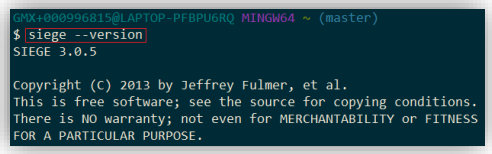
“SCALING & AUTOSCALING”

El siguiente documento detallará el paso a paso, el manejo del: **SCALING & AUTOSCALING** de los ***CONTENEDORES*** en **OPENSHIFT**. La ***AGENDA*** que se manejará será la siguiente:

- ✓ **REQUERIMIENTOS.**
- ✓ **ESCALAMIENTO DE 'CONTENEDORES'.**
  - ***SCALING INSTANCES.***
  - ***HORIZONTAL AUTOSCALING INSTANCES.***

DESCRIPCIÓN	DETALLE
1. REQUERIMIENTOS:	
El único requerimiento es tener ***INSTALADA*** la herramienta: **SIEGE** nivel de: **VARIABLES DE ENTORNO**. Esta herramienta permitirá ***PROBAR*** más adelante la aplicación del ***RECURSO***: **AUTOSCALING**. ▪ **WINDOWS:**  siege-windows.zip *$ siege –version* ▪ **LINUX:** *$ sudo apt update -y* *$ sudo apt install siege -y* *$ siege --version* ▪ **MAC:** *$ brew install siege* *$ siege --version*	

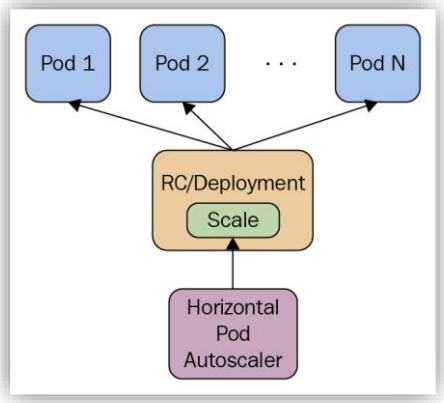
```
cguerra@LAPTOP-PFBPU6RQ:~$
cguerra@LAPTOP-PFBPU6RQ:~$ siege --version
New configuration template added to /home/cguerra/.siege
Run siege -C to view the current settings in that file
SIEGE 4.0.4

Copyright (C) 2017 by Jeffrey Fulmer, et al.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS
FOR A PARTICULAR PURPOSE.
```

```
gianniguatame — -zsh — 80×24
gianniguatame@Giannis-MacBook-Pro ~ % siege --version
New configuration template added to /Users/gianniguatame/.siege
Run siege -C to view the current settings in that file
SIEGE 4.1.3

Copyright (C) 2022 by Jeffrey Fulmer, et al.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS
FOR A PARTICULAR PURPOSE.
```

2. ESCALAMIENTO DE *'CONTENEDORES'*:



El objetivo aquí es el poder ***CRECER*** a nivel de ***INSTANCIAS (PODs),*** para aplicarlos en los escenarios requeridos. Estos pueden ser: ***MANUÁLMENTE*** o ***AUTOMÁTICAMENTE***.

A. SCALING *INSTANCES*:

Este procedimiento es el **MANUAL** & para ello se debe crear un ***ESCENARIO*** de prueba, corriendo el **Script *YAMEL***:

Script_Microservicio
.yaml

*$ cat > **Script_Microservicio.yaml***
…
…..
…….
$ oc create -f Script_Microservicio.yaml
$ oc get all -n dummy-csm-crga

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ cat > Script_Microservicio.yaml
#### -------- [NAMESPACE] -------- ####
apiVersion: v1
kind: Namespace
metadata:
  name: dummy-csm-crga

