4.7 การตรวจสอบค่า Metrics ต่างๆ ผ่าน Grafana

Objective: ให้ Student ทำการทดสอบตรจสอบ ค่า Metrics ต่างของ workload ที่ถูกติดตั้งด้วย Grafana

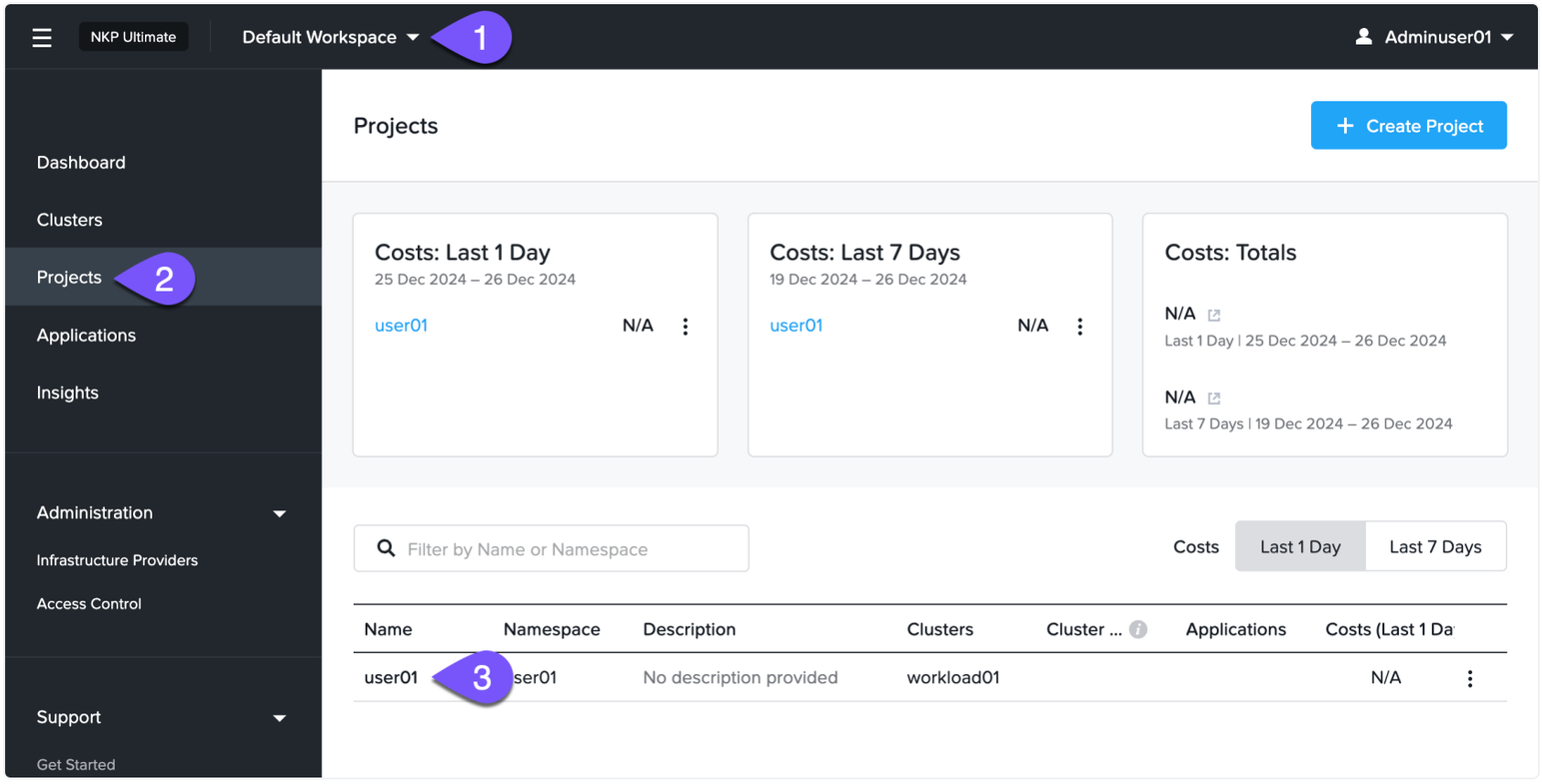
NKP offers centralized observability for Kubernetes environments with 27 built-in Grafana dashboards and support for custom dashboards, enabling comprehensive monitoring and management across multi-cloud and hybrid setups.

```
Did You Know?
Grafana is included only with NKP Pro and Ultimate licensing.
```

