fuse
Management Central

Fuse Management Central

Installation and Administration Guide

Version 1.9.1

VILT

Contents

1. Introduction **2**
2. Overview of Fuse Management Central Architecture **3**
3. Install Fuse Management Central **5**
3.1. Pre-Installation Tasks 5
3.1.1. Communication Ports Availability 5
3.1.2. Install Java 8
3.1.3. Install NTP (recommended) 9
3.1.4. Enabling SSL 9
3.1.4.1. Register SSL certificate 9
3.1.4.2. Expose Fuse Management Central as HTTPS server 10
3.1.4.3. Self-signed certificate support 11
3.1.4.4. Enable SSL in HyperLens 11
3.2. Microsoft Windows 12
3.2.1. Installation 12
3.2.2. Upgrade 13
3.3. Linux 14
3.3.1. Pre-requirements 14
3.3.2. Installation and Configuration 15
3.3.3. Upgrade 16
3.4. Docker 17
3.4.1. Pre-requirements 17
3.4.2. Run with Docker Compose 18
3.4.3. Advanced Configuration 18
3.4.3.1. Data Persistence 18
3.4.3.2. JVM Options 19
3.4.3.3. Third-Party Components 19
3.4.4. Upgrade 19
3.5. Kubernetes and OpenShift Helm Deployment 20
3.5.1. What This Chart Deploys 21
3.5.2. Package Artifacts 21
3.5.3. Platform Prerequisites 21
3.5.4. Prepare the Image 22
3.5.5. Review Values 22
3.5.6. Install 23
3.5.7. Validate 23
3.5.8. Upgrade 24
3.6. Validate Fuse Management Central Installation 25
3.7. Post-installation 25
3.8. Next Steps 25
4. Install Fuse Management Client **27**
4.1. Install Fuse Management Client for OpenText Content Server 27
4.1.1. Install Fuse Management Client using an extended Content Server image . . . 28

4.1.2. Install Fuse Management Client using an OpenText init container 31
4.1.3. (Optional) Install Fuse Management Client for OpenText Content Server using Opentext System Center Manager 33
4.1.4. Configure Fuse Management Client logs 34
4.1.5. Patching Fuse Management Client for OpenText Content Server 34
4.1.6. Custom SSL Configuration 35
4.1.6.1. Overview 35
4.1.6.2. Configuration 35
4.1.6.3. Parameters 35
4.2. Install Fuse Management Client for OpenText Archive Center 36
4.2.1. User Requirements 37
4.2.2. Installation on Microsoft Windows 37
4.2.3. Installation on Linux 38
4.2.3.1. Systemd 38
4.2.3.2. Init.d 38
4.2.3.2.1. Daemon 39
4.2.3.3. Fuse Management Client for OpenText Archive Center Configuration . . 39
4.2.4. Kubernetes and OpenShift Helm Deployment 39
4.2.4.1. What This Chart Deploys 40
4.2.4.2. Package Artifacts 40
4.2.4.3. Platform Prerequisites 40
4.2.4.4. Prepare the Image 41
4.2.4.5. RBAC Requirements 42
4.2.4.6. Review Values 42
4.2.4.7. Install 43
4.2.4.8. Validate 43
4.2.4.9. Register the OTAC Client in Fuse Management Central 44
4.2.4.10. Upgrade 44
4.2.4.11. Troubleshooting 45
4.2.5. Upgrade Fuse Management Client for OpenText Archive Center 45
4.2.6. Failover Cluster Scenario for Fuse Management Client for OpenText Archive Center 46
4.2.6.1. How the Client Works (Architecture) 46
4.2.6.2. The Failover Concept 46
4.2.6.3. Configuration Steps 47
4.2.7. Additional Settings 48
4.2.7.1. OpenText Archive Center URL 49
4.2.7.2. Fuse Management Client for OpenText Archive Center Port 49
4.2.7.3. Configuration Requirements for Fuse Management Client for OpenText Archive Center UDP Ports 49
4.2.7.4. Configuration Requirements for a Document Pipeline Cluster Server . . 49
4.2.7.5. SSL Support 50
4.2.7.6. Register SSL certificate 50
4.2.7.7. Expose Fuse Management Client for OpenText Archive Center as

HTTPS server . . . 51
4.2.7.8. Self-signed certificate support . . . 51
4.2.8. Post-installation steps . . . 51
4.2.8.1. Validate installation . . . 51
4.2.8.2. Checking for possible hotfixes . . . 52
5. Fuse Management Central Administration . . . 53
5.1. Security . . . 53
5.1.1. Change *fuseadmin* password . . . 53
5.1.2. Change *fuseadmin* email . . . 53
5.2. General . . . 53
5.3. License . . . 54
5.3.1. Fuse Management Central for Content Server License . . . 54
5.3.2. Fuse Management Central for Archive Center Server License . . . 55
5.3.3. Request License . . . 55
5.3.4. Apply License . . . 56
5.3.5. Validate License Status . . . 56
5.4. OTDS Integration . . . 56
5.4.1. Create OTDS Resource . . . 57
5.4.1.1. Add users and/or groups to the created Resource . . . 57
5.4.2. Activate OTDS Resource . . . 57
5.4.3. Configuring Fuse Management Central Access Roles . . . 58
5.5. Add New System . . . 59
5.5.1. Activation Request . . . 59
5.5.2. Authorize Activation . . . 61
5.5.2.1. For OpenText Content Server . . . 61
5.5.2.2. For OpenText Archive Center . . . 61
5.6. Integration Channels . . . 62
5.6.1. SMTP . . . 62
5.6.1.1. Custom Email Settings . . . 63
5.6.2. Checkmk Integration . . . 64
5.6.2.1. Checkmk Plug-in Download . . . 65
5.6.2.2. Checkmk Plug-in Installation and Configuration . . . 65
5.6.2.3. Instance Service . . . 67
5.6.2.4. Other Services . . . 67
5.6.3. OpenText Service Management Automation X (SMAX) . . . 68
5.6.3.1. OpenText SMAX Integration Setup . . . 68
5.6.3.2. OpenText SMAX Custom Settings . . . 70
5.6.3.3. OpenText SMAX Incident Flow . . . 70
5.6.3.4. OpenText SMAX API Calls . . . 71
5.6.4. ServiceNow Integration . . . 71
5.6.4.1. ServiceNow Integration Setup . . . 71
5.7. Alert Manager . . . 73
5.7.1. Integration Channels . . . 73
5.7.2. Metric Thresholds . . . 74

5.7.3. Dismissing Alerts . . . 74
5.8. Alerts API . . . 74
5.8.1. Alerts List Endpoint . . . 75
5.8.1.1. System Type . . . 76
5.8.1.2. Alert Type and Component Type . . . 76
5.8.2. Layout Endpoint . . . 85
5.8.3. Alerts Summary Endpoint . . . 85
5.8.3.1. Summary entry use cases . . . 86
5.8.3.2. Systems . . . 86
5.8.3.3. Environments . . . 86
5.8.3.4. Admin . . . 87
5.9. Backup and Restore . . . 87
5.9.1. Backup Fuse Management Central Data . . . 87
5.9.2. Restore a Backup . . . 88
6. HyperLens (Experimental Feature) . . . 89
6.1. HyperLens Processor . . . 89
6.1.1. Windows Installation . . . 89
6.1.2. Windows Upgrade . . . 89
6.2. HyperLens Collector . . . 89
6.2.1. Windows Installation . . . 89
6.2.2. Windows Upgrade . . . 90
6.3. HyperLens Administration Guide . . . 90
6.3.1. HyperLens Processor Administration . . . 90
6.3.1.1. Configuration Properties . . . 90
6.3.2. HyperLens Collector Administration . . . 91
6.3.2.1. Configuration Properties . . . 91
7. Uninstall Fuse Management Central . . . 93
7.1. Uninstall on Microsoft Windows . . . 93
7.2. Uninstall Helm Deployments . . . 93
7.2.1. Uninstall Fuse Management Central . . . 93
7.2.2. Uninstall Fuse Management Client for OpenText Archive Center . . . 93
7.3. Uninstall HyperLens . . . 94
7.3.1. Uninstall HyperLens Processor . . . 94
7.3.2. Uninstall HyperLens Collector . . . 94
8. Appendix A - Troubleshooting . . . 95
8.1. Known Issues and Workarounds . . . 95
8.1.1. Metrics not available after installation or upgrade . . . 95
8.1.2. Error when adding a new System with https . . . 95
8.1.3. Fuse Metrics Database corrupted files . . . 95
8.1.4. Uninstall Fuse Management Client for OpenText Content Server (16.2.2, 16.2.3, 16.2.4) . . . 96
8.1.5. Fuse Management Central unable to connect with ServiceNow . . . 96
8.1.6. CPU Usage issues in Windows systems . . . 97
8.1.7. SELinux blocking Fuse Management Client for OpenText Archive Center

execution . . . 97
8.1.8. "Could not find the class definition" llserverworker JVM error . . . 98
8.1.9. Deactivated or Deleted Systems still collecting HyperLens data . . . 98
8.2. Helm Troubleshooting . . . 98
8.2.1. ImagePullBackOff . . . 99
8.2.2. Namespace mismatch . . . 99
8.2.3. PVC remains Pending . . . 99
8.2.4. Fuse Management Central is not reachable . . . 99
8.2.5. OTAC Client cannot resolve the OTAC workload . . . 100
8.2.6. OTAC Client reports Metrics API unavailable . . . 100
8.2.7. OTAC Client cannot execute approved OTAC utilities . . . 100
8.2.8. `storagedevices` reports no local storage manager configured . . . 101
8.2.9. NetworkPolicy blocks Helm deployment traffic . . . 101
9. Appendix B - How-Tos . . . 102
9.1. How to install and configure Prometheus on a Linux Server . . . 102
9.1.1. Pre-requirements . . . 102
9.1.2. Setup Prometheus . . . 102
9.1.3. Setup Prometheus Configuration . . . 103
9.1.4. Setup Prometheus as a Service . . . 104
9.1.5. Validate Prometheus installation . . . 104
9.1.6. Possible issues and workarounds . . . 105
9.1.6.1. Running Prometheus in a different Port . . . 105
9.1.6.2. Firewall blocking external access to Prometheus . . . 105
9.1.6.3. SELinux blocking Prometheus binary execution . . . 105
9.2. How to install and configure AlertManager on a Linux Server . . . 106
9.2.1. Pre-requirements . . . 106
9.2.2. Setup AlertManager . . . 106
9.2.3. Setup AlertManager Configuration . . . 107
9.2.4. Setup AlertManager as a Service . . . 107
9.2.5. Validate AlertManager installation . . . 108
9.2.6. Possible issues and workarounds . . . 108
9.2.6.1. Running AlertManager in a different Port . . . 108
9.2.6.2. Firewall blocking external access to AlertManager . . . 108
9.2.6.3. SELinux blocking AlertManager binary execution . . . 109
9.3. How to upgrade Prometheus on a Linux Server . . . 109
9.3.1. Requirements and Assumptions . . . 109
9.3.2. Upgrade Prometheus . . . 109
9.4. How to upgrade AlertManager on a Linux Server . . . 110
9.4.1. Requirements and Assumptions . . . 111
9.4.2. Upgrade AlertManager . . . 111
9.5. How to use a specific Java version with Fuse Management Central or Fuse Management Central Client for OpenText Archive Center . . . 112
9.6. How to setup Direct Server SSL Configuration . . . 112

Copyright Notice

© 2026 **VILT Group, S.A.**. All rights reserved.

This document and its contents are proprietary to VILT Group, S.A.. Unauthorized reproduction, distribution, or modification of this document, in whole or in part, is strictly prohibited without prior written consent from VILT Group, S.A..

For more information, visit www.vilt-group.com or contact us at info@vilt-group.com.

Disclaimer

No Warranties and Limitation of Liability: Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, VILT Group, S.A. and its affiliates accept no responsibility and provide no warranty whether expressed or implied, for the accuracy of this publication.

Document Revision Date

2026-10-21

1. Introduction

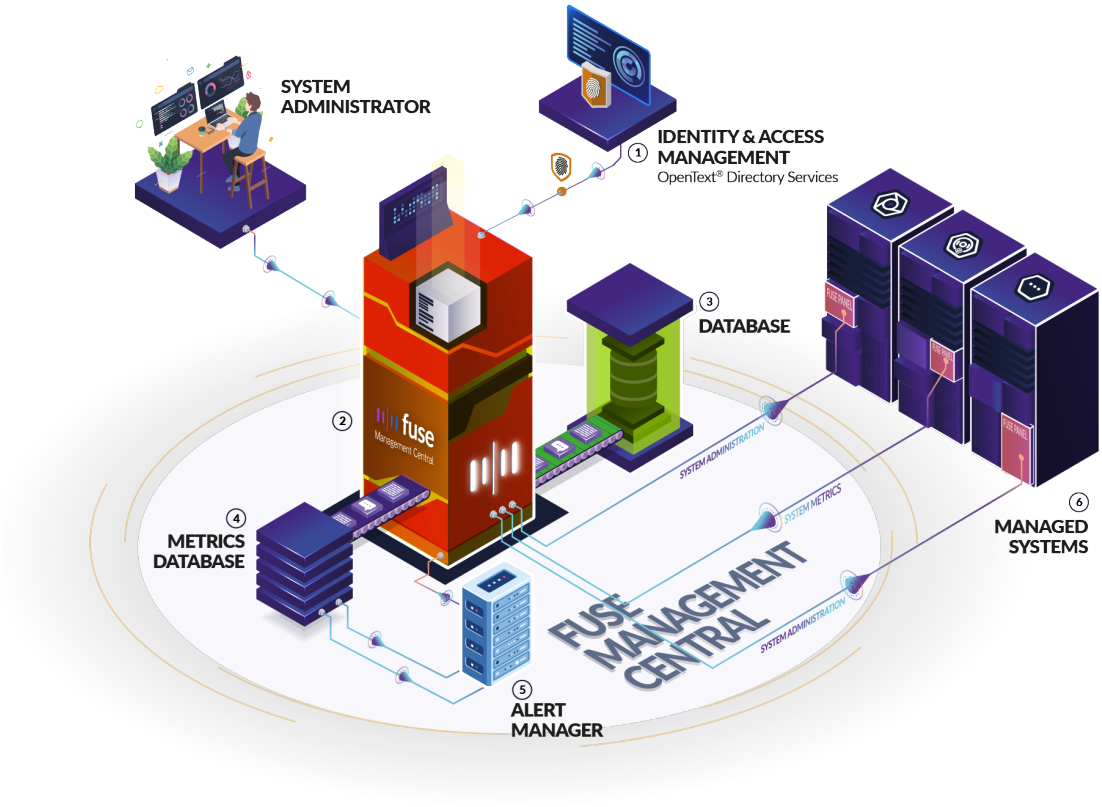
This guide walks you through the installation and administration of Fuse Management Central 1.9.1.

Fuse Management Central is a centralized web administration console for OpenText™ solutions, providing a Unified Management Experience to Self-Managed Customers or Managed Service Providers. With Fuse Management Central intuitive user interface, as well as its simplified deployment, OpenText™ system administrators can efficiently manage components, maintaining the context and understanding of them, while always having the option to schedule any operation.

Fuse Management Central also distinguishes system administration from content administration, introducing an additional layer of security on top of the traditional OpenText™ administration tools.

2. Overview of Fuse Management Central Architecture

The diagram below illustrates and provides a brief description of the conceptual architecture of Fuse Management Central, designed for high performance, scalability and security.



1. Fuse Management Central seamlessly integrates with OpenText™ Directory Services for user management and authentication purposes. OpenText™ Directory Services offers a scalable identity management solution by incorporating various authentication services, such as Active Directory or Google.
2. Fuse Management Central serves as the central orchestrator, acting as the centerpiece that coordinates all system monitoring and management activities, regardless of its cluster type, be it productive or non-productive.
3. Fuse Management Central Database stores all application-related data, such as administration settings, access roles, etc.
4. Fuse Management Central Metrics Database is utilized for long term metric storage, enabling system administrators to conduct temporal searches on system metrics. This functionality combines them into aggregated system metric snapshots over time.
5. Fuse Management Central Alert Manager is responsible for interpreting, deduplicating, grouping, and routing alerts to Fuse Management Central. It also provides the option for silencing and inhibiting alerts.

6. All managed systems must have Fuse Management Client installed and activated. Fuse Management Client is responsible not only for collecting and dispatching metric data from all system components but also for making the system management interface available, ensuring the security of data interchange.

3. Install Fuse Management Central

3.1. Pre-Installation Tasks

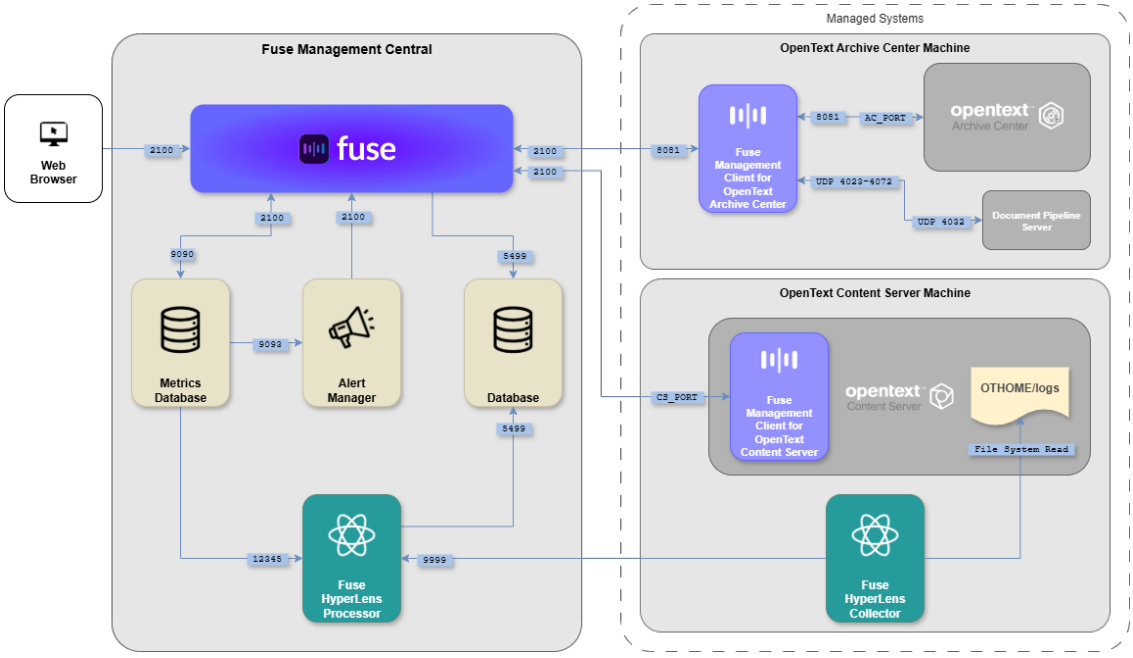
- Review Operating System Support
- Review Hardware Requirements
- Review Communication Ports Availability

> Prior to initiating the Fuse Management Central installation process, it is crucial to refer to the **Release Notes** document for a complete listing of supported systems and compatibility. Failure to do so may result in installation issues or system incompatibility.

3.1.1. Communication Ports Availability

Fuse Management Central relies on defined network ports to enable communication between its internal components and associated services. When a firewall or network security control is present between any of these components, the corresponding ports must be explicitly opened to ensure proper connectivity and system operation.

The diagram below provides a high-level overview of communication paths and port relationships between the main Fuse Management Central components.



The following table enumerates the communication ports required to establish connectivity between the different components.

From	To	Port	Description
External Web Browser	Fuse Management Central	`2100`	User access to Fuse Management Central app
Content Server Machines	Fuse Management Central	`2100`	Fuse Management Client sends metrics to Fuse Management Central
Fuse Management Central	Content Server Machines	`CS_PORT`	Fuse Management Central sends operation requests to Fuse Management Client
Archive Center Machines	Fuse Management Central	`2100`	Fuse Management Client sends metrics to Fuse Management Central
Fuse Management Central	Archive Center Machines	`8081`	Fuse Management Central sends operation requests to Fuse Management Client
Content Server Machines	Fuse Management Central	`9999`	HyperLens Collector sends logs data to HyperLens Processor

! If any of the above ports are being used by other processes or applications, Fuse Management Central will not be able to properly operate.

The connectivity requirements described above are organized by major endpoint. Each of the following sections details the communication requirements for a specific Fuse Management Central component.

Fuse Management Central

The central piece for all communication is the Fuse Management Central service, installed along with its dependency services: Database, Metrics Database and Alert Manager.

- These services should be installed in the same machine and all the communication between them will happen internally.
- Communication with external endpoints will be done through Fuse Management Central service on port `2100`.
    - Fuse Management Central will connect to Archive Center Client on port `8081`.
    - Fuse Management Central will connect to Content Server Client on the `CS_PORT` where the Content Server is available.

- Both Content Server and Archive Center Clients will connect to Fuse Management Central on port `2100`.
- Users will be able to access Fuse Management Central UI on port `2100`.

OpenText Content Server Machines

For all OpenText Content Server systems that will be monitored, the Fuse Management Client for OpenText Content Server should be installed and it will be the connecting point to Fuse Management Central.

- Fuse Management Client for OpenText Content Server is a Content Server module installed inside the Content Server app itself. Because of that, all the communication goes through the usual Content Server port (`CS_PORT`) where the app is running.
- Content Server, and consequently the Fuse Client, should be able to access Fuse Management Central on port `2100`.
- There is no requirement for connection from the Client to the Content Server, since the Client is already running inside the Content Server app itself.
- Fuse Management Central should be able to access Content Server on the `CS_PORT` where the Content Server app is running.

For OpenText Content Server, the Fuse Management Client operates as a module within Content Server, eliminating the need for any special ports to be accessed, other than the actual Content Server port.

OpenText Archive Center Machines

For all OpenText Archive Center systems that will be monitored, the Fuse Management Client for OpenText Archive Center should be installed and it will be the connecting point to Fuse Management Central.

- Fuse Management Client for OpenText Archive Center is a web service installed in the same machine as OpenText Archive Center.
- Fuse Management Client should be able to access Fuse Management Central on port `2100`.
- Fuse Management Central should be able to access Fuse Management Client on port `8081`.
- Fuse Management Client should be able to access OpenText Archive Center on the `AC_PORT` where the Archive Center is available.
- In case Document Pipelines Monitoring will be activated, Fuse Management Client will need UDP access to the Document Pipeline Server on `UDP 4032`. On the inverse side, Document Pipeline Server will need access to Fuse Management Client on `UDP` ports in range `4023-4072`.

Fuse Management Client for OpenText Archive Center should be installed on the same system where Archive Center is running.

HyperLens (Experimental)

HyperLens can be installed as an add-on to the Fuse Management Central architecture. For environments where HyperLens is installed and enabled, the HyperLens components integrate with the Fuse Management Central architecture to collect and process request data.

- HyperLens Collector is installed on the same machine as OpenText Content Server.
- HyperLens Collector should be able to access HyperLens Processor on port `9999`.
- HyperLens Processor is installed on the same machine as Fuse Management Central.
- HyperLens Processor will receive data from HyperLens Collector on port `9999`.
- The Metrics Database should be able to access HyperLens Processor on port `12345`.
- No external access to HyperLens Processor is required beyond the ports explicitly listed above.

3.1.2. Install Java

Prior to initiating the Fuse Management Central installation, please ensure that a supported JDK is already installed.

