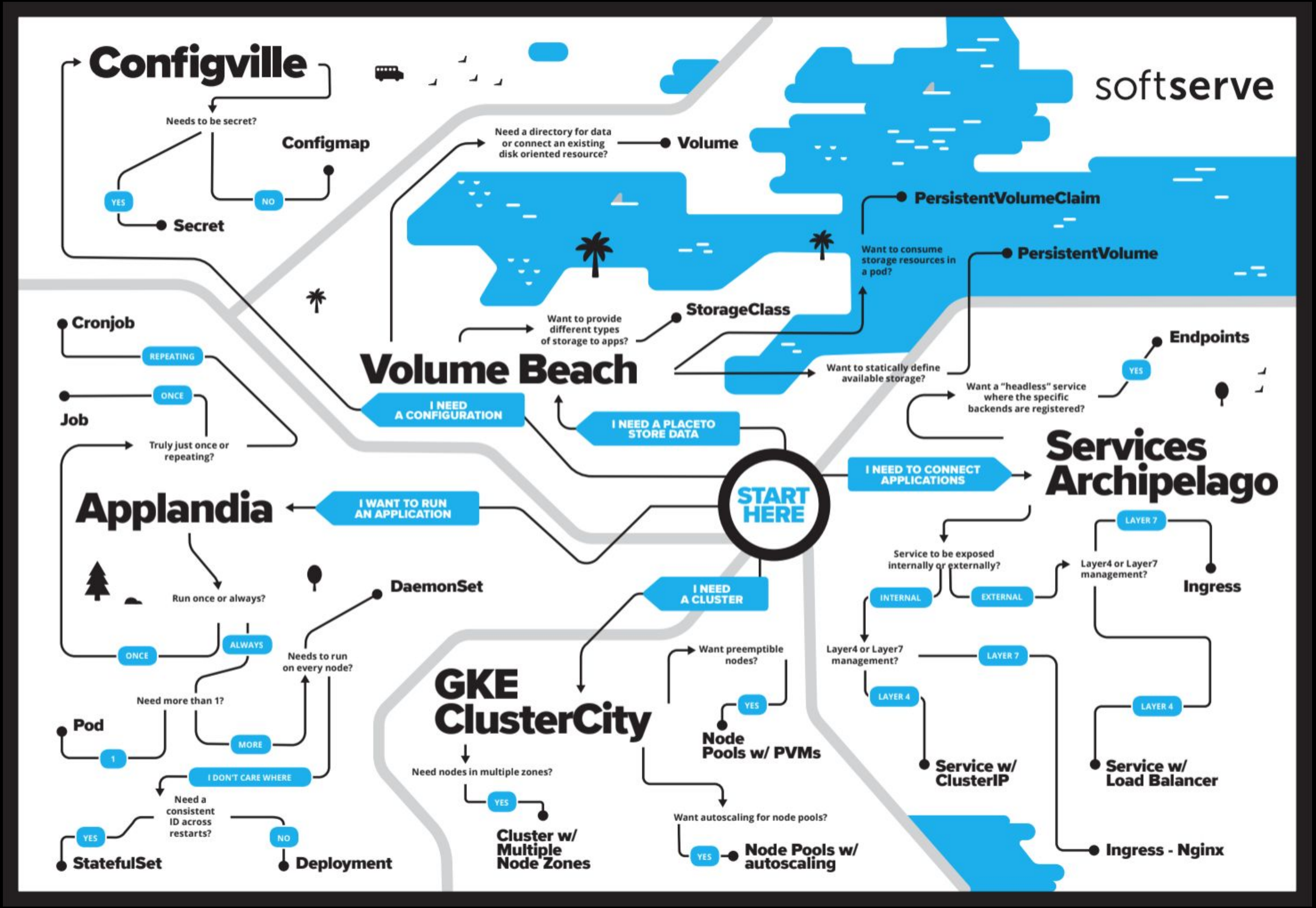
COST EFFECTIVE KUBERNETES

Myroslav Rys — Softserve
Ryan Richard — Google

softserve



softserve

START HERE

Configville

Needs to be secret?

YES

Secret

NO

Configmap

Volume Beach

Need a directory for data or connect an existing disk oriented resource?

Volume

Want to provide different types of storage to apps?

StorageClass

Want to consume storage resources in a pod?

PersistentVolumeClaim

Want to statically define available storage?

PersistentVolume

I NEED A CONFIGURATION

I NEED A PLACETO STORE DATA

I NEED TO CONNECT APPLICATIONS

Services Archipelago

Want a "headless" service where the specific backends are registered?

YES

Endpoints

Service to be exposed internally or externally?

INTERNAL

Layer4 or Layer7 management?

LAYER 4

Service w/ ClusterIP

LAYER 7

Ingress - Nginx

EXTERNAL

Layer4 or Layer7 management?

LAYER 7

Ingress

LAYER 4

Service w/ Load Balancer

I WANT TO RUN AN APPLICATION

Applandia

Truly just once or repeating?

REPEATING

Cronjob

ONCE

Job

Run once or always?

ONCE

ALWAYS

Need more than 1?

1

Pod

MORE

Needs to run on every node?

DaemonSet

I DON'T CARE WHERE

Need a consistent ID across restarts?

YES

StatefulSet

NO

Deployment

I NEED A CLUSTER

GKE ClusterCity

Need nodes in multiple zones?

YES

Cluster w/ Multiple Node Zones

Want preemptible nodes?

YES

Node Pools w/ PVMs

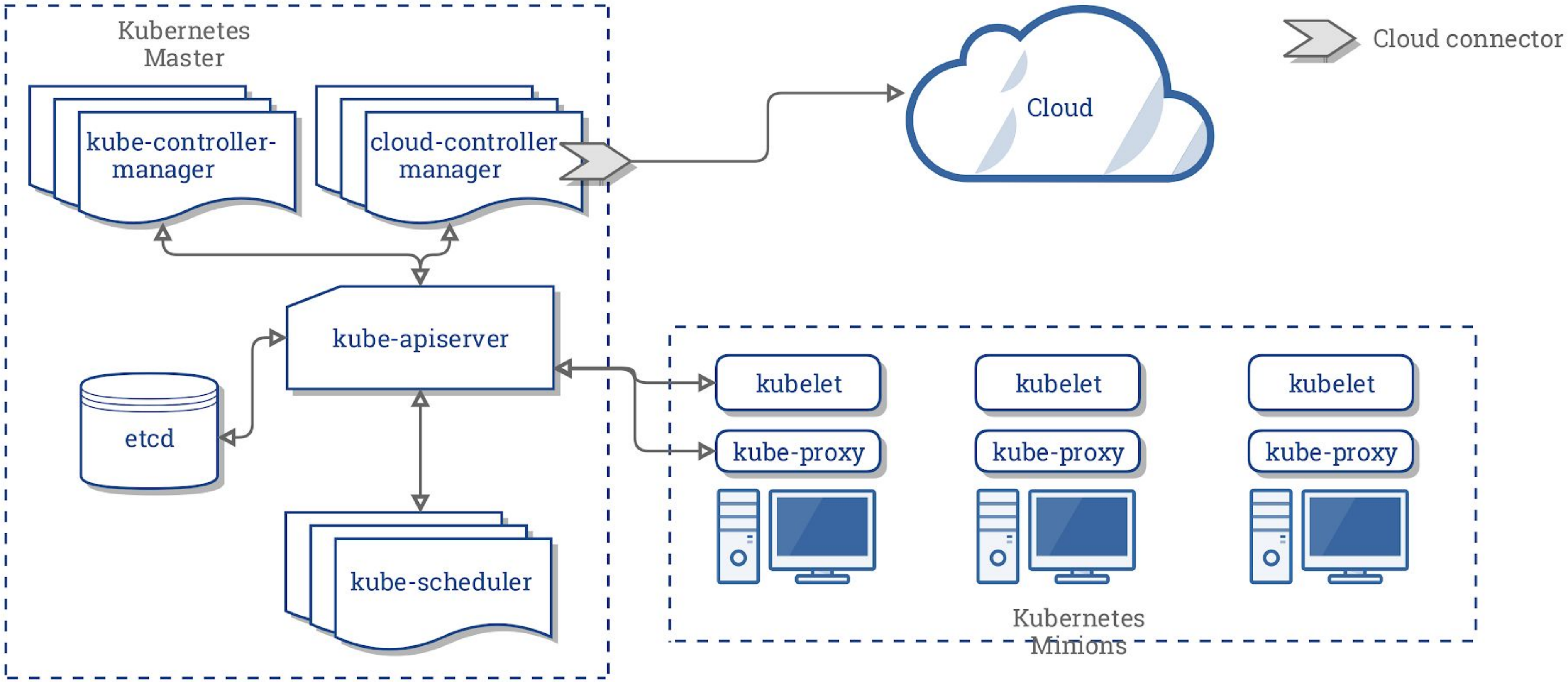
Want autoscaling for node pools?

