

k8s report

The screenshot of kubectl get nodes

```
● student@ds-student12:~/NTHU-Scheduler-Plugin$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
kind-control-plane	Ready	control-plane	7d5h	v1.29.2
kind-worker	Ready	<none>	7d5h	v1.29.2
kind-worker2	Ready	<none>	7d5h	v1.29.2

The screenshot of passing all the unit tests

```
root@87b2c296fd0b:/go/src/app# go test -v ./...
?      my-scheduler-plugins/cmd/scheduler      [no test files]
=== RUN   TestCustomScheduler_PreFilter
=== RUN   TestCustomScheduler_PreFilter/pod_is_accepted
finish adding2024/05/28 14:36:58 Pod  is in Prefilter phase.
=== RUN   TestCustomScheduler_PreFilter/pod_is_just_accepted
finish adding2024/05/28 14:36:58 Pod  is in Prefilter phase.
=== RUN   TestCustomScheduler_PreFilter/pod_is_rejected
finish adding2024/05/28 14:36:58 Pod  is in Prefilter phase.
--- PASS: TestCustomScheduler_PreFilter (0.00s)
    --- PASS: TestCustomScheduler_PreFilter/pod_is_accepted (0.00s)
    --- PASS: TestCustomScheduler_PreFilter/pod_is_just_accepted (0.00s)
    --- PASS: TestCustomScheduler_PreFilter/pod_is_rejected (0.00s)
=== RUN   TestCustomScheduler_Score
=== RUN   TestCustomScheduler_Score/least_mode
2024/05/28 14:36:58 Custom scheduler runs with the mode: Least.
2024/05/28 14:36:58 Pod  is in Score phase. Calculate the score of Node m1.
2024/05/28 14:36:58 Pod  score is 0.
2024/05/28 14:36:58 Pod  is in Score phase. Calculate the score of Node m2.
2024/05/28 14:36:58 Pod  score is 0.
=== RUN   TestCustomScheduler_Score/most_mode
2024/05/28 14:36:58 Custom scheduler runs with the mode: Least.
2024/05/28 14:36:58 Pod  is in Score phase. Calculate the score of Node m1.
2024/05/28 14:36:58 Pod  score is 100.
2024/05/28 14:36:58 Pod  is in Score phase. Calculate the score of Node m2.
2024/05/28 14:36:58 Pod  score is 200.
--- PASS: TestCustomScheduler_Score (0.00s)
    --- PASS: TestCustomScheduler_Score/least_mode (0.00s)
    --- PASS: TestCustomScheduler_Score/most_mode (0.00s)
=== RUN   TestCustomScheduler_NormalizeScore
=== RUN   TestCustomScheduler_NormalizeScore/scores_in_range
=== RUN   TestCustomScheduler_NormalizeScore/scores_out_of_range
=== RUN   TestCustomScheduler_NormalizeScore/negative_score
--- PASS: TestCustomScheduler_NormalizeScore (0.00s)
    --- PASS: TestCustomScheduler_NormalizeScore/scores_in_range (0.00s)
    --- PASS: TestCustomScheduler_NormalizeScore/scores_out_of_range (0.00s)
    --- PASS: TestCustomScheduler_NormalizeScore/negative_score (0.00s)
PASS
ok      my-scheduler-plugins/pkg/plugins      0.030s
root@87b2c296fd0b:/go/src/app#
```

3 scenarios

scenario 1

3個pod

1. pod nginx:

- podGroup: "A"
- minAvailable: "2"
- requests & limits memory :100Mi

2. pod nginx-2:

- podGroup: "B"
- minAvailable: "2"
- requests & limits memory :100Mi

3. pod nginx-3:

- podGroup: "A"
- minAvailable: "2"
- requests & limits memory :100Mi

這邊主要是檢查prefilter正確性預期是nginx和nginx-2第一次檢查時無法通過prefilter，因為沒有達到minAvailable，nginx-3則是第一次就能通過prefilter被schedule。之後下一round(這邊看是5分鐘後)nginx因為nginx-3的加入滿足minAvailable的條件，會被schedule，而nginx-2還是無法被schedule。

下面是最後每個pod被assign到哪一個node的結果。

