



Critical
manufacturing
an ASMPPT company



Critical
manufacturing **11.0**

Installation Guide

March 2026



Table of Contents

1	Installation Guide	9
2	Introduction	10
2.1	Target Audience	10
2.2	Definitions	10
2.3	Naming Conventions	11
2.4	Documentation Set	12
2.5	Document History	13
2.6	Licensing	13
3	Preparation	14
3.1	Infrastructure	14
3.2	Storage and Network Shares	15
3.3	Accounts, Permissions and Active Directory Objects	15
3.4	SAP Connection Setup	15
3.5	Other infrastructure requirements	16
4	Planning Guides	17
5	Infrastructure	18
5.1	Optional components	18
6	AWS - Amazon Web Services	19
6.1	Time Estimate	19
6.2	Preconditions	19
6.2.1	Required Skills and Knowledge	19
6.2.2	Billable AWS Services	20
6.2.3	AWS Regions Support	21
6.2.4	AWS Service Limits	21
6.2.5	Secrets Management	22
6.2.6	Publicly Available Components	22
6.2.7	Required Software	22
6.3	Requirements	22
6.4	Configuration Sections	22
7	Network	23
7.1	Key Components	23
7.2	Configuration Steps	23
8	EKS - Elastic Kubernetes Service	26
8.1	Amazon Elastic Kubernetes Service	26
8.1.1	Cluster Nodes	26
8.1.2	Network	27
8.1.3	High Availability	27
8.2	Cluster Configuration	27
8.3	Troubleshooting	30
9	Storage	31
9.1	Cluster Storage	31
9.1.1	Option 1. EFS - Amazon Elastic File System	31
9.1.2	Option 2. NFS	34



9.2	EKS and Database Shared Storage	44
10	Ingress	45
10.1	Ingress Controller	45
10.2	Ports	46
10.3	TLS	46
10.4	Load Balancer	47
11	Encryption Data Configuration	48
11.1	Encryption in transit	48
11.1.1	Features	48
11.1.2	Volumes	48
11.2	Encryption at rest	48
11.2.1	Features	48
11.2.2	Volumes	49
12	External Services	50
13	SQL Server	51
13.1	EC2 Instances	51
13.2	AWS RDS Custom	51
14	Kafka	52
14.1	Amazon Managed Streaming for Apache Kafka - Amazon MSK	52
14.2	Confluent Cloud	52
15	RabbitMQ	53
15.1	Amazon MQ	53
16	S3	54
16.1	Amazon S3	54
17	ClickHouse	55
17.1	ClickHouse - AWS Marketplace	55
18	Accounts and Security	56
18.1	Critical Manufacturing Windows Services Account	56
18.2	SQL Server Accounts	56
18.3	ClickHouse	56
18.4	Kafka	56
18.5	RabbitMQ	57
18.6	S3	57
19	Database Servers	58
19.1	Planning for disaster recovery and high-availability	58
19.2	Database Server pre-requisites	58
19.3	SQL Server Licensing	59
19.4	Always On for Availability Groups pre-requisites	59
19.5	Preparing Windows Server Failover Cluster	59
19.6	Microsoft Distributed Transaction Coordinator (MS-DTC) Configuration	60
19.7	SQL Server Installation	61
19.8	SQL Server Reporting Services	70
19.8.1	Configuring Reporting Services for Critical Manufacturing	70



19.9	Creating Availability Groups in SQL Server	71
19.10	Microsoft Advanced Firewall Configuration	84
19.11	SQL Server Installation Advanced Topics	85
19.12	Installing to an Azure managed instance	85
20	Always On Availability Groups	86
20.1	Always On on Availability Groups versus on Failover Cluster Instances	86
20.2	Always On Terms	86
21	Breaking Up SQL Server Databases into Multiple Files	87
22	Database Filegroups and Data Files	88
23	Recommendations for SQL Server Installation Options	89
23.1	Install Analysis Services in Multidimensional and Data Mining Mode	89
23.2	Provision Storage For The Operating System and for SQL Server	89
23.3	Operating System Configuration	89
23.4	Service Accounts and Permission Granting	89
23.5	SQL Server Installation and Configuration	90
23.6	TempDB Configuration	90
23.7	Move TempDB to its own drive	90
23.8	Configuration of SQL Server Max Degree of Parallelism	91
23.9	Configuration of SQL Server Cost Threshold for Parallelism	91
23.10	Configuration of SQL Server Max Memory	91
23.11	SQL Server Maintenance Setup	91
23.12	Set Compatibility level	92
23.13	Other SQL Server Settings (to check after Critical Manufacturing has been installed)	92
23.14	Manually Enable Backup Jobs	92
23.15	Optional Settings	93
24	Storage and RAID Levels	94
25	TempDB Database Configuration	95
26	Additional required components	96
27	ClickHouse Connection Setup	97
27.1	Deployment	97
27.1.1	Fully managed ClickHouse service	97
28	Kafka Connection Setup	98
28.1	Authentication Methods	98
28.1.1	Using Mutual TLS authentication	98
28.1.2	Using SASL SSL Plain	98
28.2	Deployment	98
29	RabbitMQ Connection Setup	100
30	S3 Connection Setup	101
31	Application Servers	102
31.1	Planning for disaster recovery and high-availability	102
31.2	Application Server pre-requisites	102
31.3	.Net Framework Installation	103



32	Application Clients	104
33	File Shares	105
34	Email	106
34.1	Email configuration	106
35	Installation	107
35.1	Optional Component Installation	121
35.1.1	Main Installation Process	123
35.1.2	License Agreement	123
35.1.3	Package Sources	125
35.1.4	Package Selection	126
35.1.5	Update Product License	137
36	Post Installation	150
37	Connect IoT Introduction	151
38	IoT Runtime Components Configuration	152
38.1	Base Structure	152
38.2	Repository Structure	153
38.2.1	Type Npm Settings	153
38.2.2	Type Directory Settings	153
38.2.3	Directory Examples	153
38.3	Storage Structure	154
38.3.1	Storage Structure Examples	154
38.4	System Structure	154
38.4.1	Authentication Structure	155
38.4.2	Authentication type Password settings	155
38.4.3	Authentication type SecurityPortal settings	155
38.4.4	Examples	156
38.5	Logging Structure	156
38.5.1	Common Transport Options	157
38.5.2	Console Transport Options	158
38.5.3	Example	158
38.5.4	File Transport Options	158
38.5.5	File Transport Options Example	159
38.5.6	HTTP Transport Options	159
38.5.7	HTTP Transport Options Example	160
38.5.8	OTLP Transport Options	160
38.5.9	OTLP Transport Options Example	160
38.6	Monitor Structure	161
38.6.1	Monitor Structure Example	161
38.7	Controller Structure	161
38.7.1	Controller Structure Example	162
38.8	Driver Structure	162
38.8.1	Driver Structure Example	162
38.9	SslConfig structure	162
38.9.1	SslConfig Structure Example	163
38.10	Full Example	163



39	Connect IoT Installation	166
39.1	Package Repository	166
39.2	Deploy Connect IoT Packages	166
39.2.1	Package Selection	166
39.2.2	Connect IoT Configuration	167
39.2.3	MES Configurations	168
39.2.4	Connect IoT Repository Settings	169
39.2.5	Connect IoT Managers Configurations	170
39.3	Manually Deploy Packages	171
39.3.1	Directory Repository	171
39.3.2	NPM Repository	172
39.4	Install Automation Manager	173
39.5	Troubleshooting	174
39.5.1	“Unable to verify the first certificate”	174
39.5.2	“Unable to get local issuer certificate”	174
40	Theobald ERPConnect License	175
40.1	ERP Application Configuration Entry Files	175
40.2	Configuring ERP Integration	175
40.2.1	IntegrationSystem (Lookup Table)	176
40.2.2	IntegrationHandler (Generic Table)	176
40.2.3	IntegrationHandlerResolution (Smart Table)	176
41	Manually Set TRUSTWORTHY Database Property on Always On Systems	177
42	Report Server Security	178
43	Accessing SQL Server Reporting Services	179
44	Assigning or modifying a User/Group to a System Role	180
45	Defining Role-based security for the Reports Homepage	181
46	Defining Security by Folder	182
47	Critical Manufacturing Upgrade Procedure	183
47.1	Upgrade an MES Customer Environment	183
48	Uninstall	185
	Legal Information	186
	Disclaimer	186
	Confidentiality Notice	186
	Copyright Information	186
	Trademark Information	186



Summary of Tables

1	Definitions	10
2	Snippet visual conventions	11
3	Visual conventions	11
4	Guides available online	12
5	Included PDF documents	13
6	Servers	14
7	Accounts, Permissions and Active Directory Objects	15
8	SAP Connection Setup	16
9	Other infrastructure requirements	16
11	Database Server pre-requisites	58
12	Always On Availability Groups pre-requisites	59
13	SQL Server pre-requisites	61
14	Advanced Firewall Configuration	84
15	Always On Terms	86
16	ClickHouse connection setup	97
18	Kafka connection setup using Mutual TLS authentication	98
19	Kafka connection setup using SASL SSL Plain	98
20	RabbitMQ connection setup	100
21	S3 connection setup	101
22	Application Server pre-requisites	102
23	Application Server components	102
24	Connect IoT Configuration Tokens	152
25	Connect IoT Base Structure	152
26	Connect IoT Repository Structure	153
27	Npm specific settings	153
28	Directory specific settings	153
29	Connect IoT Storage Structure	154
30	Connect IoT System Structure	155
31	Authentication structure	155
32	Authentication type Password structure	155
33	Authentication type SecurityPortal structure	155
34	Connect IoT Logging Structure	157
35	Connect IoT Common Transport Options	157
36	Connect IoT Console Transport Options	158
37	Connect IoT File Transport Options	159
38	Connect IoT HTTP Transport Options	160
39	Connect IoT OTLP Transport Options	160
40	Connect IoT Monitor Structure	161
41	Connect IoT Controller Structure	161
42	Connect IoT Driver Structure	162
43	Connect IoT SslConfig Structure	163
44	Package Repository types	166
45	ERP Application Configuration Entry Files	175
46	IntegrationSystem Lookup Table	176
47	IntegrationHandler Generic Table	176
48	IntegrationHandlerResolution Smart Table	176



Summary of Figures

1	Application Modules	13
2	Infrastructure architecture and components	19
3	nodes_cluster_info	33
4	cluster_subnets	34
5	storage_class_example	34
6	setup_gateway	35
7	setup_gateway_platform_options	36
8	connect_to_aws	37
9	configure_gateway	38
10	gateway_overview	38
11	create_bucket	39
12	buckets_dashboard	39
13	file_share_settings	40
14	file_share_settings_2	41
15	amazon_s3_storage_settings	42
16	file_access_settings	43
17	file_access_settings_2	44
18	file_share_overview	44
19	Component Services	60
20	Local DTC Properties	61
21	SQL Server Installation - Feature Selection	62
22	SQL Server Installation - Name	63
23	SQL Server Installation - Service Account	64
24	SQL Server Installation - Collation	65
25	SQL Server Installation - Customization	65
26	SQL Server Installation - Database Engine	66
27	SQL Server Installation - Data Directories	67
28	SQL Server Installation - TempDB	68
29	SQL Server Installation - Analysis Services	69
30	SQL Server Installation - Analysis Services	70
31	image20211008134948048	71
32	image20211008140247218	72
33	image20211008140314467	73
34	image20211008135213480	74
35	image20211008140118280	75
36	image20211008140504827	76
37	image20211008140853232	77
38	image20211008141014208	78
39	image20211008141729589	79
40	image20211008142003760	80
41	image20211008142312338	81
42	image20211008142526796	82
43	image20211008142618283	82
44	image20211008142956229	83
45	image20211008143106749	83
46	image20211008143211260	84
47	SQL Server Always On Availability Groups	84



48	azureblobstorage	85
49	SQL Server Backup Jobs	92
50	create_env_button	107
51	create_action	108
52	step_target_package	108
53	step_target_package_configuration	109
54	step_configuration_general_data	110
55	step_configuration_database_1	111
56	step_configuration_database_2	112
57	step_configuration_dependencies_1	113
58	step_configuration_dependencies_2	113
59	step_configuration_security_ad	114
60	step_configuration_reports	115
61	step_configuration_printing	115
62	step_configuration_ecad	115
63	step_configuration_email	116
64	step_configuration_sap	116
65	step_configuration_servicerresources	117
66	step_configuration_services_cacerts	117
67	step_deploy_connect_to_cluster	119
68	step_deploy_check_installation	119
69	step_summary	119
70	access_mes_reset_password_on_first_login	120
71	home_page	120
72	assign_mes_oauth_role	121
73	iso_properties	121
74	Installation - Welcome screen	122
75	Installation - Field Validation	123
76	Installation - License Agreement review	124
77	Installation - License Agreement acceptance	125
78	Installation - Package Sources	126
79	Installation - Package Selection	127
80	ECAD Configuration.1	128
81	ECAD Configuration.2	129
82	ECAD Configuration.3	130
83	ECAD Configuration - PCBI Floating Service.2	131
84	ECAD Configuration.6	131
85	ECAD Configuration Summary	132
86	ECAD Installation Export	133
87	Printing Selection	134
88	Printing Installation	135
89	Printing Installation Summary	136
90	Printing Installation Export	137
91	installation_setup	138
92	installation_update_license	138
93	installation_authentication1	139
94	installation_import_parameter1	139
95	installation_import_parameter2	140
96	installation_license_select	141



97	installation_license_success	142
98	Installation - Welcome screen	143
99	installation_activation_code	144
100	Installation - Activation - Step 2	145
101	Installation - Activation - Step 3	145
102	Installation - Activation - Step 4	146
103	Installation - Activation - Step 5	146
104	Installation - Activation - Step 6	147
105	installation_offline1	147
106	installation_console1	148
107	installation_console2	149
108	installation_console3	149
109	Deploy Connect IoT Packages - Package Selection	167
110	Deploy Connect IoT Packages - Connect IoT Configurations	168
111	Deploy Connect IoT Packages - MES Configurations	169
112	Deploy Connect IoT Packages - Connect IoT Repository Settings	170
113	Deploy Connect IoT Packages - Connect IoT Managers Configurations	171
114	Rebuild database Powershell script	172
115	Running npm commands	173
116	Install Automation Manager	173



1 Installation Guide

The Critical Manufacturing MES installation guide provides you with detailed instructions on how to successfully deploy Critical Manufacturing MES. Critical Manufacturing MES is a high availability product that is optimized for transaction throughput with critical system requirements. For an overview of the system and its main components please refer to the [System Architecture](#) section.

Due to its demanding nature Critical Manufacturing MES requires careful planning and preparation. Before starting a new installation please follow the steps in the [Preparation](#) section very carefully as this is where you will find instructions that will help you prepare a new system to host Critical Manufacturing MES. This section provides a checklist of the system requirements and links to the detailed instructions on how to prepare each item in the checklist. You can access the complete PDF file of the Installation Guide here: [PDF](#), in PDF



2 Introduction

This document is a guide for planning, installing and configuring the Critical Manufacturing application.

- [Preparation](#) - Instructions and recommendations about how to prepare an infrastructure for installing Critical Manufacturing MES. It's not intended to be an exhaustive document about infrastructure but instead a quick guide on proven solutions for successfully deploying the solution.
- [Installation](#) - A visual walk-through of the installation wizard and the settings that must be entered before initiating a deployment.
- [Post Installation](#), in [Post Installation](#) - Best practices that are required to double check the health status of the solution and advanced configuration options that can be made after installation.
- [Upgrade](#) - The steps necessary to upgrade an existing installation.
- [Removal](#) - How to remove Critical Manufacturing MES from your system.

Should you require further assistance, please open a support ticket in the Critical Manufacturing Customer Portal at <https://portal.criticalmanufacturing.com>

2.1 Target Audience

This Installation Guide is intended for application administrators and should be used to support the Critical Manufacturing installation process.

2.2 Definitions

The next table identifies some common terms used throughout the document:

Table 1: Definitions

Acronym	Definition
ACL	Access Control List
AD	Active Directory
CMF	Critical Manufacturing
COM	Component Object Mode
DACL	Discretionary Access Control List
DB	DataBase
DWH	Data Warehouse
EFC	Electronic Failure Catalogue
HPC	High Performance Cluster
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IIS	Microsoft Internet Information Services
IP	Internet Protocol
MS-DTC	Microsoft Distributed Transaction Coordinator



Acronym	Definition
MSMQ	Microsoft Message Queue
NLB	Network Load Balancing
ODS	Operational Data Store
OLAP	Online Analytical Processing
OS	Operating System
SAN	Storage Area Network
SDDL	Security Descriptor Definition Language
SID	Security Identifier
SMTP	Simple Mail Transfer Protocol
URL	Uniform Resource Locator

2.3 Naming Conventions

The following conventions are applied throughout the document:


!!! info Helpful hints to assist in particular situations.

!!! note A note with important information.

!!! warning A warning or a word of caution.



These callouts carry important information can be very useful. However, in some situations, other smaller icons are visible and should be regarded as a visual identifier that a larger information callout of the same type can be found below. These icons include:

Table 2: Snippet visual conventions










Convention	Callout
:information_source:	Info
	Warning
	Note

Other visual indicators are used in the documentation to indicate information on specific actions or operations found in the MES GUI:

Table 3: Visual conventions

Convention	Description
	Described functionality is associated with the stated security feature
edit	Icon that symbolizes an edit action
	Icon that symbolizes an add action



Convention	Description
	Icon that symbolizes a remove action
	Icon that symbolizes a move up action
	Icon that symbolizes a move down action
	Icon that symbolizes the use of a camera or scanner
	Icon that symbolizes the use of a keypad
	Icon that symbolizes a browser modal window view action
	Icon that symbolizes a download action
	Icon that symbolizes a copy action
	Icon symbolizing PDF availability

2.4 Documentation Set

This Critical Manufacturing release comes with the following set of documents available online:

Table 4: Guides available online

Document	Target Audience	Description
User Guide	End Users	The standard application user interface manual (with a few exceptions which are covered in the Operations Guide and the Developer Guide/).
System Requirements	System Administrators	A description of the system requirements for Critical Manufacturing MES installation and operation.
Installation Guide	System Administrators	The application installation guide.
Operations Guide	System Administrators	A guide for maintaining, optimizing and troubleshooting the application.
Developer Portal	System Integrators	A guide to explain how to extend and customize the system.
Data Dictionary	System Administrators	A description of the database schema including the relationships of main objects.
Tutorials	End Users	Scenario creation and system operability for factory-floor and planning situations.

This Critical Manufacturing release also includes the following separate PDF documents:



Table 5: Included PDF documents

Document	File Name	Description
Installation Guide	InstallationGuide.pdf	The application installation guide.
Release Notes	ReleaseNotes.pdf	The release notes that includes a description of the new features as well as the description of any changes.
System Requirements	SystemRequirements.pdf	A description of the system requirements for Critical Manufacturing installation and operation.

!!! note The PDF version of this document included in the Critical Manufacturing release is an exported version of the web-based guide.

2.5 Document History

This document was developed prior to the product release to manufacturing and as such, it cannot be guaranteed that all details included herein will be exactly as what is found in the shipping product. Critical Manufacturing cannot guarantee the accuracy of any information presented after the date of publication.

The date of the document changes whenever a new edition is released. However, it must be noticed that some product updates do not necessarily require document changes, and as such, versions of the product may be released without accompanying documentation and vice versa.

2.6 Licensing

The Critical Manufacturing base license includes the Core module. All other modules are licensed separately.

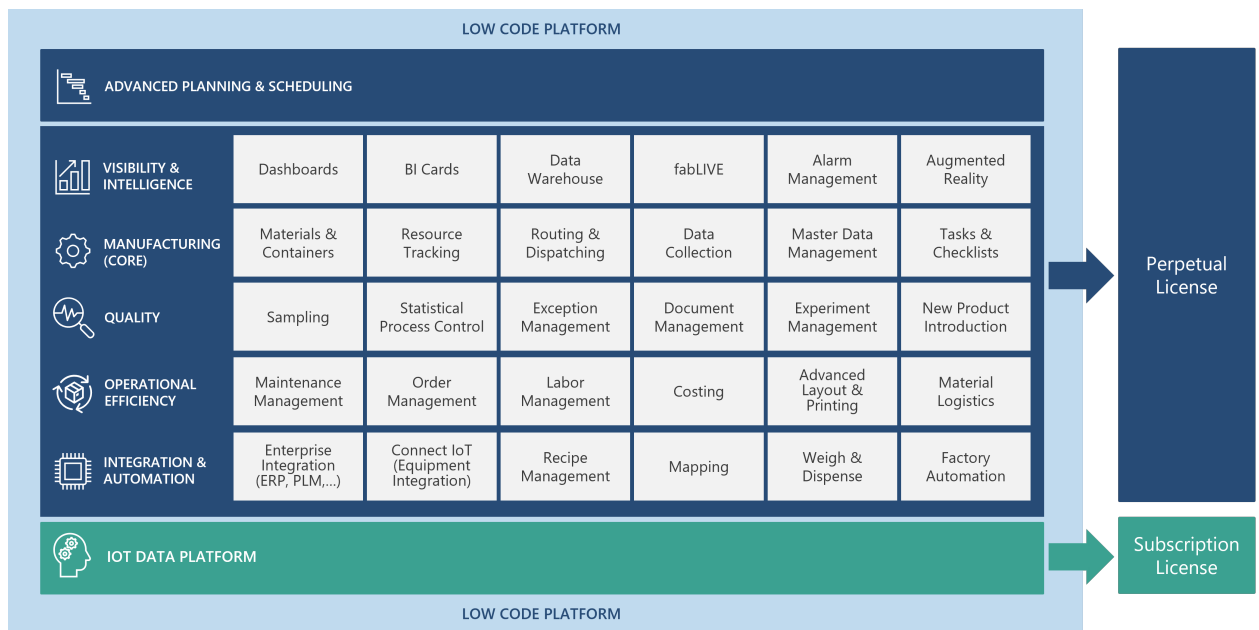


Figure 1: Application Modules



3 Preparation

Critical Manufacturing provides you with the following checklist that serves as a quick guide and helps determine the preliminary procedures and requirements for the installation to take place. Below you can see links to information resources to get started:

- [\[\[operation-guide-container-stack-index|Container Stack Documentation\]\]](#) section of the Operations Guide to know the components of the stack.
- [DevOps Center Documentation](#) in the Critical Manufacturing Customer Portal for installation instructions.
- [\[\[system-requirements-application-layer-applicationlayercontainers|Software Requirements\]\]](#) in the System Requirements.