YES

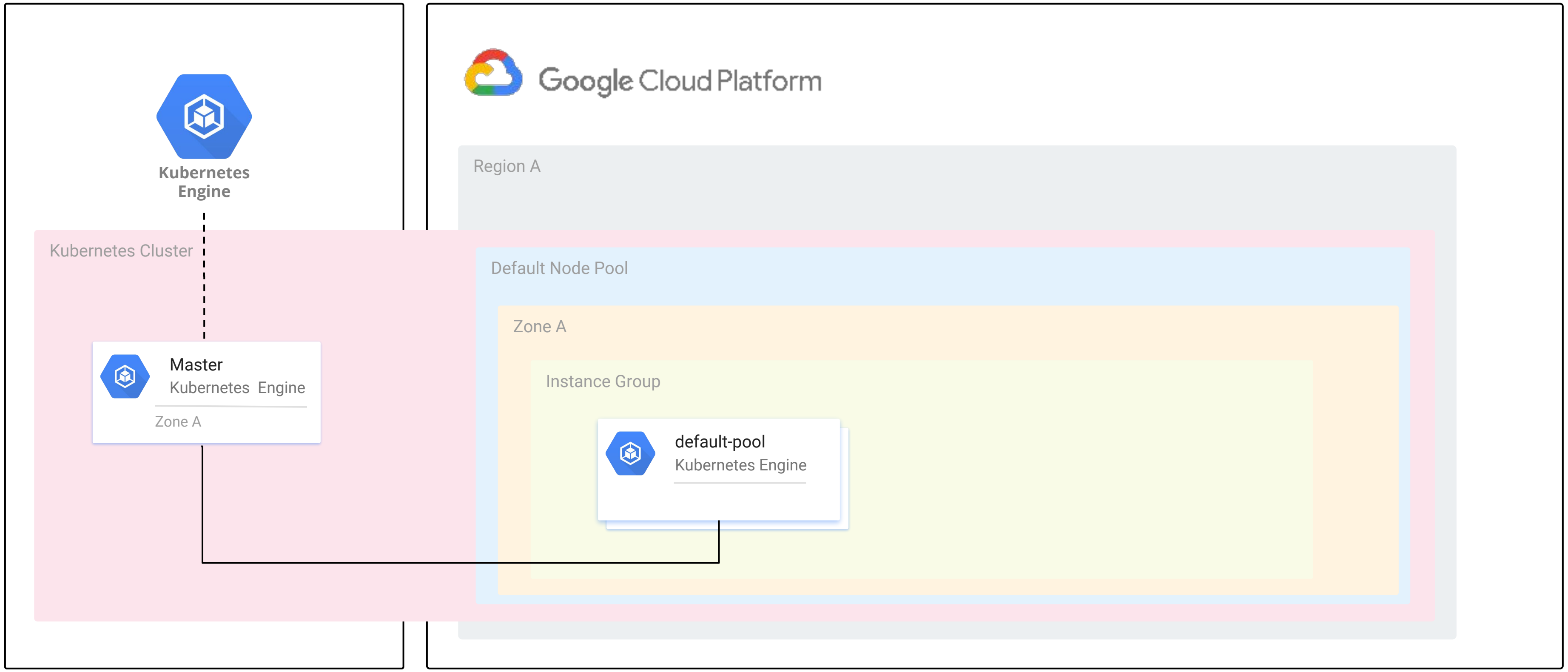
Node Pools w/ autoscaling

softserve

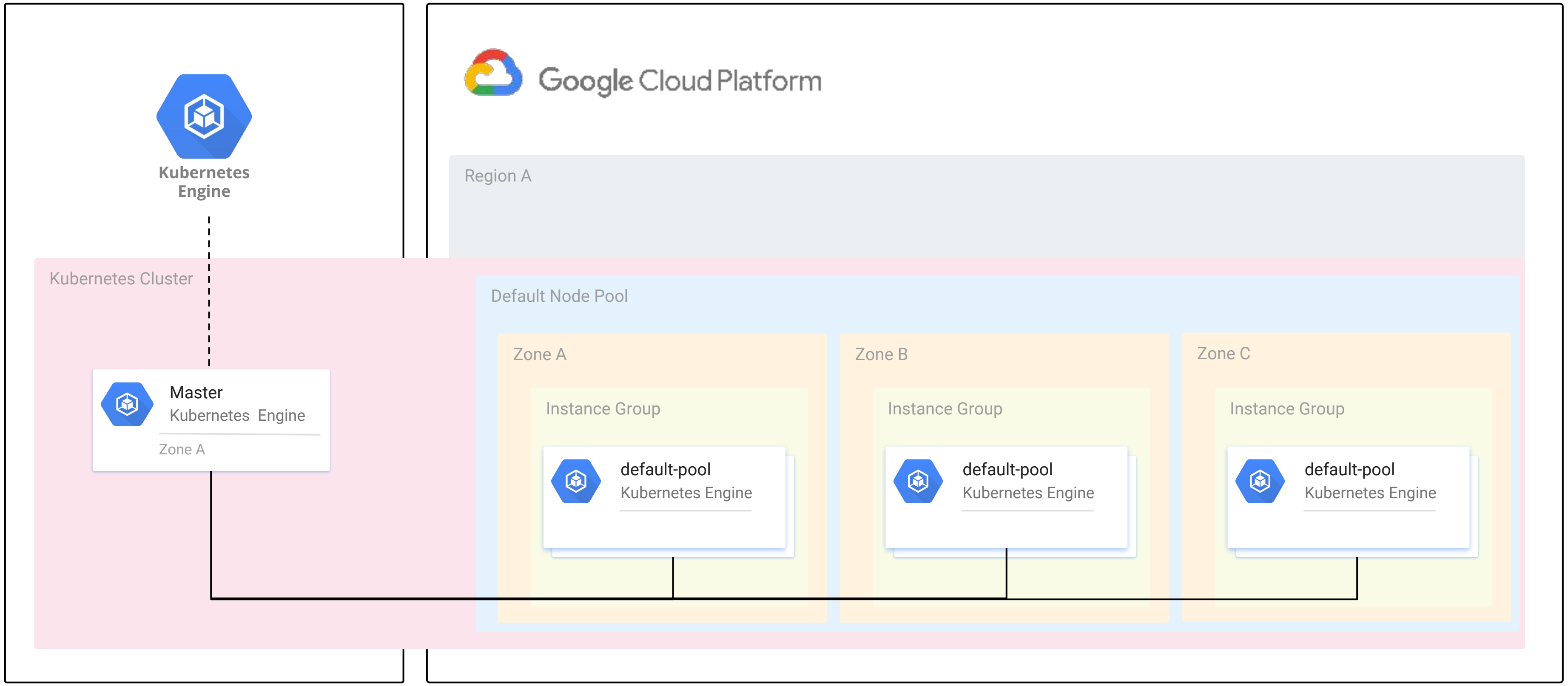
KUBERNETES ARCHITECTURE FOUNDATION



GKE — SINGLE ZONE / NODE POOL

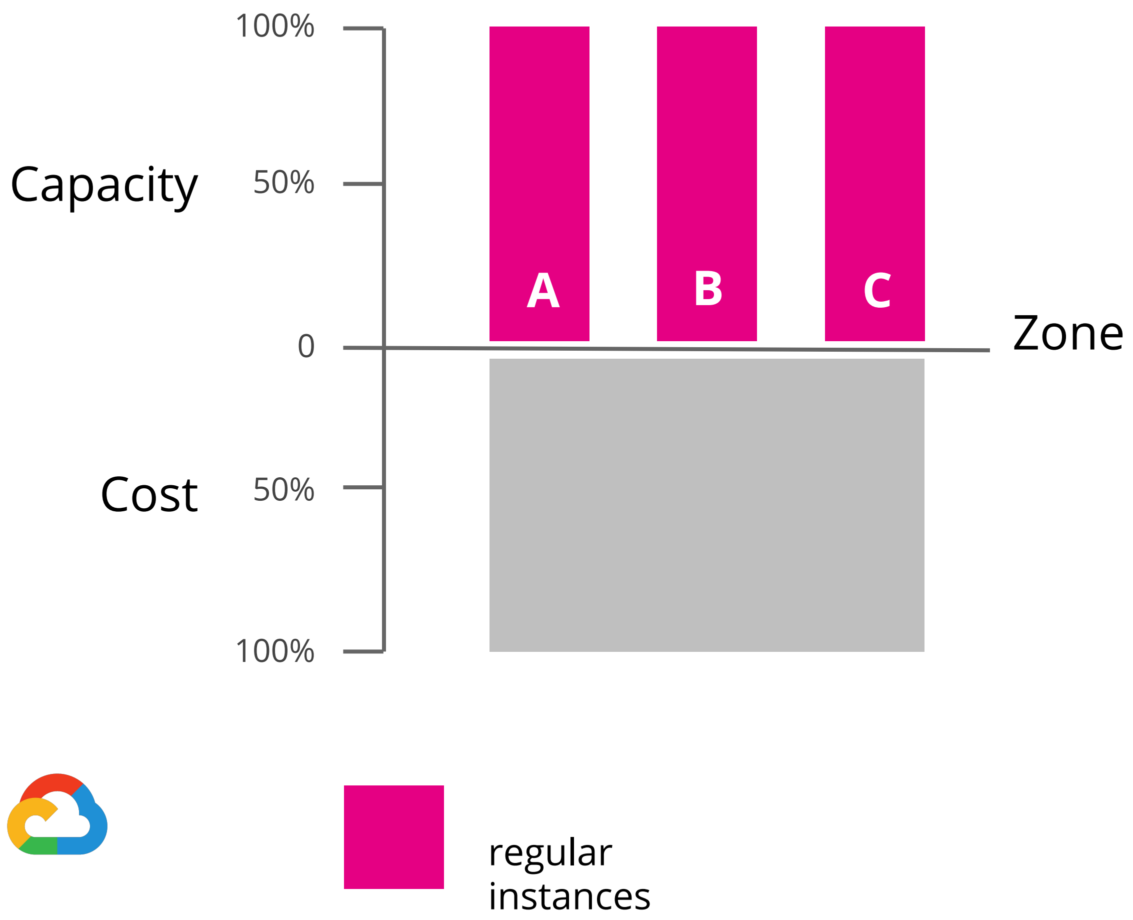


GKE — SINGLE NODE POOL, MULTI-ZONE



softserve

CAPACITY vs COST



100% of spend

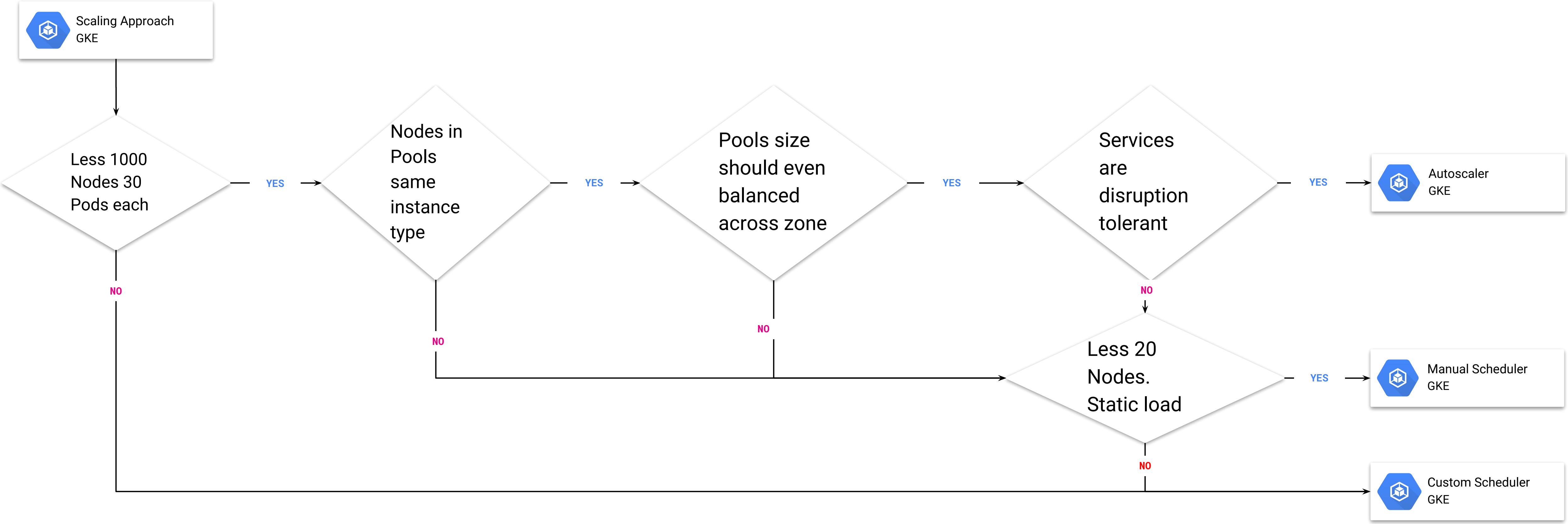
100% of the time

GKE CLUSTER AUTOSCALER

Allows for Autoscaling of the compute nodes which make up the cluster.

```
gcloud beta container clusters create meetup \
--enable-autoscaling \
--min-nodes 2 \
--max-nodes 6
...
```