```
student@ds-student12:~/NTHU-Scheduler-Plugin$ kubectl get po -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP            NODE           NOMINATED NODE   READINESS GATES
my-scheduler-69cfc986c7-n5q6x      1/1     Running   0           32m   10.244.2.8    kind-worker2   <none>           <none>
nginx                                1/1     Running   0           31m   10.244.2.10   kind-worker2   <none>           <none>
nginx-2                             0/1     Pending   0           31m   <none>        <none>         <none>           <none>
nginx-3                             1/1     Running   0           31m   10.244.2.9    kind-worker2   <none>           <none>
```

scenario 2

3個pod

1. pod nginx:

- podGroup: "A"
- minAvailable: "1"
- requests & limits memory :1000Mi

2. pod nginx-2:

- podGroup: "A"
- minAvailable: "1"
- requests & limits memory :1000Mi

3. pod nginx-3:

- podGroup: "A"
- minAvailable: "1"
- requests & limits memory :500Mi

這邊主要是測試most mode正確性，預期是nginx會被schedule給原先可分配的memory就比較大的kind-worker2，nginx-2會被schedule給目前可分配的memory比較大的kind-worker(kind-worker2被分配1000Mi給nginx後，剩餘可分配的memory小於kind-worker)，最後nginx-3會分配給kind-worker2，因為kind-worker和kind-worker2都各被分配1000Mi給其他pod，此時kind-worker2剩餘可分配的memory比較大。

下面是最後每個pod被assign到哪一個node的結果。

```
student@ds-student12:~/NTHU-Scheduler-Plugin$ kubect1 get po -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED	NODE	READINESS	GATES
my-scheduler-69cfc986c7-n5q6x	1/1	Running	0	51m	10.244.2.8	kind-worker2	<none>		<none>	
nginx	1/1	Running	0	4s	10.244.2.11	kind-worker2	<none>		<none>	
nginx-2	1/1	Running	0	4s	10.244.1.2	kind-worker	<none>		<none>	
nginx-3	1/1	Running	0	4s	10.244.2.12	kind-worker2	<none>		<none>	

```
student@ds-student12:~/NTHU-Scheduler-Plugin$
```

scenario 3

3個pod

1. pod nginx:

- podGroup: "A"
- minAvailable: "1"
- requests & limits memory :500Mi

2. pod nginx-2:

- podGroup: "A"
- minAvailable: "1"
- requests & limits memory :1000Mi

3. pod nginx-3:

- podGroup: "A"
- minAvailable: "1"
- requests & limits memory :500Mi

這邊主要是測試least mode正確性，預期是nginx會被schedule給原先可分配的memory就比較小的kind-worker，此時可分配memory較小的node應該是kind-worker，但nginx-2會被schedule給kind-worker2，因為kind-worker剩餘可分配的記憶體已經不夠分配1000Mi給nginx-2，最後nginx-3會分配給kind-worker，因為目前剩餘可分配的memory是kind-worker比較小。

下面是最後每個pod被assign到哪一個node的結果。

```
student@ds-student12:~/NTHU-Scheduler-Plugin$ kubect1 get po -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED	NODE	READINESS	GATES
my-scheduler-69cfc986c7-j2kxh	1/1	Running	0	5m51s	10.244.2.15	kind-worker2	<none>		<none>	
nginx	1/1	Running	0	3s	10.244.1.5	kind-worker	<none>		<none>	
nginx-2	1/1	Running	0	3s	10.244.2.17	kind-worker2	<none>		<none>	
nginx-3	1/1	Running	0	3s	10.244.1.6	kind-worker	<none>		<none>	

```
student@ds-student12:~/NTHU-Scheduler-Plugin$
```