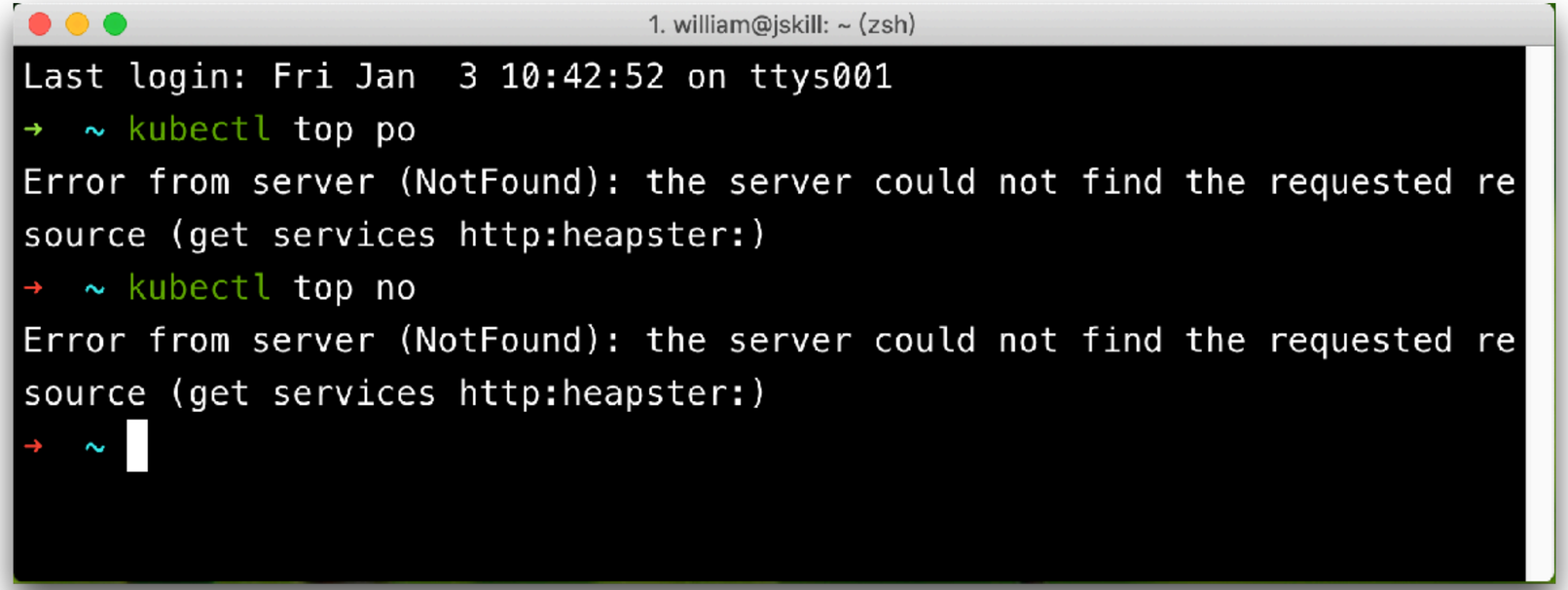
第 4 节

Java/Spring应用在K8s中的内存配置

本课内容

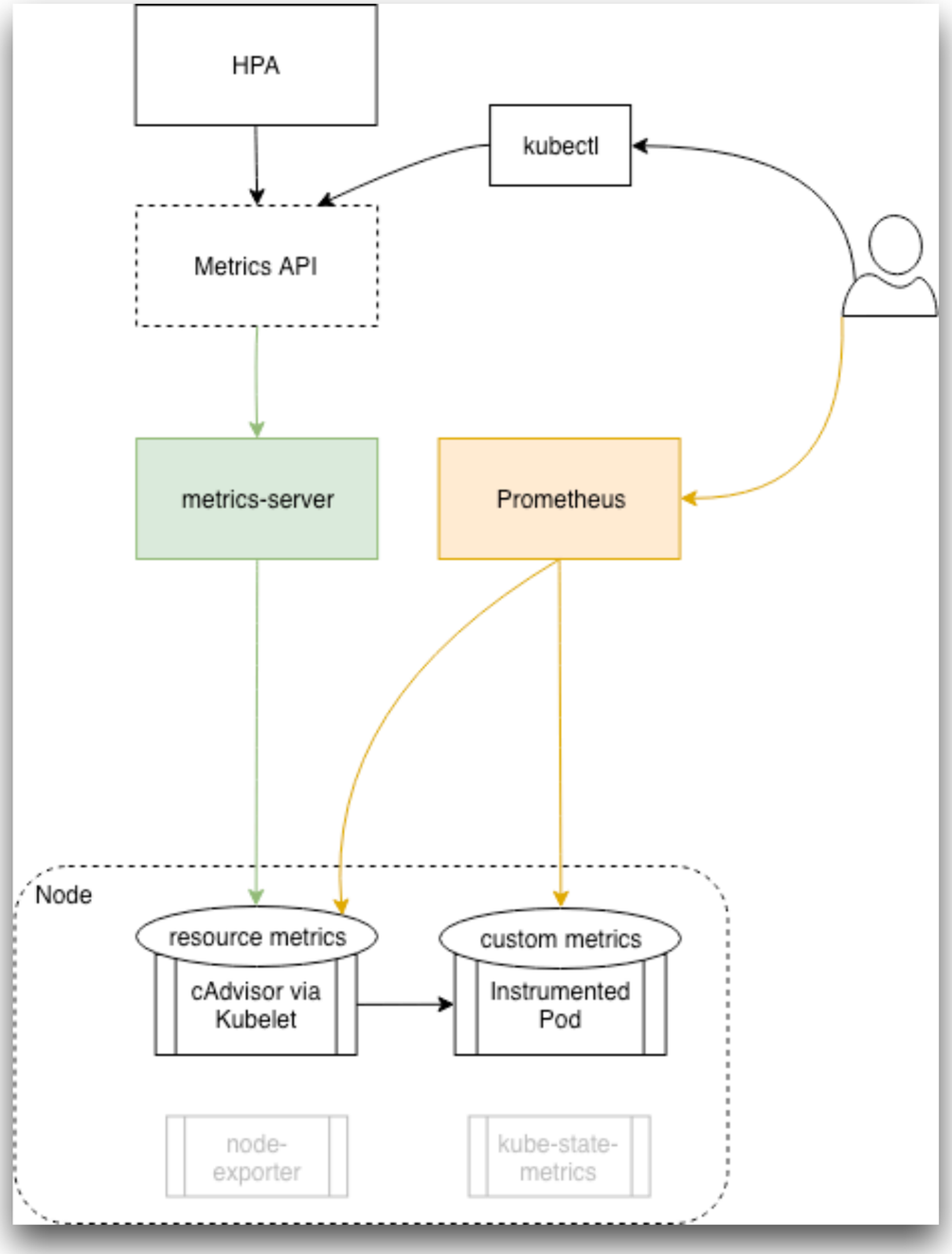
- 启用K8s的Metrics Server组件
- JVM内存结构
- Java/Spring应用在K8s中内存设置实践