3.1 Infrastructure

This section contains a list of the computer, server and client roles required to run Critical Manufacturing MES. Generically, we recommend that you deploy each of the application tiers on separate machines. For development environments, you can deploy all tiers on a single machine. If a machine is hosting several tiers, it must fulfill all the requirements of the current tier.

Table 6: Servers

Item	Description	Planning and installation guides	Checked
Hardware infrastru	Plan the number of servers and the computing, storage and memory requirements for each one	[[system-requirements-application-layer-applicationlayercontainers]] [[system-requirements-application-layer-applicationlayeroptional]]	
Database Servers	The database servers will host Critical Manufacturing persistency layer components	[[installation-guide-database-servers-index	Planning and installing database servers]]
Applicatio Servers	The application servers will host Critical Manufacturing application layer components	[[installation-guide-planningapplicationservers	Planning and installing the application servers]]
Applicatio Clients	The application clients will run Critical Manufacturing presentation layer components in a browser	[[installation-guide-planningapplicationclients	Planning and preparing application clients]]



3.2 Storage and Network Shares

Information on file shares and volume configurations is available on the Customer Portal support website [here](#).

3.3 Accounts, Permissions and Active Directory Objects

In this section, we summarize what must be prepared in terms of account configurations:

Table 7: Accounts, Permissions and Active Directory Objects

Item	Description	Planning and installation guide	Checked
Deployment Account	The account that will be used to deploy Critical Manufacturing	[[installation-guide-accountsandsecurity]]	
Critical Manufacturing Windows Services Account	The account that will be used to run the Critical Manufacturing Windows Service applications	[[installation-guide-accountsandsecurity]]	
Microsoft SQL Server User Account	The account that will be used to connect to the Microsoft SQL Server instance	[[installation-guide-accountsandsecurity]]	
Microsoft SQL Server Analysis Service User Account	The account that will be used to connect to the OLAP cubes created by Critical Manufacturing	[[installation-guide-accountsandsecurity]]	
Microsoft SQL Server Reporting Services User Account	The account that will be used in Reporting Services to host Critical Manufacturing reports	[[installation-guide-accountsandsecurity]]	
Active Directory Domain Services FQDN	The FQDN for the Active Directory domain services server	[[installation-guide-accountsandsecurity]]	
Active Directory Domain Name	The shorter domain name that is used to prefix user accounts	[[installation-guide-accountsandsecurity]]	
ClickHouse User Account Name	The account that will be used to connect to ClickHouse	[[installation-guide-accountsandsecurity]]	
Kafka User Account Name	The account that will be used to connect to Kafka	[[installation-guide-accountsandsecurity]]	
S3 User Account Name	The account that will be used to connect to S3	[[installation-guide-accountsandsecurity]]	

3.4 SAP Connection Setup

This section outlines the configuration required to support SAP integration when using the Theobald ERP Connector. If the integration is implemented via web services or other protocols, this setup is not required.



Table 8: SAP Connection Setup

Item	Description	Planning and installation guide	Checked
SAP Host	Critical Manufacturing will connect to this SAP instance to receive inbound messages and to send outbound messages	SAP Integration	
SAP System Number	The SAP system name to connect to	SAP Integration	
SAP Service	The SAP RFC gateway hostname	SAP Integration	
SAP Program Id	The name of the Remote Server Program that will be used for IDOC exchange	SAP Integration	
SAP User Name	The user name used to authenticate in SAP	SAP Integration	
SAP Password	The password used to authenticate in SAP	SAP Integration	
SAP Client	The SAP Client number	SAP Integration	
ERP Connector-License Key	The Theobald ERP Connect license required to communicate with SAP	SAP Integration	

3.5 Other infrastructure requirements

In this section, we summarize the configurations for the email infrastructure required for notifications:

Table 9: Other infrastructure requirements

Item	Description	Planning and installation guide	Checked
SMTP Server Address	Critical Manufacturing requires an email server to send notifications through email (SSL is both supported and recommended)	[[installation-guide-email]]	
SMTP Server User Name	The account used to authenticate at the email server	[[installation-guide-email]]	
SMTP Server Password	The password used to authenticate at the email server	[[installation-guide-email]]	
Support Email Address	Critical Manufacturing will send email messages to this destination	[[installation-guide-email]]	
Support Email From Name	Critical Manufacturing will send email messages using this text to identify the email sender	[[installation-guide-email]]	



4 Planning Guides

In this section you have step-by-step instructions on how to plan and prepare systems for deploying Critical Manufacturing MES. Refer to the [Preparation](#) article for an overview and checklist of the installation process requirements.

- [Infrastructure](#)
- [AWS - Amazon Web Services](#), in [AWS - Amazon Web Services](#)
- [Database Servers](#), in [Database Servers](#)
- [Additional Components](#), in [Additional Components](#)
- [Application Servers](#)
- [Application Clients](#)
- [File Shares](#)
- [Accounts and Security](#)
- [Email](#)



5 Infrastructure

This guide contains step-by-step instructions on how to successfully plan and deploy an infrastructure for Critical Manufacturing MES. See the System Requirements sections below to find out more:

- [Application Layer - Container Stack](#)

5.1 Optional components

Some components of the application layer of Critical Manufacturing MES require traditional methods of installation, unlike the main stack of container-based installation:

- [Application Layer - Optional Components](#)

6 AWS - Amazon Web Services

This guide will walk you through the steps to configure the necessary resources to have an AWS infrastructure ready to host Critical Manufacturing MES. The configuration includes several components needed to run the MES, such as a Kubernetes cluster, database optionally running on AWS, shared storage solutions, network configurations for secure and reliable communication, and load balancers for traffic management.

The following architecture diagram shows how the Critical Manufacturing MES can be deployed on AWS cloud infrastructure, including, among others, the kubernetes cluster for the application servers, SQL Server database, and external services such as Rabbit MQ and ClickHouse.

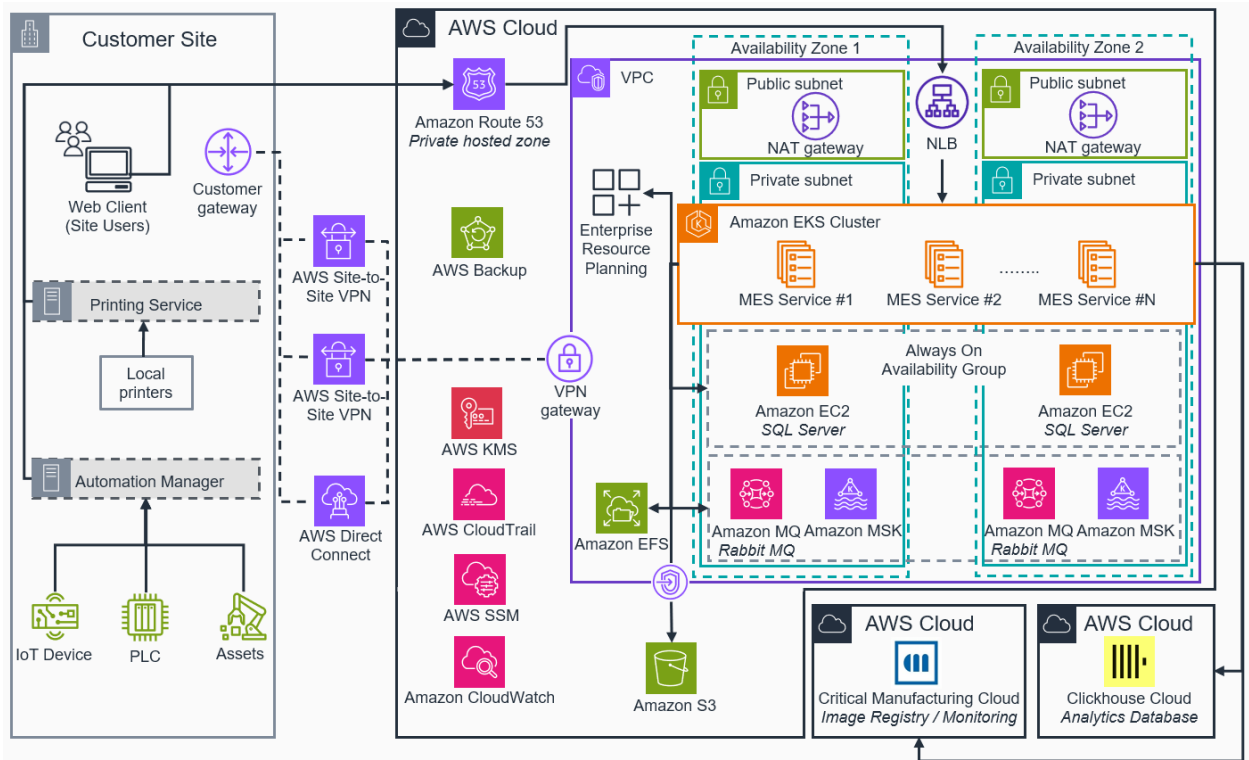


Figure 2: Infrastructure architecture and components

6.1 Time Estimate

This guide includes the creation of several infrastructure resources and may take up to an hour to complete.

6.2 Preconditions

6.2.1 Required Skills and Knowledge

It is assumed the user has basic knowledge of some AWS services, such as:

1. **AWS Services:** Understanding of Amazon Web Services (AWS), specifically VPC and networking, EKS, EC2, IAM, and Route 53.
2. **Kubernetes:** Familiarity with Kubernetes concepts such as clusters, nodes, pods, and services.
3. **EKS:** Experience with Amazon EKS and the deployment of Kubernetes clusters on AWS.



4. **Command Line Interface:** Ability to work with AWS CLI and `eksctl` for infrastructure management.
5. **IAM Roles and Policies:** Knowledge of AWS IAM, including the creation of policies and roles for managing service accounts and permissions.
6. **Storage Classes:** Familiarity with Amazon EFS, S3, and Storage Gateways, and their integration with Kubernetes.

6.2.2 Billable AWS Services

When deploying an EKS-based environment on AWS, there are several AWS services that will incur costs. Below is a list of the billable services and an indication of whether each is **mandatory** or **optional** for the deployment.

Service	Mandatory	Description	Pricing
Amazon VPC (Virtual Private Cloud)	:white_checkmark	Required to create the network environment. There are costs for NAT gateways, VPC endpoints, Load Balancers, and data transfer.	https://aws.amazon.com/vpc/pricing/
Amazon EKS (Elastic Kubernetes Service)	:white_checkmark	Required for running Kubernetes clusters. Costs include control plane fees and per-node pricing.	https://aws.amazon.com/eks/pricing/
Amazon EC2 (Elastic Compute Cloud)	:white_checkmark	Required for provisioning worker nodes. EC2 instance costs are based on the instance type, and pricing is pay-per-hour or reserved.	https://aws.amazon.com/ec2/pricing/
Amazon EFS (Elastic File System)	:octocat	Used for shared storage across the cluster. If persistent storage is needed, EFS may be necessary.	https://aws.amazon.com/efs/pricing/



Service	Manda	Description	Pricing
Amazon S3 (Simple Storage Service)	:octicon-dash-16: (at least one of the storage solutions is require	Used for object storage. Costs include storage fees and data retrieval.	https://aws.amazon.com/s3/pricing/
AWS Storage Gateway (S3 File Storage Gateway)	:octicon-dash-16: (at least one of the storage solutions is require	Depending on the chosen Volume Types, it may needed to create a Storage Gateway to share data in Amazon S3 and access it locally via file gateways.	https://aws.amazon.com/storagegateway/pricing/
Amazon Route 53	:octicon-dash-16:	If domain name resolution and custom DNS are required for your deployment, Route 53 will be used for managing DNS.	https://aws.amazon.com/route53/pricing/

6.2.3 AWS Regions Support

Critical Manufacturing MES does not impose any AWS Region restriction by itself. For details on regional support for each required AWS service, please consult <https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/>.

6.2.4 AWS Service Limits

When deploying to Amazon EKS, you might encounter AWS service limits. These limits, also known as [AWS Service Quotas](#), include constraints on the number of EC2 instances, EBS volumes, NAT gateways, and other resources that can be provisioned within an AWS account.

Before starting a deployment, it is recommended to check your current service limits to avoid interruptions during cluster creation and management. To view your limits:

1. Go to the [Service Quotas Dashboard](#).



2. In the AWS Management Console, you can view your limits for different services such as EC2, VPC, and EKS.
3. If any limit is too low, submit a request to increase the limit by clicking **Request quota increase**.

6.2.5 Secrets Management

Critical Manufacturing does not require any specific service for managing secrets, as all the sensitive data is stored as Kubernetes Secrets. For more details, check https://portal.criticalmanufacturing.com/Help/devops-center/deployment-targets/kubernetes-section/kubernetes_sensitive_data/.

6.2.6 Publicly Available Components

Which components and services need to be publicly available to outside of the AWS infrastructure depends on your specific configuration and requirements. In a deployment following the suggested architecture, where all the components are deployed and running in the AWS Cloud, the only resource that needs to be publicly exposed and available to ingress traffic is the Load Balancer / DNS managing service.

6.2.7 Required Software

It is possible to create the cluster in three different ways:

- [AWS Management Console Web GUI](#)
- [AWS CLI](#)
- [eksctl](#)

In this guide, we will use [eksctl](#), version 0.114.0, and AWS CLI, so these tools must be installed. More information on how to install this command can be found [here](#).

6.3 Requirements

If the cluster will be used to deploy a Critical Manufacturing MES installation, the requirements must be taking into account. You can check the MES requirements in the [\[\[system-requirements-application-layer-applicationlayercontainers\]\]](#) page.

6.4 Configuration Sections

The following sections will cover:

-
-
-
-
-
-

Each section will focus on the key configurations and requirements to ensure the MES system operates efficiently on AWS infrastructure.



7 Network

This section describes how to create and configure the AWS Network components required to allow communication between the several application components, the database, storage, external components service, and incoming web traffic.

Critical Manufacturing MES requires a well-configured network to ensure secure, fast, and reliable communication between its components (Kubernetes pods, SQL Server database, and external services). The following configurations are essential for proper operation.

7.1 Key Components

- **VPC:** Create an isolated Virtual Private Cloud (VPC) for the MES system, ensuring that the EKS cluster, SQL Server, and other components are secure and have controlled access.
- **Subnets:** Use both public and private subnets:
 - **Public Subnets** for exposing load balancers and external services.
 - **Private Subnets** for keeping the MES containers and SQL Server secure from direct internet exposure.
- **Security Groups:** Define security groups to control traffic, such as allow ingress from the load balancers to the EKS services.
- **NAT Gateways:** Ensure private subnets have outbound internet access for software updates and external communication needs of MES pods.
- **Internet Gateways:** Allows resources in the public subnets, with a public IP, to access the internet, both inbound and outbound traffic.

This setup provides an isolated and secure network for the MES components while maintaining access to essential external services.

!!! note More information: <https://docs.aws.amazon.com/vpc/>

7.2 Configuration Steps

=== "AWS Management Console"

1. Open the [AWS Management Console] (<https://console.aws.amazon.com/console/home>)
1. Search for **VPC** and open
1. Click **Create VPC**
1. Choose **VPC and more**
1. If required, change the configuration according to your needs
1. ![Create VPC] (images/vpc_create.png)
1. Click **Create VPC**

=== "AWS CLI"

Example steps for the creation of a VPC, with two Availability Zones, each one with a private subnet, a public subnet, and outbound Internet connection:

```
```bash
Step 1: Create VPC
echo "Creating VPC..."
VPC_ID=$(aws ec2 create-vpc --cidr-block "10.0.0.0/16" --tag-specifications
 'ResourceType=vpc,Tags=[{Key=Name,Value=cm-vpc}]' --query 'Vpc.VpcId' --output text)
echo "VPC ID: $VPC_ID"
```



```
Step 2: Modify VPC attributes (Enable DNS hostnames)
echo "Enabling DNS hostnames for the VPC..."
aws ec2 modify-vpc-attribute --vpc-id $VPC_ID --enable-dns-hostnames "{\"Value\":true}"

Step 3: Create public subnets
echo "Creating public subnets..."
SUBNET_PUBLIC1=$(aws ec2 create-subnet --vpc-id $VPC_ID --cidr-block "10.0.0.0/20" --availability-zone
 "us-east-1a" --tag-specifications
 'ResourceType=subnet,Tags=[{Key=Name,Value=cm-subnet-public1-us-east-1a}]' --query
 'Subnet.SubnetId' --output text)
SUBNET_PUBLIC2=$(aws ec2 create-subnet --vpc-id $VPC_ID --cidr-block "10.0.16.0/20" --availability-zone
 "us-east-1b" --tag-specifications
 'ResourceType=subnet,Tags=[{Key=Name,Value=cm-subnet-public2-us-east-1b}]' --query
 'Subnet.SubnetId' --output text)
echo "Public Subnet 1 ID: $SUBNET_PUBLIC1"
echo "Public Subnet 2 ID: $SUBNET_PUBLIC2"

Step 4: Create private subnets
echo "Creating private subnets..."
SUBNET_PRIVATE1=$(aws ec2 create-subnet --vpc-id $VPC_ID --cidr-block "10.0.128.0/20"
 --availability-zone "us-east-1a" --tag-specifications
 'ResourceType=subnet,Tags=[{Key=Name,Value=cm-subnet-private1-us-east-1a}]' --query
 'Subnet.SubnetId' --output text)
SUBNET_PRIVATE2=$(aws ec2 create-subnet --vpc-id $VPC_ID --cidr-block "10.0.144.0/20"
 --availability-zone "us-east-1b" --tag-specifications
 'ResourceType=subnet,Tags=[{Key=Name,Value=cm-subnet-private2-us-east-1b}]' --query
 'Subnet.SubnetId' --output text)
echo "Private Subnet 1 ID: $SUBNET_PRIVATE1"
echo "Private Subnet 2 ID: $SUBNET_PRIVATE2"

Step 5: Create and attach an Internet Gateway
echo "Creating Internet Gateway..."
IGW_ID=$(aws ec2 create-internet-gateway --tag-specifications
 'ResourceType=internet-gateway,Tags=[{Key=Name,Value=cm-igw}]' --query
 'InternetGateway.InternetGatewayId' --output text)
echo "Attaching Internet Gateway to VPC..."
aws ec2 attach-internet-gateway --internet-gateway-id $IGW_ID --vpc-id $VPC_ID

Step 6: Create a public route table and routes
echo "Creating public route table..."
RTB_PUBLIC=$(aws ec2 create-route-table --vpc-id $VPC_ID --tag-specifications
 'ResourceType=route-table,Tags=[{Key=Name,Value=cm-rtb-public}]' --query 'RouteTable.RouteTableId'
 --output text)
echo "Creating route for Internet access in the public route table..."
aws ec2 create-route --route-table-id $RTB_PUBLIC --destination-cidr-block "0.0.0.0/0" --gateway-id
 $IGW_ID

Step 7: Associate public route table with public subnets
echo "Associating public route table with public subnets..."
aws ec2 associate-route-table --route-table-id $RTB_PUBLIC --subnet-id $SUBNET_PUBLIC1
aws ec2 associate-route-table --route-table-id $RTB_PUBLIC --subnet-id $SUBNET_PUBLIC2

Step 8: Allocate an Elastic IP for NAT Gateway
echo "Allocating Elastic IP for NAT Gateway..."
```



```
EIP_ALLOC_ID=$(aws ec2 allocate-address --domain "vpc" --tag-specifications
 'ResourceType=elastic-ip,Tags=[{Key=Name,Value=cm-eip-us-east-1a}]' --query 'AllocationId'
 --output text)

Step 9: Create a NAT Gateway in the public subnet
echo "Creating NAT Gateway..."
NAT_GW_ID=$(aws ec2 create-nat-gateway --subnet-id $SUBNET_PUBLIC1 --allocation-id $EIP_ALLOC_ID
 --tag-specifications 'ResourceType=natgateway,Tags=[{Key=Name,Value=cm-nat-public1-us-east-1a}]'
 --query 'NatGateway.NatGatewayId' --output text)
echo "NAT Gateway ID: $NAT_GW_ID"

Step 10: Create private route tables and routes through NAT Gateway
echo "Creating private route table 1..."
RTB_PRIVATE1=$(aws ec2 create-route-table --vpc-id $VPC_ID --tag-specifications
 'ResourceType=route-table,Tags=[{Key=Name,Value=cm-rtb-private1-us-east-1a}]' --query
 'RouteTable.RouteTableId' --output text)
echo "Creating route in private route table 1 through NAT Gateway..."
aws ec2 create-route --route-table-id $RTB_PRIVATE1 --destination-cidr-block "0.0.0.0/0"
 --nat-gateway-id $NAT_GW_ID

echo "Creating private route table 2..."
RTB_PRIVATE2=$(aws ec2 create-route-table --vpc-id $VPC_ID --tag-specifications
 'ResourceType=route-table,Tags=[{Key=Name,Value=cm-rtb-private2-us-east-1b}]' --query
 'RouteTable.RouteTableId' --output text)
echo "Creating route in private route table 2 through NAT Gateway..."
aws ec2 create-route --route-table-id $RTB_PRIVATE2 --destination-cidr-block "0.0.0.0/0"
 --nat-gateway-id $NAT_GW_ID

Step 11: Associate private route tables with private subnets
echo "Associating private route tables with private subnets..."
aws ec2 associate-route-table --route-table-id $RTB_PRIVATE1 --subnet-id $SUBNET_PRIVATE1
aws ec2 associate-route-table --route-table-id $RTB_PRIVATE2 --subnet-id $SUBNET_PRIVATE2

