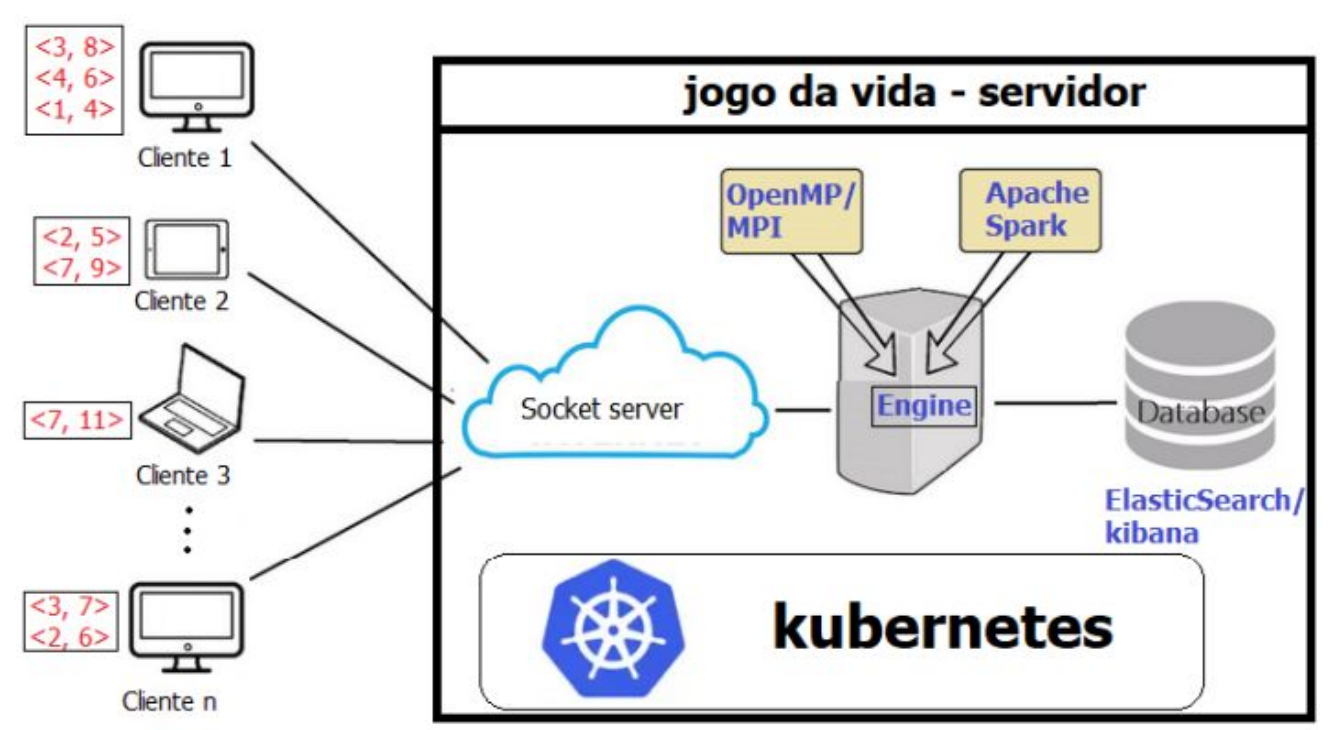
Projeto Final - PSPD

Gabriela da Gama Pivetta - 180052845
Murilo Gomes de Souza - 180025601

Arquitetura



Sequencial

Logs

Ok, RESULTADO CORRETO tam=8;
tempos: init=0.0000141, comp=0.0000451, fim=0.0000198, tot=0.0000789
Ok, RESULTADO CORRETO tam=16;
tempos: init=0.0000050, comp=0.0004020, fim=0.0000041, tot=0.0004110
Ok, RESULTADO CORRETO tam=32;
tempos: init=0.0000110, comp=0.0035961, fim=0.0000172, tot=0.0036242
Ok, RESULTADO CORRETO tam=64;
tempos: init=0.0000479, comp=0.0297630, fim=0.0000410, tot=0.0298519
Ok, RESULTADO CORRETO tam=128;
tempos: init=0.0001879, comp=0.1201141, fim=0.0000439, tot=0.1203458
Ok, RESULTADO CORRETO tam=256;
tempos: init=0.0003171, comp=0.8231528, fim=0.0001521, tot=0.8236220
Ok, RESULTADO CORRETO tam=512;
tempos: init=0.0012391, comp=6.2695191, fim=0.0005629, tot=6.2713211
Ok, RESULTADO CORRETO tam=1024;
tempos: init=0.0049760, comp=51.7422149, fim=0.0022430, tot=51.7494340

