1. We need to have Anthos Discovery tool to check successful migration of a VM to Anthos
2. We need to have Kubernetes Processing cluster to Examine the source VM and create the migration plan and execute the actual migration.
3. Command line tool to interact with the Processing Cluster:

Migctl (just like kubctl)

Migrate for Anthos workflow:

* Enable service accounts and APIs.
* Deploy Processing cluster.
* Qualify your workload.
* Define migration source.
* Generate and review plan.
* Generate and review artifacts.
* Deploy containerized application.

Compatible windows workloads:

* Microsoft windows server 2008R2 or higher
* Microsoft IIS 7 or higher web applications
* ASP.NET and .NET Framework version 3.5 or higher

Check whether windows VM fit for Anthos migration or not?

RDP into the windows VM and do the following:

Windows discovery tool is used to check Windows VM is fit to migrate windows VM to Anthos or not.

Download windows discovery tool from the below link:

<https://storage.googleapis.com/anthos-migrate-release/v1.9.0/windows/amd64/collect_info.exe>.

run collect\_info.exe file and check the compatible of VM to fit for Anthos migration.

* info-hostname-timestamp.zip indicates that the VM is a potential fit to be migrated.
* info-hostname-timestamp-NOFIT.zip indicates that the VM is not fit to be migrated. Typically, a NOFIT results from a missing or incorrect version of IIS.

Windows VM can be only migrated when the instance is in stopped state.

Then go to Anthos and select migrate to containers and select sources.

Click on Processing cluster and give it to a “source name and type”

And last click “add source”

And then on same page go to migrations and click “create migration”

Give:

1. Migration name
2. Select source: source name
3. VM OS type: as windows
4. VM ID: windows Application VM ID
5. Migration intent: as Image

Then click on Create Migration 🡺 it will start the migration plan

It will create migration plan once plan done, please review the plan and click “save and Generate Artifacts”

Get the generated artifacts using the following Migctl command

🡺**migctl** migration get-artifacts “Migration name”

Above command will gives you the generated artifacts source

Copy the artifacts to desired location and unzip the artifacts.

Go to the unzipped folder and build docker image using Dockerfile

**docker build -t gcr.io/project-name/windows-application-name:v1.0.0 .\artiafcts\**

push the docker image to google container

* **docker push** **gcr.io/ project-name /windows-application-name:v1.0.0**

then go the clusters and deploy the

gcloud container clusters get-credentials “cluster-name” –zone “zone-name” –project “project-name”

kubectl get nodes

create windows-deployment.yml file to deploy the application with the image that we already pushed to container registry “**gcr.io/ project-name /windows-application-name:v1.0.0”**

* kubectl apply -f windows-deployment.yml

kubectl get pods.

kubectl get deploy.

kubectl expose deployment windows-deployment.yml --type=loadbalancer –name “application name”

kubectl get services.