---

#### -------- [DEPLOYMENT] -------- ####
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dummy-micro-deploy
  namespace: dummy-csm-crga
  labels:
    app: dummy-micro-service
    version: v1
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dummy-micro-service
      version: v1
  template:
    metadata:
      labels:
        app: dummy-micro-service
        version: v1
    spec:
      containers:
      - image: maktup/dummy-micro-01:latest
        name: dummy-micro-container
        resources:
          limits:
            cpu: 250m
            memory: 250Mi
          requests:
            cpu: 100m
            memory: 100Mi
        ports:
        - containerPort: 8080
---

#### -------- [SERVICE] -------- ####
```

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc create -f Script_Microservicio.yaml
namespace/dummy-csm-crga created
deployment.apps/dummy-micro-deploy created
service/dummy-micro-service created
route.route.openshift.io/dummy-micro-route created

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc get all -n dummy-csm-crga
NAME                                    READY   STATUS    RESTARTS   AGE
pod/dummy-micro-deploy-c54c7cf69-xs95l   1/1     Running   0          10s

NAME                         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
service/dummy-micro-service   ClusterIP   172.21.30.139   <none>        8080/TCP   10s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/dummy-micro-deploy   1/1     1            1           10s

NAME                                           DESIRED   CURRENT   READY   AGE
replicaset.apps/dummy-micro-deploy-c54c7cf69   1         1         1       10s

NAME                                          HOST/PORT
      PATH   SERVICES             PORT    TERMINATION   WILDCARD
route.route.openshift.io/dummy-micro-route   dummy-micro-route-dummy-csm-crga.cluster-cla-crga-
loud          dummy-micro-service   <all>                 None
```

Luego, se ingresa el ***COMANDO*** para el **AUMENTO *MANUAL*** hacia 5 ***INSTANCIAS***:

$ oc scale deployment dummy-micro-deploy --replicas=5 -n dummy-csm-crga

$ oc get pods -n dummy-csm-crga

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc scale deployment dummy-micro-deploy --replicas=5 -n dummy-csm-crga
deployment.apps/dummy-micro-deploy scaled

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc get pods -n dummy-csm-crga
NAME                                 READY   STATUS    RESTARTS   AGE
dummy-micro-deploy-c54c7cf69-2csfl   1/1     Running   0          11s
dummy-micro-deploy-c54c7cf69-h2rdr   1/1     Running   0          11s
dummy-micro-deploy-c54c7cf69-phrtj   1/1     Running   0          11s
dummy-micro-deploy-c54c7cf69-rpmw8   1/1     Running   0          11s
dummy-micro-deploy-c54c7cf69-xs95l   1/1     Running   0          11m
```

Luego, se realiza una prueba del ***MICROSERVICIO*** desplegado. Los ***REQUEST*** enviados serán distribuidos a nivel de ***TODAS*** las ***INSTANCIAS*** desplegadas:

$ curl http://dummy-micro-route-dummy-csm-crga.cluster-cla-crga-ccc03eca20d26e6ac64511f874a64b9b-0000.br-sao.containers.appdomain.cloud/dummy-micro-01/get/personas

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ curl http://dummy-micro-route-dummy-csm-crga.cluster-cla-crga-ccc03eca20d26e6ac64511f874a64b9b-0000.br-sao.co
ntainers.appdomain.cloud/dummy-micro-01/get/personas
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   196  100   196    0     0     58      0  0:00:03  0:00:03 --:--:--    60[{ 'nombre': 'PAOLO GUERRERO', 'e
dad': 35, 'rol': 'CONSULTOR'   },{ 'nombre': 'LUIS GUADALUPE', 'edad': 40, 'rol': 'PROGRAMADOR' },{ 'nombre': '
PEDRO SALAZAR',  'edad': 30, 'rol': 'ARQUITECTO'  }]
```

Finalmente, para el **DECREMENTO** ***MANUAL*** de ***INSTANCIAS*** hacia: 1 *(dejarlo como estaba inicialmente)*, se ingresa:

$ oc scale deployment dummy-micro-deploy --replicas=1 -n dummy-csm-crga

$ oc get pods -n dummy-csm-crga

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc scale deployment dummy-micro-deploy --replicas=1 -n dummy-csm-crga
deployment.apps/dummy-micro-deploy scaled

