



CAN NETWORK FUNCTION INSTALLATION - MOP (METHOD OF PROCEDURE) VERSION FOR WINDRIVER

Cognitive Assistant for Networks (CAN) Release 5.5



JULY 6, 2021
AVANSEUS TECHNOLOGY PVT. LTD.

REVISION HISTORY

Version	Date	Change description	Created by	Updated by	Reviewed by
V 1.0	July, 2021	Initial Release	Hemanth/Umesh	Sandeep Singh	Chiranjib

Table of Contents

1. Installation of WRCP	3
2. Prerequisites	3
2.1. Istio Installation.....	3
2.2. Enabling Istio Add-Ons	4
3. Pre-Installation Steps	4
4. Creating Configmaps	5
5. Creating Persistent Volume.....	6
6. Installing Database and Importing Default Schema	7
7. Installing Helm Charts.....	7
8. Routing Traffic	8
9. Accessing the Application and Monitoring Tools from the Browser.....	9

1. Installation of WRCP

- Download WindRiver cloud platform patch 19.
- Get the Docker registry details from the WindRiver official site.
- Configure the localhost.yml file by specifying IP details, server credentials and docker registry details as shown below:

localhost.yml

```
dns_servers:
  - 8.8.8.8

external_oam_subnet: 192.168.29.0/24
external_oam_gateway_address: 192.168.29.1
external_oam_floating_address: 192.168.29.100

admin_username: admin
admin_password: wrsPWD123*
ansible_become_pass: wrsPWD123*

docker_registries:
  defaults:
    username: AKIAZDKOSK7ZIUISZSHF
    password: cLqGHHw8g3jv7rTA/o3STqP6ii5o1gArTB7B6Z/4

additional_local_registry_images:
  - docker.io/wind-river/cloud-platform-deployment-manager:WRCP_20.06
  - gcr.io/kubebuilder/kube-rbac-proxy:v0.4.0
```

- Install the downloaded WRCP ISO image and bootstrap the server with the below command:

```
ansible-playbook /usr/share/ansible/stx-ansible/playbooks/bootstrap.yml
```

- Follow the steps provided in the first-setup-wrcp.sh file to complete the WRCP installation and verify docker, kubernetes and Helm are working correctly.

2. Prerequisites

Avanseus Cognitive Assistant for Networks (CAN) is integrated with Istio service mesh to provide several features like Authorization Policy, Traffic Management, Peer Authentication Policy, Envoy sidecar Proxy Injection, Monitoring etc., Istio needs to be installed before the installation of CAN related HELM charts.

1. Istio v1.8 or higher.
2. Istio add-ons (Grafana, Kiali, Prometheus) have to be enabled.

Note that Avanseus CAN application uses MongoDB as its persistent storage. Importing the default schema into the mongo database is one of the prerequisites which will be discussed in the later section.

2.1. Istio Installation

Follow the below steps to install the Istio:

1. You can use the link as a reference to install the Istio:

<https://istio.io/latest/docs/setup/getting-started/>

2. Download the Istio installation files using the below command:

```
$curl -L https://istio.io/downloadIstio | sh - //This will download the latest version.
```

3. Move to the Istio package directory.

```
$cd istio-1.9.3
$export PATH=$PWD/bin:$PATH
```

4. Install Istio:

```
$istioctl install --set profile=demo -y
```

5. The installation of Istio is complete. To check all the pods are up, use the below commands:

```
$kubectl get all -n istio-system
```

```
controller-0:~$
controller-0:~$ kubectl get all -n istio-system
NAME                                READY    STATUS    RESTARTS   AGE
pod/grafana-94f5bf75b-jbkjr         1/1      Running   4           5d1h
pod/istio-egressgateway-668944d785-f4jvr 1/1      Running   4           5d1h
pod/istio-ingressgateway-6b47fc8-qn72p 1/1      Running   4           5d1h
pod/istiod-7b7bcbfb5-hxr7p          1/1      Running   5           5d1h
pod/jaeger-5c7675974-qj4dq          1/1      Running   6           5d1h
pod/kiali-d4fdb9cdb-q5v2k           1/1      Running   4           5d1h
pod/prometheus-7d76687994-rgkk7      2/2      Running   8           5d1h

NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)
service/grafana                     ClusterIP           10.107.242.10   <none>            3000/TCP
service/istio-egressgateway          ClusterIP           10.109.226.221  <none>            80/TCP,443/TCP,15443/TCP
service/istio-ingressgateway         LoadBalancer       10.96.135.16    <pending>         15021:32228/TCP,80:30000/TCP
```

2.2. Enabling Istio Add-Ons

To install the Istio add-ons, execute the two commands:

```
$kubectl apply -f samples/addons
$kubectl rollout status deployment/kiali -n istio-system
```

Note: If there are errors while trying to install the add-ons, try to run the command again. There may be some timing issues which will be resolved when you run the command again.

