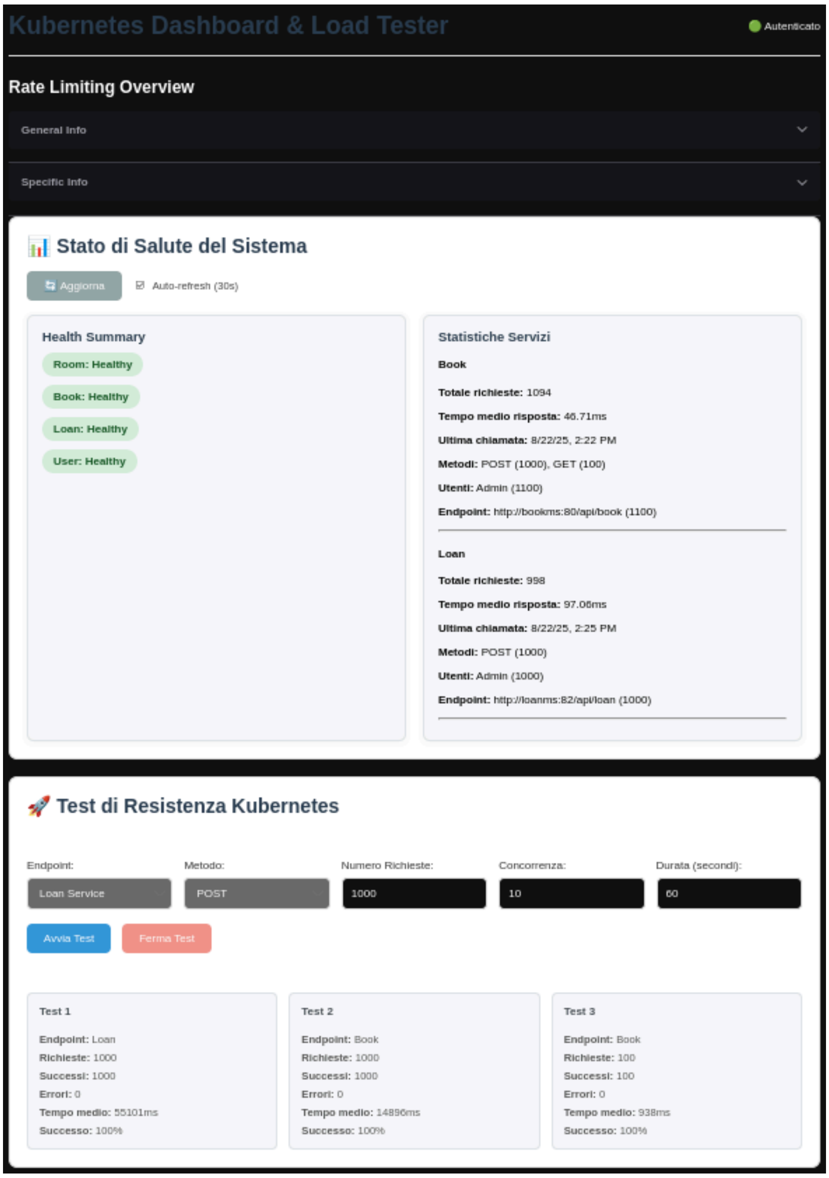
API Gateway Demo

Introduction

From the frontend application (developed specifically to test the resilience of my containers) you can login either as a **regular user** (without the privilege of viewing the health status of the microservices) or as an **administrator**.

The main (and only) page displays the general system status, retrieved from the health API, along with:

- A summary of the requests made by the currently logged-in user
- A testing section where you can choose which API and which microservice to invoke



Book

Totale richieste: 1094

Tempo medio risposta: 46.71ms

Ultima chiamata: 8/22/25, 2:22 PM

Metodi: POST (1000), GET (100)

Utenti: Admin (1100)

Endpoint: http://bookms:80/api/book (1100)

Loan

Totale richieste: 998

Tempo medio risposta: 97.06ms

Ultima chiamata: 8/22/25, 2:25 PM

Metodi: POST (1000)

Utenti: Admin (1000)

Endpoint: http://loanms:82/api/loan (1000)