Please validate your current JDK version:

- *Option 1:* On Windows go to **Control Panel > Programs and Features** to check which JDK version is installed.
- *Option 2:* Verify if the JDK is already installed by opening a command line and entering the following command:

```
java -version
```

Refer to the Fuse Management Central **Release Notes** document to determine the supported JDK versions for your specific Fuse Management Central version. If no JDK is installed or if the installed version is not supported, take the following action: * Download and install Java with the default option selected, ensuring it is available in your system's `path`.

Note that free long-term support (LTS) versions of JDK are available from Adoptium Eclipse Temurin and Oracle. It is highly recommended to install a long-term support version for optimal compatibility with Fuse Management Central.

By default, Fuse Management Central will use the JDK available in the System Path. If you want to use a specific JDK, you will need to edit the file **FuseManagementCentral.xml** after installation and replace the `java` command by the complete path to the JDK you would like to use.

3.1.3. Install NTP (recommended)

To ensure consistent metric data and as a general best practice, it is highly advisable to keep all servers clocks synchronized.

To fulfill this purpose, installing the Network Time Protocol (NTP) is strongly recommended on both the Fuse Management Central server and all of your configured Systems.

NTP helps maintain a consistent time of day across all the service nodes in the cloud. If you enable NTP in a network, ensure that the service nodes are configured to obtain their time over the network.

3.1.4. Enabling SSL

In order to establish a secure communication channel between the user and Fuse Management Central, HTTPS can be used by enabling SSL.

The recommendation is to use a proxy web server, such as NGINX, to redirect all traffic to HTTP port 2100. With this approach, there is no need to change any configuration in Fuse and everything should work as expected.

Optionally, it is possible to enable SSL security directly on Fuse. Please refer to the **Configure SSL** section in the Spring Boot Reference Documentation.

3.1.4.1. Register SSL certificate

Self-signed or internal certificates can be configured to propagate as a truststore for all internal SSL communications, if needed, and as a keystore for exposing the server via HTTPS.

The recommended approach for registering SSL certificates is to configure a `fuse` SSL bundle in the `application.yml` file.

For example, to register a signed certificate with its private Certificate Authority (CA):

```
spring:
  ssl:
    bundle:
      pem:
        fuse:
          # change these values accordingly
          truststore:
            certificate: c:/path/ca.pem
          keystore:
            certificate: c:/path/ca-signed.pem
            private-key: c:/path/private.key
            private-key-password: secret
```

Communications with secured systems, such as OpenText Content Server or OpenText Archive Center, will utilize the up above certificate.

It is also possible to configure either PEM-encoded files or Java keystore files, as well as other trust material. For detailed information, please refer to the **SSL** section in the Spring Boot Reference Documentation.

3.1.4.2. Expose Fuse Management Central as HTTPS server

After registering the SSL certificate, you can enable HTTPS for Fuse Management Central by simply adding the following configuration to the `application.yml` file:

```
server:
  # {product-full-name} HTTPS Port
  port : 8443
  ssl:
    bundle: fuse
```

After this step, it is necessary to update the configurations of the Alert Manager and the Metrics Database to ensure correct communication with Fuse.

To configure the Alert Manager go to `<fuse_installation_folder>/alertManager/alertmanager.yml` and update the URL with HTTPS and the new port.

```
receivers:
  - name: fuse
    webhook_configs:
      - url: 'https://127.0.0.1:8443/api/alert'
        send_resolved: true
        http_config:
          tls_config:
            ca_file: c:/path/ca.pem
            server_name: "localhost"
```

To configure the Metrics Database go to `<fuse_installation_folder>/metricsDatabase/prometheus.yml` and update each scrape configuration with the new URL, adding the property `scheme: https`.

```
scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['127.0.0.1:9090']
  - job_name: 'fuse-spring-boot'
    metrics_path: '/actuator/prometheus'
    static_configs:
      - targets: ['127.0.0.1:8443']
    scheme: https
  - job_name: 'fuse-metrics-5'
    metrics_path: '/api/metrics/5'
    scrape_interval: 5s
    static_configs:
      - targets: ['127.0.0.1:8443']
    scheme: https
  - job_name: 'fuse-metrics-30'
    metrics_path: '/api/metrics/30'
    scrape_interval: 30s
    static_configs:
      - targets: ['127.0.0.1:8443']
    scheme: https
  - job_name: 'fuse-metrics-60'
    metrics_path: '/api/metrics/60'
```

```
    scrape_interval: 60s
    static_configs:
      - targets: ['127.0.0.1:8443']
    scheme: https
  - job_name: 'fuse-metrics-120'
    metrics_path: '/api/metrics/120'
    scrape_interval: 120s
    static_configs:
      - targets: ['127.0.0.1:8443']
    scheme: https
```

3.1.4.3. Self-signed certificate support

If the configured SSL certificate is self-signed, it must be declared explicitly by configuring in the `application.yml` file:

```
spring:
  ssl:
    bundle:
      jks:
        fuse:
          keystore:
            # pointing to a self-signed certificate
            location: c:/path/self-signed.jks
            password: secret
            type: PKCS12

http-client:
  ssl:
    trust-self-signed: true
```

3.1.4.4. Enable SSL in HyperLens

To enable SSL in HyperLens, follow these steps:

1. **Configure SSL in the HyperLens Processor** Update the `config/config.properties` file to enable TLS and provide the paths to the truststore and keystore files:

```
[source,properties]
----
postgres.url=jdbc:postgresql://localhost:5432/postgres
postgres.user=postgres
postgres.password=myPassword
otlp.port=9999
prometheus.port=12345
fuse.window-size=10
```

```
tls.enabled=true
tls.truststore-path=/path/to/truststore.crt
tls.keystore-path=/path/to/keystore.key
----
```

2. **Configure SSL in the HyperLens Collector** Update the `config/fmc.ini` file to enable TLS and provide the paths to the TLS certificate, private key, and CA files:

```
[source,ini]
----
[hyperlens]
HyperLensURL=http://localhost:9999
HyperLensTLS=true
HyperLensCertFile=/path/to/certificate.crt
HyperLensKeyFile=/path/to/private.key
HyperLensCAFile=/path/to/ca.crt
----
```



The `HyperLensCertFile` and `HyperLensKeyFile` properties are required if `HyperLensTLS` is set to `true`. The `HyperLensCAFile` property is required if a Certificate Authority (CA) is needed for the TLS connection.



The `HyperLensURL` property should be updated to reflect the correct URL of the HyperLens Processor, including the HTTPS protocol and the port number, only if Fuse Server (or protocol) is different from the server where HyperLens is.

3.2. Microsoft Windows

3.2.1. Installation

To run Fuse Management Central installer on Windows:

1. Log in to Windows as a user who is a member of the **Local Administrators** group.
2. **Start Fuse Management Central installation** wizard, by double-clicking the installation file (`/Fuse Management Central/Windows/`**`Fuse Management Central1.9.1-Winx64.exe`**).
3. In the **Choose Components** dialog box, leave the default values selected and click **Next**.
4. In the **Choose Install Location** dialog box, accept the default **Destination Folder** or click Browse to select a different folder, and then click **Next**.
5. In the **Choose Data Location** dialog box, accept the default **Data Directory** folder or click Browse to select a different folder, and then click **Next**.



To ensure business continuity, the **Data Directory** path should have a backup policy applied, enabling data recovery in the event of a disaster.

6. In the **Choose Start Menu Folder** dialog box, click **Install**.
7. When the installation process is complete, click **Close**.
8. Open the Windows Services console and start **Fuse Management Central** service. Once started, all dependency services will start automatically.
   The following Windows services must be running:

- **Fuse Management Central**
- **Fuse Management Central (Alert Manager)**
- **Fuse Management Central (Database)**
- **Fuse Management Central (Metrics Database)**

3.2.2. Upgrade

As the PostgreSQL version has been upgraded, it is crucial to take proper precautions before upgrading Fuse Management Central. Failure to do so could result in the loss of all existing PostgreSQL data! To ensure the preservation of your data, before starting the Fuse Management Central upgrade, you must first backup your PostgreSQL data, as described in this chapter. Once the upgrade is complete, you can then restore your data.

If you have a previous version of Fuse Management Central installed, follow the procedures below:

1. Stop the following services:
   a. `Fuse Management Central`.
   b. `Fuse Management Central (Alert Manager)`.
   c. `Fuse Management Central (Metrics Database)`.

   Ensure that only the `Fuse Management Central (Database)` service remains running.

2. Backup your **Fuse Data Directory**.
   a. The Fuse Data Directory is set up according to the installation instructions in the Installation on Microsoft Windows chapter, for example `C:\ProgramData\Fuse Management Central`
3. Create a database backup by creating a dump of all the Postgres databases:
   a. Refer to the official `pg_dumpall` command documentation on how to create a dump of all databases, for example:

```
cd "<fuse_installation_folder>\database\bin"
.\pg_dumpall -U postgres -p 5499 -f c:\\database_backup
```

   The database dump may take several minutes, **so please make sure it completes successfully**.

4. Stop `Fuse Management Central (Database)` service.

5. Delete the `database` folder in **Fuse Data Directory**, For example, `C:\ProgramData\Fuse Management Central\database`



Ensure that you have previously made a proper backup of the folder

6. Uninstall Fuse Management Central following the instructions in the Uninstall Fuse Management Central chapter.
7. Install Fuse Management Central by following the instructions in the Install Fuse Management Central chapter, configuring the same data directory used previously when prompted.
    a. **Do NOT start any service after installation**, only `Fuse Management Central (Database)`
8. Import the Postgres database dump, by running the script generated previously
    a. For example:

```
cd "<fuse_installation_folder>\database\bin"
.\psql.exe -p 5499 -U postgres -d postgres -f
c:\\database_backup
```



This may take several minutes, **so please make sure it completes successfully**.

9. Start `Fuse Management Central` service and make sure all the processes are running (`Fuse Management Central (Alert Manager)`, `Fuse Management Central (Metrics Database)` and `Fuse Management Central (Database)`).

3.3. Linux

Linux and manual installation resources are available in the `Linux` folder inside the Fuse Management Central package.

It is highly recommended to refer to the **Fuse Management Central architecture** chapter and understand the various components that are part of the product. Each of these components should be installed manually.

3.3.1. Pre-requirements

- Install Java 17 or higher
- Install Prometheus 3.5.5 or any latest patch version (https://github.com/prometheus/prometheus/releases/tag/v3.5.5).
- Install AlertManager 0.33.0 or any latest patch version (https://github.com/prometheus/alertmanager/releases/tag/v0.33.0)
- Install PostgreSQL 16.14 or any latest minor version (https://www.postgresql.org/download/), including the **postgresql-contrib** subpackage (https://www.postgresql.org/

docs/16.14/contrib.html)

- Install PostgreSQL **uuid-ossp** extension module (https://www.postgresql.org/docs/16.14/uuid-ossp.html)

Please refer to each third-party component documentation about procedures on how to install them. Alternatively, we provide general guides for third-party software installation in the **Appendix B - How-Tos**.

The next steps of the manual installation will assume that all components are installed on the same machine and are running with the default ports.

- All configurations use 127.0.0.1 or localhost for communication.
- All third-party software is using default ports:
  - PostgreSQL: `5432`
  - Prometheus: `9090`
  - AlertManager: `9093`

For a different setup, please review the configuration files supplied by Fuse Management Central package as you go through each step of the installation:

- `config/application.yml`
- `prometheus_config/prometheus.yml`
- `alertmanager_config/alertmanager.yml`

3.3.2. Installation and Configuration

1. Unzip the `Linux` folder from Fuse Management Central package into the desired installation directory. For example: `/opt/vilt/fuse/`.
2. Update the `config/application.yml` file to validate the datasource configuration, ensuring it matches your PostgreSQL installation. Additionally, add the username and password configuration. For example:

```
spring:
  datasource:
    driver-class-name: org.postgresql.Driver
    url: jdbc:postgresql://localhost:5432/postgres
    username: postgres
    password: myPassword
```

3. Redirect your Prometheus `--config.file` argument in your Prometheus startup script to `<fuse_installation_folder>/prometheus_config/prometheus.yml`.
4. Modify your Prometheus startup script to include the following arguments, alongside any other arguments you may have:

```
--query.max-concurrency=32
```

5. Redirect your AlertManager `--config.file` argument in your AlertManager startup script to `<fuse_installation_folder>/alertmanager_config/alertmanager.yml`.
6. Start Fuse Management Central as a standalone runnable jar:

```
java -jar fuse.jar
```

7. The startup process may take some time to complete. Once finished, you can open Fuse Management Central in your browser:
    - http://localhost:2100/
8. **Optional:** To integrate with **systemd** in *nix systems, Fuse Management Central integrates jsystemd. A sample service unit can be created like this:

    Sample `fuse.service`:

```
[Unit]
Description=Fuse Management Central
Requires=network.target
After=network.target
After=syslog.target
[Service]
Type=notify
WorkingDirectory=<fuse_installation_folder>
ExecStart=/usr/bin/java -jar <fuse_installation_folder>/fuse.jar
SuccessExitStatus=143
KillMode=mixed
TimeoutStopSec=10
TimeoutStartSec=120
[Install]
WantedBy=multi-user.target
```

3.3.3. Upgrade

As the PostgreSQL version has been upgraded, it is crucial to take proper precautions before upgrading Fuse Management Central. Failure to do so could result in the loss of all existing PostgreSQL data! To ensure the preservation of your data, before starting the Fuse Management Central upgrade, you must first backup your PostgreSQL data, as described in this chapter. Once the upgrade is complete, you can then restore your data.

If you have a previous version of Fuse Management Central installed, follow these procedures:

1. Stop all Fuse services: Fuse Management Central, Prometheus, AlertManager and PostgreSQL.
2. Upgrade Prometheus to 3.5.5 version or any latest patch version.
3. Upgrade AlertManager to 0.33.0 version or any latest patch version.

4. Upgrade PostgreSQL to 16.14 version or any latest minor version.

Please refer to each third-party component documentation about procedures on how to upgrade them. Always backup your data before making changes. Alternatively, we provide general guides for third-party software upgrade in the **Appendix B - How-Tos**.

After upgrading all third-party services, please follow the next steps:

1. Update Fuse configuration file with the new one:
    a. Go to your Fuse installation folder.
    b. Inside the `config` folder, back up your `application.yml` file.
    c. If you have not modified the file manually and have default configurations, simply replace the old `application.yml` file with the new one.
    d. If you have custom configurations, review the new application.yml file for any additional configurations and merge them with your current file as needed.
2. Update Prometheus configurations:
    a. Go to your Fuse installation folder.
    b. Inside the `prometheus_config` folder, back up all files and folders.
    c. Replace current configuration files and folders with the new ones.
3. Update AlertManager configurations:
    a. Go to your Fuse installation folder.
    b. Inside the `alertmanager_config` folder, back up all files and folders.
    c. Replace current configuration files and folders with the new ones.

    Please make sure that Prometheus and AlertManager startup scripts are configured to load configurations from your Fuse installation folder. Refer to the installation section for further information on this.

4. Update Fuse Management Central runnable jar with the new one:
    a. Go to your Fuse installation folder.
    b. Back up your current `fuse.jar` file.
    c. Replace your current `fuse.jar` file with the new one.
5. Restart all Fuse services: Prometheus, AlertManager, PostgreSQL and Fuse Management Central.

3.4. Docker

3.4.1. Pre-requirements

Before proceeding, please make sure you have installed the latest version of Docker and Docker Compose as defined in the official documentation:

- Docker: https://docs.docker.com/install/
- Docker Compose: https://docs.docker.com/compose/install/

! Please refer to the **Release Notes** for the required minimum versions. This information is crucial for a successful installation and operation.

3.4.2. Run with Docker Compose

1. Load the docker image with:

```
docker load < fuse-management-central-image-<release-version>.tar
```

This will create an image with the tag `vilt-group/fuse-management-central:<release-version>`.

2. In the same directory as the provided `docker-compose.yml` file, start all the services with:

```
docker-compose up -d
```

After startup, open Fuse Management Central login page in your browser: http://localhost:2100

! Ensure that port `2100` is available on your system. Otherwise, change the exposed port in the `docker-compose.yml`file to one that is available.

3.4.3. Advanced Configuration

3.4.3.1. Data Persistence

Data is persisted in volumes, however, it is recommended to create backups for all volumes. For backup recommendations, please refer to the official Docker documentation.

```
volumes:
  postgresql:
  prometheus:
  alertmanager:
```

Alternatively, you can also map the data directories to filesystem mount points. Please refer to the official documentation for the third-party containers for guidance.

3.4.3.2. JVM Options

To configure advanced JVM options in Fuse service, utilize the `JAVA_TOOL_OPTIONS` environment variable:

```
    environment:
      - JAVA_TOOL_OPTIONS=
```

3.4.3.3. Third-Party Components

For more information regarding third-party configurations, please refer to the official documentation for the respective third-party images:

- Postgres
- Prometheus
- AlertManager

For information on running the Docker image with kubernetes, please contact us via email at product.support@vilt-group.com.

3.4.4. Upgrade

As the PostgreSQL version has been upgraded, it is crucial to take proper precautions before upgrading Fuse Management Central. Failure to do so could result in the loss of all existing PostgreSQL data! To ensure the preservation of your data, before starting the Fuse Management Central upgrade, you must first backup your PostgreSQL data, as described in this chapter. Once the upgrade is complete, you can then restore your data.

To upgrade your Docker installation, first, you need to backup your database using `pg_dumpall`:

1. Stop all containers.

```
docker compose stop
```

2. Perform a backup of the database data.

```
# start the fuse-database, so you can dump it
docker compose start fuse-database

# dump the database (feel free to change the target file to a folder
with enough space available)
docker compose exec fuse-database pg_dumpall -U postgres -p 5432
--no-role-passwords | gzip > db_backup.sql.gz
```

```
# stop fuse-database and remove all containers
docker compose stop fuse-database
docker compose rm
```

3. Perform a backup of your existing `docker-compose.yml` file.
4. Copy the new Docker files from the release bundle (`docker-compose.yml` and `fuse-management-central-image-<release-version>.tar`).
5. Optionally, backup the data from all named volumes by following the instructions provided in the Docker User Guide.
6. Copy the provided `docker-compose.yml` file and make any necessary adjustments as required for your environment.
7. Delete existing `postgres` volume.

```
# find and remove the postgres volume
docker volume ls | grep postgres

# this will output something like:
# local     fuse-server_postgresql

# use the volume name to remove it
docker volume rm fuse-server_postgresql
```

8. Load the Docker image.

```
docker load < fuse-management-central-image-<release-version>.tar
```

9. Start the database and restore the data from the backup.

```
# start the fuse-database to import the dumped file
docker compose up -d fuse-database

# import the dump (replace db_backup.sql.gz with your postgres
backup file)
gunzip -c db_backup.sql.gz | docker compose exec --no-TTY fuse-
database psql -U postgres -p 5432
```

10. Start all the remaining services.

```
docker-compose up -d
```

11. Verify that the new images were pulled correctly and that the containers were recreated successfully.

3.5. Kubernetes and OpenShift Helm Deployment

This section describes how to deploy Fuse Management Central on Kubernetes or OpenShift using the Helm artifacts shipped in the Fuse Management Central package.

3.5.1. What This Chart Deploys

The `fuse-management-central` Helm release deploys:

- `fuse-management-server`: Fuse Management Central application `Deployment`, fixed to one replica.
- `database`: internal PostgreSQL `StatefulSet`, enabled by default and externalizable through values.
- `metrics-database`: mandatory Prometheus `StatefulSet`.
- `alertmanager`: mandatory Alertmanager `Deployment`.
- Supporting `Service`, `ConfigMap`, `Secret`, bootstrap `Job`, and optional `Ingress` and `NetworkPolicy` resources.

3.5.2. Package Artifacts

From the Fuse Management Central package root, use:

- `Deployments/Helm/Fuse Management Central/fuse-management-central-<release-version>.tgz`
- `Deployments/Helm/Fuse Management Central/values-example.yaml`
- `Containers/Fuse Management Central/fuse-management-central-image-<release-version>.tar`



The chart package contains a `README.md` file with chart-specific instructions. The provided `values-example.yaml` file is commented and documents the available values that should be reviewed before installation.

3.5.3. Platform Prerequisites

Before installing the chart, confirm the following requirements:

- A supported Kubernetes or OpenShift cluster is available.
- Helm 3 is installed on the operator workstation or automation runner.
- `kubectl` or `oc` is configured for the target cluster.
- The target namespace is created by the operator or by using Helm `--create-namespace`.
- The Fuse Management Central container image is available to every node that can schedule the application Pod.
- A default `StorageClass`, or explicitly configured storage classes, are available for PostgreSQL and Prometheus PVCs.
- Ingress, route, load balancer, NodePort, or port-forward access is planned for the Fuse

Management Central user interface.

The chart does not create or manage:

- Kubernetes namespaces.
- Cluster-wide RBAC.
- Storage classes.
- Container registries.
- Image import or node image preload processes.
- Ingress controllers, OpenShift Routes, external DNS, TLS certificates, or corporate proxy configuration.

! Review the Fuse Management Central Release Notes before installation for supported platforms and minimum resource requirements.

3.5.4. Prepare the Image

The Fuse Management Central package includes an offline image tar file. Make the image available to the cluster using the method approved for your platform.

For a Docker-backed local or single-node cluster, load the image with:

```
docker load -i "Containers/Fuse Management Central/fuse-management-
central-image-<release-version>.tar"
```

For containerd-based clusters, use the platform-approved import command. Example:

```
nerdctl -n k8s.io load -i "Containers/Fuse Management Central/fuse-
management-central-image-<release-version>.tar"
```

For multi-node clusters, loading the image on only one node is not enough unless scheduling is constrained to that node. Prefer pushing the image to an internal registry and setting the image repository, tag, digest, and image pull secrets in `values-example.yaml`.

3.5.5. Review Values

Open `Deployments/Helm/Fuse Management Central/values-example.yaml` and review at least:

- `global.namespace`
- `global.imagePullSecrets`
- `fuse.image.repository`
- `fuse.image.tag` or `fuse.image.digest`

- `fuse.service.type`
- `fuse.env.restUrl`
- `ingress.*`
- `postgresql.enabled`
- `postgresql.persistence.*`
- `externalDependencies.postgresql.*`
- `prometheus.persistence.*`
- `prometheus.config.retentionTime`
- `prometheus.config.retentionSize`
- `alertmanager.persistence.enabled`
- `networkPolicy.*`



If `global.namespace` is set, it must match the Helm `--namespace` value. The chart fails rendering when these values differ.

If exposing Fuse Management Central through `NodePort`, Ingress, OpenShift Route, reverse proxy, or any platform-specific entrypoint, set `fuse.env.restUrl` to the browser-visible URL before first production use. This value is bootstrapped into the database.

