Professional (L3) Unified Inventory R24 Engineer Certification

Practical Work Assignment
To be completed by the Candidate

Certification Track ID	GS02-UIV-R24-S31000-F
Certification Track name	Professional (L3) Unified Inventory R24 Engineer Certification
Name of candidate	<To be filled by Candidate>
Candidate Nokia ID	<To be filled by Candidate>
Mail address of candidate	<To be filled by Candidate>
Submission date (dd-mm-yy)	<To be filled by Candidate>

[TypeDocIDHere]

Contents

[TypeDocIDHere]

1   Introduction

This practical exercise is a **<u>mandatory</u>** step on the way to get certified as a L3 engineer.

The candidate must work **<u>independently</u>** to answer all questions as guided in this document and ensure sufficient details are provided as guided in the question.

Candidates will be asked to confirm the following when submitting their completed assignment:

I confirm that the work assignment I am submitting is entirely my own work and that I have not copied, used another person's work, or worked with others inappropriately to complete this work assignment.

Any suspicion of deviation will be passed to the individuals LM and HR Business Partner and in case of evidence of violation, the case will be reported to <u>Ethics & Compliance</u>.

All answers must be recorded in this document in the sections marked in **green** and the completed document should be submitted for evaluation as guided below.

Please follow instructions for Assignment Upload in Nokia Knowledge Center, here

Once your work assignment has been submitted then update your curriculum by clicking on the "Notify Observer" – button in NLDH (Nokia Learning & Development Hub).

Assignments usually take 1 week to evaluate.

<u>Do you need support?</u>

- For Certification **Technical** support queries, please use the <u>supporting community in NLDH</u> which is mentioned in the description of the curriculum. The community is monitored by the SME.
- For any queries, use the <u>Jira Service Management (JSM) system</u>.

This document describes the exercises that need to be completed after successful completion of the **Professional (L3) Unified Inventory R24 Engineer Certification** training content. The assignment is structured such that it is taken from real life cases and combined in a way to give the candidates good insight in their day-to-day work. The tasks will focus on practical work assignment and shall be handled like in real customer case situation.

[TypeDocIDHere]

There will be on each chapter an overall description/explanation of the requirements and then there is separately written what exactly is asked as a deliverable from you as candidate. You only need to answer and deliver what has been specifically asked in the requirement section.

2   Lab Access Information

For some certification programs (e.g., engineer programs), you may need access to lab facilities to be able to complete the exercises. If this is the case, then this section will describe how to gain remote access to the supporting lab environment.

Typically, access will be granted for a period of 6 days (Monday to Saturday). Afterwards (on Sunday) an automatic clean-up will be performed to get the environment ready for new trainees. Therefore, please make sure that you can fully concentrate on these exercises during this period.

Note: Due to limited Lab resources the number of vouchers to access our Lab is limited per week.

How to get a voucher to access the lab? Please follow instructions <u>here</u>.

Be aware that the credentials provided will only be valid on the 6 specified days on the voucher.

3 Exercises

3.1 IHUB Use-Case

To develop a cytoscape topology based on the attached LLD (Sample_LLD.docx).

Sample_LLD.docx

3.1.1 Important Links Reference

These Links can be referred later in the document for use-case Implementation.

	DESCRIPTION	LINK
Jenkins	Jenkins link to create the build pipeline.	https://build14.cci.nokia.net/job/NSWPS/job/BADOP/job/BADOPALLTTEUIVKMCUIV/
Gerrit	Repository Link to push your code.	https://gerrit.ext.net.nokia.com/gerrit/admin/repos/NSWPS/BADOP/BADOPALLTTEUIVKMCUIV
SDK	For downloading the required SDK.	https://repo3.cci.nokia.net/artifactory/uiv-mvn-releases/uiv-sdk/

Follow below steps to finish the hands-on exercise for the use case.

3.1.2 DOR (Definition of Readiness)

This section defines the prerequisites necessary to commence the implementation of the use case. Should any of the items specified as part of the Definition of Ready (DOR) be absent, implementation should not proceed.