Endpoint:	Metodo:	Numero Richieste:	Concorrenza:	Durata (secondi):
Loan Service	POST	1000	10	60

Test 1

Endpoint: Loan
Richieste: 1000
Successi: 1000
Errori: 0
Tempo medio: 55101ms
Successo: 100%

Test 2

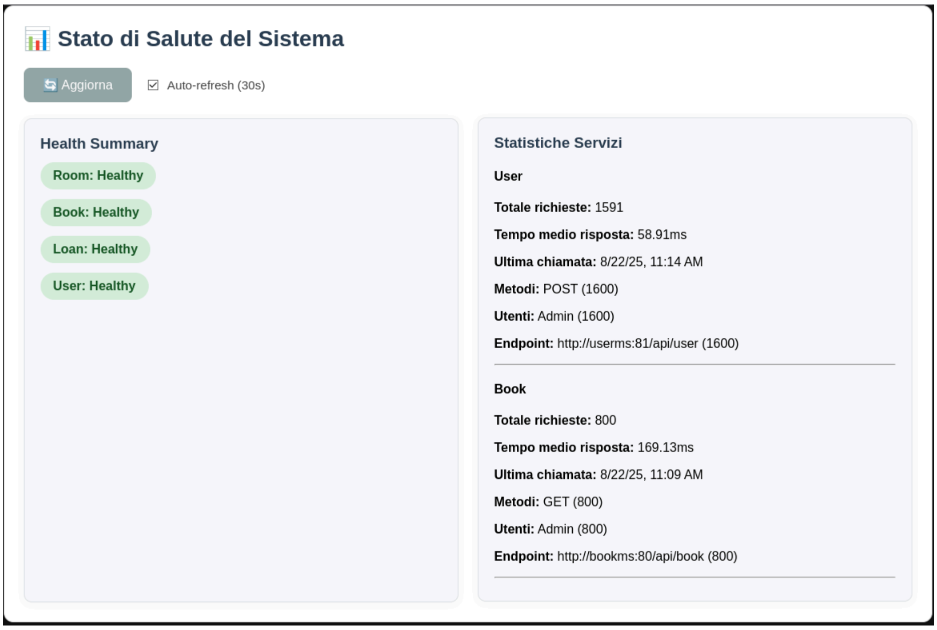
Endpoint: Book
Richieste: 1000
Successi: 1000
Errori: 0
Tempo medio: 14896ms
Successo: 100%

Test 3

Endpoint: Book
Richieste: 100
Successi: 100
Errori: 0
Tempo medio: 938ms
Successo: 100%

Usage example

After sending a high number of requests to the **Book** and **User** microservices:



📊 Stato di Salute del Sistema

☑ Auto-refresh (30s)

Health Summary

Room: Healthy

Book: Healthy

Loan: Healthy

User: Healthy

Statistiche Servizi

User

Totale richieste: 1591

Tempo medio risposta: 58.91ms

Ultima chiamata: 8/22/25, 11:14 AM

Metodi: POST (1600)

Utenti: Admin (1600)

Endpoint: http://userms:81/api/user (1600)

