REQUESTS & LIMITS IN KUBERNETES

Requests and Limits are part of **resource management** and help control the CPU and memory (resources) consumed by containers within a pod.

1. Resource Requests

- A **request** is the amount of CPU or memory guaranteed to a container.
- Kubernetes uses requests to determine which node can schedule a pod.
- If a node has enough resources to fulfill the request, the pod will be scheduled on that node.

2. Resource Limits

- A **limit** is the maximum amount of CPU or memory a container can use.
- If a container exceeds its memory limit, it will be terminated, and if it exceeds its CPU limit, its usage will be throttled.

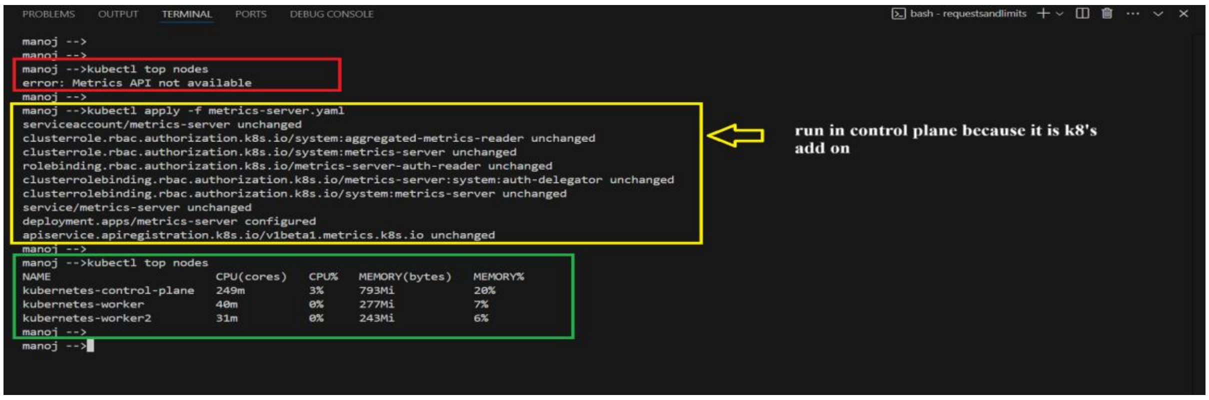
➔ **Some of the issues we can face in request and limits are:**

- **Insufficient Memory (or Insufficient CPU)**: The node doesn't have enough memory or CPU available to fulfill the pod's resource request.
- **OOMKilled**: The container was terminated because it exceeded its memory limit.
- **PodFitsResources**: The pod can't be scheduled because no node has sufficient resources to meet the pod's requests.
- **ResourceQuotaExceeded**: The pod exceeds the resource quota set for the namespace, preventing it from being created.
- **LimitRangeExceeded**: The pod's resource requests or limits exceed the constraints defined by the LimitRange policy in the namespace.
- **PodEvicted**: The pod was evicted due to resource pressure on the node, often because of insufficient memory.
- **ContainerCannotRun**: The container fails to start due to insufficient resources or incorrect configuration.
- **FailedScheduling**: Kubernetes was unable to find a suitable node to schedule the pod due to resource constraints.

For metric-server yaml use the link below to find the code

https://drive.google.com/file/d/1CIbZkM_71xn4psUrGeIhfWTaKfD_5oZu/view?usp=drive_link