REQUIREMENT	DESCRIPTION	COMMENT
LLD	LLD Document	It captures initial requirement based on which

		implementation can be started
Lab Details	Lab URL and Credentials	Details of the environment where IHUB project can be deployed, and testing can be performed.
Minio Details	Lab URL and Credentials	Details of the Minio server where the project jars can be uploaded.
Lab Access	Host Entry/ IP Whitelisting/ Project specific VPN Access	Details on how to access the lab environment, via VPN/Firewall Login, Host file entry or Whitelisting.
Openshift/AWS/NCS details	Required for checking pod status	Either the access or superadmin credentials.
Repository Details	Repository URL	Details for repo where code can be checked in.
Repository Access	Read/ Write Access for the repository	Required to push and pull the changes in the repository
Jenkins Pipeline	Jenkins Pipeline URL	Details for pipeline for the build generation
Jenkins Access	Read/ Write Access for the Jenkins.	Required to create a build pipeline for IHUB projects

3.1.3 Environment Setup

Follow below steps to setup the environment:

1. As part of environment setup make sure to download java and maven. Install Java JDK version 11.0.8 or higher and Apache Maven version 3.5.1 or higher in your machine. In case of UIV24 install JAVA version 17
2. Based on the UIV version corresponding IHUB SDK can be downloaded from the SDK link provided in the important links section.
3. Setting file at "C:\Program Files\apache-maven-<version>\conf" can be updated or below attached file can be used.

settings.xml

Note – Attached settings.xml file works till UIV sdk 23

4. IHUB SDK downloaded can be installed using the following command: *java -jar ihub-cli -i*

3.1.4 Implementation

Follow below steps for the use case implementation:

i) Generate the project from the SDK as per the use case.
ii) Implement the project based on the Sample_LLD.docx.
iii) Make sure to commit the code in *Gerrit Repository* on regular basis in **IHUB** Branch.
iv) Make sure to follow the below folder structure in Gerrit while pushing the code: <nsnId>_<ProjectName>

3.1.5 Deployment

Follow below steps for the use case deployment:

i) Create a Jenkins pipeline (Duplicate name for pipeline is not allowed, so make it with a unique name). A *Sample Jenkins* pipeline is provided at the same location for reference.
ii) Build the project using the Jenkins pipeline.

Sample Jenkins File –

Jenkinsfile.zip

iii) Upload the generated jar to the appropriate bucket in minio.
iv) Onboard and apply the jar using the API.
v) Check for the pods, all the pods should be up and running.

3.1.6 Unit Test

Create a UT report for the deployed artifact.

[TypeDocIDHere]

Sample UT Report:

CFS_TOPOLOGY_UT.
xlsx

3.1.7   Use-Case Specific Exercise

QUESTION: Name the bucket in which the artifact will be uploaded in minio?

ANSWER:

QUESTION: Share the TopologyPayload.json file.

ANSWER:

QUESTION: Share the globalConfig.json file.

ANSWER:

[TypeDocIDHere]

QUESTION: Share the screenshot of the generated topology view.

ANSWER:

3.2 UIV Custom Action Use-Case

To develop a CreateVLANPool Custom Action based on the attached LLD (SampleLLDCustomAction.docx).

SampleLLDCustomAc
tion.docx

3.2.1 Important Links Reference

These Links can be referred later in the document for use-case Implementation.

	DESCRIPTION	LINK
Jenkins	Jenkins link to create the build pipeline.	https://build14.cci.nokia.net/job/NSWPS/job/BADOP/job/BADOPALLTTEUIVKMCUIV/
Gerrit	Repository Link to push your code.	https://gerrit.ext.net.nokia.com/gerrit/admin/repos/NSWPS/BADOP/BADOPALLTTEUIVKMCUIV
SDK	For downloading the required SDK.	https://repo3.cci.nokia.net/artifactory/uiv-mvn-releases/uiv-sdk/

Follow below steps to finish the hands-on exercise for the use case:

3.2.2 DOR (Definition of Readiness)

This section defines the prerequisites necessary to commence the implementation of the use case. Should any of the items specified as part of the Definition of Ready (DOR) be absent, implementation should not proceed.

REQUIREMENT	DESCRIPTION	COMMENT
LLD	LLD Document	It captures initial requirement based on which implementation can be started
Lab Details	Lab URL and Credentials	Details of the environment where Custom Action project can be deployed,

		and testing can be performed.
Minio Details	Lab URL and Credentials	Details of the Minio server where the project jars can be uploaded.
Lab Access	Host Entry/ IP Whitelisting/ Project specific VPN Access	Details on how to access the lab environment, via VPN/Firewall Login, Host file entry or Whitelisting.
Openshift/AWS/NCS details	Required for checking pod status	Either the access or superadmin credentials.
Repository Details	Repository URL	Details for repo where code can be checked in.
Repository Access	Read/ Write Access for the repository	Required to push and pull the changes in the repository
Jenkins Pipeline	Jenkins Pipeline URL	Details for pipeline for the build generation
Jenkins Access	Read/ Write Access for the Jenkins.	Required to create a build pipeline for Custom Action projects

