Deploying the Denodo Solution Manager in Kubernetes

Revision 20221018

Denodo Inc.: 525 University Avenue, Suite 31, Palo Alto, CA 94301. USA www.denodo.com

CONTENTS

1 INTRODUCTION

The Denodo Solution Manager can be containerized to be run in a container platform such as Docker. The usage of containers eases the adoption of modern architectures, for instance, microservices, which can be orchestrated with Kubernetes.

This document explains how to deploy Denodo Solution Manager containers using Kubernetes.

2 DEPLOYING IN KUBERNETES

Kubernetes is an orchestration system for containers, it allows the IT teams not only to manage the deployment of containerized applications but also to scale deployments by increasing the number of replicas for the deployment. Kubernetes also allows other actions, for instance, to update the current containers with its new version released.

When working with containers and Kubernetes it is very important to take into account that containers do not persist the data by default. If we do not expect changes in our application this is not a problem, but if the container configuration changes we should take this into account. For more information see Data Persistence in Containers.

This document will make use of the Kubernetes command-line tool (`kubectl`) that interacts with the Kubernetes cluster via the Kubernetes API. This article assumes that Kubernetes and Docker are already installed and working in your local environment and that Kubernetes is enabled on Docker, as this enables a single-node cluster when Docker is started. It is also recommended to follow the Denodo Platform Container QuickStart Guide as a prerequisite for this article, as it serves to check that the Solution Manager container is working successfully in the Docker installation. The Denodo users that are new to Kubernetes can also check the Kubernetes Basics tutorial to learn the basic concepts and usage of Kubernetes and the "`kubectl`" command-line tool that communicates with Kubernetes to execute commands in the cluster.

2.1 LICENSE CONFIGMAP

The Solution Manager requires a valid license in order to start, so in order to load this license we are going to create a configmap that will embed the file. Hence, the following statement creates the map with the contents of the license file that will be referenced later from the `solution-manager-service.yaml` file:

```
$ kubectl create configmap solution-manager-license --from-file=denodo.lic=<pathToLicenseFile>
```

2.2 SOLUTION MANAGER KUBERNETES SERVICE

Once the configmap is created in the Kubernetes cluster, the Solution Manager can be started. The following YAML configuration file defines a Kubernetes Service and Deployment that will deploy the Solution Manager 8.0 container generated by Denodo in the Kubernetes cluster:

```
apiVersion: v1
kind: Service
metadata:
  name: solution-manager-service
spec:
  selector:
    app: solution-manager-app
  ports:
  - name: svc-license-manager
```

```
    protocol: "TCP"
    port: 10091
    targetPort: license-manager
  - name: svc-web
    protocol: "TCP"
    port: 19090
    targetPort: web-container
  type: LoadBalancer
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: solution-manager-deployment
spec:
  selector:
    matchLabels:
      app: solution-manager-app
  replicas: 1
  template:
    metadata:
      labels:
        app: solution-manager-app
    spec:
      hostname: solution-manager-hostname
      containers:
      - name: solution-manager-container
        image: solution-manager:8.0-latest
        command: ["/opt/denodo/tools/container/entrypoint.sh"]
        args: ["--lmserver", "--smserver", "--smadmin"]
        ports:
        - name: license-manager
          containerPort: 10091
        - name: web-container
          containerPort: 19090
        volumeMounts:
        - name: config-volume
          mountPath: /opt/denodo/conf/denodo.lic
          subPath: denodo.lic
      volumes:
      - name: config-volume
        configMap:
          name: solution-manager-license
```

solution-manager-service.yaml for Denodo 8.0

NOTE: This YAML file only applies to the Denodo Containers released with Denodo 8.0 Update 20220815 and later.

If you are using your own custom Denodo containers, you need to change both the "`command:`" and "`args:`" lines in the previous YAML files to match the requirements from your containers.

To create both elements, service, and deployment, in the Kubernetes environment save the script to a file and name it to something like `solution-manager-service.yaml`.

Then, execute the following Kubernetes command in a console:

```
> kubectl create -f solution-manager-service.yaml
```

```
C:\Denodo\Kubernetes>kubectl create -f solution-manager-service.yaml
service/solution-manager-service created
deployment.apps/solution-manager-deployment created

C:\Denodo\Kubernetes>kubectl get service,deployment,pod
NAME                                TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)                          AGE
service/kubernetes                  ClusterIP      10.96.0.1      <none>        443/TCP                          7h8m
service/solution-manager-service    LoadBalancer   10.99.76.44    localhost     10091:31070/TCP,19090:31464/TCP  4s

NAME                                          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/solution-manager-deployment   1/1     1            1           4s

NAME                                              READY   STATUS    RESTARTS   AGE
pod/solution-manager-deployment-6c48dfdbc8-r559l  1/1     Running   0          4s
```

Execution of the solution-manager-service.yaml

Although this article does not try to explain how Kubernetes works or how YAML files are created, it is interesting to outline the following from the YAML file definition:

- The service for Solution Manager 8.0 exposes two ports: 10091 and 19090. These are the ports published in the service that are mapped to ports in the container, in our example we are using the same ports in the service and the container for both 10091 and 19090.
- The configuration of the YAML file is assuming that a license is available in the configmap `solution-manager-license`. The configmap can be for instance created with the following `kubectl` statement:

  ```
  kubectl create configmap solution-manager-license
        --from-file=denodo.lic=<pathToLicenseFile>
  ```