Step 12: Verify Route Tables
echo "Verifying route tables..."
aws ec2 describe-route-tables --route-table-ids $RTB_PRIVATE1 $RTB_PRIVATE2
...

```



## 8 EKS - Elastic Kubernetes Service

This section describes how to create and configure the Kubernetes cluster that runs, manages, and orchestrates the several application containers.

After the creation of the basic network resources, we need to create the Kubernetes Cluster that will run and manage the several application containers of the Critical Manufacturing MES. To that end, we will use Amazon EKS.

### 8.1 Amazon Elastic Kubernetes Service

The Amazon Elastic Kubernetes Service (EKS) is AWS's managed Kubernetes cloud platform.

In the DevOps Center, you can remotely deploy a Customer Environment to an EKS cluster.

!!! note More information: <https://docs.aws.amazon.com/eks/>

#### 8.1.1 Cluster Nodes

Cluster Nodes are the actual machines that provide computational power to the cluster, it's where the pods and containers are deployed and run.

For EKS, Critical Manufacturing recommends the usage of **Managed nodes**. Managed nodes are EC2 instances that AWS automatically provisions, manages, and scales within a Kubernetes cluster. These nodes are part of EKS Managed Node Groups, which simplify node lifecycle management, including upgrades and security patches, while integrating with Kubernetes and EC2 features.

EKS Managed nodes have several key features, highlighting the following:

- **Automated Provisioning**
  - AWS automatically creates EC2 instances, registers them with the Kubernetes control plane, and configures them to run as worker nodes in the cluster.
- **Simplified Upgrades**
  - Managed node groups allow easy updates to the latest Amazon EKS-optimized AMI versions, which include the latest Kubernetes versions, security patches, and performance enhancements.
  - During updates, nodes are drained (pods are evicted), upgraded, and then brought back online with minimal disruption.
- **Auto-scaling**
  - Managed node groups integrate with EC2 Auto Scaling, enabling automatic scaling of nodes based on workload demands. If more compute capacity is needed, the node group scales out; when demand drops, it scales in to optimize resource usage and cost.
- **Integrated Security**
  - Managed nodes are launched with the latest EKS-optimized AMI, which includes built-in security configurations for enhanced protection.
  - IAM roles can be assigned to the nodes to control access to AWS resources.
  - Managed node groups can be configured with Security Groups to define network traffic rules.

!!! warning Currently, **Fargate nodes are not supported**. Critical Manufacturing recommends the usage of Managed nodes.



## 8.1.2 Network

The network in an EKS cluster is built using Amazon VPC (Virtual Private Cloud), and it leverages AWS networking services for scaling, security, and performance.

You can find more information on the subject in [\[installation-guide-aws-network\]](#).

## 8.1.3 High Availability

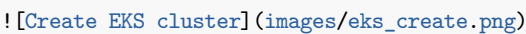
High availability in an EKS cluster ensures that workloads and applications remain accessible, even in the event of failures or disruptions. EKS, running on AWS, allows the building of highly available clusters with several built-in features.

- Multi-AZ (Availability Zone) Deployments
  - Node Groups: EKS enables the creation of worker node groups spread across multiple Availability Zones (AZs) within a region. By distributing nodes across AZs, the impact of an outage in a single zone is minimized.
  - Load Balancing: AWS Load Balancers can route traffic across nodes in multiple AZs to ensure consistent traffic handling even if some nodes go down.
- Managed Control Plane
  - Master Nodes: EKS provides a managed Kubernetes control plane (master nodes), which is distributed across multiple AZs within a region by default. AWS automatically maintains and ensures the high availability of the control without manual intervention.
  - Scaling: The EKS control plane scales horizontally and can handle disruptions due to hardware failure or updates, ensuring continuous availability.
- Auto Scaling
  - Cluster Autoscaler: Dynamically adjusts the number of worker nodes (EC2 Instances) in the cluster based on resource demand. It can automatically add new nodes when workloads require more capacity and remove them when demand decreases.

## 8.2 Cluster Configuration

To create the EKS cluster, it is assumed that a VPC has already been configured according to [Network](#) section.

=== "AWS Management Console"

1. On [\[AWS Management Console\]](https://console.aws.amazon.com/console/home) (<https://console.aws.amazon.com/console/home>) search for **EKS** and open
1. Click **Add cluster**, "Create"  

1. Set a **Cluster Name** and a **Cluster IAM role**
1. On the **Networking** section, select the VPC, and the appropriate subnets
1. Proceed until the end of the wizard and create the EKS cluster
1. After a few minutes the cluster is created and ready.

=== "AWS CLI"

!!! note

The following command must be run on the same console window as the VPC creation; otherwise, the variables value must be set manually.



## 1. Create the required [IAM

Roles] (<https://docs.aws.amazon.com/eks/latest/userguide/cluster-iam-role.html>)

```
```bash
# Create IAM Role for EKS Cluster
echo "Creating IAM Role for EKS Cluster..."
EKS_ROLE_NAME="eks-cluster-role"
EKS_ROLE_ARN=$(aws iam create-role \
  --role-name $EKS_ROLE_NAME \
  --assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "eks.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }' \
  --query 'Role.Arn' --output text)
echo "IAM Role ARN for EKS Cluster: $EKS_ROLE_ARN"

# Attach necessary policies to the EKS role
echo "Attaching AmazonEKSClusterPolicy to EKS Cluster Role..."
aws iam attach-role-policy --role-name $EKS_ROLE_NAME --policy-arn arn:[] (images\.png){ width=20px
}iam:[] (images\.png){ width=20px }policy/AmazonEKSClusterPolicy
aws iam attach-role-policy --role-name $EKS_ROLE_NAME --policy-arn arn:[] (images\.png){ width=20px
}iam:[] (images\.png){ width=20px }policy/AmazonEKSServicePolicy

# Create IAM Role for EKS Node Group
echo "Creating IAM Role for EKS Node Group..."
NODE_ROLE_NAME="eks-nodegroup-role"
NODE_ROLE_ARN=$(aws iam create-role \
  --role-name $NODE_ROLE_NAME \
  --assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "ec2.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }' \
  --query 'Role.Arn' --output text)
echo "IAM Role ARN for EKS Node Group: $NODE_ROLE_ARN"

# Attach necessary policies to the Node Group role
echo "Attaching AmazonEKSServicePolicy and other policies to EKS Node Group Role..."
aws iam attach-role-policy --role-name $NODE_ROLE_NAME --policy-arn arn:[] (images\.png){ width=20px
```



```
    }iam:!(images\.png){ width=20px }policy/AmazonEKSEKSWorkerNodePolicy
aws iam attach-role-policy --role-name $NODE_ROLE_NAME --policy-arn arn:!(images\.png){ width=20px
    }iam:!(images\.png){ width=20px }policy/AmazonEC2ContainerRegistryReadOnly
aws iam attach-role-policy --role-name $NODE_ROLE_NAME --policy-arn arn:!(images\.png){ width=20px
    }iam:!(images\.png){ width=20px }policy/AmazonEKSCNIPolicy
...

```

1. Create the EKS cluster

```
```bash
Ensure you have the necessary environment variables set up from the previous script:
VPC_ID, SUBNET_PUBLIC1, SUBNET_PUBLIC2, SUBNET_PRIVATE1, SUBNET_PRIVATE2, EKS_ROLE_ARN,
 NODE_ROLE_ARN

Create EKS Cluster (public and private cluster endpoint access)
echo "Creating EKS Cluster 'cm-cluster'..."
aws eks create-cluster \
 --name cm-cluster \
 --role-arn $EKS_ROLE_ARN \
 --resources-vpc-config subnetIds=$SUBNET_PUBLIC1,$SUBNET_PUBLIC2,$SUBNET_PRIVATE1,
 $SUBNET_PRIVATE2,endpointPublicAccess=true,endpointPrivateAccess=true \
 --kubernetes-version 1.31 \
 --region eu-east-1

Wait for the EKS Cluster to be active (this can take a few minutes)
echo "Waiting for the EKS cluster to become ACTIVE..."
aws eks wait cluster-active --name cm-cluster

Create Node Group (with 2 nodes of type m5.large)
echo "Creating Node Group 'cm-nodegroup' with 2 m5.large instances..."
aws eks create-nodegroup \
 --cluster-name cm-cluster \
 --nodegroup-name cm-nodegroup \
 --scaling-config minSize=2,maxSize=2,desiredSize=2 \
 --disk-size 20 \
 --subnets $SUBNET_PUBLIC1 $SUBNET_PUBLIC2 \
 --instance-types m5.large \
 --ami-type AL2_x86_64 \
 --node-role $NODE_ROLE_ARN \
 --region eu-east-1

Wait for the Node Group to be active
echo "Waiting for the Node Group to become ACTIVE..."
aws eks wait nodegroup-active --cluster-name cm-cluster --nodegroup-name cm-nodegroup

echo "EKS Cluster and Node Group successfully created!"
...

```

=== "eksctl"

#### !!! note

The following command must be run on the same console window as the VPC creation; otherwise, the variables value must be set manually.

```
```bash
```



```
eksctl create cluster \  
  --name cm-cluster \  
  --region us-east-1 \  
  --vpc-public-subnets $SUBNET_PUBLIC1,$SUBNET_PUBLIC2 \  
  --vpc-private-subnets $SUBNET_PRIVATE1,$SUBNET_PRIVATE2 \  
  --managed \  
  --node-type m5.large \  
  --nodes 2 \  
  --node-private-networking  
...
```

Cluster creation takes several minutes but some output lines can be seen. When all operations are done, the following line is shown:

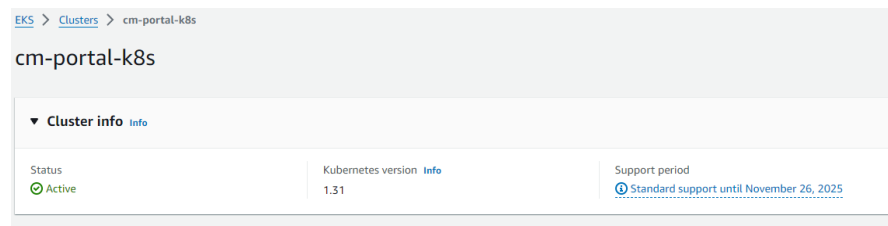
```
![cluster_creation] (images/cluster_creation.png)
```

After this, your EKS cluster is running and ready.

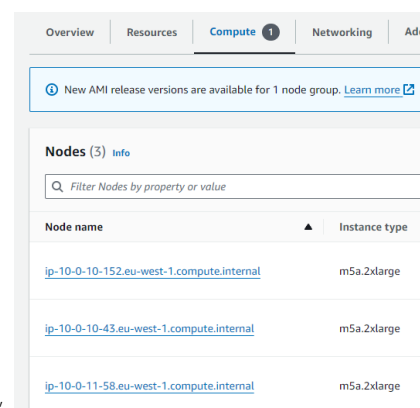
8.3 Troubleshooting

A user can follow several steps to troubleshoot issues with the cluster.

1. Verify the cluster status to ensure it is 'Active'.



2. On the Compute tab of the EKS Cluster page, check the node status to confirm they are 'Ready'. If a node is not 'Ready,' you can click on it to view its events at the bottom of the page.





9 Storage

This section describes how to configure storage to serve as backend for volumes or storage classes in your EKS cluster.

9.1 Cluster Storage

We will present two different options for the cluster storage, allowing you to choose the one that best fits your needs.

- **EFS** - Amazon Elastic File System
- **NFS File Share** (backed-up by an S3 File Gateway)

!!! info We recommend using EFS as it is a fully managed service and therefore does not require manual management or maintenance.

9.1.1 Option 1. EFS - Amazon Elastic File System

!!! note More information: <https://docs.aws.amazon.com/eks/latest/userguide/efs-csi.html>

- Before creating the storage class, verify that an Amazon EFS Container Storage Interface (CSI) Driver is deployed to the previously created Amazon EKS cluster. Also, verify that an AWS Identity and Access Management (IAM) OpenID Connect (OIDC) provider exists for the cluster. To check this topic, follow [this link](#).

– Create an IAM policy and role

1. Create an IAM policy

- * Download the IAM policy that allows the CSI driver's service account to make calls to AWS APIs. This IAM policy is designed for the AWS EFS CSI (Container Storage Interface) driver, which allows Kubernetes to manage Amazon EFS (Elastic File System) volumes. This policy grants permissions needed for EFS operations from within a Kubernetes cluster.

```
curl -o iam-policy-example.json
https://raw.githubusercontent.com/kubernetes-sigs/aws-efs-csi-driver/master/docs/iam-policy-exa
```

- * Create the policy based on the previously downloaded json.

```
aws iam create-policy --policy-name AmazonEKS_EFS_CSI_Driver_Policy --policy-document
file://iam-policy-example.json
```

2. Create an IAM role and attach the IAM policy to it.

- * First, you need to find the cluster's OpenID Connect (OIDC) provider URL. This is possible through the following command:

```
aws eks describe-cluster --name my-cluster --query "cluster.identity.oidc.issuer" --output text
```

The output must be as follows:

```
https://oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE
```

- * Create the IAM role, granting the Kubernetes service account the AssumeRoleWithWebIdentity action. This action allows the role to be assumed by an entity using a Web Identity Federation token (in this case, an OIDC token from Kubernetes).
- * Copy the following contents to a file named trust-policy. Replace **111122223333** with your account ID. Replace **EXAMPLED539D4633E53DE1B71EXAMPLE** and **region-code** with the values returned in the previous step.



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub",
        "iam::111122223333:oidc-provider/oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:kube-system:efs-csi-controller-sa"
        }
      }
    }
  ]
}
```

* Now, create the role.

```
aws iam create-role
--role-name AmazonEKS_EFS_CSI_DriverRole
--assume-role-policy-document file://"trust-policy.json"
```

* Attach the IAM policy to the role with the following command. Replace **111122223333** with your account ID.

```
aws iam attach-role-policy
--policy-arn arn:aws:iam::111122223333:policy/AmazonEKS_EFS_CSI_Driver_Policy
--role-name AmazonEKS_EFS_CSI_DriverRole
```

* Create a Kubernetes service account that is annotated with the ARN of the IAM role that you created.

* Create a file with the following contents named `efs-service-account.yaml` and replace **111122223333** with your account ID.

```
---
apiVersion: v1
kind: ServiceAccount
metadata:
  labels:
    app.kubernetes.io/name: aws-efs-csi-driver
  name: efs-csi-controller-sa
  namespace: kube-system
  annotations:
    eks.amazonaws.com/role-arn: arn:aws:iam::111122223333:role/AmazonEKS_EFS_CSI_DriverRole
```

* Run the following command to create the storage class. The Kubernetes service account named `efs-csi-controller-sa` is annotated with the IAM role that you created named `AmazonEKS_EFS_CSI_DriverRole`.

```
kubectl apply -f efs-service-account.yaml
```

– Install the Amazon EFS driver

* There are three ways to install the driver: Helm, Manifest (private registry), or Manifest (public registry). We used the last option.



1. Download the manifest through the following command:

```
kubectl kustomize
"github.com/kubernetes-sigs/aws-efs-csi-driver/deploy/kubernetes/overlays/stable/?ref=v1.4.0" >
public-ecr-driver.yaml
```

2. Edit the file and remove the lines that are responsible for the creation of the storage class named efs-csi-controller-sa because it was created in the previous step. After that, apply the manifest:

```
kubectl apply -f public-ecr-driver.yaml
```

– Create an Amazon EFS file system

- * The Amazon EFS CSI driver supports the Amazon EFS access points, which are application-specific entry points into an Amazon EFS file system that make it easier to share a file system between multiple points.
- * Below are the different steps the you need to create an Amazon EFS file system for the previously created cluster.

1. Get the VPC (Virtual Private Cloud) ID where the cluster is in.

```
vpc_id=$(aws eks describe-cluster --name my-cluster --query "cluster.resourcesVpcConfig.vpcId"
--output text)
```

2. Get the CIDR range for the cluster's VPC.

```
cidr_range=$(aws ec2 describe-vpcs --vpc-ids $vpc_id --query "Vpcs[].CidrBlock" --output text)
```

3. Create a security group with an inbound rule that allows inbound NFS traffic for the Amazon EFS mount points.

- Create a security group.

```
security_group_id=$(aws ec2 create-security-group --group-name MyEfsSecurityGroup --description
security group" --vpc-id $vpc_id --output text)
```

- Create an inbound rule that allows inbound NFS traffic from the CIDR for the cluster's VPC.

```
aws ec2 authorize-security-group-ingress --group-id $security_group_id --protocol tcp --port
--cidr $cidr_range
```

4. Create an Amazon EFS file system for the previously created cluster.

- Create a file system.

```
file_system_id=$(aws efs create-file-system --region region-code --performance-mode generalP
--query 'FileSystemId' --output text)
```

- Create mount targets. First you need to find the ip address of the cluster nodes. This can be done through the following command:

```
kubectl get nodes
```

The output needs to be as follows:

NAME	STATUS	ROLES	AGE	VERSION
ip-192-168-56-0.region-code.compute.internal	Ready	<none>	19m	v1.19.6-eks-49a6c0

Figure 3: nodes_cluster_info

- After that, determine the ids of the subnets in the VPC and which Availability Zone the subnet is in.

```
aws ec2 describe-subnets
--filters "Name=vpc-id,Values=$vpc_id" --query 'Subnets[*].{SubnetId: SubnetId,AvailabilityZone:
AvailabilityZone,CidrBlock:CidrBlock}' --output table --region region-code
```



DescribeSubnets		
AvailabilityZone	CidrBlock	SubnetId
region-codec	192.168.128.0/19	subnet-EXAMPLE6e421a0e97
region-codeb	192.168.96.0/19	subnet-EXAMPLEd0503db0ec
region-codec	192.168.32.0/19	subnet-EXAMPLEe2ba886490
region-codeb	192.168.0.0/19	subnet-EXAMPLE123c7c5182
region-codea	192.168.160.0/19	subnet-EXAMPLE0416ce588p

Figure 4: cluster_subnets

- Finally, add mount targets for the subnets that the nodes are in. Through this [link](#) you can find out in which CidrBlock the node's ip address is in.

```
aws efs create-mount-target
--file-system-id $file_system_id
--subnet-id subnet-EXAMPLEe2ba886490
--security-groups $security_group_id --region region-code
```

- To ensure that everything was created correctly, you can deploy the sample application with dynamic provisioning presented at the end of [this guide](#).
- After that, you can create a storage class. Take what follows as an example:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: efs-sc
provisioner: efs.csi.aws.com
parameters:
  provisioningMode: efs-ap
  fileSystemId: fs-0e40f7a1635c2391d
  directoryPerms: "700"
```

Figure 5: storage_class_example

- Replace the fileSystemId value with the previously created file system.

9.1.2 Option 2. NFS

!!! note More information: <https://docs.aws.amazon.com/filegateway/latest/files3/create-gateway-file.html> and <https://docs.aws.amazon.com/filegateway/files3/create-nfs-file-share.html>

1. Create AWS Storage Gateway

1. Open the [AWS Storage Gateway console](#) and select **Create Gateway**.
2. **Gateway Settings**



- For **Gateway name**, enter a name for the gateway.
- For **Gateway time zone**, select the local time zone for the part of the world where you want to deploy the gateway.

3. Gateway Options

- For **Gateway Type**, select **Amazon S3 File Gateway**.

Storage Gateway > Gateways > Create gateway

Set up gateway [Info](#)

► **How it works** [Info](#)

Gateway settings

Gateway name

The name must be between 2 and 255 characters and cannot include a slash (/).

Gateway time zone

Choose the local time zone based on where you are deploying your gateway.

GMT +1:00 Brussels, Copenhagen, Madrid, Paris ▼

Gateway options [Info](#)

Gateway type

- Amazon S3 File Gateway**
Store and access objects in Amazon S3 from NFS or SMB file data with local caching.
- Amazon FSx File Gateway**
Access fully managed file shares in Amazon FSx for Windows File Server using SMB.
- Tape Gateway**
Store virtual tapes in Amazon S3 using iSCSI-VTL, and store archived tapes in Amazon S3 Glacier Flexible Retrieval or Amazon S3 Glacier Deep Archive.
- Volume Gateway**
Store and access iSCSI block storage volumes in Amazon S3.

Figure 6: setup_gateway

4. Platform Options

- For **Host platform**, select the platform on which the gateway is supposed to be deployed. On our side, we chose **Amazon EC2**. The gateway must be created in the same *Virtual private cloud (VPC)* as the cluster, and the selected *VPC subnet* can be one that is available in the VPC.
- After filling out all the fields with the correct data, select **Launch Instance**. This step can take up to two minutes. Select **Next** when the previous task is completed.



Platform options [Info](#)

Host platform

- VMware ESXi
- Microsoft Hyper-V
- Linux KVM
- Amazon EC2
- Hardware appliance

Launch EC2 instance
Standard Amazon EC2 instance pricing applies. [Learn more](#)

Use default settings
The default settings use an instance type of m5.xlarge, 150 GiB of cache storage, and minimally-required inbound security ports.

Customize your settings
Customize your settings using the Amazon EC2 launch instance wizard.

Virtual private cloud (VPC) network
Choose a VPC for your instance.

vpc-0b747eda1471f4195 | eksctl-cluster-cluster/VPC ▼ ↻

VPC subnet
Choose a VPC subnet for your instance.

subnet-0e7888728d6d62331 | eksctl-cluster-cluster/SubnetPublicUSWEST2A | us-west-2a ▼ ↻

Key pair
Choose an existing key pair that you have access to, or create a new key pair to securely connect to your instance.

nfs-keypair ▼ ↻

[Create new key pair](#)

Launch instance

Cancel **Next**

Figure 7: setup_gateway_platform_options

5. Connect to AWS

- For **Endpoint options** you can select **VPC hosted**, but you must have created all the inherent resources in advance. Otherwise, continue with **Publicly accessible**.
- For **Gateway connection options**, continue with the default options.



Storage Gateway > Gateways > Create gateway

Activation

Step 1
Set up gateway

Step 2
Connect to AWS

Step 3
Review and activate

Configuration

Step 4
Configure gateway

Connect to AWS Info

Endpoint options Info

Service endpoint
Choose whether the endpoint is publicly accessible or hosted inside your VPC.

Publicly accessible
Your gateway communicates with AWS over the public internet.

VPC hosted
Accessible within your Virtual Private Cloud (VPC) only. Your gateway communicates with AWS through a private connection with your VPC, allowing you to control your network settings.

FIPS enabled
Select whether the endpoint should comply with Federal Information Processing Standards (FIPS).

FIPS enabled endpoint

Gateway connection options

Connection options
You can use the gateway IP address. If that isn't available, use the activation key.

IP address
Your gateway's IP address must be public or accessible from within your current network. Your web browser must be able to connect to this IP address.

Activation key
Enter the activation key of your gateway's virtual machine (VM).

IP address
This is pre-populated with the public IPv4 address of your EC2 instance. [i-091ef4f60a9f54878 | storagegateway-wizard 0ef6a89c](#)

Cancel Previous **Next**

Figure 8: connect_to_aws

6. Review and activate

- In this step, you can edit the pre-filled settings. After proceeding, all data cannot be changed.