3.2.3   Environment Setup

Follow below steps to setup the environment:

5. As part of environment setup make sure to download java and maven. Install Java JDK version 11.0.8 or higher and Apache Maven version 3.5.1 or higher in your machine. In case of UIV24 install JAVA version 17
6. Based on the UIV version corresponding UIV SDK can be downloaded from the SDK link provided in the important links section.
7. Setting file at "C:\Program Files\apache-maven-<version>\conf" can be updated or below attached file can be used.

settings.xml

Note – Attached settings.xml file works till UIV sdk 23

8. UIV SDK downloaded can be installed using the following command: *java -jar uiv-cli -i*

3.2.4 Implementation

Follow below steps for the use case implementation:

1. Generate the project from the SDK as per the use case. Custom Action Provided is a number management use case so generate the number management project.
2. Implement the project based on the *SampleLLDCustomAction.docx.*
3. Make sure to commit the code in *Gerrit Repository* on regular basis in **UIV_Custom_Actions** Branch.
4. Make sure to follow the below folder structure in Gerrit while pushing the code –

   <nsnId>_<ProjectName>

3.2.5 Deployment

Follow below steps for the use case deployment:

1. Create a Jenkins pipeline (Duplicate name for pipeline is not allowed, so make it with a unique name). A *Sample Jenkins* pipeline is provided at the same location for reference.
2. Build the project using the Jenkins pipeline.
3. Sample Jenkins File:

   Jenkinsfile

4. Upload the generated jar to the appropriate bucket in minio.
5. Onboard and apply the jar using the API.
6. Check for the pods, all the pods should be up and running.

3.2.6 Unit Test

Create a UT report for the deployed artifact. Sample UT Report:

CreateVLANPool_UT.
docx

3.2.7 Use-Case Specific Exercise

QUESTION: Name the bucket in which the Integration artifact will be uploaded in minio?

ANSWER:

QUESTION: Share the Action – CreateVLANPool.java file.

ANSWER:

QUESTION: Share the Request Parameters Considered for the Trigger

ANSWER:

QUESTION: Please share the Postman request created for the action trigger, ensuring that both the request and response details are visible after the action is triggered.

ANSWER:

QUESTION: Share the Output Response of the Trigger

ANSWER:

QUESTION: Attach the Screenshot From IHUB UI of the VLAN Pool Created with all the attributes and properties.

ANSWER:

[TypeDocIDHere]

3.3 Jasper Use-Case

To develop a report based on the attached LLD (Jasper LLD.docx).

Jasper LLD.docx

3.3.1 Important Links Reference

These Links can be referred later in the document for use-case Implementation.

	DESCRIPTION	LINK
Gerrit	Repository Link to push your code.	https://gerrit.ext.net.nokia.com/gerrit/admin/repos/NSWPS/BADOP/BADOPALLTTEUIVKMCUIV
License	For designing reports in Jasper studio.	https://confluence.ext.net.nokia.com/display/GSDO/JasperReports+-+Ramp+up?preview=/1716013373/1818639390/jasperserver.license

Follow below steps to finish the hands-on exercise for the use case:

3.3.2 DOR (Definition of Readiness)

This section defines the prerequisites necessary to commence the implementation of the use case. Should any of the items specified as part of the Definition of Ready (DOR) be absent, implementation should not proceed.

S.No	ITEM	DESCRIPTION
1	Baselined LLD	Nokia Reviewed and Approved LLD is provided. LLD Walkthrough will be provided by the Project Team Ensure LLD doc is latest one and contains all the details. - Associations between entities is clearly mentioned. - UIV Properties clearly mentioned and how to fetch (Traversal Path).

2	Sample Data	Data should be available in Ihub
3	VPN connectivity	VPN Access to the the identified by Team members on Day-1. Client should provide that as requested.
4	Dev Environment details and no connectivity issues	Jasper server URL is provided by them Credentials to connect to IHUB,CSFP Neo4j credentials for the respective environment for testing queries Jasper licence need to be provided by client
5	JIRA EPIC	Jira EPIC with initial estimates
6	GIT link and access permissions	GIT link for code checkin.