3.5.6. Install

Create or select the namespace:

```
kubectl create namespace fuse-management-central --dry-run=client -o yaml
| kubectl apply -f -
```

OpenShift:

```
oc new-project fuse-management-central
```

Install or upgrade the release from the Fuse Management Central package root:

```
helm upgrade --install fuse-management-central \
  "Deployments/Helm/Fuse Management Central/fuse-management-central-
<release-version>.tgz" \
  --namespace fuse-management-central \
  --create-namespace \
  -f "Deployments/Helm/Fuse Management Central/values-example.yaml"
```

3.5.7. Validate

Check the release and workloads:

```
helm status fuse-management-central -n fuse-management-central
kubectl get pods,svc,deploy,statefulset,pvc -n fuse-management-central
kubectl rollout status deployment/fuse-management-server -n fuse-
management-central
```

OpenShift:

```
helm status fuse-management-central -n fuse-management-central
oc get pods,svc,deploy,statefulset,pvc -n fuse-management-central
oc rollout status deployment/fuse-management-server -n fuse-management-
central
```

Inspect logs if the rollout does not complete:

```
kubectl logs deployment/fuse-management-server -n fuse-management-central
--tail=200
kubectl get events -n fuse-management-central --sort-by=.lastTimestamp
```

For the default `ClusterIP` service, validate browser access with port-forward:

```
kubectl port-forward service/fuse-management-server 2100:2100 -n fuse-
management-central
```

Then open:

```
http://127.0.0.1:2100/
```

3.5.8. Upgrade

Before upgrading:

1. Back up PostgreSQL if using the internal database.
2. Review the new `values-example.yaml` and compare it with your environment-specific values.
3. Confirm that the image for the new version is available to the cluster.
4. Run `helm upgrade --install` with the new chart package and updated values.
5. Validate rollout and application health.

Example:

```
helm upgrade --install fuse-management-central \
  "Deployments/Helm/Fuse Management Central/fuse-management-central-<new-
release-version>.tgz" \
  --namespace fuse-management-central \
  -f "Deployments/Helm/Fuse Management Central/values-example.yaml"
```

StatefulSet PVC fields, including storage class decisions, are not safe ordinary upgrade changes. Treat storage backend changes as a controlled migration.

For Helm-specific issues, see Helm Troubleshooting.

3.6. Validate Fuse Management Central Installation

To confirm if Fuse Management Central was successfully installed, open Fuse Management Central Administration page using one of the following methods:

1. Either on Windows, click **Start**, then navigate to **Programs** > **[Program Folder Name]** (*default: Fuse Management Central*), and click the **Fuse Management Central Administration** shortcut.
2. Or, open the following URL:

```
http://<fuse-management-central-host>:2100/
```

3. Log in with your authentication credentials:
   - **Username**: `fuseadmin` *(default)*
   - **Password**: `fuseadmin` *(default)*
4. Navigate to ***Fuse Administration > Administration***.
5. On Fuse Management Central Administration page, click on **Status**.

If Fuse Metrics Database, Fuse Database and Fuse Alert Manager components are green and healthy, Fuse Management Central was **successfully installed**! Otherwise, please refer to the **Troubleshooting** chapter for guidance.

3.7. Post-installation

After installing Fuse Management Central, please check for possible hotfixes. Hotfixes are cumulative, so you only need to install the latest one. Available hotfixes can be found in the Fuse Management Central repository at https://sw.vilt-group.com/, under the `Fuse Management Central/Hotfixes` folder within the version folder of your Fuse Management Central installation.

Please follow the `README.txt` file that will be available inside the hotfix package to get instructions on how to apply the hotfix, as well as information about the changes introduced.

3.8. Next Steps

Once Fuse Management Central is installed, it is **mandatory to perform a set of initial configurations required for Fuse Management Central to properly and securely operate**.

Post-installation checklist:

- [x] Review the **Security** settings

- Update **General** settings
- Request and apply a valid **License**

4. Install Fuse Management Client

4.1. Install Fuse Management Client for OpenText Content Server

1. Extract the Fuse Management Client for OpenText Content Server ZIP file (`Clients/Fuse Management Client for OpenText Content Server 1.9.1/fuse-management-client-otcs-1.9.1.zip`) outside of the OpenText™ Content Server® installation folder*.
2. Copy all the extracted `fuse-management-client-otcs-1.9.1` folder contents to the `<Content Server home>` directory, overriding the existing `staging` folder.

> ! If you are installing Fuse Management Client on a UNIX/Linux system, ensure that you perform the setup actions with the user who installed OpenText™ Content Server® and runs the Content Server service.

3. Open **Content Server Administration** page in a Web browser.
4. If prompted, enter the Administrator password, and then click **Log-in**.
5. Install or upgrade Fuse Management Client:
    a. If you already have a previous version of Fuse Management Client:
        i. Select:
            - *(OpenText™ Content Server 16.2.5 and below)* **Module Administration > Upgrade Modules**
            - *(OpenText™ Content Server 16.2.6 and above)* **Core System > Module Configuration > Upgrade Modules**
    b. For new installations:
        i. Select:
            - *(OpenText™ Content Server 16.2.5 and below)* **Module Administration > Install Modules**
            - *(OpenText™ Content Server 16.2.6 and above)* **Core System > Module Configuration > Install Modules**
6. From the **Installable Modules** list, install/upgrade **Fuse Management Client** module.
7. After the installation of **Fuse Management Client** module is completed, restart **Content Server**.

> ⚠ For some Content Server versions, particularly earlier ones, we have found that the standard soft-restart is not sufficient to reload all the loaders required for Fuse Management Client. Therefore, we strongly recommend performing a second hard-restart to make sure everything was properly

updated. Please refer to the troubleshooting section for further assistance.

4.1.1. Install Fuse Management Client using an extended Content Server image

When OpenText™ Content Server runs in Kubernetes/OpenShift, you can install the Fuse Management Client module using the standard OpenText™ Content Server image extension process.

The Fuse Management Central release package includes an additional script for this scenario only:

```
Deployments/Helm/Fuse Management Client for OpenText Content
Server/Kubernetes OpenShift/k8s-post-installation.sh
```

This script is not included in the standard Fuse Management Client ZIP or TAR.GZ package. It is intended only for Kubernetes/OpenShift Content Server images, where the Content Server container startup process may restore configuration files after the module files are copied.

1. Set the values used by the commands in this section:

```
NAMESPACE=<content-server-namespace>
CONTENT_SERVER_VERSION=<content-server-version>
FUSE_CLIENT_MODULE_VERSION=<fuse-client-module-version>
REGISTRY=<registry>
REPOSITORY=<repository>
OTCS_BASE_IMAGE="registry.opentext.com/otxecm:${CONTENT_SERVER_VERSI
ON}"
EXTENDED_IMAGE_TAG="${CONTENT_SERVER_VERSION}-
fuse${FUSE_CLIENT_MODULE_VERSION}"
EXTENDED_IMAGE="${REGISTRY}/${REPOSITORY}/otxecm-
fuseclient:${EXTENDED_IMAGE_TAG}"
FUSE_CLIENT_PACKAGE="<path-to-extracted-fuse-client-package>"
BUILD_CONTEXT=<path-to-build-context>/otcs-fuseclient-extend
```

   For example, if the package contains `staging/fuseclient_1_10_0`, set `FUSE_CLIENT_MODULE_VERSION=1_10_0`.

2. Create the image extension build context:

```
rm -rf "$BUILD_CONTEXT"
mkdir -p "$BUILD_CONTEXT/cs_install/modules"
mkdir -p "$BUILD_CONTEXT/custom_config/pre_config_scripts"
mkdir -p "$BUILD_CONTEXT/custom_config/post_config_scripts"

cp <path-to-OpenText-Dockerfile_extend>
"$BUILD_CONTEXT/Dockerfile_extend"

tar -cf
"$BUILD_CONTEXT/cs_install/modules/fuseclient_${FUSE_CLIENT_MODULE_V
ERSION}.tar" \
  -C "$FUSE_CLIENT_PACKAGE" \
  "staging/fuseclient_${FUSE_CLIENT_MODULE_VERSION}"
```

> ! The module archive name, the folder under `staging`, and the image tag should represent the same Fuse Management Client version. Do not build an image tagged as a newer Fuse Client version using an older module archive.

3. Copy `k8s-post-installation.sh` from the release package to both custom configuration hook folders:

```
custom_config/pre_config_scripts/k8s-post-installation.sh
custom_config/post_config_scripts/k8s-post-installation.sh
```

Example

```
cp "Deployments/Helm/Fuse Management Client for OpenText Content
Server/Kubernetes OpenShift/k8s-post-installation.sh" \
  "$BUILD_CONTEXT/custom_config/pre_config_scripts/k8s-post-
installation.sh"

cp "Deployments/Helm/Fuse Management Client for OpenText Content
Server/Kubernetes OpenShift/k8s-post-installation.sh" \
  "$BUILD_CONTEXT/custom_config/post_config_scripts/k8s-post-
installation.sh"
```

4. Ensure the image extension Dockerfile copies both hook files into the Content Server container hook locations:

```
COPY --chmod=0755 ./custom_config/pre_config_scripts/k8s-post-
installation.sh
/opt/opentext/container_files/custom_config/pre_config_scripts/k8s-
post-installation.sh
COPY --chmod=0755 ./custom_config/post_config_scripts/k8s-post-
installation.sh
/opt/opentext/container_files/custom_config/post_config_scripts/k8s-
post-installation.sh
```

5. Build and publish the extended OpenText™ Content Server image.

Kubernetes or external registry example

```
docker build -f "$BUILD_CONTEXT/Dockerfile_extend" "$BUILD_CONTEXT"
\
  --build-arg base_image="$OTCS_BASE_IMAGE" \
  --build-arg customize=true \
  --build-arg system_center_apply_patches=false \
  -t "$EXTENDED_IMAGE"

docker push "$EXTENDED_IMAGE"
```

OpenShift internal registry example

```
oc -n "$NAMESPACE" new-build \
  --binary \
  --name=otxecm-fuseclient \
```

```
  --strategy=docker \
  --to=otxecm-fuseclient:"$EXTENDED_IMAGE_TAG" \
  --build-arg base_image="$OTCS_BASE_IMAGE" \
  --build-arg customize=true \
  --build-arg system_center_apply_patches=false

oc -n "$NAMESPACE" set build-secret --pull bc/otxecm-fuseclient
<opentext-registry-pull-secret>

oc -n "$NAMESPACE" start-build otxecm-fuseclient \
  --from-dir="$BUILD_CONTEXT" \
  --follow
```

6. Configure the OpenText™ Content Server Helm values to use the extended image, then run the normal Helm upgrade for your Content Server release:

```
otcs:
  image:
    source: <registry>/<repository>
    name: otxecm-fuseclient
    tag: "<extended-image-tag>"
```

For the OpenShift internal registry example, the image source is:

```
image-registry.openshift-image-registry.svc:5000/<namespace>
```

7. Apply the updated values to the Content Server Helm release:

```
helm upgrade <content-server-release-name> <content-server-chart-
path-or-reference> \
  -n "$NAMESPACE" \
  -f <content-server-values-file>
```

8. Wait for every Content Server instance to restart and finish the OpenText module configuration flow.

```
oc -n "$NAMESPACE" rollout status statefulset/otcs-admin
oc -n "$NAMESPACE" rollout status statefulset/otcs-frontend
oc -n "$NAMESPACE" rollout status statefulset/otcs-da
oc -n "$NAMESPACE" rollout status statefulset/otcs-backend-search
```

9. Validate the module from the Content Server Administration page.

A build failure containing `Module did not extract to the staging folder correctly!` usually means that the module archive or build context is not in the layout expected by the OpenText `Dockerfile_extend` flow. Check that `cs_install/modules/fuseclient_<version>.tar` contains `staging/fuseclient_<version>/…` and that the version used in the archive name, staging folder, and image tag is consistent.

4.1.2. Install Fuse Management Client using an OpenText init container

OpenText™ Content Server also supports installing modules in Kubernetes/OpenShift through an init container image. Use this option when your deployment process should keep the original Content Server image unchanged and deliver module files at deployment time.



Use either the extended image approach or the init container approach for a given deployment. Do not configure both at the same time.

The init container image must be built according to the OpenText™ Content Server container deployment documentation, using the OpenText-provided `Dockerfile_init`. The Fuse Management Client module archive must be placed in the module extension folder expected by that Dockerfile.

1. Set the values used by the commands in this section:

```
NAMESPACE=<content-server-namespace>
FUSE_CLIENT_MODULE_VERSION=<fuse-client-module-version>
REGISTRY=<registry>
REPOSITORY=<repository>
INIT_IMAGE_TAG=<tag>
INIT_IMAGE="${REGISTRY}/${REPOSITORY}/otcs-fuseclient-
init:${INIT_IMAGE_TAG}"
FUSE_CLIENT_PACKAGE="<path-to-extracted-fuse-client-package>"
INIT_BUILD_CONTEXT=<path-to-build-context>/otcs-fuseclient-init
```

2. Create the init container build context:

```
rm -rf "$INIT_BUILD_CONTEXT"
mkdir -p "$INIT_BUILD_CONTEXT/extensions/modules"

cp <path-to-OpenText-Dockerfile_init>
"$INIT_BUILD_CONTEXT/Dockerfile_init"

tar -cf
"$INIT_BUILD_CONTEXT/extensions/modules/fuseclient_${FUSE_CLIENT_MOD
ULE_VERSION}.tar" \
  -C "$FUSE_CLIENT_PACKAGE" \
  "staging/fuseclient_${FUSE_CLIENT_MODULE_VERSION}"
```



The module file used by the init container must be a `.tar` archive in the OpenText module extension folder. Do not extract the module contents inside that folder.

3. Build and publish the init container image to a registry that is accessible by every Content Server Pod.

Kubernetes or external registry example

```
docker build -f "$INIT_BUILD_CONTEXT/Dockerfile_init"
"$INIT_BUILD_CONTEXT" \
  -t "$INIT_IMAGE"

docker push "$INIT_IMAGE"
```

OpenShift internal registry example

```
oc -n "$NAMESPACE" new-build \
  --binary \
  --name=otcs-fuseclient-init \
  --strategy=docker \
  --to=otcs-fuseclient-init:"$INIT_IMAGE_TAG"

oc -n "$NAMESPACE" start-build otcs-fuseclient-init \
  --from-dir="$INIT_BUILD_CONTEXT" \
  --follow
```

4. Configure the OpenText™ Content Server Helm values to enable extensions and reference the init container image:

```
otcs:
  config:
    extensions:
      enabled: true
  initContainers:
    - name: otcs-fuseclient-init
      image:
        source: <registry>/<repository>
        name: otcs-fuseclient-init
        tag: "<init-image-tag>"
```

5. Ensure the Content Server workload has pull access to both the regular OpenText™ Content Server image and the init container image.

> **!** If the regular Content Server image and the init container image are stored in different registries, make sure the Helm values include all required image pull secrets before running `helm upgrade`. Do not patch generated StatefulSets manually for permanent installations, because later Helm upgrades can overwrite those changes.

6. Apply the Helm upgrade using the same values strategy used by your existing Content Server deployment. .Example

```
helm upgrade <content-server-release-name> <content-server-chart-
path-or-reference> \
  -n "$NAMESPACE" \
  -f <content-server-values-file>
```

7. Wait for every Content Server instance to restart and finish the OpenText module

configuration flow.

```
oc -n "$NAMESPACE" rollout status statefulset/otcs-admin
oc -n "$NAMESPACE" rollout status statefulset/otcs-frontend
oc -n "$NAMESPACE" rollout status statefulset/otcs-da
oc -n "$NAMESPACE" rollout status statefulset/otcs-backend-search
```

8. Validate the module from the Content Server Administration page.

4.1.3. (Optional) Install Fuse Management Client for OpenText Content Server using Opentext System Center Manager

Alternatively, the Fuse Management Client 1.9.1 module can also be deployed using OpenText System Center Manager (OTSCM):

1. Upload Fuse Management Client ZIP file:
    a. Open OTSCM, navigate to **Settings** and on the left menu choose **External Vendor Files**.
    b. In the header row labeled **Available Files**, there is a button on the right side named **Upload** that allows you to upload a new file.
    c. Upload Fuse Management Client ZIP file (`Clients/Fuse Management Client 1.9.1/`**`fuse-management-client-otcs-1.9.1.zip`**).
2. Create an installation plan for Fuse Management Client:
    a. Navigate to the **Plans** tab and use the button **Add Item** to create a new plan for installing Fuse Management Client.
    b. Add each configured system where fuse-management-client-otcs-1.9.1 is to be installed, and for each one of them configure the required attributes:

Field	Description
Module Vendor	Third Party
Thirt party Module	Previously uploaded Fuse Client ZIP (e.g. *"fuse-management-client-otcs-1.9.1.zip"*)
Instance path	Path to OpenText Content Server installation
Admin username	OpenText Content Server Admin user
Host name	OpenText Content Server hostname

Field	Description
Site name	OpenText Content Server site name (configured in mappings.tbl)

c. Save the plan.

3. Execute plan:

a. The plan can be executed by pressing the **play** button under actions.

4.1.4. Configure Fuse Management Client logs

The Fuse Management Client module has its own log configurations, which can be changed according to your needs.

In order to change Fuse Management Client log configurations, follow these steps:

1. Open **Content Server Administration** page in a Web browser.
2. If prompted, enter the Administrator password, and then click **Log-in**.
3. Access **Fuse System Administration > Log Settings**.
4. Change the settings accordingly to your needs:

Field	Description
Log Level	Desired log level (OFF, ERROR, WARN, INFO, DEBUG, TRACE)
Location	Path where logs should be stored
Use rolling logs	Create a rotation mechanism for the log file
Number of log files	How many files should be stored after rotation
Size of each log file	Size of the log file to be rotated
Compress completed log files	If rotated log files should be compressed

5. Click on **Save Changes**.

4.1.5. Patching Fuse Management Client for OpenText Content Server

To complete the installation of Fuse Management Client or resolve known issues, you may need to apply one or more Content Server patch files.

Fuse Management Client patches are distributed as common OpenText Content Server patch files. Available patch files can be found in the Fuse Management Central repository https://sw.vilt-group.com/, under the `Clients/Fuse Management Client for OpenText Content Server x.x.x/Patches` folder of each Fuse Management Central version folder. Additionally, patches can be sent directly to you in support cases.

To apply Fuse Management Client patches follow the steps below:

1. **Download** Fuse Management Client patch files for OpenText Content Server, for example `pat140000001.txt`.
2. **Stop** OpenText Content Server service.
3. **Copy** the patch file(s) to the OpenText™ Content Server® patches folder (`<Content_Server_Home>/patch/`).
4. **Start** OpenText Content Server service.

! In the absence of patch files available for your Fuse Management Client version, it is imperative to note that there is no requirement to apply any patches.

4.1.6. Custom SSL Configuration

4.1.6.1. Overview

It is possible to enable and configure SSL (including optional mutual TLS) directly in your `fmc.ini` file. This allows secure HTTPS connections.

4.1.6.2. Configuration

Add the following section to your `fmc.ini`:

```
[ssl]
IsEnabled=true
Timeout=200
DefaultKeystoreFolderPath=/etc/ssl/certs
PemFilePath=/etc/ssl/certs/cert.pem
PrivateKeyFilePath=/etc/ssl/certs/cert.key
```

4.1.6.3. Parameters

Parameter	Description
`IsEnabled`	Enables or disables `custom SSL configuration` for connections. If set to `false`, the system will follow the default certificate validation process provided by the OpenText Content Server.
`Timeout`	Request timeout in seconds (default: 200, minimum: 15).
`DefaultKeystoreFolderPath`	Path to a keystore folder containing trusted certificates for validation. **Linux**: If set to a valid folder path, it will be used for certificate validation. If empty or not defined, the system certificate store will be used automatically. **Windows**: Ignored (system store is always used).
`PemFilePath`	Path to the client certificate in PEM format (for mutual TLS). Omit this key if mutual TLS is not required.
`PrivateKeyFilePath`	Path to the client private key in PEM format. Omit this key if mutual TLS is not required or if the private key is bundled with the certificate.

Once configured, the system will automatically use these values for secure HTTPS connections.

Any changes to the `[ssl]` section require restarting the OpenText Content Server to take effect. Avoid disabling server certificate validation in production environments.

Do not set parameters that are not required for your environment, simply omit them from `fmc.ini`. On Linux, if `DefaultKeystoreFolderPath` is empty or not defined, the system certificate store will be used automatically.

4.2. Install Fuse Management Client for OpenText Archive Center

Before starting the Fuse Management Client for OpenText Archive Center installation process, please verify if a supported Java version is already installed by following the next steps:

1. Check the Fuse Management Central **Release Notes** document to determine which Java versions are supported for your specific Fuse Management Central version.

2. If Java is not installed or if the installed version is not supported:
    - Download and install a supported Java version, ensuring that it is added to your system's PATH, on the same host where OpenText Archive Center is installed.



Fuse Management Client for OpenText Archive Center must be installed on the same host as OpenText Archive Center.



Free long term support (LTS) versions of JDK are provided by Adoptium Eclipse Temurin and Oracle. We strongly recommend installing a long term support version to ensure compatibility and stability with Fuse Management Central.

4.2.1. User Requirements

User permissions in OpenText Archive Center directly influence the monitoring and management actions available in Fuse Management Central. These permissions are determined by the credentials used to connect Fuse Management Client with OpenText Archive Center.

Monitoring

In order to have full monitoring metrics, it is recommended to use a user from the **aradmins** group. While users from other groups might also work, they may have limited access to certain data.

Management

In order to access all available actions in Fuse Management Central, it is necessary to use the **dsadmin** user, since Fuse Management Client uses both API and also `dsclient` and `spawncmd` calls to perform these actions.

Users from **aradmins** can perform API actions but are unable to execute `dsclient` calls. Consequently, actions performed by `dsclient` will not be successful for these users. Below is the current list of actions performed by Fuse Management Central using `dsclient`, which require the use of **dsadmin** user specifically:

- Delete OpenText Archive Center Disk Volume.

Users from other groups may work, but their access to API actions is limited.

4.2.2. Installation on Microsoft Windows

1. Extract Fuse Management Client for OpenText Archive Center ZIP file (`Clients/Fuse Management Client for OpenText Archive Center 1.9.1/`**`fuse-management-client-otac-1.9.1-windows.zip`**) to the desired installation folder (e.g. C:\Program Files\Fuse Management Client for OpenText Archive Center).
2. Navigate to the new folder location and execute the batch file named `install.bat`. This will install the client in the current location and add it as a Windows service.
3. When the installation process is complete, if needed, close the console window.

4. Check the client settings in the `application.yml` file, following the specifications outlined in Fuse Management Client for OpenText Archive Center Configuration.
5. Open the Windows Services console and start the following Windows service:
    - **Fuse Management Client for OpenText Archive Center**

Optional: The path to the Java executable can be configured in the file `FuseClientArchiveCenter.xml` by editing the value of the tag `<executable>java</executable>`. This is optional and can be useful when you have multiple JREs installed and need to select a specific one.

4.2.3. Installation on Linux

On Unix Systems, installation supports both Init.d and Systemd to start the process as a service. Alternatively, there is a script available to start the process manually in the background as a daemon.

4.2.3.1. Systemd

1. Extract Fuse Management Client for OpenText Archive Center ZIP file (`Clients/Fuse Management Client for OpenText Archive Center 1.9.1/`**`fuse-management-client-otac-1.9.1-unix.zip`**) to the folder `/opt/vilt/fuse-client-otac/` (The installation instructions assume this installation path).
2. Copy the script located at `/opt/vilt/fuse-client-otac/bin/fuse-client-otac.service` to the folder `/etc/systemd/system/` (If the path differs from the previous step, ensure to update this script to reflect the correct path).
3. The script allows configuration of both the application path (default: /opt/vilt/fuse-client-otac/) and the user running the service (default: root) by editing it.
4. Reload the available services by running the command: `sudo systemctl daemon-reload`.
5. To start the application, use the command: `sudo systemctl start fuse-client-otac`.

To enable automatic startup of the application on system boot, use the command: `systemctl enable fuse-client-otac`.

For additional configuration options, please refer to the **Spring Boot systemd Service documentation** official website.

