Dynamic Volume Provisioning

Concept

Udemy

Objectives

Concept

a. Overview of Dynamic Provisioning of Volume

Review Demo

a. Storage Class

b. Persistent Volume Claim (PVC)

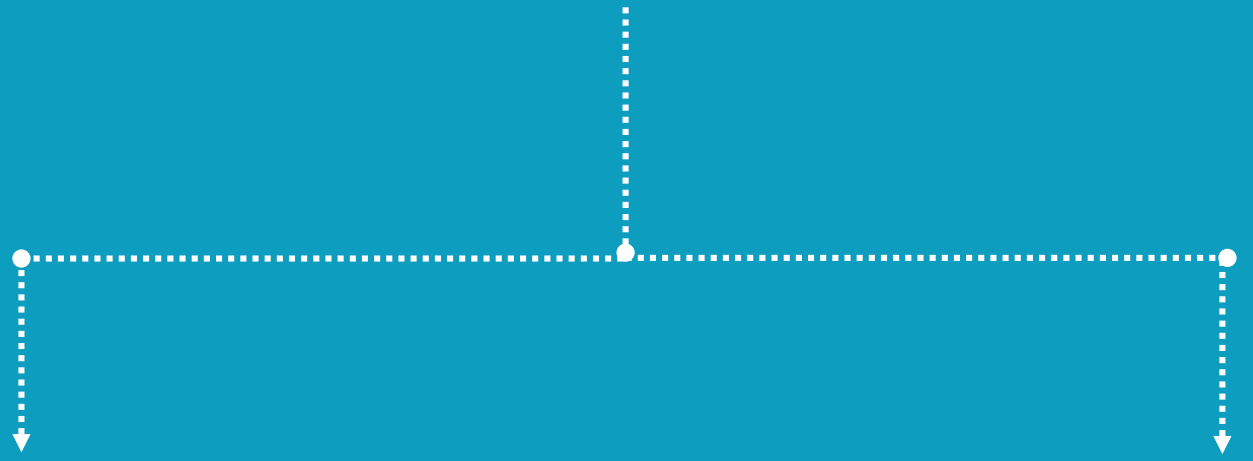
c. Reference claim in Pod

d. Test use case

srinathchalla@outlook.com

Provisioning → Binding → Using → Reclaiming

Provisioning



Static

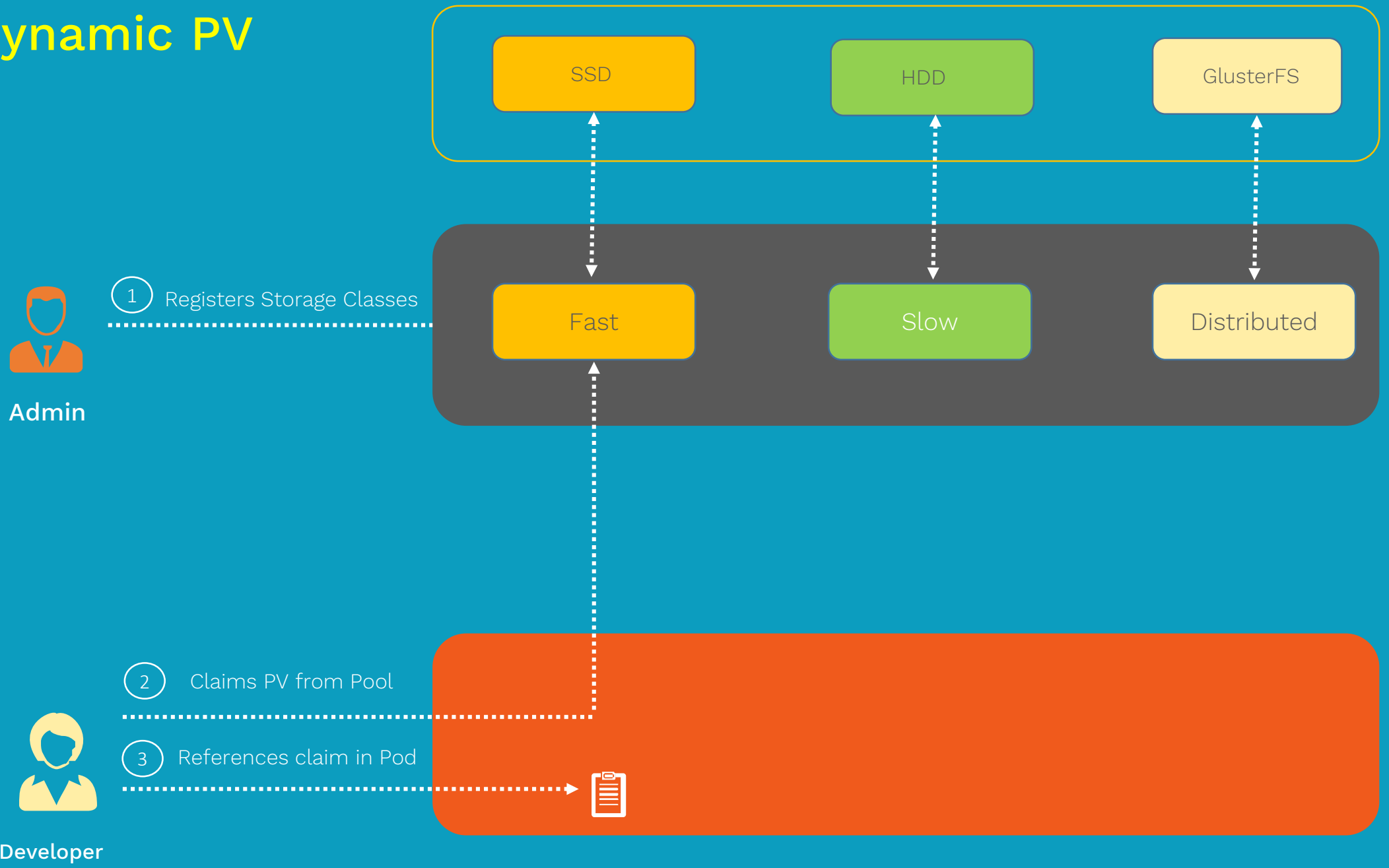
PV needs to be **created before** PVC

Dynamic

PV is **created at same time** of PVC

srinathchalla@outlook.com

Dynamic PV



srinathchalla@outlook.com

Review Demo

Storage Class

Persistent Volume Claim

Referencing claim in Pod

Test use case

srinathchalla@outlook.com

1. StorageClass

srinathchalla@outlook.com

StorageClass – Manifest file

1. Storage Class

```yaml
# sc.yaml
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: fast
provisioner: kubernetes.io/gce-pd
parameters:
  type: pd-ssd
```

StorageClass – Create & Display

```
schalla@master:$ kubectl create -f sc.yaml
storageclass "fast" created
```

```
schalla@master:$ kubectl get storageclass
NAME                 PROVISIONER            AGE
fast                 kubernetes.io/gce-pd   37s
standard (default)   kubernetes.io/gce-pd   4d
```

```
schalla@master:$ kubectl describe storageclass fast
Name:            fast
IsDefaultClass:  No
Annotations:     <none>
Provisioner:     kubernetes.io/gce-pd
Parameters:      type=pd-ssd
ReclaimPolicy:   Delete
Events:          <none>
```

2. Persistent Volume Claim (PVC)

srinathchalla@outlook.com

Persistent Volume Claim(PVC) - Config

1 Storage Class

```yaml
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: fast
provisioner: kubernetes.io/gce-pd
parameters:
  type: pd-ssd
```

2 Persistent Volume Claim (PVC)

```yaml
# pvc-dv1.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my-disk-claim-1
spec:
  resources:
    requests:
      storage: 30Gi
  accessModes:
    - ReadWriteOnce
  storageClassName: fast
```