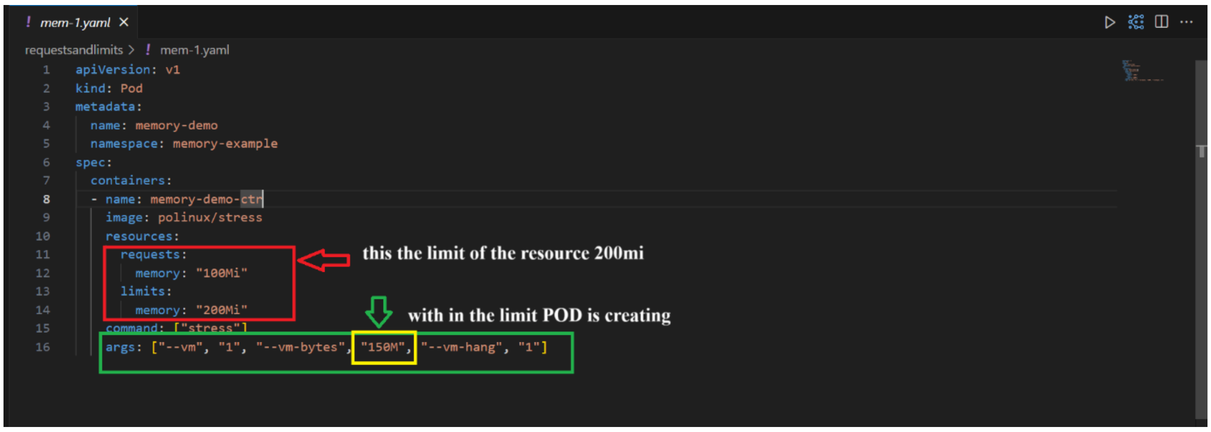
Run the metric-server on kube-system namespace present in the control plane



```
manoj -->
manoj -->
manoj -->kubectl top nodes
error: Metrics API not available
manoj -->
manoj -->kubectl apply -f metrics-server.yaml
serviceaccount/metrics-server unchanged
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader unchanged
clusterrole.rbac.authorization.k8s.io/system:metrics-server unchanged
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader unchanged
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator unchanged
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server unchanged
service/metrics-server unchanged
deployment.apps/metrics-server configured
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io unchanged
manoj -->
manoj -->kubectl top nodes
NAME                      CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
kubernetes-control-plane  249m         3%     793Mi           20%
kubernetes-worker         40m          0%     277Mi           7%
kubernetes-worker2        31m          0%     243Mi           6%
manoj -->
manoj -->
```

REQUESTS & LIMITS IN KUBERNETES

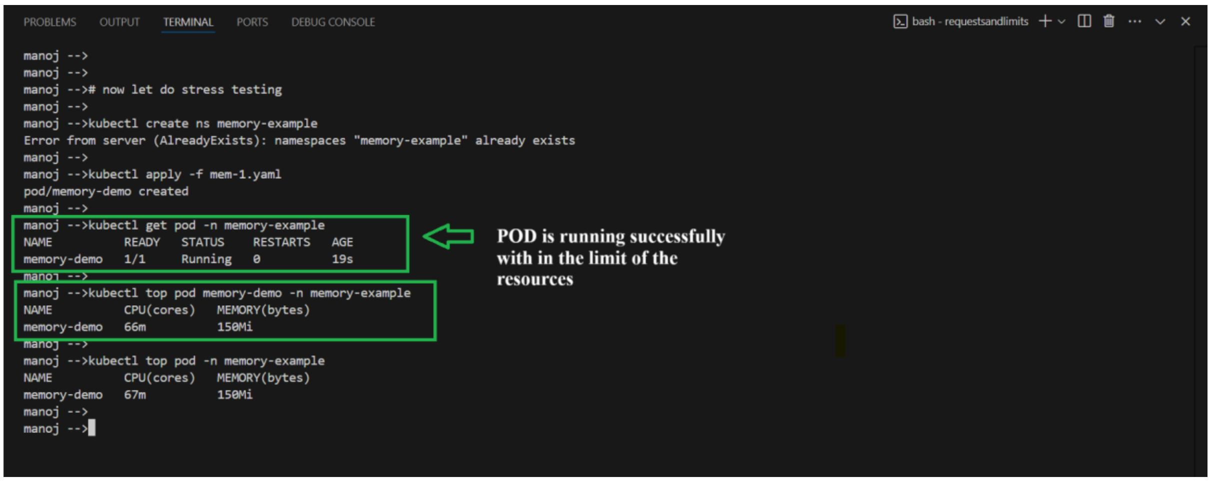
Case 1: Request memory “100Mi” and limit the memory to “200Mi”. with in the limit POD is created.



```yaml
apiVersion: v1
kind: Pod
metadata:
  name: memory-demo
  namespace: memory-example
spec:
  containers:
  - name: memory-demo-ctr
    image: polinux/stress
    resources:
      requests:
        memory: "100Mi"
      limits:
        memory: "200Mi"
    command: ["stress"]
    args: ["--vm", "1", "--vm-bytes", "150M", "--vm-hang", "1"]
```