OpenMP

Logs

Ok, RESULTADO CORRETO tam=8;
tempos: init=0.0000451, comp=0.0000858, fim=0.0000551, tot=0.0001860
Ok, RESULTADO CORRETO tam=16;
tempos: init=0.0000110, comp=0.0007579, fim=0.0000110, tot=0.0007799
Ok, RESULTADO CORRETO tam=32;
tempos: init=0.0000191, comp=0.0036299, fim=0.0000031, tot=0.0036521
Ok, RESULTADO CORRETO tam=64;
tempos: init=0.0000150, comp=0.0099549, fim=0.0000100, tot=0.0099800
Ok, RESULTADO CORRETO tam=128;
tempos: init=0.0001199, comp=0.0808120, fim=0.0000322, tot=0.0809641
Ok, RESULTADO CORRETO tam=256;
tempos: init=0.0003560, comp=0.6609581, fim=0.0001211, tot=0.6614351
Ok, RESULTADO CORRETO tam=512;
tempos: init=0.0011411, comp=5.3329060, fim=0.0006120, tot=5.3346591
Ok, RESULTADO CORRETO tam=1024;
tempos: init=0.0046771, comp=43.7467699, fim=0.0018570, tot=43.7533040

OpenMP e MPI

Sem hosts

Logs

Ok, RESULTADO CORRETO
tam=8; tempos: init=0.0000050, comp=0.4391639, fim=0.0039990, tot=0.4431679
Ok, RESULTADO CORRETO
tam=16; tempos: init=0.0000050, comp=1.2719800, fim=0.0000520, tot=1.2720370
Ok, RESULTADO CORRETO
tam=32; tempos: init=0.0000319, comp=2.6392059, fim=0.0000122, tot=2.6392500
Ok, RESULTADO CORRETO
tam=64; tempos: init=0.0000720, comp=5.8830879, fim=0.0001140, tot=5.8832738
Ok, RESULTADO CORRETO
tam=128; tempos: init=0.0000870, comp=11.9198620, fim=0.0000968, tot=11.9200459
Ok, RESULTADO CORRETO
tam=256; tempos: init=0.0003691, comp=24.4154499, fim=0.0001731, tot=24.4159920
Ok, RESULTADO CORRETO
tam=512; tempos: init=0.0012319, comp=45.4470401, fim=0.0004740, tot=45.4487460
Ok, RESULTADO CORRETO
tam=1024; tempos: init=0.0128441, comp=51.8523459, fim=0.0019290, tot=51.8671191

OpenMP e MPI

Com hosts

Logs

Ok, RESULTADO CORRETO
tam=8; tempos: init=0.0000088, comp=0.0010130, fim=0.0002611, tot=0.0012829
Ok, RESULTADO CORRETO
tam=16; tempos: init=0.0000129, comp=0.0006340, fim=0.0001440, tot=0.0007908
Ok, RESULTADO CORRETO
tam=32; tempos: init=0.0000200, comp=0.0016940, fim=0.0001290, tot=0.0018430
Ok, RESULTADO CORRETO
tam=64; tempos: init=0.0000451, comp=0.0045581, fim=0.0001109, tot=0.0047140
Ok, RESULTADO CORRETO
tam=128; tempos: init=0.0000761, comp=0.0154891, fim=0.0000660, tot=0.0156312
Ok, RESULTADO CORRETO
tam=256; tempos: init=0.0003111, comp=0.1179428, fim=0.0001571, tot=0.1184111
Ok, RESULTADO CORRETO
tam=512; tempos: init=0.0012279, comp=0.9280231, fim=0.0004671, tot=0.9297180
Ok, RESULTADO CORRETO
tam=1024; tempos: init=0.0054169, comp=7.4560680, fim=0.0019190, tot=7.4634039

