

PLAN DE TESTS

Ref. document VISHNU_D2_1b-FMS-PlanTests
Objet du document Référentiel du plan de tests du module FMS du projet Vishnu

TABLE DES REVISIONS

Version	Date	Modifications	Auteur
V1.0	25/03/11	Version initiale	Ibrahima Cissé, Daouda Traoré, Benjamin Isnard, Eugène Pamba Capo-Chichi
V1.1	15/12/11	Prise en compte de l'expression des besoins d'évolution de VISHNU : - ajout du test F1.CR2-B2:Create New directories et F1.CR2-E4: Create new directories	Ibrahima Cissé, Daouda Traoré
V1.2	21/02/12	Correction du nom de la commande dans F1.DI4-B:Display content of directories	Eugène Pamba Capo-Chichi

VALIDATION DU DOCUMENT

Date	Personne

REFERENCES

VISHNU_D2_1b-FMS-PlanTests	Politique de tests du module FMS
----------------------------	----------------------------------

Ref. document	Description
VISHNU_D1_1e-POL-TESTS	Politique de tests du projet VISHNU
VISHNU_D1_1a-SPECSGEN	Spécifications générales du projet VISHNU
VISHNU_D1_1b-STB	Spécifications Techniques des besoins
VISHNU_D1_1g-ARCH	VISHNU Technical Architecture

1 Présentation du document

1.1 Objectifs du document

Les objectifs de ce document sont les suivants :

- Présenter les fonctionnalités du module FMS à tester.
- Présenter les différents types de tests permettant de valider les cas d'utilisation de chaque fonctionnalité du module FMS.
- Présenter les méthodes et outils utilisés pour ces tests.
- Donner des détails de l'environnement de test

1.2 Structure du document

Le chapitre 1 présente le document.

Le chapitre 2 présente les environnements et outils de tests.

La chapitre 3 décrit la liste des tests qui s'appliquent aux fonctionnalités à tester.

La chapitre 4 présente les outils utilisés pour ces tests.

1.3 Glossaire, abréviations et acronymes

1.3.1 Glossaire

- **Script de test** : c'est un programme qui contient les instructions permettant de réaliser les tests.
- **Certificat** : correspond à la clé de session (identifiant de session crypté)

1.3.2 Acronymes

- STB : Spécification Technique des Besoins
- DSG : Document de Spécification générale
- DSD : Document de Spécification Détaillées
- API : Application Programming Interface (Interface de programmation d'applications)
- MPI : Message Passing Interface

1.3.3 Abréviations

- Id. Identifiant

- Réf. Référence
- Rel.# Numéro de release/version
- Resp. Responsable(s)

2 Environnements et outils de tests

2.1 Environnements de tests

Pour ces tests, nous allons utiliser les deux environnements de tests suivants avec des systèmes dédiés aux tests:

Environnement 1 :

- Système d'exploitation : c'est le système Linux décrit dans [STB] qui sera installé sur toutes les machines clientes et serveurs.
- Nombre de machines clientes : une machine cliente décrite dans [STB] pour pouvoir lancer des tests à distance (selon le cas de test).
- Nombre de machines serveurs : un serveur FMS décrit dans [ARCH] .
- Un serveur SysFera-DS dédié décrit dans [ARCH].
- Mode de transfert : scp

Environnement 2 :

- Système d'exploitation : c'est le système Linux décrit dans [STB] qui sera installé sur toutes les machines clientes et serveurs.
- Nombre de machines clientes : une machine cliente décrite dans [STB] pour pouvoir lancer des tests à distance (selon le cas de test).
- Nombre de machines serveurs: un serveur TMS décrit dans [ARCH]
- Un serveur SysFera-DS dédié décrit dans [ARCH].
- Mode de transfert : rsync

2.2 Outils de tests

Les outils suivants seront utilisés pour les tests :

- Hudson (serveur d'intégration) sera utilisé pour les tests d'intégration continue (compilation, installation).
- Boost test sera utilisé pour la réalisation des tests unitaires.
- Des scripts batchs seront utilisés pour la réalisation des scripts de tests, des tests fonctionnels.
- Square et Logiscope seront utilisés régulièrement pour la qualité du code.
- Valgrind sera utilisé pour la détection des fuites mémoires.

3 Cycle de vie du test

Le schéma suivant donne une vue du plan de test sur un test à réaliser jusqu'à la réalisation du

rapport de test :

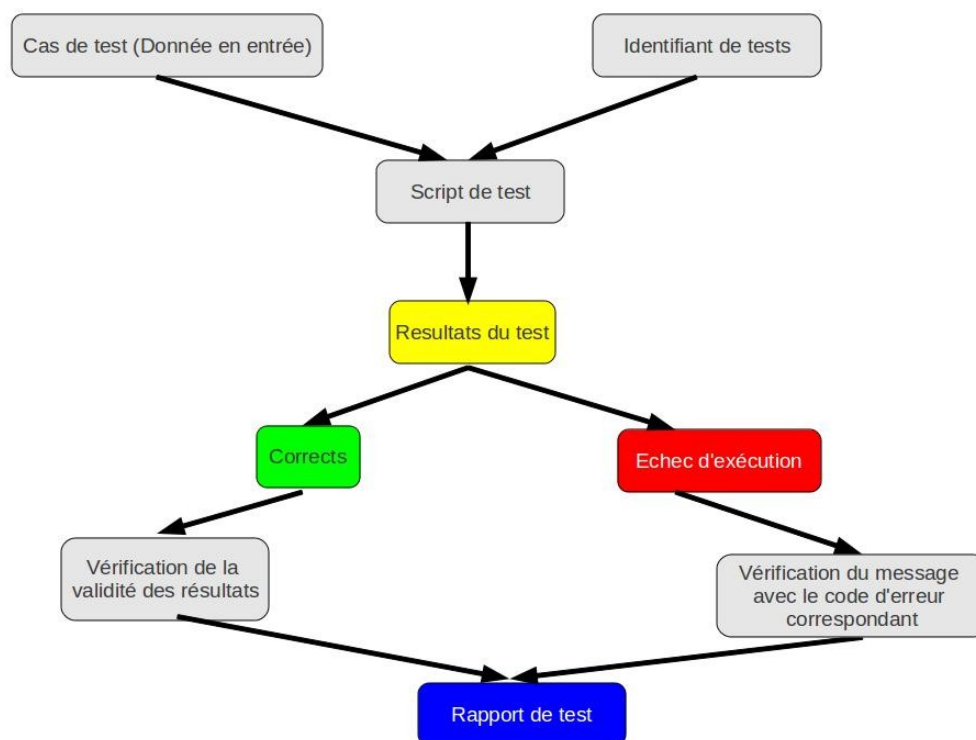


Illustration 1: Schema du test

4 Liste des tests

4.1 Tests fonctionnels

Les tests fonctionnels ou tests de boîtes noires permettent à partir des données en entrée d'observer les réactions du logiciel à tester.