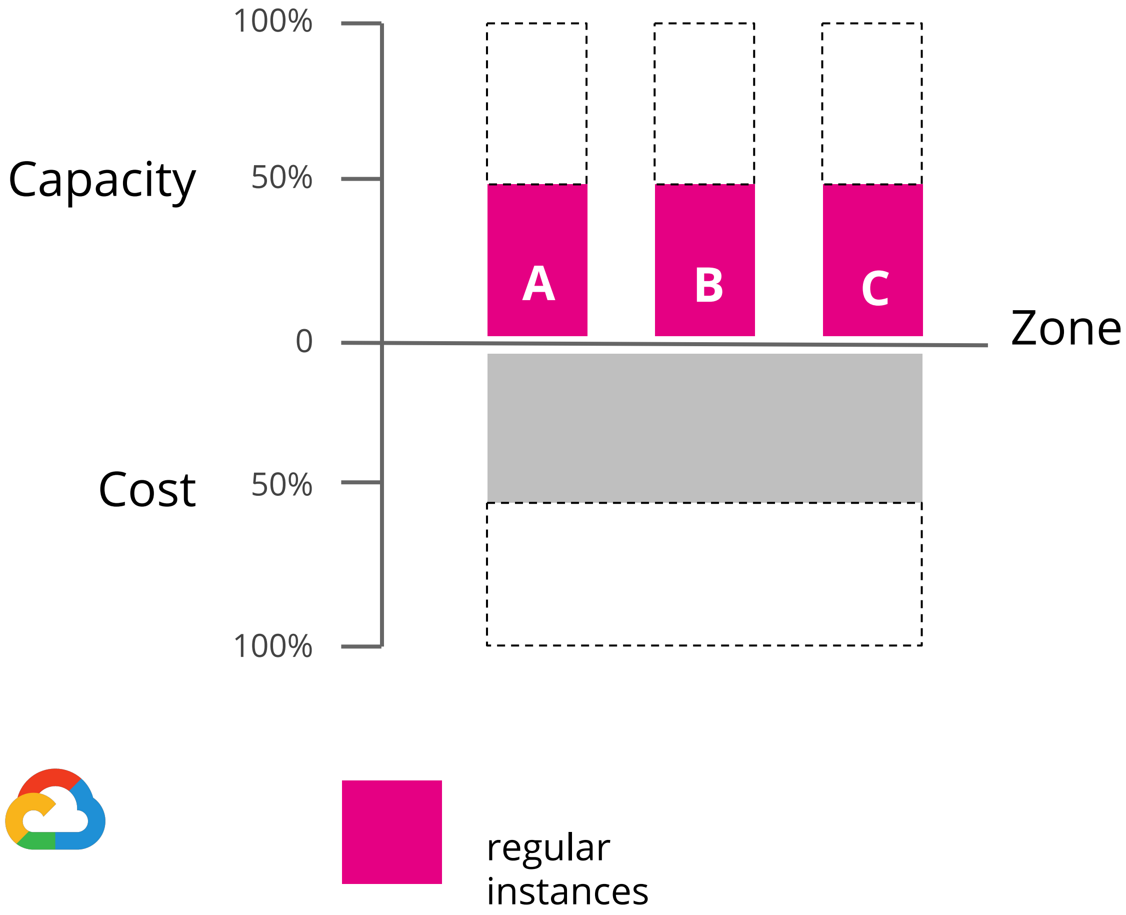
CLUSTER AUTOSCALER VS CUSTOM CONTROLLER OR MANUAL MANAGEMENT


Scaling Approach
GKE
Less 1000 Nodes 30 Pods each
YES
NO
Nodes in Pools same instance type
YES
NO
Pools size should even balanced across zone
YES
NO
Services are disruption tolerant
YES
NO
Autoscaler
GKE
Less 20 Nodes. Static load
YES
NO
Manual Scheduler
GKE
Custom Scheduler
GKE

GKE CLUSTER AUTOSCALER LIMITATIONS

- Cluster autoscaler works based on **Pod resource requests**
- Cluster autoscaler does **not track labels manually added** after initial cluster or node pool creation
- Cluster autoscaler considers the relative cost of each instance type in the node pool and attempts to expand the **least expensive** possible node pool
- Cluster with multiple node pools with the same instance type, cluster autoscaler will attempt to keep those node pools' sizes **balanced**
- Maximum period for **graceful termination** for a Pod up to 10 minutes

CAPACITY VS COST (autoscaling)



BEST CASE:

50% of spend,
100% of the time

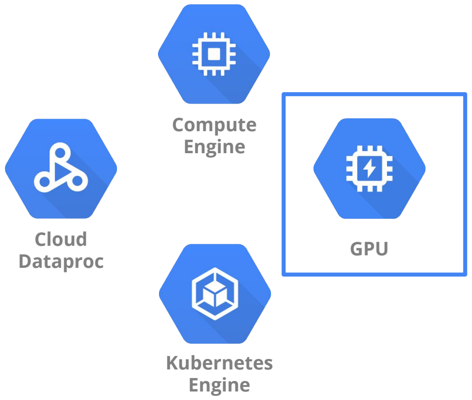
WORST CASE:

100% of spend,
100% of the time

PREEMPTIBLE VMS

A preemptible VM (PVM) is an instance that you can create and run at a much lower price than normal instances. However, Compute Engine might terminate (preempt) these instances if it requires access to those resources for other tasks.

~80% Discount!



24HR

Life (max)

PREEMPTIBLE VMS STATS

580,000

10% - 15%

20,000

softserve

PREEMPTIBLE VMS STATS

580,000

cores for 1 HPC workload

10% - 15%

Average Preemption Rate*

$20,000

over a weekend for HPC workload

softserve

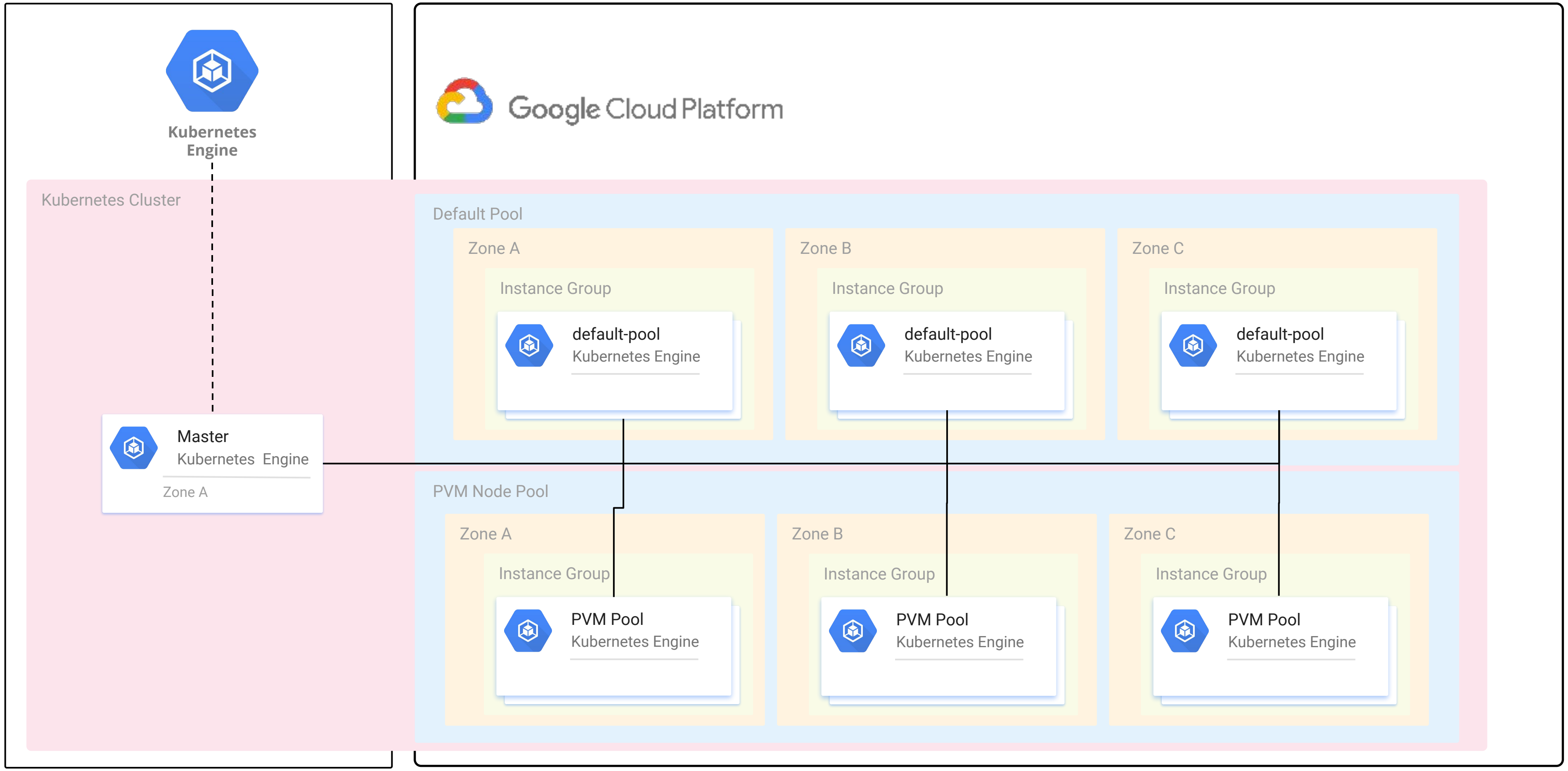
EXTEND GKE CLUSTER WITH PVM NODE POOL DEMO