4.2.3.2. Init.d

1. Extract Fuse Management Client for OpenText Archive Center ZIP file (`Clients/Fuse Management Client for OpenText Archive Center 1.9.1/`**`fuse-management-client-otac-1.9.1-unix.zip`**) to the folder `/opt/vilt/fuse-client-otac/` (The installation instructions assume this installation path).

2. Create a symlink, as follows:

```
sudo ln -s /opt/vilt/fuse-client-otac/fuse-client-otac.jar
/etc/init.d/fuse-client-otac
```

3. To start the application, use the command:

```
service fuse-client-otac start
```


You can also flag the application to start automatically by using your standard operating system tools. For example, on Debian, you could use the following command: `update-rc.d fuse-client-otac defaults <priority>`.

For additional configuration options, please refer to the Spring Boot systemd Service documentation official website.

4.2.3.2.1. Daemon

1. Extract Fuse Management Client for OpenText Archive Center ZIP file (`Clients/Fuse Management Client for OpenText Archive Center 1.9.1/`**`fuse-management-client-otac-1.9.1-unix.zip`**) to the folder `/opt/vilt/fuse-client-otac/` (The installation instructions assume this installation path).
2. Change to the script directory:

```
cd /opt/vilt/fuse-client-otac/bin/
```

3. Execute the script to run the client in background:

```
./startup.sh
```

4.2.3.3. Fuse Management Client for OpenText Archive Center Configuration

Check the client settings in the `application.yml` file, following the specifications outlined in Fuse Management Client for OpenText Archive Center Configuration.

4.2.4. Kubernetes and OpenShift Helm Deployment

This section describes how to deploy Fuse Management Client for OpenText Archive Center on Kubernetes or OpenShift using the Helm artifacts shipped in the Fuse Management Central package.

This installation is optional. Use it only when OpenText Archive Center is deployed on

Kubernetes/OpenShift and must be monitored or managed.

4.2.4.1. What This Chart Deploys

The `fuse-management-client-for-otac` Helm release deploys:

- One `Deployment` named `fuse-management-client-for-otac`.
- One `Service` for the OTAC Client HTTP endpoint.
- One `ConfigMap` containing the rendered `application.yml` and the seeded Kubernetes-only `initial-data.json`.
- One `ServiceAccount` when `global.serviceAccount.create=true`.
- Namespace-scoped `Role` and `RoleBinding` resources for OTAC workload discovery and approved OTAC utility execution.
- PVC-backed OTAC Client state, enabled by default and expected for supported deployments.
- Optional PVC-backed OTAC Client log persistence.
- Optional `Ingress`.
- Optional `NetworkPolicy`.

4.2.4.2. Package Artifacts

From the Fuse Management Central package root, use:

- `Deployments/Helm/Fuse Management Client for OpenText Archive Center/fuse-management-client-for-otac-<release-version>.tgz`
- `Deployments/Helm/Fuse Management Client for OpenText Archive Center/values-example.yaml`
- `Containers/Fuse Management Client for OpenText Archive Center/fuse-management-client-for-otac-image-<release-version>.tar`

The chart package contains a `README.md` file with chart-specific instructions. The provided `values-example.yaml` file is commented and documents the available values that should be reviewed before installation.

4.2.4.3. Platform Prerequisites

Before installing the chart, confirm the following requirements:

- OpenText Archive Center is already deployed and reachable from inside the target namespace.
- The OTAC Client will be installed in the same namespace as OpenText Archive Center.
- Helm 3 is installed on the operator workstation or automation runner.

- `kubectl` or `oc` is configured for the target cluster.
- The OTAC Client image is available to every node that can schedule the OTAC Client Pod.
- `archiveCenter.url` points to the OpenText Archive Center service URL reachable from the OTAC Client Pod.
- If automatic service discovery cannot resolve the OTAC service from `archiveCenter.url`, `clusterClient.target.serviceName` is configured.
- If the OTAC pod has more than one container or the target cannot be inferred safely, `clusterClient.target.containerName` is configured.
- If OTAC log collection is required, the OTAC logs PVC exists and is mountable by the OTAC Client Pod.
- The Kubernetes/OpenShift Metrics API is available through `metrics.k8s.io` to collect OTAC workload CPU and RAM metrics.

The chart does not create or manage:

- Kubernetes namespaces.
- Cluster-wide RBAC.
- Storage classes.
- Container registries.
- Image import or node image preload processes.
- Kubernetes Metrics Server or OpenShift monitoring stack.
- OpenText Archive Center itself.



Review the Fuse Management Central Release Notes before installation for supported platforms and minimum resource requirements.

4.2.4.4. Prepare the Image

The Fuse Management Central package includes an offline image tar file. Make the image available to the cluster using the method approved for your platform.

For a Docker-backed local or single-node cluster, load the image with:

```
docker load -i "Containers/Fuse Management Client for OpenText Archive
Center/fuse-management-client-for-otac-image-<release-version>.tar"
```

For containerd-based clusters, use the platform-approved import command. Example:

```
nerdctl -n k8s.io load -i "Containers/Fuse Management Client for OpenText
Archive Center/fuse-management-client-for-otac-image-<release-
version>.tar"
```

For multi-node clusters, loading the image on only one node is not enough unless scheduling is constrained to that node. Prefer pushing the image to an internal registry and setting the image repository, tag, digest, and image pull secrets in `values-example.yaml`.

4.2.4.5. RBAC Requirements

The OTAC Client chart creates namespace-scoped `Role` and `RoleBinding` resources for the supported `k8s` runtime path. It does not create `ClusterRole` resources and does not access other namespaces.

The required namespace-scoped permissions are:

API group	Resource	Verbs
core	`pods`	`get`, `list`
core	`pods/exec`	`get`, `create`
core	`services`	`get`, `list`
core	`endpoints`	`get`, `list`
`discovery.k8s.io`	`endpointslices`	`get`, `list`
core	`persistentvolume claims`	`get`, `list`
`metrics.k8s.io`	`pods`	`get`, `list`

4.2.4.6. Review Values

Open `Deployments/Helm/Fuse Management Client for OpenText Archive Center/values-example.yaml` and review at least:

- `global.namespace`
- `global.imagePullSecrets`
- `client.image.repository`
- `client.image.tag` or `client.image.digest`
- `client.service.*`
- `archiveCenter.url`
- `clusterClient.target.serviceName`
- `clusterClient.target.containerName`

- `otacLogs.enabled`
- `otacLogs.existingClaim`
- `statePersistence.*`
- `logsPersistence.*`
- `documentPipelines.*`
- `ingress.*`
- `networkPolicy.*`

The default OTAC namespace in the values example is `opentext`. Change it if your OpenText Archive Center deployment uses another namespace.

If Document Pipelines integration is enabled, the platform must allow UDP traffic from the OTAC Client Pod to the Document Pipelines host and return UDP traffic to the configured client callback port range.

4.2.4.7. Install

Create or select the namespace where OpenText Archive Center is already deployed:

```
kubectl create namespace opentext --dry-run=client -o yaml | kubectl apply
-f -
```

OpenShift:

```
oc new-project opentext
```

Install or upgrade the OTAC Client release from the Fuse Management Central package root:

```
helm upgrade --install fuse-management-client-for-otac \
  "Deployments/Helm/Fuse Management Client for OpenText Archive
Center/fuse-management-client-for-otac-<release-version>.tgz" \
  --namespace opentext \
  --create-namespace \
  -f "Deployments/Helm/Fuse Management Client for OpenText Archive
Center/values-example.yaml"
```

4.2.4.8. Validate

Check the release and workload:

```
helm status fuse-management-client-for-otac -n opentext
kubectl get pods,svc,deploy,pvc,role,rolebinding -n opentext
kubectl rollout status deployment/fuse-management-client-for-otac -n
opentext
```

OpenShift:

```
helm status fuse-management-client-for-otac -n opentext
oc get pods,svc,deploy,pvc,role,rolebinding -n opentext
oc rollout status deployment/fuse-management-client-for-otac -n opentext
```

Validate ServiceAccount permissions:

```
kubectl auth can-i get pods \
  --as system:serviceaccount:opentext:fuse-management-client-for-otac \
  -n opentext

kubectl auth can-i create pods/exec \
  --as system:serviceaccount:opentext:fuse-management-client-for-otac \
  -n opentext

kubectl auth can-i get pods.metrics.k8s.io \
  --as system:serviceaccount:opentext:fuse-management-client-for-otac \
  -n opentext
```

For the default `ClusterIP` service, validate OTAC Client application access with port-forward:

```
kubectl port-forward service/fuse-management-client-for-otac 8081:8081 -n
opentext
```

Then open the OTAC Client URL:

```
http://127.0.0.1:8081/
```

Log in with a user and password that have access to OpenText Archive Center. A successful login confirms that the OTAC Client application is reachable and can authenticate against the configured OTAC environment.

4.2.4.9. Register the OTAC Client in Fuse Management Central

After the OTAC Client is running, add it to Fuse Management Central as an OpenText Archive Center system.

Use the URL that Fuse Management Central can reach. Examples:

- Internal cluster URL: `http://fuse-management-client-for-otac.opentext.svc.cluster.local:8081`
- Ingress or route URL: `https://otac-client.example.com`
- Port-forward URL for validation only: `http://127.0.0.1:8081`

For production, use a stable service, ingress, route, or load balancer entrypoint. Do not use port-forward URLs for permanent system registration.

4.2.4.10. Upgrade

Before upgrading:

1. Review the new `values-example.yaml` and compare it with your environment-specific values.
2. Confirm that the image for the new version is available to the cluster.
3. Run `helm upgrade --install` with the new chart package and updated values.
4. Validate rollout and application health.

> ! If you are upgrading from a version **prior to 1.9.1**, a security update to the encryption library used by this client means it can no longer read the Archive Center password saved by your previous version. **After the rollout completes, in the Fuse UI go to Edit System for this OTAC system and re-enter the Archive Center password once**, so it gets sent to the client and stored under the new encryption. If you are already running 1.9.1 or later, this was already done and no further action is needed. No other configuration, activation status, or history is affected.

4.2.4.11. Troubleshooting

For Helm-specific issues, see Helm Troubleshooting.

4.2.5. Upgrade Fuse Management Client for OpenText Archive Center

If you currently have a version of Fuse Management Client for OpenText Archive Center installed, you can easily upgrade it following these steps:

1. **Stop** the Fuse Management Client for OpenText Archive Center service.
2. Backup your Fuse Management Client for OpenText Archive Center installation folder.
3. Extract Fuse Management Client for OpenText Archive Center ZIP file (`Clients/Fuse Management Client for OpenText Archive Center 1.9.1/fuse-management-client-otac-1.9.1-[windows/unix].zip`) and copy all files inside `**fuse-client-otac-1.9.1** directory to your existing Fuse Management Client for OpenText Archive Center installation folder, replacing current files with the new ones.
4. Review your existing configuration file for any custom configurations you might have and apply them on the new `application.yml` file.
5. Check for new configurations that should be added to the `application.yml` file, following the specifications outlined in Fuse Management Client for OpenText Archive Center Configuration.
6. **Start** the Fuse Management Client for OpenText Archive Center service.

> ! If you are upgrading from a version **prior to 1.9.1**, a security update to the encryption library used by this client means it can no longer read the Archive Center password saved by your previous version. **After starting the service, in the Fuse UI go to Edit**

System for this OTAC system and re-enter the Archive Center password once, so it gets sent to the client and stored under the new encryption. If you are already running 1.9.1 or later, this was already done and no further action is needed. No other configuration, activation status, or history is affected.

4.2.6. Failover Cluster Scenario for Fuse Management Client for OpenText Archive Center

4.2.6.1. How the Client Works (Architecture)

To successfully deploy the Fuse Management Client for OpenText Archive Center in a failover clustered environment, it is important to understand how the client manages its state and identity within the broader architecture.

- **Standalone Agent:** The Fuse Management Client operates as an independent, standalone application residing on the OpenText Archive Center machine. It acts as a crucial bridge, communicating locally with the OpenText Archive Center and Document Pipeline components, while maintaining a secure, external connection to the Fuse Management Central server.
- **State Persistence:** Because it acts as this bridge, the client is highly stateful. It persists all of its runtime data, credentials, and routing configurations locally in a file named `data.json`.
- **Identity Data Persistence:** This `data.json` file acts as the core registry for the client's unique system identity and runtime parameters. It contains:
    - The unique `systemId` and encrypted `secretKey` used to authenticate with Fuse Management Central.
    - Connection URLs and encrypted credentials.
    - User-defined directory paths for the Archive Center binaries, Spawner, and log folders.
- **Supported Cluster Architectures:** The client is fully compatible with **Active-Passive (Failover)** scenarios. However, it is **NOT compatible with Active-Active (Load Balancing)** scenarios, as multiple active clients cannot simultaneously share the same unique identity (`systemID` and `secretKey`).

4.2.6.2. The Failover Concept

In a failover scenario, the cluster contains multiple physical nodes (e.g., Node A and Node B), but they represent one single logical Archive Center.

If every node maintained its own local `data.json` file, a failover event would cause the newly active node to behave like a completely brand-new, unregistered system. The Fuse Management Central server would reject its connections, breaking the bridge until it was manually reactivated.

To prevent this, the `data.json` file **must** be physically moved to a highly available, **shared storage location** (such as the shared cluster disk). By instructing every node to read from this

single, shared `data.json`, whichever node becomes active will instantly assume the exact same state, identity, security keys, and configurations. The transition happens seamlessly without requiring manual intervention.

4.2.6.3. Configuration Steps

Follow these steps to configure the client for an active-passive cluster:

Step 1: Initial Activation on the Primary Node

You only need to activate the system **once**.

1. Start the Fuse Management Client service on your primary node.
2. Access the Fuse Management Central web interface to register and activate the Fuse Archive Center client system.
    - The activation must be done using the IP/Hostname of the Cluster, and not the IP/Hostname of the client node.
3. Within the Fuse Archive Center client web interface, fully configure all necessary directory paths (OpenText Archive Center Binaries folder, Spawner Bin folder, and Log folders).
    - The directory paths must be identical on all nodes. If the physical folder structure differs between Node A and Node B, the client will fail to locate them after a failover.
4. This will fully populate the local `data.json` file with your paths and unique identity data.



Do not activate the system again on your secondary/passive nodes. The activation must only happen once on the primary node to generate the master `data.json` configuration file.

Step 2: Relocate the Configuration File

1. Stop the Fuse Management Client service on the primary node.
2. Locate the newly populated `data.json` file in your client installation directory.
3. Move this `data.json` file to your highly available, shared cluster storage location (for example, the shared `E:\` drive that moves between nodes during a failover).

Step 3: Update `application.yml` Across All Nodes

Now that the master `data.json` file is sitting on the shared drive, you must instruct every Fuse Management Client node in the cluster to point to this exact location.

1. On **every** node in your cluster, navigate to the client installation folder and open the `application.yml` file.
2. Under the `archive-center:` section, add the `datafile` property, pointing it to the absolute path of the shared drive location.

```
archive-center:
  # Archive Center Instance URL
  url: http://localhost:8080
  # Path to the shared configuration file
  datafile: 'file:///E:/data.json'
```

3. Save the `application.yml` file.
4. Restart the Fuse Management Client service.

Once configured on all nodes, your failover system is ready. When a failover occurs and the secondary node boots up, it will automatically read the shared `data.json`, instantly validating its connection to the Fuse Server without the need to do anything else, with all base functionalities available.

4.2.7. Additional Settings

In this section, you will find a comprehensive list of available settings for the Fuse Management Client for OpenText Archive Center, along with their respective functionalities. These configurations are customizable to accommodate infrastructure requirements and can be accessed and modified in the `application.yml` file located in the client installation folder:

```
archive-center:
  # Archive Center Instance URL
  url: http://localhost:8080
  # Document Pipelines configuration
  document-pipelines:
    host: localhost
    port: 4032
    timeout: 15s
    client-ports:
      offset: 4023
      range: 50

server:
  # Fuse Management Client HTTP Port
  port : 8081
```

- **`url`**: the OpenText Archive Center application URL.
- **`host`**: the hostname or ip from the server.
- **`port`**: the port for the Document Pipelines is 4032 by default.
- **`timeout`**: the timeout that will be used in the calls performed by the client.
- **`client-ports`**: the port for Fuse Client to receive requests from the Document Pipeline Server.
- **`offset`**: the desired port number for communication.
- **`range`**: the range parameter determines the range of ports available for communication, with the starting point being the offset port previously defined.
- **`port`**: the Fuse Management Client port, which will be utilized by the Fuse Management Central to establish a connection, e.g. `8081`.

Please note that by default the Fuse Management Client port will be 8081.

Please note that each Fuse Management Client for OpenText Archive Center instance can only be connected to one OpenText Document Pipeline Server.

You must open the specified `offset` port range in your Fuse Client system **firewall** to enable the asynchronous UDP communication between the Document Pipeline Server and the Fuse Client. This is necessary because the Fuse Client dynamically opens ports randomly allocated within the specified range.

4.2.7.1. OpenText Archive Center URL

Please insert the URL of the OpenText Archive Center application into the `[url]` property under the `[archive-center]` section. This URL will be utilized by the Fuse Management Client to establish a connection to OpenText Archive Center and should follow this format: `http://[otac.server.host]:[otac.server.port]`.

4.2.7.2. Fuse Management Client for OpenText Archive Center Port

Please insert the port of the Fuse Management Client application into the `[port]` property under the `[server]` section. This port will be used when adding the system to Fuse Management Central. For example, in Fuse Management Central, the URL for the Fuse Management Client will be `http://[fuse.otac.client.server.host]:[fuse.otac.client.port]`.

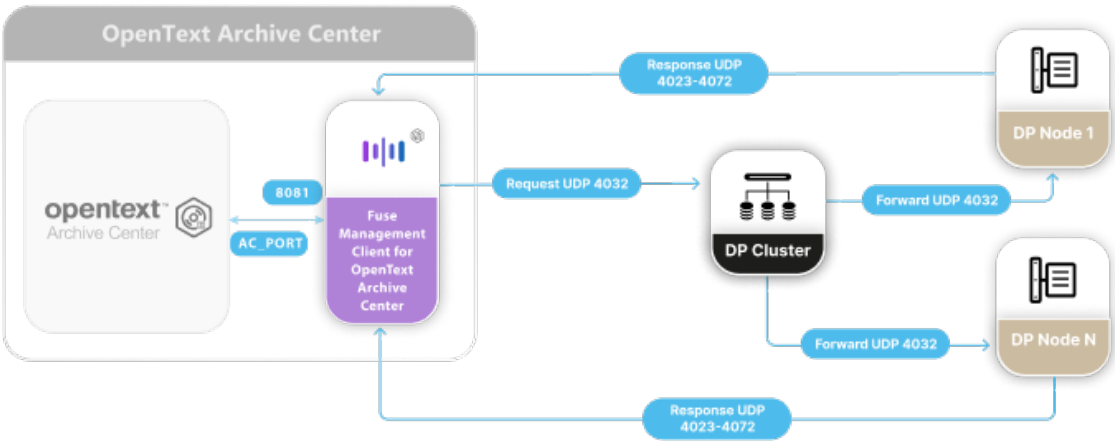
4.2.7.3. Configuration Requirements for Fuse Management Client for OpenText Archive Center UDP Ports

Fuse Management Client for OpenText Archive Center operates using UDP communication, similar to the Document Pipeline Info. This means that all communication is asynchronous. Requests are sent to port 4032 of the Document Pipeline Server, and in order to receive responses, a UDP socket must be open and actively listening. The Fuse Client dynamically allocates ports for each response, requiring configuration of a range of dynamic ports, akin to what is required for Document Pipeline Info. Therefore, this UDP port range must be open in the firewall of the machine where the Fuse Client is running.

4.2.7.4. Configuration Requirements for a Document Pipeline Cluster Server

A Document Pipeline Cluster Server requires Inbound rules to open port 4032 in order to

receive requests from Fuse Management Client for OpenText Archive Center. Additionally, the OpenText Archive Center machine running the Fuse Client needs Inbound rules to open the port range 4023-4072 to receive responses from the cluster nodes. Below, you can find a visual aid illustrating this topic within a diagram:



4.2.7.5. SSL Support

It is possible to enable SSL security directly on Fuse Management Client for OpenText Archive Center and Fuse Management Central.

4.2.7.6. Register SSL certificate

Self-signed or internal certificates can be configured to be propagated as a truststore for all internal SSL communications if needed and as key store for exposing the server as HTTPS.

The recommendation for registering SSL certificates is to configure a `fuse` SSL bundle in `application.yml`.

For example, to register a signed certificate with its private CA:

```
spring:
  ssl:
    bundle:
      pem:
        fuse:
          # change these values accordingly
          truststore:
            certificate: c:/path/ca.pem
          keystore:
            certificate: c:/path/ca-signed.pem
            private-key: c:/path/private.key
            private-key-password: secret
```

Communications with SSL-secured Fuse Management Central or OpenText Archive Center will make use of the above certificate.

It is possible to configure either PEM-encoded files or Java keystore files, as well as other trust material. Please refer to the SSL section in the Spring Boot Reference Documentation.

4.2.7.7. Expose Fuse Management Client for OpenText Archive Center as HTTPS server

After registering the SSL certificate, HTTPS at Fuse Management Client for OpenText Archive Center can be enabled by simply adding the following configuration to `application.yml` file:

```
server:
  # Fuse Management Client HTTPS Port
  port : 8444
  ssl:
    bundle: fuse
```

4.2.7.8. Self-signed certificate support

If the configured SSL certificate is self-signed, it must be declared explicitly by configuring in `application.yml`:

```
spring:
  ssl:
    bundle:
      jks:
        fuse:
          keystore:
            # pointing to a self-signed certificate
            location: c:/path/self-signed.jks
            password: secret
            type: PKCS12

http-client:
  ssl:
    trust-self-signed: true
```

4.2.8. Post-installation steps

4.2.8.1. Validate installation

To validate if Fuse Management Client for OpenText Archive Center was successfully installed and is up and running, open the following URL and login using OpenText Archive Center credentials:

```
http://<otac.server.host>:8081
```



In order for the Fuse Management Client feature to function correctly, it is mandatory to configure the following directories in the client settings:

- **`Binaries folder`** – Path to the OpenText Archive Center binaries folder, e.g. in Windows `/Program Files/OpenText/Archive Server 24.4/bin` or in UNIX `/opt/opentext/ac/binaries/bin`
- **`Spawner bin folder`** – Path to the OpenText Archive Center spawner folder, e.g. in Windows `/Program Files/Common Files/Open Text/Spawner/bin` or in UNIX `/opt/opentext/ac/spawner/bin`

If these directories are not properly set in the `directories` section, Fuse Management Client will fail to send metrics to Fuse Management Central.

If Fuse Management Client for OpenText Archive Center is not running, check the **Appendix A - Troubleshooting** for possible known issues and workarounds.

4.2.8.2. Checking for possible hotfixes

After installing Fuse Management Client for OpenText Archive Center please check for possible released hotfixes. Hotfixes are cumulative, so you only need to install the latest one. Available hotfixes can be found in Fuse Management Central repository (https://sw.vilt-group.com/), under the `Fuse Management Central/X.X.X/Clients/Fuse Management Client for OpenText Archive Center X.X.X/Hotfixes` folder of each Fuse Management Central version.

Please follow the `README.txt` file that will be available inside the hotfix package to get instructions on how to apply the hotfix, as well as information about the changes introduced.

5. Fuse Management Central Administration

This chapter explains how to configure Fuse Management Central interactively using its Administration pages, allowing Fuse administrators to adjust all of the application features.