Les fonctionnalités sont classées par catégories de tests :

- Catégorie 1 : regroupe les fonctionnalités permettant la création et suppression de fichiers qui sont : *createFile*, *createDir*, *removeFile*, *removeDir*.

- Catégorie 2 : regroupe les fonctionnalités permettant la consultation des propriétés d'un fichier/répertoire sont : `listDir`, `headOfFile`, `tailOfFile`, `contentOfFile`, `getFilesInfo`.
- Catégorie 3 : regroupe les fonctionnalités permettant la modification des propriétés d'un fichier/répertoire qui sont : `chmod`, `chGrp`.
- Catégorie 4 : regroupe les fonctionnalités permettant la gestion des transferts de fichiers qui sont : `copyFile`, `copyAsyncFile`, `moveFile`, `moveAsyncFile`, `stopFileTransfer`, `listFileTransfers`.

Chaque test fonctionnel sera réalisé en utilisant chacune des interfaces d'accès suivantes : fonctions API/C++ et API/Python, ligne de commande utilisateur et API web services.

Nous allons dans cette section présenter les différentes stratégies de tests en utilisant des programmes exécutables ou des fonctions de l'API FMS puis en observant les résultats obtenus. Les deux conditions à vérifier sont :

- Est ce que le programme réalise les fonctions attendues en accord avec les spécifications?
- Est ce que le programme fonctionne correctement ?

4.1.1 Identifiant de tests

Les identifiants des tests sont de la forme: <ID DSG>-[B|E]:<NOM_TEST>-<INTERFACE> où 'ID DSG' est l'identifiant du cas d'utilisation dans le document [SDG] , 'B' la séquence de base (exécution du cas d'utilisation avec des paramètres valides), 'E' la séquence d'exception (comportement du cas d'utilisation en cas d'erreurs), 'NOM_TEST' le nom du test décrit dans [SDG] et 'INTERFACE' les différents codes des interfaces utilisées par VISHNU : fonctions API/C++(CPP), API/Python (PY), ligne de commande utilisateur (CLI) et API web services (WS). L'absence du code de l'interface signifie que le test s'applique à toutes les interfaces précédentes.

Il faut noter que les tests avec des paramètres corrects sont effectués dans une session active avec un mode de fermeture à la déconnexion (CLOSE_ON_DISCONNECT) et un certificat de session valide.

4.1.2 Tests de la Catégorie 1

Identifiant de tests	Actions	Résultats	Considérations particulières
F1.CR1-B:Create New files	<p>Cas de test : création du fichier file1 dans le répertoire /home/rep1 de la machine d'identifiant MA_1 en donnant les droits de lecture à tous et les droits d'écriture en plus pour le propriétaire (644 en octal). Supposons que les paramètres file1, /home/rep1 et MA_1 sont corrects. Supposons que /home/rep1 existe dans la machine d'identifiant MA_1 et ne contient aucun fichier de nom file1.</p> <ol style="list-style-type: none"> 1. Lancer la commande vishnu_create_file avec comme paramètres MA_1:/home/rep1/file1 et 644. 2. Le fichier file1 est créé dans le répertoire /home/rep1 et il possède des droits de lecture et écriture pour le propriétaire et des droits de lecture uniquement pour les autres. 	-Un fichier nommé file1 est créé dans le répertoire /home/rep1 avec des droits de lecture et d'écriture pour le propriétaire et des droits de lecture uniquement pour les autres.	L'utilisateur doit avoir les droits d'écriture et d'exécution sur le répertoire rep1.

	<p>autres.</p> <p>3. Lancer la commande vishnu_list_dir avec comme paramètres MA_1:/home/rep1 et comme option -l et vérifier que le fichier file1 a bien été créé avec les droits spécifiés. La création du fichier a bien été validée.</p>		
<p>F1.CR1-E1: Create new files</p> <p>F1.CR1-E2: Create new files</p> <p>F1.CR1-E3: Create new files</p>	<p>1. Cas de test : paramètres non valides</p> <ul style="list-style-type: none"> ● Cas E1 : le chemin du fichier est invalide ● Cas E2 : il manque des droits d'accès pour la création du fichier ● Cas E3 : l'identifiant machine est invalide <p>2. Lancer la commande vishnu_create_file avec le cas de test correspondant (E1, E2 ou E3)</p> <p>3. Vérifier que son exécution échoue avec le message d'erreur renvoyé.</p>	- Un message correspondant au cas d'erreur est affiché	
F1.CR2-B:Create New directories	<p>Cas de test : Création du répertoire rep2 dans le répertoire /home/rep1 de la machine d'identifiant MA_1 en donnant les droits de lecture à tous et les droits d'écriture en plus pour le propriétaire (644 en octal). Supposons que les paramètres rep1, /home/rep2 et MA_1 sont corrects. Supposons aussi que /home/rep1 existe dans la machine d'identifiant MA_1 et ne contient aucun répertoire de nom rep2.</p> <p>1. Lancer la commande vishnu_create_dir avec comme paramètres MA_1:/home/rep1/rep2.</p> <p>2. Le répertoire rep2 est créé dans le répertoire /home/rep1 et il possède des droits de lecture et écriture pour le propriétaire et des droits de lecture uniquement pour les autres.</p> <p>3. Lancer la commande vishnu_list_dir avec comme paramètres MA_1:/home/rep1 et comme option -l et vérifier que le répertoire rep2 a bien été créé avec les droits spécifiés.</p> <p>4. Lancer la commande vishnu_create_file /home/rep1/rep2/file1 et vérifier que rep2 contient bien le fichier file1. La création du répertoire a bien été validée.</p>	-Un répertoire nommé rep2 est créé dans le répertoire /home/rep1 avec des droits de lecture et d'écriture pour le propriétaire et des droits de lecture uniquement pour les autres.	L'utilisateur doit avoir les droits d'écriture et d'exécution sur le répertoire rep1 .
F1.CR2-B2:Create New directories	<p>Cas de test : Création récursive de l'arborescence /home/rep1/rep2 en supposant que les répertoires rep1 et rep2 n'existent pas sur la machine d'identifiant MA_1.</p> <p>1. Lancer la commande vishnu_create_dir avec comme paramètres MA_1:/home/rep1/rep2, et en spécifiant</p>	-Les répertoires nommé rep1 et rep2 sont créés et rep2 est créé dans le répertoire /home/rep1 .	

