CAN INSTALLATION AND CONFIGURATION

Cognitive Assistant for Networks (CAN) Release 5.5

JULY 6, 2021

AVANSEUS TECHNOLOGIES PVT. LTD.

Table of Contents

Copyright © 2021 Avanseus Holdings Pte. Ltd. All rights reserved.

avanseus

1. Initial Configuration Steps

1. Setup the LDAP, MongoDB, VBI, PredictionController & PredictionWorker modules using the Helm charts mentioned in the MOP documents.

2. Setup the CAN pod by configuring the “config.properties” ConfigMap correctly. i.e. IP’s & ports. See the sample content below:

```
#Domain details
avanseus.app.cas.domain=avanseuscanserver.com
avanseus.app.can.domain=avanseuscanserver.com

#VBI application info
avanseus.vbi.ip=vbicontainer
avanseus.vbi.port=12001

#Domain protocol
avanseus.protocol=https

#Host details
avanseus.app.cas.host=canapp:2003
avanseus.app.can.host=canapp:2000

#MongoDb configuration
avanseus.mongodb.ssl.enabled=true
avanseus.mongodb.validHostName=false
avanseus.mongodb.keystore.path=/data/workspace/pemfile/mongo.pkcs12
avanseus.mongodb.keystore.password=hR4YvpJGIfRH/JYilf5NOdOWEyMZlBVX

avanseus.mongodb.ip=database
avanseus.mongodb.port=27017
avanseus.mongodb.username=<username>
avanseus.mongodb.password=<password>
avanseus.mongodb.dbName=<dbname>

#Note: This configuration is needed to get admin rights.
avanseus.mongodb.admin.username=<admin_username>
avanseus.mongodb.admin.password=<admin_password>
avanseus.mongodb.admin.dbName=<admin_dbname>

#log configuration
avanseus.log.path=/data/workspace/logs/
avanseus.log.thread.poolsize=20

#LDAP server configuration
avanseus.ldap.organisation=employee
avanseus.ldap.domain=avanseus
avanseus.ldap.url1=ldap://ldap-container:1389
avanseus.ldap.url2=ldap://ldap-container:1389
avanseus.ldap.userDn=cn=Directory Manager
avanseus.ldap.password=PbcLjZLEL6xWgAt7JxDFN9IInI21gHE8

#NAS path
```


Copyright © 2021 Avanseus Holdings Pte. Ltd. All rights reserved.

```
avanseus.app.nas.path=/data/workspace/nasmount/

#Email Config
avanseus.email.fromId=can.admin@avanseus.com
avanseus.email.pwd=<password>
avanseus.email.host=<email_server>
avanseus.email.fromName=Admin
avanseus.email.port=587

#Cluster node configuration
avanseus.predictionNode.name=nodeA
```

3. Create a user in MongoDB and import the collections that are available in "MasterTables" directory of release repository. The collections and their purposes are as follows:

Entity name	Purpose
5bed7c01fd9b9d519fedd798, 5bed7c22fd9b9d519fedd79b, 5d0ce5a1cbaaab22bc8aa286, 5bed7f6afd9b9d519ffb5776, Resource & ResourceEntity	Contains default data for resource file load screen
Cause	Default alarm causes
AnnouncementRuleConfiguration	Contains sample data for Announcement rule configuration screen
AlertEmailConfiguration	Contains initial data for Notification handler screen
DefaultComplexCodeROE	Contains default configuration for ROE screen
DefaultPolicyConfiguration	Contains default policy configuration which will be used to retrieve the original state of a default policy in the policy configuration screen
DomainIcon	Contains default icon paths for default domains
Config	Generic configuration table
EmployeeAccess	Has the initial user id as 'canadmin'.
EntityFilterMaster, EntityFilterMasterQuery & EquipmentIcon	Contains default values for Cross domain correlation screen
EquipmentComponent	Default equipment components
EventFileFormat	Contains default data for Parser screen
EventFileFormatTemplate	Has master code template used for parser
ExcelPageConfiguration	Contains default data for Excel page configuration screen
ExcelPageConfigurationTemplate	Has master code template used for excel report generation
ExcelReportConfiguration	Contains default data for Excel report configuration screen
FieldLearntCauses	Contains default data of Root cause learnt from the

Copyright © 2021 Avanseus Holdings Pte. Ltd. All rights reserved.