默认未启用Metrics Server



metrics-server

https://github.com/kubernetes-sigs/metrics-server

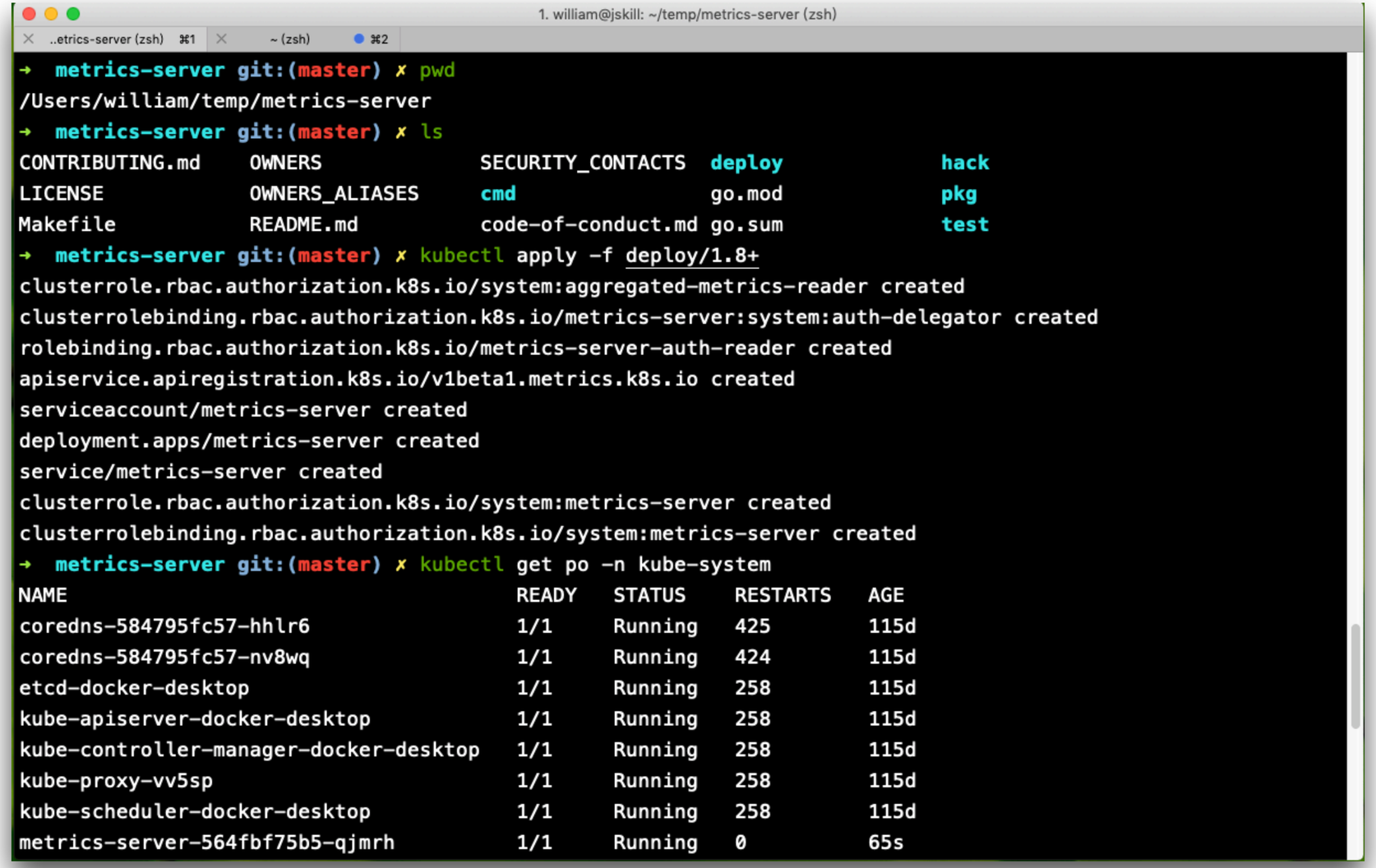


更新metrics-server-deployment.yml

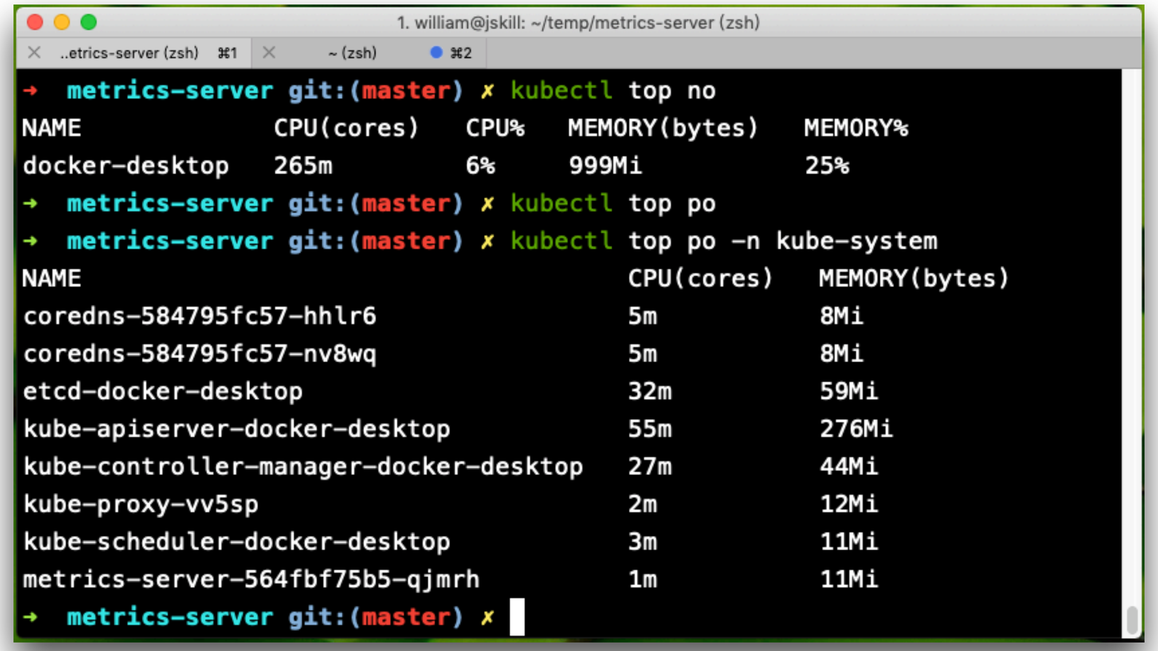
```yaml
      containers:
        - name: metrics-server
          image: k8s.gcr.io/metrics-server-amd64:v0.3.6
          args:
            - --cert-dir=/tmp
            - --secure-port=4443
          ports:
            - name: main-port
              containerPort: 4443
              protocol: TCP
          securityContext:
            readOnlyRootFilesystem: true
            runAsNonRoot: true
            runAsUser: 1000
          imagePullPolicy: Always
          command:
            - /metrics-server
            - --kubelet-insecure-tls
          volumeMounts:
            - name: tmp-dir
              mountPath: /tmp
```

deploy/1.8+/metrics-server-deployment.yml

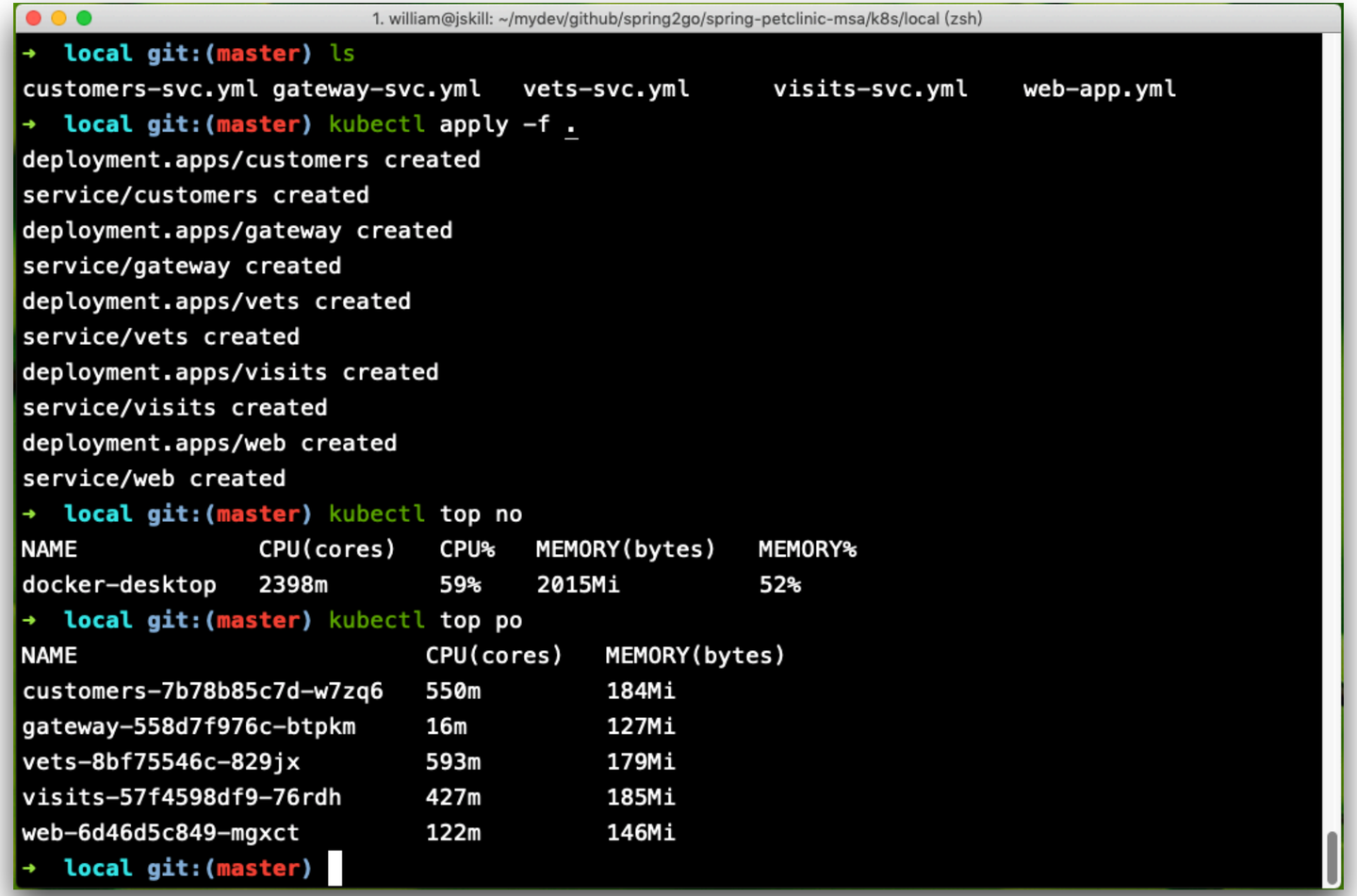
发布metrics-server



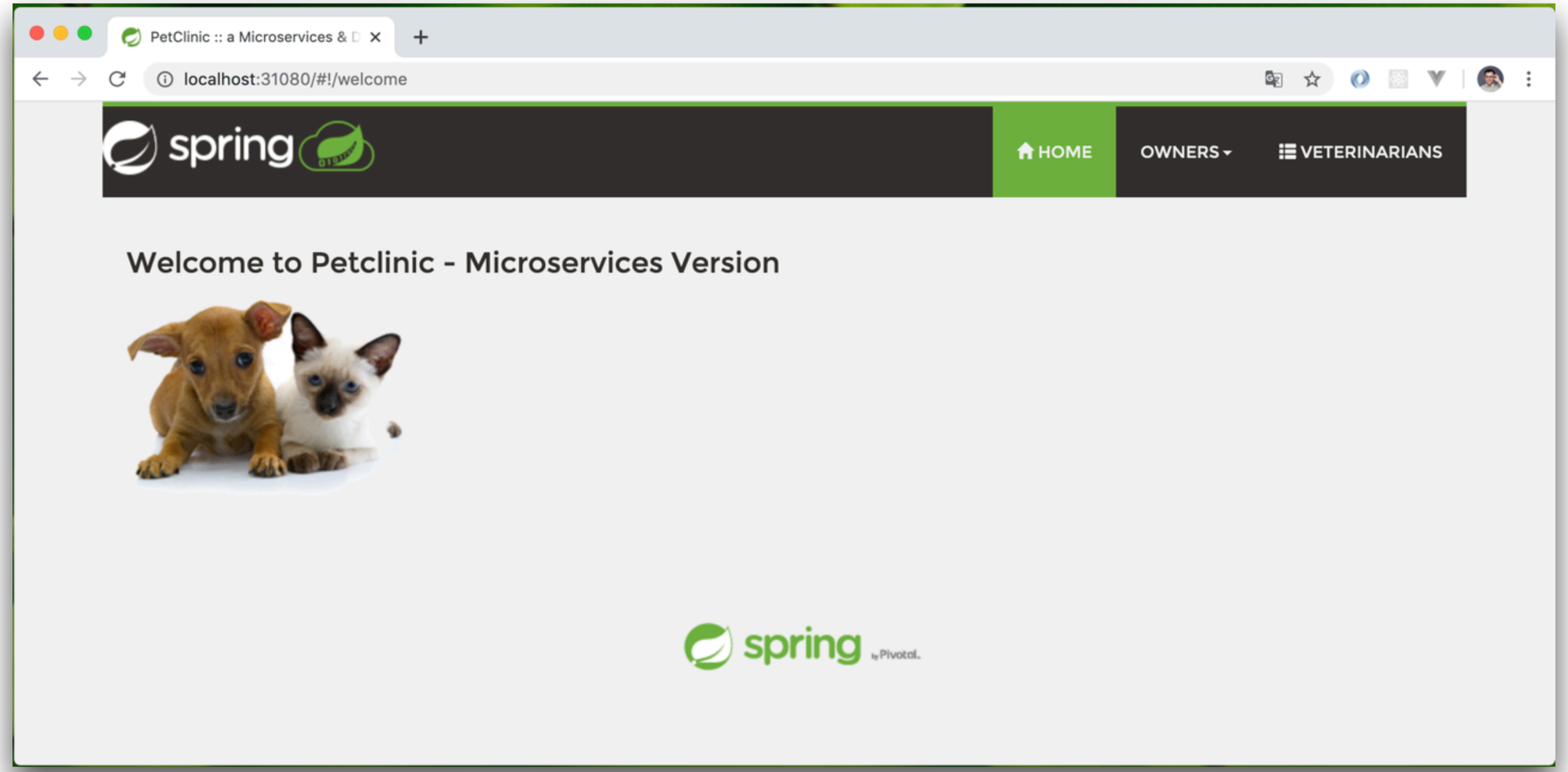
kubectl top


1. william@jskill: ~/temp/metrics-server (zsh)
..etrics-server (zsh) ⌘1
~ (zsh) ⌘2
→ metrics-server git:(master) ✗ kubectl top no
NAME CPU(cores) CPU% MEMORY(bytes) MEMORY%
docker-desktop 265m 6% 999Mi 25%
→ metrics-server git:(master) ✗ kubectl top po
→ metrics-server git:(master) ✗ kubectl top po -n kube-system
NAME CPU(cores) MEMORY(bytes)
coredns-584795fc57-hhlr6 5m 8Mi
coredns-584795fc57-nv8wq 5m 8Mi
etcd-docker-desktop 32m 59Mi
kube-apiserver-docker-desktop 55m 276Mi
kube-controller-manager-docker-desktop 27m 44Mi
kube-proxy-vv5sp 2m 12Mi
kube-scheduler-docker-desktop 3m 11Mi
metrics-server-564fbf75b5-qjmrh 1m 11Mi
→ metrics-server git:(master) ✗