To access **Fuse Management Central Administration** area:

1. **Open** Fuse Management Central:

```
http://<fuse-management-central-host>:2100
```

2. Login with your authentication credentials:
    - **Username**: `fuseadmin` *(default)*
    - **Password**: `fuseadmin` *(default)*
3. Click **Fuse Administration** on the navigation menu.

5.1. Security

By default, Fuse Management Central has a built-in administrator user account named `fuseadmin`, which cannot be deleted.

This chapter describes how to change this user account password and email.

5.1.1. Change *fuseadmin* password

For security reasons, is highly recommended to change the `fuseadmin` user default password.

To change `fuseadmin` default password:

1. On Fuse Management Central Administration area, click **Security**
2. Fill the following fields and click **Change Password**:
    - **Current password** *(Default: `fuseadmin`)*
    - **New password**
    - **Confirm password**

5.1.2. Change *fuseadmin* email

To change `fuseadmin` email address:

1. On Fuse Management Central Administration area, click **Security**.
2. Insert or update the email address and click **Submit**.

5.2. General

To allow your systems to communicate with Fuse Management Central, the **API Endpoint** URL

must be updated with its FQDN URL.

> To allow systems to dispatch their metrics to Fuse Management Central, the **API Endpoint** URL must be accessible by all managed systems.

To update the **API Endpoint** URL:

1. On Fuse Management Central Administration area, click **General**.
2. Under the **Fuse Management Central URL** section, update the **API Endpoint** URL and click **Update**:

> Please note that the **API Endpoint URL is built-in on your license key file. Changing it will invalidate your current license and automatically deactivate all systems**.
> Before changing API Endpoint URL, please request an updated license providing the new license data.

5.3. License

A valid license is required for Fuse Management Central to operate properly. By default, **when installed for the first time, Fuse Management Central has no license applied**.

Please note that **under the following license scenarios, Fuse Management Central will have limited functionality**:

- **Not Licensed** *(No license file found in the `license` folder)*
- **Invalid License** *(License data mismatch Fuse Management Central API Endpoint)*
- **Trial License Expired** *(The current trial license period has expired)*
- **Subscription License Expired** *(The current subscription license period has expired)*

Fuse Management Central can support multiple OpenText system types, currently OpenText Content Server and/or OpenText Archive Center, each one requiring its own license key file to enable its management functionalities.

5.3.1. Fuse Management Central for Content Server License

Fuse Management Central for Content Server has the following license models available:

Type	Description
Perpetual License - Per User	Limited to a total number of Standard Named Users for OpenText Content Server/Extended ECM.

Type	Description
Perpetual License - Per Managed System *(For MSP Only)*	Limited to a total number of managed systems.
Subscription License - Per User	License issued monthly and limited to the total number of Standard Named Users in OpenText Content Server/Extended ECM.
Subscription License - Per Managed System *(For MSP Only)*	License issued monthly and limited to a total number of managed systems.

5.3.2. Fuse Management Central for Archive Center Server License

Fuse Management Central for Archive Center has the following license models available:

Type	Description
Perpetual License - Per Managed System	Limited to a total number of managed systems.
Subscription License - Per Managed System	License issued monthly and limited to a total number of managed systems.

5.3.3. Request License

When requesting your license, either for OpenText Content Server and/or OpenText Archive Center, please provide the following information when contacting the software **Support** channel or your **Account Executive**:

- **System Type** *(OpenText Content Server or OpenText Archive Center)*
- **Trial Period** *(Trial License Only)*
- **Customer Name**
- **Fuse Management Central URL (API Endpoint)**
- **Total Managed Systems** *(For "Per Managed System" license models)*
- **Total System Named Users** *(Total OpenText Content Server/Extended ECM total Standard Named Users)*

Please note that each OpenText solution requires its own Fuse Management Central license

key file to enable its functionalities. Upon receiving your license file(s), you must ensure that each solution license file has the correct name:

- **OpenText Content Server** license file: `otcs-key.license`
- **OpenText Archive Center** license file: `otac-key.license`



Please be aware that once a license is issued, all the above data will be hardcoded into it. Therefore, any changes to this data will require an updated license.

5.3.4. Apply License

To apply your license, please follow these steps:

1. On Fuse Management Central main menu, navigate to **Fuse Administration > Administration**.
2. You will be redirected to the **License** section.
3. For each license file, according to the license type (OTCS and/or OTAC), **upload** the license file in the corresponding area.
4. Validate in the respective **License Information** card whether the license was updated successfully.

5.3.5. Validate License Status

To validate your license status:

1. On Fuse Management Central main menu, navigate to **Fuse Administration > Administration**.
2. Validate whether your **License Information** data is correct and whether Fuse Management Central license status is valid.



When Fuse Management Central is running with an invalid license (such as a trial expiration or Fuse Management Central URL mismatch), all managed systems will automatically deactivate, thereby limiting the functionality of Fuse Management Central.

5.4. OTDS Integration

Fuse Management Central has a built-in administrator user account (`fuseadmin`), which cannot be deleted.
To allow other users access to Fuse Management Central, it must be integrated with OpenText™ Directory Services (OTDS).

Fuse Management Central integrates natively with OTDS, leveraging its authentication capabilities while allowing centralized user management.

5.4.1. Create OTDS Resource

To integrate Fuse Management Central with OTDS, a resource must be created in OTDS.

To create the OTDS resource, follow these steps:

1. Open OTDS Administration by navigating to its URL (e.g. `http(s)://otds.company.com:8080/otds-admin`).
2. In the web administration menu, click **Resources**.
3. Click on the **Add** button in the button bar. The New Resource wizard will then guide you through the steps to create a new resource.
4. On the **General** page:
    a. In the **Resource Name** box, type a descriptive name for this resource (e.g. *Fuse Management Central*).

    Please note that the name you type here cannot be edited later.

    b. *(Optional)* In the **Display Name** box, you can optionally type a different resource name.
    c. *(Optional)* In the **Description** box, you can optionally type a short resource description.
    d. Leave all other options with default values and then click **Next**.
5. On the **Synchronization** page, make sure that **User and group synchronization** option is not checked, and click **Next**.
6. On the **Principal Attribute** page, leave all options with default values and click **Save**.
7. In the **Resource Activation** window, copy or write down the resource identifier.

5.4.1.1. Add users and/or groups to the created Resource

Once the OTDS Resource for Fuse Management Central is created, OTDS will automatically create an Access Role named "Access to <*ResourceName*>". Users and/or groups who will be able to login to Fuse Management Central must be added to this Access Role.

For more detailed information regarding OTDS functionality, please refer to OpenText™ Directory Services documentation.

5.4.2. Activate OTDS Resource

To activate Fuse Management Central with OTDS:

1. On Fuse Management Central Administration area, click **OTDS Integration**.
2. Fill the following fields and click **Activate**:
    - **OTDS URL**: *The FQDN address of the OTDS Server (e.g. `http(s)://otds.company.com:8080`)*
    - **OTDS Resource ID**: *The ID of the resource that has been created in OTDS*

Once activated, the OTDS resource activation status will only be displayed when authenticating in Fuse Management Central using an OTDS account with administrative privileges (e.g. `otadmin@otds.admin`).

5.4.3. Configuring Fuse Management Central Access Roles

To manage user privileges, a set of access roles is available in Fuse Management Central, each with specific privilege sets:

Access Role	Privilege Description
Fuse Admin	Permits access to the Fuse Administration area, allowing full control over Fuse Management Central. In addition to these privileges, this role also has all privileges of **System Admin** role.
System Admin	This role can manage all systems, allowing users to perform actions such as Restarting, Applying Configurations, etc.
Guest	Limited privileges role, for users with "read-only" access, meaning that no management actions can be performed allowing only to observe monitoring metrics.

To allow users to authenticate in Fuse Management Central using OTDS, these access roles **must be mapped with one or more OTDS groups** from both synchronized or unsynchronized partitions, depending on your OTDS partition scenarios.

To map an OTDS group with a Fuse Management Central access role:

1. Login to Fuse Management Central using the `otadmin@otds.admin` OTDS user account.
2. On Fuse Management Central Administration area, click **OTDS Integration**.
3. **Map** each role by selecting or inserting one or more OTDS groups to each role field.
4. Click **Save roles**.

Please note that **all OTDS groups mapped with Fuse Management Central access roles must be added to Fuse Management Central OTDS Access Role**.
This access role is automatically created when the Fuse Management Central resource is created in OTDS.

5.5. Add New System

This section will guide you through the process of adding a new OpenText™ system to Fuse Management Central.

Before starting, ensure that:

✅ You have the appropriate **OpenText Content Server** or **OpenText Archive Center** Fuse Management Client installed on your system.

✅ Fuse Management Central can access your system:

- OpenText Content Server
  - Fuse Management Client for OpenText Content Server runs as a module installed inside OpenText Content Server, so you need to be able to access OpenText Content Server CGI URL *(e.g.* `http(s)://otcs.company.com/otcs/cs.exe`*)*

Please note that when adding an OpenText™ Content Server system to Fuse Management Central for the first time, it cannot be running under **Eclipse (CSIDE)**. If it is, it is mandatory to **close Eclipse (CSIDE)** and run OpenText Content Server service. Then, wait until Fuse Management Central scans all of your system's components.

- OpenText Archive Center
  - Fuse Management Client for OpenText Archive Center runs as a standalone application, so you need to be able to access Fuse Management Client for OpenText Archive Center URL: *(e.g.* `http(s)://otac.company.com:8081`*)*

✅ Your system can access Fuse Management Central (e.g. `http://fuse.company.com:2100`).

5.5.1. Activation Request

1. Access Fuse Management Central:

```
http://<fuse-management-central-host>:2100
```

2. **Login** with your authentication credentials:
   - **Username**: `fuseadmin` *(default)*
   - **Password**: `fuseadmin` *(default)*

3. Click **Systems** on the navigation menu.
4. Click **Add System**.
5. Complete the required fields, following all the wizard steps:

Field	Description
System Type	*OpenText Content Server* or *OpenText Archive Center*.
System URL	* For OpenText Content Server: URL to Content Server CGI *(e.g. `http(s)://otcs.company.com/otcs/cs.exe`)*. * For OpenText Archive Center: URL to Fuse Management Client for OpenText Archive Center *(e.g. `http(s)://otac.company.com:8081`)*.
Environment	Select the Environment name (e.g. *"DEVELOPMENT"*) **Please note that only systems belonging to the same cluster can be added to the same environment. Mixing systems from different clusters within the same Fuse Management Central environment will result in system deactivation!**
(Optional) **Credentials**	Select the checkbox to inherit the credentials previously set from the Environment.
Username	User account with sufficient OTCS login privileges. **At minimum, the account must have the "Log-in enabled" option enabled to allow FMC authentication. No additional user account permissions are required.**
Password	User account password.
* Test Connection*	Validate that your system fulfills all the requirements. **If the connection test is not successful, please review all system parameters (System URL, credentials, etc.) and try again.**
System Name	System name or alias (e.g. *"LV181"*).
Advanced Options	

Field	Description
(Optional) **Description**	System description to help system identification (e.g. *"Partner sandbox"*)
(Optional) **Owners**	System owner(s) email(s) (e.g. *"john.doe@company.com"*), for event email notifications.
(Optional) **Tags**	System tags (e.g. *"front-end"*).



System tags are valuable for logically grouping systems, enabling you to filter them when applying configurations, performing bulk actions, and more.

6. Click **Add System**.
7. Next, copy the **System ID** and send it to your system administrator for authorization of the Fuse Management Central activation request.

5.5.2. Authorize Activation

5.5.2.1. For OpenText Content Server

1. Open **Content Server Administration** page in a web browser.
2. If prompted, enter the Administrator password, and then click **Log-in**.
3. Select **Fuse System Administration > Fuse System Activation**.
4. Insert the provided **System ID** and click **Activate**.
5. Your system is now **activated** and Fuse Management Central can begin managing and monitoring it.

5.5.2.2. For OpenText Archive Center

1. Open **Fuse Management Client for OpenText Archive Center** web page in a web browser. See Post-installation setps for details.
2. If prompted, login with OpenText Archive Center authentication.
3. You should see a pending activation request.
4. Insert the provided **System ID** and click **Activate**.
5. Your system is now **activated** and Fuse Management Central can begin managing and monitoring it.
6. **Next step:** Configure the log settings for OpenText Archive Center. You should indicate the ECM logs path and the Tomcat logs path.

The logs path configuration is **required** in order to be able to remotely open, view and download OpenText Archive Center logs directly from Fuse Management Central.

If the authorize activation process fails, please check if all requirements are fulfilled and review the procedure.

5.6. Integration Channels

Integration channels allow Fuse Management Central to integrate with SMTP and/or 3rd Party incident management or alert systems, to easily notify teams about OpenText performance or health issues.

Notifications Timezone

Communications made from Integration Channels have a specific timezone setting. This timezone is used to compose the messages sent to the configured Integration Channels, for example for the alert dates sent in email notifications or dates in ServiceNow incident comments.

The timezone used in these notifications can be changed in **Fuse Administration > Administration > Integration Channels**, in the **General Configurations** section.

5.6.1. SMTP

To enable email notifications for Fuse Management Central alerts, operations, etc… you must first configure the SMTP Settings.

1. On Fuse Management Central Administration area, click **Integration Channels**.
2. On the **SMTP** panel, fill in the following information:
    a. **Enabled**: *Enable or disable the SMTP integration.*
    b. **Sender Email**: *Type the email address that will be used as the "From" address in all email notifications sent by Fuse Management Central.*
    c. **SMTP Host**: *The FQDN hostname of the SMTP server to which Fuse Management Central will connect in order to send email.*
    d. **SMTP Port**: *The port number used by the SMTP server.*
    e. *(Optional)* **SMTP Username**: *If your SMTP server requires it, type the username to be used in the connection to the SMTP server.*
    f. *(Optional)* **SMTP Password**: *If your SMTP server requires it, type the password for the username you typed in the previous step.*
    g. **Enable StartTLS**: *Enable this option if your SMTP server requires TLS.*
    h. **Enable SSL**: *Enable this option if your SMTP server requires SSL.*
3. Click **Send test email** and validate if you have received a test email notification.

> ! The test email notification will be sent to the email defined on your user account. If you are authenticated with the `fuseadmin` user account, this user account email must be properly set.

4. Click **Update** to save your SMTP configurations.

5.6.1.1. Custom Email Settings

Custom email settings in Fuse Management Central allow you to personalize email alerts by creating a custom email template, containing the most relevant and up-to-date information for effective problem resolution and communication. This template can include placeholders dynamically filled with the details of each alert, providing users with tailored notifications that are relevant to their specific needs.

To enable custom email settings for Fuse Management Central alerts, follow these steps:

1. Navigate to the **SMTP** panel.
2. Enable the **Custom Email Settings** option.
3. Set the following information:
    - **Email To**: *Add additional email addresses where the notifications should be sent, complementing any existing ones.*
    - **Subject**: *Customize the email alert subject, using one or more placeholders to include dynamic information.*
    - **Body**: *Customize the email alert body, using one or more placeholders to include dynamic information.*
4. Click **Update** to save your SMTP configurations.

Below are the placeholders that can be used to customize the email alerts:

Placeholder	Description
`@|SystemName|`	The name of the affected system, e.g., "WEB4193".
`@|EnvironmentName|`	The name of the affected environment, e.g., "PRODUCTION".
`@|ComponentType|`	The type of component affected by the problem, e.g., "System Object Volume".
`@|ComponentName|`	The name of the specific component impacted by the problem, e.g., "Document Conversion Server For Default".

Placeholder	Description
`@|AlertStatus|`	The current status of the alert, indicating whether it is active, resolved, or dismissed.
`@|AlertSeverity|`	The severity level of the alert, such as "Error" or "Warning".
`@|AlertStartTimeStamp|`	The start timestamp of the alert, displayed in the Fuse Management Central server's time zone format, e.g., "25 Aug 2021 16:59:39 (GMT+2)".
`@|AlertFinishTimeStamp|`	The finish timestamp of the alert, also in the Fuse Management Central server's time zone format, e.g., "25 Aug 2021 17:12:21 (GMT+2)".
`@|AlertURL|`	The URL of the alert details, redirecting to Fuse Management Central.
`@|AlertTitle|`	The title or headline of the alert, summarizing the nature of the problem.
`@|AlertDescription|`	A detailed description of the alert, providing context and additional information about the problem.

Please note that custom email settings will only be applied for single email alerts. Digest email alerts are currently excluded from the customized email template.

Digest email alerts consolidate multiple alerts into a single summary email, avoiding receiving separate emails for each alert. They provide users with a comprehensive overview of all the issues detected within a specified timeframe.

5.6.2. Checkmk Integration

Checkmk is one of the leading tools for Infrastructure and Application Monitoring, offering both

Open Source and Enterprise license models.

Fuse Management Central offers a seamless integration with **Checkmk**, with an agent plug-in specifically designed to connect your Fuse Management Central instance to Checkmk.

Using the data provided by the Alerts API, the plug-in will add a complete list of Services to be monitored in Checkmk, from your OpenText Content Suite.

The shared information is related to **active alerts** grouped by **System Type**, **System Name** and **Component**. It also includes alert monitoring for specific **Environment** scope alerts, as well as related to the **Fuse Management Central** instance and administration alerts.

5.6.2.1. Checkmk Plug-in Download

To download the Fuse Management Central plug-in for Checkmk go to **Checkmk Exchange** and download the following package:

- **OpenText Fuse Management Central Plug-in**

5.6.2.2. Checkmk Plug-in Installation and Configuration

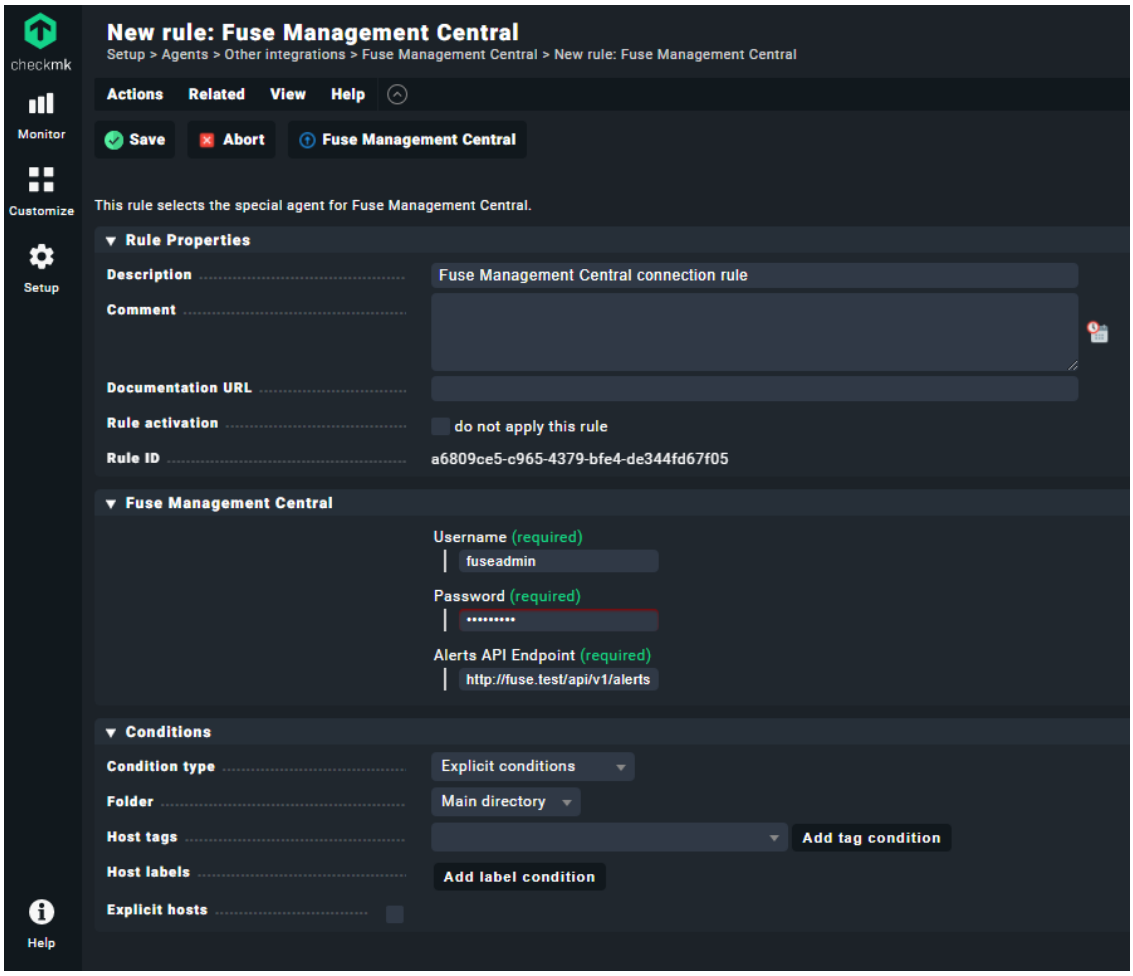
To install the Fuse Management Central plug-in in Checkmk follow these steps:

- Navigate to **Setup > Extension packages**.
- Click in **Upload package** and upload the recently downloaded Fuse Management Central package.
- Click in **Upload & Install**.

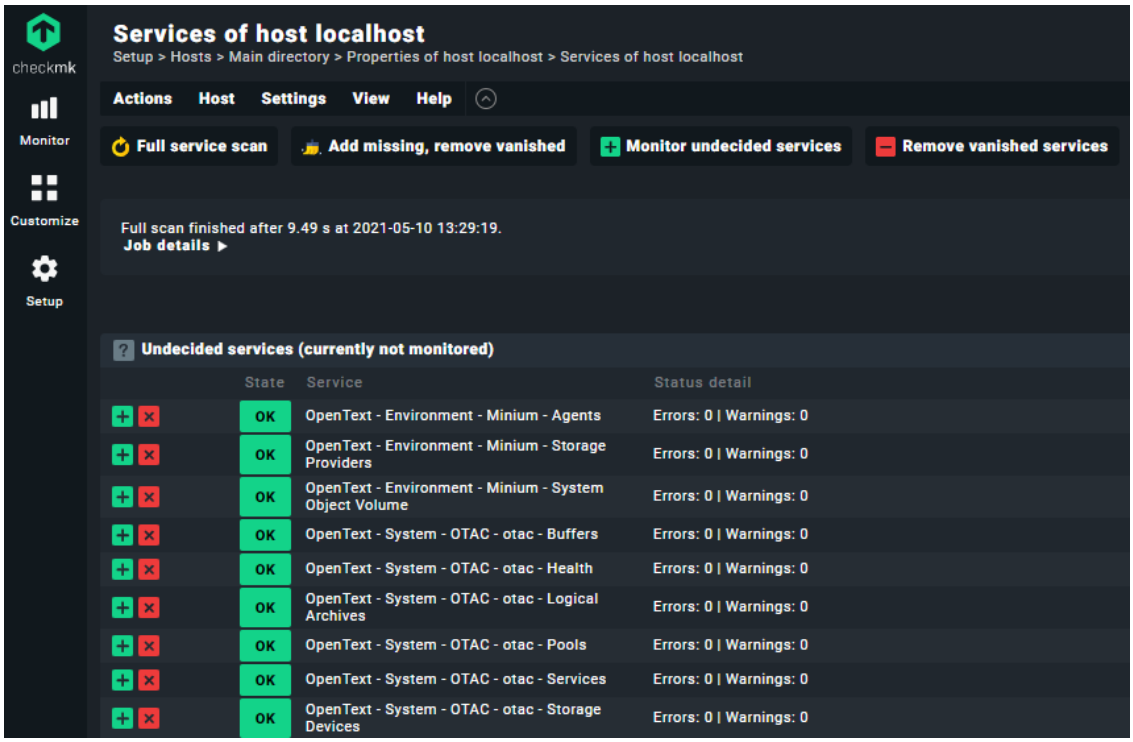
Configuration steps

Now you have the Fuse Management Central plug-in in your Checkmk and you can configure it:

- Navigate to **Setup > Other integrations > Fuse Management Central**
- **Create a new rule** with the Fuse Administrator credentials (username and password) and the URL for the Alerts API.
    - The Alerts API URL should be `http://[fuse-host]:[fuse-port]/api/v1/alerts`.