3. Pre-Installation Steps

Follow the below steps before you install the HELM charts related to CAN application:

1. Create a namespace.

```
$ kubectl create ns avanseus-workspace
```

2. Enable Istio injection on the desired namespace i.e., avanseus-workspace

```
$ kubectl label namespace avanseus-workspace istio-injection=enabled
$kubectl get namespace -L istio-injection
```

(Use this command to check if Istio-injection is enabled or not in the avanseus-workspace).

```
controller-0:~$
controller-0:~$ kubectl get namespace -L istio-injection
NAME              STATUS    AGE      ISTIO-INJECTION
avanseus-workspace Active    5d2h     enabled
cert-manager      Active    6d20h
default           Active    6d20h
istio-system      Active    5d2h     disabled
kube-node-lease   Active    6d20h
kube-public       Active    6d20h
kube-system       Active    6d20h
controller-0:~$
```

3. Installing Metric Server in the cluster.

Go to **kubernetes_resources/Gateway_Metric_HPAI** folder. Use the file named **metric_server.yaml** and execute the below command:

```
$ kubectl apply -f metric_server.yaml
```

Alternative command:

```
$ kubectl apply -f
https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
```

Use the below command to check, if metric server is running:

```
$ kubectl get all -n kube-system | grep metric
```

```
controller-0:~$
controller-0:~$ kubectl get all -n kube-system | grep metric
pod/metrics-server-6d9f9dd965-46c49          1/1      Running    4          5d2h
service/metrics-server                      ClusterIP 10.105.161.38 <none>     443/TCP
deployment.apps/metrics-server              1/1      1          1          5d2h
replicaset.apps/metrics-server-6d9f9dd965    1        1          1          5d2h
controller-0:~$
controller-0:~$
controller-0:~$
```

4. Creating Secrets

This will be used internally by HELM charts to pull Avanseus specific Docker images from the Avanseus public Docker repository.

Run the below command to create the secret. You can find the config.json file in the **Avanseus_files/Docker_secret/** folder.

```
$ kubectl create secret generic avanseus-imagepull-secret
--from-file=.dockerconfigjson=Avasnesu_files/Docker_sercret/config.json
--type=kubernetes.io/dockerconfigjson -n avanseus-workspace
```

Use below command to check if secret has been created

```
$ kubectl get secrets -n avanseus-workspace
```

4. Creating Configmaps

Go to **Avanseus_files/ConfigMap_Files/** folder and execute the below command to create the configmap as follows:

```
$ kubectl create configmap controllerconfig --from-file=controller-config/ --namespace=avanseus-workspace
$ kubectl create configmap workerconfig --from-file=worker-config/ --namespace=avanseus-workspace
$ kubectl create configmap canconfig --from-file=can-config/ --namespace=avanseus-workspace
$ kubectl get configmap -n avanseus-workspace
```

```
controller-0:~$
controller-0:~$ kubectl get configmap -n avanseus-workspace
NAME                DATA      AGE
canconfig            3          5d2h
controllerconfig     1          5d2h
istio-ca-root-cert   1          5d3h
workerconfig         1          5d2h
controller-0:~$
controller-0:~$
```

Note:

- If you have purchased a domain name (ex: <https://can.avanseus.com>) then please update domain details in the config map using `kubectl edit configmap` command.

```
$ kubectl edit configmap canconfig -n avanseus-workspace
```

- Keys such as **avanseus.app.cas.domain** and **avanseus.app.can.domain** have to be updated as "can.avanseus.com" or whichever domain name you have procured.

5. Creating Persistent Volume

Four persistence volumes have to be created for the whole application deployment.