Book

Totale richieste: 800

Tempo medio risposta: 169.13ms

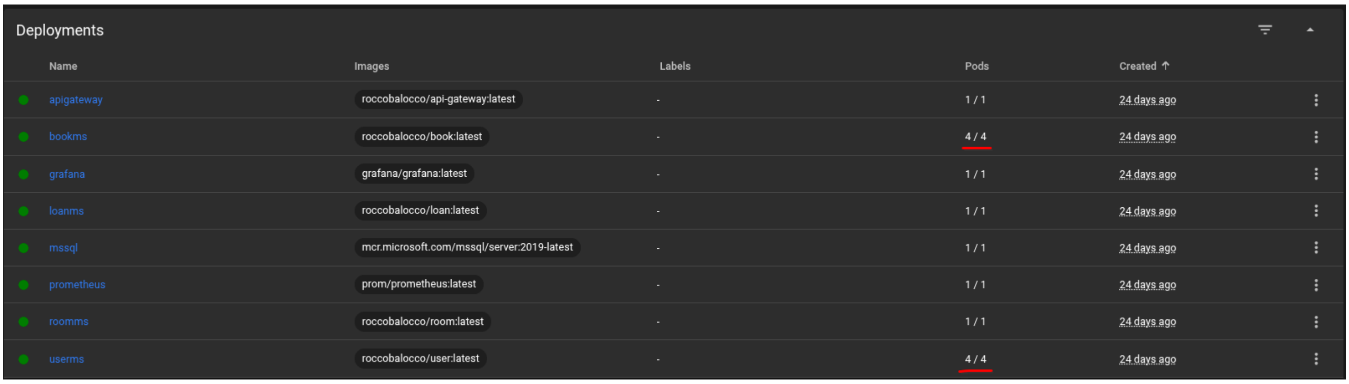
Ultima chiamata: 8/22/25, 11:09 AM

Metodi: GET (800)

Utenti: Admin (800)

Endpoint: http://bookms:80/api/book (800)

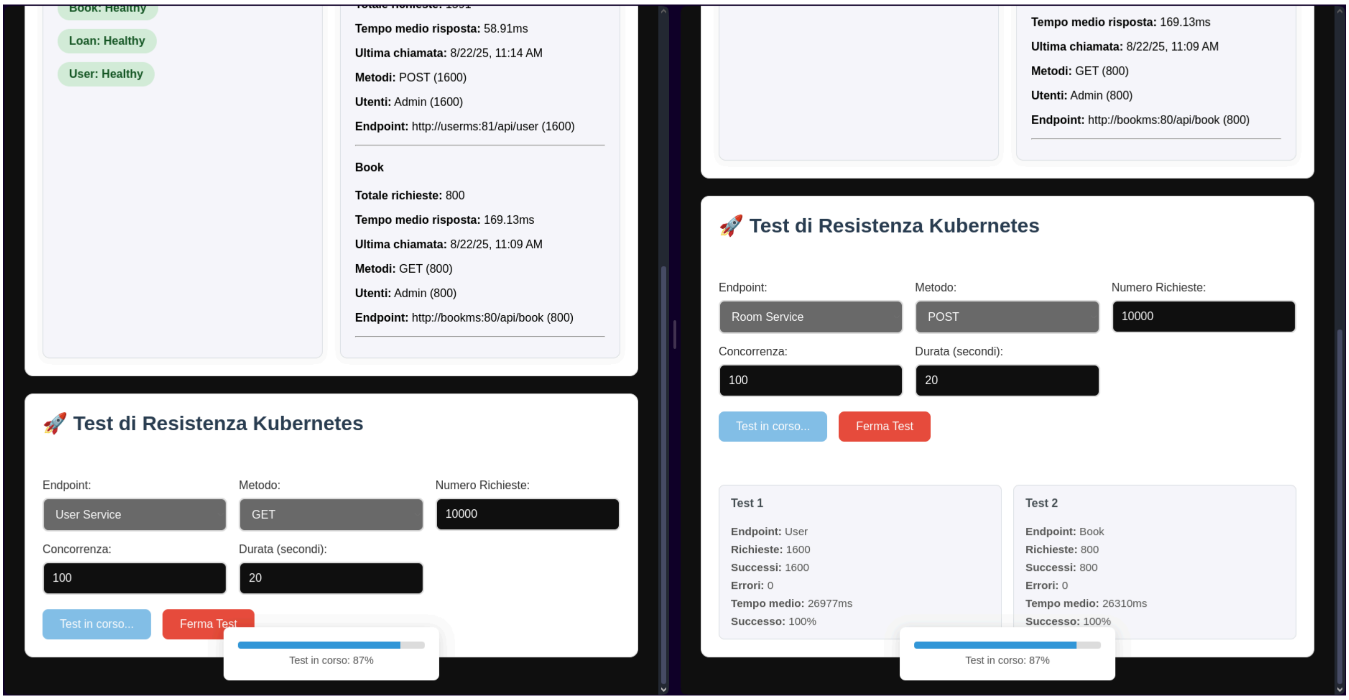
We can observe that the **Horizontal Pod Autoscaler (HPA)** starts scaling the pods accordingly.