Entity name	Purpose
	field
FileCollectionConfiguration	Contains default data for file collection screen
KafkaConfigProperties	Contains default configurations for Kafka
PerformanceCounterCause	Contains distinct KPI names for which we will run the prediction.
PostPredictionProcess	Contains default data for Post prediction process screen
PostProcessorTemplate	Has master code template used for parser's post-processor
Postprocessor	Contains default data for post processor screen
PreProcessorTemplate	Has master code template used for parser's pre-processor
PredictionFilter	Has master code logic for prediction filtration
PredictionFilterTemplate	Has master code template for prediction filtration
PredictionKeyFilter	Contains default data for prediction filter screen
PredictionFilterRule	Contains filter rules
PredictionNode	Contains default data for Prediction nodes. Default is 3 nodes as of now.
PredictedFaultToTicketMappingTemplate	Contains template needed to compile PredictionToTicketMapping code
Preprocessor	Contains default data for pre-processor screen
RemedyConfig	Contains remedy config values such as cron , mean threshold closed days
RemedyFieldValue	Contains fieldId and field names of remedy internal fields
RoeConfig	Stores default RoE configuration
RoeConfigTemplate	Has master code template for RoE configuration
RoePolicyConfiguration	Stores default policy configuration for Policy configuration screen
RoeSheetConfiguration	Stores default sheet configuration for roe sheet configuration screen
Role	Has initial role for "canadmin". Only Super Admin role would be present in Role table by default.
RootCauseAnalysis & RootCauseAnalysisEntity	Contains default data for Root cause analysis configuration screen
SplunkFields	Has splunk search job related fields and their default values
TechnicalAnalysedCauses	Contains default data of Root cause learnt from technical analysis
ThresholdConfiguration	Contains threshold type (min/max) and the threshold value for each of the KPI
VBIAnnotations	Contains necessary information for VBI module related to possible questions voice based module accepts
WeatherStatusResponseCode	Contains weather codes, weather description, color

Copyright © 2021 Avanseus Holdings Pte. Ltd. All rights reserved.

Entity name	Purpose
	code and image icon path
WebServiceConfig	Contains an entry with a threshold limit used for the maximum number of Prediction delivery API requests being served from the CAN application to the client in a day

4. Run the file with Index scripts on MongoDB available in “CAN_Indices.txt” file located in “MasterTables” directory of GIT. These will create initial set of indices required for CAN application.

5. Also update the nasmount path in few DB collections by executing the below script.

```
var nasPath = "/data/workspace/nasmount/"

db.getCollection('5d23282ebb19a8c46c88c105').update({},{$set:{"filePath":nasPath+"/CAN/Resources/RANSharedSites/Report_SSI_H3G.XLS"}}, {multi:true})
db.getCollection('5d23285dbb19a8c46c88c128').update({},{$set:{"filePath":nasPath+"/CAN/Resources/PlannedWorks/Planned 01.05-05.11.2018.xlsx"}}, {multi:true})
db.getCollection('5d232805bb19a8c46c88c0f1').update({},{$set:{"filePath":nasPath+"/CAN/Resources/LongPendingTickets/Sol.xlsx"}}, {multi:true})
db.getCollection('5d232845bb19a8c46c88c117').update({},{$set:{"filePath":nasPath+"/CAN/Resources/SitePriority/Localization_sites.xls"}}, {multi:true})
db.getCollection('FieldLearntCauses').find({}).forEach(function(e){db.getCollection('FieldLearntCauses').update({filePath:e.filePath},{$set:{filePath:nasPath+e.filePath}})})
db.getCollection('PostPredictionProcess').update({"classFilePath":"/CAN/Generated_Dir"},{$set:{"classFilePath":nasPath+"/CAN/Generated_Dir"}},{})
db.getCollection('PostPredictionProcess').update({"javaFilePath":"/CAN/Generated_Dir/postPredictionProcessUploadedFile"},{$set:{"javaFilePath":nasPath+"/CAN/Generated_Dir/postPredictionProcessUploadedFile"}},{})
db.getCollection('ResourceEntity').find({}).forEach(function(e){db.getCollection('ResourceEntity').update({filePath:e.filePath},{$set:{filePath:nasPath+e.filePath}})})
db.getCollection('RootCauseAnalysisEntity').find({}).forEach(function(e){db.getCollection('RootCauseAnalysisEntity').update({filePath:e.filePath},{$set:{filePath:nasPath+e.filePath}})})
db.getCollection('TechnicalAnalysedCauses').update({"filePath":"/CAN/RootCauseAnalysis/technicalAnalysis/SampleFileForTechnicalAnalysis.xlsx"},{$set:{"filePath":nasPath+"/CAN/RootCauseAnalysis/technicalAnalysis/SampleFileForTechnicalAnalysis.xlsx"}},{multi:true})
```

In the above script, replace the nasmount path value with respective nasmount directory of deployment environment.