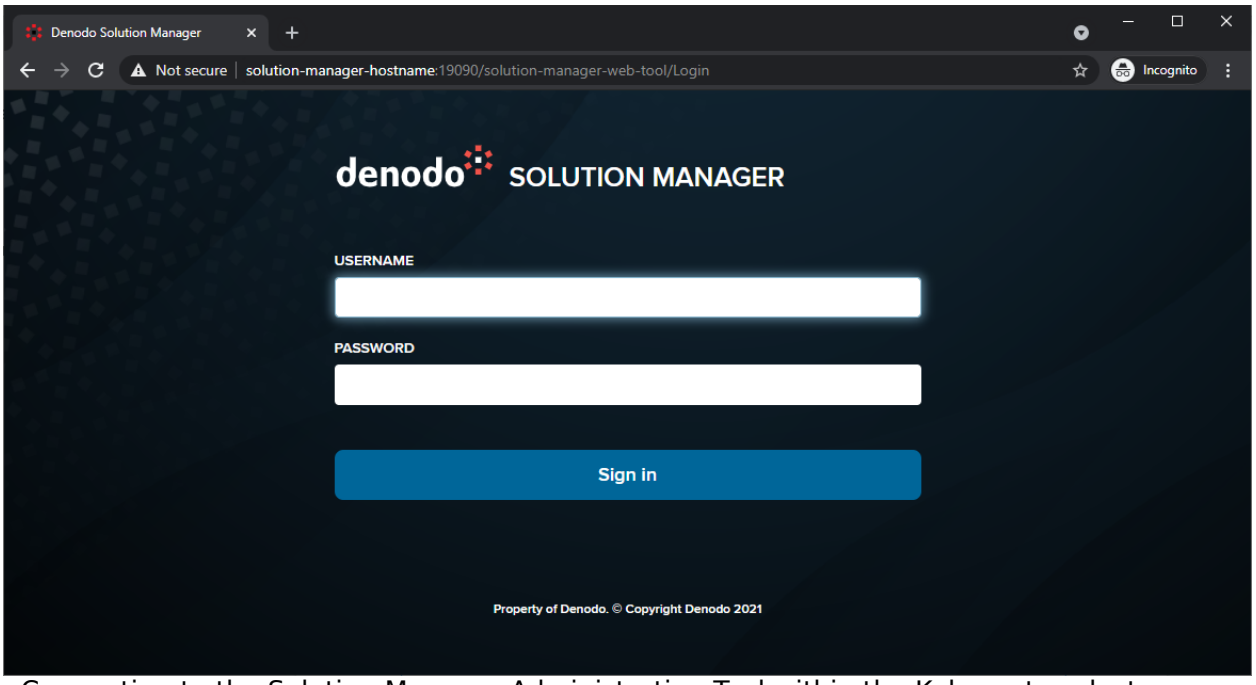
- All the Denodo pods will share the hostname *solution-manager-hostname*. Notice that the field *hostname* cannot include dots, but If needed, Kubernetes also provides a subdomain field so you can assign a FQDN to the pod.

2.3 TESTING

After the Service deployment is completed, it will be possible to connect to the new Solution Manager instance from a browser with the following HTTP URL:

```
http://solution-manager-hostname:19090/solution-manager-web-tool
```

In addition, notice that for using *solution-manager-hostname* as the connection host in your browser, it will be necessary that the client computer is able to resolve that hostname to the IP address of the *solution-manager-service*, either by defining the *solution-manager-hostname* in the DNS server or in the *hosts* file of the client computer. Please, ensure that you have network connectivity from the client computer to the Kubernetes Service, by configuring the network routes appropriately.



Connecting to the Solution Manager Administration Tool within the Kubernetes cluster

The following command will provide all the information required regarding the Kubernetes Service that is deployed:

```
> kubectl describe service solution-manager-service

Name:                     solution-manager-service
Namespace:                default
Labels:                   <none>
Annotations:              <none>
Selector:                 app=solution-manager-app
Type:                     LoadBalancer
IP:                       10.99.76.44
LoadBalancer Ingress:     localhost
Port:                     svc-license-manager  10091/TCP
TargetPort:               license-manager/TCP
NodePort:                 svc-license-manager  31070/TCP
Endpoints:                10.1.0.126:10091
Port:                     svc-web  19090/TCP
TargetPort:               web-container/TCP
NodePort:                 svc-web  31464/TCP
Endpoints:                10.1.0.126:19090
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
```

kubectl describe solution-manager-service output

Once the Solution Manager is running, you can start defining the environments of the Solution Manager so it can serve the licenses to the Denodo servers, for this you can

check the Knowledge Base article Denodo Solution Manager: First Steps.

In addition, notice that the containers are ephemeral, so in order to persist the configuration set you will need to use volumes in the YAML or to configure an external metadata database for the Solution Manager, so the configuration is stored externally.

2.4 CLEAN UP

Finally, in case that it is needed to clean the environment, the following commands can be executed to delete the configmap for the license and the Solution Manager service deployed in Kubernetes:

```
> kubectl delete -f solution-manager-service.yaml
> kubectl delete configmap solution-manager-license
```

3 REFERENCES

Denodo Platform Container QuickStart Guide
Kubernetes Documentation
Learn Kubernetes Basics
Kubernetes on Docker
Data Persistence in Containers