Deployments

	Name	Images	Labels	Pods	Created ↑	
●	apigateway	roccobalocco/api-gateway:latest	-	1 / 1	24 days ago	⋮
●	bookms	roccobalocco/book:latest	-	4 / 4	24 days ago	⋮
●	grafana	grafana/grafana:latest	-	1 / 1	24 days ago	⋮
●	loanms	roccobalocco/loan:latest	-	1 / 1	24 days ago	⋮
●	mssql	mcr.microsoft.com/mssql/server:2019-latest	-	1 / 1	24 days ago	⋮
●	prometheus	prom/prometheus:latest	-	1 / 1	24 days ago	⋮
●	roomms	roccobalocco/room:latest	-	1 / 1	24 days ago	⋮
●	userms	roccobalocco/user:latest	-	4 / 4	24 days ago	⋮

To stress-test the deployment, I sent **20,000 requests**, split between two different microservices.



Book: Healthy

Loan: Healthy

User: Healthy

Totale richieste: 1591

Tempo medio risposta: 58.91ms

Ultima chiamata: 8/22/25, 11:14 AM

Metodi: POST (1600)

Utenti: Admin (1600)

Endpoint: http://userms:81/api/user (1600)

Book

Totale richieste: 800

Tempo medio risposta: 169.13ms

Ultima chiamata: 8/22/25, 11:09 AM

Metodi: GET (800)

Utenti: Admin (800)

Endpoint: http://bookms:80/api/book (800)

🚀 Test di Resistenza Kubernetes

Endpoint: User Service

Metodo: GET

Numero Richieste: 10000

Concorrenza: 100

Durata (secondi): 20

Test in corso: 87%

Tempo medio risposta: 169.13ms

Ultima chiamata: 8/22/25, 11:09 AM

Metodi: GET (800)

Utenti: Admin (800)

Endpoint: http://bookms:80/api/book (800)

🚀 Test di Resistenza Kubernetes

Endpoint: Room Service

Metodo: POST

Numero Richieste: 10000

Concorrenza: 100

Durata (secondi): 20

Test 1

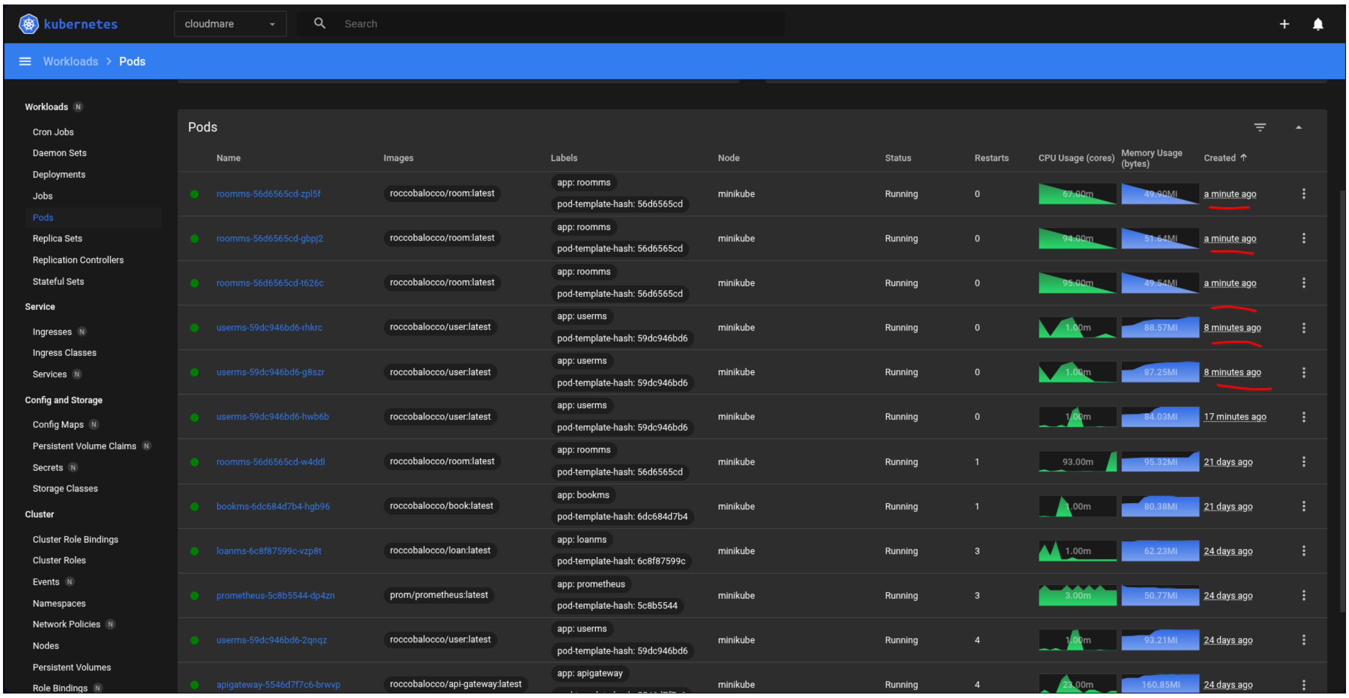
Endpoint: User
Richieste: 1600
Successi: 1600
Errori: 0
Tempo medio: 26977ms
Successo: 100%