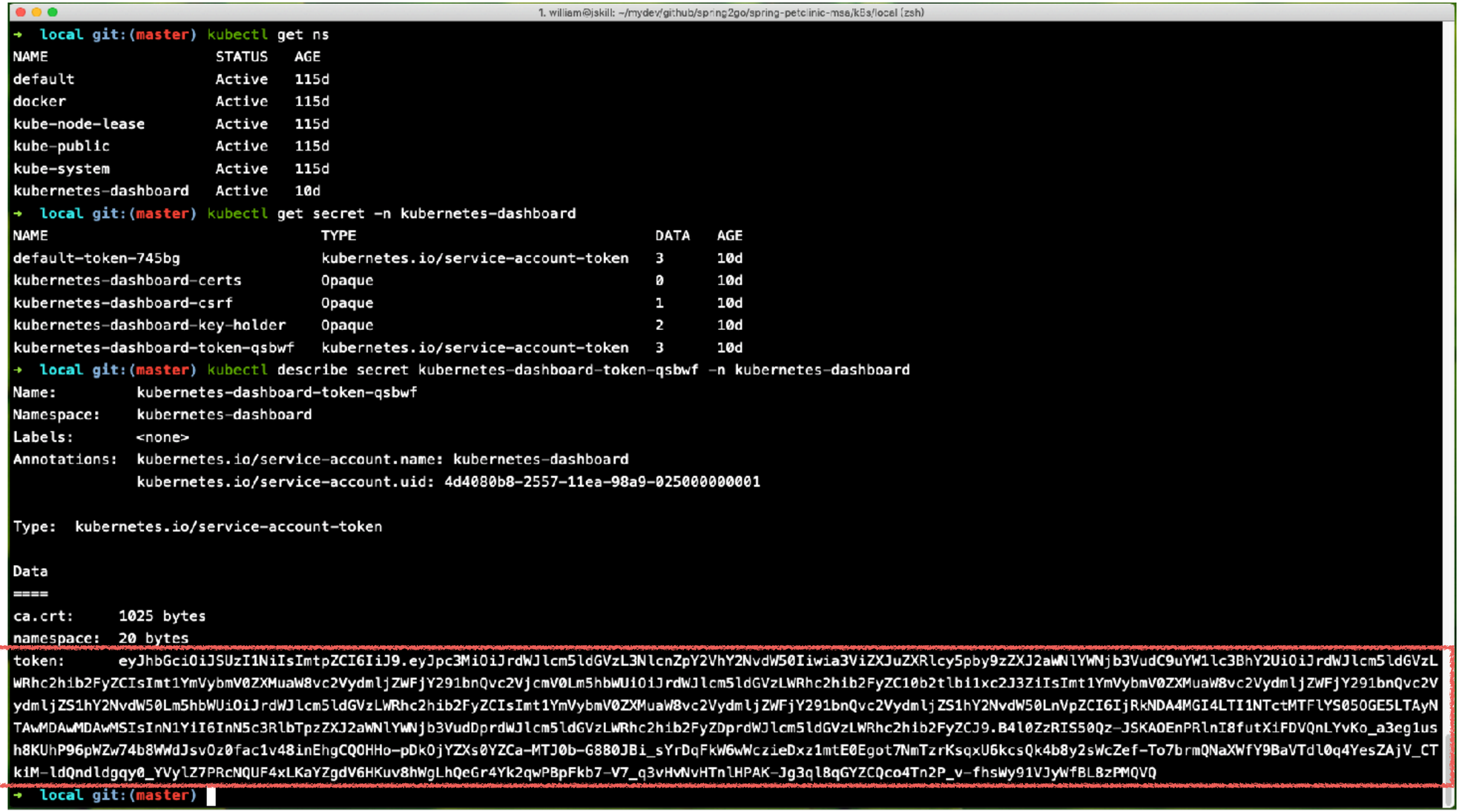
发布Petclinic微服务版 + 查看资源使用



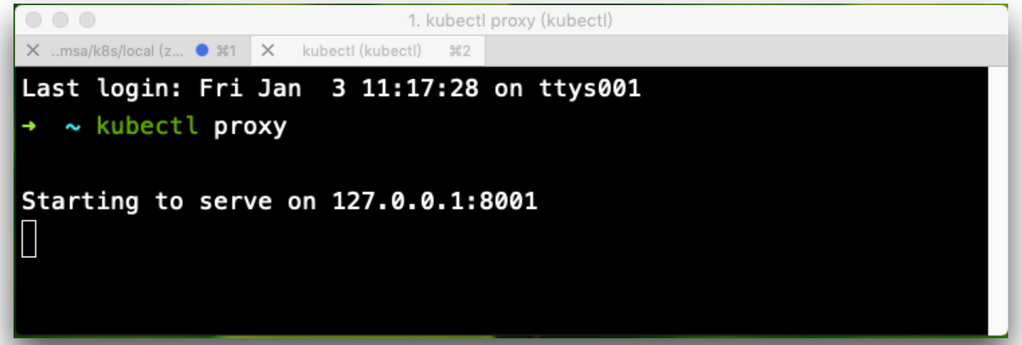
浏览器校验成功

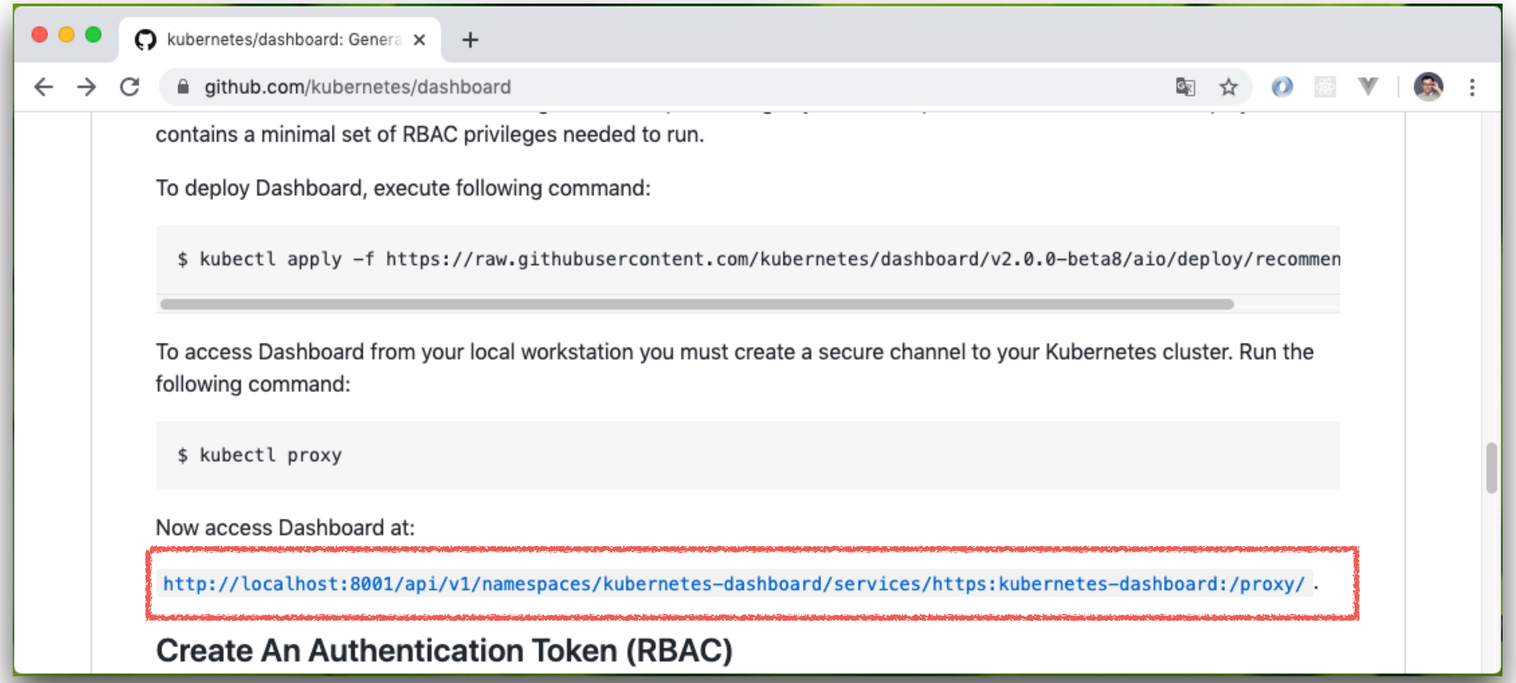

PetClinic :: a Microservices &
localhost:31080/#!/welcome
spring
HOME
OWNERS
VETERINARIANS
Welcome to Petclinic - Microservices Version
spring by Pivotal