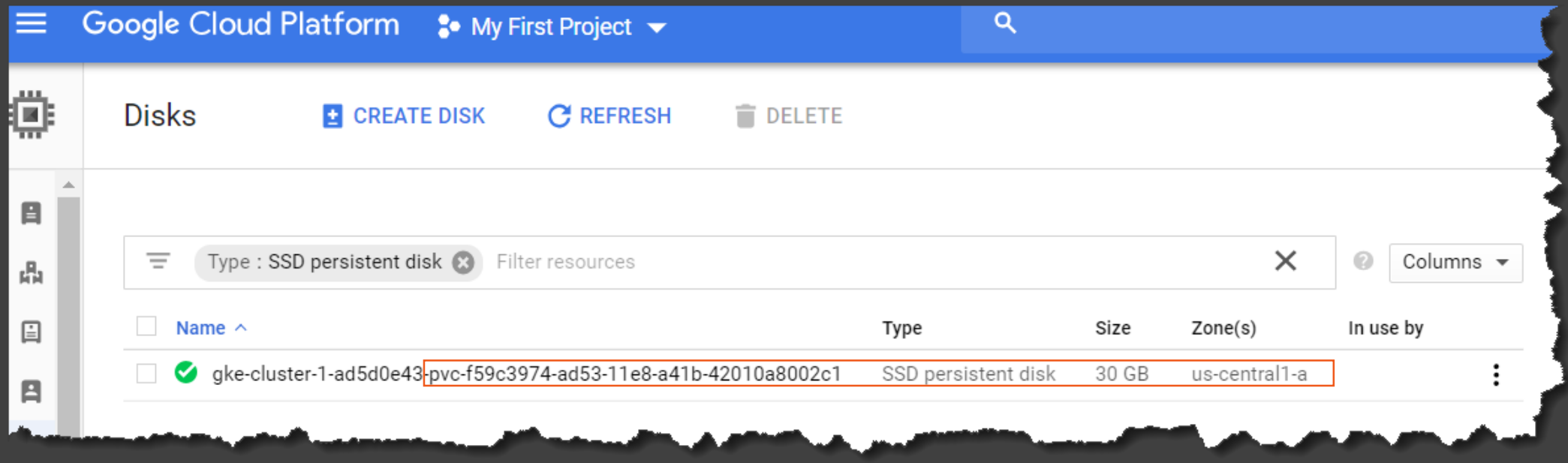
```yaml
# pvc-dv2.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my-disk-claim-2
spec:
  resources:
    requests:
      storage: 40Gi
  accessModes:
    - ReadWriteOnce
  storageClassName: fast
```

srinathchalla@outlook.com

Persistent Volume Claim(PVC) - Create & Display

```
schalla@master:$ kubectl create -f pvc-dv1.yaml
persistentvolumeclaim "my-disk-claim-1" created
```

```
schalla@master:$ kubectl get pvc
NAME              STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
my-disk-claim-1   Bound     pvc-f59c3974-ad53-11e8-a41b-42010a8002c1   30Gi       RWO            fast           4m
```