After creating the rule for the Fuse Management Central plug-in, create a new host, and in the **Service Configuration** page you can discover the Fuse services and add them to be monitored.



In order to correctly see the Fuse services summary, you need to go to **Setup > Services > Service monitoring rules > Escape HTML in service output** and create a rule for the host with the Fuse services to **don't escape html**.

5.6.2.3. Instance Service

The Fuse Management Central instance service has the name **Fuse Management Central - Instance**, this service will always appear.

If Checkmk can connect to the configured Fuse, it will have the **OK** state. If it can't connect to Fuse, it will have the **CRIT** state and the summary will have more information about why it could not connect to Fuse.

5.6.2.4. Other Services

For each pair **System - Component** type, you will have a service with the name **OpenText - System - [system type] - [system name] - [component type name]**.

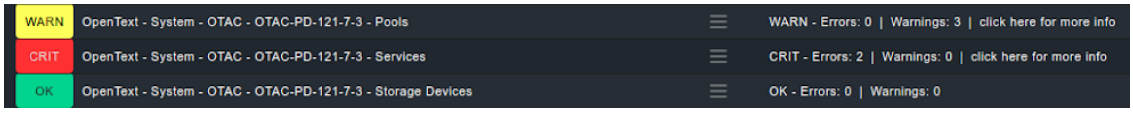
The same happens for the Environments, for each pair **Environment - Component** type, you will have a service named **OpenText - Environment - [environment name] - [component type name]**.

For the Admin component types, you will have one service for each, named **Fuse Management Central - [component type name]**.

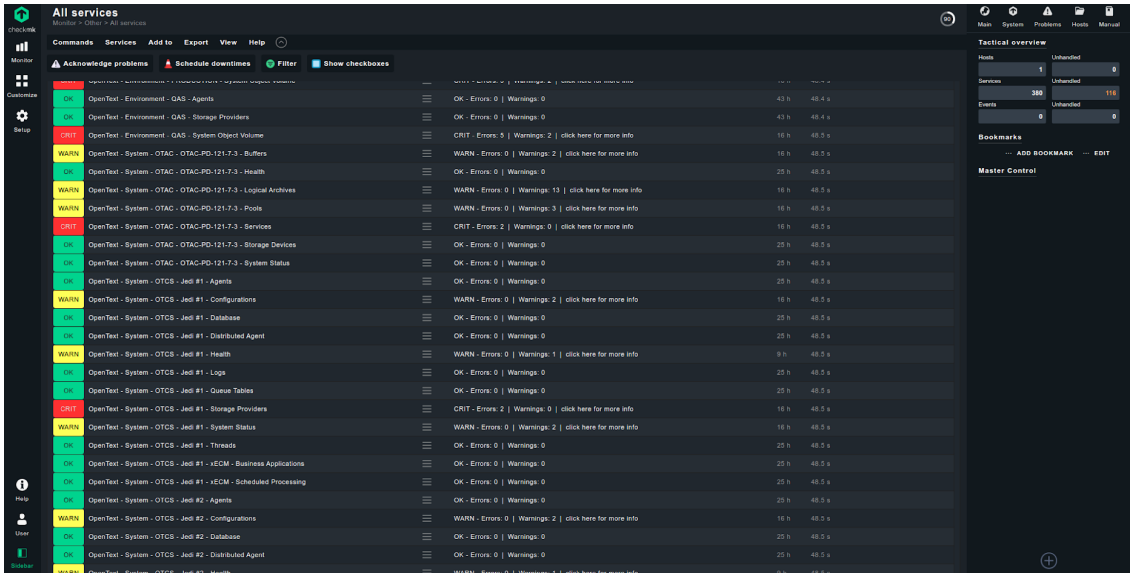
In these services you will be able to see the number of errors and warnings in their summary.

Regarding the state, they can have one of the following states:

- **OK** - there are no errors or warnings
- **WARN** - there are some warnings but no errors
- **CRIT** - there are errors



If a service is in a **WARN** or **CRIT** state, you will have a link to Fuse Management Central in its summary. The link will redirect you to the Fuse Management Central Alerts page with the correct filters selected, so you can see more details about the errors/warnings.



5.6.3. OpenText Service Management Automation X (SMAX)

OpenText SMAX uses built-in AI and analytics to deliver smart IT service management (ITSM). This integration allows Fuse Management Central to integrate seamlessly with OpenText SMAX incident management, allowing teams to receive notifications regarding OpenText performance, operations or health issues detected by Fuse Management Central.

5.6.3.1. OpenText SMAX Integration Setup

To enable OpenText SMAX notifications for Fuse Management Central alerts, you must first configure the OpenText SMAX Settings.

1. On Fuse Management Central Administration area, click **Integration Channels**.
2. On the **OpenText SMAX** panel, fill the following information:
    a. **Enabled**: *Enable or disable the OpenText SMAX integration.*
    b. **OpenText SMAX URL**: *Your OpenText SMAX server URL, for example*

`https://us23-smax.saas.microfocus.com`.

c. **Tenant ID**: _Your OpenText SMAX tenant ID, for example `962991158`.

d. **Username**: *Username to be used in the connection to the OpenText SMAX service.*

> ! The provided user must have permissions to create and update incidents. Also, to be able to get all the possible configurations, the user needs to have permission to perform the following API calls listed in the end of the section.

e. **Password**: *Password for the username typed in the previous step.*

3. Click **Connect** to validate the provided configuration. When settings are correct, the **Incident Settings** area is expanded.

> On this area, you can configure additional settings. All these settings are required to ensure that OpenText SMAX incidents are categorized correctly.

a. Incident Classification:

i. **Service**: *The Actual Service that will be assigned to the incident upon creation.*

ii. **Category**: *The category that will be assigned to the incident upon creation.*

iii. Impact Classification:

A. **Warning Alerts**: *Mapping between Fuse Alert severity type WARNING into SMAX impact incident option.*

B. **Error Alerts**: *Mapping between Fuse Alert severity type ERROR into SMAX impact incident option.*

iv. Urgency Classification:

A. **Warning Alerts**: *Mapping between Fuse Alert severity type WARNING into SMAX impact incident option.*

B. **Error Alerts**: *Mapping between Fuse Alert severity type ERROR into SMAX impact incident option.*

b. Incident Assignment:

i. **Service Desk Group**: *The service group that will be assigned to the incident upon creation.*

c. Incident Resolution:

i. **Completion Code**: *The Code selected will be used as default when an incident is resolved.*

> The fields Service, Category, and Service Desk Group can be changed directly within the OpenText SMAX application. All other

fields can be customized in Fuse Management Central `application.yml` file.

4. Once all configurations are set according to your needs, press **Update** to save the configuration.

5.6.3.2. OpenText SMAX Custom Settings

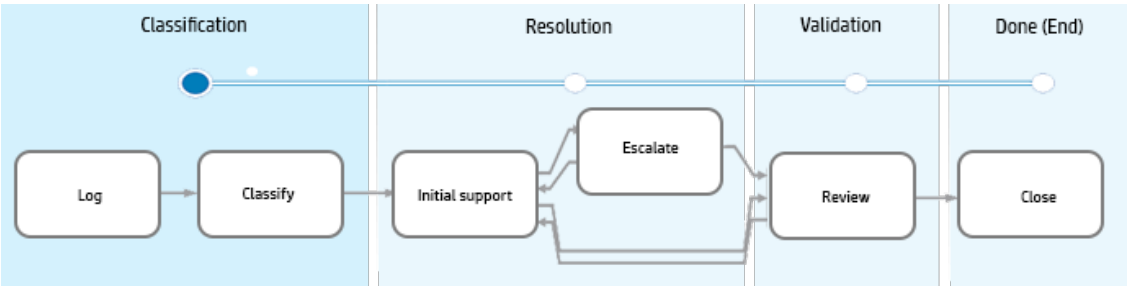
In this section, you will find a comprehensive list of the available settings for Fuse Management Central, along with their respective functionalities. These configurations are customizable and can be modified in the `application.yml` file located in the Fuse Management Central installation folder.

```
smax:
  # Name: Display Name
  enums:
    urgency:
      TotalLossOfService: Total Loss of Service
      SevereDisruption: Severe Disruption
      SlightDisruption: Slight Disruption
      NoDisruption: No Disruption
    impact:
      Enterprise: Enterprise
      SiteOrDepartment: Site or Department
      MultipleUsers: Multiple Users
      SingleUsers: Single Users
    completionCode:
      SuccessfulDiagnosis: Successful Diagnosis
      NoFaultFound: No fault found
      NoUserResponse: No user response
      NotReproducible: Not reproducible
      OutofScope: Out of scope
      RequestRejected: Request rejected
      Resolvedbyfix: Resolved successfully
      ResolvedWorkaround: Resolved with workaround
      UnabletoSolve: Unable to solve
      WithdrawnbyUser: Withdrawn by user
```

The properties (urgency, impact and completionCode) are divided between the value sent to the API and the display name shown in Fuse.

5.6.3.3. OpenText SMAX Incident Flow

This integration was specifically developed for the incident settings displayed in Fuse. Any additional settings that may be required by the incident form may not function properly. This design ensures that the incident flow comes pre-configured out of the box, as shown below:



5.6.3.4. OpenText SMAX API Calls

To work as expected, Fuse needs access to the following OpenText SMAX's endpoints:

HTTP method	API call
GET	{OPENTEXT_SMAX_URL}/auth/authentication-endpoint/authenticate/token?TENANTID={TENANTID}
GET	{OPENTEXT_SMAX_URL}rest/{TENANTID}/ems/ActualService?filter=PhaseId+!%3D+%27pRetired%27+or+PhaseId+%3D+null&layout=Id,DisplayLabel,SubType,BusinessCriticality,Environment,AssetModel,AssetModel.Id,AssetModel.DisplayLabel,AssetModel.AssetType,AssetModel.DeviceSubType,AssetModel.InfrastructurePeripheralSubType,AssetModel.IsDeleted&order=DisplayLabel+asc&meta=totalCount&size=9999&skip=0
GET	{OPENTEXT_SMAX_URL}rest/{TENANTID}/ems/PersonGroup?layout=FULL_LAYOUT&order=Name+asc&size=9999
GET	{OPENTEXT_SMAX_URL}/rest/{TENANTID}/ems/ITProcessRecordCategory?filter=IsActive+%3D+%27TRUE%27+and+Level1ParentId+%3D+null&layout=FULL_LAYOUT&order=DisplayLabel+asc&size=9999
POST	{OPENTEXT_SMAX_URL}/rest/{TENANTID}/ems/bulk"

5.6.4. ServiceNow Integration

This integration allows Fuse Management Central to integrate with ServiceNow incident management to easily notify teams about OpenText performance, operation or health issues detected by Fuse Management Central.

5.6.4.1. ServiceNow Integration Setup

To enable ServiceNow notifications for Fuse Management Central alerts, you must first configure the ServiceNow Settings.

1. On Fuse Management Central Administration area, click **Integration Channels**.
2. On the **ServiceNow** panel, fill the following information:
    a. **Enabled**: *Enable or disable the ServiceNow integration.*
    b. **ServiceNow URL**: *Your ServiceNow instance URL, for example https://dev115171.service-now.com.*
    c. **Username**: *Username to be used in the connection to the ServiceNow service.*

> ! User provided must have permissions to create and update incidents. Also, to be able to get all the possible configurations, the user needs to have permission to perform the following API calls listed in the end of the section.

d. **Password**: *Password for the username typed in the previous step.*

3. Click **Connect** to validate the provided configuration. When settings are correct, the **Incident Settings** area is expanded.

> On this area you can configure some extra settings. None of those settings are required, but they will help ServiceNow incidents to be typified.

a. Incident Status:

i. *(Optional)* **New Incident**: *State selected will be used as default state when an incident is created.*

ii. *(Optional)* **Resolve Incident**: *State selected will be used as default state when an incident is resolved.*

b. Incident Severity:

i. *(Optional)* **Warning Alert**: *Mapping between Fuse Alert severity type WARNING into ServiceNow severity option.*

ii. *(Optional)* **Error Alert**: *Mapping between Fuse Alert severity type ERROR into ServiceNow severity option.*

c. Additional Incident Fields *(depending on your ServiceNow configurations, settings below may or may not be displayed)*:

i. *(Optional)* **Assignment Group**: *Group selected will be used as default assignment group when an incident is created.*

ii. *(Optional)* **Contact Type**: *Contact type selected will be used as default contact type when an incident is created.*

iii. *(Optional)* **Incident Area**: *Area selected will be used as default incident area when an incident is created.*

d. Static Settings *(these settings cannot be changed)*:

i. **Caller**: *The Caller is a static field based on the user serviceNow account provided previously.*

ii. **Short Description**: *Incident title used when an incident is created.*

iii. **Description**: *Incident description used when an incident is created.*

4. Once all configurations are set according to your needs, press **Update** to save the configuration.

To work as expected, Fuse needs access to the following ServiceNow's endpoints:

HTTP method	API call
GET	`http://[ServiceNow_URL]/api/now/table/sys_user?sysparm_query=user_name={username}&sysparm_fields=user_name,sys_id,roles`
GET	`http://[ServiceNow_URL]/api/now/table/sys_user_group?sysparm_fields=sys_id,name`
GET	`http://[ServiceNow_URL]/api/now/table/u_category?sysparm_fields=sys_id,u_incident_area`
GET	`http://[ServiceNow_URL]/api/now/table/sys_choice?name=incident&element={field}&language=EN&sysparm_fields=label,value`
POST	`http://[ServiceNow_URL]/api/now/table/incident`
PUT	`http://[ServiceNow_URL]/api/now/table/incident/{id}`

5.7. Alert Manager

Fuse Management Central uses an Alert Manager to automatically detect system anomalies and consequently triggering real-time alerts. These alerts are used to report on warning or error situations, such as performance degradations, failing agent schedule, lack of resources, among others.

5.7.1. Integration Channels

The current supported integration channels are:

Channel	Description
User Interface *(Default)*	Notification events are displayed on Fuse Management Central user interface, being displayed on the events list and adjusting the failing component style, providing real-time feedback to users.
SMTP	If the SMTP Settings are properly set, alert notifications will be sent by email to the system owners.
ServiceNow	If the ServiceNow Setup is properly set, alert notifications will be sent to ServiceNow.

To manage an alert integration channel:

1. On Fuse Management Central Administration area, click **Alert Manager**.

a. **Click** on the ON/OFF toggle button to fully disable the alert for all integration channels.

b. **Click** on the specific integration channel (e.g. "SMTP") toggle button to disable it from being dispatched to that integration channel.

2. Click **Update** to save your new settings.

5.7.2. Metric Thresholds

Fuse Management Central system monitoring is based on numerous built-in, predefined metric thresholds. These default thresholds are set based on common usage scenarios but can be adjusted to fit your organization requirements.

To change the default metric thresholds:

1. On Fuse Management Central Administration area, click **Alert Manager**.
2. **Adjust** each alert threshold, to fit your requirements.
3. Click **Update** to save your new settings.

5.7.3. Dismissing Alerts

Fuse Management Central allows to dismiss alerts for specific components inside a specific system or environment.

On **Alert Manager** page, Fuse Administrators can review existing Dismiss Rules for any alert. These rules are listed within each alert section and can be easily removed by **clicking** on the **Trash icon**.

You can filter the alerts list to show only those with Dismiss Rules by using the **Dismissed Alerts** filter at the top of the list and selecting the option **With Dismissed Rules**. This makes it easier to find a specific Dismiss Rule.

5.8. Alerts API

Fuse Management Central provides a REST API to deliver a summary of all active alerts, allowing alerts integration with third-party centralized monitoring solutions.

Fuse Management Central Alert API is available in the following endpoints:

Endpoint	Description
`http://[host]:[port]/api/v1/alerts`	**Alerts List Endpoint** - Lists all active alerts, optionally filtered by alert attributes.
`http://[host]:[port]/api/v1/alerts/layout`	**Layout Endpoint** - Fetches the layout and details of all existent Environments, Systems and Component Types.

Endpoint	Description
`http://[host]:[port]/api/v1/alerts/summary`	Alerts Summary Endpoint - Fetches a summary of all active alerts, grouped by Environment, System and Component Type.

5.8.1. Alerts List Endpoint

The Alerts List API endpoint provides a list of all active alerts, which can optionally be filtered by alert attributes.

The following attributes are provided for each entry:

Attribute	Description
`alertDescription`	The alert description. E.g. `"Storage provider 'Default' write performance is below 2MiB/s for more than 6m 0s"`.
`alertDetailsURL`	The URL to Fuse Management Central alert details page.
`alertId`	The alert unique identifier. E.g. `"7e7a397f-a4c8-3a59-ac88-b865f0d91ee4"`.
`alertObservedValue`	The component's last observed value, which has triggered the alert. E.g. `"1.70MB"`.
`alertSeverity`	The alert severity (`warning`or `error`). E.g. `"warning"`.
`alertStartTime`	The alert start time, in timestamp format ("yyyy-MM-dd'T'HH:mm:ssZZZZ"). E.g. `"2023-04-01'T'13:21:45ZZZZ"`.
`alertTitle`	The alert title. E.g. `"Storage Provider 'Default' Write Performance"`.
`alertType`	The alert type. E.g. `"otac_buffer_free_space_high"`.

Attribute	Description
`componentName`	The affected component name. E.g. `"Local_Buffer"`.
`componentType`	The affected component type. E.g. `"BUFFERS"`.
`environmentId`	The affected environment unique identifier. E.g. `"098390f5-969f-4e43-a6c4-01d3d011864f"`.
`environmentName`	The environment name. E.g. `"Production - Local Site"`.
`systemId`	The affected system unique identifier. E.g. `"0bdab526-9218-45de-ad2c-53e87b8f3e47"`.
`systemName`	The affected system name. E.g. `"PD-121-146"`.
`systemType`	The affected system type. E.g. `"OTAC"`.

The alerts list may optionally be filtered, combining one or more of the available alert attribute filters, narrowing the scope of returned results. The filter usage is done by adding one or more alert attributes as URL parameters, e.g. `/api/v1/alerts?systemType=OTAC`.

The following alert attribute filter enumerators are available:

5.8.1.1. System Type

	systemType
OpenText Content Server	`OTCS`
OpenText Archive Center	`OTAC`

5.8.1.2. Alert Type and Component Type

	alertType	componentType
System is offline	heartbeat_missing	SYSTEM
Authentication failed	authentication_error	SYSTEM
Low application performance	performance_low	SYSTEM
CSIDE (Development) mode is enabled	cside_mode_on	SYSTEM
Low application disk space	app_disk_space_low	SYSTEM
Low disk space	noapp_disk_space_low	SYSTEM
Low CPU performance	cpu_high	SYSTEM
Low memory available	memory_high	SYSTEM
Admin server default password	default_admin_server_password	SYSTEM
Admin user default password	default_admin_user_password	SYSTEM
JVM crash	dump_files	SYSTEM
LLClient connection issue	ll_client_port_available	SYSTEM
Fuse Management Client compatibility issue	invalid_client_version	FUSE_CLIENT
Fuse Management Client secret key issue	secretkey_error	FUSE_CLIENT
Fuse Management Client pending log configurations	fuse_client_pending_log_configurations	FUSE_CLIENT
Fuse Management Client pending directory configurations	fuse_client_pending_directories_configurations	FUSE_CLIENT
System [SYSTEM_NAME] threads usage	thread_usage_high	THREADS

	alertType	componentType
System [SYSTEM_NAME] threads queue time	(Avg. Queue Time)	THREADS
System [SYSTEM_NAME] threads queue time	(Avg. Queue Time)	THREADS
System [SYSTEM_NAME] threads avg. request execution time	(Avg. Execution Time)	THREADS
System [SYSTEM_NAME] threads avg. request execution time	(Avg. Execution Time)	THREADS
System [SYSTEM_NAME] thread #[THREADID] request execution	threadExecutionTime	THREADS
System [SYSTEM_NAME] exceptions	thread_exceptions	THREADS
System [SYSTEM_NAME] threads avg. queue time high	thread_avg_queue_time_high	THREADS
System [SYSTEM_NAME] threads avg. execution time high	thread_avg_execution_time_high	THREADS
System [SYSTEM_NAME] thread current execution time high	thread_current_execution_time_high	THREADS
System [SYSTEM_NAME] exceptions	trace_files	LOGS
System [SYSTEM_NAME] logs size	log_files_size_high	LOGS
Hyperlens minimum log level required	hyperlens_otcs_log_level	LOGS
Storage provider [PROVIDER_NAME] availability	storage_provider_unavailable	STORAGE_PROVIDERS
Storage provider [PROVIDER_NAME] write performance	storage_provider_extshared_write_performance	STORAGE_PROVIDERS

	alertType	componentType
Storage provider [PROVIDER_NAME] read performance	storage_provider_extshared_read_performance	STORAGE_PROVIDERS
Storage provider [PROVIDER_NAME] write performance	storage_provider_entrepriseArchive_write_performance	STORAGE_PROVIDERS
Storage provider [PROVIDER_NAME] read performance	storage_provider_entrepriseArchive_read_performance	STORAGE_PROVIDERS
Storage provider [PROVIDER_NAME] write error	storage_provider_entrepriseArchive_cant_write	STORAGE_PROVIDERS
Storage provider [PROVIDER_NAME] write error	storage_provider_extshared_cant_write	STORAGE_PROVIDERS
Storage provider [PROVIDER_NAME] write performance	storage_provider_rdbms_write_performance	STORAGE_PROVIDERS
Storage provider [PROVIDER_NAME] Free Space	storage_provider_extshared_free_space	STORAGE_PROVIDERS
Storage provider [PROVIDER_NAME] read performance	storage_provider_rdbms_read_performance	STORAGE_PROVIDERS
Storage provider [PROVIDER_NAME] node version error	fuse_client_otcs_sp_test_binary	STORAGE_PROVIDERS
Configuration policy [POLICY_NAME] compliance	configuration_policy_compliance_failures	CONFIGURATIONS
Configuration policy [POLICY_NAME] compliance conflicts	configuration_policy_compliance_conflicts	CONFIGURATIONS
Environment 'VerifyAgent' process error	verify_agent_multiple_instances	AGENTS
Environment 'Notify' process error	notify_loader_multiple_instances	AGENTS
System 'agents' loader configuration warning	agents_loader_false_configuration	LOADERS