SL No	Application name	PVC name
1	Mongodb(database)	mongopvc
2	CAN-logs	canlogpvc
3	CAN-prediction related Files	candatapvc
4	LDAP	ldappvc

Go to `Avanseus_files/Helm/` folder and execute the below commands for creating all the four persistent volume claims.

```
kubectl apply -f avanseus_pvc.yaml -n avanseus-workspace
```

Use the below code to verify whether all the 4 PVC are created:

```
$ kubectl get pvc -n avanseus-workspace
```

```
controller-0:~/Avanseus_Files/Helm$
controller-0:~/Avanseus_Files/Helm$ kubectl get pvc -n avanseus-workspace
NAME          STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
candatapvc    Bound     pvc-e39c8f9d-5e5e-401e-a7f8-5303f80d3598  10Gi       RWO             general        5d3h
canlogpvc     Bound     pvc-fd6482d4-9ad2-42e1-8a94-96061dbf21c8  10Gi       RWO             general        5d3h
mongopvc      Bound     pvc-d634681b-b5ed-4d8c-a996-4e51d4517aac  100Gi      RWO             general        5d3h
controller-0:~/Avanseus_Files/Helm$
controller-0:~/Avanseus_Files/Helm$
```

6. Installing Database and Importing Default Schema

Avanseus CAN application uses Mongo database as its backend database.

Stay in home folder and execute the below command:

```
$ helm install --name mongo Avanseus_Files/Helm/avanseus-mongodb-chart/ --namespace  
avanseus-workspace
```

After the above step, mongo database will be successfully installed. Load the database with master/mandatory tables by creating an appropriate schema.

Go to **Avanseus_file/Mongo/mongodb-linux-x86_64-3.4.6/bin/** folder and execute the following commands:

```
./mongo <ip_address_of_any_nodes>:30001/admin -u "admin" -p "Avanseus$0"  
use cangeneric  
db.createUser({user: "cangeneric", pwd: "Avanseus$0", roles: ["userAdmin", "dbAdmin",  
"readWrite"]});  
./mongorestore -h <ip_address_of_any_nodes>:30001 -u cangeneric -p '<password>' -d cangeneric  
Avanseus_file/Mongo/cangeneric/
```

7. Installing Helm Charts

There are 6 helm charts to be installed while deploying the CAN application. All the required helm charts are present under **Avanseus_files/Helm/** folder.

The sequences in which the charts have to be installed are as follows:

1. avanseus-mongo-chart
2. avanseus-ldapchart-chart
3. avanseus-pyvbi-chart
4. avanseus-workerapp-chart
5. avanseus-controllerapp-chart
6. avanseus-canmaster-chart

The Mongo chart is already installed in previous section. Proceed to install the remaining charts by executing the below commands in sequence:

```
helm install ldap avanseus-ldap-chart/ -n avanseus-workspace  
helm install vbi avanseus-pyvbi-chart/ -n avanseus-workspace  
helm install controller avanseus-controllerapp-chart/ -n avanseus-workspace  
helm install worker avanseus-workerapp-chart/ -n avanseus-workspace  
helm install can avanseus-canmaster-chart/ -n avanseus-workspace
```

Verify if all the pods have been deployed successfully using the below command.

```
$kubectl get pods -n avanseus-workspace
```

```

controller-0:~$
controller-0:~$ kubectl get pods -n avanseus-workspace
NAME                                READY   STATUS    RESTARTS   AGE
can-75899b49b7-cttqf               2/2     Running   2          2d20h
controller-65b887c556-26zhh        2/2     Running   8          5d2h
ldap-7c4d96756b-tzqq9              2/2     Running   8          5d3h
mongo-0                             2/2     Running   8          5d2h
vbl-56474d7944-bl2jn               2/2     Running   8          5d3h
worker-766b668578-8mgHW            2/2     Running   8          5d2h
controller-0:~$
controller-0:~$

```

8. Routing Traffic

By default, Avanseus CAN application uses Istio ingress controller for routing the traffic from outside of the cluster to services within the cluster. Routing rules are specified in the gateway.yaml file.

Go to Avanseus_files/redirection/ folder and execute the below command:

```
$kubectl apply -f gateway.yaml -n istio-system
```

Note:

- In WRCF Nginx ingress controller is pre-installed and it is listening to port 80. So we need to redirect all the traffic of nginx server to istio gateway. This is done by executing the below command inside Avanseus_files/redirection/ folder.