softserve

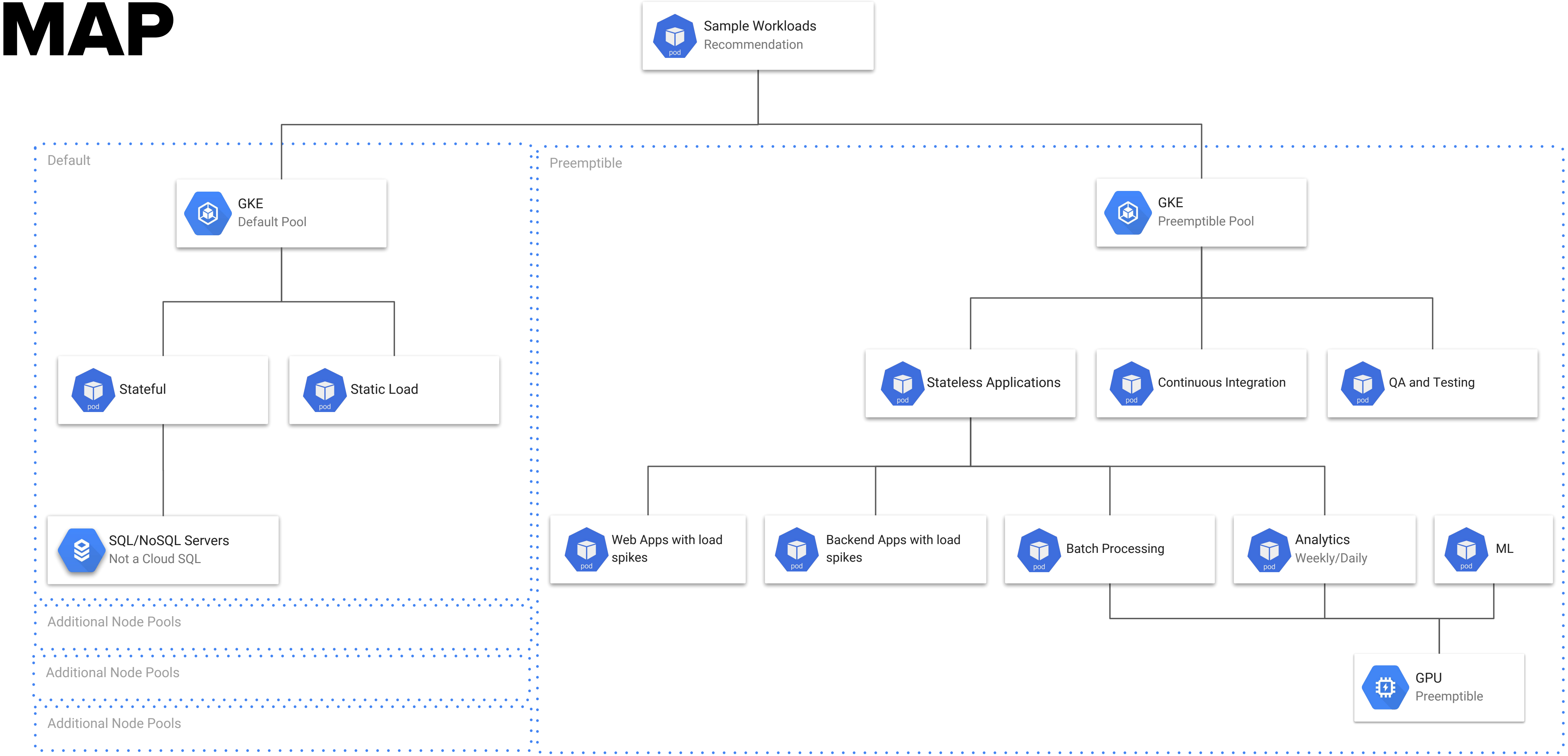
RECOMMENDED WORKLOAD FOR PVM NODE POOL

```
gcloud container node-pools create pvm-pool \
  --cluster $CLUSTER_NAME \
  --zone $CLUSTER_ZONE \
  --scopes cloud-platform \
  --enable-autoupgrade \
  --preemptible \
  --num-nodes 1 --machine-type g1-small \
  --enable-autoscaling --min-nodes=1 --max-nodes=6
```

GKE — MULTI-ZONE / MULTIPLE NODE POOLS



GKE — PREEMPTIBLE POOL DECISION MAP

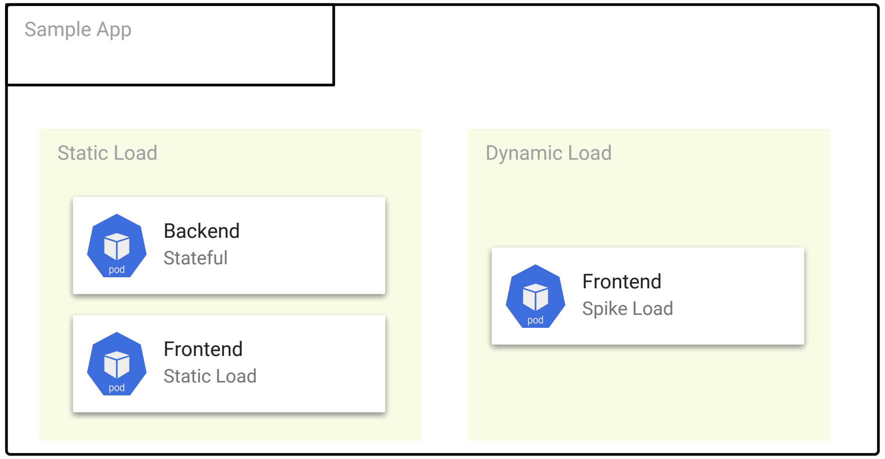


EXAMPLE 1

Simple App manager by Autoscaler

softserve

SAMPLE APP



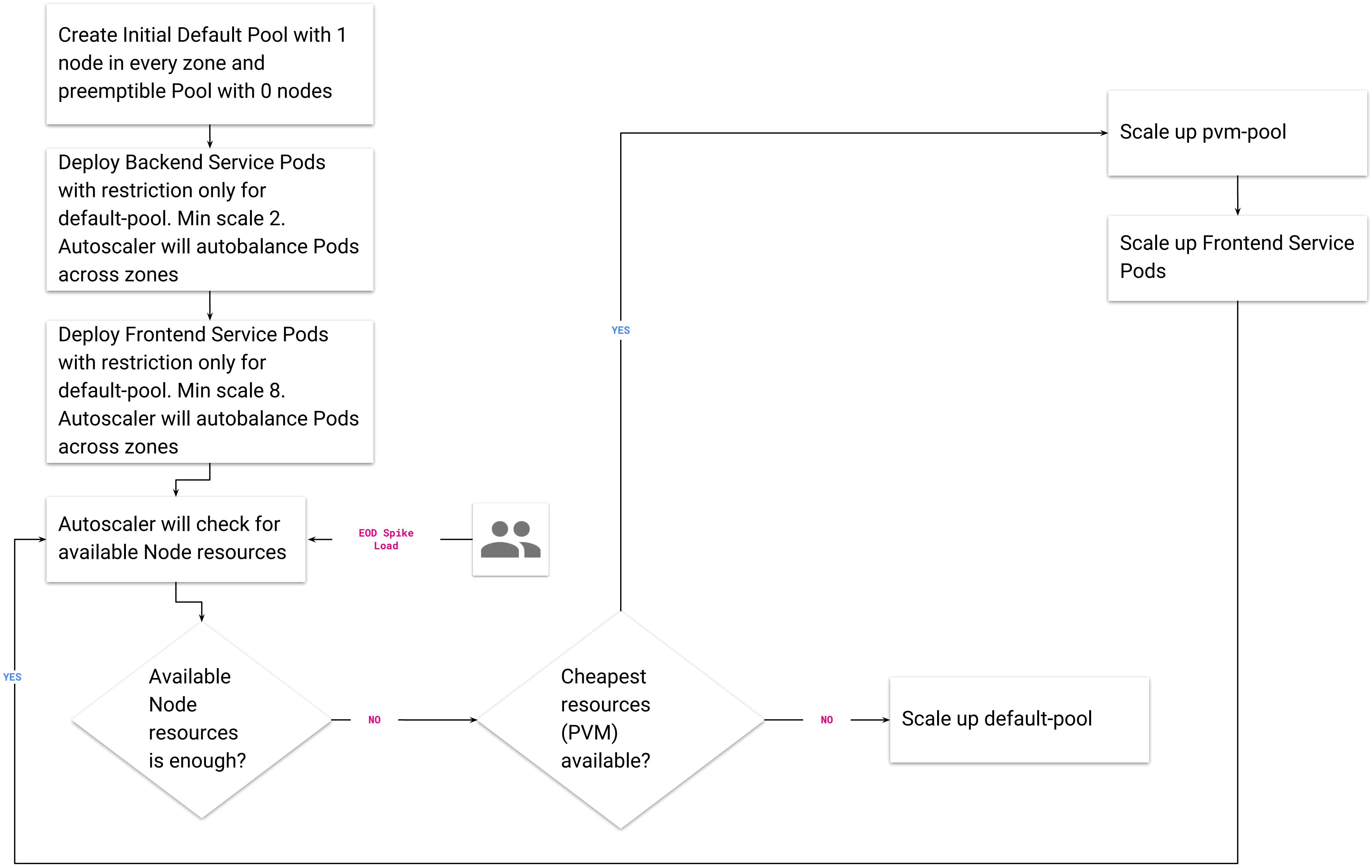
Old-style Java monolithic enterprise app splitted for two major parts

- **Backend Java** old-style stateful service
- New and shiny **Node.js Frontend** service
  - Predictable static load
  - High load spikes at EOD/EOW

Nodes in preemptible pool will have label

```
cloud.google.com/gke-preeptible: true
```

softserve



Create Initial Default Pool with 1 node in every zone and preemptible Pool with 0 nodes

Deploy Backend Service Pods with restriction only for default-pool. Min scale 2. Autoscaler will autobalance Pods across zones

Deploy Frontend Service Pods with restriction only for default-pool. Min scale 8. Autoscaler will autobalance Pods across zones

Autoscaler will check for available Node resources

Available Node resources is enough?

Cheapest resources (PVM) available?

