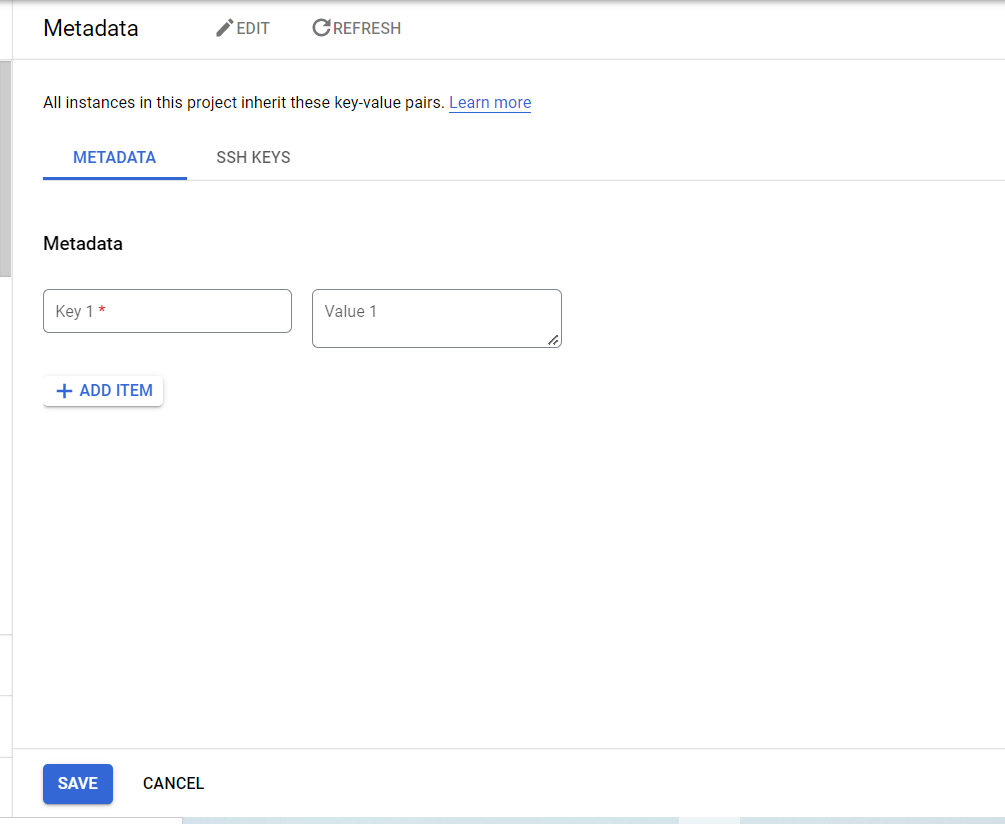
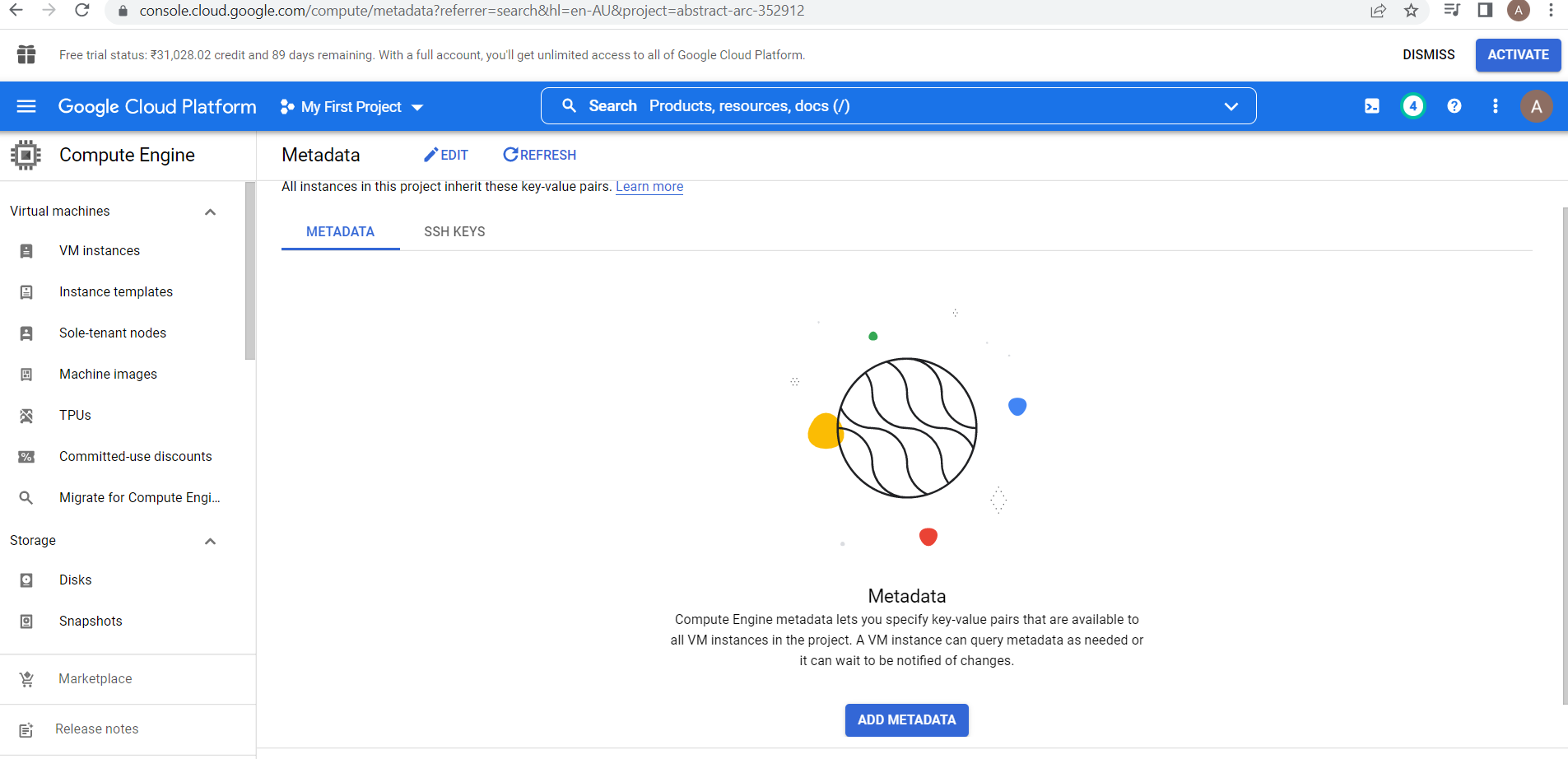
In a similar vein, in GCP as well, a role is needed to provide cross project access. A service account in project A is granted access (access being a compute viewer) to a compute resource in project B, by assigning the appropriate role (see gcloud example below).