7. Configure gateway

- This is the last step and all configurations can be left with the default values. Select **Next** to finish the gateway configuration.



Configure gateway [Info](#)

Configure cache storage [Info](#)

Storage allocation is pre-populated using the EBS volumes of your EC2 instance [i-06d7f6a7657653acb](#) | [storagegateway-wizard 0ef6f94e](#)

Configure cache storage by allocating one or more local disks with at least 150 GiB to **Cache**. The local disks correspond to the storage that you provisioned on your host platform.

Disk ID	Capacity	Allocated to
/dev/sdb	150 GiB	Cache

CloudWatch log group [Info](#)

You can monitor the health of your gateway using Amazon CloudWatch log groups.

Choose how to set up log group
You can activate or deactivate logging at any time.

- Create a new log group**
A new CloudWatch log group will be created.
- Use an existing log group**
Choose an existing CloudWatch log group.
- Deactivate logging**
No CloudWatch log group will be created.

Pricing
Standard Amazon CloudWatch Logs pricing applies based on your usage. [Learn more](#)

Figure 9: configure_gateway

8. Now, on the **Gateway overview dashboard**, the resource is available. You may need to wait a few minutes for the status to turn green. At this point, the gateway is ready to use.

With the gateway ready, it is time to create the file shares. Besides having the gateway up and running, a S3 bucket resource needs to be created for each file share. Buckets are containers for data stored in S3.

Storage Gateway > Gateways

Gateway overview

Gateways (1) [Info](#) [Refresh](#) [Actions](#) [Create tapes](#) [Create volume](#) [Create file share](#) [Attach FSx file system](#) [Create gateway](#)

Filter by gateway name, ID, status, type, tag key, or tag value.

Name	Gateway ID	Status	Alarm state	Gateway type	Storage resources
storage-gateway	sgw-2B2CD842	Running	None	S3 File	0 file shares

Figure 10: gateway_overview

2. S3 bucket creation

1. To create this resource, go to [this dashboard](#) and select **Create bucket**. A page similar to the one below comes up. Fill out the **bucket name** and make sure that the AWS Region is the same as in the previous resources. All other configurations can be left with the default values.



Amazon S3 > Buckets > Create bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

Bucket name must be globally unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Object Ownership [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

Figure 11: create_bucket

2. If all goes well, the bucket is created and ready to use.

Buckets (1) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

Name	AWS Region	Access	Creation date
<input type="radio"/> s3-bucket-tutorial-eks	US West (Oregon) us-west-2	Bucket and objects not public	October 18, 2022, 15:32:03 (UTC+01:00)

Figure 12: buckets_dashboard

3. File share creation

1. File share settings

- For **Gateway**, select the previously created gateway.
- For **Amazon S3 location**, select **S3 bucket name** and fill out the corresponding field with its name.



Storage Gateway > File shares > Create file share

Step 1
File share settings

Step 2
Amazon S3 storage settings

Step 3
File access settings

Step 4
Review and create

File share settings [Info](#)

File share settings configuration

Gateway
storage-gateway (sgw-2B2CD842) ▼

Amazon S3 location [Info](#)

S3 bucket name
Connect directly to your bucket.

S3 access point
Connect through a named network endpoint that is attached to your bucket.

S3 access point alias
Connect using a bucket-style alias that points to an S3 access point connected to your bucket.

Amazon S3 bucket name / S3 prefix name - optional

Bucket name must be between 3 to 63 characters. Prefix name must end with a "/".

Versioning-enabled S3 buckets
Use of S3 buckets with versioning enabled can increase the amount of storage used in Amazon S3. Each modification to a file creates a new version of the object.

AWS region
Choose the region where the S3 endpoint is located
 ▼

Figure 13: file_share_settings

- By default, the file share name is prefilled with the S3 bucket name, but it can be changed to a different one.
- For now, leave the other configurations with the default values and select **Next**.



File share name [Info](#)

The default name is pre-filled with the S3 bucket name. Once the file share is created, the file share name can't be deleted.

File share name must be between 1 to 255 characters.

Private Link for S3 - Optional

Use VPC endpoint for S3

Access objects using

- Network File System (NFS)
 Server Message Block (SMB)

Audit logs [Info](#)

You can monitor your fileshare using Amazon CloudWatch log groups.

- Create a new log group
A new CloudWatch log group will be created.
- Use an existing log group
Choose an existing CloudWatch log group.
- Deactivate logging
No CloudWatch log group will be created.

Automated cache refresh from S3 [Info](#)

- None
 Set refresh interval

File upload notification [Info](#)

- None
 Set settling time

Figure 14: file_share_settings_2

2. Amazon S3 storage settings

- Keep the configurations with the default values and proceed.



Amazon S3 storage settings [Info](#)

Amazon S3 storage configuration

Amazon S3 bucket name / Prefix name
[s3-bucket-tutorial-eks](#) [🔗](#)

Storage class for new objects
S3 Standard ▼

Object metadata [Info](#)

- Guess MIME type
- Give bucket owner full control
- Enable requester pays

Access to your S3 bucket [Info](#)

- Create a new IAM role.
- Use an existing IAM role

Encryption [Info](#)

- S3-Managed Keys (SSE-S3)
- KMS-Managed Keys (SSE-KMS)

Cancel

Previous

Next

Figure 15: amazon_s3_storage_settings

3. File access settings


- Keep the configurations with the default values and proceed.



File access settings [Info](#)

Access object

Allowed clients is a list of clients that are allowed to access the file gateway.

 This file share will accept connections from any NFS clients. Add one or more clients to restrict access.

No clients associated with the resource.

[Add client](#)

Mount options [Info](#)

Squash level

Root squash (default) ▼

Export as

- Read-write
- Read-only

Figure 16: file_access_settings



File metadata defaults [Info](#)

Objects in Amazon S3 which were not created or modified by your gateway will have the following metadata defaults. These defaults include Unix permissions for files and folders.

ⓘ The values provided are the default file access settings. You can also edit the default metadata for your file share later.

Directory permissions

Directory permissions must be between 1 and 4 characters.

File permissions

File permissions must be between 1 and 4 characters.

User ID

User ID must have a value between 0 and 4294967294.

Group ID

Group ID must have a value between 0 and 4294967294.

[Cancel](#) [Previous](#) [Next](#)

Figure 17: file_access_settings_2

4. Review and create

- Check that all settings are correct and proceed with creating the file share.
5. File share is ready to use when its status changes to **Available**. It may take a few minutes between creation and this status update.

File share overview

File shares (1) [Info](#) 🔄 Actions ▾ Create file share

🔍 share-7C971518 ✕ 1 match ⚙️

<input type="checkbox"/>	File share ID ▲	File share name ▼	Status ▼	Type ▼	S3 location 🔗	Gateway ▼
<input type="checkbox"/>	share-7C971518	s3-bucket-tutorial-eks	Available	NFS	s3-bucket-tutorial-eks 🔗	storage-gateway

Figure 18: file_share_overview

9.2 EKS and Database Shared Storage

The EKS cluster and the SQL Server must have a shared storage space that allows both to read and write data, ensuring seamless access and interaction between them.



10 Ingress

This section describes how to manage the incoming ingress traffic onto the EKS cluster at the Ingress Controller level, including its exposure, TLS and how to route to different environments.

10.1 Ingress Controller

An Ingress Controller is responsible to handle traffic coming from outside the Kubernetes cluster, acting as a reverse proxy and routing the requests to the backend services as defined in Ingress specifications.

Critical Manufacturing MES includes a Traefik instance that acts as an internal reverse proxy, routing the traffic to the namespace services, and an Ingress definition to be consumed by the controller. These Ingresses define a single HTTPS rule that matches a particular host, indicating that all traffic that gets matched is to be delivered to the Traefik service which takes care of the rest.

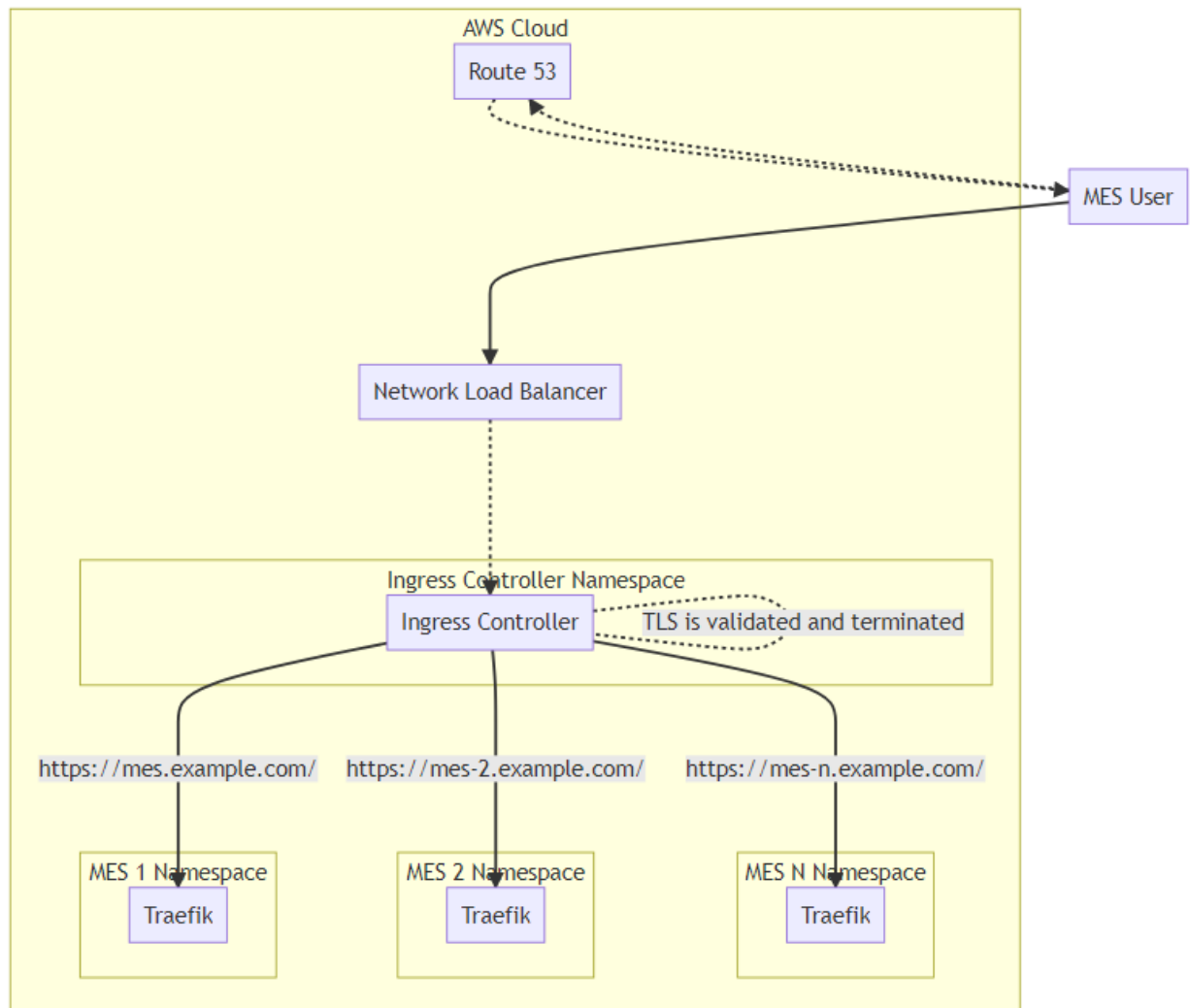
Having this in mind, the Ingress Controller must read Ingresses from different namespaces (in order to serve more than one MES environment), which requires a set of RBAC permissions (these vary depending on the Ingress Controller of choice). Currently, the [Critical Manufacturing Infrastructure Agent](#) includes a Traefik instance that is configured to run as the cluster's Ingress Controller with all the necessary requirements:

1. Two endpoints for incoming traffic, HTTP and HTTPS (HTTP traffic is permanently redirected to the HTTPS endpoint)
2. RBAC definitions allowing it to read and manage Ingresses from all namespaces
3. Matches all Ingresses with the label `app: traefik`
4. Only allows HTTPS and deals with TLS validation and termination

!!! warning "Deprecation warning" The **Infrastructure Agent's Traefik is deprecated** starting from Infrastructure Agent 11.1, and will be removed when MES 12.0 is released.

!!! note If using a custom Ingress Controller, these requirements must be met.

The following diagram illustrates how the traffic is managed:



10.2 Ports

The Ingress Controller needs to expose two ports:

1. 80 - Handle HTTP traffic. Security-wise it should permanently redirect all traffic to the HTTPS endpoint.
2. 443 - Handle HTTPS traffic.

10.3 TLS

All incoming traffic that hits the HTTPS endpoint is validated and TLS is terminated before being forwarded to the backend service. As such, TLS configurations must be put into place:

- Using a TLS secret with a certificate that matches different subdomains through SANs or using a wildcard.
- Using external services to handle certificate management, such as Let's Encrypt with Route 53.

!!! note If using the Infrastructure Agent's Traefik, you'll have the possibility to reference the name of the secrets that contain the Certificate and Key, or include configures for external providers such as Route 53 through Let's Encrypt.



!!! note Defining different certificates at the Ingress level (per MES) is not supported.

10.4 Load Balancer

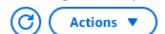
Having the ingress traffic reaching the Ingress Controller requires a Load Balancer that respects the [Ports](#) requirements. Since we are dealing with Layer 7 (Application) in the Ingress Controller, we require a Layer 4 (Transport) Load Balancer - a Network Load Balancer (NLB). As the [AWS documentation](#) refers, it is recommended to use the [AWS Load Balancer Controller](#). This controller will automatically manage the NLB during its whole lifecycle, including the creation and destruction.

Having the AWS Load Balancer Controller, you should add the following annotations to the Load Balancer Kubernetes Service that exposes the Ingress Controller:

- service.beta.kubernetes.io/aws-load-balancer-scheme - The scheme of the Load Balancer. The value can be **internet-facing** or **internal**, depending if you want to route requests from over the internet or just with private IP addresses.
- service.beta.kubernetes.io/aws-load-balancer-type - Type of the Load Balancer which should be set to **external**. This will define, by default, that all traffic is routed to the port of the service.

After having these changes, AWS should automatically create an NLB for you with an A Record, similar to the following example:

k8s-ingressc-traefik-3d1f325084



▼ **Details**

Load balancer type Network	Status Active	VPC [redacted]	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone [redacted]	Availability Zones 1b (euw1-az2) [redacted] eu-west-1 1a (euw1-az1) [redacted] eu-west-1	Date created [redacted]
Load balancer ARN arn:aws:elasticloadbalancing:eu-west-1:[redacted]:loadbalancer/net/k8s-ingressc-traefik-3d1f325084/[redacted]		DNS name Info k8s-ingressc-traefik-3d1f325084-[redacted].elb.eu-west-1.amazonaws.com (A Record)	

!!! note The ARN, Zone, VPC, Region, Availability Zones and DNS name will vary.

The A Record resolves to the NLB which will, in turn, route all the traffic to the Ingress Controller. Now, since the latter operates at Layer 7, and to be able to support other domains, you should configure a DNS provider to resolve a particular DNS name to this A record. You can leverage Route53 by adding a CNAME record to your domain. Taking the *mes.example.com* as an example:

1. Go to Route53
2. Choose the Hosted Zone where your top domain (example.com) resides
3. Create a Record:
 - Name: the name of the subdomain (mes)
 - Type: A Record
 - Target: Alias to Network Load Balancer and choose your value

For more information, see <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-to-elb-load-balancer.html>.



11 Encryption Data Configuration

Amazon EFS provides encryption for both data in transit and data at rest, ensuring that the data remains secure when stored and while being accessed. Amazon S3 Gateway can be used to mount volumes that map S3 buckets, while encryption is handled by S3 and through the network protocol.

!!! info More information about the two above mentioned methods can be found [here](#).

11.1 Encryption in transit

Protects the data while it is being moved from one place to another. This ensures that it can't be intercepted or read while it's being sent over a network.

11.1.1 Features

- It acts when the data is moved between two systems, such as a computer and a cloud service or between services within a cloud.
 - The data is encrypted during transmission using protocols like TLS, which ensures that no one can intercept and read the data while it's being sent over a network.
 - The main goal of this process is to protect data from being intercepted while it is moving across a network.
 - It is supported using TLS to ensure that the communication between the client (Kubernetes) and EFS is encrypted.
 - !!! info To learn more about this topic, go to the [Encrypting data in transit](#) page in the Amazon Elastic File System User Guide.

11.1.2 Volumes

- When using S3 Gateway, the data in transit between the application and the gateway may not be encrypted by default over NFS.
- However, once the data is transferred from the gateway to Amazon S3, it is encrypted using HTTPS that ensures that data is secure during the process between the gateway and the S3 bucket.

11.2 Encryption at rest

Protects the data once it reaches its destination and is stored on a disk, database, or in the cloud. This ensures that if someone gains unauthorized access to the physical storage, they won't be able to read the data without the decryption keys.

11.2.1 Features

- Amazon EFS encrypts the data at rest using AWS Key Management Service.
- The data and metadata are encrypted with the AES-256 encryption algorithm.
- It is possible to choose between AWS-managed key (default) or a customer-managed key in AWS KMS for greater control over the encryption.
- This method is enabled when the EFS file system is created, and after that, all data written to the file system will be encrypted.
 - !!! info To learn more about this topic, go to the [Encrypting data at rest](#) page in the Amazon Elastic File System User Guide.



11.2.2 Volumes

- Data stored in Amazon S3 is encrypted at rest using Server-Side Encryption (SSE). There are three main options:
 - **SSE-S3** (Server-Side Encryption with S3-Managed Keys):
 - * S3 automatically encrypts your data using AES-256 (Advanced Encryption Standard) without requiring user involvement.
 - * AWS manages both the encryption and decryption processes seamlessly. Suitable for use cases where simplicity and minimal configuration are required.
 - **SSE-KMS** (Server-side Encryption with Customer-Provided Keys)
 - * Data is encrypted using keys managed by AWS KMS.
 - * The user can control and audit the usage of the encryption keys.
 - * It provides additional security features like fine-grained permissions and key rotation.
 - **SSE-C** (Server-Side Encryption with Customer-Provided Keys):
 - * The user provides the encryption keys for S3 to use when encrypting and decrypting data.
 - * S3 does not store the encryption key, so it must be provided with each request.



12 External Services

This section contains information regarding services external to Critical Manufacturing MES that are dependencies and required for the MES system to run.

- [\[\[installation-guide-aws-external-sqlserver\]\]](#)
- [\[\[installation-guide-aws-external-kafka\]\]](#)
- [\[\[installation-guide-aws-external-rabbitmq\]\]](#)
- [\[\[installation-guide-aws-external-s3\]\]](#)
- [\[\[installation-guide-aws-external-clickhouse\]\]](#)



13 SQL Server

As described in [[system-requirements-persistence-layer-index]] and [[installation-guide-database-servers-index]], an SQL Server database is required for Critical Manufacturing to run.

Depending on your infrastructure configuration, the SQL Server database instances may or may not be part of the AWS deployment. There is no requirement on where the database servers are deployed and running, as long as there is connectivity with the EKS cluster.

The following sections give some guides on how to install SQL Server in AWS if you want to have the database running in the AWS infrastructure as well.

13.1 EC2 Instances

One way to install SQL Server in AWS is using EC2 instances. Amazon Elastic Compute Cloud is a service that allows to create and run virtual machines with resizable capacity.

Using EC2, you will create **Windows Server EC2** instances following the software and hardware requirements documented in [[system-requirements-persistence-layer-index]]. When the instances are running, SQL Server needs to be installed and configured on these instances according to the documentation that you can find in [[installation-guide-database-servers-index]].

Keep in mind that, similarly to local on-premises servers, these EC2 instances need to be managed, updated, and maintained manually.

13.2 AWS RDS Custom

An alternative way to have your databases running on AWS infrastructure is using AWS **RDS Custom for SQL Server**. Managed database services for applications that require operating system and database customization. The idea behind RDS Custom is to allow the users to initially setup the machine where the database is running, giving access to the underlying EC2 instance to perform specific configurations or installing any required additional software. After this, RDS Custom service will monitor and manage the SQL Server database inside this EC2 instance.

For more information, see <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/rds-custom.html>.

!!! warning Presently, Amazon RDS databases **are not supported** by Critical Manufacturing.



14 Kafka

Managing Kafka clusters manually can be complex, especially when considering factors like scalability, monitoring, and security. Fortunately, it is possible to access several managed Kafka solutions via AWS Marketplace. The two primary options are Amazon MSK and Confluent Cloud.

14.1 Amazon Managed Streaming for Apache Kafka - Amazon MSK

Amazon Managed Streaming for Apache Kafka (MSK) is a fully managed service that simplifies deploying, managing, and scaling Apache Kafka clusters on AWS. It allows to use Apache Kafka without handling the operational complexity of running it manually.

14.2 Confluent Cloud

Confluent Cloud is a managed Kafka service that extends the Apache Kafka with advanced features and tools. Built by the Kafka creators, it provides additional capabilities on top of Kafka that helps to build robust data streaming pipelines.

This solution can be subscribed through AWS Marketplace. With this, the setup is simplified, as the marketplace allows to subscribe and deploy the Kafka cluster in a easy way, it offers flexible pricing models, the integration is seamless with other AWS services.



15 RabbitMQ

RabbitMQ is a reliable and mature messaging and streaming broker, which is easy to deploy of cloud environments, on-premises, and on local machines.

15.1 Amazon MQ

AWS offers a solution for RabbitMQ called **Amazon MQ**. It is a managed message broker service that supports message brokers such as RabbitMQ and Apache ActiveMQ. It allows developers to use RabbitMQ's robust messaging features without worrying about the complexity of broker management, scaling, or infrastructure maintenance.

There are many key features of Amazon MQ. Below, some of them are highlighted:

- **Fully Managed** - it takes care of the operational overhead, including provisioning the message broker, ensuring high availability, patching, monitoring, and failure recovery.
- **High Availability** - it deploys RabbitMQ brokers across multiple availability zones (AZs) in a region, providing built-in redundancy and automatic failover for improved reliability.
- **Security** - it integrates with AWS Identity and Access Management (IAM) and supports encryption in transit and at rest using TLS and AWS Key Management Service, ensuring secure messaging.
- **Monitoring and Metrics** - Integrated with Amazon CloudWatch, it provides metrics and logs to monitor broker performance and health.
- **Cost-Effective** - *Pay-As-You-Go* approach.