this the limit of the resource 200mi

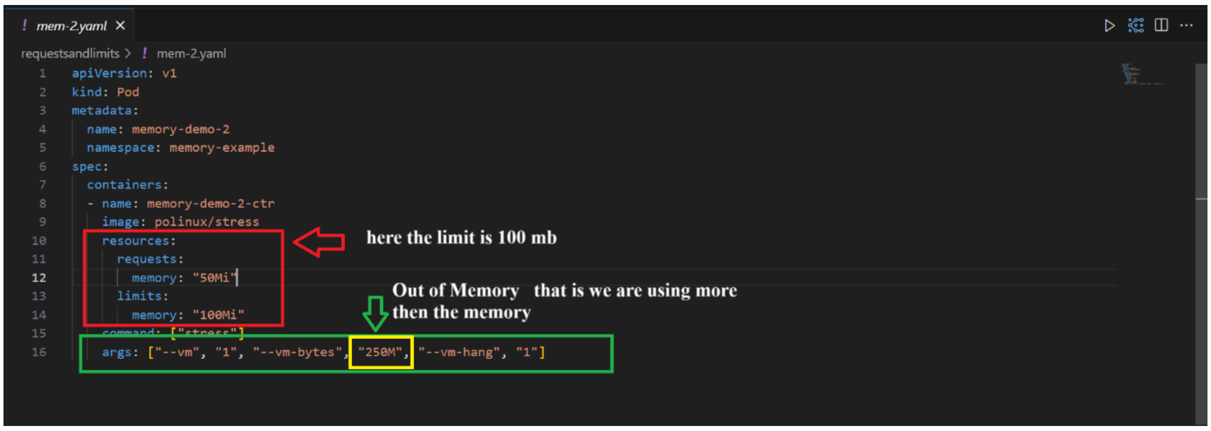
with in the limit POD is creating



```
manoj -->
manoj -->
manoj -->#now let do stress testing
manoj -->
manoj -->kubectl create ns memory-example
Error from server (AlreadyExists): namespaces "memory-example" already exists
manoj -->
manoj -->kubectl apply -f mem-1.yaml
pod/memory-demo created
manoj -->
manoj -->kubectl get pod -n memory-example
NAME          READY   STATUS    RESTARTS   AGE
memory-demo   1/1     Running   0          19s
manoj -->
manoj -->kubectl top pod memory-demo -n memory-example
NAME          CPU(cores)   MEMORY(bytes)
memory-demo   66m          150Mi
manoj -->
manoj -->kubectl top pod -n memory-example
NAME          CPU(cores)   MEMORY(bytes)
memory-demo   67m          150Mi
manoj -->
manoj -->
```

POD is running successfully with in the limit of the resources

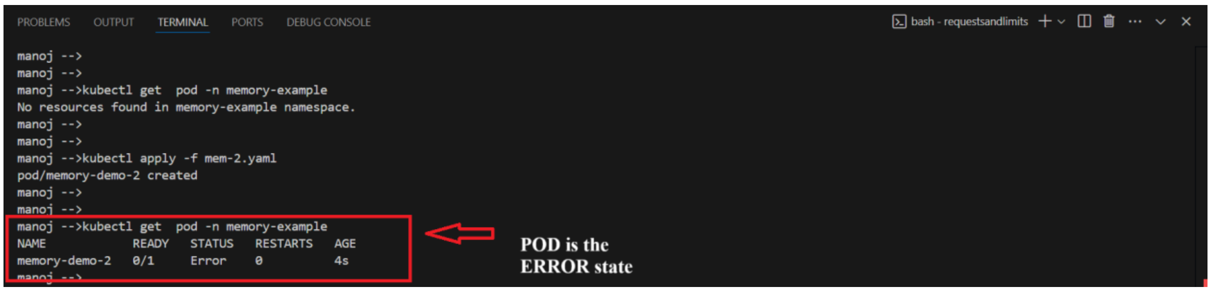
Case 2: Request memory “50Mi” and limit the memory to “100Mi”. Although the container tries to allocate 250 MiB of memory, Kubernetes enforces a 100 MiB memory limit, so the container will likely be terminated due to exceeding the limit.