	<p><i>l'option isRecursive.</i></p> <ol style="list-style-type: none"> Les répertoires rep1 et rep2 sont créés et rep2 existe bien dans rep1. Lancer la commande vishnu_list_dir avec comme paramètres MA_1:/home/rep1 et vérifier que le répertoire rep2 a été créé dans rep1. Lancer la commande vishnu_create_file /home/rep1/rep2/file1 et vérifier que rep2 contient bien le fichier file1. La création du répertoire a bien été validée. 		
<p>F1.CR2-E1: <i>Create new directories</i></p> <p>F1.CR2-E2: <i>Create new directories</i></p> <p>F1.CR2-E3: <i>Create new directories</i></p> <p>F1.CR2-E4: <i>Create new directories</i></p>	<p>Cas de test : paramètres non valides</p> <ul style="list-style-type: none"> Cas E1 : le chemin du répertoire est invalide Cas E2 : il manque des droits d'accès pour la création du répertoire Cas E3 : l'identifiant machine est invalide Cas E4 : création récursive des répertoires avec une arborescence donnée et l'option isRecursive est omise. <ol style="list-style-type: none"> Lancer la commande vishnu_create_dir avec le cas de test correspondant (E1, E2 ou E3 ou E4) Vérifier que son exécution échoue avec le message d'erreur renvoyé. 	- Un message correspondant au cas d'erreur est affiché	
F1.DE1-B : <i>Delete files</i>	<p>Cas de test : Suppression du fichier file1 du répertoire /home/rep1 de la machine d'identifiant MA_1. Supposons que les paramètres file1, /home/rep1 et MA_1 sont corrects. Supposons également que le répertoire /home/rep1 existe dans la machine d'identifiant MA_1 et qu'il contient un fichier nommé file1.</p> <ol style="list-style-type: none"> Lancer la commande vishnu_create_file avec comme paramètres MA_1:/home/rep1/file1 Lancer la commande vishnu_list_dir avec comme paramètres MA_1:/home/rep1 et vérifier que le fichier file1 a bien été supprimé. La suppression du fichier a bien été validée. 	- Le fichier file1 est supprimé du répertoire /home/rep1 sur la machine d'identifiant MA_1.	L'utilisateur doit avoir les droits d'écriture et d'exécution sur le répertoire rep1.
<p>F1.DE1-E1: <i>Delete files</i></p> <p>F1.DE2-E2: <i>Delete files</i></p>	<p>Cas de test : paramètres non valides</p> <ul style="list-style-type: none"> Cas E1 : le chemin du fichier est invalide Cas E2 : il manque des droits d'accès pour la suppression du fichier. 	- Un message correspondant au cas d'erreur est affiché	

F1.DE3-E3: Delete files	<ul style="list-style-type: none"> ● Cas E3 : l'identifiant machine est invalide <ol style="list-style-type: none"> 1. Lancer la commande vishnu_remove_file avec le cas de test correspondant (E1, E2 ou E3) 2. Vérifier que son exécution échoue avec le message d'erreur renvoyé. 		
F1.DE2-B : Delete directories	<p>Cas de test : Suppression du répertoire rep2 contenu dans le répertoire /home/rep1 de la machine d'identifiant MA_1. Supposons que les paramètres rep2, /home/rep1 et MA_1 sont corrects. Supposons également que le répertoire /home/rep1 existe dans la machine d'identifiant MA_1 et qu'il contient un répertoire nommé rep2.</p> <ol style="list-style-type: none"> 1. Lancer la commande vishnu_remove_dir avec comme paramètres MA_1:/home/rep1/rep2 2. Lancer le commande vishnu_list_dir avec comme paramètres MA_1:/home/rep1 et vérifier que le répertoire rep2 a bien été supprimé. La suppression du répertoire a bien été validée. 	- Le répertoire rep2 est supprimé du répertoire /home/rep1 sur la machine d'identifiant MA_1.	L'utilisateur doit avoir les droits d'écriture et d'exécution sur le répertoire rep1.
F1.DE1-E1: Delete directories F1.DE2-E2: Delete directories F1.DE3-E3: Delete directories	<p>Cas de test : paramètres non valides</p> <ul style="list-style-type: none"> ● Cas E1 : le chemin du répertoire est invalide ● Cas E2 : il manque des droits d'accès pour la suppression du répertoire. ● Cas E3 : l'identifiant machine est invalide <ol style="list-style-type: none"> 1. Lancer la commande vishnu_remove_dir avec le cas de test correspondant (E1, E2 ou E3) 2. Vérifier que son exécution échoue avec le message d'erreur renvoyé. 	- Un message correspondant au cas d'erreur est affiché	

4.1.3 Tests de la catégorie 2

Identifiant de tests	Actions	Résultats	Considérations particulières
F1.DI1-B: Display head of	Cas de test : Consultation des n premières lignes du fichier /home/rep1/file1 contenant m lignes ($m \geq n$) de la machine	- Les n premières	L'utilisateur doit avoir les

files	<p><i>d'identifiant MA_1. Supposons que les paramètres /home/rep1/file1 et MA_1 sont corrects et correspondent respectivement à un chemin vers un fichier existant et à une machine existante.</i></p> <ol style="list-style-type: none"> 1. Lancer la commande vishnu_head_of_file avec les paramètres MA1:/home/rep1/file1 et n. 2. Lancer la commande vishnu_content_of_file du fichier MA1:/home/rep1/file1 et vérifier que les n premières retournées correspondent bien aux n premières lignes du fichier. 	lignes du fichier sont retournées	droits d'exécution sur le répertoire rep1 et les droits de lecture sur le fichier file1.
<p>F1.DI1-E1: Display head of files</p> <p>F1.DI1-E2: Display head of files</p> <p>F1.DI3-E3: Display head of files</p>	<p>Cas de test : paramètres non valides</p> <ul style="list-style-type: none"> ● Cas E1 : le chemin du répertoire est invalide ● Cas E2 : il manque des droits d'accès pour pour la récupération du contenu du fichier. ● Cas E3 : l'identifiant machine est invalide <ol style="list-style-type: none"> 1. Lancer la commande vishnu_head_of_file avec le cas de test correspondant (E1, E2 ou E3) 2. Vérifier que son exécution échoue avec le message d'erreur renvoyé. 	- Un message correspondant au cas d'erreur est affiché	
F1.DI2-B: Display tail of files	<p>Cas de test : Consultation des n dernières lignes du fichier /home/rep1/file1 contenant m lignes ($m \geq n$) de la machine d'identifiant MA_1. Supposons que les paramètres /home/rep1/file1 et MA_1 sont corrects et correspondent respectivement à un chemin vers un fichier existant et à une machine existante.</p> <ol style="list-style-type: none"> 1. Lancer la commande vishnu_tail_of_file avec les paramètres MA1:/home/rep1/file1 et n. 2. Lancer la commande vishnu_content_of_file du fichier MA1:/home/rep1/file1 et vérifier que les n dernières retournées correspondent bien aux n premières lignes du fichier. 	- Les n dernières ligne du fichier sont retournées	L'utilisateur doit avoir les droits d'exécution sur le répertoire rep1 et les droits de lecture sur le fichier file1.
<p>F1.DI2-E1: Display tail of files</p> <p>F1.DI2-E2: Display tail of files</p> <p>F1.DI2-E3:</p>	<p>Cas de test : paramètres non valides</p> <ul style="list-style-type: none"> ● Cas E1 : le chemin du répertoire est invalide ● Cas E2 : il manque des droits d'accès pour pour la récupération du contenu du fichier ● Cas E3 : l'identifiant machine est invalide <ol style="list-style-type: none"> 1. Lancer la commande vishnu_tail_of_file avec le cas de test 	- Un message correspondant au cas d'erreur est affiché	