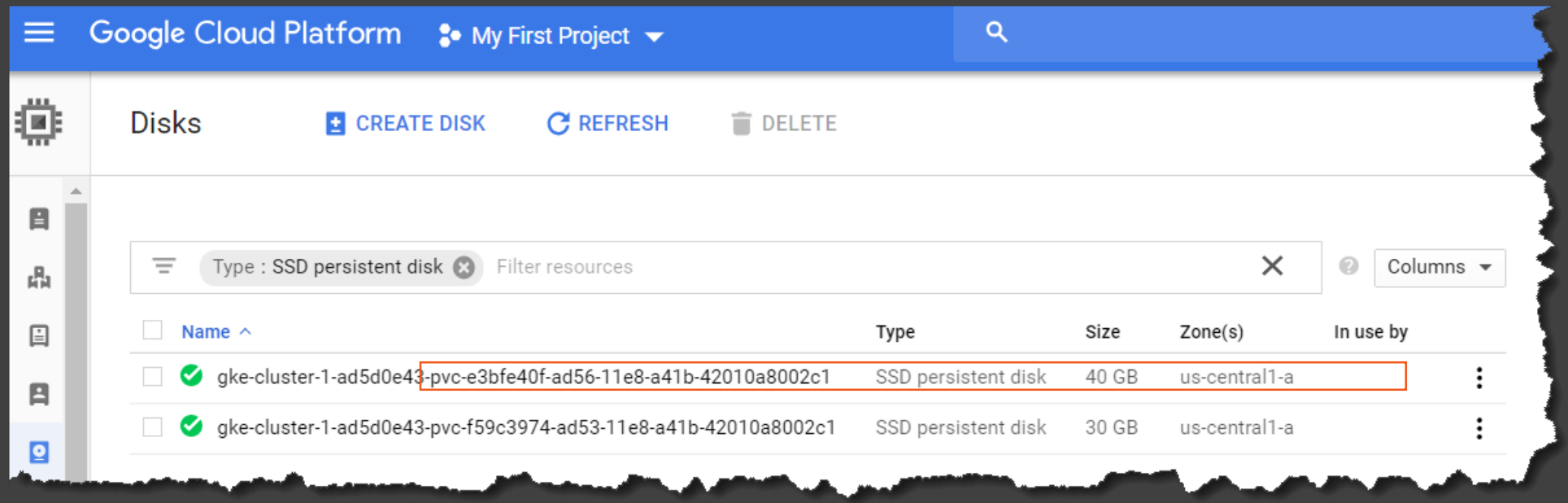


srinathchalla@outlook.com

Persistent Volume Claim(PVC) - Create & Display

```
schalla@master:$  kubectl create -f pvc-dv2.yaml
persistentvolumeclaim "my-disk-claim-2" created
```

```
schalla@master:$  kubectl get pvc
NAME             STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
my-disk-claim-1  Bound     pvc-f59c3974-ad53-11e8-a41b-42010a8002c1   30Gi       RWO            fast           21m
my-disk-claim-2  Bound     pvc-e3bfe40f-ad56-11e8-a41b-42010a8002c1   40Gi       RWO            fast           12s
```



srinathchalla@outlook.com

3. Referencing claim in Pod

srinathchalla@outlook.com

Persistent Volume Claim(PVC) - Config

1 Storage Class

```yaml
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: fast
provisioner: kubernetes.io/gce-pd
parameters:
  type: pd-ssd
```

2 Persistent Volume Claim (PVC)

```yaml
# pvc-dv1.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my-disk-claim-1
spec:
  resources:
    requests:
      storage: 30Gi
  accessModes:
    - ReadWriteOnce
  storageClassName: fast
```

3 Referencing claim in Pod

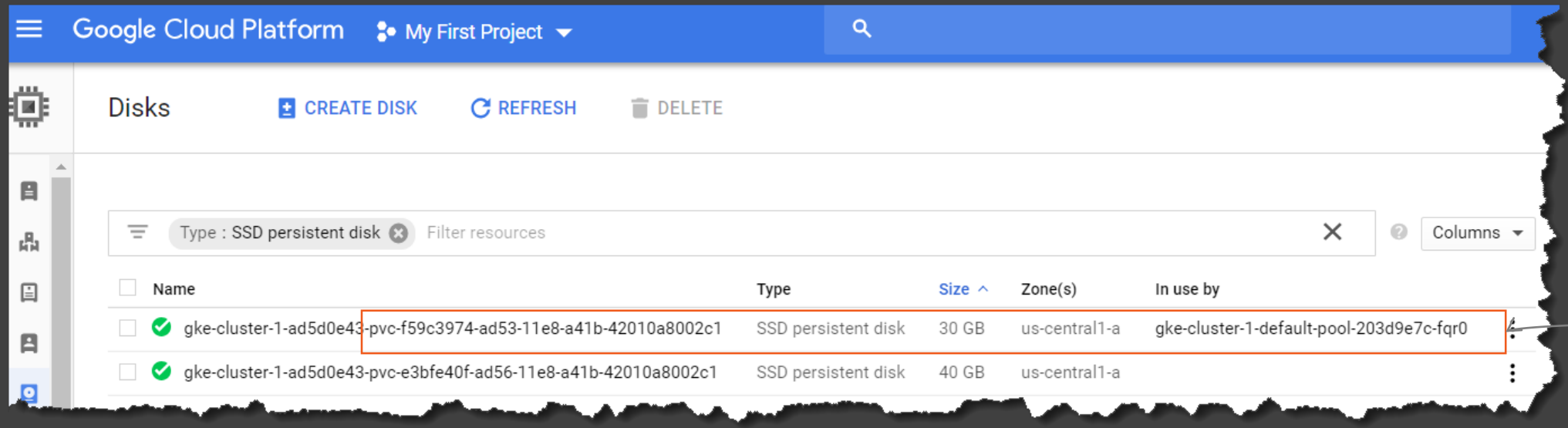
```yaml
# nginx-pv.yaml
apiVersion: v1
kind: Pod
metadata:
  name: pv-pod
spec:
  containers:
  - name: test-container
    image: nginx
    volumeMounts:
    - mountPath: /test-pd
      name: test-volume
  volumes:
  - name: test-volume
    persistentVolumeClaim:
      claimName: my-disk-claim-1
```

srinathchalla@outlook.com

Referencing claim in Pod – Create & Display

```
schalla@master:$ kubectl create -f nginx-pv.yaml
pod "pv-pod" created
```

```
schalla@master:$ kubectl get po -o wide
NAME          READY     STATUS    RESTARTS   AGE       IP           NODE
pv-pod        1/1       Running   0          11m       10.8.3.12    gke-cluster-1-default-pool-203d9e7c-fqr0
```