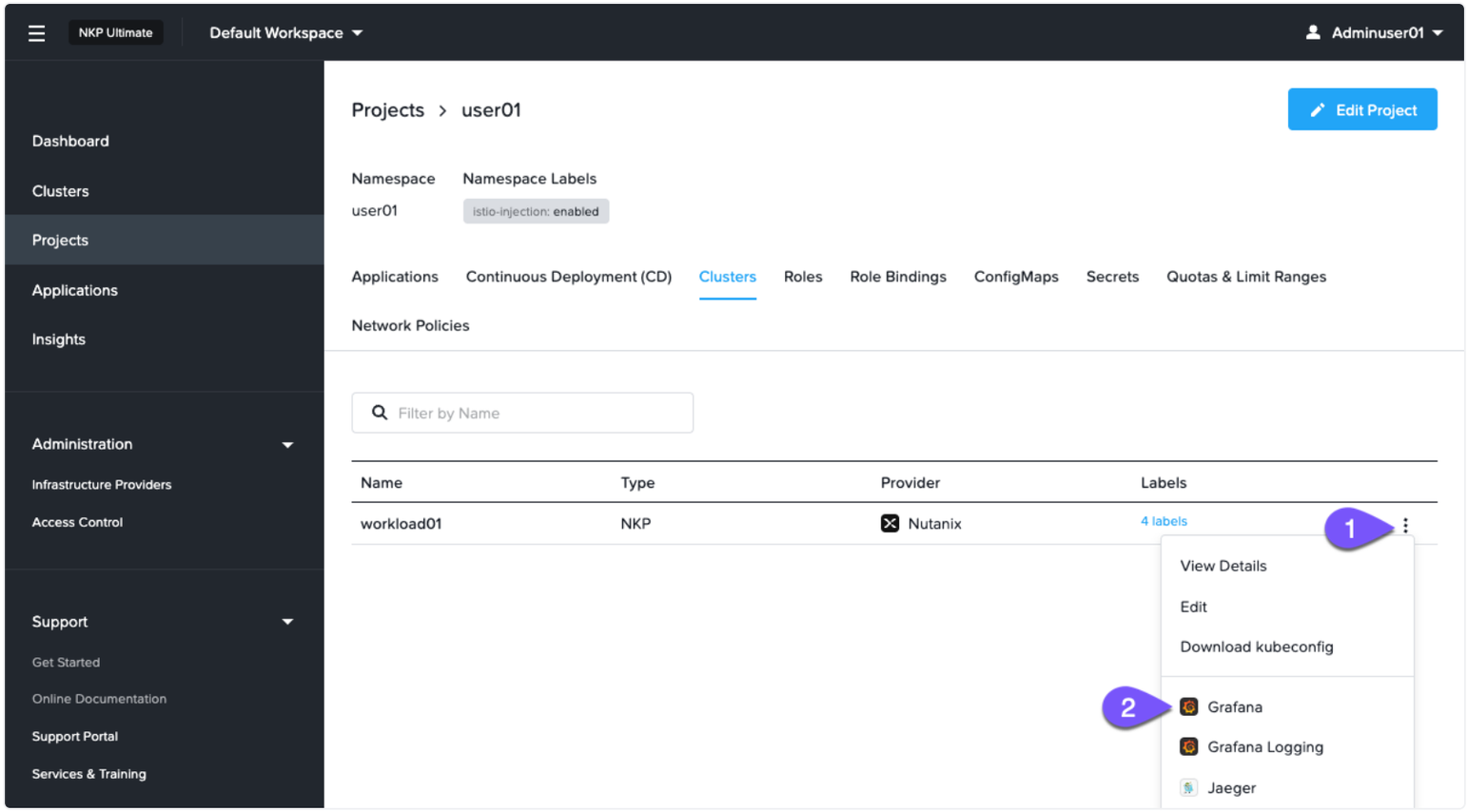
In this exercise, you will explore Grafana dashboards to examine the WordPress application you previously deployed.

Re-Run วิธีการเข้าใช้งาน ด้วยหัวข้อ 4.1

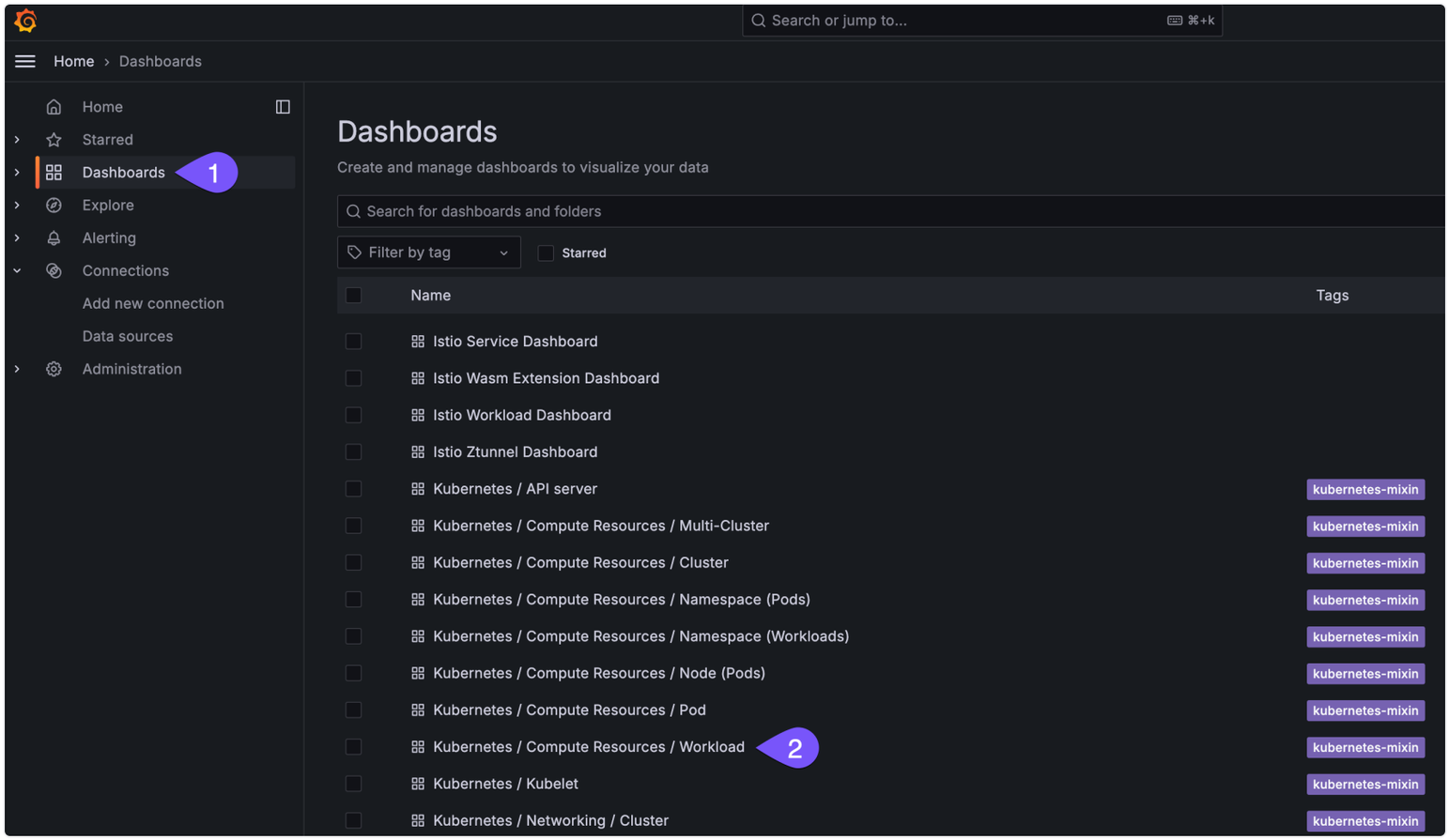
1. Navigate to your specific project where the WordPress application is deployed.