Display tail of files	correspondant (E1, E2 ou E3) 2. Vérifier que son exécution échoue avec le message d'erreur renvoyé.		
F1.DI3-B: Display content of files	<p>Cas de test : Consultation du contenu du fichier file1 contenu dans le répertoire /home/rep1 de la machine MA_1. Supposons que les paramètres /home/rep1/file1 et MA_1 sont corrects et correspondent respectivement à un chemin vers un fichier existant et à une machine existante.</p> <ol style="list-style-type: none"> Lancer la commande vishnu_content_of_file avec les paramètres MA1:/home/rep1/file1 Lancer la commande scp sur la machine MA_1 pour récupérer le fichier file1 et vérifier que son contenu correspond à celui retourné par la commande vishnu_content_of_file 	- Le contenu du fichier est retourné.	L'utilisateur doit avoir les droits d'exécution sur le répertoire rep1 et les droits de lecture sur le fichier file1.
F1.DI3-E1: Display content of files F1.DI3-E2: Display content of files F1.DI3-E3: Display content of files	<p>Cas de test : paramètres non valides</p> <ul style="list-style-type: none"> Cas E1 : le chemin du fichier est invalide Cas E2 : il manque des droits d'accès pour la récupération du contenu du fichier. Cas E3 : l'identifiant machine est invalide <ol style="list-style-type: none"> Lancer le programme vishnu_content_of_file avec le cas de test correspondant (E1, E2 ou E3) Vérifier que son exécution échoue avec le message d'erreur renvoyé. 	- Un message correspondant au cas d'erreur est affiché	
F1.DI4-B: Display content of directories	<p>Cas de test : Consultation du contenu du répertoire rep2 contenu dans le répertoire /home/rep1 de la machine MA_1. Supposons que les paramètres /home/rep1/rep2 et MA_1 sont corrects et correspondent respectivement à un chemin vers un répertoire existant et à une machine existante.</p> <ol style="list-style-type: none"> Lancer la commande vishnu_list_dir avec les paramètres MA1:/home/rep1/rep2 Lancer la commande scp -r sur la machine MA_1 pour récupérer le répertoire rep2 et vérifier que son contenu correspond à celui retourné par la commande vishnu_content_of_file 	- Le contenu du fichier est retourné.	L'utilisateur doit avoir les droits d'exécution sur le répertoire rep1 et les droits de lecture sur le fichier file1.
F1.DI4-E1: Display content of directories F1.DI3-E2: Display content of	<p>Cas de test : paramètres non valides</p> <ul style="list-style-type: none"> Cas E1 : le chemin du répertoire est invalide Cas E2 : il manque des droits d'accès pour la récupération du contenu du répertoire 	- Un message correspondant au cas d'erreur est affiché	

directories F1.DI3-E3: Display content of directories	<ul style="list-style-type: none"> ● Cas E3 : l'identifiant machine est invalide <ol style="list-style-type: none"> 1. Lancer le programme vishnu_content_of_dir avec le cas de test correspondant (E1, E2 ou E3) 2. Vérifier que son exécution échoue avec le message d'erreur renvoyé. 		
F1.DI5-B: Get information about remote files	<p>Cas de test : Consultation des informations sur les propriétés du fichier <i>file1</i> contenu dans le répertoire <i>/home/rep1</i> de la machine <i>MA_1</i>. Supposons que les paramètres <i>/home/rep1/file1</i> et <i>MA_1</i> sont corrects et correspondent respectivement à un chemin vers un fichier existant et à une machine existante.</p> <ol style="list-style-type: none"> 1. Lancer la commande vishnu_get_files_info avec les paramètres <i>MA1:/home/rep1/file1</i> 2. Récupérer le fichier <i>file1</i> sur la machine d'identifiant <i>MA_1</i> à l'aide de la commande. 3. Vérifier à l'aide de la fonction « fstat » que les propriétés retournées par la commande vishnu_get_files_info sont les mêmes que celles retournées par « fstat ». 	- Les propriétés du fichier sont retournées.	L'utilisateur doit avoir les droits d'exécution sur le répertoire <i>rep1</i> .
F1.DI5-E1: Get information about remote files F1.DI5-E2: Get information about remote files F1.DI5-E3: Get information about remote files	<p>Cas de test : paramètres non valides</p> <ul style="list-style-type: none"> ● Cas E1 : le chemin du répertoire est invalide ● Cas E2 : il manque des droits d'accès aux propriétés du fichier ● Cas E3 : l'identifiant machine est invalide <ol style="list-style-type: none"> 1. Lancer la commande vishnu_get_files_info avec le cas de test correspondant (E1, E2 ou E3) 2. Vérifier que son exécution échoue avec le message d'erreur renvoyé. 	- Un message correspondant au cas d'erreur est affiché	

4.1.4 Tests de la catégorie 3

Identifiant de tests	Actions	Résultats	Considérations particulières
F1.CH1-B: Change access rights of files	<p>Cas de test : Changement des droits d'accès du fichier <i>file1</i> contenu dans le répertoire <i>/home/rep1</i> de la machine</p>	- Les droits du fichiers ont été mis à jour.	L'utilisateur doit avoir les droits d'exécution sur le