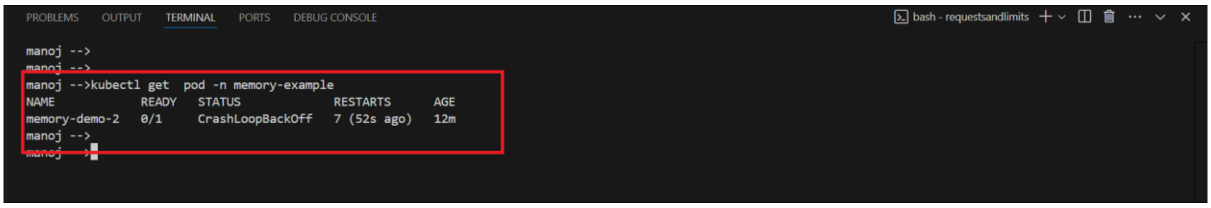
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: memory-demo-2
  namespace: memory-example
spec:
  containers:
  - name: memory-demo-2-ctr
    image: polinux/stress
    resources:
      requests:
        memory: "50Mi"
      limits:
        memory: "100Mi"
    command: ["stress"]
    args: ["--vm", "1", "--vm-bytes", "250M", "--vm-hang", "1"]
```

here the limit is 100 mb

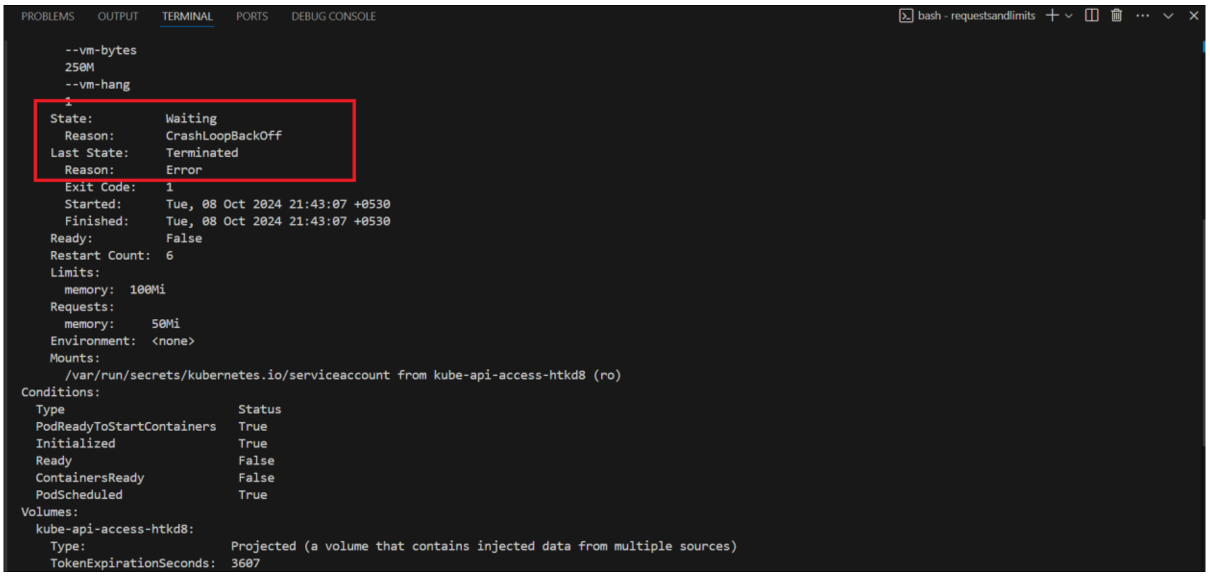
Out of Memory that is we are using more then the memory