	alertType	componentType
System 'notify' loader configuration warning	notify_loader_false_configuration	LOADERS
System [SYSTEM_NAME] agent	agent_nextstart_failed	AGENTS
Distributed Agent [AGENT_NAME] standing by	da_agent_status_stand_by	DISTRIBUTED_AGENT
Distributed Agent [AGENT_NAME] stopped	da_agent_status_stopped	DISTRIBUTED_AGENT
Distributed Agent [AGENT_NAME] status offline	da_agent_status_offline	DISTRIBUTED_AGENT
Worker agent [AGENT_NAME] paused	da_worker_status_paused	DISTRIBUTED_AGENT
Worker agent [AGENT_NAME] stopped	da_worker_status_stopped	DISTRIBUTED_AGENT
Worker agent [AGENT_NAME] status offline	da_worker_status_offline	DISTRIBUTED_AGENT
Primary distributed agent not defined	da_primary_agent_undefined	DISTRIBUTED_AGENT
Distributed agent task error	da_total_task_error	DISTRIBUTED_AGENT
Distributed agent worker availability error	da_is_da_worker_available	DISTRIBUTED_AGENT
Distributed agent availability error	da_is_da_agent_available	DISTRIBUTED_AGENT
Distributed agent queued items	da_queue_queued	DISTRIBUTED_AGENT
Distributed agent backlog items	da_queue_backlog	DISTRIBUTED_AGENT
Distributed agent processed items error	da_queue_processed	DISTRIBUTED_AGENT
SOV process [PROCESS_NAME] is idle	sov_processes_status_idle	SOV

	alertType	componentType
SOV process [PROCESS_NAME] unknown status	sov_processes_status_unknown	SOV
SOV process [PROCESS_NAME] error	sov_processes_status_error	SOV
SOV process [PROCESS_NAME] admin server error	sov_processes_status_admin_server_error	SOV
SOV process [PROCESS_NAME] does not exist	sov_processes_status_not_exist	SOV
SOV process [PROCESS_NAME] error 11	sov_processes_status_error_11	SOV
Admin Server [ADMINSERVER_NAME] is offline	sov_admserv_active	SOV
Admin Server [ADMINSERVER_NAME] is in safe mode	sov_admserv_safe_mode	SOV
iPool [IPOOL_NAME] quarantine warning	sov_ipool_quarantined	SOV
iPool [IPOOL_NAME] pending messages processing warning	sov_ipool_pending	SOV
iPool [IPOOL_NAME] is idle	sov_ipool_idle	SOV
Data Flow [DATAFLOW_NAME] is suspended	sov_dataflow_idle	SOV
Partition [PARTITION_NAME] is unavailable	sov_partition_unavailable	SOV
Partition [PARTITION_NAME] is full	sov_partition_full	SOV
Partition [PARTITION_NAME] is almost full	sov_partition_almost_full	SOV
Search Federator "[SF_NAME]" connection error	sov_search_federator_connection	SOV

	alertType	componentType
Search Federator "[SF_NAME]" network health warning	sov_search_federator_network_health	SOV
Search is not available	search.isAvailable	SOV
"DTreeNotify" queue table object process performance	queuetables_d_tree_notify	QUEUE_TABLES
"LLEventQueue" queue table object process performance	queuetables_ll_event_queue	QUEUE_TABLES
"NotifyEvents" queue table object process performance	queuetables_notify_events	QUEUE_TABLES
"NotifyMessages" queue table object process performance	queuetables_notify_messages	QUEUE_TABLES
"ProviderRetry" queue table object process performance	queuetables_provider_retry	QUEUE_TABLES
"RenditionQueue" queue table object process performance	queuetables_rendition_queue	QUEUE_TABLES
"WorkerQueue" queue table object process performance	queuetables_worker_queue	QUEUE_TABLES
"WorkerQueuePending" queue table object process performance	queuetables_worker_queue_pending	QUEUE_TABLES
"RetentionUpdateOrder" queue table object process performance	queuetables_retention_update_order	QUEUE_TABLES
"BlobData" queue table object process performance	queuetables_blob_data	QUEUE_TABLES
Database query performance issue in [TABLE_NAME]	database_query_performance	DB_PERFORMANCE

	alertType	componentType
Business Application [BUSINESS_APPLICATION_NAME] is unavailable	extendedecm_system_status	XECM
Business Application [BUSINESS_APPLICATION_NAME] authentication error	extendedecm_system_auth_status	XECM
Extended ECM licensing issues	extendedecm_licensing_userlicense_issue	XECM
Scheduled Job [JOB_NAME] status error	extendedecm_job_status	XECM
Scheduled Job [JOB_NAME] has error items	extendedecm_job_error_items	XECM
[SERVICE_NAME] status error	otac_service_status_error	OTAC_SERVICE
Logical Archive [ARCHIVE_NAME] pool warning	otac_archive_no_pools	OTAC_ARCHIVE
Pool [POOL_NAME] is almost full	otac_pool_free_space_high	OTAC_POOL
Pool [POOL_NAME] is full	otac_pool_free_space_limit	OTAC_POOL
Pool [POOL_NAME] has no volumes	otac_pool_no_volumes	OTAC_POOL
Buffer [BUFFER_NAME] free space warning	otac_buffer_free_space_high	OTAC_BUFFER
Buffer [BUFFER_NAME] free space error	otac_buffer_free_space_limit	OTAC_BUFFER
Buffer [BUFFER_NAME] volumes warning	otac_buffer_no_volumes_attached	OTAC_BUFFER
Storage Device [DEVICE_NAME] free space warning	otac_storage_device_free_space_high	OTAC_STORAGE_DEVICE

	alertType	componentType
Storage Device [DEVICE_NAME] free space error	otac_storage_device_free_space_limit	OTAC_STORAGE_DEVICE
Storage Device [DEVICE_NAME] is dettached	otac_storage_device_detached	OTAC_STORAGE_DEVICE
Storage Device [DEVICE_NAME] has unavailable volume(s)	otac_storage_device_aborted	OTAC_STORAGE_DEVICE
Storage Device [DEVICE_NAME] aborted connection	devices.device.isAborted	OTAC_STORAGE_DEVICE
Storage Device [DEVICE_NAME] has unavailable volume(s)	otac_storage_device_unavailable_volumes	OTAC_STORAGE_DEVICE
Job [JOB_NAME] error	otac_job_has_protocol_error	OTAC_JOB
Job scheduler not running	otac_job_scheduler_not_running	OTAC_JOB
DocTool [DOCTOOL_NAME] has error document(s)	otac_dp_queue_has_errors	OTAC_DOCUMENT_PIPELINES
Doctool [DOCTOOL_NAME] is disabled	otac_dp_queue_disabled	OTAC_DOCUMENT_PIPELINES
DocTool [DOCTOOL_NAME] has failed to load	otac_dp_queue_failed	OTAC_DOCUMENT_PIPELINES
Unknown DocTool [DOCTOOL_NAME]	otac_dp_queue_unknown	OTAC_DOCUMENT_PIPELINES
Metric is stale	stale_metric	HEALTH
Disk Space Status	health_disk	HEALTH
Disk Space Status	health_disk	HEALTH
Metrics Database Status	health_prometheus	HEALTH

	alertType	component Type
Mail Status	`health_mail`	`HEALTH`
Database Status	`health_database`	`HEALTH`
Alert Manager Status	`health_alert_manager`	`HEALTH`
ServiceNow Status	`health_service_now`	`HEALTH`
SMAX Status	`health_smax`	`HEALTH`
Hyperlens collector pending restart	`hyperlens_collector_pending_restart`	`HYPERLENS`

5.8.2. Layout Endpoint

The Layout API endpoint provides an overview on the existent Systems, Environments and Component Types, as well as the Component Types used to classify Fuse Administration alerts.

This information works as metadata to be cross-referenced by other endpoints of the API.

5.8.3. Alerts Summary Endpoint

The Alerts Summary API endpoint provides a summary list of current active alerts, grouped by Environment, System and Component Type.

The following attributes are provided for each entry:

Attribute	Description
`envId`	Unique identifier of the Environment.
`systemId`	Unique identifier of the System.
`componentType`	Unique identifier of the Component Type.
`errors`	Total number of error alerts.
`warnings`	Total number of warning alerts.

Attribute	Description
`link`	Link to Fuse Management Central alert details page, with pre-configured filters for the alerts included in the summary entry.

5.8.3.1. Summary entry use cases

Each summary entry may have one or multiple attributes missing, depending on the type of alerts it represents:

Attributes	Description
`systemId` is null but `envId` exists	These are Environment alerts, exclusive to the Environment scope.
`systemId` is null and `envId` is null	These are Administration alerts, exclusive to Fuse Administration scope.
`errors` is null	No error alerts to report.
`warnings` is null	No warning alerts to report.

5.8.3.2. Systems

List of all Systems in Fuse Management Central.

The following attributes are provided for each System:

Attribute	Description
`id`	Unique identifier of the System.
`name`	Name of the System in Fuse Management Central.
`type`	Type of the System (*OTCS* or *OTAC*).
`componentTypes`	List of Component Types existent in that System, identified by `id` and `displayName`.

5.8.3.3. Environments

List of all Environments in Fuse Management Central.

The following attributes are provided for each Environment:

Attribute	Description
`id`	Unique identifier of the Environment.
`name`	Name of the Environment in Fuse Management Central.
`componentTypes`	List of Component Types existent in that Environment, exclusive to the Environment scope, identified by `id` and `displayName`.

5.8.3.4. Admin

Fuse Administration alerts metadata.

The following attributes are provided:

Attribute	Description
`componentTypes`	List of Component Types existent to classify Admin alerts, identified by `id` and `displayName`.

5.9. Backup and Restore

Fuse Management Central deals with large amounts of data in different databases. It is highly recommended that you keep recurrent backups of the stored data, avoiding losing any data in case of unfortunate events.

5.9.1. Backup Fuse Management Central Data

All data stored by Fuse Management Central is saved into two different databases:

1. **Fuse Management Central Database** - A PostgreSQL database, used to store the application model and system metadata information.
2. **Fuse Management Central Metrics Database** - A Prometheus time series database, used to store system metrics and alerts data.

Both databases are located in the "Data Directory" that you selected during installation, usually at `C:\ProgramData\Fuse Management Central`.





We strongly recommend to keep recurrent backups of this folder and its subfolders, since the databases can be later associated with a clean installation of Fuse Management Central if needed.



It is highly recommended that you stop all Fuse Management Central services before taking a backup of the folder. Backups during runtime can result in incomplete or corrupted data.

5.9.2. Restore a Backup

If you have a previous backup of the Data Directory folder being used by Fuse Management Central (usually at `C:\ProgramData\Fuse Management Central`), you can use that folder as a restore point to a new installation of Fuse Management Central.

Restore to new installation

In order to restore Fuse Management Central with that data, simply make a new installation pointing the Data Directory to the existing restore folder that you have, instead of a new location. Bear in mind that this new installation will use this new folder as its Data Directory from now on, so choose a convenient location for that folder before the installation (we recommend `C:\ProgramData\Fuse Management Central`).

Restore to current installation

If you don't want to do a new installation, you can always replace the current Data Directory by your backup. Just make sure to **stop all Fuse Management Central services** before replacing the current data with the backup.

Restore Considerations

After restoring a previous backup, some controlled errors may occur, that may require your attention:

- Data loss may occur between the restore point and the current point.
- Licensing issues that may lead to deactivated systems:
    - You may need to acquire and apply a new license, as well as reactivate all systems, if your license information has changed.
    - The new installation won't have a license applied by default.
- Fuse Management Central configurations, such as Hostname, API Endpoint and License, may differ from the previous settings.
- OTDS Integration may need to be checked in case the hostname has changed.



There are several possible problems that may have led to this situation, and while we strive to address them all, it is possible that other errors may arise. If you encounter additional problems or experience difficulties restoring your data, please contact product.support@vilt-group.com for assistance.

6. HyperLens (Experimental Feature)

This chapter describes how to deploy HyperLens on an existing and operational Fuse Management Central installation.

Currently, HyperLens only supports OpenText™ Content Server systems.

6.1. HyperLens Processor

6.1.1. Windows Installation

1. Extract the downloaded HyperLens Processor package to your desired installation directory.
2. Navigate to the extracted directory.
3. Run `install.bat` to install the HyperLens Processor as a service.
4. The service **Fuse Management Central (HyperLens Processor)** will be installed with default configuration which is suitable for most installations.
5. In order to start and stop HyperLens Processor you can start and stop the service **Fuse Management Central (HyperLens Processor)**.

The default configuration values will work without any changes for most installations. If you have specific requirements or a non-standard setup, you may need to modify the configuration properties following the instructions in the section HyperLens Processor Administration.

6.1.2. Windows Upgrade

1. Extract the downloaded HyperLens Processor package .
2. Stop the service **Fuse Management Central (HyperLens Processor)**.
3. Navigate to the existing HyperLens Processor directory.
4. Delete **lib** folder and **FuseHyperLensProcessor.jar**.
5. Replace them for the ones from the downloaded package.
6. Start the service **Fuse Management Central (HyperLens Processor)**.

6.2. HyperLens Collector

6.2.1. Windows Installation

Currently, the HyperLens Collector only supports OpenText™ Content Server and

must be installed in each OpenText™ Content Server managed system.

1. Extract the downloaded HyperLens Collector package to your desired installation directory on the Content Server machine.
2. Navigate to the extracted directory.
3. Run `install.bat` to install the HyperLens Collector as a service.
4. The service **Fuse Management Client (HyperLens Collector)** will be installed with default configuration which is suitable for most installations.
    a. By default, the service will point to a configuration file located at `%OTHOME%\config\hyperlens-config.yaml`.
    b. The Fuse Management Client will be responsible for creating and updating the `hyperlens-config.yaml` file as needed.
    c. If you don't have an `OTHOME` environment variable set, you can either create it pointing to your Content Server installation directory, or manually update the configuration file path in the service settings to use the full directory path instead of `%OTHOME%`.
        i. Open the `FuseHyperLensCollector.xml` file.
        ii. Check the `<startargument>--config=%OTHOME%\config\hyperlens-config.yaml</startargument>` and make sure the path to the Content Server installation is correct.
5. In order to start and stop HyperLens Collector you can start and stop the service **Fuse Management Client (HyperLens Collector)**.

No additional configuration of the HyperLens Collector is required for most installations. For environments with specific requirements or non-standard setups, please refer to HyperLens Collector Administration.

6.2.2. Windows Upgrade

1. Extract the downloaded HyperLens Collector package to your Content Server machine.
2. Stop HyperLens Collector service.
3. Swap otelcol-contrib.exe for the one from the downloaded package.
4. Start HyperLens Collector service.

6.3. HyperLens Administration Guide

6.3.1. HyperLens Processor Administration

6.3.1.1. Configuration Properties

For most installations, the default configuration values will work without any changes. These settings should only be modified if you have specific requirements or a non-standard setup.

HyperLens Processor configurations are available in the `config/config.properties` file. The following properties can be configured:

Property	Description
`postgres.url`	Database URL for the Fuse Database.
`postgres.user`	Username for the Fuse Database connection.
`postgres.password`	Password for the Fuse Database connection.
`otlp.port`	Port for HyperLens Collector communication.
`prometheus.port`	Port number on which the HyperLens Processor will expose HyperLens metrics.
`fuse.window-size`	Size of the window for the HyperLens Processor to collect metrics.
`tls.enabled`	Whether TLS is enabled for the HyperLens Processor.
`tls.truststore-path`	Path to the truststore file for TLS (valid if `tls.enabled=true`).
`tls.keystore-path`	Path to the keystore file for TLS (valid if `tls.enabled=true`).
`sampler.enabled`	Whether to enable Sampling for tracing.
`sampler.ratio`	Sampling ratio for tracing.

After making changes to the `config.properties` file, the HyperLens Processor service must be restarted for the changes to take effect.

6.3.2. HyperLens Collector Administration

6.3.2.1. Configuration Properties

For most installations, the default configuration values will work without any changes. These settings should only be modified if you have specific requirements or a non-standard setup.

HyperLens Collector configurations can be customized in the `config/fmc.ini` file available in the installation directory of OpenText™ Content Server.

These changes require the Content Server process to be **stopped** before editing the file.

The following properties could exist or be placed under the `[hyperlens]` section. If the `[hyperlens]` section does not exist, it can be created.

Property	Description
`IsHyperLensEnabled`	No need to change, this is managed automatically by Fuse.
`HyperLensURL`	URL of the HyperLens Processor. Configure this if it differs from the default Fuse Management Central machine or is running under a different protocol.
`HyperLensTLS`	Whether to enable TLS for the connection.
`HyperLensCertFile`	Path to the TLS certificate file (required if `HyperLensTLS=true`).
`HyperLensKeyFile`	Path to the TLS private key file (required if `HyperLensTLS=true`).
`HyperLensCAFile`	Path to the CA file (required if `HyperLensTLS=true` and a Certificate Authority file is needed).

After making changes to the `fmc.ini` file, the Content Server process must be restarted followed by a restart of the HyperLens Collector service for the changes to take effect.

7. Uninstall Fuse Management Central

This chapter describes how to remove Fuse Management Central from a host server.
If you are upgrading to a newer version of Fuse Management Central, it may be necessary to uninstall the older version.

7.1. Uninstall on Microsoft Windows

Fuse Management Central uses a Windows Installer to remove components from a Windows platform. The program is designed to remove all program files installed at the time of the Fuse Management Central installation.

> ! The uninstall process **will not remove any configuration and long term metric data**. This is beneficial because you can retain these data files for use if you upgrade your Fuse Management Central software.
> To force the deletion of all Fuse Management Central data files, please contact product.support@vilt-group.com.

To uninstall Fuse Management Central on Windows:

1. Stop all Fuse Management Central services.
2. Using the Windows application for removing programs (for example, **Programs and Features**), select Fuse Management Central installer and then click **Uninstall**.
3. Use the uninstall wizard automatically to remove all Fuse Management Central installed components.

7.2. Uninstall Helm Deployments

Use this section only for Kubernetes/OpenShift deployments installed with Helm.

7.2.1. Uninstall Fuse Management Central

```
helm uninstall fuse-management-central -n fuse-management-central
```

Helm uninstall does not delete PVC data by default. Review PostgreSQL, Prometheus, and optional Alertmanager PVCs before deleting them manually.

7.2.2. Uninstall Fuse Management Client for OpenText Archive Center

```
helm uninstall fuse-management-client-for-otac -n opentext
```

Replace `opentext` with the namespace used for the OTAC Client deployment.

Helm uninstall does not delete PVC data by default. Review OTAC Client state and log PVCs before deleting them manually.

7.3. Uninstall HyperLens

7.3.1. Uninstall HyperLens Processor

1. Navigate to the directory where the HyperLens Processor was installed.
2. Run `uninstall.bat` to uninstall the HyperLens Processor service.
3. The service **Fuse Management Central (HyperLens Processor)** will be removed from your system.
4. Ensure that all related files and configurations are removed if necessary.
5. If you encounter any issues during uninstallation, refer to the troubleshooting section in the documentation.

7.3.2. Uninstall HyperLens Collector

1. Navigate to the directory where the HyperLens Collector was installed.
2. Run `uninstall.bat` to uninstall the HyperLens Collector service.
3. The service **Fuse Management Client (HyperLens Collector)** will be removed from your system.
4. Ensure that all related files and configurations are removed if necessary.
5. If you encounter any issues during uninstallation, refer to the troubleshooting section in the documentation.

8. Appendix A - Troubleshooting

8.1. Known Issues and Workarounds

This section describes scenarios that users may run into and how to troubleshoot and workaround or fix them.

8.1.1. Metrics not available after installation or upgrade

In some Content Server installations, after installing/upgrading and activating a system, the metrics might not be available and your System will appear offline. In that case there are several things to check:

- If the installation or upgrade of the Fuse Management Client module fails to complete the system restart correctly. Try restarting Content Server manually.
- Check if the firewall is blocking HTTP requests between Fuse Management Central installation and the System being activated.
- Check Fuse Management Client logs to make sure metrics are being dispatched:
  - Change the log level to 'DEBUG', using `http://[Content_Server_URL]?func=fuseclient.ConfigureLogging` or using Fuse Management Central logs widget configuration (please refer to the **Logs** section of Fuse Management Central User Guide).
  - Check the logs in the OTCS log directory with the name `fuseclient<date>.log`.

8.1.2. Error when adding a new System with https

When adding a new system with HTTPS, an error message similar to the following one might be displayed:

```
Failed to connect
Resource not reachable: I/O error on POST request ()...) unable to find
valid certification path to requested target
```

In this case, the corresponding SSL certificate must be configured in Fuse Management Central, as documented in section **Enabling SSL** of **Installation and Administration Guide**.

8.1.3. Fuse Metrics Database corrupted files

Sometimes some data files in the Fuse Metrics Database can become corrupted, preventing the Metrics service to start. We found that this issue commonly occurs in the following scenarios:

- Fuse Metrics Database service was not shutdown correctly.

- Disk ran out of space.

If this happens, we suggest shutting down Fuse Management Central (all services) and then restart Fuse Management Central (all services).

At this stage, some errors may occur when restarting the Fuse Metrics Database if there are corrupted data files present. If you encounter an error while starting the Fuse Metrics Database, please check the Fuse Metrics Database logs. The logs may contain entries similar to the following, indicating potential issues:

```
err="opening storage failed: block dir:
\"data\\\\01E270EBZ1YPKF7BB2WZ38H5SV\": open
data\\01E270EBZ1YPKF7BB2WZ38H5SV\\meta.json: The system cannot find the
file specified."
```

or

```
err="opening storage failed: found unsequential head chunk files 23 and
25"
```

In order to fix this issue, the folder or file specified in the log message, for example *01E270EBZ1YPKF7BB2WZ38H5SV* or chunk files *23* and *25*, should be deleted. These folders can be found in the Fuse Metrics Database installation directory, typically within the **data** or **data/chunks_head** folders.

This process should be repeated for each folder or file mentioned in the logs until Fuse Metrics Database is able to start without any errors.

8.1.4. Uninstall Fuse Management Client for OpenText Content Server (16.2.2, 16.2.3, 16.2.4)

For the Content Server versions **16.2.2**, **16.2.3** and **16.2.4**, we found out that the standard soft-restart is not enough to reload all the loaders required for Fuse Management Client. Therefore, we recommend restarting the processes/services of Content Server to make sure everything is updated properly.

8.1.5. Fuse Management Central unable to connect with ServiceNow

To configure ServiceNow in Fuse, the provided user must explicitly have one of the following roles: **admin**, **itil** or **itil_admin**.

If the user has one of the required roles, but you still get the error message indicating that the user lacks permission to create or update incidents when trying to connect, please add the following configuration to **aplication.yml** file located at `[fuse_installation_folder]/config/` (e.g. `C:\Program Files (x86)\Fuse Management Central\config\`).

```
service-now:
```

```
    roles:
      - admin
      - itil_admin
      - itil
```

8.1.6. CPU Usage issues in Windows systems

Fuse Client requires permission to access CPU performance on Windows systems. If the CPU usage chart is not displaying any data or displaying negative values, it is possible that Fuse Client does not have the required permissions to retrieve it.

This can happen for either Fuse OTCS Client or Fuse OTAC Client, and it is usually caused by the Windows User running the Content Server (OTCS) or Fuse Management Client for OTAC service not having the required permissions to access the Windows system counters.

The Windows User running Fuse Client needs to belong to one of the following groups:

- Performance Log Users
- Performance Monitor Users

In order to add a user to one of these groups, follow these steps:

1. Access **Windows Services** and identify which user is running the service:
    a. Content Server (OTCS) service, if the issue is with Fuse Management Client for Content Server.
    b. Fuse Management Client for OTAC service, if the issue is with Fuse Management Client for OTAC.
    c. By right-clicking on the service and selecting **Properties**, the **Log On** tab will show the user account running the service.
2. Acess **Windows Control Panel** and searh for **Edit Local Users and Groups**.
3. Search for the groups **Performance Log Users** or **Performance Monitor Users**.
4. Add the user identified in the first step to one of these groups.
5. **Restart** the Content Server (OTCS) or Fuse Management Client for OTAC service.