	<p><i>MA_1. Supposons que les paramètres /home/rep1/file1 et MA_1 sont corrects et correspondent respectivement à un chemin vers un répertoire existant et à une machine existante.</i></p> <ol style="list-style-type: none"> 1. Lancer la commande vishnu_ch_grp avec les paramètres MA1:/home/rep1/file1 avec les nouveaux droits 2. Lancer la commande vishnu_get_files_info et vérifier que les droits fournis en paramètre en ont été appliqués. 		répertoire rep1 et les droits de lecture sur le fichier file1.
<p>F1.CH1-E1: Change access rights of files</p> <p>F1.CH1-E2: Change access rights of files</p> <p>F1.CH1-E3: Change access rights of files</p>	<p>Cas de test : paramètres non valides</p> <ul style="list-style-type: none"> ● Cas E1 : le chemin du répertoire est invalide ● Cas E2 : il manque des droits d'accès pour changer les propriétés du fichier ● Cas E3 : l'identifiant machine est invalide <ol style="list-style-type: none"> 1. Lancer la commande vishnu_ch_mod avec le cas de test correspondant (E1, E2 ou E3) 2. Vérifier que son exécution échoue avec le message d'erreur renvoyé. 	- Un message correspondant au cas d'erreur est affiché	
F1.CH2-B: Change group owner of files	<p>Cas de test : Changement du groupe propriétaire du fichier file1 contenu dans le répertoire /home/rep1 de la machine MA_1. Supposons que les paramètres /home/rep1/file1 et MA_1 sont corrects et correspondent respectivement à un chemin vers un répertoire existant et à une machine existante.</p> <ol style="list-style-type: none"> 1. Lancer la commande vishnu_ch_mod avec les paramètres MA1:/home/rep1/file1 avec le nouveau nom de groupe à associer au fichier 2. Lancer la commande vishnu_get_files_info et vérifier que le 	- Le nom du groupe associé au fichier a été mis à jour.	L'utilisateur doit avoir les droits d'exécution sur le répertoire rep1 et les droits de lecture sur le fichier file1.

	nom du groupe associé au fichier à changer et que ce nom correspond à celui donné en paramètre de <i>vishnu_ch_mod</i> .		
<i>F1.CH2-E1: Change group owner of files</i> <i>F1.CH2-E2: Change group owner of files</i> <i>F1.CH2-E3: Change group owner of files</i>	Cas de test : paramètres non valides <ul style="list-style-type: none"> ● Cas E1 : le chemin du répertoire est invalide ● Cas E2 : il manque des droits d'accès pour changer les propriétés du fichier ● Cas E3 : l'identifiant machine est invalide <ol style="list-style-type: none"> 1. Lancer la commande <i>vishnu_ch_grp</i> avec le cas de test correspondant (E1, E2 ou E3) 2. Vérifier que son exécution échoue avec le message d'erreur renvoyé. 	- Un message correspondant au cas d'erreur est affiché.	

4.1.5 Tests de catégorie 4

Identifiant de tests	Actions	Résultats	Considérations particulières
<i>F2.CP1-B:Execute a synchronous copy of files</i>	<ol style="list-style-type: none"> 1. Cas de test : les paramètres sont corrects. <ul style="list-style-type: none"> ● Cas 1 : copie d'un fichier (ou répertoire) local vers un répertoire local. ● Cas 2 : copie d'un fichier (ou répertoire) local vers un répertoire distant. ● Cas 3 : copie d'un fichier (ou répertoire) distant vers un répertoire local. ● Cas 4 : copie d'un fichier (ou répertoire) distant vers un répertoire distant. 2. Lister le répertoire destination pour les cas 1, 2, 3 et 4 à l'aide de la commande <i>vishnu_list_dir</i> et vérifier que le fichier ou répertoire à copier 	Le fichier (ou répertoire) a été copié sur la destination spécifiée.	L'utilisateur doit avoir les droits en lecture, écriture et exécution sur le répertoire destination et les droits de lecture, écriture et exécution sur le fichier (ou répertoire) à copier. Le transfert de gros fichiers a été pris

	<p>n'existe pas.</p> <p>3. Lancer la commande vishnu_copy_file avec le nom du répertoire destination et les identifiants des machines sources et destinations pour les cas 1, 2, 3 et 4 .</p> <p>4. Utiliser la commande scp pour récupérer le fichier (ou répertoire) copié et l'enregistrer sous un autre nom.</p> <p>5. Comparer le contenu des deux fichiers (ou répertoires) .</p>		en compte.
<p>F2.CP1-E1:Execute a synchronous copy of files</p> <p>F2.CP1-E2:Execute a synchronous copy of files</p> <p>F2.CP1-E3:Execute a synchronous copy of files</p> <p>F2.CP1-E4:Execute a synchronous copy of files</p>	<p>Cas de test : les paramètres sont incorrects.</p> <ul style="list-style-type: none"> ● Cas E1 : chemin du fichier (ou répertoire) source est incorrect ● Cas E2 : chemin du fichier (ou répertoire) destination est incorrect ● Cas E3 : absence des droits d' accès au fichier (ou répertoire) source et/ou destination ● Cas E4 : l'identifiant de la machine n'est pas enregistré dans VISHNU <p>1. Lancer la commande vishnu_copy_file avec le cas de test correspondant (E1, E2, E3 ou E4) et vérifier que son exécution échoue.</p>	- Un message correspondant au cas d'erreur est affiché.	
F2.CP2-B:Execute a asynchronous copy of files	<p>1. Cas de test : les paramètres sont corrects.</p> <ul style="list-style-type: none"> ● Cas 1 : copie d'un fichier (ou répertoire) local vers un répertoire local. ● Cas 2 : copie d'un fichier (ou répertoire) local vers un répertoire distant. ● Cas 3 : copie d'un fichier (ou répertoire) distant vers un répertoire local. ● Cas 4 : copie d'un fichier (ou répertoire) distant vers un répertoire distant. <p>2. Lister le répertoire destination pour les cas 1, 2, 3 et 4 à l'aide de la commande vishnu_list_dir et vérifier que le fichier ou répertoire à copier n'existe pas.</p> <p>3. Lancer la commande vishnu_async_copy_file avec le nom du répertoire destination et les identifiants des machines sources et destinations pour les cas 1, 2, 3 et 4 et conserver l'identifiant</p>	Le fichier (ou répertoire) a été copié sur la destination spécifiée.	L'utilisateur doit avoir les droits en lecture, écriture et exécution sur le répertoire destination et les droits de lecture, écriture et exécution sur le fichier (ou répertoire) à copier.

	<p>du transfert.</p> <p>4. Attendre la fin du transfert et utiliser la commande scp pour récupérer le fichier (ou répertoire) copié et l'enregistrer sous un autre nom.</p> <p>5. Comparer le contenu des deux fichiers (ou répertoires) .</p>		
<p>F2.CP2-E1:Execute a asynchronous copy of files</p> <p>F2.CP2-E2:Execute a asynchronous copy of files</p> <p>F2.CP2-E3:Execute asynchronous copy of files</p> <p>F2.CP2-E4:Execute asynchronous copy of files</p>	<p>1. Cas de test : les paramètres sont incorrects.</p> <ul style="list-style-type: none"> ● Cas E1 : chemin du fichier (ou répertoire) source est incorrect ● Cas E2 : chemin du fichier (ou répertoire) destination est incorrect ● Cas E3 : absence des droits d' accès au fichier (ou répertoire) source et/ou destination ● Cas E4 : l'identifiant de la machine n'est pas enregistré dans VISHNU <p>2. Lancer la commande vishnu_async_copy_file avec le cas de test correspondant (E1, E2, E3 ou E4) et vérifier que son exécution échoue.</p>	- Un message correspondant au cas d'erreur est affiché.	
<p>F2.MV1-B:Execute a synchronous move of files</p>	<p>1. Cas de test : les paramètres sont corrects.</p> <ul style="list-style-type: none"> ● Cas 1 : déplacer un fichier (ou répertoire) local vers un répertoire local. ● Cas 2 : déplacer d'un fichier (ou répertoire) local vers un répertoire distant. ● Cas 3 : déplacer d'un fichier (ou répertoire) distant vers un répertoire local. ● Cas 4 : déplacer d'un fichier (ou répertoire) distant vers un répertoire distant. <p>2. Lister le répertoire destination pour les cas 1, 2, 3 et 4 à l'aide de la commande vishnu_list_dir et vérifier que le fichier ou répertoire à déplacer n'existe pas.</p> <p>3. Faire une copie du fichier (ou répertoire) à déplacer dans le répertoire local .</p> <p>4. Lancer la commande vishnu_move_file avec le nom du répertoire destination et les identifiants des machines sources et destinations pour les cas 1, 2, 3 et 4 .</p>	Le fichier (ou répertoire) a été déplacé sur la destination spécifiée.	L'utilisateur doit avoir les droits en lecture, écriture et exécution sur le répertoire destination et les droits de lecture, écriture et exécution sur le fichier (ou répertoire) à copier.

	<p>5. Vérifier que le fichier ou répertoire à déplacer n'existe plus à l'aide de la commande <i>vishnu_list_dir</i>.</p> <p>6. Utiliser la commande <i>scp</i> pour récupérer le fichier ou répertoire déplacé et l'enregistrer sous un autre nom.</p> <p>7. Comparer le contenu du fichier ou du répertoire de l'étape 3 avec le fichier (ou le répertoire) obtenu à l'aide de la commande <i>scp</i>.</p>		
<i>F2.MV1-E1:Execute a synchronous move of files</i> <i>F2.MV1-E2:Execute a synchronous move of files</i> <i>F2.MV1-E3:Execute a synchronous move of files</i> <i>F2.MV1-E4:Execute a synchronous move of files</i>	<p>Cas de test : les paramètres sont incorrects.</p> <ul style="list-style-type: none"> ● Cas E1 : chemin du fichier (ou répertoire) source est incorrect ● Cas E2 : chemin du fichier (ou répertoire) destination est incorrect ● Cas E3 : absence des droits d'accès au fichier (ou répertoire) source et/ou destination ● Cas E4 : l'identifiant de la machine n'est pas enregistré dans VISHNU <p>1. Lancer la commande <i>vishnu_move_file</i> avec le cas de test correspondant (E1, E2, E3 ou E4) et vérifier que son exécution échoue.</p>	- Un message correspondant au cas d'erreur est affiché	
<i>F2.MV2-B:Execute a asynchronous move of files</i>	<p>1. Cas de test : les paramètres sont corrects.</p> <ul style="list-style-type: none"> ● Cas 1 : déplacer un fichier (ou répertoire) local vers un répertoire local. ● Cas 2 : déplacer d'un fichier (ou répertoire) local vers un répertoire distant. ● Cas 3 : déplacer d'un fichier (ou répertoire) distant vers un répertoire local. ● Cas 4 : déplacer d'un fichier (ou répertoire) distant vers un répertoire distant. <p>2. Lister le répertoire destination pour les cas 1, 2, 3 et 4 à l'aide de la commande <i>vishnu_list_dir</i> et vérifier que le fichier ou répertoire à déplacer n'existe pas.</p> <p>3. Faire une copie du fichier (ou répertoire) à déplacer dans le répertoire local.</p> <p>4. Lancer la commande <i>vishnu_move_async_file</i> avec le nom du répertoire destination et les identifiants des machines sources et destinations</p>	Le fichier (ou répertoire) a été déplacé sur la destination spécifiée.	L'utilisateur doit avoir les droits en lecture, écriture et exécution sur le répertoire destination et les droits de lecture, écriture et exécution sur le fichier (ou répertoire) à copier.

	<p>pour les cas 1, 2, 3 et 4 .</p> <p>5. Vérifier que le fichier ou répertoire à déplacer n'existe plus à l'aide de la commande <i>vishnu_list_dir</i>.</p> <p>6. Attendre la fin du transfert et utiliser la commande <i>scp</i> pour récupérer le fichier ou répertoire déplacé et l'enregistrer sous un autre nom.</p> <p>7. Comparer le contenu du fichier ou du répertoire de l'étape 3 avec le fichier (ou le répertoire) obtenu à l'aide de la commande <i>scp</i>.</p>		
<i>F2.MV2-E1:Execute a asynchronous move of files</i> <i>F2.MV2-E2:Execute a asynchronous move of files</i> <i>F2.MV2-E3:Execute a asynchronous move of files</i> <i>F2.MV2-E4:Execute a asynchronous move of files</i>	<p>Cas de test : les paramètres sont incorrects.</p> <ul style="list-style-type: none"> ● Cas E1 : chemin du fichier (ou répertoire) source est incorrect ● Cas E2 : chemin du fichier (ou répertoire) destination est incorrect ● Cas E3 : absence des droits d'accès au fichier (ou répertoire) source et/ou destination ● Cas E4 : l'identifiant de la machine n'est pas enregistré dans VISHNU <p>1. Lancer la commande <i>vishnu_move_async_file</i> avec le cas de test correspondant (E1, E2, E3 ou E4) et vérifier que son exécution échoue.</p>	- Un message correspondant au cas d'erreur est affiché	
<i>F2.CA1-B:Cancel file transfers</i>	<p>1. Cas de test : les paramètres sont corrects.</p> <p>2. Lancer deux fois le programme <i>vishnu_copy_async_file</i> pour transférer deux répertoires qui nécessitent un temps de transfert estimé à cinq minutes chacun en fonction des caractéristiques du réseau et de la taille du répertoire.</p> <p>3. Vérifier que les transferts sont en cours d'exécution à l'aide de la commande <i>vishnu_list_file_transfers</i>.</p> <p>4. Lancer la commande <i>vishnu_stop_file_transfer</i> .</p> <p>5. Lancer la commande <i>vishnu_list_file_transfers</i> avant la fin des deux transferts et vérifier qu'ils sont annulés.</p>	Le transfert est annulé.	L'utilisateur doit avoir les droits en lecture, écriture et exécution sur le répertoire destination et les droits de lecture, écriture et exécution sur le répertoire à copier.