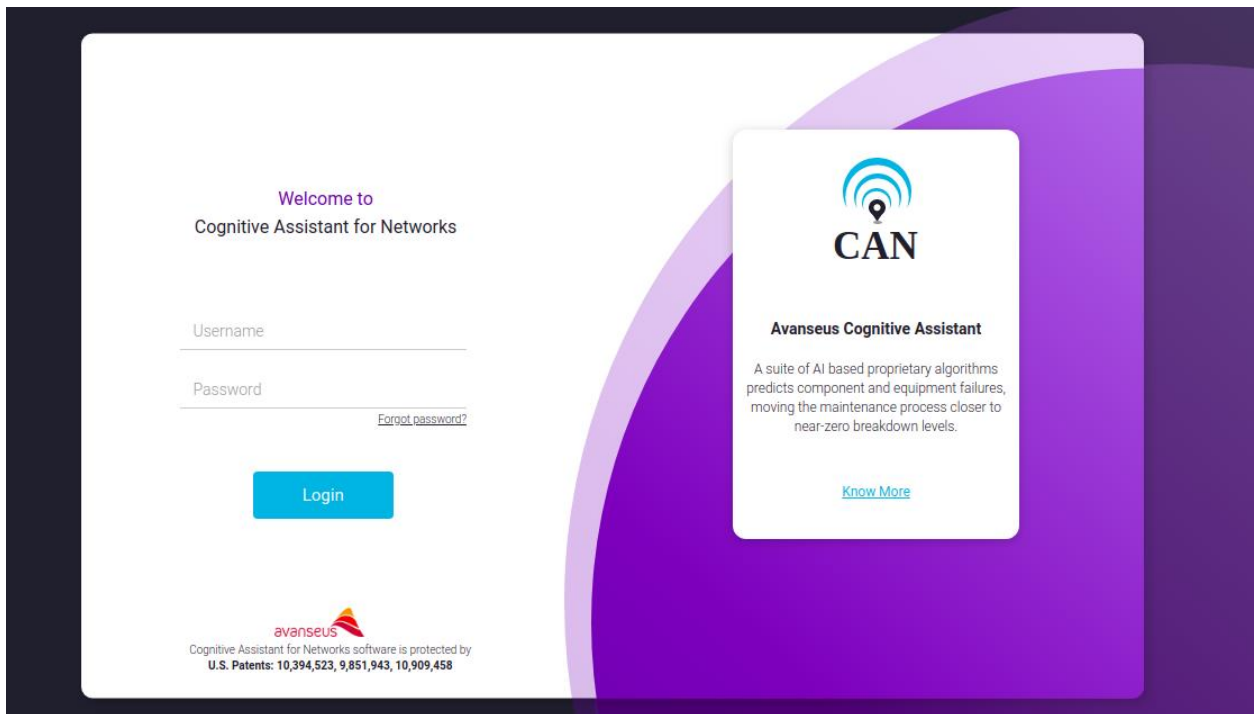
```
$kubectl apply -f redirection-service.yaml
```

- If hardware load balancer is available, then we will get an external ip for istio ingress controller which can be used for accessing the application.