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc get pods -n dummy-csm-crga
NAME                                  READY   STATUS        RESTARTS   AGE
dummy-micro-deploy-c54c7cf69-2csfl    0/1     Terminating   0          10m
dummy-micro-deploy-c54c7cf69-h2rdr    1/1     Running       0          10m
dummy-micro-deploy-c54c7cf69-phrtj    0/1     Terminating   0          10m
dummy-micro-deploy-c54c7cf69-rpmw8    0/1     Terminating   0          10m
dummy-micro-deploy-c54c7cf69-xs95l    0/1     Terminating   0          21m

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc get pods -n dummy-csm-crga
NAME                                  READY   STATUS    RESTARTS   AGE
dummy-micro-deploy-c54c7cf69-h2rdr    1/1     Running   0          13m
```

B. HORIZONTAL AUTOSCALING *INSTANCES*:

El objetivo aquí es el poder ***CRECER*** a nivel de ***INSTANCIAS (PODs),*** esto aplicado en los escenarios de ***TRÁFICO CAMBIANTE,*** sobre todo, pero de manera: ***AUTOMÁTICA***.

IMPORTANTE: *"Antes que nada, se debe conocer que el **HORIZONTAL AUTOSCALING**, requiere que esté **ACTIVO** algo que se llama: **METRIC-SERVER**, esto sirve para poder desde **OPENSHIFT** medir el control **CPU** & **MEMORIA** tanto a nivel de: **NODOS** & **PODs** respectivamente.*
*En **KUBERNETES** no está instalado, pero si en **OPENSHIFT** **(NO se requiere)**":*

$ git clone https://github.com/kodekloudhub/kubernetes-metrics-server.git
$ cd kubernetes-metrics-server
$ oc create -f ./

Luego, para ***PROBAR*** este escenario se requiere ejecutar los ***COMANDOS IMPERATIVOS*** siguientes:

$ oc autoscale deployment dummy-micro-deploy --min=1 --max=4 --cpu-percent=40 -n dummy-csm-crga

$ oc get pods,hpa -o wide -n dummy-csm-crga

IMPORTANTE: *"Esta modalidad a nivel de **COMANDOS**, aplica una **REGLA** que funciona solo a nivel de **CPU**".*

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc autoscale deployment dummy-micro-deploy --min=1 --max=4 --cpu-percent=40 -n dummy-csm-crga
horizontalpodautoscaler.autoscaling/dummy-micro-deploy autoscaled

GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc get pods,hpa -o wide -n dummy-csm-crga
NAME                                      READY   STATUS    RESTARTS   AGE   IP              NODE             NOMINAT
pod/dummy-micro-deploy-c54c7cf69-h2rdr    1/1     Running   0          26m   172.30.240.218  10.150.181.189   <none>

NAME                                                          REFERENCE                          TARGETS   MINPODS   MAXPODS
horizontalpodautoscaler.autoscaling/dummy-micro-deploy        Deployment/dummy-micro-deploy      3%/40%    1         4
```

Luego, es necesario obtener el **ROUTE** creado del ***MICROSERVICIO***, ejecutando:

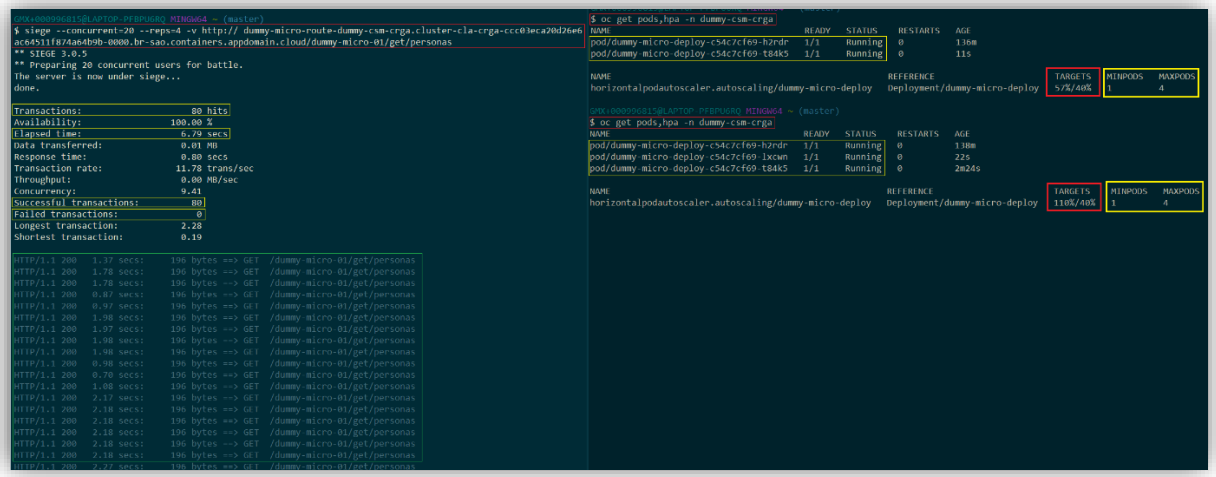
$ oc get routes -n dummy-csm-crga

```
GMX+000996815@LAPTOP-PFBPU6RQ MINGW64 ~ (master)
$ oc get routes -n dummy-csm-crga
NAME                  HOST/PORT
     PATH   SERVICES                PORT    TERMINATION   WILDCARD
dummy-micro-route     dummy-micro-route-dummy-csm-crga.cluster-cla-crga-ccc03eca20d26e6ac64511f874a64b9b-0000.br-sao.containers.appdomain.cloud
            dummy-micro-service   <all>                 None
```

Luego, se requiere probar ***MASIVAMENTE*** contra el ***MICROSERVICIO*** desplegado. Así mismo, los ***REQUEST*** serán distribuidos a nivel de ***TODAS*** las ***INSTANCIAS*** desplegadas y/o se vayan **ACTIVANDO**:

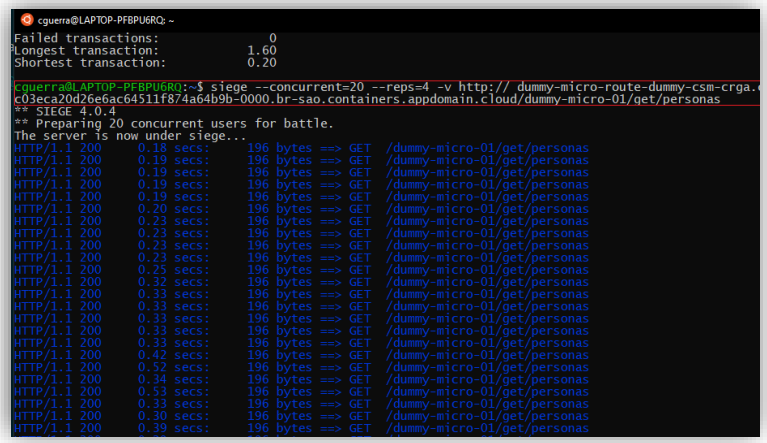
SIEGE permitirá enviar ***REQUEST*** de manera ***PARALELA***, esto significa que en la demostración se enviarán: ***4 PETICIONES***, cada una con ***20 CONCURRENCIAS*** *(en **TOTAL** se deberán ejecutar: **80 Mensajes**)*.