F2.CA1-E1:Cancel file transfers	<ol style="list-style-type: none"> Cas de test : les paramètres sont incorrects. <ul style="list-style-type: none"> Cas E1 : Aucun transfert en cours Lancer la commande <i>vishnu_stop_file_transfer</i> avec le cas de test correspondant et vérifier que son exécution échoue. 	- Un message correspondant au cas d'erreur est affiché	
F2.CA2-B:Cancel all users file transfers	<ol style="list-style-type: none"> Cas de test : Un administrateur liste les transferts de fichiers de l'utilisateur U_X. Cas 1 : Supposons que l'utilisateur U_X est ajouté pour ce test à l'aide de la commande <i>vishnu_add_user</i> et que l'administrateur se substitue à l'utilisateur U_X lors de l'ouverture de session avec l'option -s. Cas 2 : Supposons que l'administrateur liste les transferts de fichiers de l'utilisateur U_X. Créer deux fichiers file1 et file2 dans votre répertoire local. Copier les fichiers file1 et file2 de votre répertoire local vers un répertoire distant à l'aide du programme <i>vishnu_copy_async_file</i>. Cas 1 : Lancer le programme <i>vishnu_list_file_transfers</i> et vérifier que les deux transferts de fichiers sont retournés. Cas 2 : Lancer le programme <i>vishnu_list_file_transfers</i> avec l'option -i suivie de U_X et vérifier que les deux transferts de fichiers précédents s'affichent. 	Le transfert est annulé.	L'utilisateur doit être un administrateur
F2.LT1-B:List all file transfers F2.LT2-B:List all users file transfers	<ol style="list-style-type: none"> Cas de test : Un administrateur liste les transferts de fichiers de l'utilisateur donné. Cas 1 : Supposons que l'utilisateur donné est ajouté pour ce test à l'aide de la commande <i>vishnu_add_user</i> et que l'administrateur se substitue à l'utilisateur lors de l'ouverture de session avec l'option -s. Cas 2 : Supposons que l'administrateur liste les transferts de fichiers de l'utilisateur donné. 	- La liste des fichiers en cours de transfert est affichée.	L'utilisateur qui effectue le cas de test est un administrateur

	<ol style="list-style-type: none"> Créer deux fichiers file1 et file2 dans votre répertoire local. Copier les fichiers file1 et file2 de votre répertoire local vers un répertoire distant à l'aide du programme <i>vishnu_copy_async_file</i>. <p>Cas 1 :</p> <ol style="list-style-type: none"> Lancer le programme <i>vishnu_list_file_transfers</i> et vérifier que les deux transferts de fichiers sont retournés. <p>Cas 2 :</p> <ol style="list-style-type: none"> Lancer le programme <i>vishnu_list_file_transfers</i> avec l'option -i suivie de U_X et vérifier que les deux transferts de fichiers précédents s'affichent sont retournés. 		
<i>F2.LT1-E1:List all file transfers</i> <i>F2.LT1-E2:List all file transfers</i> <i>F2.LT1-E3:List all file transfers</i> <i>F2.LT2-E1:List all users file transfers</i> <i>F2.LT2-E2:List all users file</i> <i>F2.LT2-E3:List all users file</i> <i>F2.LT2-E4:List all users file</i>	<ol style="list-style-type: none"> Cas de test : paramètres non valides <ul style="list-style-type: none"> Cas E1 : le chemin du répertoire est invalide Cas E2 : il manque des droits d'accès aux propriétés du fichier Cas E3 : l'identifiant machine est invalide Cas E4 : l'utilisateur n'est pas un administrateur Lancer le programme <i>vishnu_list_file_transfers</i> avec les cas d'exception précédents. Vérifier que son exécution échoue avec le message d'erreur renvoyé. 		

4.2 Tests de stress

4.2.1 Présentation

Ces tests permettent de vérifier que la cible de test peut réussir à basculer et à récupérer pour différents dysfonctionnements matériels, logiciels ou réseaux sans qu'il y ait perte de données. Ils vérifient que lorsqu'une défaillance survient un système alternatif ou un système de sauvegarde prends le relai correctement sans perte de données ou de transactions.

4.2.2 Environnement des tests

Nous utiliserons l'environnement de test 1 dans un environnement isolé (réseau et systèmes non partagés).

4.2.3 Identifiants de tests

Les identifiants des tests sont de la forme: STR-<Element stressé>-<Type de stress>.

4.2.4 Liste des tests

Identifiant de tests	Technique	Critère de complétion
STR-SERV-DOWN	<ul style="list-style-type: none"> - Lancer la commande de transfert de fichier - Arrêter le serveur FMS avant la fin d'exécution de la commande Cas 1 : Redémarrer le serveur FMS avant la fin de la copie Cas 2 : <i>Redémarrer le serveur FMS après la fin de la copie</i>	<ul style="list-style-type: none"> - Cas 1:Terminaison normale du transfert avec son statut mis à COMPLETED -Cas 2 : Statut du transfert mis à FAILED
STR-AGENT-DOWN	<ul style="list-style-type: none"> - Lancer la commande <i>vishnu_copy_file</i> - Arrêter l'agent SysFera-DS après l'exécution de la commande - Redémarrer l'agent SysFera-DS - Lancer la commande <i>vishnu_remove_file</i> 	<ul style="list-style-type: none"> - Terminaison normale de la commande <i>vishnu_remove_file</i>

4.3 Contrôle des performances

4.3.1 Présentation

Ces tests ont pour objectif de vérifier que les exigences de performance quantifiées sont effectivement satisfaites.

4.3.2 Environnement de tests

Nous utiliserons l'environnement de test 1.

4.3.3 Identifiants des tests

Les identifiants des tests sont de la forme: CDP-<ID STB>-[NomdeCommande|IdentifiantDeTest] où 'ID STB' est l'identifiant de l'exigence spécifiée dans le document [STB] au paragraphe « performances, fiabilité et robustesse ».