6. Deploy CAN master helm chart

7. Configure the file collection & parser.

2. Manual Verification Steps (Using Utilities)

NOTE: URL Utilities mentioned below are run using the Javascript developer console in the browser after CAN login is successful. Due to security reason, we do not allow direct URL

Copyright © 2021 Avanseus Holdings Pte. Ltd. All rights reserved.

access via GET method and hence “window.redirect()” JS utility is provided in invoke POST requests to mentioned below URL utilities.

1. Run the file collection & parsing:

   URL: http://<domain>/CAN/pickupData

   In developer console, use the following after login:

   ```
   window.redirect("pickupData", {}, "_blank");
   ```

   Eg: window.redirect("pickupData", {}, "_blank");

   This step is to load the data.

2. Create the "PredictionKeyFilter" table entries for all causes having default rule (0, 0):

   URL: http://<domain>/CAN/functionalConfigurations/generateFilterKeys

   In developer console, use the following after login:

   ```
   window.redirect("functionalConfigurations/generateFilterKeys", {filterId: "<filter_id>"},
   "_blank");
   ```

   Eg: window.redirect("functionalConfigurations/generateFilterKeys", {filterId:
   "5746ab0a66bba21bf99dc004"}, "_blank");

3. Running clustering:

   URL: http://<domain>/CAN/functionalConfigurations/performCluster

   In developer console, use the following after login:

   ```
   window.redirect("functionalConfigurations/performCluster", {}, "_blank");
   ```

   Eg: window.redirect("functionalConfigurations/performCluster", {}, "_blank");

4. Running prediction:

   URL: http://<domain>/CAN/predictionService/predict

   In developer console, use the following after login:

   ```
   window.redirect("predictionService/predict", {"time": "<dd-mm-yyyy>, <dd-mm-yyyy>…" },
   "_blank");
   ```


Copyright © 2021 Avanseus Holdings Pte. Ltd. All rights reserved.

Eg: window.redirect("predictionService/predict", {"time": "12-04-2021, 13-04-2021, 14-04-2021" }, "_blank");

5. Generate the fault trace:

   URL: http://<domain>/CAN/functionalConfigurations/generateAndUpdateFaultTrace

   In developer console, use the following after login:

```
window.redirect("functionalConfigurations/generateAndUpdateFaultTrace", {"historyDays": "<days>","startDate": "<dd-mm-yyyy>", "endDate": "<dd-mm-yyyy>" }, "_blank");
```

   Eg: window.redirect("functionalConfigurations/generateAndUpdateFaultTrace", {"historyDays": 30","startDate": "11-01-2018", "endDate": "15-01-2018" }, "_blank");

   NOTE: The “endDate” parameter is optional. It may not be given, if only one day fault trace has to be updated.

6. Generate the fault history:

   URL: http://<domain>/CAN/functionalConfigurations/updateFaultHistory

   In developer console, use the following after login:

```
window.redirect("functionalConfigurations/updateFaultHistory", {"startDate": "<dd-mm-yyyy>", "endDate": "<dd-mm-yyyy>" }, "_blank");
```

   Eg: window.redirect("functionalConfigurations/updateFaultHistory", {"startDate": "11-01-2018", "endDate": "15-01-2018" }, "_blank");

   NOTE: The “endDate” parameter is optional. It may not be given, if only one day fault history has to be updated.

7. Configure the Excel report format & mailing list.

8. Generate the particular day’s report:

   URL: http://<domain>/CAN/excelReport/downloadReport

   In developer console, use the following after login:

```
window.redirect("excelReport/downloadReport", {"time": "<dd-mm-yyyy>" }, "_blank");
```

   Eg: window.redirect("excelReport/downloadReport", {"time": "12-04-2021" }, "_blank");

9. Generate RoE for an interval of dates:

Copyright © 2021 Avanseus Holdings Pte. Ltd. All rights reserved.

URL: http://<domain>/CAN/functionalConfigurations/updateRoePredictions

In developer console, use the following after login:

```
window.redirect("functionalConfigurations/updateRoePredictions", {"startDate": "<dd-mm-yyyy>", "endDate": "<dd-mm-yyyy>" }, "_blank");
```

Eg: window.redirect("functionalConfigurations/updateRoePredictions", {"startDate": "11-01-2018", "endDate": "15-01-2018" }, "_blank");

10. Generate the particular prediction window's matching report:

    URL:
    http://<domain>/CAN/predictiveFaultAnalytics/downloadConsolidatedAlarmMatchingReport

    In developer console, use the following after login:

```
window.redirect("predictiveFaultAnalytics/downloadConsolidatedAlarmMatchingReport", { windowIdentifier: "<time_in_milliseconds_of_the_first_day_of_prediction_window>" }, "_blank");
```