```
manoj -->
manoj -->
manoj -->kubectl get  pod -n memory-example
No resources found in memory-example namespace.
manoj -->
manoj -->
manoj -->kubectl apply -f mem-2.yaml
pod/memory-demo-2 created
manoj -->
manoj -->
manoj -->kubectl get  pod -n memory-example
NAME            READY   STATUS    RESTARTS   AGE
memory-demo-2   0/1     Error     0          4s
manoj -->
```

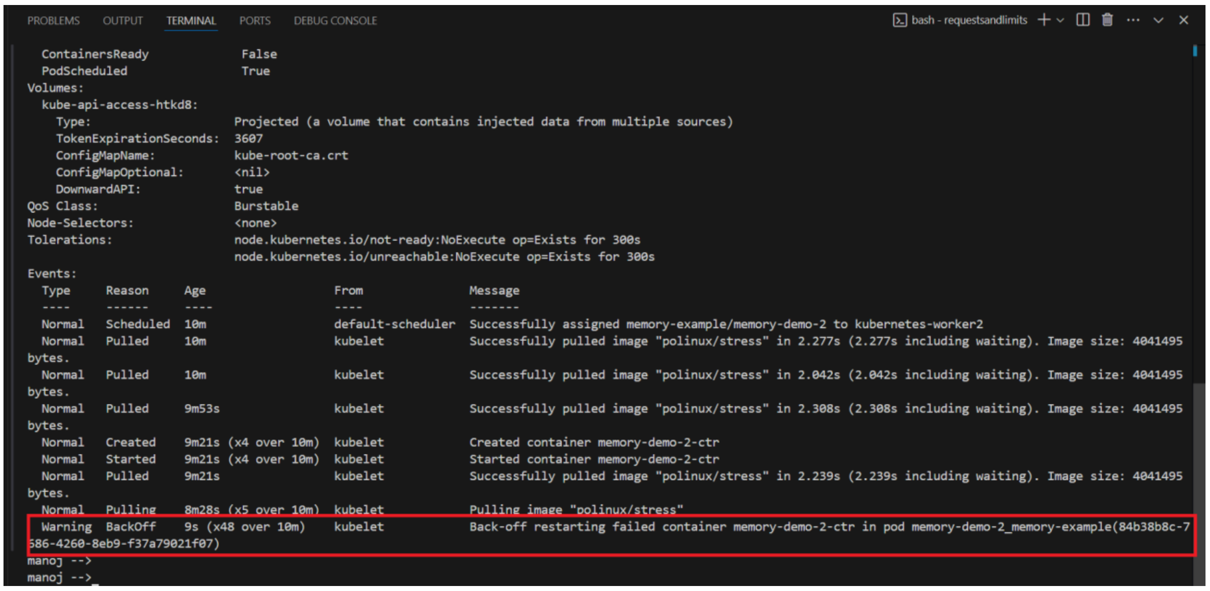