It is possible to find more information about this topic and how to setup it in AWS [here](#).



16 S3

16.1 Amazon S3

Amazon S3 is a scalable object storage service offered by AWS (Amazon Web Services). It allows to store and retrieve large amounts of data at any time. S3 is designed for high durability, availability, and scalability, making it suitable for a variety of use cases, including data backups, content storage and distribution, big data analytics, and more.

It is widely used for both small applications and enterprise-level storage solutions due to its flexibility and ease of use.



17 ClickHouse

ClickHouse is a high-performance, column-oriented SQL database management system (DBMS) for online analytical processing (OLAP). OLAP scenarios require real-time responses on top of large datasets for complex analytical queries.

17.1 ClickHouse - AWS Marketplace

Through AWS Marketplace, it is possible to deploy ClickHouse as a fully managed service, eliminating the operational complexity of running and maintaining the ClickHouse clusters. Managed service providers handle tasks like infrastructure setup, scaling, backup, and security.

There are different benefits in subscribing ClickHouse through the Marketplace. Firstly, it is possible to get a fully managed ClickHouse deployment with automated maintenance, backups, and scaling, reducing operational overhead. Moreover, this makes it easier to integrate with other AWS Services like S3 storage. Finally, as other AWS Services, it has the *Pay-as-You-Go* pricing so the user only pays for the resources used.

Thus, ClickHouse on AWS Marketplace is a robust solution for the users who need fast, scalable analytics without the overhead of managing their own infrastructure. It is possible to find more information about this topic [here](#).



18 Accounts and Security

This guide will walk you through the process of planning and preparing the security objects required by Critical Manufacturing MES.

18.1 Critical Manufacturing Windows Services Account

All Critical Manufacturing services will be created to run under an account that is configured in deployment time in the installation wizard. To better understand Service User Accounts, please refer to this [section](#) on Microsoft documentation.

As a reminder, please make sure that your service user account:

- Has been granted the **Log on as service** permission in the host computer
- Has permissions to access the network shares and the deployment folder
- If Remote Shipping is able to read/write the queues created for Remote Shipping
- The password never expires or there is a company mechanism to renew it before it expires

18.2 SQL Server Accounts

If the database system was deployed in Always On on Availability Groups it is fundamental to run all instances of the same component (eg. Database Engine) under the same account. Additionally, unless there is a critical security requirement forcing to do otherwise, it is recommend to use the same account for all the SQL Server Components:

- Microsoft SQL Server User Account
- Microsoft SQL Server Analysis Service User Account
- Microsoft SQL Server Reporting Services User Account

If the account hosting Reporting Services is not the same as the one hosting Critical Manufacturing services the Critical Manufacturing services user must be granted administration privileges in Reporting services.

18.3 ClickHouse

To communicate with ClickHouse, Critical Manufacturing MES uses traditional username and password authentication. In order to operate correctly, the user must have permissions to create and alter databases related to MES (including but not limited to the default ClickHouse database), which typically have the **SystemName** prefix, followed by a pertinent suffix (e.g. **SystemNameCDM**). Permissions must also include create, alter, and delete tables, as well as CRUD operations (create, read, update and delete) on table records.

It is also vital that the ClickHouse system can access the Kafka IP and MES IP addresses.

18.4 Kafka

To communicate with Kafka, Critical Manufacturing MES provides two forms of authentication:

- Mutual TLS (client certificates)
- SASL Plain (username and password)

In terms of access, the user must be granted the following permissions:



- Topic Permissions:
 - Alter
 - AlterConfigs
 - Create
 - Delete
 - Describe
 - DescribeConfigs
 - Read
 - Write
- Consumer Group Permissions:
 - Read
 - Delete
 - Describe
- Cluster Permissions:
 - AlterConfigs
 - Create
 - Describe
 - DescribeConfigs

!!! note These permissions must be granted to resources with the prefix **SystemName** and ****_SystemName****.

18.5 RabbitMQ

No additional account or security requirements are required for RabbitMQ installation.

18.6 S3

No additional account or security requirements are required for S3 installation.



19 Database Servers

This guide will walk you through the steps to successfully plan and install a database server to host Critical Manufacturing MES databases.

19.1 Planning for disaster recovery and high-availability

Critical Manufacturing recommends adopting the Always On for Availability Groups disaster recovery and high-availability solution for database systems hosting Critical Manufacturing databases. If you are not familiar with the process, the [Always On Guide](#) contains technical information on the subject. This guide is intended to serve as a summary and does not cover the technical details included in the above-mentioned guide.

An individual cluster for online, for ODS and for DWH to maximize the availability should be created, though the solution will also work if you decide to create a single cluster. In both cases, it is necessary to create an availability group for each of the product databases. This process is detailed below.

19.2 Database Server pre-requisites

The following software requirements must be met in all database nodes:

Table 11: Database Server pre-requisites

Name	Requisite	Support Versions	Checked
Operating System	Microsoft Windows Server	2016 or above	
Microsoft Windows Server Roles and Features Configuration	Microsoft Distributed Transaction Coordinator Configuration Microsoft Advanced Firewall Configuration Windows Server Failover Clustering 1		
Microsoft SQL Server	Microsoft SQL Server Database Engine Microsoft SQL Server Reporting Services Microsoft SQL Server Analysis Services	2019 or 2022	
Microsoft OLE DB Driver for SQL Server	Microsoft OLE DB Driver for SQL Server 2		

!!! note **1** Failover clustering is only necessary when building an Always On for Availability Groups solution.

!!! note **2** Only required if the Analysis Services and SQL Server are installed on different machines. In that case, please make sure to install Microsoft OLE DB Driver for SQL Server on the Analysis Services machine. More information on the installation of Microsoft OLE DB Driver for SQL Server here: <https://learn.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-server-ver16>.



19.3 SQL Server Licensing

This section contains licensing information regarding SQL Server 2019 and SQL Server 2022.

SQL Server 2019 and SQL Server 2022 Standard editions are licensed in Core-Based and Server + CAL modes, whereas SQL Server 2019 and SQL Server 2022 Enterprise editions are licensed only in Core-Based mode.

When running Critical Manufacturing on SQL Server Standard edition, the following limitations apply:

- Database instance maximum hardware consists of 128 GB or the lesser of 4 CPUs or 24 Cores (SQL Server 2019 and SQL Server 2022).
- Mobile reports and KPIs are not available.
- Power BI Report Server is not available.

For more information about SQL Server licensing, see the URLs below:

- [SQL Server 2019](#)
- [SQL Server 2022](#)

19.4 Always On for Availability Groups pre-requisites

If you have decided to implement an high-available SQL Server solution using Always On for Availability Groups, these additional requirements and preparation steps must be met:

Table 12: Always On Availability Groups pre-requisites

Task	Notes	Checked
Define the Availability Groups	We recommend adopting the following naming convention to name the Availability Groups and to use the same names for the Availability Group Listeners. At this time, you should have chosen the name to give to this Critical Manufacturing installation, referred to as SystemName AG<SystemName> - the AG that will host the online database AG<SystemName>ODS - the AG that will host the ODS database AG<SystemName>DWH - the AG that will host the DWH database	
Install WSFC	The Failover Cluster role must be enabled in all database nodes	
Enable Always On for Availability Groups in SQL Server		

19.5 Preparing Windows Server Failover Cluster

To prepare the WSFC cluster, follow the instructions on this [guide](#). Because we use Always On for Availability Groups, we do not require any kind of storage resource in the cluster. You can also choose the quorum solution that best fits your environment.

19.6 Microsoft Distributed Transaction Coordinator (MS-DTC) Configuration

This section describes how to install and configure Microsoft Distributed Coordinator (MS-DTC). MS-DTC is a component that is used to managed distributed transactions and the two-phase commit protocol. Microsoft Distributed Coordinator (MS-DTC) is installed as part of the operating system installation but it needs to be configured in order for Critical Manufacturing to work correctly.

!!! note Microsoft IIS needs to be installed in all Critical Manufacturing Application and Database servers.

The instructions for configuring Microsoft Distributed Coordinator (MS-DTC) according to Critical Manufacturing requirements are the same for Windows Server 2008 and for Windows Server 2012 and can be found in the URL: <http://technet.microsoft.com/en-us/library/cc731495.aspx>. For convenience, the configuration procedure is described in this section.

- Start the **Component Services** management console by typing **comexp.msc** in the start command and pressing **OK**.

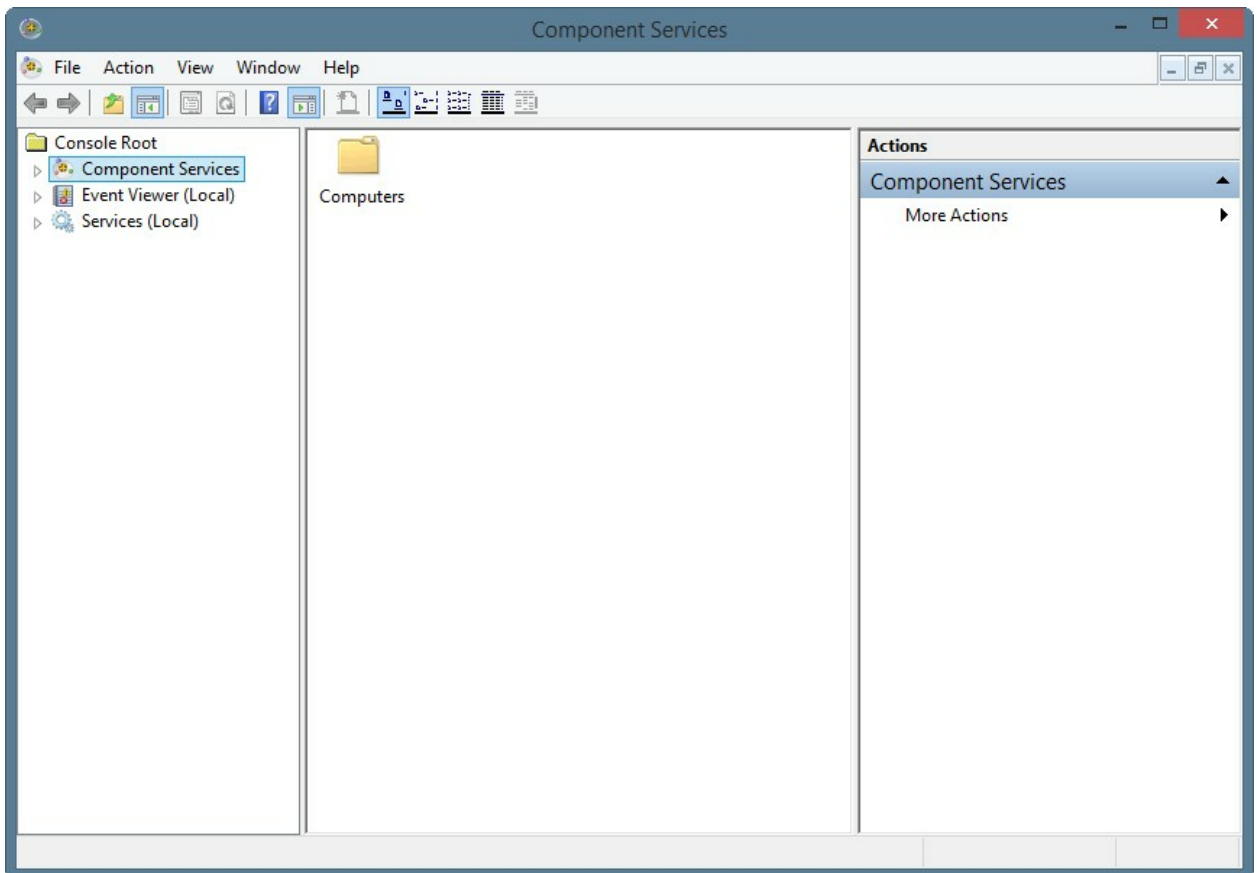


Figure 19: Component Services

- Drill-down by clicking on **Component Services** > **Computer** > **My Computer** > **Distributed Transaction Coordinator** > **Local PC**.
- Right-click on the **Local PC** under **Distributed Transaction Coordinator** and then select **Properties**.
- Click on the **Security** tab and then make sure that the following options are checked as shown in the picture below:
 - Network DTC Access
 - Allow Remote Clients
 - Allow Remote Administration
 - Allow Inbound

- Allow Outbound
- Enable XA Transactions
- Enable SNA LU 6.2 Transactions

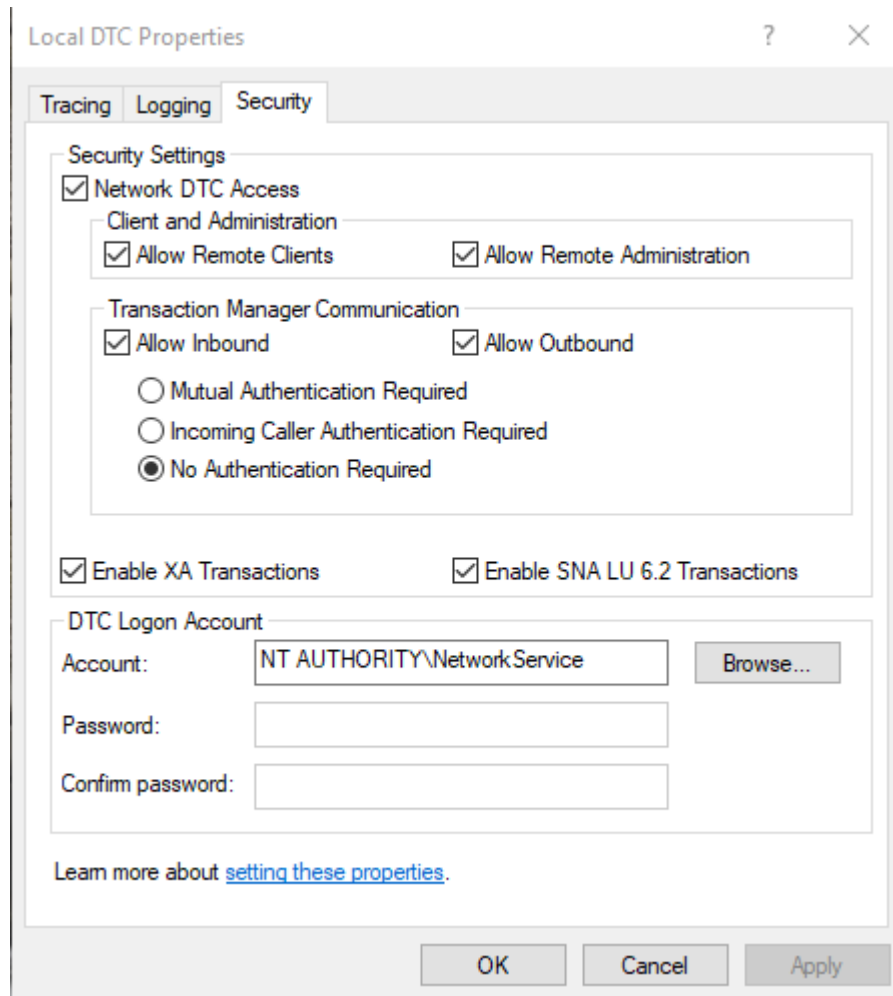


Figure 20: Local DTC Properties

19.7 SQL Server Installation

To implement Always On for Availability Groups, first you need to perform a **single node** installation of SQL Server on **each of the nodes** of the cluster. This procedure can also be used to setup a development environment where a high-availability solution is not required.

Table 13: SQL Server pre-requisites

Feature	Description	Remark
SQL Server Engine Services	The core SQL Server engine.	Mandatory for all database servers.
Analysis Services	The analysis services engine used by the data warehouse and necessary for data mining.	Mandatory for the Operational Data Store and Data Warehouse database servers.



Feature	Description	Remark
Reporting Services	The SQL Server engine used for reporting.	Mandatory for the Operational Data Store and Data Warehouse database servers.

!!! warning Reporting Services must be installed separately from this installation.

The next picture shows the *Feature Selection* screen for **SQL Server**. It represents the features required by Critical Manufacturing.

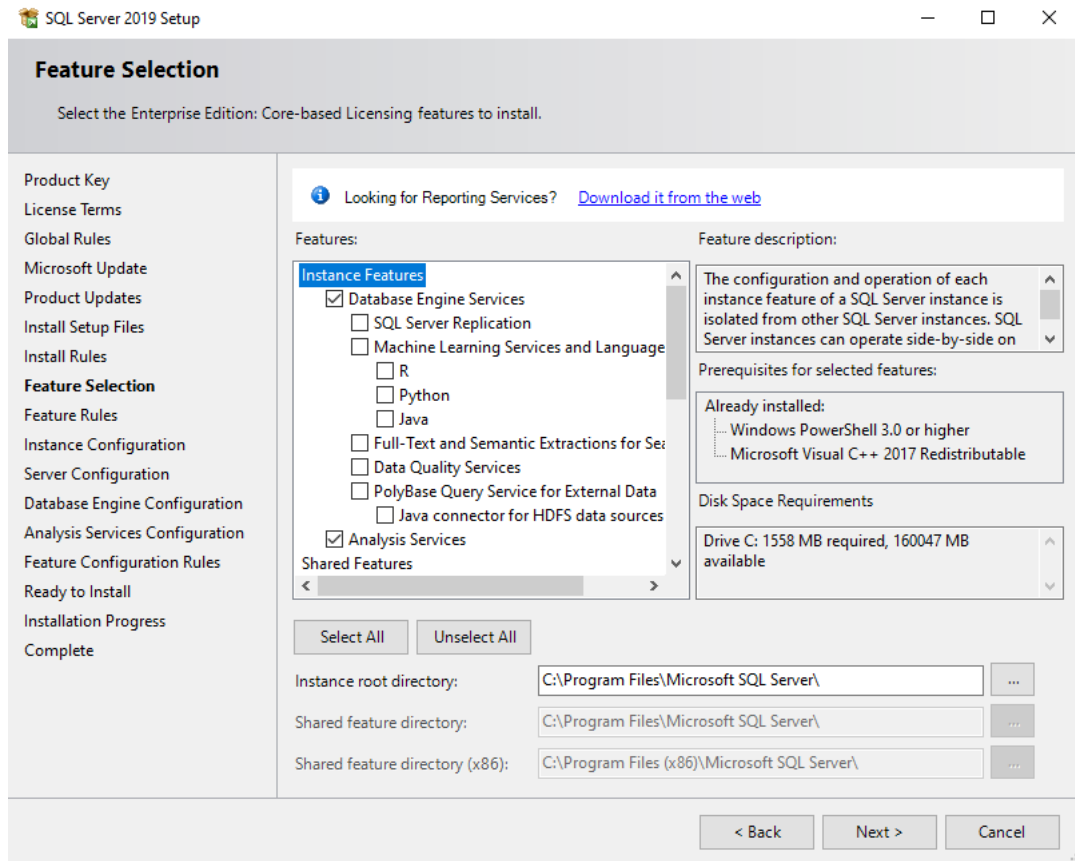


Figure 21: SQL Server Installation - Feature Selection

Next set the name of the instance to install. Critical Manufacturing recommends deploying three instances of SQL Server (Online, ODS and DWH) ideally on different physical machines. In that sense this procedure should be repeated for each of the instances.

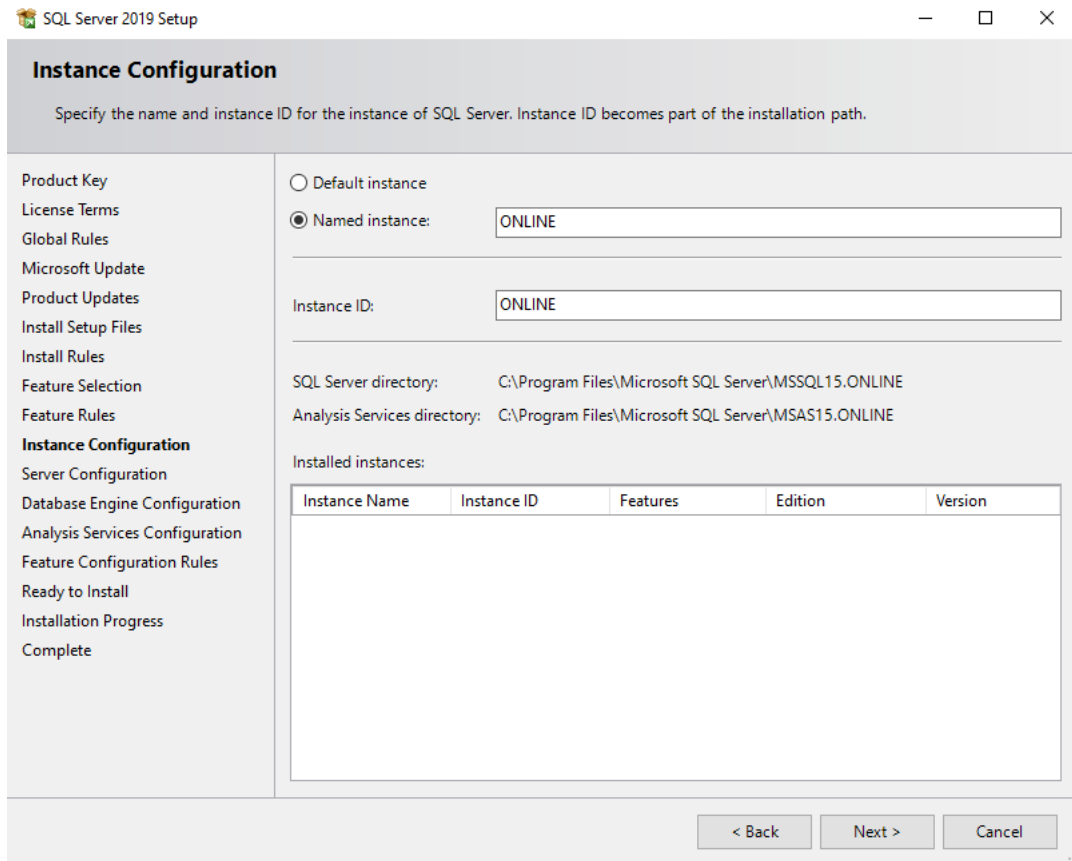


Figure 22: SQL Server Installation - Name

Next set the service account to run the engine service and the agent service. This account should be the domain account, which requires rights to access the file system.