Scale up default-pool

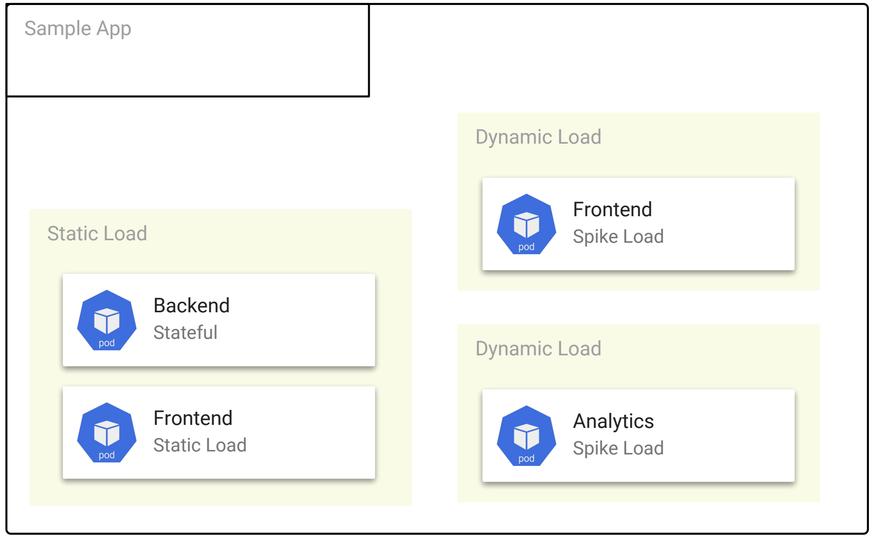
Scale up pvm-pool

Scale up Frontend Service Pods

EXAMPLE 2

Let’s add Analytics

SAMPLE APP. ANALYTICS ADDED

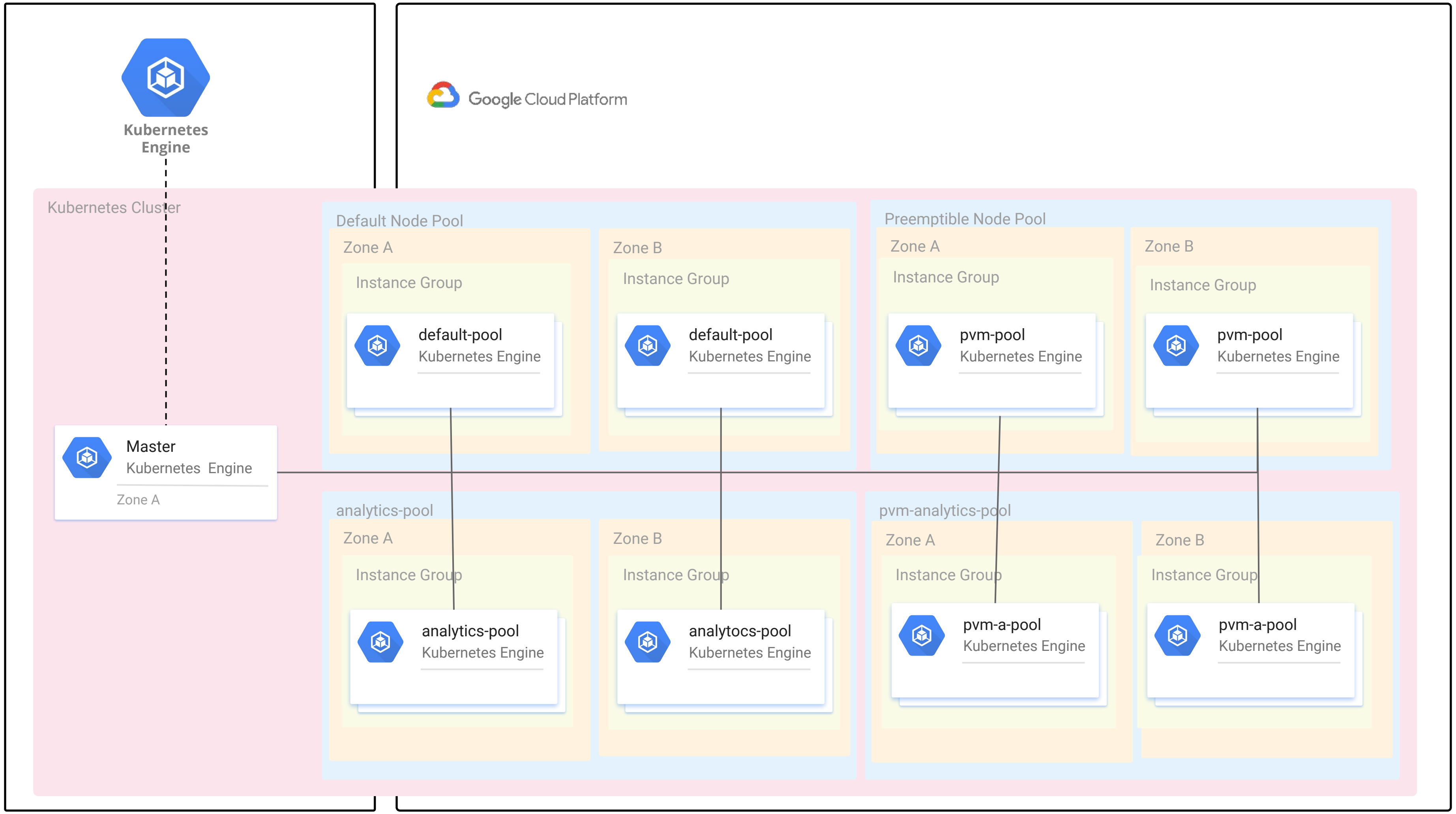


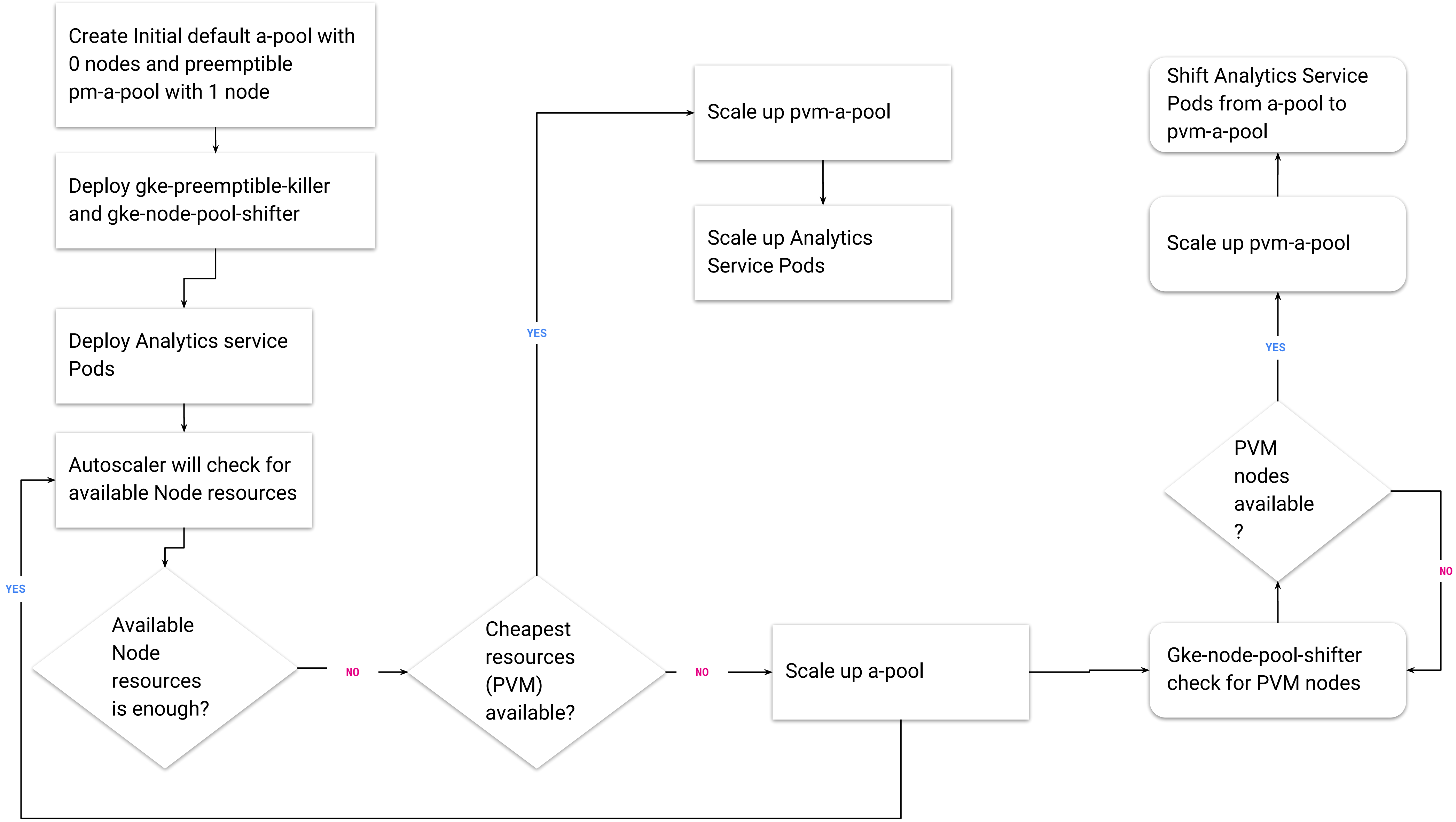
Old-style Java monolithic enterprise app splitted for two major parts

- Backend Java old-style stateful service
- New and shiny Node.js Frontend service
    - Predictable static load
    - High load spikes at EOD/EOW
- **Daily/Weekly Analytics**
    - To decrease cost let’s run heavy analytics only on PVM

softserve

GKE — MULTI-ZONE DEFAULT NODE POOL / MULTI-ZONE PREEMPTIBLE NODE POOL





Create Initial default a-pool with 0 nodes and preemptible pm-a-pool with 1 node

Deploy gke-preemptible-killer and gke-node-pool-shifter

Deploy Analytics service Pods

Autoscaler will check for available Node resources

Available Node resources is enough?

Cheapest resources (PVM) available?

Scale up pvm-a-pool

Scale up Analytics Service Pods

Scale up a-pool

Gke-node-pool-shifter check for PVM nodes

PVM nodes available ?