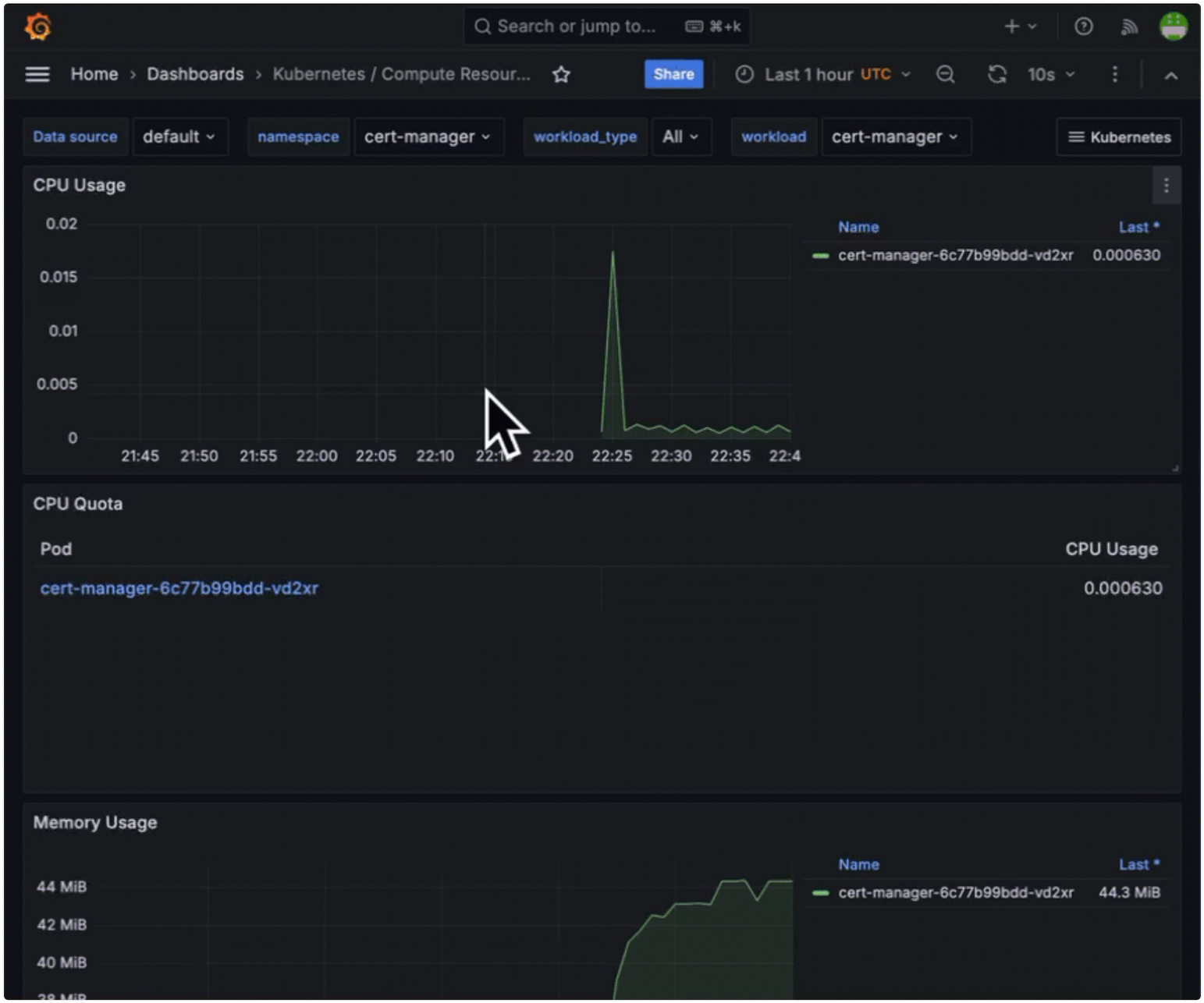
2. Open the Grafana dashboard for the cluster assigned to your project to check monitoring metrics.



3. The default dashboard on the homepage displays resource utilization for your NKP cluster. From there, navigate to the built-in workload dashboard to check compute resources



4. Analyze application-specific metrics by using filters within the namespace field with your project. This will allow you to observe the performance of the WordPress application and its MySQL database by selecting the respective workload.



In summary, Grafana in NKP provides flexibility through custom and community-driven dashboards, enhanced troubleshooting with multi-source data integration, and centralized access control for secure metrics governance. This comprehensive approach enables efficient monitoring and management of Kubernetes environments.

```
Pro tip
There are built-in Grafana dashboards to monitor K8s components in addition to user deployed workloads. The following
dashboard is an example for checking the availability of the Kubernetes API server. You can also easily change the time
interval.
```


Search or jump to...
Home › Dashboards › Kubernetes / API server
1
Add
Share
Last 1 hour UTC
2
10s
Data source
default
instance
All
Kubernetes
Notice
The SLO (service level objective) and other metrics displayed on this dashboard are for informational purposes only.
Availability (30d) > 99.000%
99.999%
ErrorBudget (30d) > 99.000%
errorbudget
99.896%
99.894%
99.892%
99.890%
99.888%
99.886%
07:40 07:45 07:50 07:55 08:00 08:05 08:10 08:15 08:20 08:25 08:30 08:35
Read Availability (30d)
99.999%
Read SLI - Requests
64 req/s
62 req/s
60 req/s
58 req/s
56 req/s
54 req/s
200
301
304
404
500
08:00 08:30
Read SLI - Errors
10000%
8000%
6000%
4000%
2000%
0%
Value
certificates
configmaps
devicetokens
helmreleases
helmrepositories
ingresses
kustomizations
leases
08:00
Read SLI - Duration
1.5 s
1 s
500 ms
0 s
alertmanagerconfigs
alertmanagers
alerts
apiservices
appdeployments
apps
assign
assignimage
assignmetadata
08:00
Write Availability (30d)
99.999%
Write SLI - Requests
42.5 req/s
42 req/s
41.5 req/s
41 req/s
40.5 req/s
200
201
403
404
409
415
500
08:00 08:30
Write SLI - Errors
federatedtypeconfigs
issuers
licenses
08:00 08:30
Write SLI - Duration
500 ms
400 ms
300 ms
200 ms
100 ms
0 s
authrequests
backupstoragelocations
cephclusters
ciliumendpoints
ciliumidentities
clusterpropagatedversion
clusterroles
configmaps
constraintpodstatuses
08:00
Work Queue Add Rate
6 ops/s
4 ops/s
2 ops/s
0 ops/s
07:40 07:50 08:00 08:10 08:20 08:30
Work Queue Depth
100
80
60
40
20
0
07:40 07:50 08:00 08:10 08:20 08:30