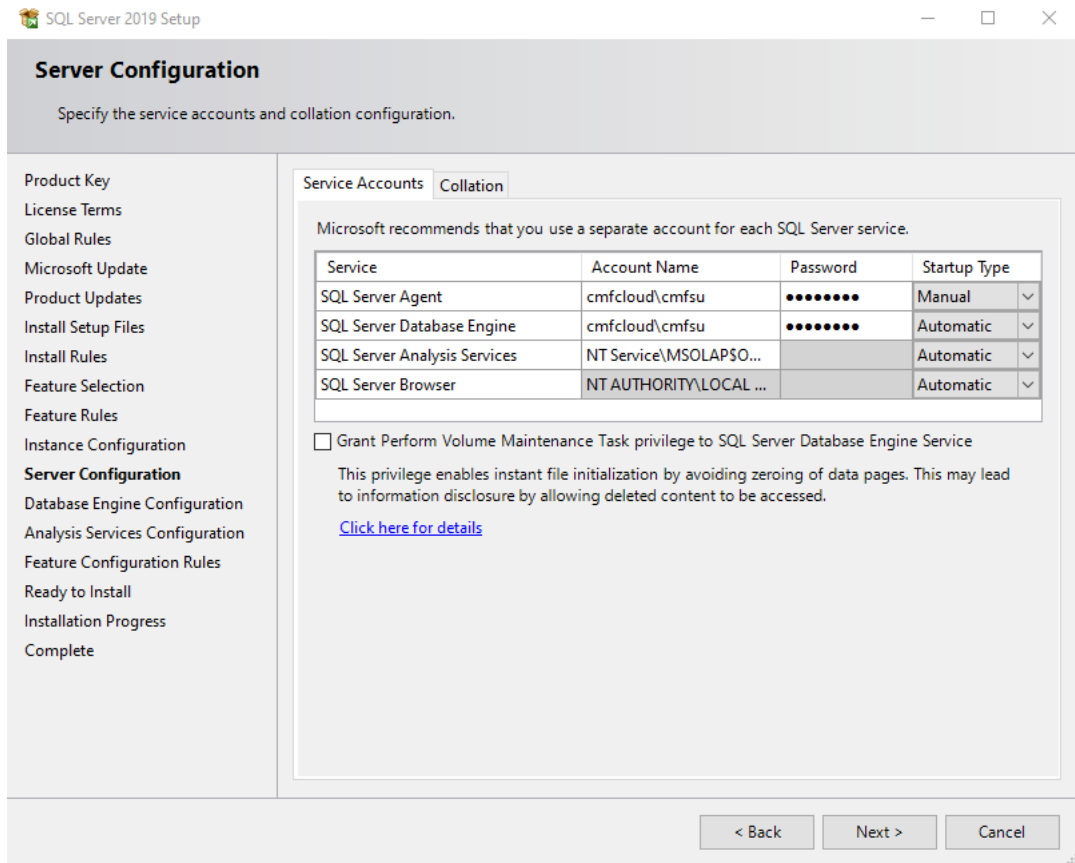


Figure 23: SQL Server Installation - Service Account

It is necessary to check that the collation is exactly **Latin1_General_CI_AS**. The deployment software will verify this setting and **will fail if it differs**.

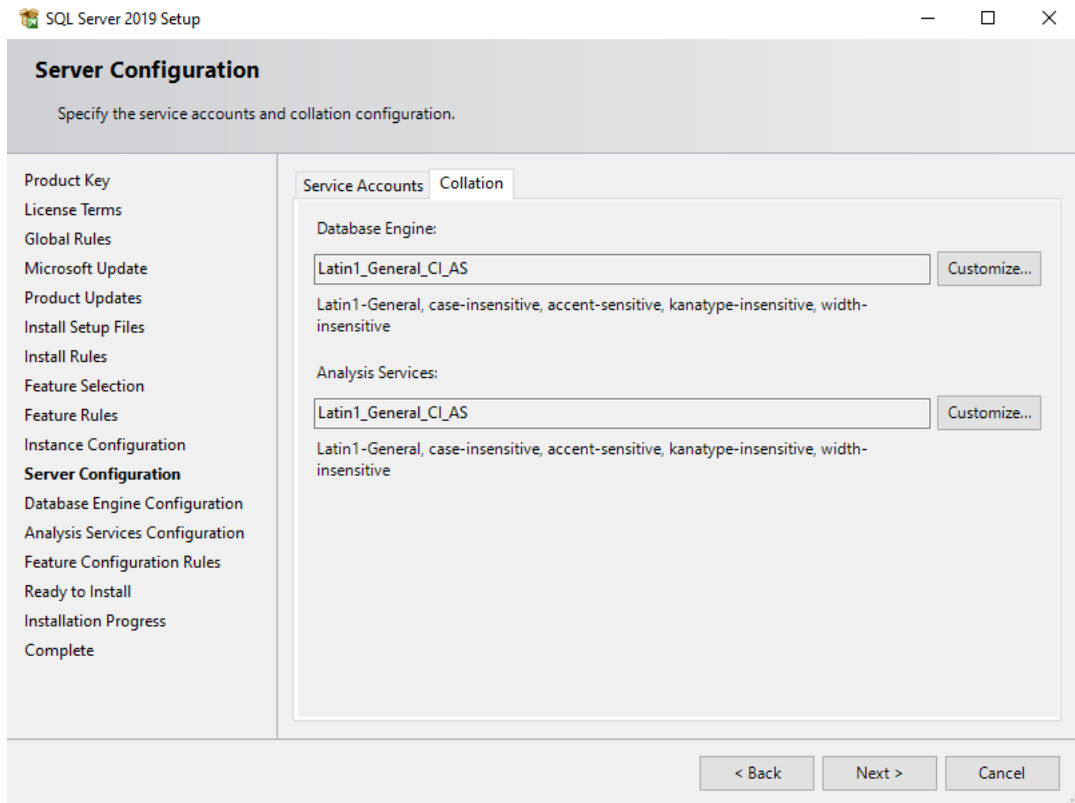


Figure 24: SQL Server Installation - Collation

If the values do not match choose the "Customize..." action and configure it as shown here:

Customize the SQL Server 2019 Database Engine Collation

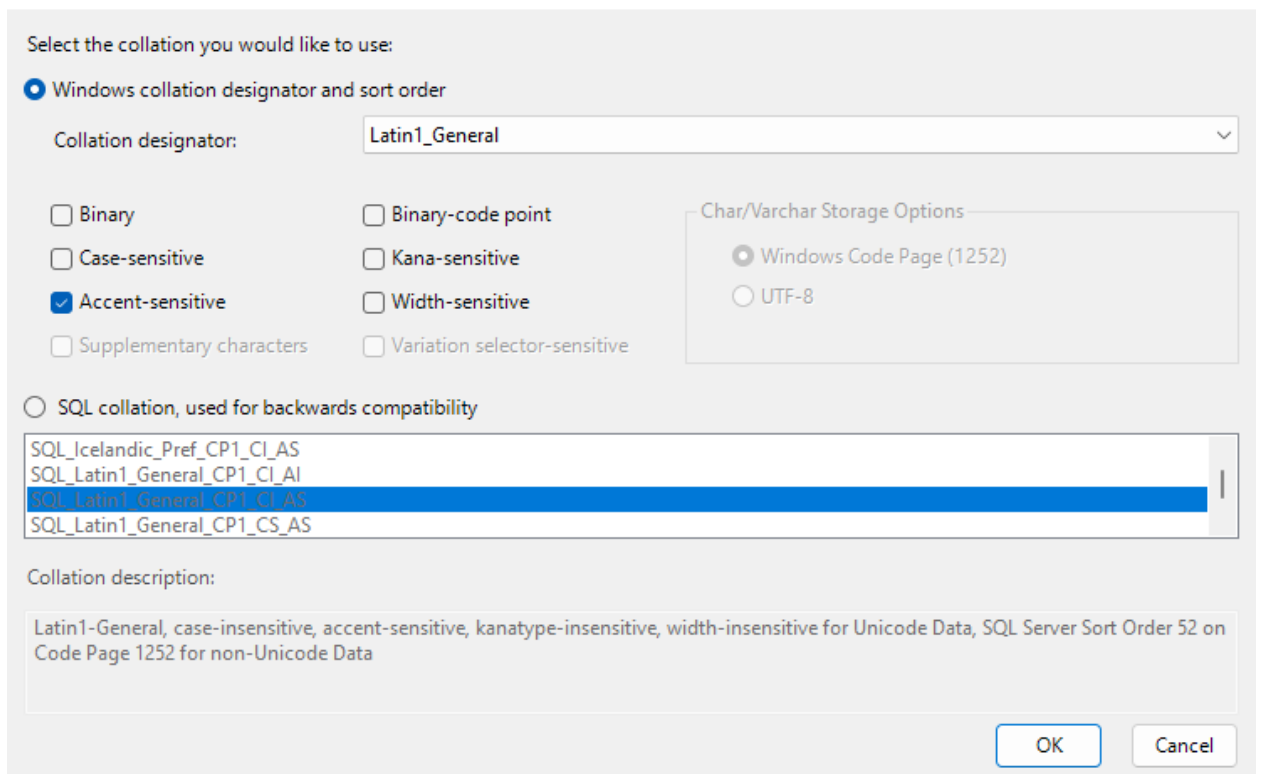


Figure 25: SQL Server Installation - Customization



Now configure the database engine. Set the authentication mode to “Mixed mode” and set the sa account password to a value of your choice.

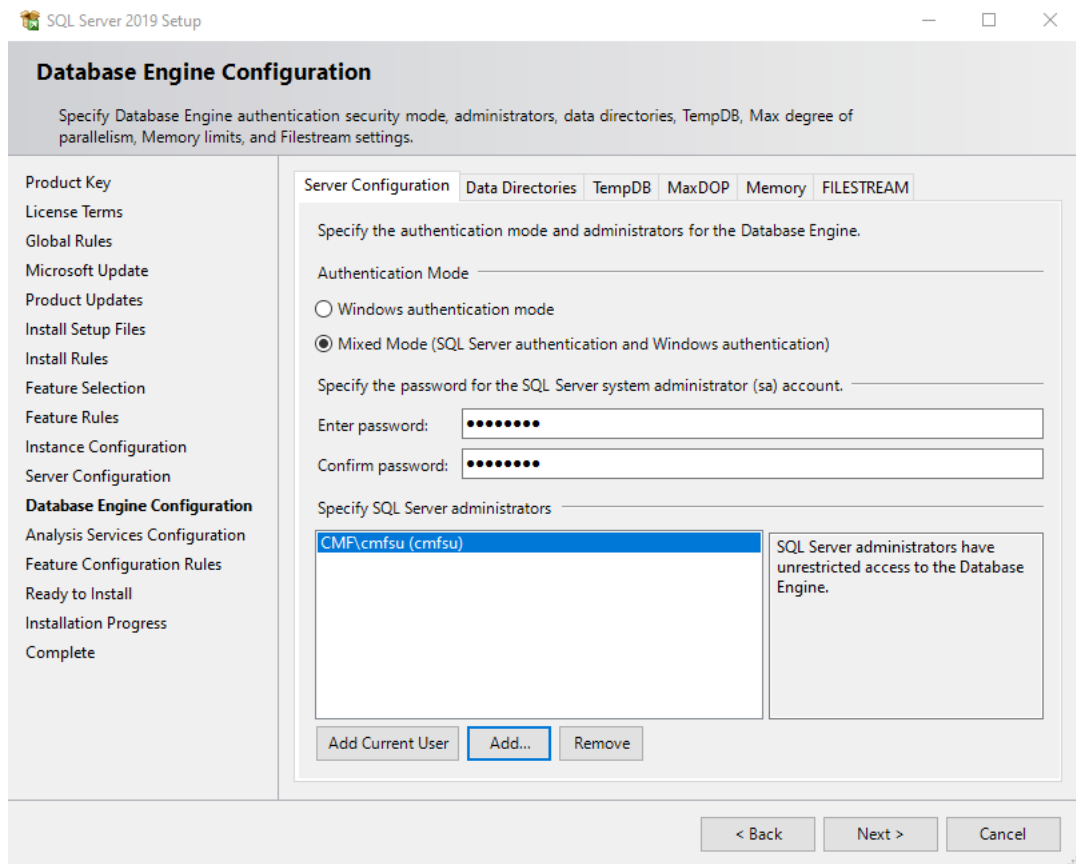


Figure 26: SQL Server Installation - Database Engine

Now configure the data directories. Critical Manufacturing recommends separating the data files and at this moment you should have already prepared the disk drives on all nodes.

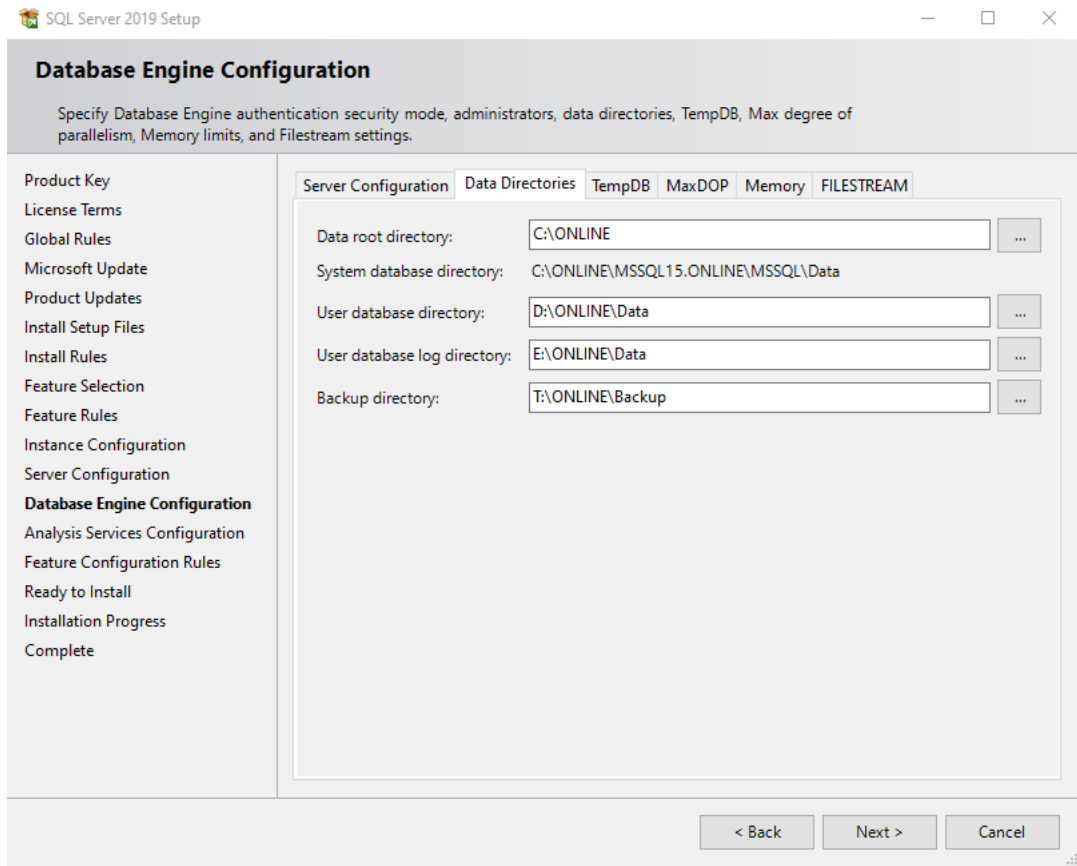


Figure 27: SQL Server Installation - Data Directories

The same recommendation is applicable to the TempDB database.

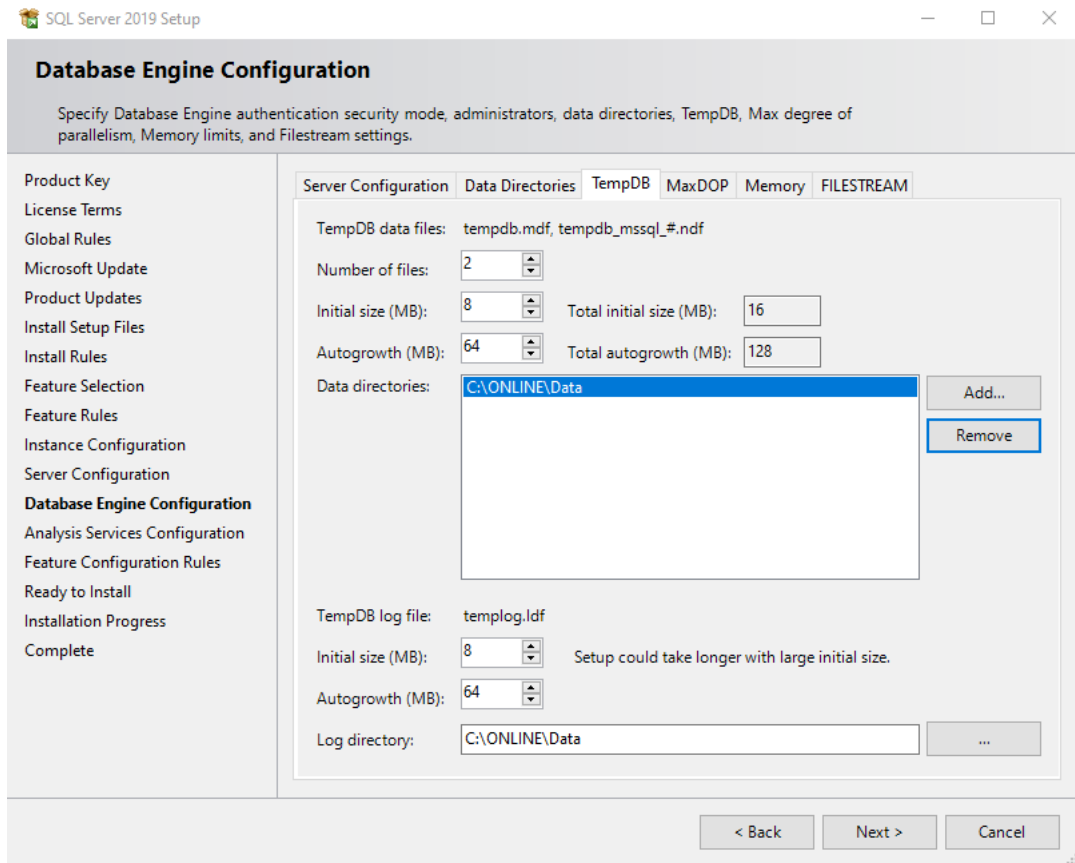


Figure 28: SQL Server Installation - TempDB

And also to the Analysis Services database files.

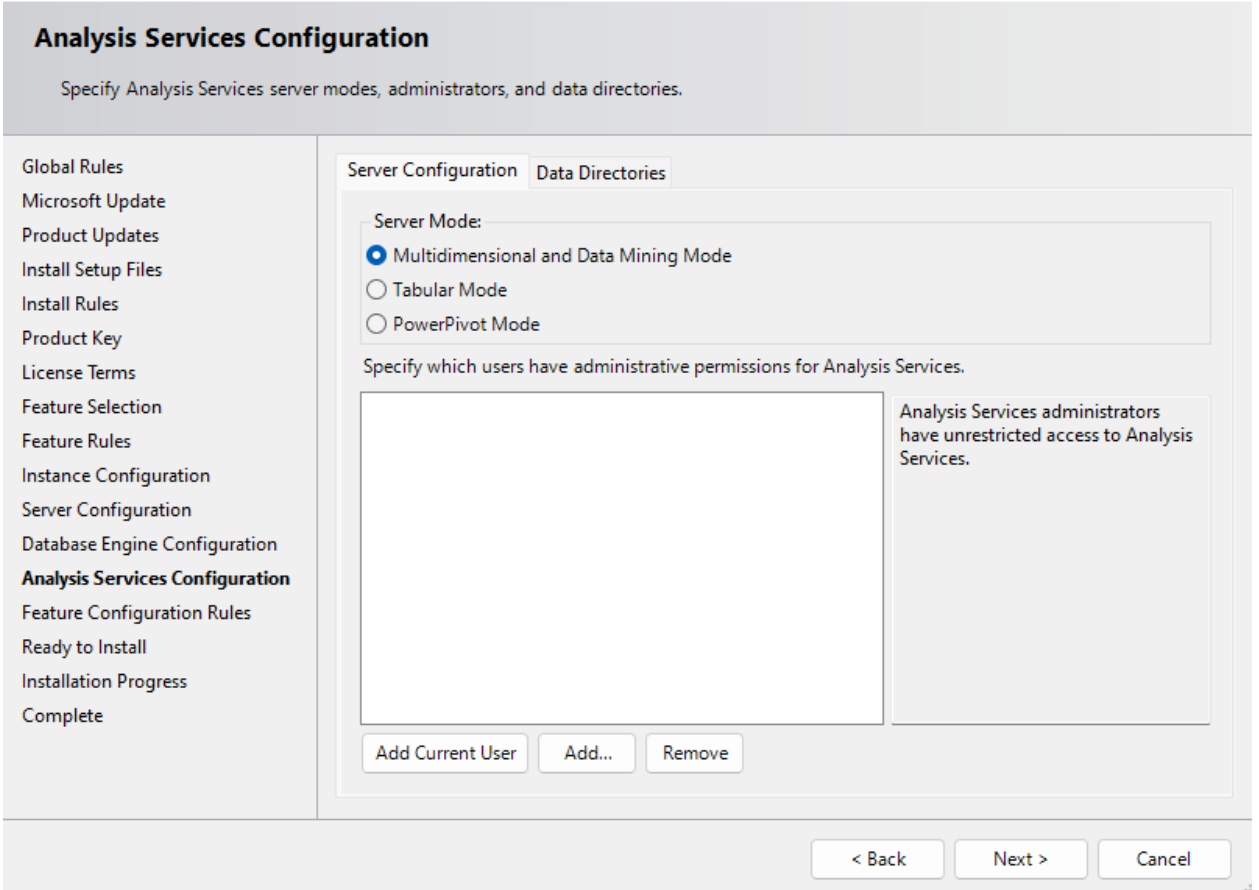


Figure 29: SQL Server Installation - Analysis Services

!!! note When installing SQL Server, the Analysis Services must be installed in **Multidimensional** mode.