srinathchalla@outlook.com

4. Testing Use Case

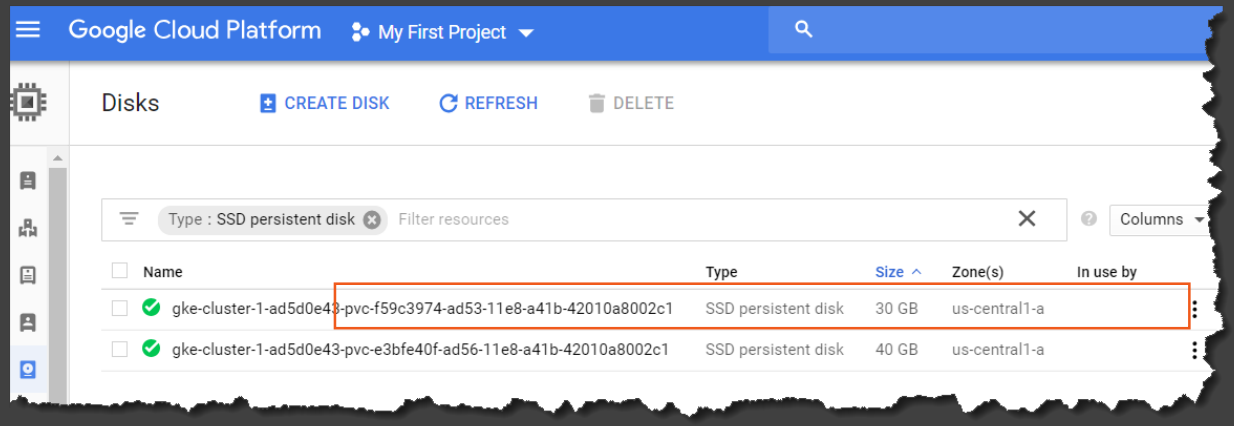
1. Create a sample test file inside the mount.
2. Delete the Pod
3. Recreate the Pod with same configuration
4. Verify the data created in step-1 is still available?

srinathchalla@outlook.com

Testing-1

```
schalla@master:$ kubectl exec pv-pod -it -- /bin/sh
# df -h /usr/share/nginx/html
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb         30G   41M   28G   1% /test-pd
#
# cd /test-pd
# echo "From first pod" > test1.txt
# exit
```

```
schalla@master:$  kubectl delete -f nginx-pv.yaml
pod "pv-pod" deleted
```