```
manoj -->
manoj -->
manoj -->kubectl get  pod -n memory-example
NAME            READY   STATUS             RESTARTS      AGE
memory-demo-2   0/1     CrashLoopBackOff   7 (52s ago)   12m
manoj -->
manoj -->
```



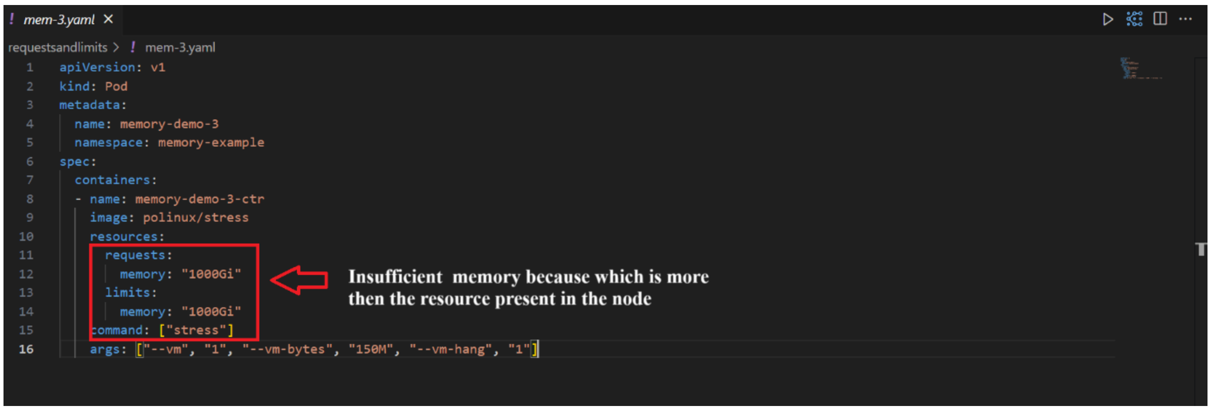
```
      --vm-bytes
      250M
      --vm-hang
      1
    State:          Waiting
      Reason:       CrashLoopBackOff
    Last State:     Terminated
      Reason:       Error
      Exit Code:    1
      Started:      Tue, 08 Oct 2024 21:43:07 +0530
      Finished:     Tue, 08 Oct 2024 21:43:07 +0530
    Ready:          False
    Restart Count:  6
    Limits:
      memory:  100Mi
    Requests:
      memory:     50Mi
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-htkd8 (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       False
  ContainersReady             False
  PodScheduled                True
Volumes:
  kube-api-access-htkd8:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
```

If the container exceeds the **100 MiB memory limit**, it will be terminated by Kubernetes (likely with an OOMKilled error).



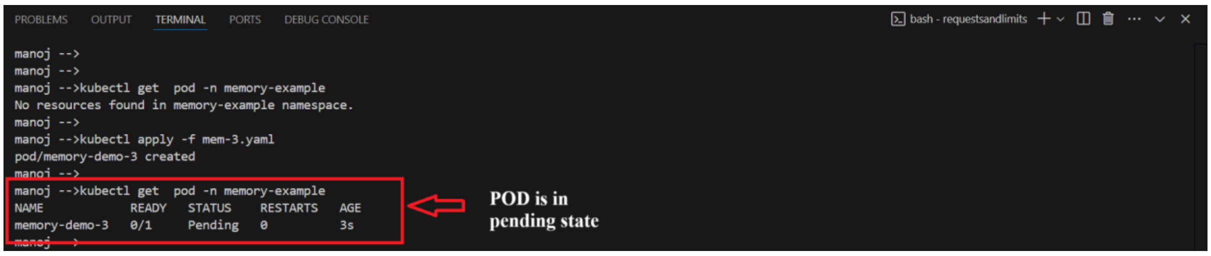
```
  ContainersReady             False
  PodScheduled                True
Volumes:
  kube-api-access-htkd8:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   Burstable
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:
  Type     Reason     Age                  From               Message
  ----     ------     ----                 ----               -------
  Normal   Scheduled  10m                  default-scheduler  Successfully assigned memory-example/memory-demo-2 to kubernetes-worker2
  Normal   Pulled     10m                  kubelet            Successfully pulled image "polinux/stress" in 2.277s (2.277s including waiting). Image size: 4041495
bytes.
  Normal   Pulled     10m                  kubelet            Successfully pulled image "polinux/stress" in 2.042s (2.042s including waiting). Image size: 4041495
bytes.
  Normal   Pulled     9m53s                kubelet            Successfully pulled image "polinux/stress" in 2.308s (2.308s including waiting). Image size: 4041495
bytes.
  Normal   Created    9m21s (x4 over 10m)  kubelet            Created container memory-demo-2-ctr
  Normal   Started    9m21s (x4 over 10m)  kubelet            Started container memory-demo-2-ctr
  Normal   Pulled     9m21s                kubelet            Successfully pulled image "polinux/stress" in 2.239s (2.239s including waiting). Image size: 4041495
bytes.
  Normal   Pulling    8m28s (x5 over 10m)  kubelet            Pulling image "polinux/stress"
  Warning  BackOff    9s (x48 over 10m)    kubelet            Back-off restarting failed container memory-demo-2-ctr in pod memory-demo-2_memory-example(84b38b8c-7
586-4260-8eb9-f37a79021f07)
manoj -->
manoj -->
```

Case 3: Request and Limit resources are specified here are more then the resource[memory] present in node