It is also possible that the Windows system counters do not exist. You can check this by attempting to open the Windows Performance Monitor. For more information about system counters and how to rebuild them, please refer to official Microsoft documentation on Manually rebuild performance counters for Windows Server.

8.1.7. SELinux blocking Fuse Management Client for OpenText Archive Center execution

If you have SELinux installed on your system, it might be blocking the execution of the Fuse Client binary (`/path/to/fuse/client/fuse-client-otac.jar`).

If you attempt to run the Fuse Client OTAC and it doesn't start, check your SELinux alerts to determine if the execution of `fuse-client-otac.jar` is being blocked. If it is, you will need to

adjust your SELinux policies and settings to allow the execution.

Sometimes, simply restoring the SELinux context is enough to unblock it:

```
/sbin/restorecon -v /path/to/fuse/client/fuse-client-otac.jar
```

By default, after extraction, the `fuse-client-otac.jar` might not be categorized as an executable. If that is the case, you might need to change the SELinux context of the file with:

```
chcon --type=java_exec_t /path/to/fuse/client/fuse-client-otac.jar
```

However, this is highly dependent on your own infrastructure and policies, so you might want to consult your System Administrator.

8.1.8. "Could not find the class definition" llserverworker JVM error

Occasionally, in **OpenText Content Server version 23.x or above**, there is a rare issue encountered when restarting OpenText Content Server via Fuse Management Central. This triggers a soft restart, but sometimes the Fuse Management Client fails to load properly due to the incorrect loading of the required Java libraries (.jar). Currently, the workaround for this issue is to manually restart the OTCS service, forcing the reload of all the Java libraries and resolving the problem.

Based on our investigations, it appears that this behavior is related to the internal restart process of OpenText Content Server and is beyond the control of our module. We are actively analyzing this behavior and investigating the issue with OpenText.

8.1.9. Deactivated or Deleted Systems still collecting HyperLens data

In some cases, it has been observed that deactivated or deleted systems in Fuse Management Central may continue to collect HyperLens data.

This happens because the HyperLens Collector is installed and running on the system, and it continues to collect data until it is manually stopped or uninstalled.

When a system is deactivated or deleted in Fuse Management Central, it does not automatically stop the HyperLens Collector on that system. Therefore, if the Collector is not manually stopped or uninstalled, it will continue to collect data and send it to Fuse Management Central, even if the system is no longer active in the management console.

To prevent this issue, it is recommended to manually stop the HyperLens Collector on any system that is deactivated or deleted in Fuse Management Central. This can be done by following the standard uninstallation process for the HyperLens Collector on the respective system.

8.2. Helm Troubleshooting

This section describes Kubernetes/OpenShift Helm deployment issues and the checks that operators should run before escalating.

8.2.1. ImagePullBackOff

Cause: the image reference in values is not available to the cluster runtime.

Check:

```
kubectl describe pod <pod-name> -n <namespace>
kubectl get events -n <namespace> --sort-by=.lastTimestamp
```

Fix:

- Import the image into every eligible node, or push it to a registry.
- Review the chart image repository, tag, digest, and image pull secret values.

8.2.2. Namespace mismatch

Cause: the chart namespace value does not match Helm `--namespace`.

Fix:

- Set `global.namespace` to the same namespace used by Helm, or leave it empty when supported by the chart.

8.2.3. PVC remains Pending

Cause: no suitable storage class or insufficient storage capacity.

Check:

```
kubectl describe pvc <pvc-name> -n <namespace>
kubectl get storageclass
```

Fix:

- Configure the persistence storage class in values.
- Ask the platform team to provide a suitable default `StorageClass`.

8.2.4. Fuse Management Central is not reachable

Check:

- `fuse.service.type`
- `ingress.enabled`
- Ingress controller or OpenShift Route configuration

- `fuse.env.restUrl`
- NetworkPolicy ingress rules, if enabled

For validation, use:

```
kubectl port-forward service/fuse-management-server 2100:2100 -n fuse-
management-central
```

8.2.5. OTAC Client cannot resolve the OTAC workload

Check:

- OTAC Client and OpenText Archive Center are in the same namespace.
- `archiveCenter.url` points to an in-cluster Service when automatic discovery is expected.
- `clusterClient.target.serviceName` is set when the URL host is not service-discoverable.
- `clusterClient.target.containerName` is set when the target OTAC container cannot be inferred safely.
- Service, Endpoints, EndpointSlices, Pod, and RBAC resources exist in the namespace.

8.2.6. OTAC Client reports Metrics API unavailable

Cause: `metrics.k8s.io` is not available or RBAC does not allow access.

Check:

```
kubectl get --raw /apis/metrics.k8s.io/v1beta1/pods
kubectl auth can-i get pods.metrics.k8s.io \
  --as system:serviceaccount:opentext:fuse-management-client-for-otac \
  -n opentext
```

Fix:

- Install or enable the Kubernetes Metrics Server or OpenShift monitoring component that provides pod metrics.
- Confirm the chart-managed Role and RoleBinding were created.

8.2.7. OTAC Client cannot execute approved OTAC utilities

Check:

- ServiceAccount token automount is present in the OTAC Client Pod.
- `pods/exec` permission includes `get` and `create`.
- The target OTAC container is correctly resolved.
- The OTAC runtime directories are present in the OTAC Client `data.json`.

The OTAC Client does not expose a generic exec endpoint and does not accept arbitrary commands. Only approved OTAC utility operations are supported.

8.2.8. `storagedevices` reports no local storage manager configured

This response comes from the OpenText Archive Center application topology when Storage Manager/STORM is not configured for the OTAC environment. It is not, by itself, a Kubernetes RBAC or OTAC Client exec failure.

Validate the OTAC Storage Manager/STORM configuration using the supported OpenText Archive Center administration procedures for your deployment.

8.2.9. NetworkPolicy blocks Helm deployment traffic

NetworkPolicies are disabled by default. If enabled, confirm that all required ingress and egress paths are explicitly allowed.

For Fuse Management Central, confirm:

- User access to Fuse Management Central.
- Fuse Management Central access to its internal PostgreSQL, Prometheus, and Alertmanager services.
- DNS access from chart workloads.

For Fuse Management Client for OpenText Archive Center, confirm:

- Access to the OTAC Client HTTP endpoint from the expected management system.
- OTAC Client access to OpenText Archive Center.
- OTAC Client UDP traffic to Document Pipelines, when enabled.
- DNS access from the OTAC Client workload.

9. Appendix B - How-Tos

This section contains some quick and basic tutorials on general topics and tools related to Fuse Management Central.

Please have into consideration that these are general tutorials with general instructions, and the correct application of them highly depends on your specific scenario and systems.



These tutorials do not replace the official documentation available for each topic or tool. Please search the web for official documentation if you want more details on a certain topic or tool.

9.1. How to install and configure Prometheus on a Linux Server



This quick guide serves as a general reference and is not intended as official documentation. It is important to note that consulting official documentation is necessary, as the effectiveness of this guide relies on your individual circumstances and system configurations.

This guide explains how to install and configure Prometheus on a Linux Server.

9.1.1. Pre-requirements

- Superuser (sudo) access to the Linux machine
- Access to the internet to download Prometheus binaries

9.1.2. Setup Prometheus

1. Go to the Prometheus downloads page (https://prometheus.io/download/#prometheus) and get the correct download link for the required version, for example https://github.com/prometheus/prometheus/releases/download/v3.5.5/prometheus-3.5.5.linux-amd64.tar.gz.
2. Download and extract Prometheus binaries to a folder called `prometheus-files`:



This guide uses `curl` to download the binaries, but you can download them on your own way, whichever is more convenient.

```
curl -LO
https://github.com/prometheus/prometheus/releases/download/v[prometh
eus-version]/prometheus-[prometheus-version].linux-amd64.tar.gz
tar -xvf prometheus-[prometheus-version].linux-amd64.tar.gz
mv prometheus-[prometheus-version].linux-amd64 prometheus-files
```

3. Create a specific user for Prometheus:

```
sudo useradd --no-create-home --shell /bin/false prometheus
```

4. Create the necessary directories and assign ownership of these directories to the **Prometheus** user:

```
sudo mkdir /etc/prometheus
sudo mkdir /var/lib/prometheus
sudo chown prometheus:prometheus /etc/prometheus
sudo chown prometheus:prometheus /var/lib/prometheus
```

5. Copy `prometheus` and `promtool` binaries from `prometheus-files` to `/usr/local/bin` and assign ownership of these binaries to the **Prometheus** user:

```
sudo cp prometheus-files/prometheus /usr/local/bin/
sudo cp prometheus-files/promtool /usr/local/bin/
sudo chown prometheus:prometheus /usr/local/bin/prometheus
sudo chown prometheus:prometheus /usr/local/bin/promtool
```

6. Copy the `consoles` and `console_libraries` directories from `prometheus-files` to `/etc/prometheus` and assign ownership of these directories to the **Prometheus** user:

```
sudo cp -r prometheus-files/consoles /etc/prometheus
sudo cp -r prometheus-files/console_libraries /etc/prometheus
sudo chown -R prometheus:prometheus /etc/prometheus/consoles
sudo chown -R prometheus:prometheus
/etc/prometheus/console_libraries
```

9.1.3. Setup Prometheus Configuration



This guide will create a default Prometheus configuration file at `/etc/prometheus/prometheus.yml`. However, during the Fuse Management Central installation, you will need to replace this configuration file with one provided by Fuse Management Central.

1. Create the `prometheus.yml` file and add a default configuration for Prometheus to monitor itself:

```
sudo vi /etc/prometheus/prometheus.yml
```

```
global:
  scrape_interval: 10s
scrape_configs:
  - job_name: 'prometheus'
    scrape_interval: 5s
    static_configs:
```

```
      - targets: ['localhost:9090']
```

2. Ensure that the **Prometheus** user is the owner of the configuration file:

```
sudo chown prometheus:prometheus /etc/prometheus/prometheus.yml
```

9.1.4. Setup Prometheus as a Service

1. Create a Prometheus service file:

```
sudo vi /etc/systemd/system/prometheus.service
```

```
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target
[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
    --config.file=/etc/prometheus/prometheus.yml \
    --storage.tsdb.path=/var/lib/prometheus/ \
    --web.console.templates=/etc/prometheus/consoles \
    --web.console.libraries=/etc/prometheus/console_libraries
[Install]
WantedBy=multi-user.target
```

2. Reload the systemd service:

```
sudo systemctl daemon-reload
```

3. Start the Prometheus service:

```
sudo systemctl start prometheus
```

4. You can check the status of the service with:

```
sudo systemctl status prometheus
```

5. You can stop the Prometheus service with:

```
sudo systemctl stop prometheus
```

9.1.5. Validate Prometheus installation

With the Prometheus service running, you should be able to access Prometheus web console on `http://[prometheus-server]:9090/query`.

9.1.6. Possible issues and workarounds

9.1.6.1. Running Prometheus in a different Port

The previous guide assumes that the port `9090` is available and that Prometheus will run there. If that is not the case, the following changes must be made to make Prometheus available in a different port.

1. Add the argument `--web.listen-address=0.0.0.0:<new-port>` to Prometheus startup script.
2. Change `prometheus.yml` file default target to new port.

```
scrape_configs:
  - job_name: 'prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:<new-port>']
```



This configuration may have already been replaced by Fuse Management Central own configuration, and the default scrape target may no longer exist if you have already proceeded with the Fuse Management Central installation.

9.1.6.2. Firewall blocking external access to Prometheus

If you cannot access Prometheus web console from an external system, your Firewall might be blocking it.

External access to Prometheus **is not mandatory**, since all components that need access to Prometheus are in the same machine.

However, if you want to test the Prometheus installation by accessing it externally, please refer to your Firewall provider official documentation and allow external access to the Prometheus port.

9.1.6.3. SELinux blocking Prometheus binary execution

If you have SELinux enabled on your system, it might block the execution of the Prometheus binary located at (`/usr/local/bin/prometheus`).

If you attempt to run Prometheus and encounter issues with it not starting, inspect your SELinux alerts to determine if it is being blocked. If SELinux is indeed blocking Prometheus, you will need to adjust your SELinux policies and settings to permit the execution of

Prometheus.

Sometimes a simple restore of the SELinux context is enough to unblock it:

```
/sbin/restorecon -v /usr/local/bin/prometheus
```

However, this is highly dependent on your own infrastructure and policies, so you might want to consult your System Administrator.

9.2. How to install and configure AlertManager on a Linux Server



This quick guide serves as a general reference and is not intended as official documentation. It is important to note that consulting official documentation is necessary, as the effectiveness of this guide relies on your individual circumstances and system configurations.

This guide explains how to install and configure AlertManager on a Linux Server.

9.2.1. Pre-requirements

- Superuser (sudo) access to the Linux machine
- Access to the internet to download AlertManager binaries

9.2.2. Setup AlertManager

1. Go to the AlertManager downloads page (https://prometheus.io/download/#alertmanager) and get the correct download link for the required version, for example https://github.com/prometheus/alertmanager/releases/download/v0.33.0/alertmanager-0.33.0.linux-amd64.tar.gz.
2. Download and extract AlertManager binaries to a folder called `alertmanager-files`:



This guide uses `curl` to download the binaries, but you can download them on your own way, whichever is more convenient.

```
curl -LO
https://github.com/prometheus/alertmanager/releases/download/v[alert
manager-version]/alertmanager-[alertmanager-version].linux-
amd64.tar.gz
tar -xvf alertmanager-[alertmanager-version].linux-amd64.tar.gz
mv alertmanager-[alertmanager-version].linux-amd64 alertmanager-
files
```

3. Create a specific user for AlertManager:

```
sudo useradd --no-create-home --shell /bin/false alertmanager
```

4. Create the necessary directories and assign ownership of these directories to the **Alertmanager** user:

```
sudo mkdir /etc/alertmanager
sudo mkdir /var/lib/alertmanager
sudo chown alertmanager:alertmanager /etc/alertmanager
sudo chown alertmanager:alertmanager /var/lib/alertmanager
```

5. Copy `alertmanager` and `amtool` binaries from `alertmanager-files` to `/usr/local/bin` and assign ownership of these binaries to the **Alertmanager** user:

```
sudo cp alertmanager-files/alertmanager /usr/local/bin/
sudo cp alertmanager-files/amtool /usr/local/bin/
sudo chown alertmanager:alertmanager /usr/local/bin/alertmanager
sudo chown alertmanager:alertmanager /usr/local/bin/amtool
```

9.2.3. Setup AlertManager Configuration



This guide will create a default AlertManager configuration file at `/etc/alertmanager/alertmanager.yml`. However, during the Fuse Management Central installation, you will need to replace this configuration file with one provided by Fuse Management Central.

1. Copy the default `alertmanager.yml` file from `alertmanager-files` directory to `/etc/alertmanager/alertmanager.yml`:

```
sudo cp alertmanager-files/alertmanager.yml
/etc/alertmanager/alertmanager.yml
```

2. Ensure that the **Alertmanager** user is the owner of the configuration file:

```
sudo chown alertmanager:alertmanager
/etc/alertmanager/alertmanager.yml
```

9.2.4. Setup AlertManager as a Service

1. Create an AlertManager service file:

```
sudo vi /etc/systemd/system/alertmanager.service
```

```
[Unit]
Description=AlertManager
```

```
Wants=network-online.target
After=network-online.target
[Service]
User=alertmanager
Group=alertmanager
Type=simple
ExecStart=/usr/local/bin/alertmanager \
    --config.file=/etc/alertmanager/alertmanager.yml \
    --storage.path=/var/lib/alertmanager/
[Install]
WantedBy=multi-user.target
```

2. Reload the systemd service:

```
sudo systemctl daemon-reload
```

3. Start the AlertManager service:

```
sudo systemctl start alertmanager
```

4. You can check the status of the service with:

```
sudo systemctl status alertmanager
```

5. You can stop the AlertManager service with:

```
sudo systemctl stop alertmanager
```

9.2.5. Validate AlertManager installation

With the AlertManager service running, you should be able to access AlertManager web console on `http://[alertmanager-server]:9093/`.

9.2.6. Possible issues and workarounds

9.2.6.1. Running AlertManager in a different Port

The previous guide assumes that the port `9093` is available and that AlertManager will run there. If that is not the case, the following changes must be made to make AlertManager available in a different port.

1. Add the argument `--web.listen-address=0.0.0.0:<new-port>` to AlertManager startup script.

9.2.6.2. Firewall blocking external access to AlertManager

If you cannot access AlertManager web console from an external system, your Firewall might be blocking it.

External access to AlertManager **is not mandatory**, since all components that need access to AlertManager are in the same machine.

However, if you want to test the AlertManager installation by accessing it externally, please refer to your Firewall provider official documentation and allow external access to the AlertManager port.

9.2.6.3. SELinux blocking AlertManager binary execution

If you have SELinux enabled on your system, it might block the execution of the Alertmanager binary located at (`/usr/local/bin/alertmanager`).

If you attempt to run AlertManager and encounter issues with it not starting, inspect your SELinux alerts to determine if it is being blocked. If SELinux is indeed blocking AlertManager, you will need to adjust your SELinux policies and settings to permit the execution of AlertManager.

Sometimes a simple restore of the SELinux context is enough to unblock it:

```
/sbin/restorecon -v /usr/local/bin/alertmanager
```

However, this is highly dependent on your own infrastructure and policies, so you might want to consult your System Administrator.

9.3. How to upgrade Prometheus on a Linux Server



This quick guide serves as a general reference and is not intended as official documentation. It is important to note that consulting official documentation is necessary, as the effectiveness of this guide relies on your individual circumstances and system configurations.

This guide explains how to upgrade an existent installation of Prometheus on a Linux Server.

9.3.1. Requirements and Assumptions

This guide assumes that the Prometheus installation was completed according to the instructions provided in How to install and configure Prometheus on a Linux Server.

You will need **superuser (sudo)** access to the Linux machine.

9.3.2. Upgrade Prometheus

1. Stop Prometheus service:

```
sudo systemctl stop prometheus
```

2. Create a backup of the Prometheus data storage directory:
    - `/var/lib/prometheus`
3. Go to the Prometheus downloads page (https://prometheus.io/download/#prometheus) and get the correct download link for the required version, for example https://github.com/prometheus/prometheus/releases/download/v3.5.5/prometheus-3.5.5.linux-amd64.tar.gz.
4. Download and extract Prometheus binaries to a folder called `prometheus-upgrade-files`:



This guide uses `curl` to download the binaries, but you can download them on your own way, whichever is more convenient.

```
curl -LO
https://github.com/prometheus/prometheus/releases/download/v{prometh
eus-version}/prometheus-{prometheus-version}.linux-amd64.tar.gz
tar -xvf prometheus-{prometheus-version}.linux-amd64.tar.gz
mv prometheus-{prometheus-version}.linux-amd64 prometheus-upgrade-
files
```

5. Replace current `prometheus` and `promtool` binaries with the new ones and assign ownership of these binaries to the **Prometheus** user:

```
sudo cp prometheus-upgrade-files/prometheus
/usr/local/bin/prometheus
sudo cp prometheus-upgrade-files/promtool /usr/local/bin/promtool
sudo chown prometheus:prometheus /usr/local/bin/prometheus
sudo chown prometheus:prometheus /usr/local/bin/promtool
```

6. Replace the current `consoles` and `console_libraries` directories with the new ones and assign ownership of these directories to the **Prometheus** user:

```
sudo cp -r prometheus-upgrade-files/consoles /etc/prometheus
sudo cp -r prometheus-upgrade-files/console_libraries
/etc/prometheus
sudo chown -R prometheus:prometheus /etc/prometheus/consoles
sudo chown -R prometheus:prometheus
/etc/prometheus/console_libraries
```

7. Start Prometheus service again:

```
sudo systemctl start prometheus
```

9.4. How to upgrade AlertManager on a Linux Server



This quick guide serves as a general reference and is not intended as official documentation. It is important to note that consulting official documentation is necessary, as the effectiveness of this guide relies on your individual circumstances and system configurations.

This guide explains how to upgrade an existent installation of AlertManager on a Linux Server.

9.4.1. Requirements and Assumptions

This guide assumes that the Alertmanager installation was completed according to the instructions provided in **How to install and configure AlertManager on a Linux Server**.

You will need **superuser (sudo)** access to the Linux machine.

9.4.2. Upgrade AlertManager

1. Stop AlertManager service:

```
sudo systemctl stop alertmanager
```

2. Create a backup of the AlertManager data storage directory:
    - `/var/lib/alertmanager`
3. Go to the AlertManager downloads page (https://prometheus.io/download/#alertmanager) and get the correct download link for the required version, for example https://github.com/prometheus/alertmanager/releases/download/v0.33.0/alertmanager-0.33.0.linux-amd64.tar.gz.
4. Download and extract AlertManager binaries to a folder called `alertmanager-upgrade-files`:



This guide uses `curl` to download the binaries, but you can download them on your own way, whichever is more convenient.

```
curl -LO
https://github.com/prometheus/alertmanager/releases/download/v{alert
manager-version}/alertmanager-{alertmanager-version}.linux-
amd64.tar.gz
tar -xvf alertmanager-{alertmanager-version}.linux-amd64.tar.gz
mv alertmanager-{alertmanager-version}.linux-amd64 alertmanager-
upgrade-files
```

5. Replace current `alertmanager` and `amtool` binaries with the new ones and assign ownership of these binaries to the **Alertmanager** user:

```
sudo cp alertmanager-upgrade-files/alertmanager
/usr/local/bin/alertmanager
sudo cp alertmanager-upgrade-files/amtool /usr/local/bin/amtool
sudo chown alertmanager:alertmanager /usr/local/bin/alertmanager
```

```
sudo chown alertmanager:alertmanager /usr/local/bin/amtool
```

6. Start AlertManager service again:

```
sudo systemctl start alertmanager
```

9.5. How to use a specific Java version with Fuse Management Central or Fuse Management Central Client for OpenText Archive Center

It is possible to force Fuse Management Central or Fuse Management Central Client for OpenText Archive Center to use a specific Java version by editing the following files:

- For Fuse Management Central:
  - `[fuse_installation_folder]/config/FuseManagementCentral.xml` (e.g. `C:\Program Files (x86)\Fuse Management Central\config\FuseManagementCentral.xml`)
- For Fuse Management Central Client for OpenText Archive Center:
  - `[fuse_otac_client_installation_folder]/FuseManagementCentral.xml` (e.g. `C:\Program Files (x86)\fuse-client-otac\FuseManagementCentral.xml`)

These paths can be different, according to the paths that where chosen to install the services. In these files, edit the `<executable>` settign and set a specific path to the Java to be used, for example:

```
<executable>C:\Program Files\Java\jdk-17.0.10.7-
hotspot\bin\java</executable>
```

9.6. How to setup Direct Server SSL Configuration

As an alternative to Spring Boot SSL Bundles, or to facilitate migration from legacy versions, SSL may be configured using standard `server.ssl.*` properties. This approach applies the configuration directly to the underlying Tomcat container, bypassing the Bundle abstraction.

Configuration example (`application.yml`):

```
server:
  port: 8443
  ssl:
    enabled: true
    key-store-type: PKCS12
    key-alias: "server"
    key-store: "classpath:server.p12"
    key-password: "keysecret"
```

```
    key-store-password: "storesecret"
```

This guide uses the **Standard Server SSL** configuration method. For a full list of valid properties, please refer to the **Official Spring Boot SSL Documentation**