Test 2

Endpoint: Book
Richieste: 800
Successi: 800
Errori: 0
Tempo medio: 26310ms
Successo: 100%

Test in corso: 87%

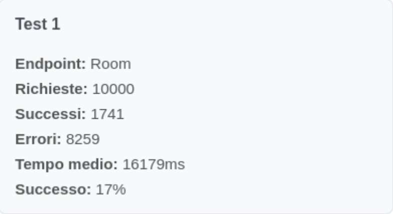
As a result, many pods crashed due to workload intensity and existing policy configurations.



Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
roomms-56d6565cd-zpl5f	roccobalocco/room:latest	app: roomms pod-template-hash: 56d6565cd	minikube	Running	0	67.00m	49.90Mi	a minute ago
roomms-56d6565cd-gbpj2	roccobalocco/room:latest	app: roomms pod-template-hash: 56d6565cd	minikube	Running	0	94.00m	51.64Mi	a minute ago
roomms-56d6565cd-t626c	roccobalocco/room:latest	app: roomms pod-template-hash: 56d6565cd	minikube	Running	0	95.00m	49.54Mi	a minute ago
userms-59dc946bd6-rhkrc	roccobalocco/user:latest	app: userms pod-template-hash: 59dc946bd6	minikube	Running	0	1.00m	88.57Mi	8 minutes ago
userms-59dc946bd6-g8szr	roccobalocco/user:latest	app: userms pod-template-hash: 59dc946bd6	minikube	Running	0	1.00m	87.25Mi	8 minutes ago
userms-59dc946bd6-hwb6b	roccobalocco/user:latest	app: userms pod-template-hash: 59dc946bd6	minikube	Running	0	1.00m	84.03Mi	17 minutes ago
roomms-56d6565cd-w4ddl	roccobalocco/room:latest	app: roomms pod-template-hash: 56d6565cd	minikube	Running	1	93.00m	95.32Mi	21 days ago
bookms-6dc684d7b4-hgb96	roccobalocco/book:latest	app: bookms pod-template-hash: 6dc684d7b4	minikube	Running	1	1.00m	80.38Mi	21 days ago
loanms-6c8f87599c-vzp8t	roccobalocco/loan:latest	app: loanms pod-template-hash: 6c8f87599c	minikube	Running	3	1.00m	62.23Mi	24 days ago
prometheus-5c8b5544-dp4zn	prom/prometheus:latest	app: prometheus pod-template-hash: 5c8b5544	minikube	Running	3	3.00m	50.77Mi	24 days ago
userms-59dc946bd6-2qnqz	roccobalocco/user:latest	app: userms pod-template-hash: 59dc946bd6	minikube	Running	4	1.00m	93.21Mi	24 days ago
apigateway-5546d7f7c6-brwvp	roccobalocco/api-gateway:latest	app: apigateway	minikube	Running	4	23.00m	160.85Mi	24 days ago

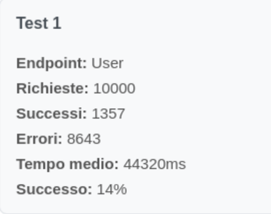
- The test on the **Room** service achieved a success rate of **17%**.



Test 1

Endpoint: Room
Richieste: 10000
Successi: 1741
Errori: 8259
Tempo medio: 16179ms
Successo: 17%

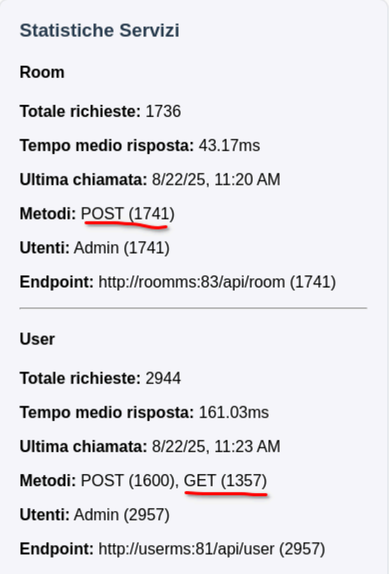
- The test on the **User** service achieved a success rate of **14%**.



Test 1

Endpoint: User
Richieste: 10000
Successi: 1357
Errori: 8643
Tempo medio: 44320ms
Successo: 14%

Although 10,000 requests were sent to each microservice, only a small number were processed successfully. The remaining requests were blocked due to circuit-breaking policies, pod crashes, or other constraints.



Statistiche Servizi

Room

Totale richieste: 1736

Tempo medio risposta: 43.17ms

Ultima chiamata: 8/22/25, 11:20 AM

Metodi: POST (1741)

Utenti: Admin (1741)

Endpoint: http://roomms:83/api/room (1741)

User

Totale richieste: 2944

Tempo medio risposta: 161.03ms

Ultima chiamata: 8/22/25, 11:23 AM

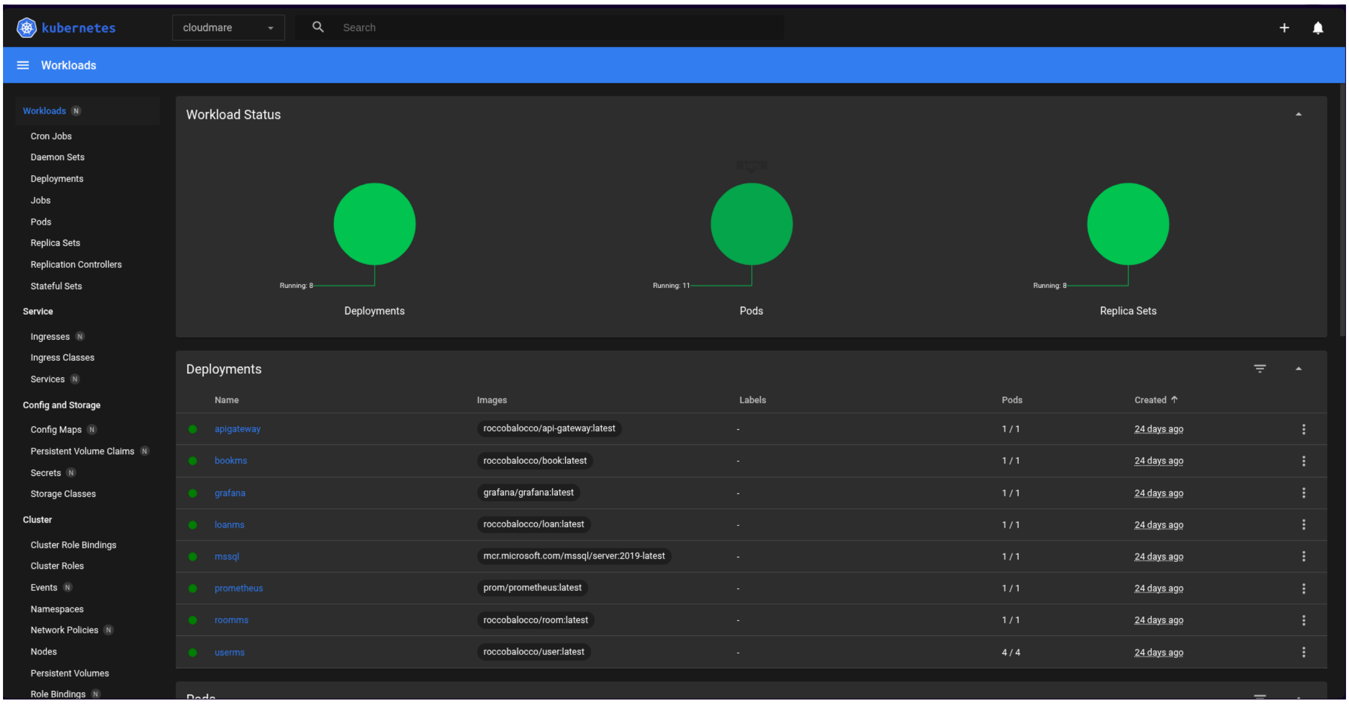
Metodi: POST (1600), GET (1357)

Utenti: Admin (2957)

Endpoint: http://userms:81/api/user (2957)

After this test, the system adjusted dynamically:

- Since the latest requests targeted the **User** microservice, the HPA scaled up its number of pods.
- At the same time, the number of pods for the **Room** microservice was scaled down.



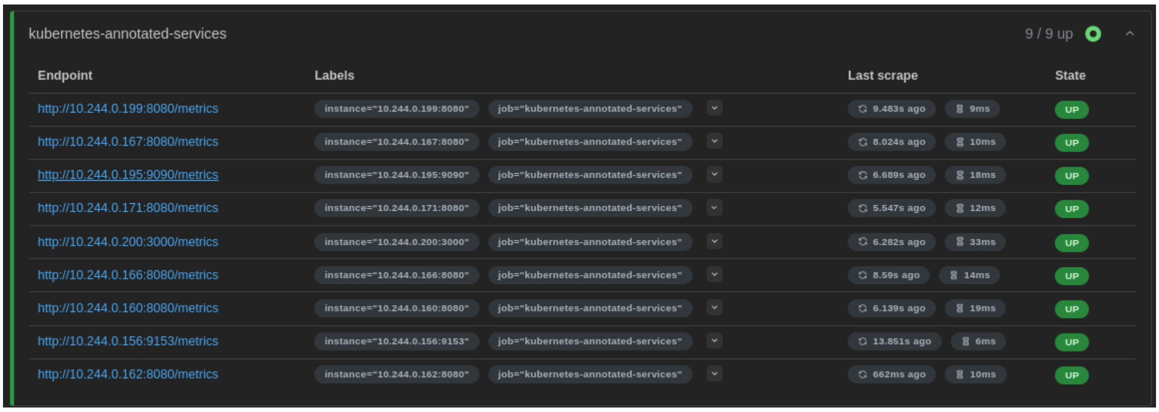
Deployments

	Name	Images	Labels	Pods	Created ↑
●	apigateway	roccobalocco/api-gateway:latest	-	1 / 1	24 days ago
●	bookms	roccobalocco/book:latest	-	1 / 1	24 days ago
●	grafana	grafana/grafana:latest	-	1 / 1	24 days ago
●	loanms	roccobalocco/loan:latest	-	1 / 1	24 days ago
●	mssql	mcr.microsoft.com/mssql/server:2019-latest	-	1 / 1	24 days ago
●	prometheus	prom/prometheus:latest	-	1 / 1	24 days ago
●	roomms	roccobalocco/room:latest	-	1 / 1	24 days ago
●	userms	roccobalocco/user:latest	-	4 / 4	24 days ago

To trigger the **Per-IP** and **Global Fixed** policies in the Angular web application, you need to send many requests, as they are not sent simultaneously.

Prometheus

Thanks to the annotations included in the manifest files, **Prometheus** can scrape the metrics of the active pods.