    Eg: window.redirect("predictiveFaultAnalytics/downloadConsolidatedAlarmMatchingReport", { windowIdentifier: "1556060000000"}, "_blank");

11. Formatted Predicted Fault table will be generated automatically after Prediction. If the table is not present in the database, then create that table manually by running:

    URL: http://<domain>/CAN/ predictiveFaultAnalytics/generateFormattedPredictedFault

    In developer console, use the following after login:

```
window.redirect("predictiveFaultAnalytics/generateFormattedPredictedFault", { startDate: "dd-MM-yyyy", endDate: "dd-MM-yyyy" }, "_blank");
```

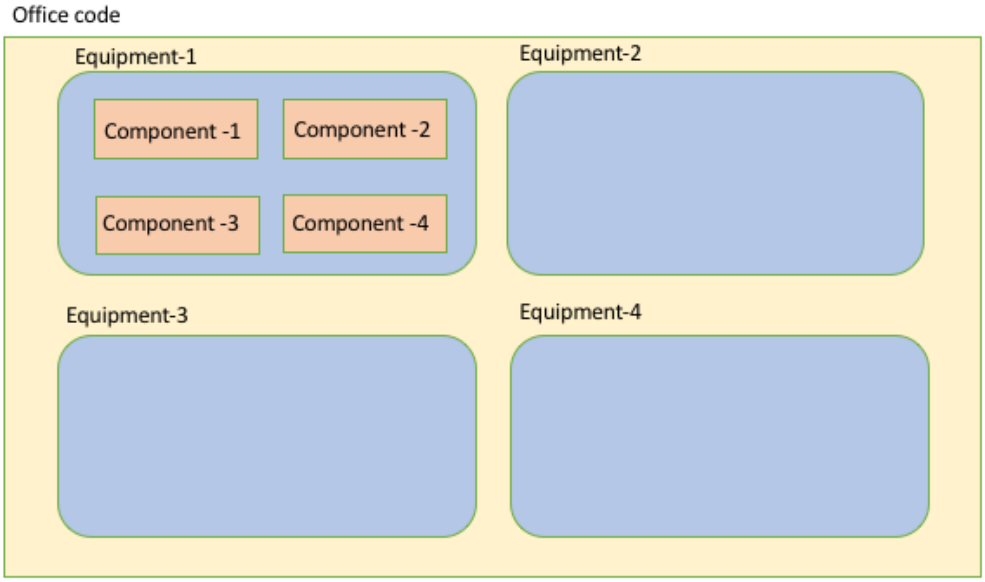
    Eg: window.redirect("predictiveFaultAnalytics/generateFormattedPredictedFault", { startDate: "29-04-2019", endDate: "30-04-2019"}, "_blank");

    NOTE: To run the above URL, please make sure that all the code snippets written in the Page Configuration screen have been compiled and saved.

3. Notes on CAN Entity Hierarchy and Their Usage

A pictorial representation of Office code and internal components is shown below.

Copyright © 2021 Avanseus Holdings Pte. Ltd. All rights reserved.



From the above picture, it is clear that a Site/Office code will have multiple equipment installed in it and each equipment will have multiple components. These components can be equipment's port, slot, rack etc.

In CAN application, we will have multiple entities that defines these Site components. The multiple entities are:

a. Office Code - DB collection would be Office Code
b. Equipment - DB collection would be Equipment
c. Equipment Component - DB collection would be Equipment Component

Above entities are made mandatory in the Parser screen for the Alarm configuration.

The table shown below contains different operations along with the Entity level.

Operation	Entity level
Prediction	Equipment component
Fault history	Equipment
Fault trace	Equipment
Clustering/Cross domain correlation	Office code
RC analysis	Equipment
ROE (Return on effort)	Equipment
Ticket history	Equipment

4. Note on Reinstallation of CAN

CAN application now runs in multiple instances in a cluster mode for high availability when accessing UI. Due to this reason, even the Cron triggers used in our application across many use

Copyright © 2021 Avanseus Holdings Pte. Ltd. All rights reserved.

cases have been made cluster aware. Quartz scheduler is now cluster aware and stores the state of each Quartz trigger in MongoDB. Whenever CAN application is completely reinstalled or installed back after a long period, there are some stale triggers in the database which may trigger the moment the application comes up. This is good since the application needs to be updated with the latest data. However there may be cases where the trigger is very old and you may not require that to run. For this, please drop the below mentioned collections from the database and then start the application.

- quartz_calenders
- quartz_jobs
- quartz_locks
- quartz_triggers

Copyright © 2021 Avanseus Holdings Pte. Ltd. All rights reserved.