Scale up pvm-a-pool

Shift Analytics Service Pods from a-pool to pvm-a-pool

SCHEDULE PODS ON PVM IF AVAILABLE

Modify your Pod or Deployment spec using

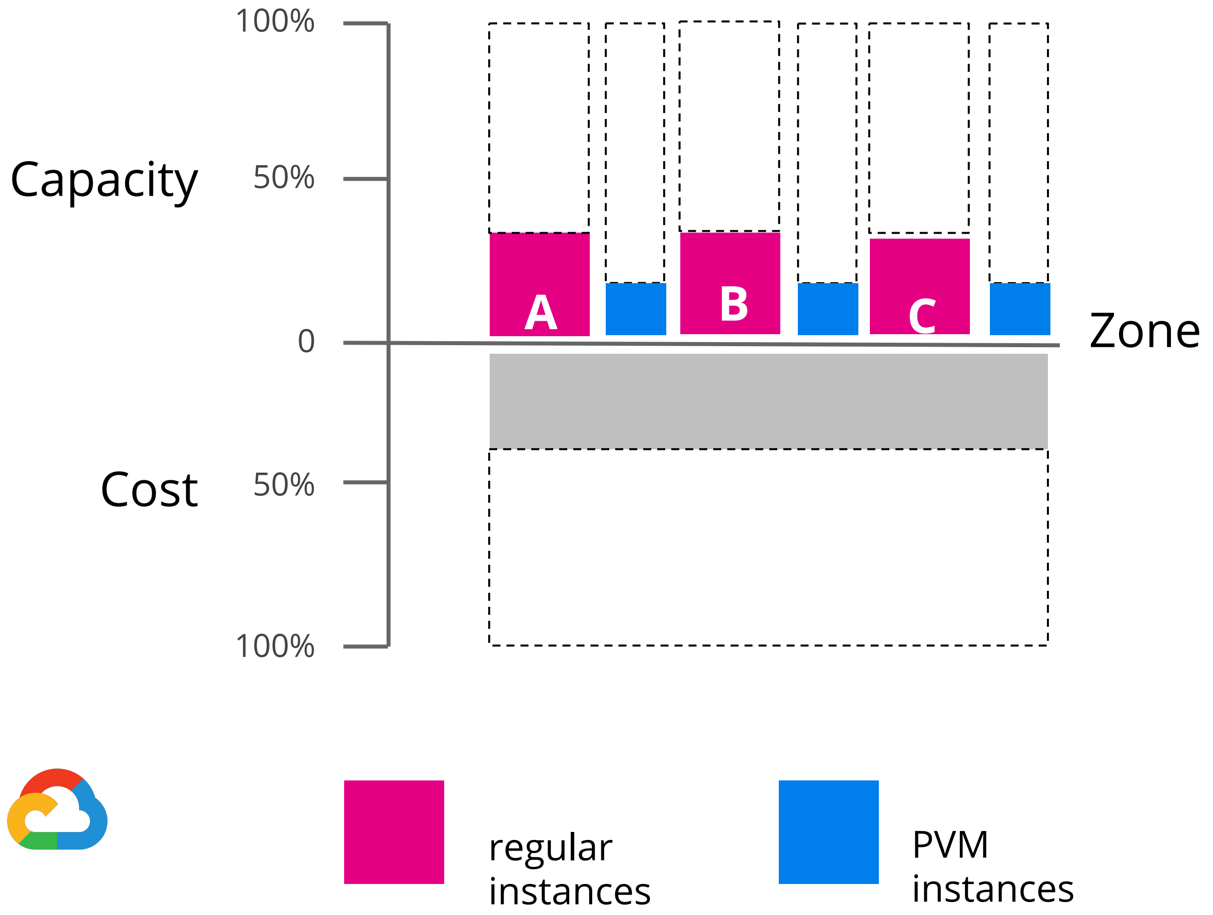
```
...
spec:
  affinity:
    nodeAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
      - preference:
          matchExpressions:
          - key: cloud.google.com/gke-preemptible
            operator: Exists
        weight: 100
...
```

SCHEDULE PODS ON NON-PVM NODES

Modify your Pod or Deployment spec using

```
...
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: cloud.google.com/gke-preemptible
            operator: DoesNotExist
...
```

CAPACITY VS COST(PVM)



BEST CASE:

34% of spend,
100% of the time

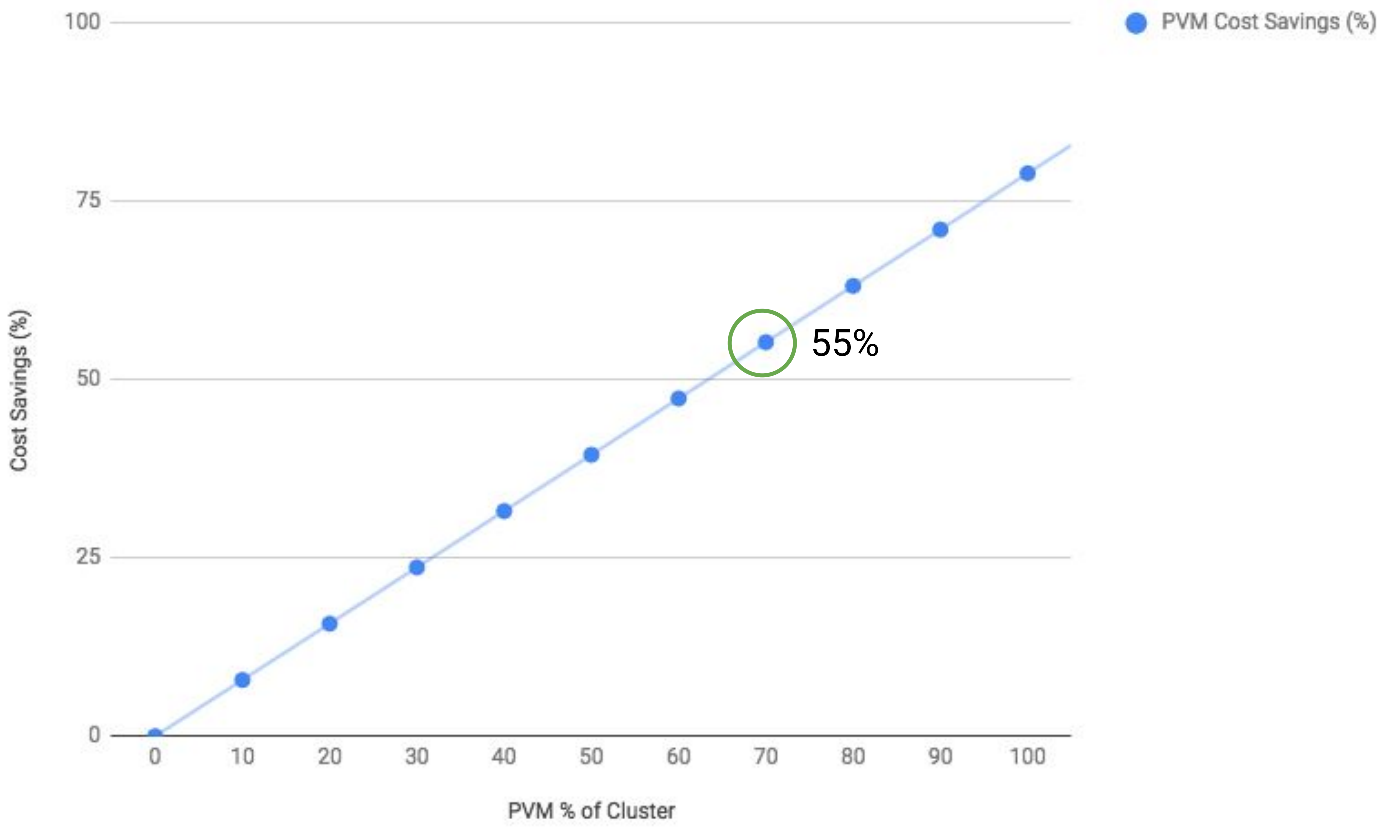
WORST CASE:

120% of spend,
100% of the time
2x capacity

SHOW STACKDRIVER

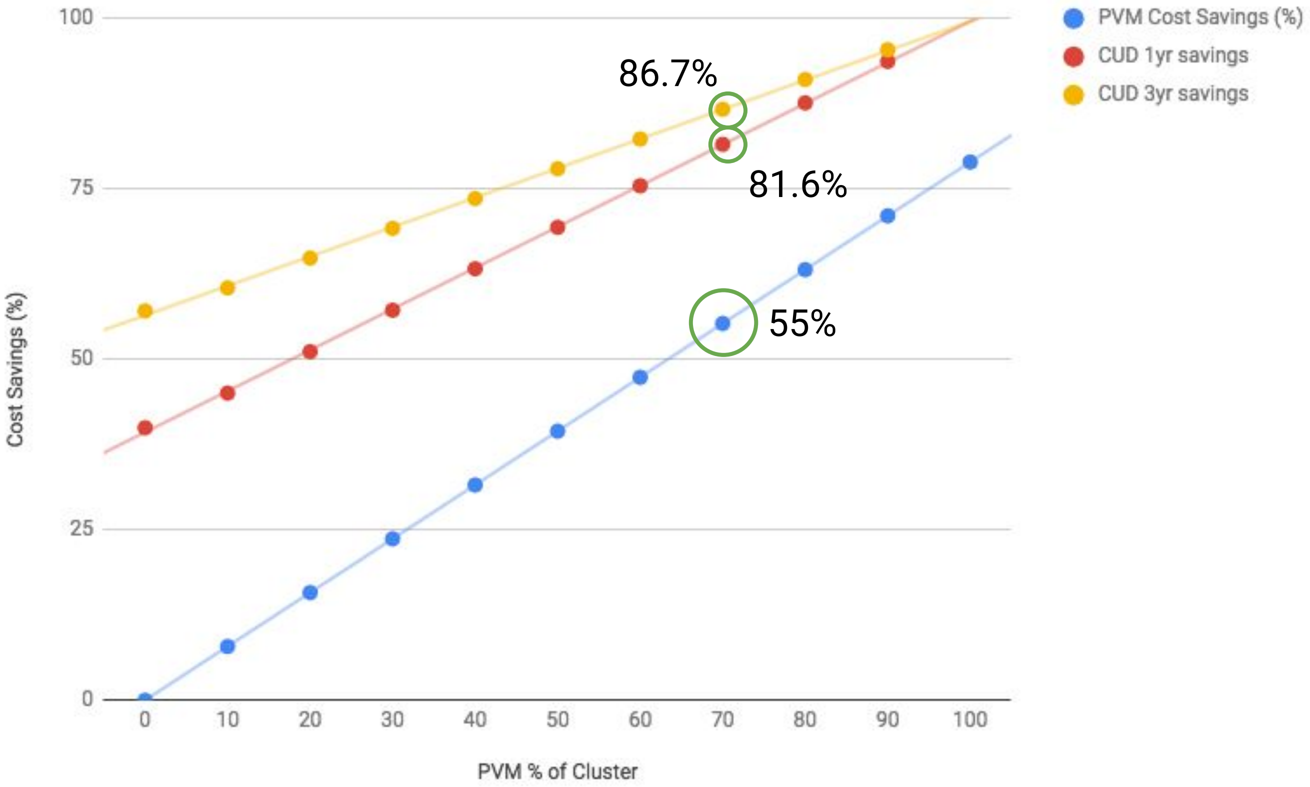
DEMO STRESS

COST SAVING WITH PVM AS % OF CLUSTER



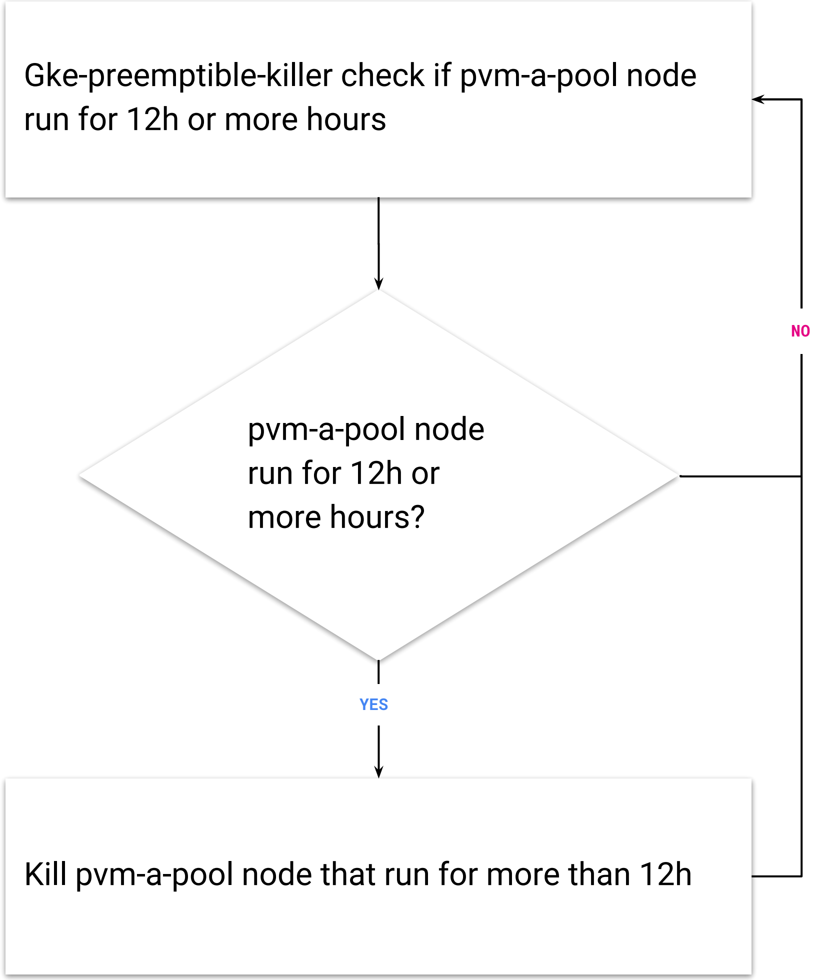
softserve

COMMITTED USE DISCOUNTS



softserve

SHOW STACKDRIVER FOR STRESS


Gke-preemptible-killer check if pvm-a-pool node run for 12h or more hours
pvm-a-pool node run for 12h or more hours?
NO
YES
Kill pvm-a-pool node that run for more than 12h

Kubernetes controller that ensures **deletion** of preemptible nodes in a GKE cluster **is spread out to avoid the risk** of all getting deleted at the same time after 24 hours

KUBERNETES CONTROLLERS FOR PREEMPTIBLE NODE POOL

- We recommend to randomly kill PVM in Preemptible Node Pool to avoid expire all nodes same time
  estafette-gke-preemptible-killer

- Another great tool will help to constantly monitor Node Pools and move nodes to preemptible PVM
  estafette-gke-node-pool-shifter

TO
THE
FUTURE
softserve

APPENDIX

DEMO repository https://github.com/stonevil/gke-meetup-demo-project

KUBERNETES AND EPHEMERAL COMPUTE

Abstract

In cloud computing, elasticity is defined as "the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible".

For those familiar with Kubernetes, this may seem like a solved problem. Not exactly. What about the underlying cluster resources? Node Autoscaling is a great feature of GKE but how can we take advantage of this in interesting, cost effective way?

In this interactive session, you will walk through few cases how-to cut Google Cloud Kubernetes Engine cluster cost with preemptible VM’s, Stackdriver and Cluster Autoscaler.

softserve

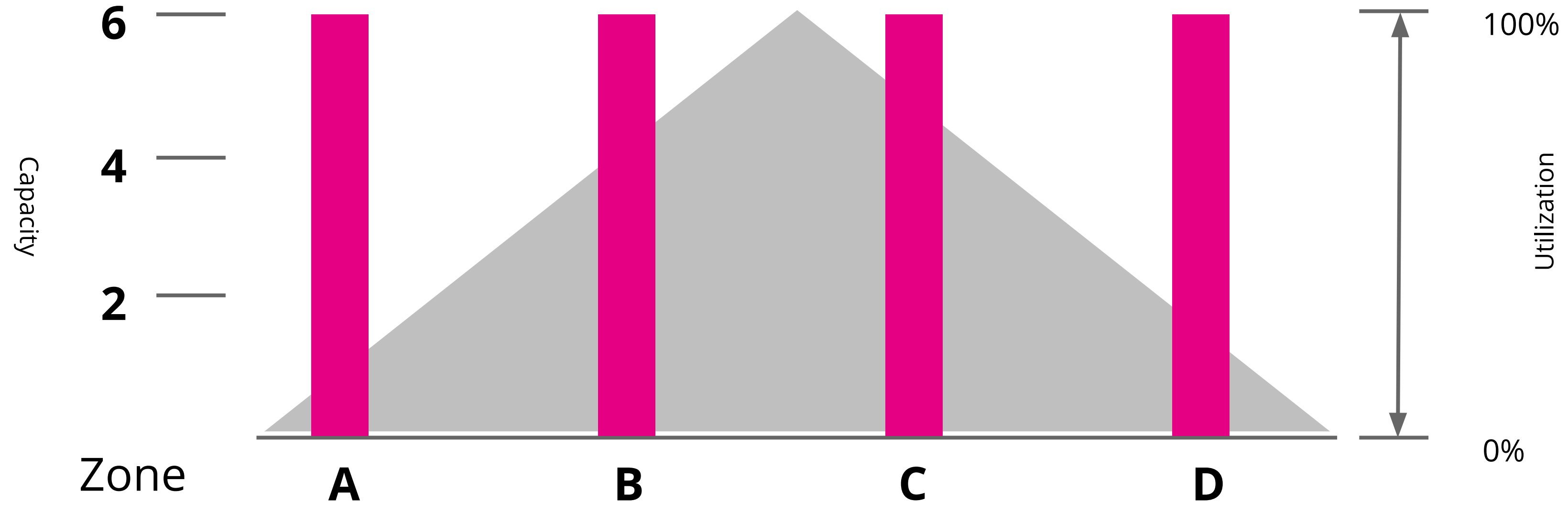
BIOS

Ryan Richard — Ryan is a Customer Engineer at Google working with enterprise customers and focused on GCP. He has background building and running services in Kubernetes and originally added the Rackspace deployment code to the repo in 2014.

Myroslav Rys — Myroslav is a Solution Architect at SoftServe Inc. More than 8 years experience in large scale enterprise solutions including SaaS / Clouds solutions. Experience building products and solutions with Kubernetes from 2015.

softserve

UTILIZATION VS CAPACITY VS COST



100% of spend,
100% of the time

SCALING (MYROSLAV)

Scaling

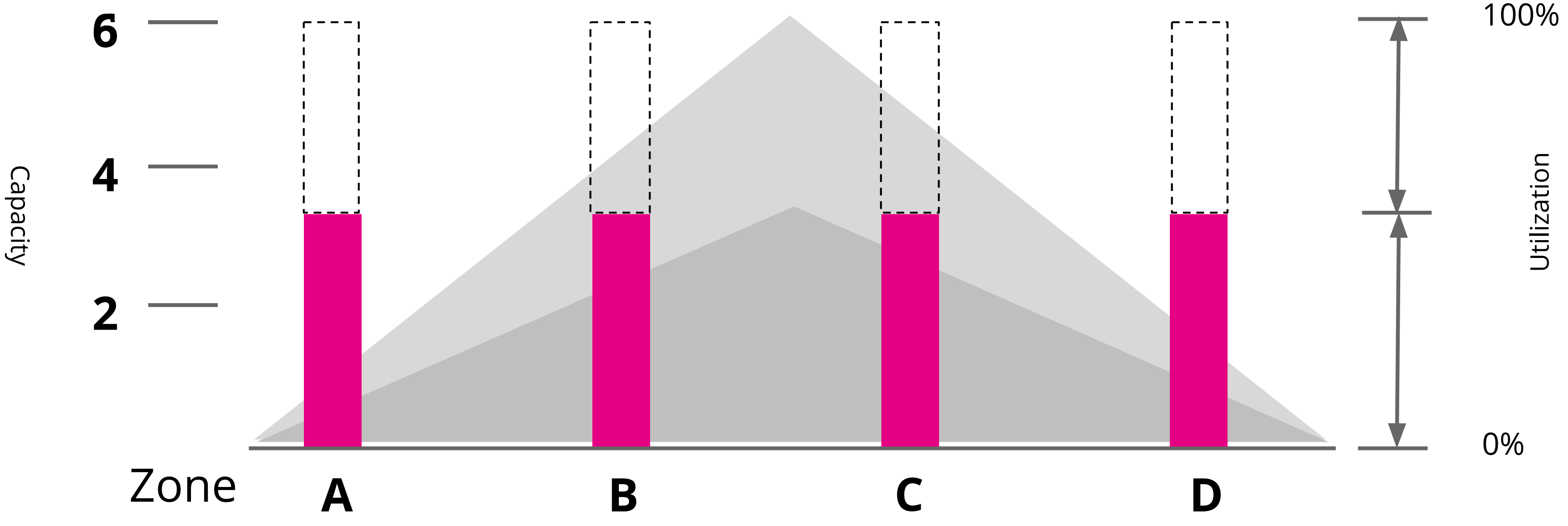
- Manually scaling, autoscaling (Cluster Autoscaler)
- Better utilization but you’re paying full price for these resources
- **what if we look at this graph, it seems that our capacity spikes only for a few hours a day (Stackdriver)**
- Or a known batch job that will increase usage for a known amount of time, want to pay the least for it.
- Is there a way to handle this capacity temporarily without paying full price?

Cluster autoscaler considers the relative cost of each instance type in the node pool and attempts to expand the least expensive possible node pool. [1]

Cluster Autoscaler only takes requested resources into account for autoscaling. It does not take current utilization into account.

[1] https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-autoscaler

UTILIZATION VS CAPACITY VS COST



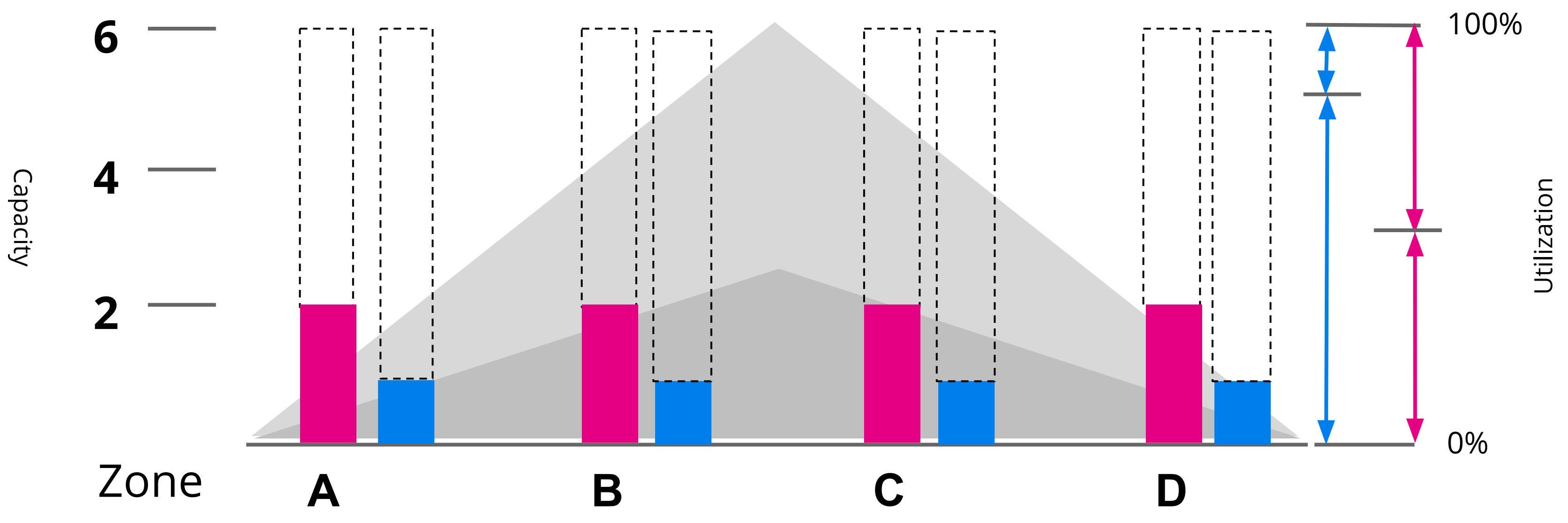
BEST CASE:

50% of spend,
100% of the time

WORST CASE:

100% of spend,
100% of the time

softserve



BEST CASE:

36.5% of spend,

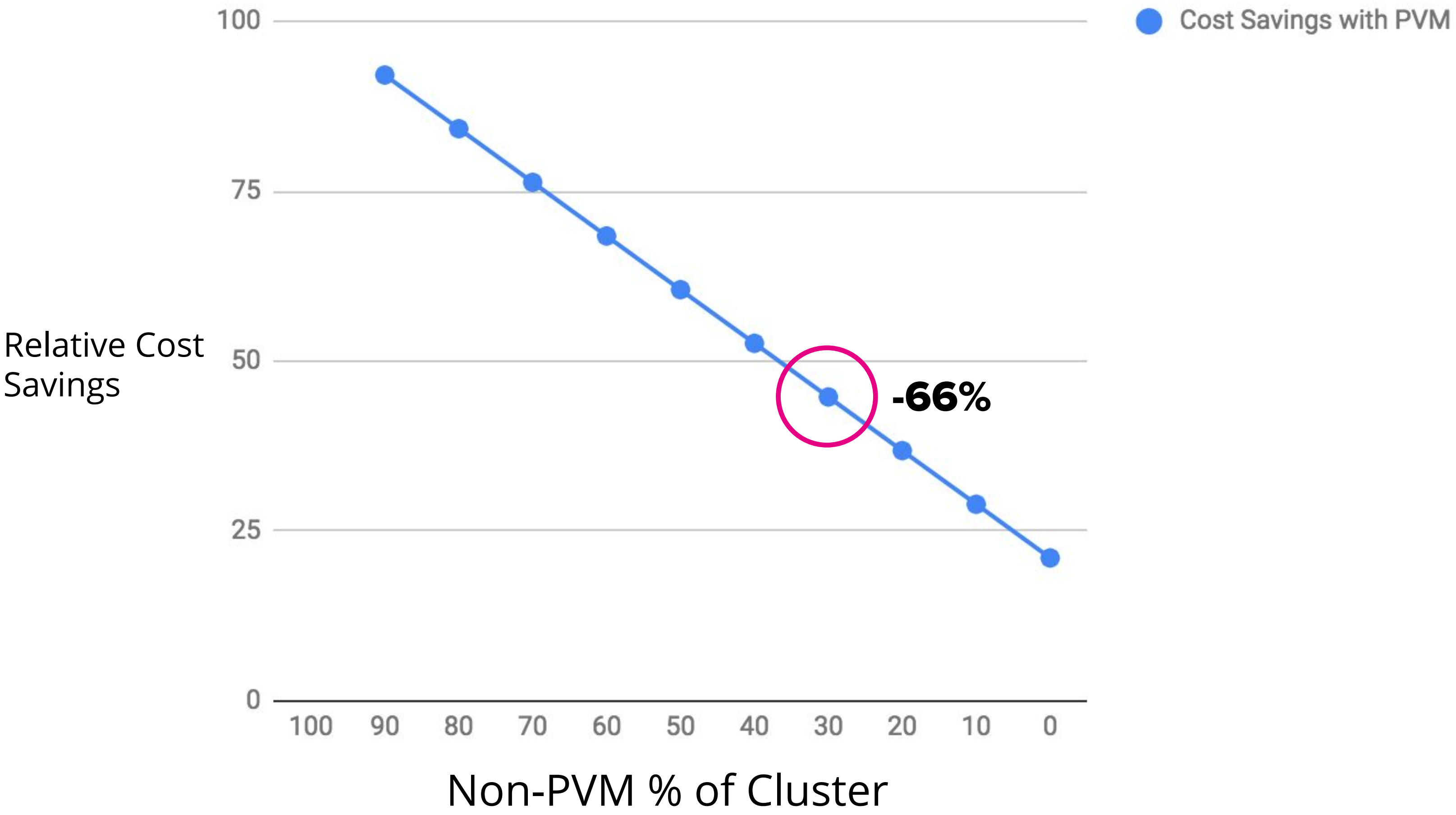
100% of the time

WORST CASE:

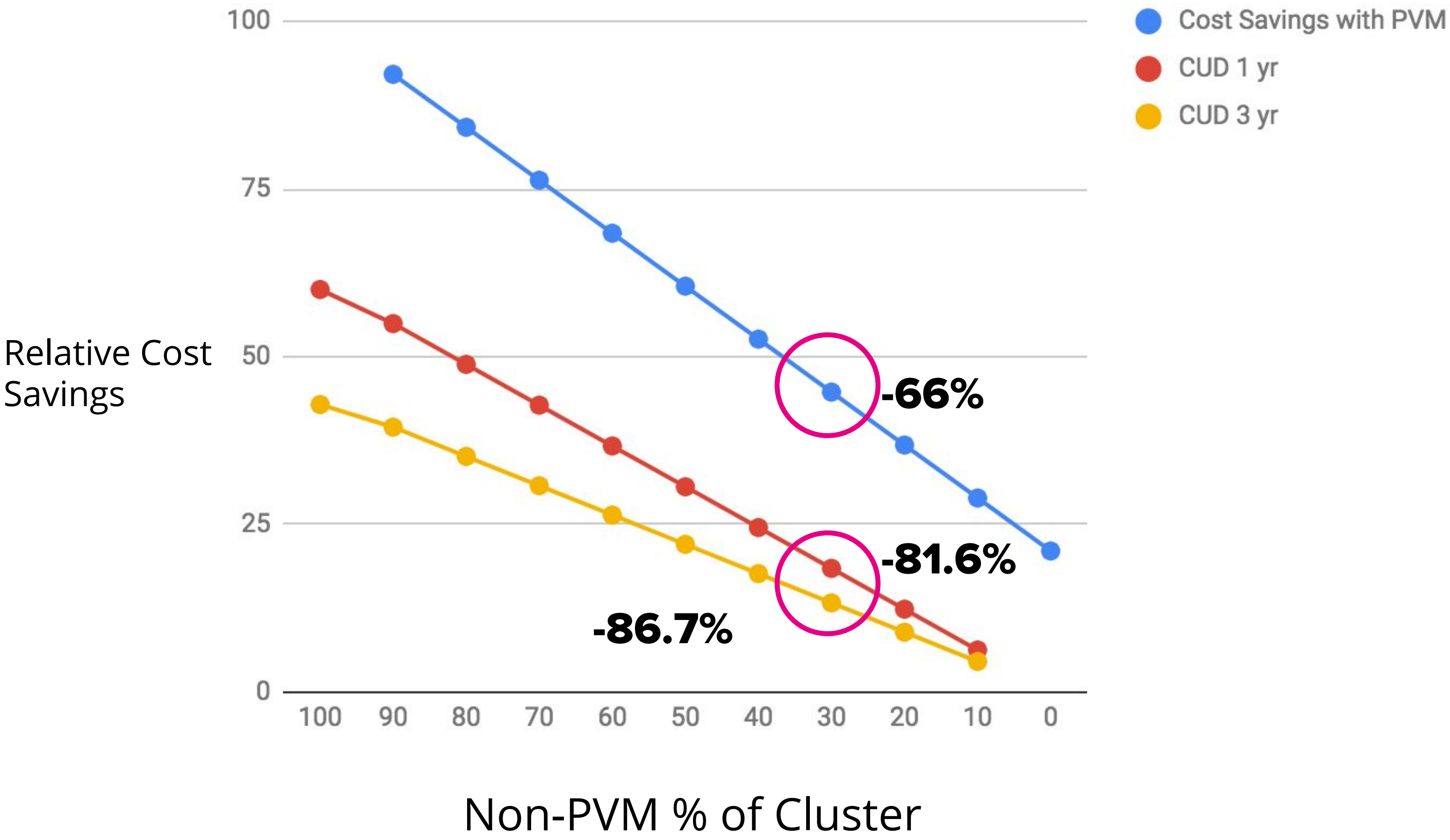
120% of spend,

100% of the time

2x capacity


Relative Cost Savings
100
75
50
25
0
100
90
80
70
60
50
40
30
20
10
0
Non-PVM % of Cluster
● Cost Savings with PVM

-66%


100
75
50
25
0
Relative Cost Savings
100
90
80
70
60
50
40
30
20
10
0
Non-PVM % of Cluster
Cost Savings with PVM
CUD 1 yr
CUD 3 yr
-66%
-81.6%
-86.7%