查看Dashboard令牌

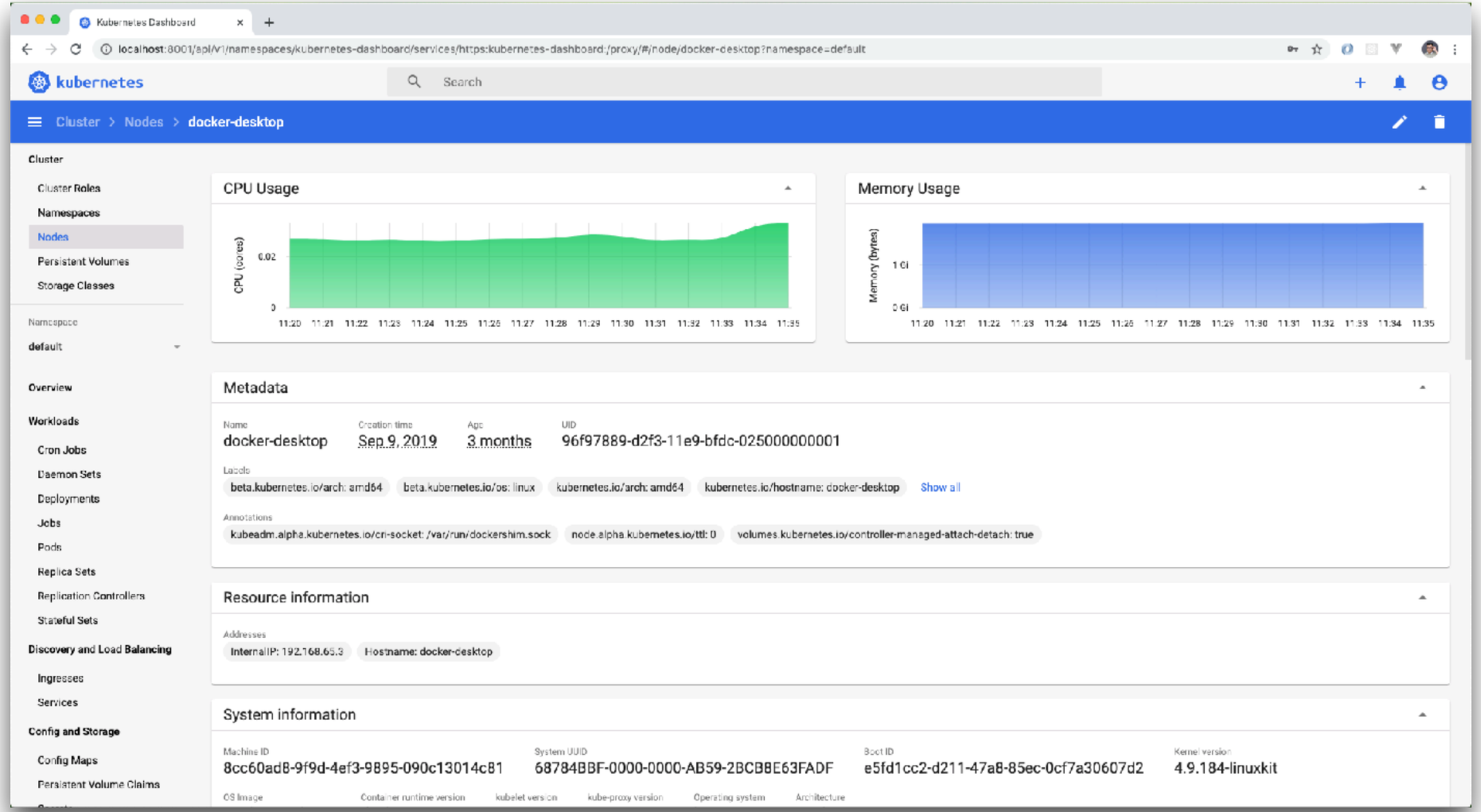


启用Dashboard

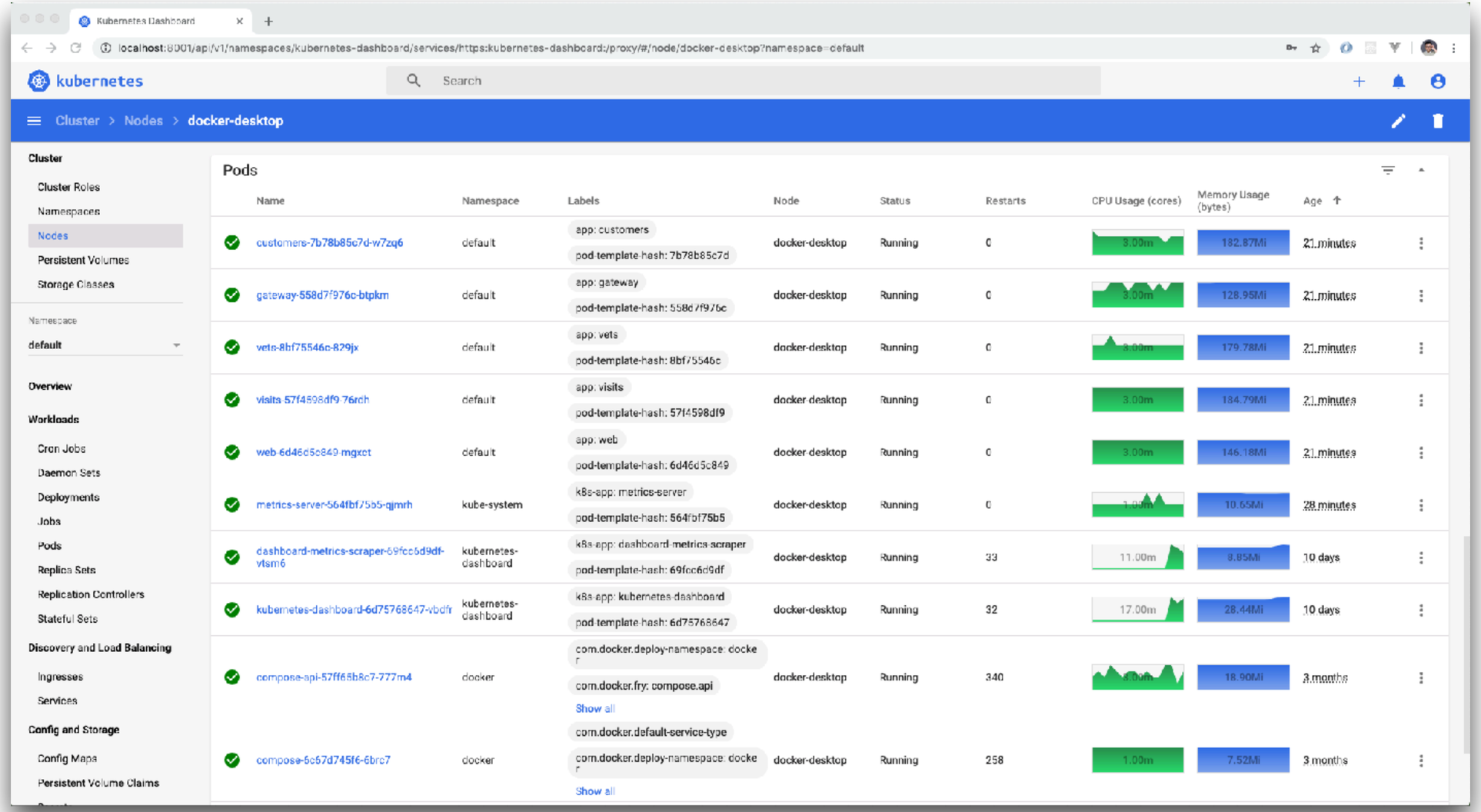




查看节点资源使用

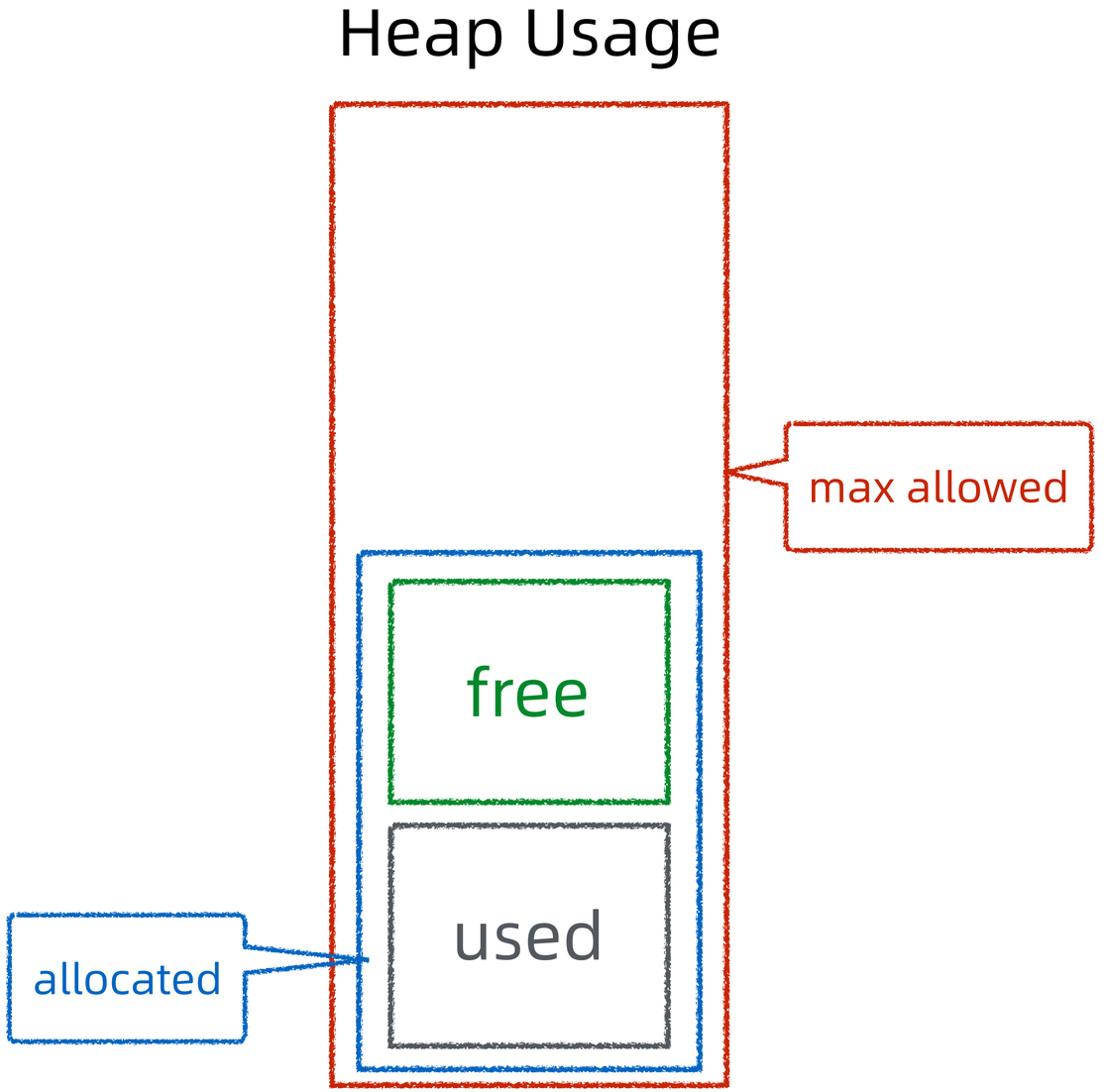

Kubernetes Dashboard
localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/node/docker-desktop?namespace=default
kubernetes
Search
Cluster > Nodes > docker-desktop
Cluster
Cluster Roles
Namespaces
Nodes
Persistent Volumes
Storage Classes
Namespace
default
Overview
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Discovery and Load Balancing
Ingresses
Services
Config and Storage
Config Maps
Persistent Volume Claims
CPU Usage
CPU (cores)
0.02
0
11:20 11:21 11:22 11:23 11:24 11:25 11:26 11:27 11:28 11:29 11:30 11:31 11:32 11:33 11:34 11:35
Memory Usage
Memory (bytes)
1 Gi
0 Gi
11:20 11:21 11:22 11:23 11:24 11:25 11:26 11:27 11:28 11:29 11:30 11:31 11:32 11:33 11:34 11:35
Metadata
Name
docker-desktop
Creation time
Sep 9, 2019
Age
3 months
UID
96f97889-d2f3-11e9-bfdc-025000000001
Labels
beta.kubernetes.io/arch: amd64
beta.kubernetes.io/os: linux
kubernetes.io/arch: amd64
kubernetes.io/hostname: docker-desktop
Show all
Annotations
kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
node.alpha.kubernetes.io/ttl: 0
volumes.kubernetes.io/controller-managed-attach-detach: true
Resource information
Addresses
InternalIP: 192.168.65.3
Hostname: docker-desktop
System information
Machine ID
8cc60ad8-9f9d-4ef3-9895-090c13014c81
System UUID
68784BBF-0000-0000-AB59-2BCB8E63FADF
Boot ID
e5fd1cc2-d211-47a8-85ec-0cf7a30607d2
Kernel version
4.9.184-linuxkit
OS Image
Container runtime version
kubelet version
kube-proxy version
Operating system
Architecture