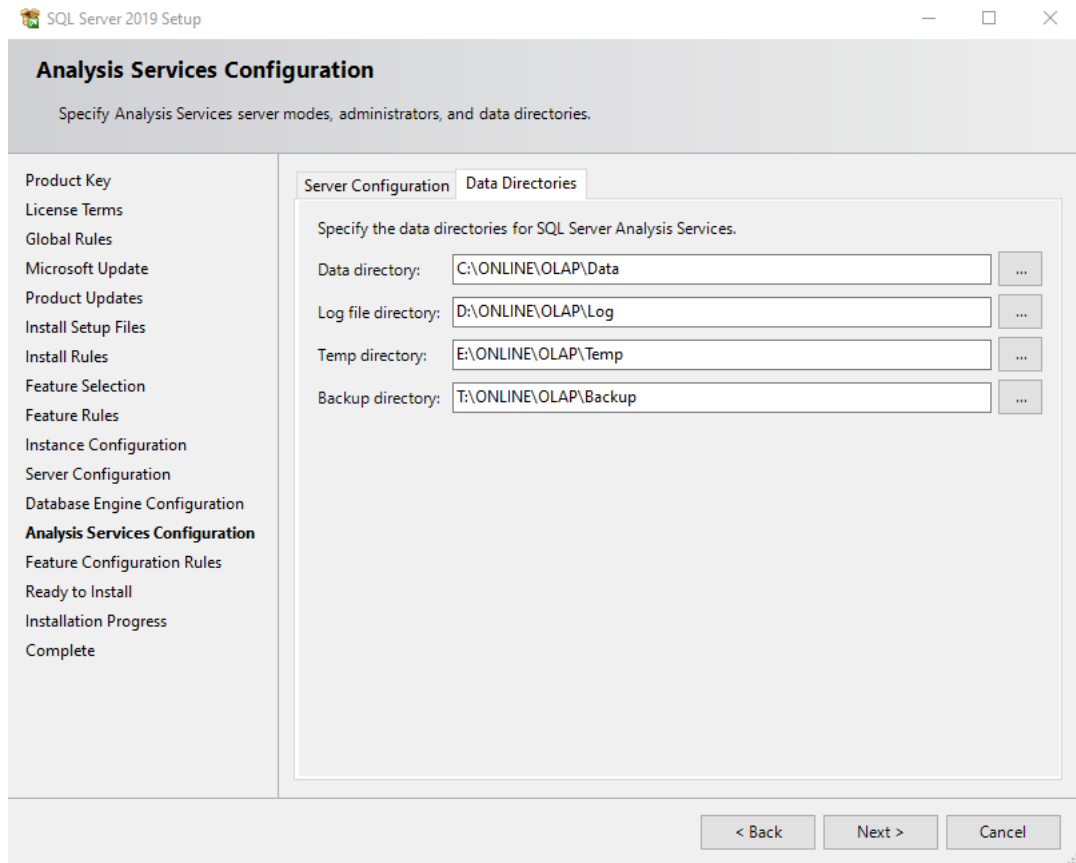


Figure 30: SQL Server Installation - Analysis Services

19.8 SQL Server Reporting Services

SQL Server Reporting Services server installation is performed outside the main SQL Server installation. This guide from Microsoft provides assistance to install this feature: <https://docs.microsoft.com/en-us/sql/reporting-services/install-windows/install-reporting-services?view=sql-server-ver15>.

19.8.1 Configuring Reporting Services for Critical Manufacturing

To configure reporting services for Critical Manufacturing in Always On on Availability Groups you need to create the database in its own availability group. Please refer to [this](#) Microsoft article for instructions on how to configure reporting server on that scenario.

The following changes should be performed on `ReportServer.Config`.

Reporting Services Authentication

!!! warning Basic authentication is a requirement for container-based installations. For more information on how to enable basic authentication, visit <https://docs.microsoft.com/en-us/sql/reporting-services/security/configure-basic-authentication-on-the-report-server?view=sql-server-ver16>. The host is able to use basic authentication but it must be manually enabled. Otherwise, it should be configured in `ReportServer.Config`:

```
<AuthenticationTypes>
  <RSWindowsNTLM/>
  <RSWindowsBasic/>
</AuthenticationTypes>
```



Reporting Services Concurrent Connections

By default Microsoft Reporting Services have the maximum number of requests per user set to 20. Given the nature of our system we recommend this limit to be increased to maximum number of users seeing reports simultaneously. If in doubt, increase it to at least 500.

```
<Add Key="MaxActiveReqForOneUser" Value="500" />
```

19.9 Creating Availability Groups in SQL Server

Pre-conditions:

- All database servers have **WSFC** installed;
- All SQL Servers have the **Always On High Availability** for Availability Groups enabled.

The most universal way (in place since the initial versions of SQL Server that support Always On High Availability) to create an Availability Group is as follows.

Create a temporary database for each planned Availability Group. If you want to create an Availability Group for Online, ODS and DWH, you need to create 3 temporary databases.

Start by creating a temp, tempODS and tempDWH:

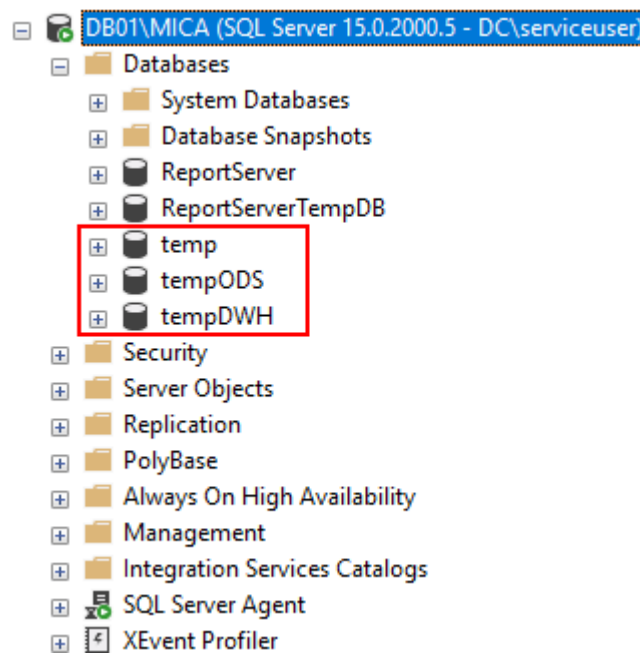


Figure 31: image20211008134948048

Before creating the Availability Group, you need to perform a full backup of each database:

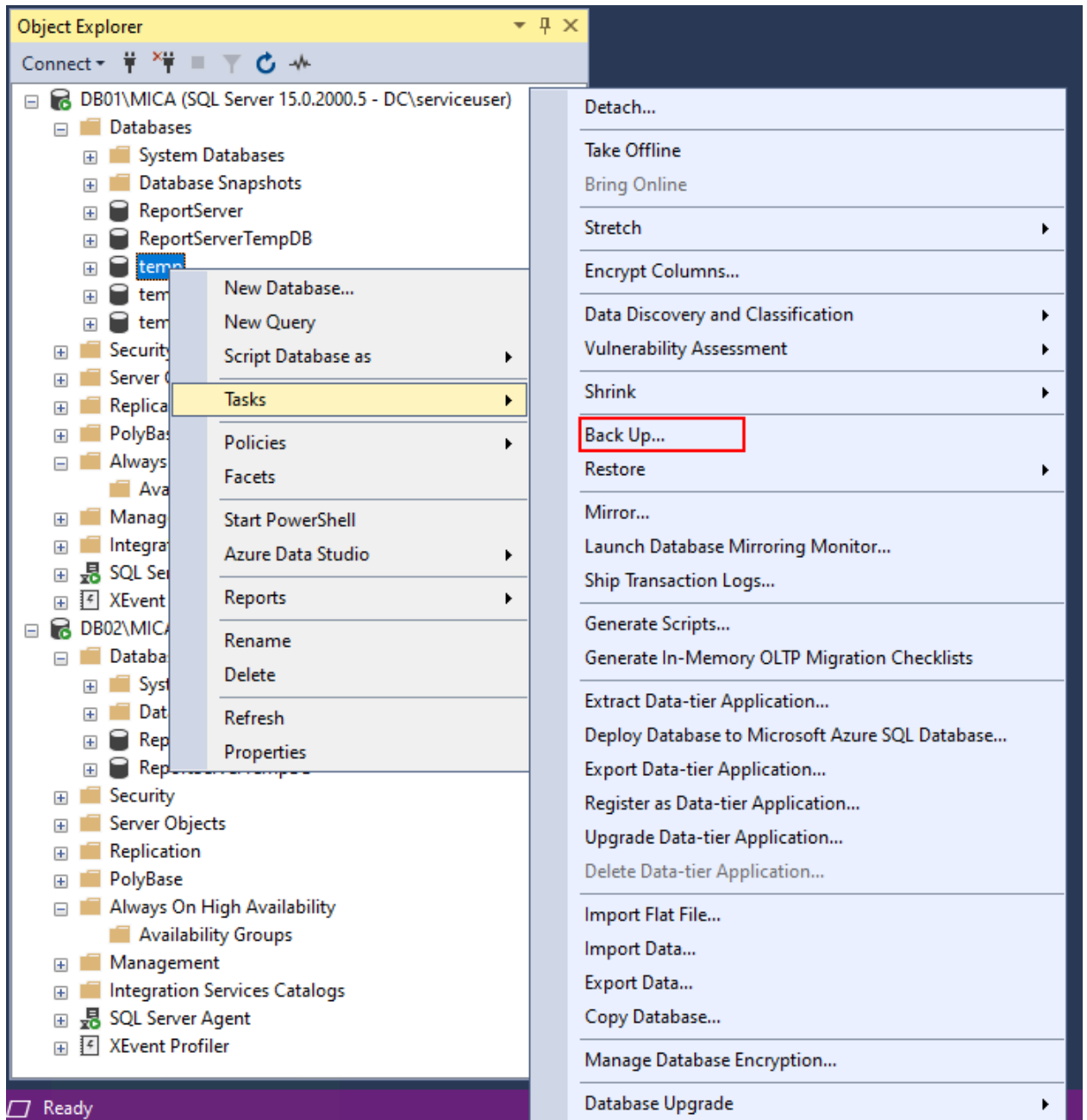


Figure 32: image20211008140247218

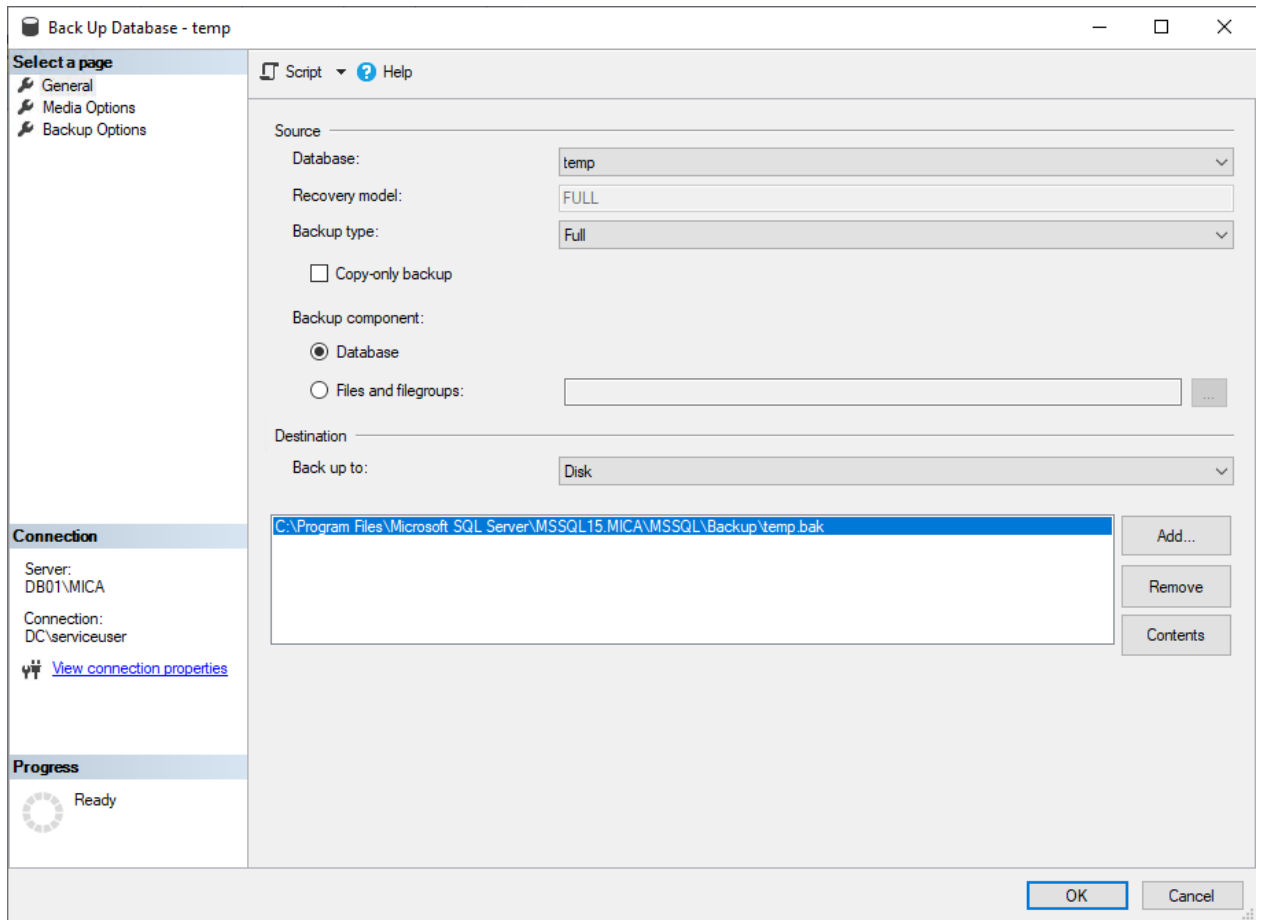


Figure 33: image20211008140314467

Then, go to the Always On High Availability folder and right click on it to start the **New Availability Group Wizard**:

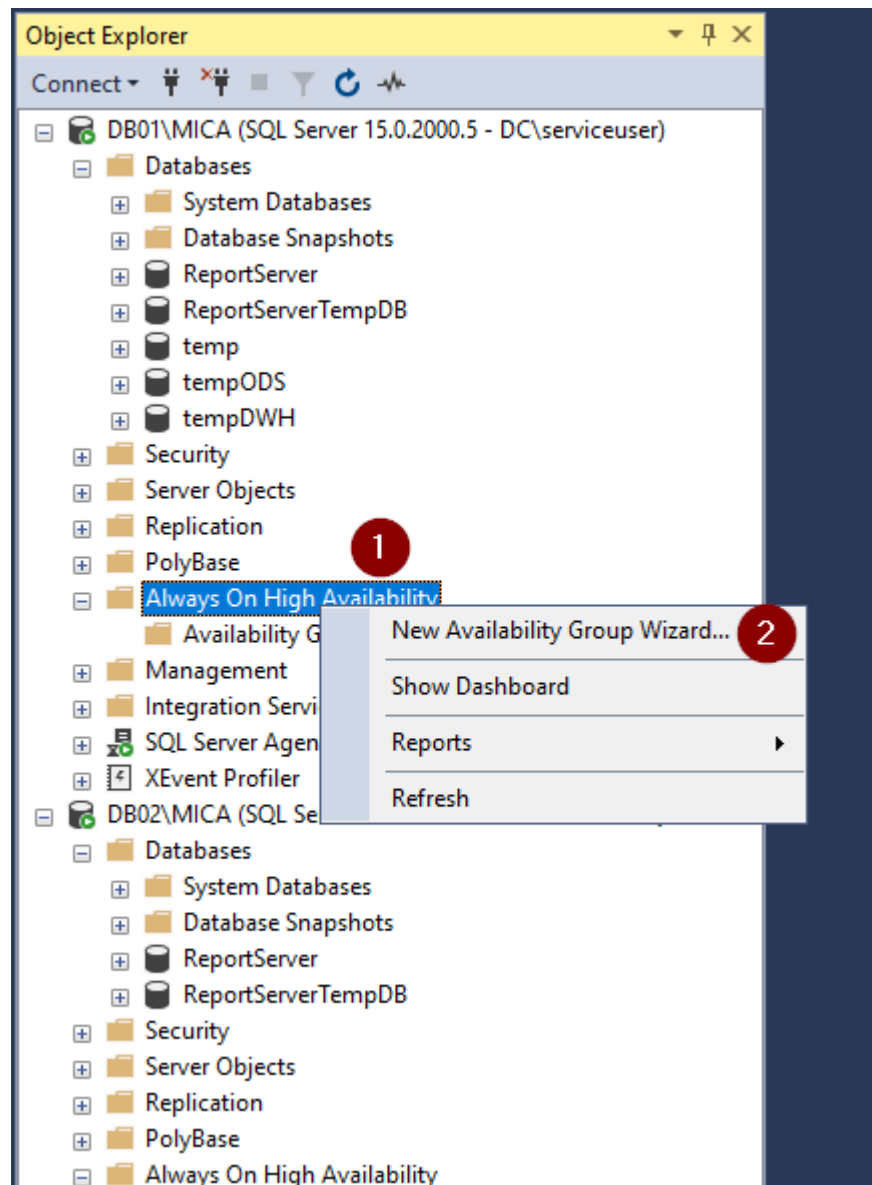


Figure 34: image20211008135213480

After this you must select the name of the Availability Group. In this case the system will be called **CMF** and three availability groups will be created (Online, ODS and DWH). According to the guidelines, the names will be **AGCMF**, **AGCMFODS** and **AGCMFDWH**:



Specify Availability Group Options

Introduction Help

Specify Options

Select Databases

Specify Replicas

Select Data Synchronization

Validation

Summary

Results

Specify availability group options

Availability group name:

Cluster type: ▾

Database Level Health Detection

Per Database DTC Support

Figure 35: image20211008140118280

Now select the database that will be associated to this Availability Group:

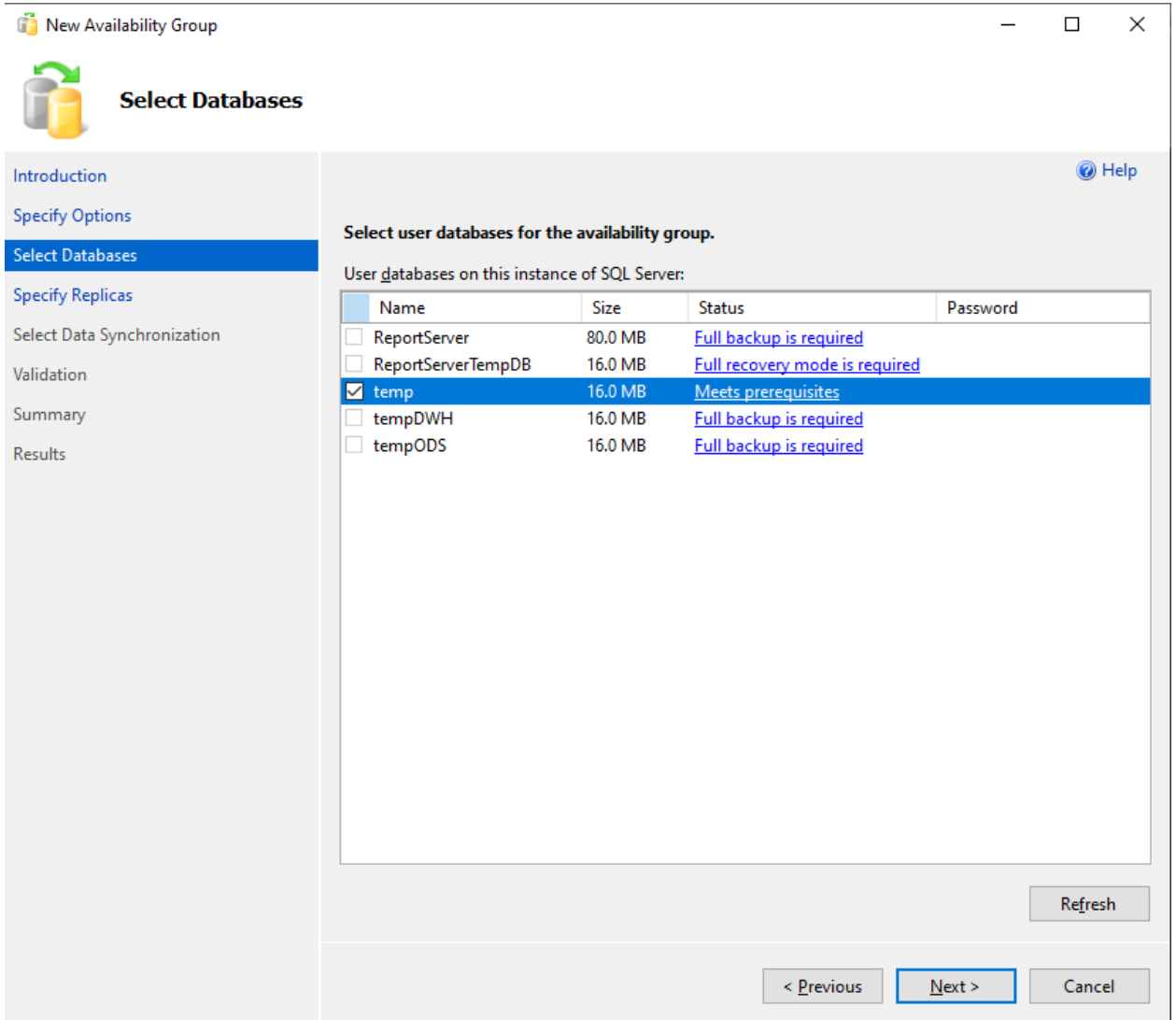


Figure 36: image20211008140504827

Next set the replicas that you need and set how you want the availability groups to work:



New Availability Group

Specify Replicas

Introduction
Specify Options
Select Databases
Specify Replicas
Select Data Synchronization
Validation
Summary
Results

Help

Specify an instance of SQL Server to host a secondary replica.

Replicas | Endpoints | Backup Preferences | Listener | Read-Only Routing

Availability Replicas:

Server Instance	Initial Role	Automatic Failover (Up to 5)	Availability Mode	Readable Secondary
DB01\MICA	Primary	<input checked="" type="checkbox"/>	Synchronous commit	Yes

Add Replica... Remove Replica

Summary for the replica hosted by DB01\MICA

Replica mode: Synchronous commit with automatic failover
This replica will use synchronous-commit availability mode and will support both automatic failover and manual failover.