Spark

Comparação

Tabela 5.1: Tabela de referência para a sigla dos algoritmos

Sigla	Algoritmo
Seq	Sequencial em C
OMP	Usando apenas OpenMP
OMP/MPI 1h	Usando OpenMP + MPI em apenas 1 host
OMP/MPI 4h	Usando OpenMP + MPI em 4 hosts diferentes
Spark	Usando PySpark

Tabela 5.2: Tabela de tempo de execução (s) para cada algoritmo

Tamanho	Seq	OMP	OMP/MPI 1h	OMP/MPI 4h	Spark
8	0.00007	0.00018	0.44316	0.00128	8.2838
16	0.00041	0.00077	1.27203	0.00079	8.3759
32	0.00362	0.00365	2.63925	0.00184	18.70081
64	0.02985	0.00998	5.88327	0.00471	40.72215
128	0.12034	0.08096	11.92	0.01563	86.22166
256	0.82362	0.66143	24.416	0.11841	234.15412
512	6.27132	5.33465	45.44874	0.92971	954.47695
1024	51.74943	43.7533	51.86711	7.4634	5257.69753

Tabela 5.3: Caption

Logs

tam=8 **Ok, RESULTADO CORRETO**
time=8.283813238143921
tam=16 **Ok, RESULTADO CORRETO**
time=8.375973463058472
tam=32 **Ok, RESULTADO CORRETO**
time=18.700819969177246
tam=64 **Ok, RESULTADO CORRETO**
time=40.72215127944946
tam=128 **Ok, RESULTADO CORRETO**
time=86.22166776657104
tam=256 **Ok, RESULTADO CORRETO**
time=234.15412831306458
tam=512 **Ok, RESULTADO CORRETO**
time=954.4769561290741
tam=1024 **Ok, RESULTADO CORRETO**
time=5257.697530269623

Socket Server

Script

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

#define PORT 8080
#define MAX_CONNECTIONS 5

int main() {
    int server_fd, new_socket, valread;
    struct sockaddr_in address;
    int addrlen = sizeof(address);
    char buffer[1024] = {0};
    char *hello = "Hello from Socket Server";

    // Criar o socket
    if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) == 0) {
        perror("Socket creation failed");
        exit(EXIT_FAILURE);
    }

    // Configurar as propriedades do socket
    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons(PORT);

    // Vincular o socket à porta e endereço
    if (bind(server_fd, (struct sockaddr *)&address, sizeof(address)) < 0) {
        perror("Bind failed");
        exit(EXIT_FAILURE);
    }

    // Esperar por conexões
    if (listen(server_fd, MAX_CONNECTIONS) < 0) {
        perror("Listen failed");
        exit(EXIT_FAILURE);
    }

    printf("Socket Server listening on port %d\n", PORT);

    // Aceitar conexões e lidar com os clientes
    while (1) {
        if ((new_socket = accept(server_fd, (struct sockaddr *)&address,
(socklen_t*)&addrlen)) < 0) {
            perror("Accept failed");
            exit(EXIT_FAILURE);
        }

        valread = read(new_socket, buffer, 1024);
        printf("Client message: %s\n", buffer);
        send(new_socket, hello, strlen(hello), 0);
        printf("Hello message sent to the client\n");

