Table of Contents

* [Quotas, LimitRanges, and Templates](https://www.opentlc.com/labs/ocp_advanced_deployment/04_3_Quotas_LimitRanges_Templates_Solution_Lab.html#_quotas_limitranges_and_templates)
* [1. Create Quotas for Cluster](https://www.opentlc.com/labs/ocp_advanced_deployment/04_3_Quotas_LimitRanges_Templates_Solution_Lab.html#labexercises)
* [2. Import 3scale AMP Template](https://www.opentlc.com/labs/ocp_advanced_deployment/04_3_Quotas_LimitRanges_Templates_Solution_Lab.html#_import_3scale_amp_template)
* [3. Deploy System](https://www.opentlc.com/labs/ocp_advanced_deployment/04_3_Quotas_LimitRanges_Templates_Solution_Lab.html#_deploy_system)
* [4. Clean Up Environment](https://www.opentlc.com/labs/ocp_advanced_deployment/04_3_Quotas_LimitRanges_Templates_Solution_Lab.html#_clean_up_environment)

Quotas, LimitRanges, and Templates

**Scenario**

MitziCom is starting production, and they are very concerned that they may run out of resources. In order to address these concerns, you need to set quotas and limits on the system.

MitziCom has decided to deploy Red Hat 3scale API Management in their RHOCP environment to evaluate its performance impact on the cluster. They have asked you to configure RHOCP to deploy 3scale API Management in a repeatable manner, taking into account its resource requirements. You need to edit the template that deploys 3scale API Management in order for users to be able to deploy it with enough resources to function properly.

**Goals**

* Create quotas for the cluster
* Import the 3scale AMP template to set up 3scale API Management
* Deploy the system
* Edit the template and LimitRange to launch 3scale properly

1. Create Quotas for Cluster

1. On your **bastion** host, make sure you have a **$GUID** environment variable set up:
2. sudo -i
3. export GUID=`hostname | awk -F. '{print $2}'`

echo "export GUID=$GUID" >> .bashrc

1. Switch to the **andrew** user:

oc login -u andrew -p r3dh4t1!

1. Create a new project for 3scale API Management:

oc new-project 3scale

1. Switch to the **system:admin** user to set a quota on **andrew**:

oc login -u system:admin

1. Create a limited cluster resource quota that sets the following hard limits:
   * **pods=25**
   * **requests.memory=6Gi**
   * **requests.cpu=5**
   * **limits.cpu=25**
   * **limits.memory=40Gi**
   * **configmaps=25**
   * **persistentvolumeclaims=25**
   * **services=25**
   * **project-annotation-selector** for the **andrew** user
2. export OCP\_USERNAME=andrew
3. oc create clusterquota clusterquota-${OCP\_USERNAME} \
4. --project-annotation-selector=openshift.io/requester=$OCP\_USERNAME \
5. --hard pods=25 \
6. --hard requests.memory=6Gi \
7. --hard requests.cpu=5 \
8. --hard limits.cpu=25 \
9. --hard limits.memory=40Gi \
10. --hard configmaps=25 \
11. --hard persistentvolumeclaims=25 \

--hard services=25

1. View the cluster resource quota:

oc get clusterresourcequota

**Sample Output**

NAME LABEL SELECTOR ANNOTATION SELECTOR

clusterquota-andrew <none> map[openshift.io/requester:andrew]

1. Describe the cluster resource quota:

oc describe clusterresourcequota clusterquota-andrew

**Sample Output**

Name: clusterquota-andrew

Created: 6 minutes ago

Labels: <none>

Annotations: <none>

Namespace Selector: ["3scale"]

Label Selector:

AnnotationSelector: map[openshift.io/requester:andrew]

Resource Used Hard

-------- ---- ----

configmaps 0 25

limits.cpu 0 25

limits.memory 0 40Gi

persistentvolumeclaims 0 25

pods 0 25

requests.cpu 0 5

requests.memory 0 6Gi

services 0 25

2. Import 3scale AMP Template

1. Switch back to the **andrew** user:

oc login -u andrew -p r3dh4t1!

1. Get the 3scale AMP template, PVCs, and wildcard router:

wget https://raw.githubusercontent.com/3scale/3scale-amp-openshift-templates/2.1.0-GA/amp/amp.yml

1. Optionally, download the APIcast template:

wget https://raw.githubusercontent.com/3scale/3scale-amp-openshift-templates/2.1.0-GA/apicast-gateway/apicast.yml

1. Create the 3scale AMP template and, optionally, the APIcast template:
2. oc create -f amp.yml

oc create -f apicast.yml

3. Deploy System

1. Create a 3scale Admin Portal with **oc new-app**:

oc new-app --template=system --param WILDCARD\_DOMAIN=apps.$GUID.example.opentlc.com

**Sample Output**

[...]