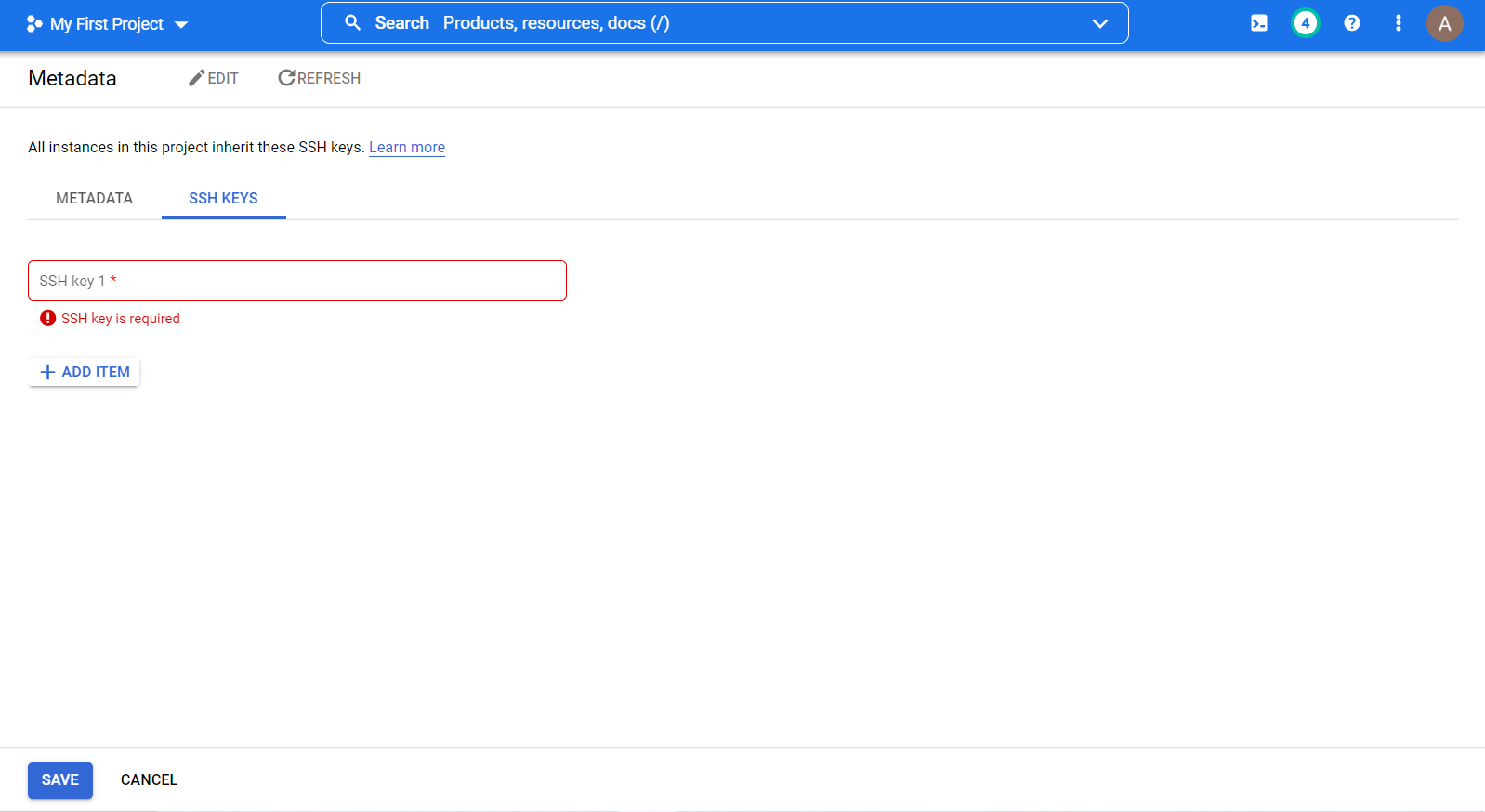
gcloud projects add-iam-policy-binding $GCP\_PROJECT\_ID\_B \

