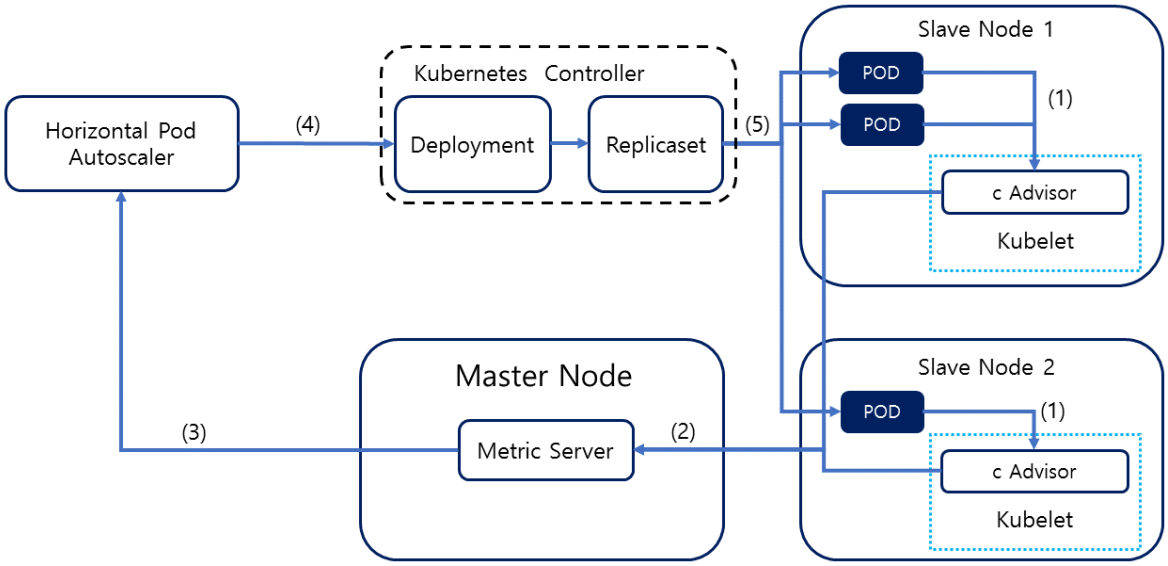
Horizontal Pod Autoscaler 메커니즘



(1) 각 Slave Node에서 c Advisor가 노드 안의 모든 pod의 metrics 정보를 수집한다.

(2) Master Node에서 Metric Server(구 Heapster)에서 노드들의 c Advisor로부터 metrics 정보를 수집한다.

(3) Autoscaler가 Metric Server에게 RestAPI를 요청하여 Metric 정보를 불러온다.

(4) Autoscaler가 불러온 Metric 정보를 기반으로 replica 수를 계산한다. 그 다음 잉여 자원과 필요한 replica 수를 비교해 rescale 가능한지 여부를 파악한다.

이 때 여러 최근 upscale을 한 경우 3분, downscale을 한 경우 5분의 delay가 지난 후 다시 rescale 할 수 있다.

Autoscaler가 수행하는 세부적인 작업 flow는 다음과 같다.

1. reconcileAutoscaler 함수를 통해 Metric Server로부터 현재 값을 가져와 요구되는 replica양을 구한다.
2. 계산된 desired replicas 값을 통해 rescale 여부를 파악한다.
3. normalizeDesiredReplicas 함수를 호출해 최근에 가장 많이 pod 요청을 한 것을 rescale한다.
4. Rescale후 current replica와 desire replica가 일치하는지 비교한다. 일치하지 않으면 다시 rescale한다.

(5) Kubernetes Controller(위 그림에서는 Deployment)가 Autoscaler의 명령에 따라 pod 수를 관리한다.

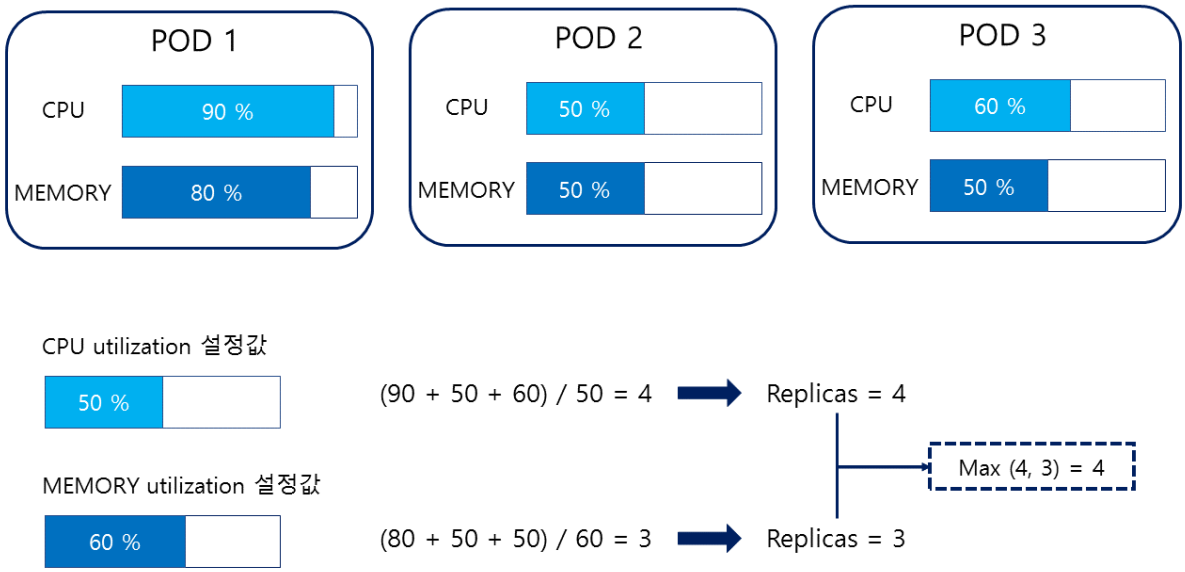
Replicas 계산 알고리즘

Horizontal Pod Autoscaler에서 replicas는 computeReplicasForMetrics Function을 통해서 계산된다.

이 Function은 원하는 metric 값과 현재 metric 값 사이의 비율로 작동하는 알고리즘이다.

`원하는 레플리카 수 = ceil[현재 레플리카 수 * ( 현재 메트릭 값 / 원하는 메트릭 값 )]`
예를 들어 현재 metric 값이 200m이고, 원하는 값이 100m인 경우 (200/100 == 2)가 되어 replica가 2배가 된다.

아래 그림은 여러 metric을 기반으로 계산을 할 때 메커니즘이다.



여러 metric으로 replicas를 계산할 때 우선 각각 metric별로 필요한 replicas의 개수를 구한다.

CPU utilization의 경우 50%를 임계값으로 설정해 이 값을 벗어나면 rescale이 진행된다. 위 경우에서 각각의 pod의 CPU utilization을 보면 90, 50, 60이다. 이를 모두 더하고 설정된 임계값으로 나눠주면 필요한 replicas의 개수가 4로 나온다. 같은 방식으로 Memory도 계산을 해주면 필요한 replicas의 개수가 3이 나온다.

설정한 모든 metric의 계산이 끝나면 결과 값 중 최대값으로 필요한 replicas 수를 설정해준다.