--> Success

Access your application via route '3scale-admin.apps.0724.example.opentlc.com'

Access your application via route 'backend-3scale.apps.0724.example.opentlc.com'

Access your application via route '3scale.apps.0724.example.opentlc.com'

Access your application via route 'api-3scale-apicast-staging.apps.0724.example.opentlc.com'

Access your application via route 'api-3scale-apicast-production.apps.0724.example.opentlc.com'

Access your application via route 'apicast-wildcard.apps.0724.example.opentlc.com'

Run 'oc status' to view your app.

|  |  |
| --- | --- |
|  | Your **system-storage** PVC may not be able to find a PV that fits its requirements. In that case, you need to create one on your NFS server that does. It is recommended that you edit one of your existing **user-vols**, but you may create a new export and PV if you choose.   * 1. Check the **system-storage** PVC:   oc get pvc system-storage -o yaml  **Sample Output**  apiVersion: v1  kind: PersistentVolumeClaim  metadata:  [ ... ]  spec:  accessModes:  - ReadWriteMany  resources:  requests:  storage: 100Mi  status:  phase: Pending   * 1. If needed, edit one of the PVs and change it to RWX (ReadWriteMany):   oc edit pv pv99   * 1. If needed, roll out deployment configurations again (they quit after 30 minutes):   2. oc rollout latest system-resque   3. oc rollout latest system-sidekiq   oc delete pod system-resque-1-deploy system-sidekiq-1-deploy |

1. Verify that no pods were launched:

oc get pods -n 3scale

**Sample Output**

No resources found.

1. Find out why:

oc get events | grep zync-database

**Sample Output**

10m 10m 1 zync-database DeploymentConfig Normal DeploymentCreated deploymentconfig-controller Created new replication controller "zync-database-1" for version 1

8m 10m 17 zync-database DeploymentConfig Warning FailedCreate deployer-controller Error creating deployer pod: pods "zync-database-1-deploy" is forbidden: failed quota: clusterquota-andrew: must specify limits.cpu,limits.memory,requests.cpu,requests.memory

8m 10m 2 zync-database DeploymentConfig Warning FailedRetry deployer-controller Stop retrying: couldn't create deployer pod for "3scale/zync-database-1": pods "zync-database-1-deploy" is forbidden: failed quota: clusterquota-andrew: must specify limits.cpu,limits.memory,requests.cpu,requests.memory

1. Deploy a limit range to specify limits on all objects in the 3scale project.
2. oc login -u system:admin
3. echo 'apiVersion: v1
4. kind: LimitRange
5. metadata:
6. creationTimestamp: null
7. name: 3scale-resource-limits
8. spec:
9. limits:
10. - type: Pod
11. max:
12. cpu: "10"
13. memory: 8Gi
14. min:
15. cpu: 50m
16. memory: 100Mi
17. - type: Container
18. min:
19. cpu: 50m
20. memory: 100Mi
21. max:
22. cpu: "10"
23. memory: 8Gi
24. default:
25. cpu: 50m
26. memory: 100Mi
27. defaultRequest:
28. cpu: 50m
29. memory: 100Mi
30. maxLimitRequestRatio:

cpu: "200" ' | oc create -f -

1. Deploy the project again.

|  |  |
| --- | --- |
|  | Deploying a template with ten deployments of one pod exceeds your pod quota. |

1. Have a look at the **ClusterResourceQuotas** that are applied in the current context

oc describe AppliedClusterResourceQuota

1. Confirm that no pods were created, then find out why:

oc get events -w

**Sample Output**

2018-04-12 04:15:19 +0000 UTC 2018-04-12 04:15:19 +0000 UTC 1 system-app DeploymentConfig Normal DeploymentCreated deploymentconfig-controller Created new replication controller "system-app-1" for version 1

2018-04-12 04:16:42 +0000 UTC 2018-04-12 04:15:19 +0000 UTC 15 system-app DeploymentConfig Warning FailedCreate deployer-controller Error creating deployer pod: pods "system-app-1-deploy" is forbidden: exceeded quota: clusterquota-andrew, requested: requests.cpu=200m, used: requests.cpu=400m, limited: requests.cpu=500m

1. Edit the template and LimitRange to launch 3scale properly.

|  |  |
| --- | --- |
|  | Examine resource utilization of all the pods with **oc adm top pod** |

1. Edit the container requests and limits in your template to restrict CPU and RAM for the **system-app** pod and the other containers in the template.
   1. Here is a working template file with a basic solution:
   2. wget https://raw.githubusercontent.com/newgoliath/ocp\_advanced\_deployment\_assets/master/Managing\_Compute\_Resources/2\_amp\_template\_solution.yaml
   3. wget https://raw.githubusercontent.com/newgoliath/ocp\_advanced\_deployment\_assets/master/Managing\_Compute\_Resources/3\_amp\_template\_solution.yaml
   4. oc login -u andrew
   5. oc project 3scale

oc delete template system

* 1. Compare your templates to see in which important ways they differ.

vimdiff 3\_amp\_template\_solution.yaml 2\_amp\_template\_solution.yaml

* 1. Find the one you think will work well, and try it out:

oc create -f <template>

4. Clean Up Environment

1. Remove the project from your environment:
2. oc login -u system:admin

oc delete project 3scale

1. Confirm that you still have cluster quotas in your system:

oc get clusterresourcequota

1. Optionally, remove the cluster quota:

oc delete clusterresourcequota clusterquota-andrew

This completes the 3scale API Management lab.

Build Version: c3147ce9f77191e30b447cc423f2f68a0c40fc03 : Last updated 2018-07-31 01:29:02 EDT