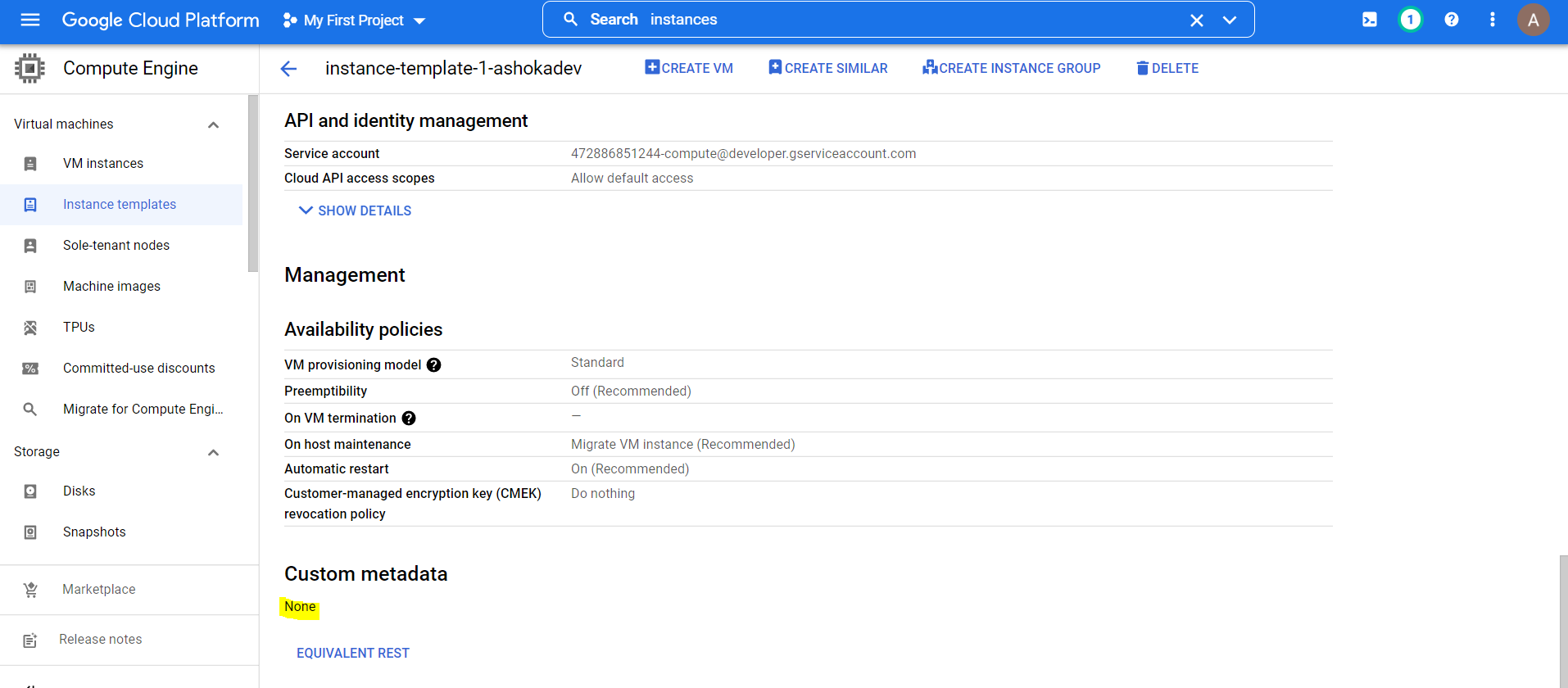
--member=serviceAccount:${GCP\_PROJECT\_ID\_A}@myaccount.gserviceaccount.com \

--role=roles/compute.viewer \

2. We need to write code that will query the meta data of an instance in the GCP







3. We have a nested object, we would like a function that you pass in the object and a key and get back the value.

# importing module

from dataclasses import dataclass

@dataclass

class A:

a: int

b: str

@dataclass

class B:

c: str

d: A

# FIRST APPROACH

# creating object for class b with following values

# c ='hello world'

# a = 8

# b ='ashok'

data ={'c':'hello world', 'd':{'a':8, 'b':'ashok'}}

b = B(\*\*data)

print (b)

# SECOND APPROACH

data ={'c':'hello world', 'd': A(\*\*{'a':8, 'b':'ashok'})}

c = B(\*\*data)

print(c)

