133 In-place Resize of pods

Before discussing about "Vertical pod autoscaler," we'll discuss about "in-place resize of pod resources"

what is in-place resize of pod resources

As of K8S v 1.32 has released

⇒ If you change resource requirements of a pod in a deployment, the default behaviour is to delete the existing pod & then spin up a new pod with the new changes

⇒ So any changes to the resource definitions on a pod doesnot happen in place, which means the pod needs to be killed & the new pod with the new resource definition needs to be created. So this is the default behaviour

⇒ we know that this can be disruptive especially for stateful workloads. So there is an improvement that is being worked upon called as inplace update of pod resources [Resize CPU and Memory Resources assigned to Containers]

This is feature that is currently in alpha as of K8S release 1.27 & is not enabled by default when it goes to beta/stable stage it will get enabled by default

To enable this feature

you must set the "feature" flag" called "InPlacePodVerticalScaling" to "True"

$FEATURE_GATES = InPlacePodVerticalScaling = true

Once this is enabled, the pod definition supports a set of resize policy parameters

⇒ The new resize policy options allow you to specify a restart policy for each resource, In our eg we have defined that a change in CPU resource will not require the pod to be restarted & a change in memory will require a restart of the pod & we make change in "resource" such as updating the cpu to one & you can see that pod doesnot need to be killed instead it can ~~be~~ simply be updated with the new resources & so it can just increase in size

nginx.yaml

```
apiVersion: apps/v1
Kind: Deployment
metadata:
  name: my-app
  spec:
    replicas: 1
    selector:
      match Labels:
        app: my-app
    template:
      metadata:
        labels:
          app: my-app
      spec:
        containers:
        - name: my-app
          Image: nginx
          resizePolicy:
            - resourceName: cpu
              restart Policy: NotRequired
            - resourceName: memory
              restartPolicy: RestartContainer
          resources:
            requests:
              cpu: "1"
              memory: "256Mi"
            Limits:
              cpu: "500m"
              memory: "512Mi"
```

so this explains how we can resize a CPU or a memory resource assigned to a container without really restarting it. This is in alpha with version 1.33 release of k8s

Still we're discussing about manually scaling of pod we haven't started about "Vertical pod autoscaler" yet"

Limitations of in-place resizing of pods

* This only works for "cpu" & "memory" resources
  Only CPU & memory resources can be changed

* Pod QoS class cannot be changed

* Init containers and Ephemeral Containers cannot be resized with this approach

* Resource requests and limits cannot be removed once set

* A container's memory limit may not be reduced below its usage. If a request puts a container in this state, the resize status will remain in Inprogress until the desired memory limit become feasible.