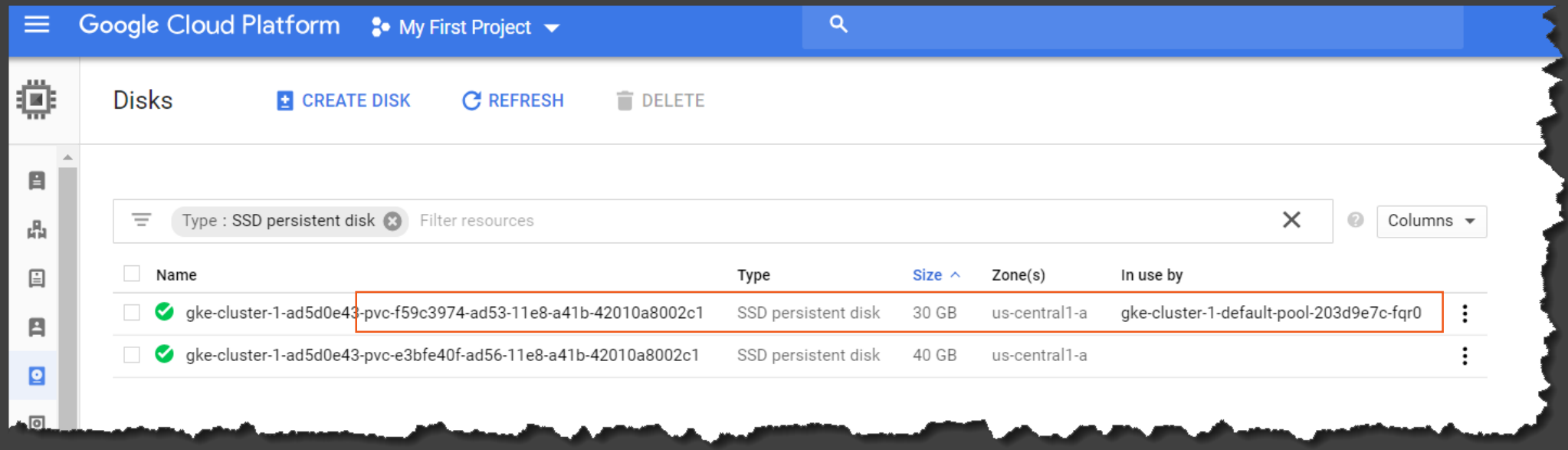


srinathchalla@outlook.com

Testing-2

```
schalla@master:$  kubectl create -f nginx-pv.yaml
pod "pv-pod" created
```

```
schalla@master:$  kubectl get po -o wide
NAME          READY     STATUS    RESTARTS   AGE       IP           NODE
pv-pod        1/1       Running   0          53s       10.8.3.13    gke-cluster-1-default-pool-203d9e7c-fqr0
```



srinathchalla@outlook.com

Validation

```
schalla@master:$ kubectl exec pv-pod df /test-pd
Filesystem     1K-blocks  Used Available Use% Mounted on
/dev/sdb        10255636 36892   9678072   1% /test-pd
```

```
schalla@master:$ kubectl exec pv-pod ls /test-pd/
lost+found
test1.txt
```

```
schalla@master:$ kubectl exec pv-pod cat /test-pd/test1.txt
From first pod
```

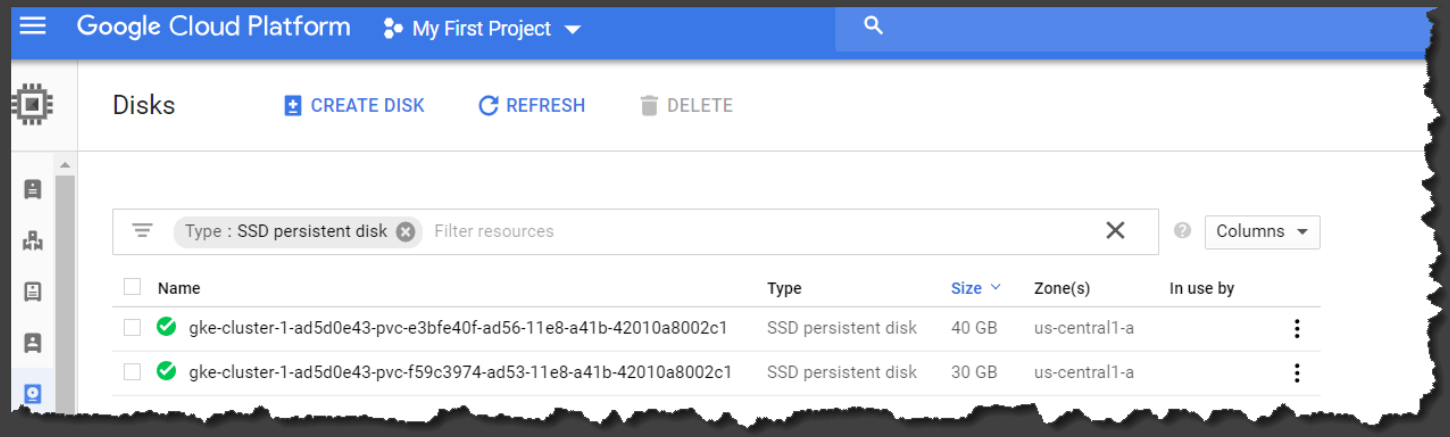
srinathchalla@outlook.com

Cleanup

```
schalla@master:$  kubectl delete -f nginx-pv2.yaml
pod "task-pv-pod2" deleted
```

```
schalla@master:$  kubectl delete -f pvc.yaml
persistentvolumeclaim "my-disk-claim" deleted
```

```
schalla@master:$  kubectl delete storageclass fast
storageclass "fast" deleted
```



srinathchalla@outlook.com

Summary

Concept

a. Overview of Dynamic Provisioning of Volume

Review Demo

a. Storage Class

b. Persistent Volume Claim (PVC)

c. Reference claim in Pod

d. Test use case

srinathchalla@outlook.com

Coming up...

Demo

Dynamic Volume Provisioning

srinathchalla@outlook.com