```yaml
apiVersion: v1
kind: Pod
metadata:
  name: memory-demo-3
  namespace: memory-example
spec:
  containers:
  - name: memory-demo-3-ctr
    image: polinux/stress
    resources:
      requests:
        memory: "1000Gi"
      limits:
        memory: "1000Gi"
    command: ["stress"]
    args: ["--vm", "1", "--vm-bytes", "150M", "--vm-hang", "1"]
```

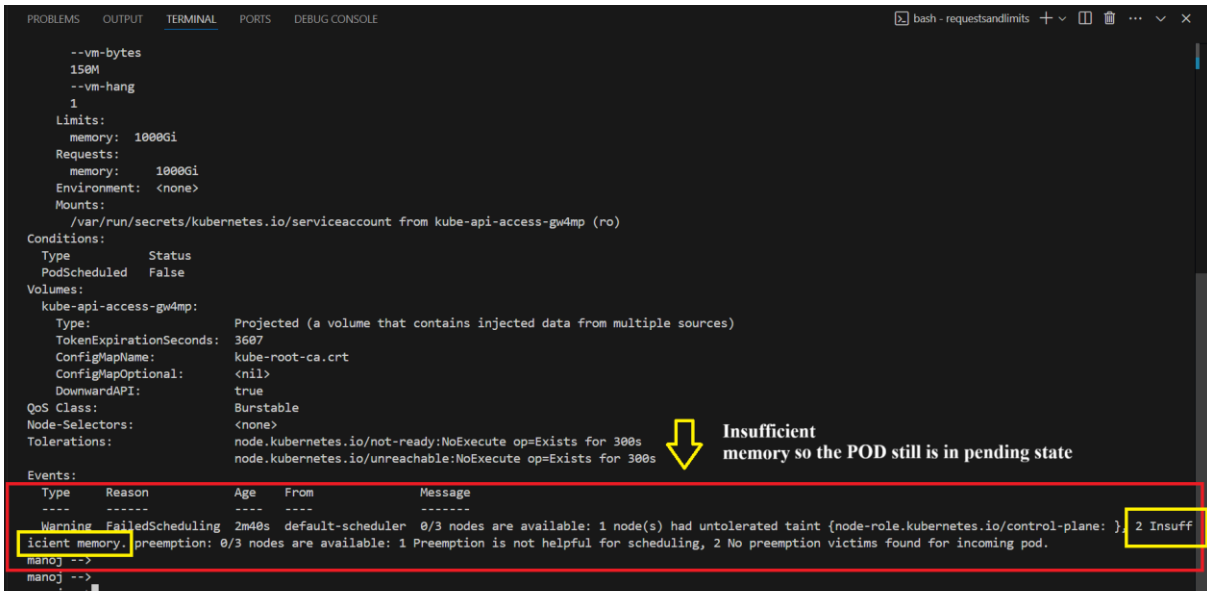
Insufficient memory because which is more then the resource present in the node



```
manoj -->
manoj -->
manoj -->kubectl get  pod -n memory-example
No resources found in memory-example namespace.
manoj -->
manoj -->kubectl apply -f mem-3.yaml
pod/memory-demo-3 created
manoj -->
manoj -->kubectl get  pod -n memory-example
NAME            READY   STATUS    RESTARTS   AGE
memory-demo-3   0/1     Pending   0          3s
manoj -->
```

POD is in pending state

In most clusters, such a pod would likely fail to be scheduled unless a node with very high memory capacity exists.



```
        --vm-bytes
        150M
        --vm-hang
        1
      Limits:
        memory:  1000Gi
      Requests:
        memory:     1000Gi
      Environment:  <none>
      Mounts:
        /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-gw4mp (ro)
Conditions:
  Type           Status
  PodScheduled   False
Volumes:
  kube-api-access-gw4mp:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   Burstable
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason            Age    From               Message
  ----     ------            ----   ----               -------
  Warning  FailedScheduling  2m40s  default-scheduler  0/3 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane: }, 2 Insuff
icient memory. preemption: 0/3 nodes are available: 1 Preemption is not helpful for scheduling, 2 No preemption victims found for incoming pod.
manoj -->
manoj -->
```

Insufficient memory so the POD still is in pending state

Key Points:

- **Requests**: Minimum resources guaranteed for a container.
- **Limits**: Maximum resources a container is allowed to consume.
- If you only set a **request**, Kubernetes will not limit resource usage beyond that.
- If you only set a **limit**, the pod might not be scheduled if the node doesn't have sufficient resources.