4.3.4 Liste des tests

Identifiant de tests	Technique	Critère de completion
CDP-3.1-vishnu_copy_file	<ul style="list-style-type: none"> - Lancer une commande 'scp <host> <fileToCopy>' sur une machine serveur en mesurant le temps de réponse (temps de 	temps du même ordre que le temps de référence (valeur moyenne sur 10 itérations)

	référence) - Lancer la commande vishnu_copy_file en mesurant le temps de réponse	
CDP-3.1-vishnu_get_file_info	- Lancer une commande 'ssh <host> stat <FileToDelete>' sur une machine serveur en mesurant le temps de réponse (temps de référence). - Lancer la commande vishnu_get_file_info en mesurant le temps de réponse	temps du même ordre que le temps de référence (valeur moyenne sur 10 itérations)
CDP-3.2.1-vishnu_copy_file	- Lancer la commande vishnu_copy_file 100 fois	Consommation en mémoire virtuelle sur le serveur FMS inférieure à 500 Mégaoctets.
CDP-3.2.2-vishnu_copy_file	- Lancer le serveur FMS	Consommation CPU inférieure à 10%.
CDP-MemServer	- Lancer le serveur FMS - Laisser le serveur fonctionner sans utiliser les services pendant 24h - Vérifier que l'espace mémoire consommé par le processus serveur n'a pas augmenté.	Consommation mémoire constante.

4.4 Tests de charge

4.4.1 Présentation

Les tests de charge ont pour objectif de vérifier que les exigences de capacité du système sont vérifiées.

4.4.2 Environnement de tests

Nous utiliserons l'environnement de test 1 avec un nombre de clients ou de serveurs tels que spécifié dans le cas de tests. Ces clients ou serveurs seront simulés sur une seule machine.

Par ailleurs, les requêtes envoyées en simultanée seront en fait envoyées par une seule machine cliente avec un programme de synchronisation (MPI).

4.4.3 Identifiants de tests

Les identifiants des tests sont de la forme: LOAD-<ID STB>-[NomdeCommande|IdentifiantDeTest] où 'ID STB' est l'identifiant de l'exigence spécifiée dans le document [STB].

4.4.4 Liste des tests

Identifiant de tests	Technique	Résultats attendus
LOAD-2.1-vishnu_create_file	- Lancer 100 commandes	- Terminaison normale de toutes les

	<i>vishnu_create_file</i> simultanées	commandes - Etat normal du serveur FMS
LOAD-2.1- vishnu_list_file_transfers	- Lancer 100 commandes <i>vishnu_liste_file_transfers</i> simultanées	- Terminaison normale de toutes les commandes - Etat normal du serveur FMS

4.5 Contrôle de la qualité du code

4.5.1 Présentation

L'objectif de ces tests est de vérifier que le code source (C/C++ ou Java) respecte les contraintes de qualité logiciel fixées dans les spécifications et le cahier des charges.

4.5.2 Identifiants des tests

Les identifiants des tests sont de la forme: [IdentifiantDeTest]-[IdentifiantDesModules] où les identifiants sont les suivants:

- Identifiants de tests
 - CRP: Conformité aux Règles de Programmation
 - MCC: Mesure de la Complexité du Code
 - DDC: Détection de Duplication du Code
- Identifiants de modules
 - CSC : C/C++ source code
 - JSC : Java source code

4.5.3 Liste des tests

Identifiant de tests	Technique	Résultats attendus
CRP-CSC	Audit du code C/C++ avec l'outil LOGISCOPE	Le code respecte toutes les règles de codage de catégorie «REQUIRED » spécifiées pour le projet [SCL] sauf dérogation clairement documentée dans le code.
CRP-JSC	Audit du code Java avec l'outil LOGISCOPE	Le code respecte toutes les règles de codage de catégorie «REQUIRED » spécifiées pour le projet [SCL] sauf dérogation clairement documentée dans le code.
MCC-CSC	Audit du code C/C++ avec l'outil SQUORE	Pas plus de 1% de composants classés « A risque »

MCC-JSC	Audit du code Java avec l'outil SQUORE	Pas plus de 1% de composants classés « A risque »
DDC-CSC	Audit du code C/C++ avec l'outil SQUORE	Moins de 10% de code dupliqué.
DDC-JSC	Audit du code Java avec l'outil SQUORE	Moins de 10% de code dupliqué.

4.6 Tests de compilation et d'installation

4.6.1 Présentation

La compilation des différents exécutables et bibliothèques composant le module UMS sera réalisée par un serveur d'intégration continue (du type HUDSON).

Par ailleurs les tests d'installation ont pour objectif de valider la procédure d'installation de l'ensemble des différents composants d'une plateforme VISHNU (clients, serveurs, agents SysFera-DS).

4.6.2 Environnement des tests

Nous utiliserons les deux environnements de tests pour la compilation et l'installation.

4.6.3 Liste des tests

Identifiant de tests	Technique	Résultats attendus
COMP-CSC	Compilation du code C/C++ avec l'outil GCC/CMake	Compilation correcte de tous les exécutables
INS-SERVER-PGSQL INS-SERVER-ORACLE	Installation des différents éléments serveur de la plateforme VISHNU <ul style="list-style-type: none"> • serveur de calcul • serveur de stockage • serveur dédié VISHNU • serveur de base de données 	<ul style="list-style-type: none"> - Tous les processus démarrés et opérationnels - Documentation disponible (pages man)
INS-CLI	Installation du client VISHNU en ligne de commande	<ul style="list-style-type: none"> - Toutes les commandes installées - Documentation disponible (pages man)
INS-PGAPI	Installation des bibliothèques API VISHNU (C++ et Python)	<ul style="list-style-type: none"> - Bibliothèque C++ installée - Bibliothèque Python installée et pouvant être chargée dans l'interpréteur Python 2.x
INS-WSAPI	Installation des services web VISHNU sur le serveur JBoss	<ul style="list-style-type: none"> - Services opérationnels (contrôle via la console webservices de JBoss)

4.7 Contrôle de la documentation utilisateur

4.7.1 Présentation

L'objectif est de vérifier que la documentation utilisateur permet à l'utilisateur de comprendre et d'utiliser les fonctions spécifiées .

4.7.2 Liste des tests

Identifiant de tests	Technique	Résultats attendus
CDU-MAN	Vérifier les pages de manuel attachées à chaque commande	Toutes les pages de manuel contiennent toutes les informations nécessaires à l'utilisateur pour utiliser les commandes.
CDU-USERSGUIDE	Suivre le guide de l'utilisateur	Le guide permet une prise en main rapide des fonctions du logiciel