        close(new_socket);
    }
    return 0;
}
```

Dockerfile

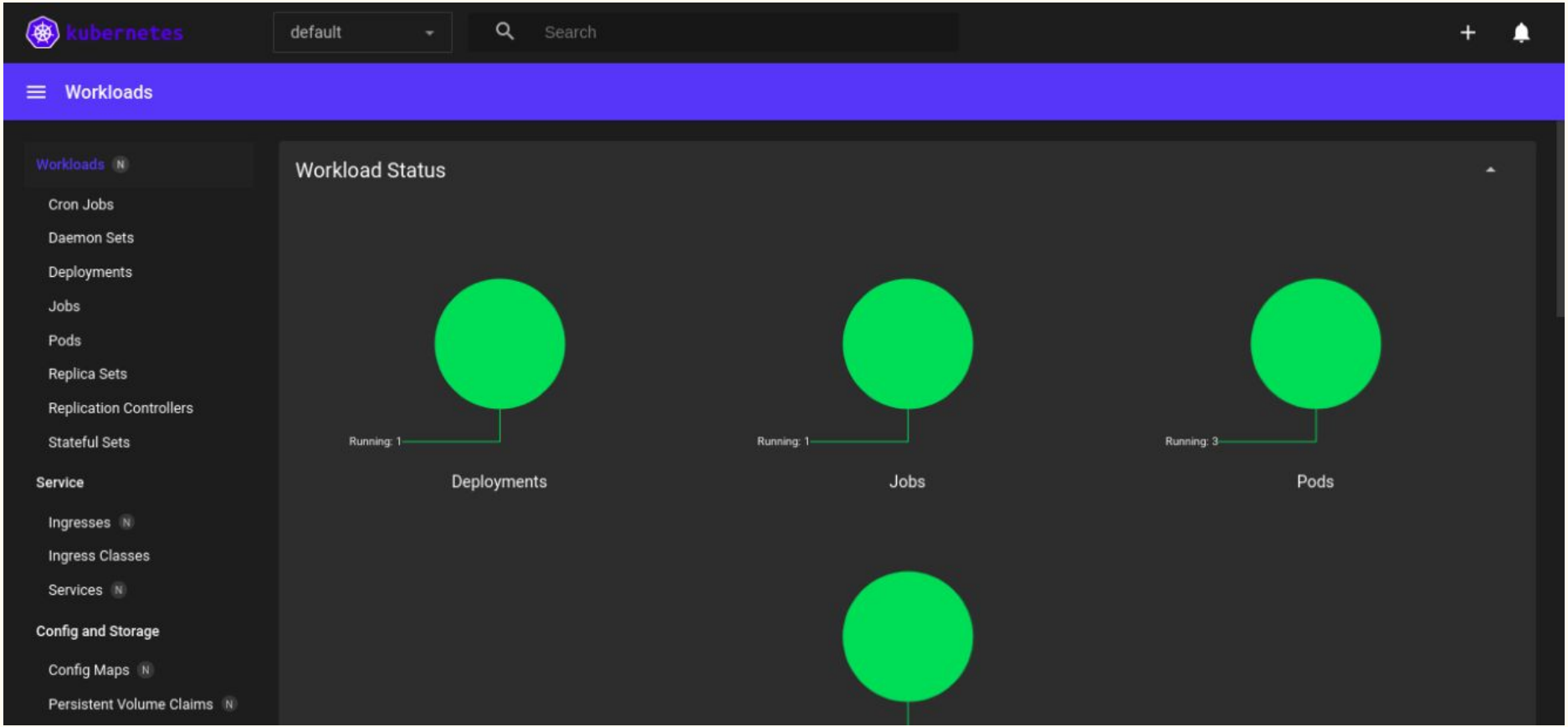
```
FROM gcc:latest