查看Pod资源使用


Kubernetes Dashboard
localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/node/docker-desktop?namespace=default
kubernetes
Search
Cluster > Nodes > docker-desktop
Cluster
Cluster Roles
Namespaces
Nodes
Persistent Volumes
Storage Classes
Namespace
default
Overview
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Discovery and Load Balancing
Ingresses
Services
Config and Storage
Config Maps
Persistent Volume Claims
Pods
Name
Namespace
Labels
Node
Status
Restarts
CPU Usage (cores)
Memory Usage (bytes)
Age
customers-7b78b85c7d-w7zq6
default
app: customers
pod-template-hash: 7b78b85c7d
docker-desktop
Running
0
3.00m
182.87Mi
21 minutes
gateway-558d7f976c-btpkm
default
app: gateway
pod-template-hash: 558d7f976c
docker-desktop
Running
0
3.00m
128.95Mi
21 minutes
vets-8bf75546c-829jx
default
app: vets
pod-template-hash: 8bf75546c
docker-desktop
Running
0
3.00m
179.78Mi
21 minutes
visits-57f4598df9-76rch
default
app: visits
pod-template-hash: 57f4598df9
docker-desktop
Running
0
3.00m
184.79Mi
21 minutes
web-6d46d5c849-mgxct
default
app: web
pod-template-hash: 6d46d5c849
docker-desktop
Running
0
3.00m
146.18Mi
21 minutes
metrics-server-564fbf75b5-qjmrh
kube-system
k8s-app: metrics-server
pod-template-hash: 564fbf75b5
docker-desktop
Running
0
10.65Mi
28 minutes
dashboard-metrics-scraper-69fcc6d9df-vtsm6
kubernetes-dashboard
k8s-app: dashboard-metrics-scraper
pod-template-hash: 69fcc6d9df
docker-desktop
Running
33
11.00m
8.85Mi
10 days
kubernetes-dashboard-6d75768647-vbdfr
kubernetes-dashboard
k8s-app: kubernetes-dashboard
pod-template-hash: 6d75768647
docker-desktop
Running
32
17.00m
28.44Mi
10 days
compose-api-57ff65b8c7-777m4
docker
com.docker.deploy-namespace: docke r
com.docker.fry: compose.api
Show all
docker-desktop
Running
340
18.90Mi
3 months
compose-6c67d745f6-6brc7
docker
com.docker.default-service-type
com.docker.deploy-namespace: docke r
Show all
docker-desktop
Running
258
1.00m
7.52Mi
3 months