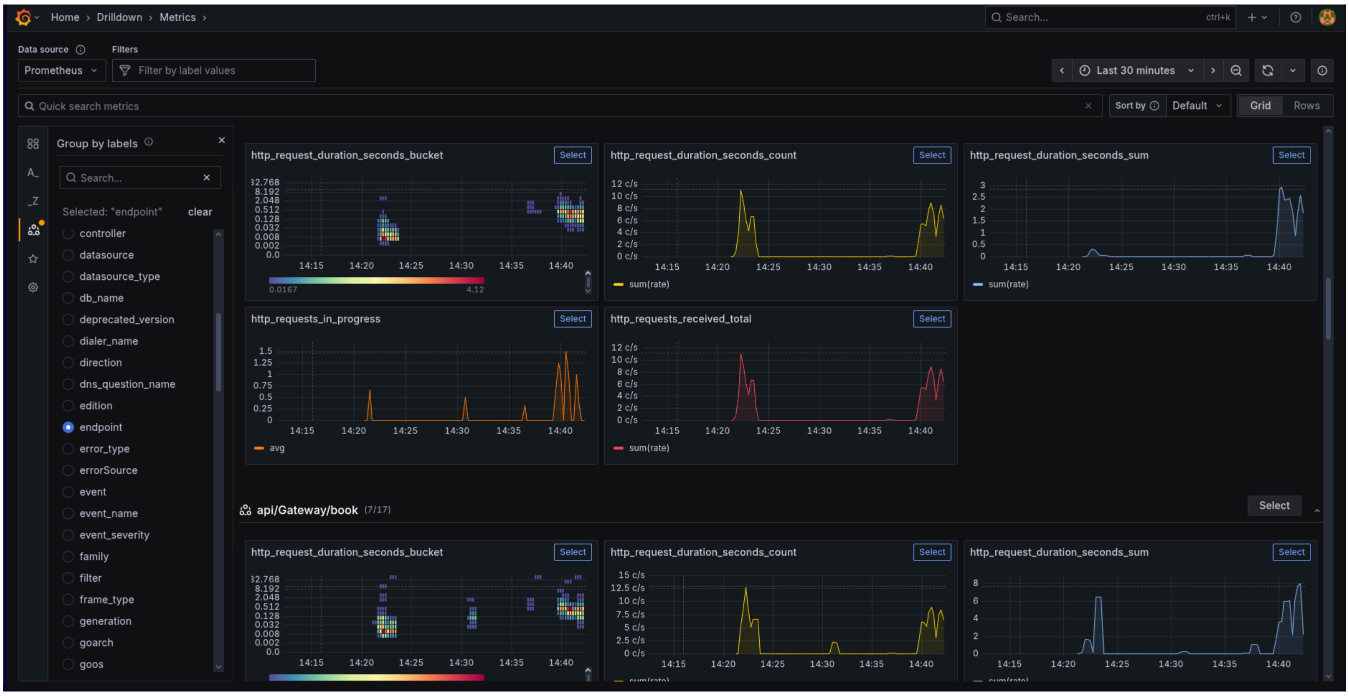


kubernetes-annotated-services 9 / 9 up

Endpoint	Labels		Last scrape		State
http://10.244.0.199:8080/metrics	instance="10.244.0.199:8080"	job="kubernetes-annotated-services"	9.483s ago	9ms	UP
http://10.244.0.167:8080/metrics	instance="10.244.0.167:8080"	job="kubernetes-annotated-services"	8.024s ago	10ms	UP
http://10.244.0.195:9090/metrics	instance="10.244.0.195:9090"	job="kubernetes-annotated-services"	6.689s ago	18ms	UP
http://10.244.0.171:8080/metrics	instance="10.244.0.171:8080"	job="kubernetes-annotated-services"	5.547s ago	12ms	UP
http://10.244.0.200:3000/metrics	instance="10.244.0.200:3000"	job="kubernetes-annotated-services"	6.282s ago	33ms	UP
http://10.244.0.166:8080/metrics	instance="10.244.0.166:8080"	job="kubernetes-annotated-services"	8.59s ago	14ms	UP
http://10.244.0.160:8080/metrics	instance="10.244.0.160:8080"	job="kubernetes-annotated-services"	6.139s ago	19ms	UP
http://10.244.0.156:9153/metrics	instance="10.244.0.156:9153"	job="kubernetes-annotated-services"	13.851s ago	6ms	UP
http://10.244.0.162:8080/metrics	instance="10.244.0.162:8080"	job="kubernetes-annotated-services"	662ms ago	10ms	UP

Grafana

Grafana connects to Prometheus as a data source and visualizes various performance and health metrics of the system.



Home › Drilldown › Metrics ›

Data source: Prometheus

Filters: Filter by label values

Last 30 minutes

Sort by: Default

Grid | Rows

Group by labels

Selected: "endpoint"