WORKDIR /app

# Copiar o código fonte do Socket Server para o contêiner

COPY socket_server.c /app/

# Compilar o código fonte

RUN gcc -o socket_server socket_server.c

# Definir o comando de inicialização do contêiner (executar o
Socket Server)

CMD ["./socket_server"]
```

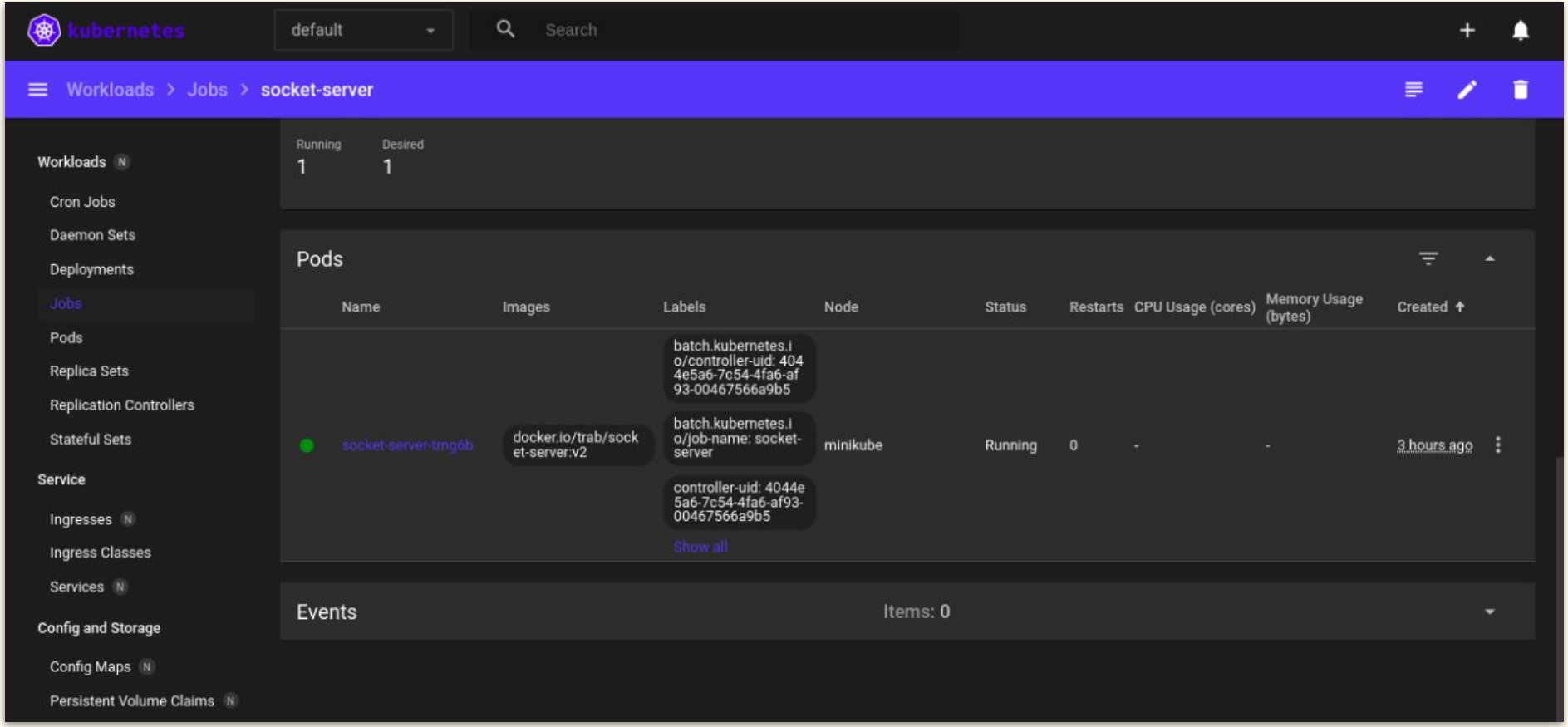
YAML com Job

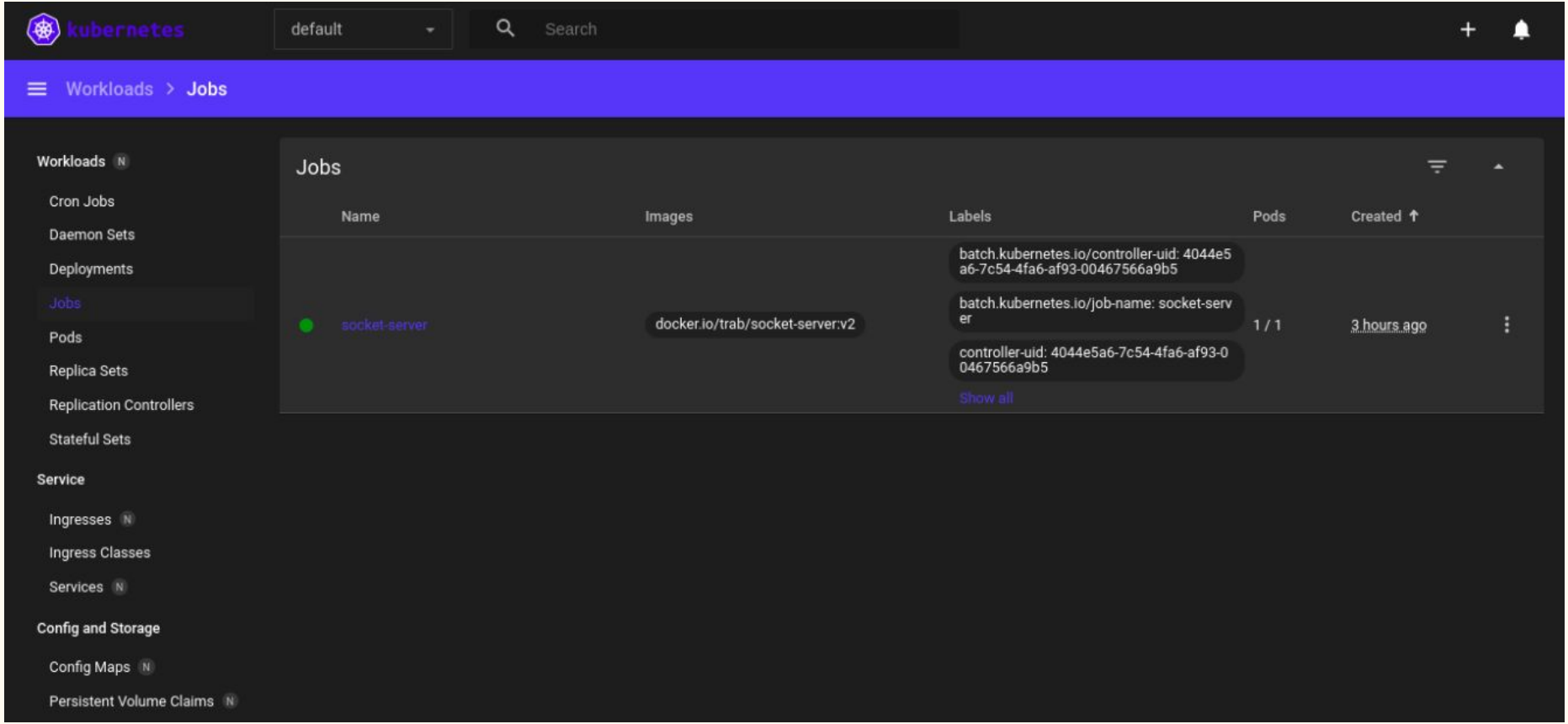
```
apiVersion: batch/v1
kind: Job
metadata:
    name: socket-server
spec:
    template:
    metadata:
        name: socket-server-pod
    spec:
        containers:
        - name: socket-server
          image: docker.io/trab/socket-server:latest
          imagePullPolicy: Never
          ports:
          - containerPort: 8080
        restartPolicy: OnFailure
```

Kubernetes

Com Minikube


kubernetes
default
Search
Workloads
Workloads N
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses N
Ingress Classes
Services N
Config and Storage
Config Maps N
Persistent Volume Claims N
Workload Status
Running: 1
Deployments
Running: 1
Jobs
Running: 3
Pods


kubernetes
default
Search
Workloads > Jobs > socket-server
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
Config Maps
Persistent Volume Claims
Running
1
Desired
1
Pods
Name
Images
Labels
Node
Status
Restarts
CPU Usage (cores)
Memory Usage (bytes)
Created
socket-server-trng6b
docker.io/trab/socket-server:v2
batch.kubernetes.io/controller-uid: 4044e5a6-7c54-4fa6-af93-00467566a9b5
batch.kubernetes.io/job-name: socket-server
controller-uid: 4044e5a6-7c54-4fa6-af93-00467566a9b5
Show all
minikube
Running
0
-
-
3 hours ago
Events
Items: 0


kubernetes
default
Search
Workloads > Jobs
Workloads N
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses N
Ingress Classes
Services N
Config and Storage
Config Maps N
Persistent Volume Claims N
Jobs
Name
Images
Labels
Pods
Created
socket-server
docker.io/trab/socket-server:v2
batch.kubernetes.io/controller-uid: 4044e5a6-7c54-4fa6-af93-00467566a9b5
batch.kubernetes.io/job-name: socket-server
controller-uid: 4044e5a6-7c54-4fa6-af93-00467566a9b5
Show all
1 / 1
3 hours ago

Fim
Obrigado