3.3.3   Environment Setup

Follow below steps to setup the environment:

1. As part of environment setup please refer the jasper ramp up confluence page where you can find the TIBCO Jaspersoft Studio.exe file.
2. Please download the file and install it in your system.
3. Get the appropriate jasper license from the Ramp up page and follow the steps provided there to install it.
4. Establish the connection between the Jasper studio to the appropriate Jasper server and   the test the connection.

3.3.4   Implementation and Deployment

Follow below steps for the use case implementation:

1. Understand the requirement and start to write the neo4j query and test it in the corresponding CSF platform by logging into the neo4j pods.
2. Once the appropriate query is formed then start to make the design changes in the Jasper studio for which you can refer the Jasper Ramp up page.

[TypeDocIDHere]

3. After updating the query in query dialog box and rendering the appropriate design , save the changes.
4. By saving the changes itself, the file will be updated in the server due to establishment of connection between Jasper studio and Jasper server.

3.3.5   Code Backup

As part of Code backup,

1.   Export the jrxml file from Jasper server to your local system in the zip format.
2.   Upload the exported file in the provided git repository for later reference and backup purposes.

3.3.6   Use-Case Specific Exercise

QUESTION: Establish the connection between the Jasper studio and Jasper Server, Provide the screenshot of the steps

ANSWER:
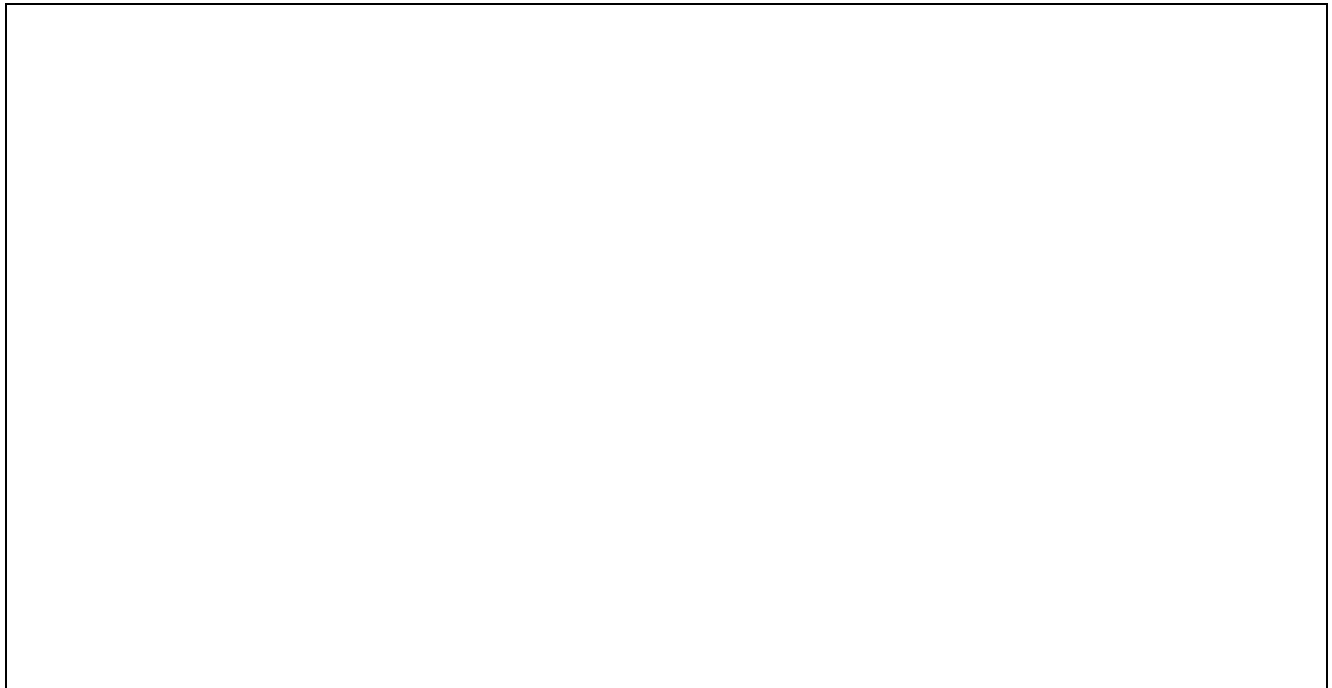
QUESTION: Provide the neo4j query for the above use-case

ANSWER:

QUESTION: Provide the designer tool (Jasper studio) image for the provided use-case.

ANSWER:

QUESTION: Provide the output of the report on the Jasper server.

ANSWER: