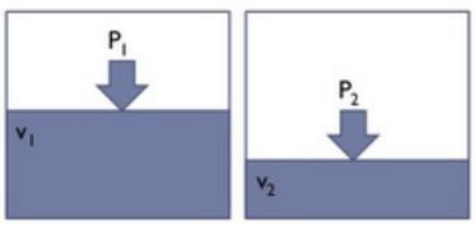
**Foundational Pattern – Predictable Demands**

- Problem:
  - K8s can manage applications written in different programming languages as long as the application can be run in a container.
  - Different languages will have different resource requirements
  - It is difficult to predict the amount of resources a container may need to function optimally.
  - Some services have a fixed CPU and memory consumption profiles, and some are spiky.
  - Some services need persistent storage to store data
  - Defining all of the application characteristics and passing them to the managing platform is fundamental pre-requisite of a cloud native application

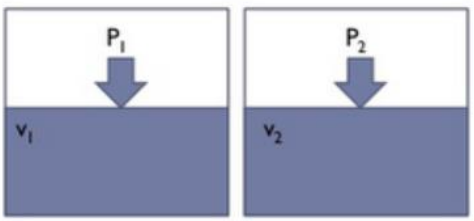
- Solution:Resource Profiles:
  - We need experimentation for figuring out the resource requirements of a container.
  - Compute resources in the context of K8s are defined as something that can be requested by and allocated to and consumed from container.
  - The resources are categorized as

    1) compressible – Can be throttled (CPU, network bandwidth)> CPU can be added dynamically.

    2) incompressible – Cannot be throttled (memory)> memory can't be added dynamically.



Compressible fluid



Incompressible fluid

  - Because of the nature and the implementation detail of your application, you need to specify the minimum amount of resources that are needed (requests) and the maximum amount of resources it can grow up to (limits)
  - The request amount is used by scheduler when placing Pods to the nodes.
  - Depending on whether we specify requests, the limits or both the platform offers a different kind of Quality of Service (QOS)

- *Best Effort*: Pods that does not specify requests and limits for its containers. Such Pod is considered as lowest priority and is most likely killed first when the node where the Pod is place run's out of incompressible resources
- *Burstable*: Pod that has requests and limits defined (limits are larger than requests). Such a Pod has minimal resource guarantees but is also willing to consume more resources up to its limit when available. When the node is under incompressible resource pressure these Pods are likely to be killed if not Best Effort Pods
- *Guaranteed*: Pods that has an equal amount resources and limit resources. These are the highest priority pods and guaranteed not be killed before Best-Effort and Burstable-Pods

Kill Criteria: Best then Burstable and then Guaranteed (Very less chances to kill)

```yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
    - image: nginx
      name: nginx-container
      ports:
        - protocol: TCP
          containerPort: 80
      resources:
        requests:
          cpu: 100m
          memory: 100Mi
        limits:
          cpu: 200m
          memory: 200Mi
```