$ siege --concurrent=20 --reps=4 -v http://dummy-micro-route-dummy-csm-crga.cluster-cla-crga-ccc03eca20d26e6ac64511f874a64b9b-0000.us-south.containers.appdomain.cloud/dummy-micro-01/get/personas

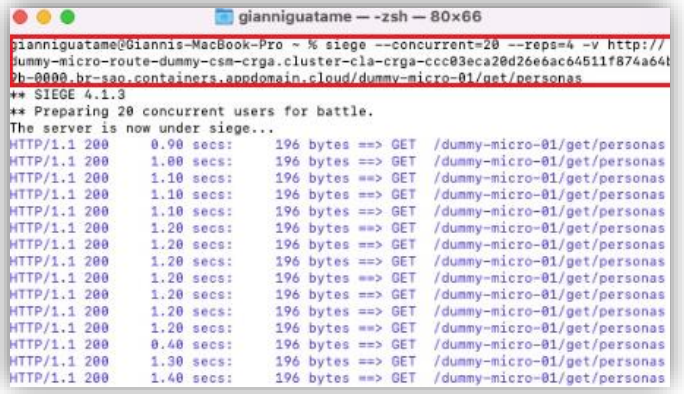


IMPORTANTE: *"La ejecución del **TRÁFICO MASIVO** hará que el **CPU**, como se muestra la **IMAGEN**, se incremente & genere la **ELASTICIDAD** a nivel de **INSTANCIAS** de **PODs"**.*

$ oc get pods,hpa -n dummy-csm-crga



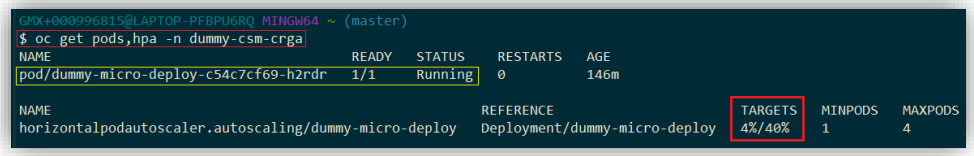
(VERSIÓN LINUX)



(VERSIÓN MAC)

Luego, se debe verificar que después de un ***TIEMPO*** de realizada la ***PRUEBA*** de: **REQUEST MASIVOS**, la plataforma se debe haber estabilizado a nivel de ***CPU***. *(La **ELIMINACIÓN** de los **PODs** adicionales es **AUTOMÁTICA**).*

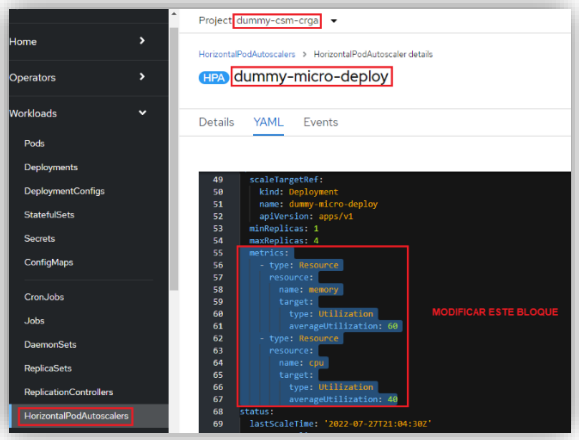
$ oc get pods,hpa -n dummy-csm-crga



Finalmente, solo para **COMPLEMENTAR** que la modalidad en ***SCRIPT*** del: **HORIZONTAL AUTOSCALING**, **NO** solamente funciona a nivel de ***CPU***, sino que también funciona ***SIMULTÁNEAMENTE*** a nivel de ***MEMORIA***.

*"Esto vuelve más **SENSIBLE** la activación del: **HORIZONTAL AUTOSCALING**".*

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: dummy-micro-deploy
  namespace: dummy-csm-crga
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: dummy-micro-deploy
  minReplicas: 1
  maxReplicas: 4
  metrics:
  - type: Resource
    resource:
      name: cpu
```



```
    target:
      type: Utilization
      averageUtilization: 40
  - type: Resource
    resource:
      name: memory
    target:
      type: Utilization
      averageUtilization: 60
```

$ oc get pods,hpa -n dummy-csm-crga

IMPORTANTE: *"Se aprecia como la **REGLA** del **HORIZONTAL AUTOSCALING**, ahora está aplicara a nivel de: **CPU** & **MEMORIA**".*