9. Accessing the Application and Monitoring Tools from the Browser

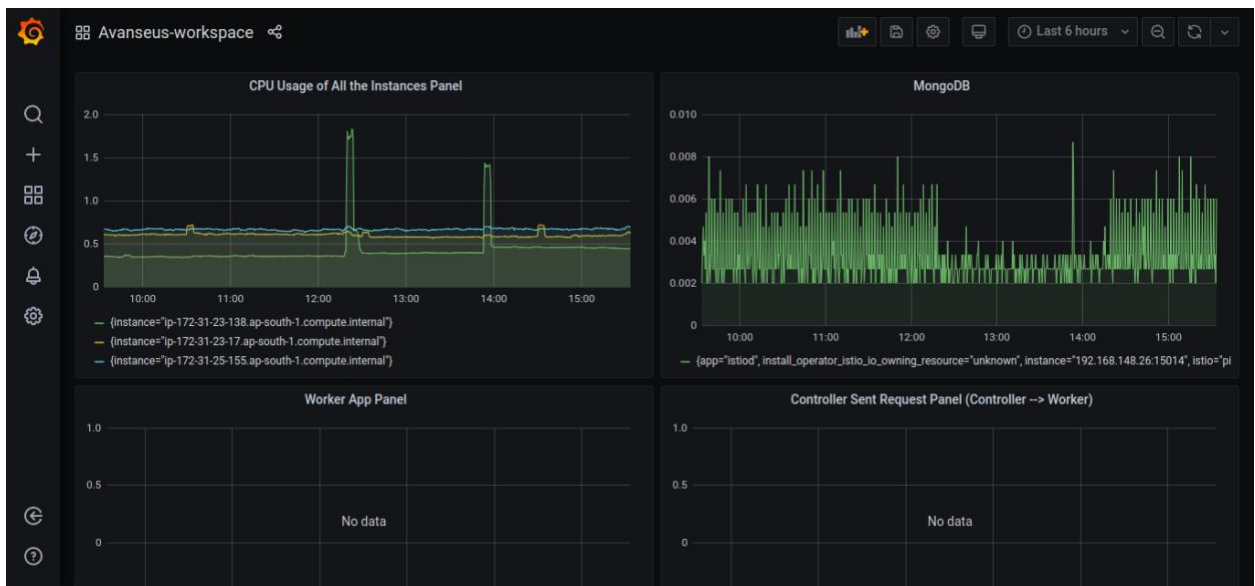
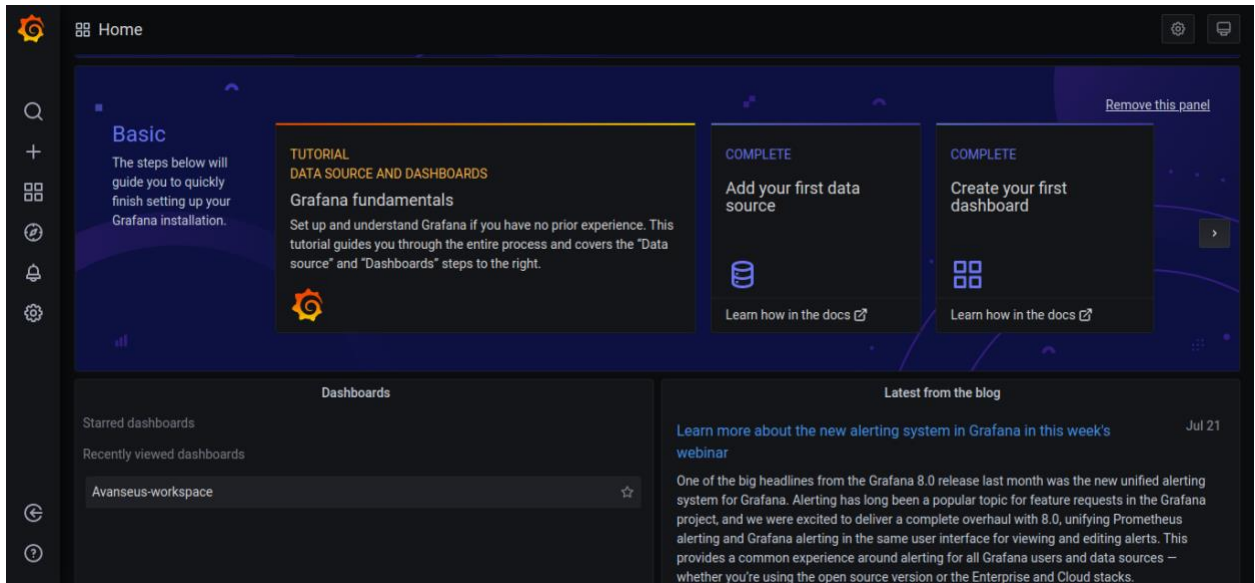
CAN Application: http://<domain_name>/CAN/

Ex: <domain_name> = 192.168.29.100 or can.avanseus.com



Grafana Dashboard: https://<domain_name>/grafana

Ex: <domain_name> = 192.168.29.100 or can.avanseus.com



Kiali Dashboard: https://<domain_name>/kiali

Ex: <domain_name> = 192.168.29.100 or can.avanseus.com