JVM内存结构

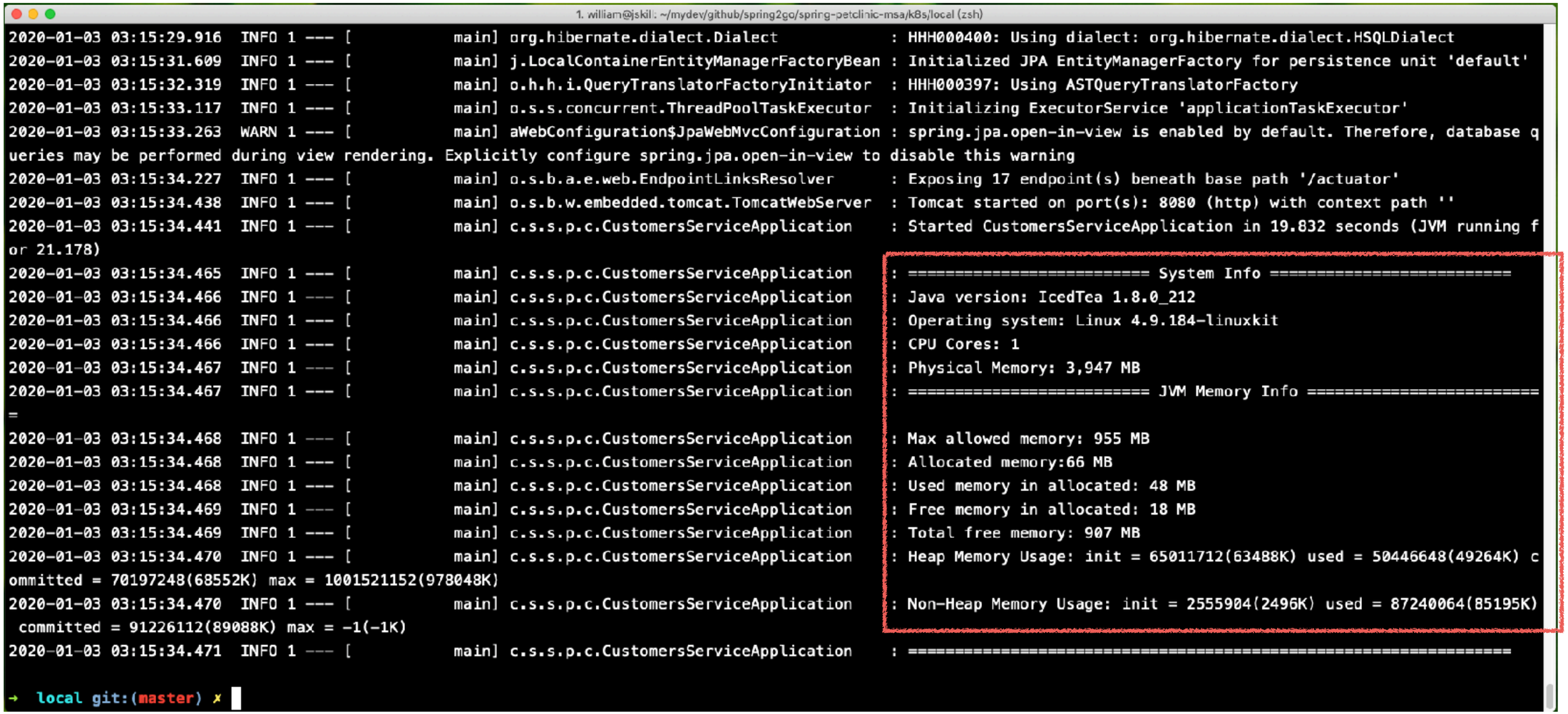

Heap Memory
Non-Heap Memory
other


Heap Usage
max allowed
free
used
allocated

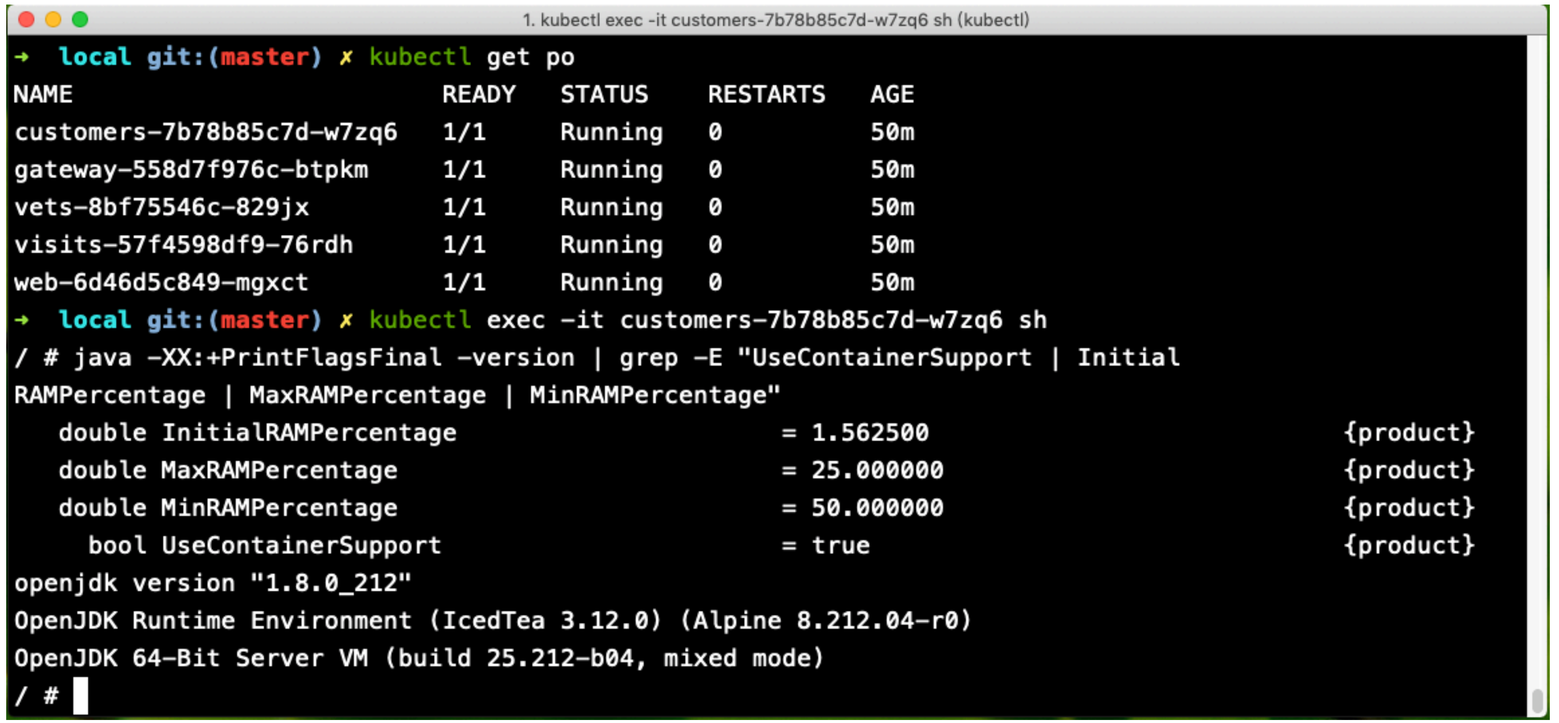
JVM内存监控代码

```
Runtime runtime = Runtime.getRuntime();

NumberFormat format = NumberFormat.getInstance();

long maxMemory = runtime.maxMemory();
long allocatedMemory = runtime.totalMemory();
long freeMemory = runtime.freeMemory();
long mb = 1024 * 1024;
String mega = " MB";

long physicalMemory;
try {
    physicalMemory = ((com.sun.management.OperatingSystemMXBean) ManagementFactory
            .getOperatingSystemMXBean()).getTotalPhysicalMemorySize();
} catch (Exception e) {
    physicalMemory = -1L;
}

int availableCores = Runtime.getRuntime().availableProcessors();

logger.info("========================= System Info ==========================");
logger.info("Java version: " + System.getProperty("java.vendor") + " " + System.getProperty("java.version"));
logger.info("Operating system: " + System.getProperty("os.name") + " " + System.getProperty("os.version"));
logger.info("CPU Cores: " + availableCores);
if (physicalMemory != -1L) {
    logger.info("Physical Memory: " + format.format( number: physicalMemory / mb) + mega);
}
logger.info("========================= JVM Memory Info ==========================");
logger.info("Max allowed memory: " + format.format( number: maxMemory / mb) + mega);
logger.info("Allocated memory:" + format.format( number: allocatedMemory / mb) + mega);
logger.info("Used memory in allocated: " + format.format( number: (allocatedMemory - freeMemory) / mb) + mega);
logger.info("Free memory in allocated: " + format.format( number: freeMemory / mb) + mega);
logger.info("Total free memory: " + format.format( number: (freeMemory + (maxMemory - allocatedMemory)) / mb) + mega);
logger.info("Heap Memory Usage: " + ManagementFactory.getMemoryMXBean().getHeapMemoryUsage());
logger.info("Non-Heap Memory Usage: " + ManagementFactory.getMemoryMXBean().getNonHeapMemoryUsage());
logger.info("=========================================================================\n");
```

查看CustomerService Pod监控日志



查看容器JVM参数



java -XX:+PrintFlagsFinal -version | grep -E "UseContainerSupport | Initial RAMPercentage | MaxRAMPercentage | MinRAMPercentage"

https://stackoverflow.com/questions/54292282/clarification-of-meaning-new-jvm-memory-parameters-initialrampercentage-and-minr

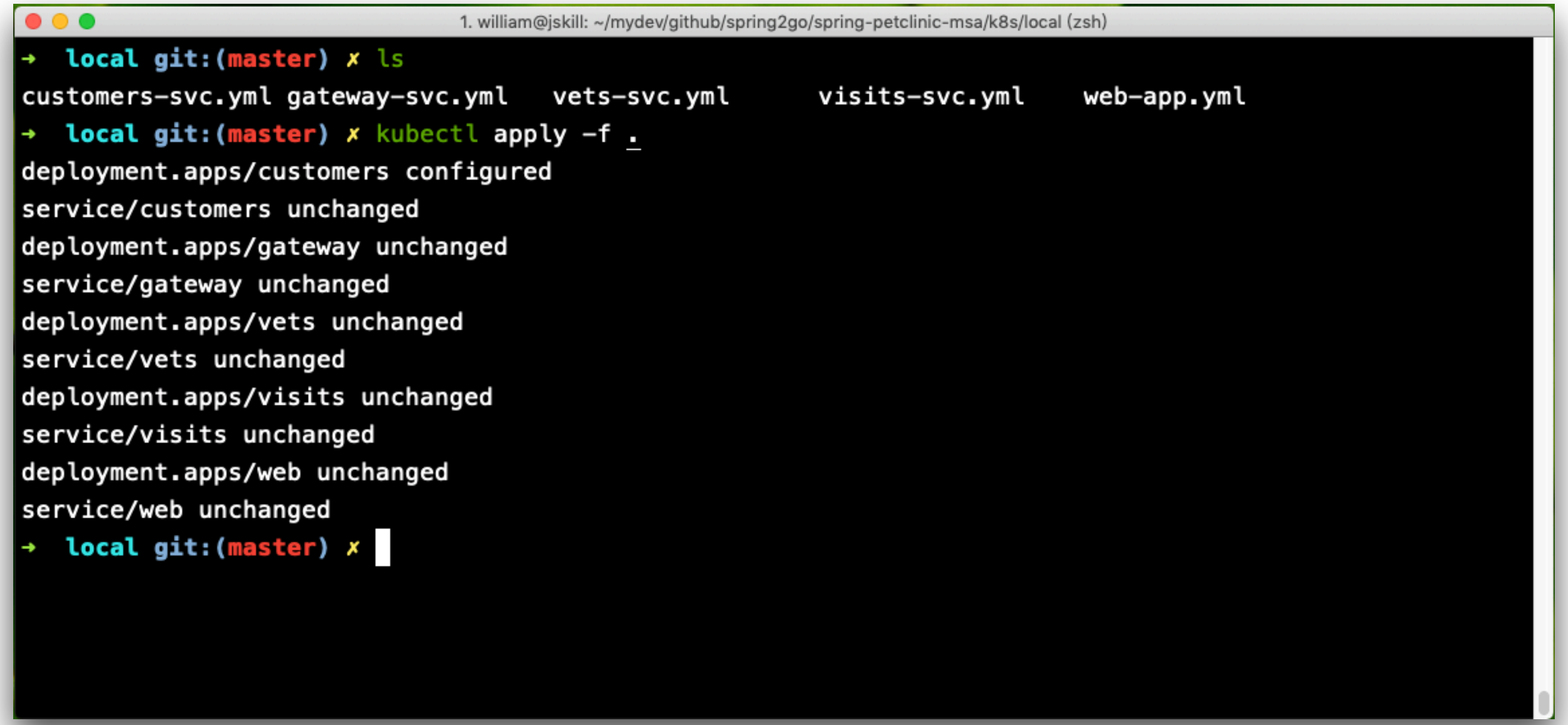
Petclinic微服务Dockerfile更新

```
FROM openjdk:8-jre-alpine
ARG ARTIFACT_NAME
ARG EXPOSED_PORT

ADD ${ARTIFACT_NAME}.jar /usr/share/app.jar

EXPOSE ${EXPOSED_PORT}
# ENTRYPOINT ["java", "-XX:+UnlockExperimentalVMOptions", "-XX:+UseCGroupMemoryLimitForHeap", "-Djava.security
  .egd=file:/dev/./urandom","-jar","/usr/share/app.jar"]
ENTRYPOINT exec java $JAVA_OPTS  -jar /usr/share/app.jar
```

