Best practices for openshift containers provisionning

As we create more and more projects in a commonly shared Quota for Service Fullfillment, it is key that developers and integrators follow these guidelines to keep "red tape" at minimum and ensure that:

1. The design of your application does not impact other applications
2. The CPU/RAM/Storage resources are optimized

Checklist for any app (Batch, API, Broker, Stream etc) deploying/provisionning on DEV/UAT /PROD

At Project/Namespace creation time

1. When requesting a new namespace/project, please write an email to the Service Fullfillment **Quota Admins** : louis-philippe.laperriere@cgi.com an d benoit.turcotte@bell.ca and include the following information:
    a. Which application does this belong to i.e. SSOM (OIL, SFAPI, SFTL), SSIM, NPE, FOCUS, EMSSM, XSA, ASA etc.
    b. Provide a highl-level description – we will help you with the naming convention: <app_name>-<componentname>-<function> where function can be batch, stream, restapi, soapapi
    c. Provide a high-level SWAG of the expected resource consumption in CPU/RAM/Storage
    d. When your deploying for the first time in Dev, let our build master (Grothe, Alain) know so that he can add your project in the "Artifactory image cleanup" scripts
    e. we'll create a default pull secret for you so openshift can pull your images from artifactory without any issues
        i. If it's the first time your application (ex: SSOM, NPE, FOCUS) deploys on openshift , you will need to create a functional ID for the DOQ integration (in My telecom warehouse), and log in once in artifactory with that FID.
        ii. Create a image pull secret from dockerfile and link it to the "deployer" and "default" service accounts. Here are the commands:
            1.
```
          oc create secret generic imagepull --from-file=.dockerconfigjson=./.
dockerconfigjson --type=kubernetes.io/dockerconfigjson
          oc secrets link default imagepull --for=pull
          oc secrets link deployer imagepull --for=pull
```

        iii. here is the format of the file dockerconfigjson:
            1.
```
{
    "auths": {
        "artifactory.bell.corp.bce.ca:5000": {
            "username": "yourFIDusername",
            "password": "yourFIDpassword"
        },
        "artifactory.bell.corp.bce.ca:5001": {
            "username": "yourFIDusername",
            "password": "yourFIDpassword"
        }
    }
}
```

When you deploy for the first time in DEV/UAT

1. For Any containers : (expecially If you are developping Java/Springboot), set the RAM, run with same RAM for memory request (RMEM) and limit (LMEM) - This will reduce chance of unnecessary pod restarts - ensure that your pod/container won't restart if it needs more memory but the server where openshift is running can't provide it (see related articles below).
2. Start your pod with minimal request CPU and let your pod burst
    a. set your request CPU to a very low value (<= **10** millicores)
    b. set your limit CPU to 20x or more the value of your request CPU
    c. adjust request (lower) and limit CPU (higher) with load testing results
3. Good example of load testing with different values but following the guidelines:
    a. OCP-Provisioning - MessageBroker2
    b. Even better: intregrate load test in your CI/CD pipeline, at merge request time!
4. In Deployment Config YAML, Please update imagePullPolicy to: IfNotPresent so that when pods restarts, we remove the dependency on Artifactory, leveraging internal openshift registry instead
5. Your application requires to be properly monitored - Most applications are instrumented with Dynatrace as the Bell standard toolset. That means that the "One Agent" image will need to be added as init Container. Feel free to reach to your team lead for more details /examples on how to implement but 2 important considerations:
    a. We are not licensed to rollout Dynatrace oneAgent on the openshift dev cluster - Thus, Dynatrace OneAgent should NOT be deployed on DEV/SIT (or if your pipeline requires it to be deployed, env variable DT_API_URL in your DEV deploymentConfig file should be pointing to dummy URL) as only UAT and PROD openshift cluster is licensed.
    b. Please set explicitly the memory and CPU value for the oneAgent
        i. Recommended memory values are 100Mb for MEM request and MEM limit

ii. Recommended cpu values are 10 mc for CPU Request and 250mc for CPU Limit

Before Deployment of new Production pods or projects

1. Implement Log rollout to avoid Persistent Volume filling too fast
2. Please share with the Quota admins (names listed above) - 1 month before the deployment, if possible - the expected incremental RAM/CPU /Storage consumption
3. Please ensure that pods using connections outside openshift, are using GLSB endpoints (applicable to Applications and Databases)
4. In Deployment Config YAML, Please update imagePullPolicy to IfNotPresent so that when pods restarts, we remove the dependency on Artifactory, leveraging internal openshift registry instead

Checklist for BATCH (cron / cronjob) deploying/provisionning on DEV/UAT/PROD

1. Please specify explicitly the concurrencyPolicy of your cron job
    a. you should set this property in the deployment config YAML to "Replace" (will replace previous failed job/pod with a new one - taking only 1 pod resource) or "Forbid" (if failed job, won't restart a new one) instead of "Allow" (default -)
2. For cleaner viewing in the openshift console, set successfulJobsHistoryLimit to a low value (ex: 3 to keep the last 3 successful daily job execution) and failedJobsHistoryLimit to a higher number (ex: 14 as you may need more occurence for troubleshooting)

On Going

1. If you no longer need the project, reach to your Quota Admins and ask for project and/or GIT repo deletions.
2. Help us by keeping the communication open and cleaning up your old images in Artifactory!

By not setting the concurrency policy, the default value is "Allow" which will spawn new pods on every cron job execution, therefore filling up the Quota and impacting all other apps!!

For more info, https://docs.openshift.com/container-platform/4.6/rest_api/workloads_apis/cronjob-batch-v1beta1.html

Related articles

Estimating RAM/CPU resources:

https://community.openshift.int.bell.ca/blog/89-cloud-native-application-resource-estimation-part-1-concepts

Container 101: 1. CONTAINERS 101

- Best practices for openshift containers provisionning
- OpenShift Troubleshooting - Basics
- How to clear OCP volume in terminating state
- How to request an OpenShift OAuth access token and use it with oc cli