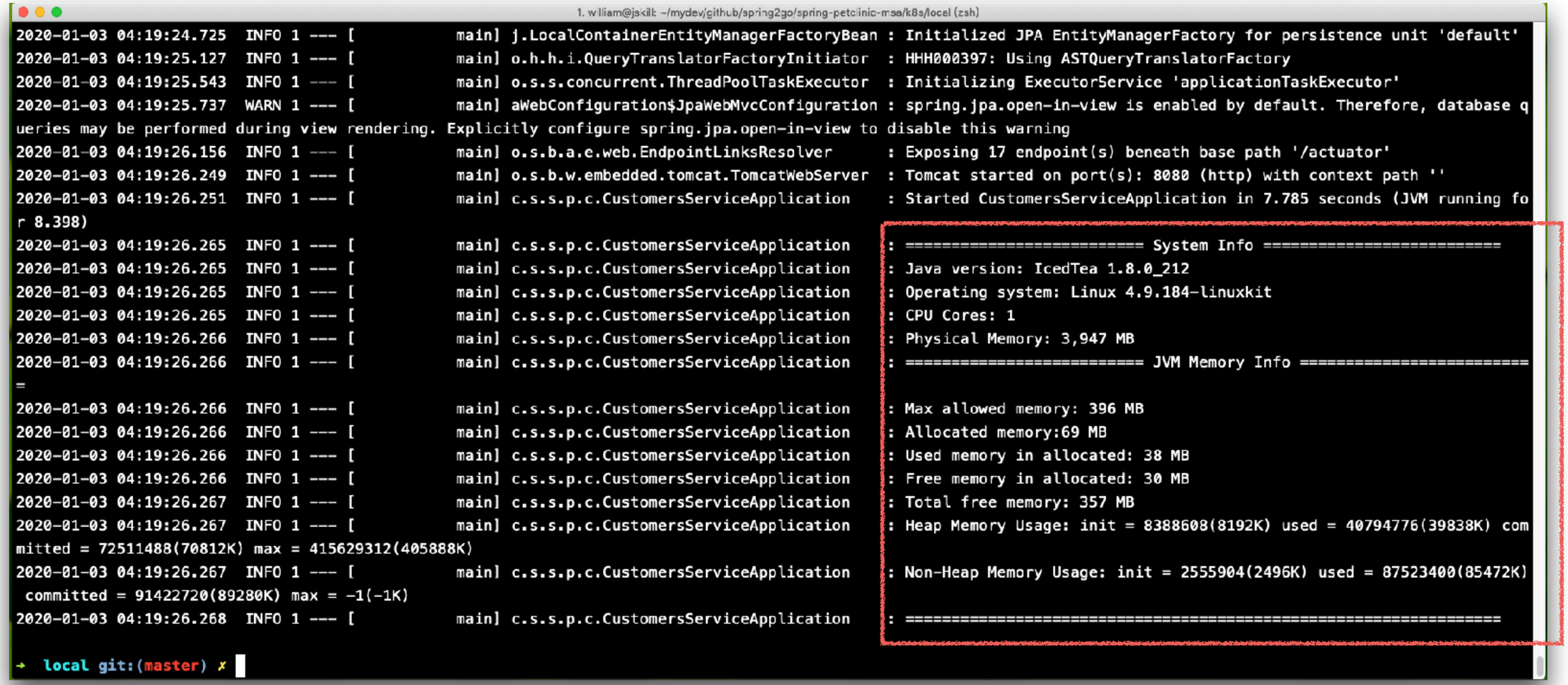
更新customers-svc.yml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: customers
spec:
  selector:
    matchLabels:
      app: customers
  replicas: 1
  template:
    metadata:
      labels:
        app: customers
    spec:
      containers:
        - name: customers
          image: spring2go/spring-petclinic-customers-service:1.0.0.RELEASE
          resources:
            requests:
              memory: "128Mi"
            limits:
              memory: "512Mi"
          env:
          - name: JAVA_OPTS
            value: "-XX:MaxRAMPercentage=80.0"
          - name: SERVER_PORT
            value: "8080"
```

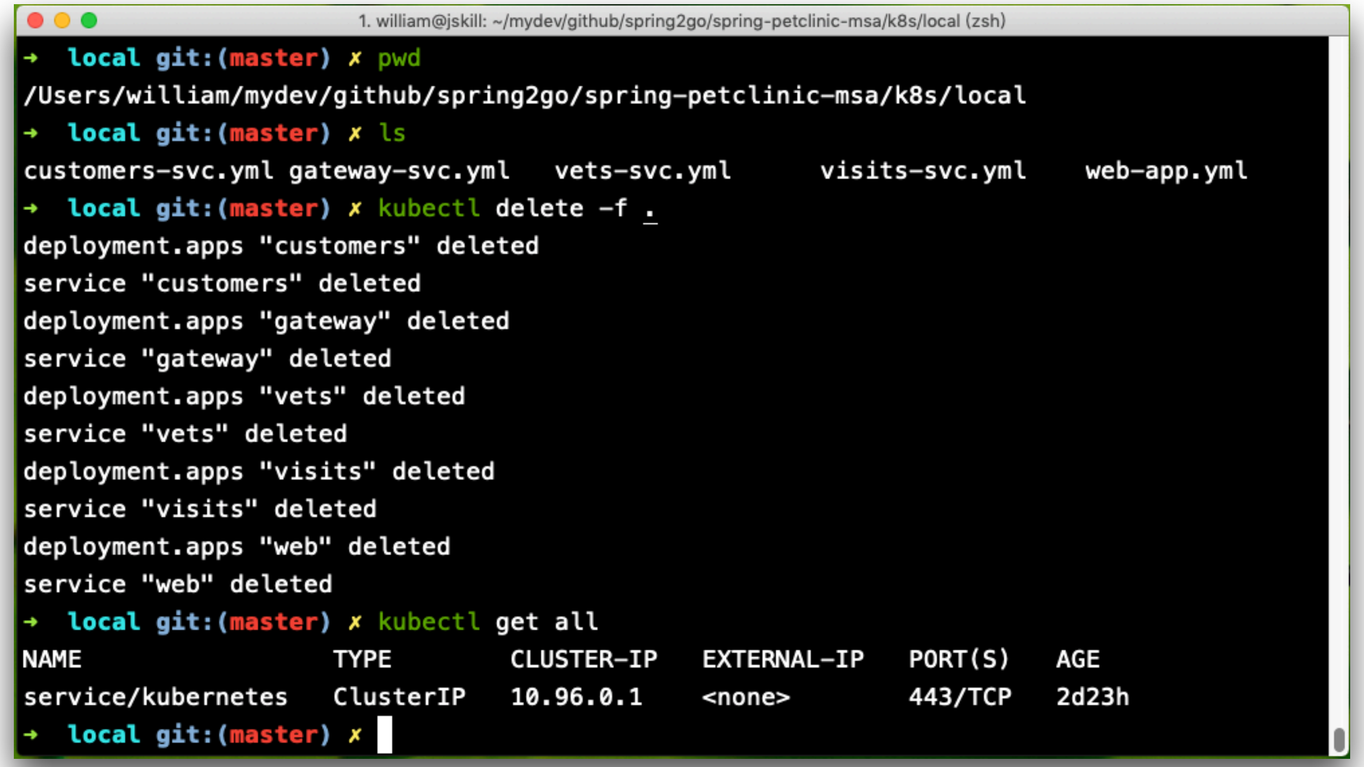
更新发布Petclinic微服务



查看CustomerService Pod监控日志



环境清理


1. william@jskill: ~/mydev/github/spring2go/spring-petclinic-msa/k8s/local (zsh)
➜ local git:(master) ✗ pwd
/Users/william/mydev/github/spring2go/spring-petclinic-msa/k8s/local
➜ local git:(master) ✗ ls
customers-svc.yml gateway-svc.yml vets-svc.yml visits-svc.yml web-app.yml
➜ local git:(master) ✗ kubectl delete -f .
deployment.apps "customers" deleted
service "customers" deleted
deployment.apps "gateway" deleted
service "gateway" deleted
deployment.apps "vets" deleted
service "vets" deleted
deployment.apps "visits" deleted
service "visits" deleted
deployment.apps "web" deleted
service "web" deleted
➜ local git:(master) ✗ kubectl get all
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 2d23h
➜ local git:(master) ✗

本课小结

- 查看node/pod资源使用
  - kubectl top(需启用Metrics Server)
  - K8s Dashboard
- JVM内存结构
  - Heap + Non Heap + 其它
  - 从初使分配开始，随实际使用增加向OS申请，直到允许的最大请求
- JVM设置建议
  - 升级到最新JDK，Java 8u191和Java 10+支持UseContainerSupport
  - 设置K8s内存Limit + MaxRAMPercentage JVM启动参数
  - 实际设置要监控调优