Readable secondary: Yes
In the secondary role, this availability replica will allow all connections for read access, including connections running with older clients.

Required synchronized secondaries to commit: 0

< Previous Next > Cancel

Figure 37: image20211008140853232

Change the default settings of the original replica and then add a another replica:

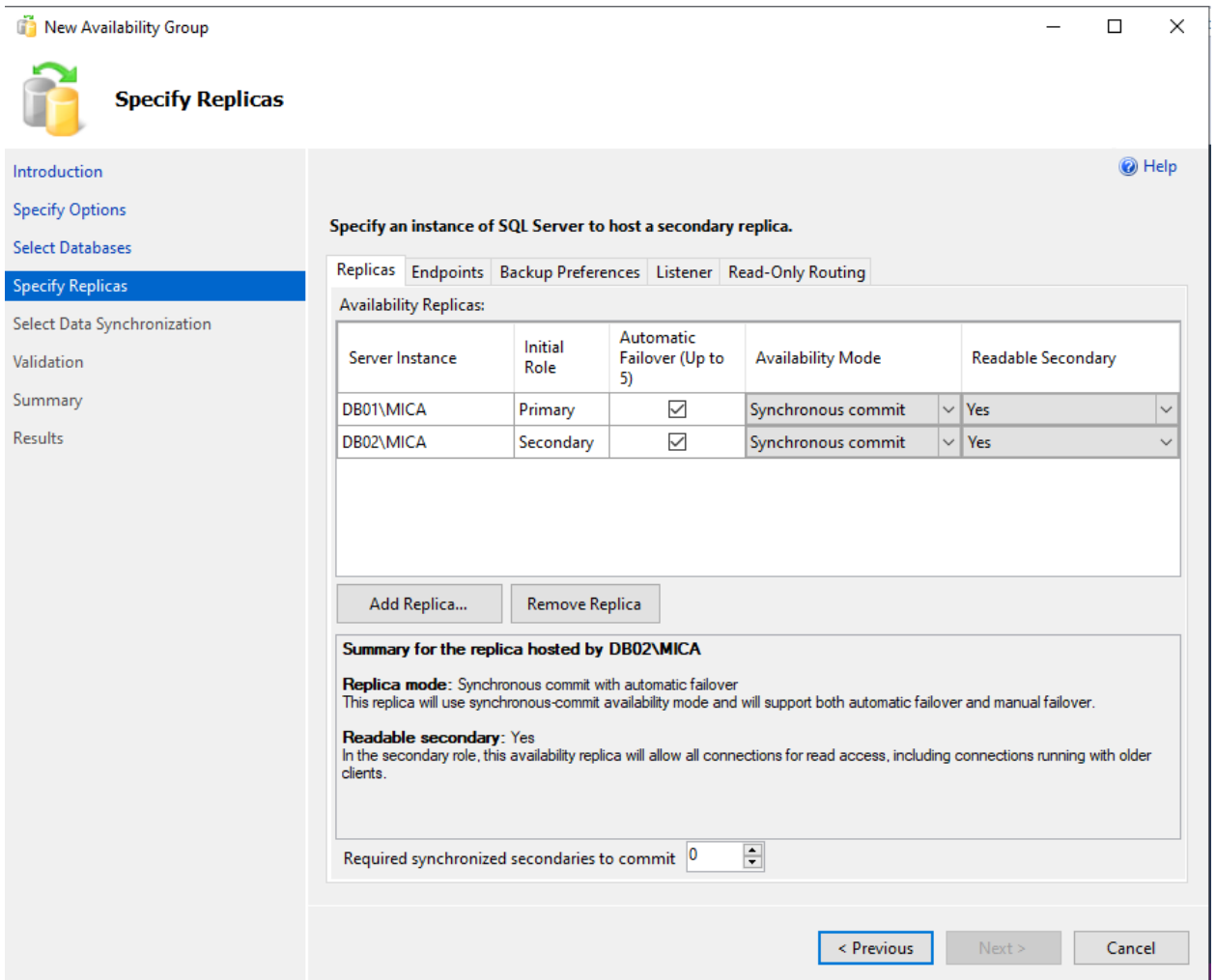


Figure 38: image20211008141014208

It is also possible to create the listeners for each Availability Group. Follow the guidelines and give the listener the same name as the Availability Group. Hence, the listener will be named [AGCMF](#):

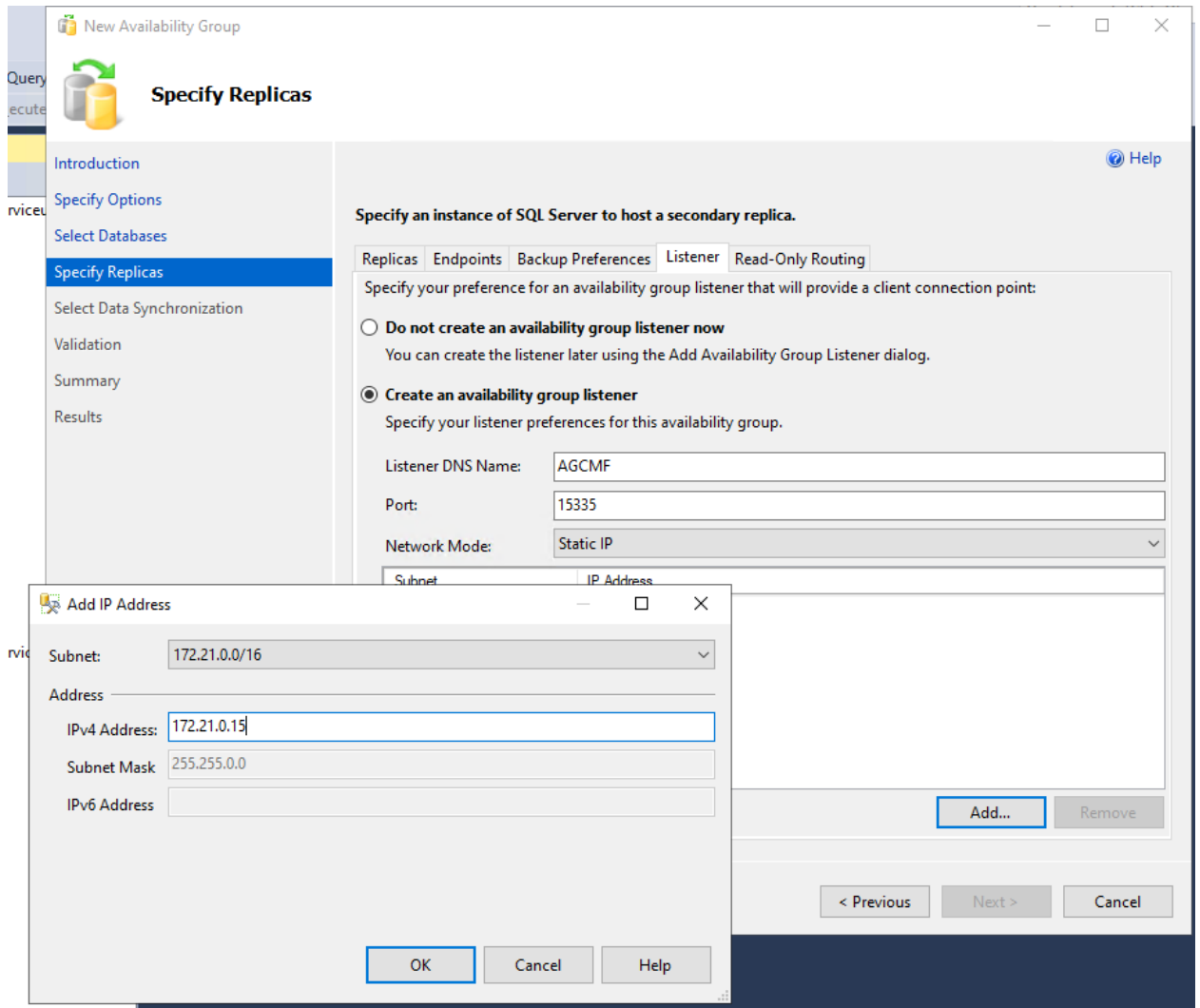


Figure 39: image20211008141729589

The port and IP Address must be selected according to the available values. Do not forget that the listener will be the **public address** of the Availability Group. Therefore, this will be the value that you will use when preparing the database server in the MES setup:

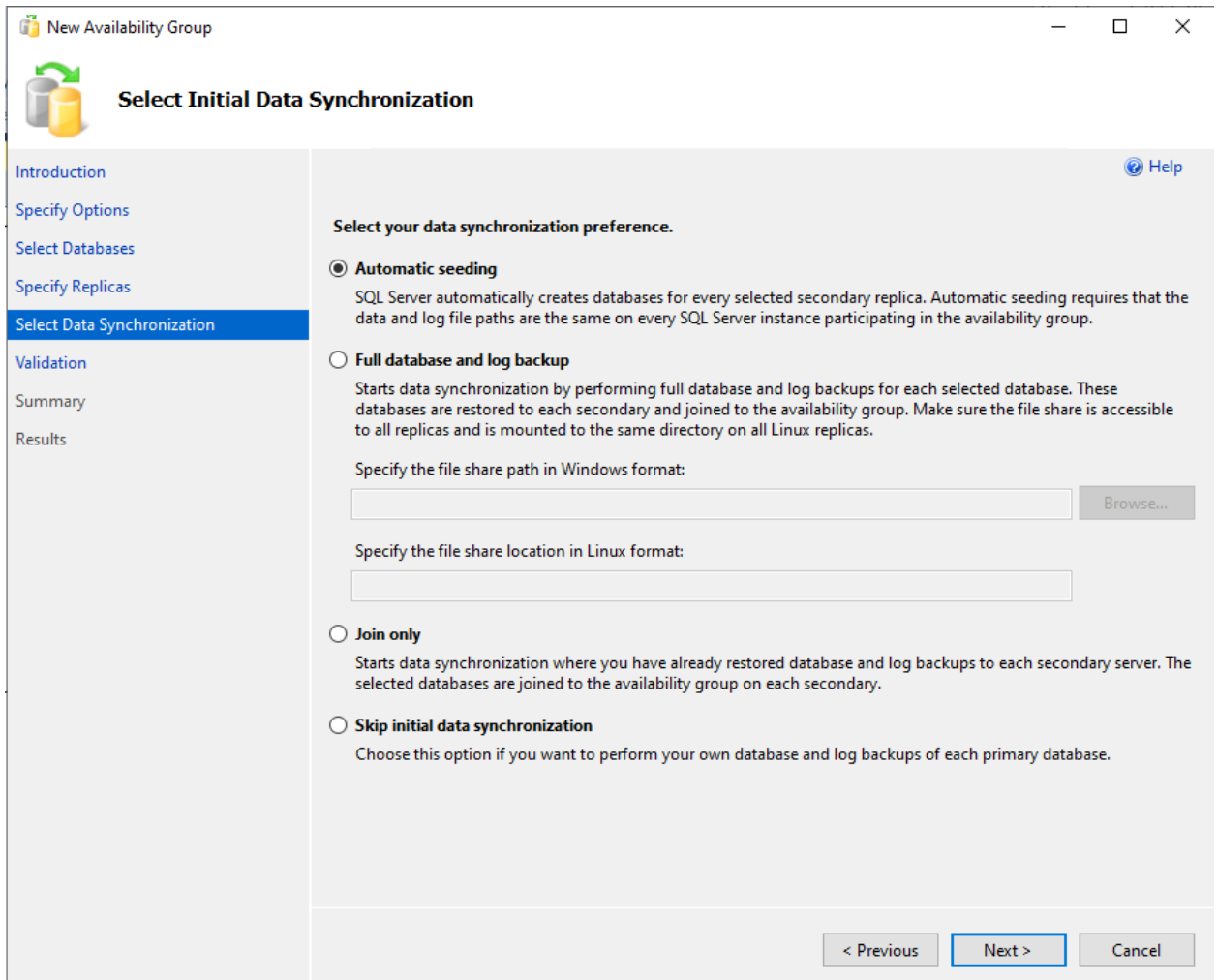


Figure 40: image20211008142003760

Use **Automatic seeding** as your data synchronization preference because experience has proved it to be the most stable, and it is recommended that all the SQL Server machines directory structure be the same.

The validation step should have an all green result:

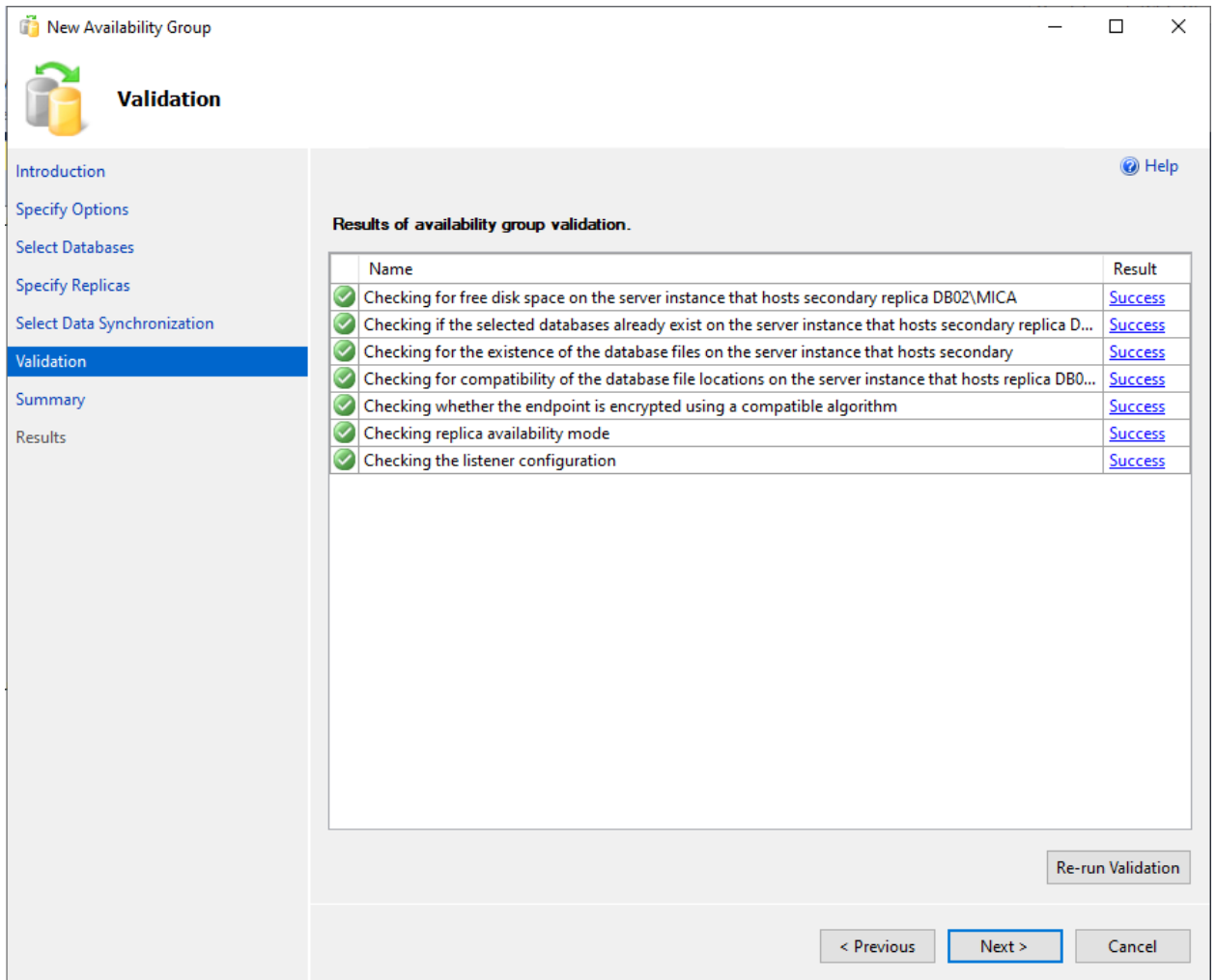


Figure 41: image20211008142312338

Finally, select **Next** and wait for the creation of the Availability Group. If no problem is found, the successful result will be clear:

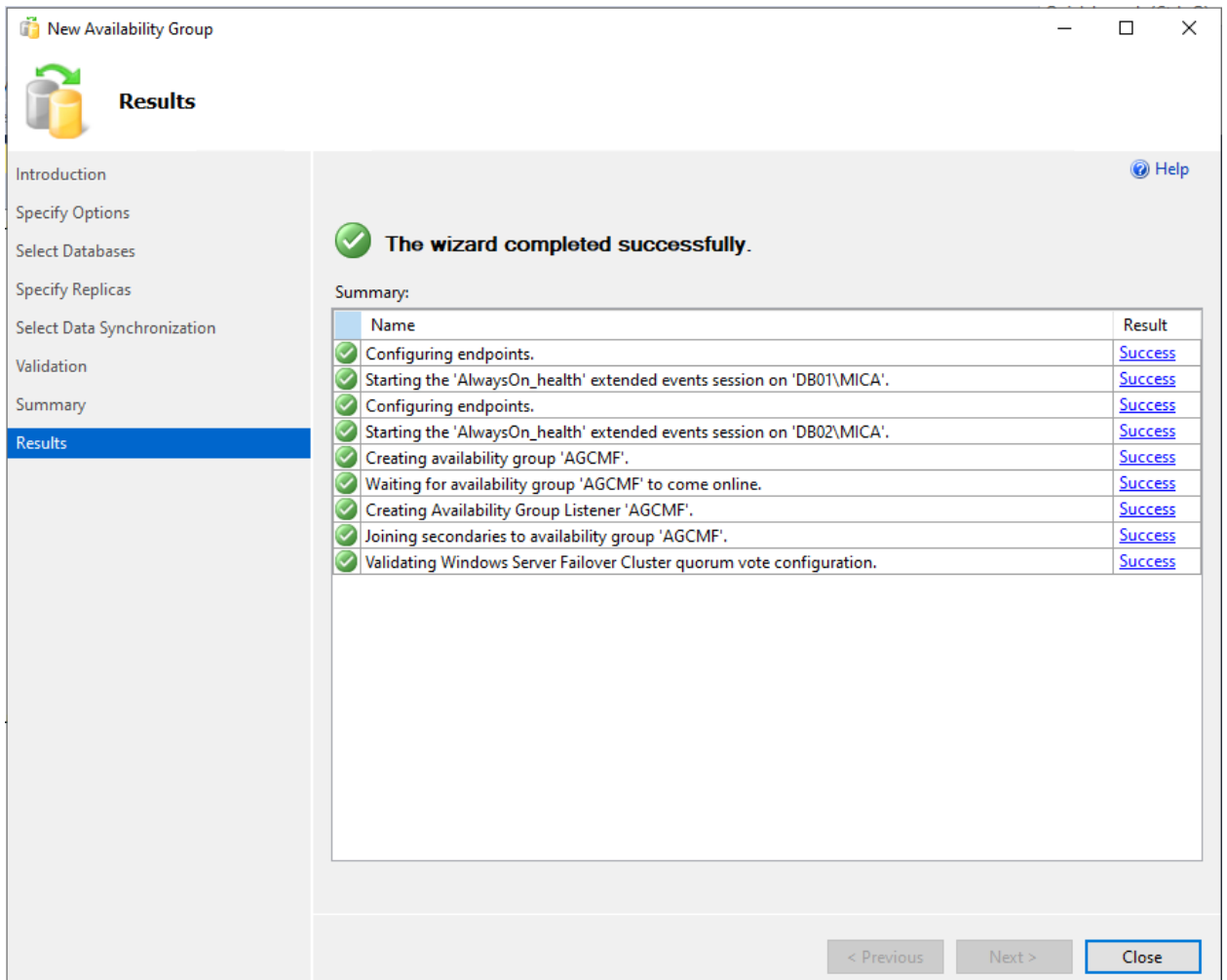


Figure 42: image20211008142526796

If you now go to the **Availability Groups** folder, you will see the new Availability Groups:

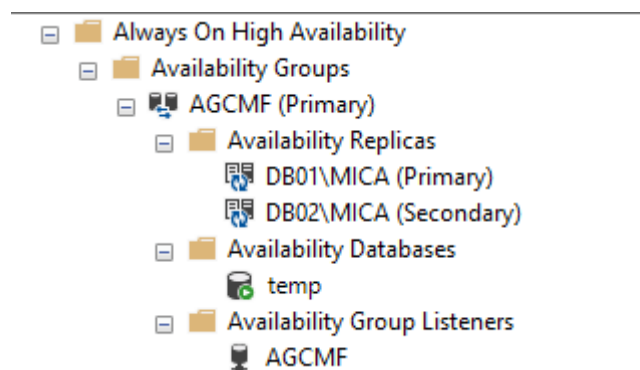


Figure 43: image20211008142618283

After creating the Availability Group, you can delete the temporary database. First remove the database from the Availability Group:

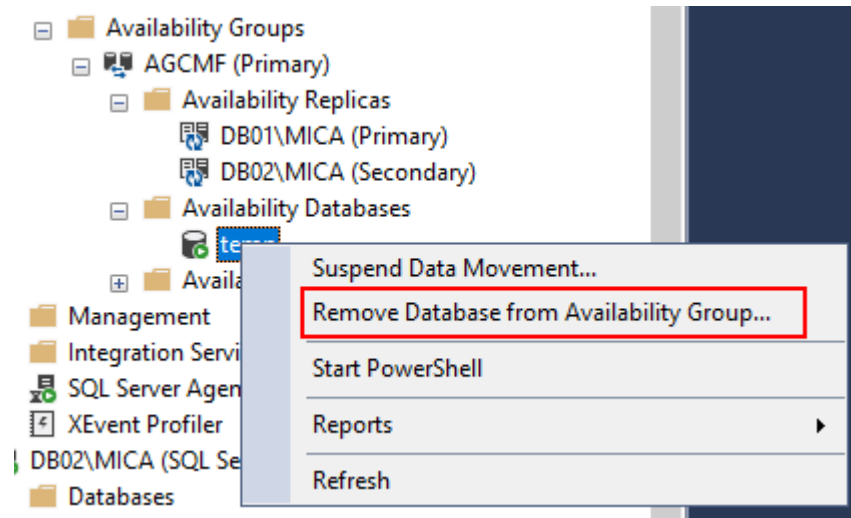


Figure 44: image20211008142956229

And then delete the database from all the nodes:

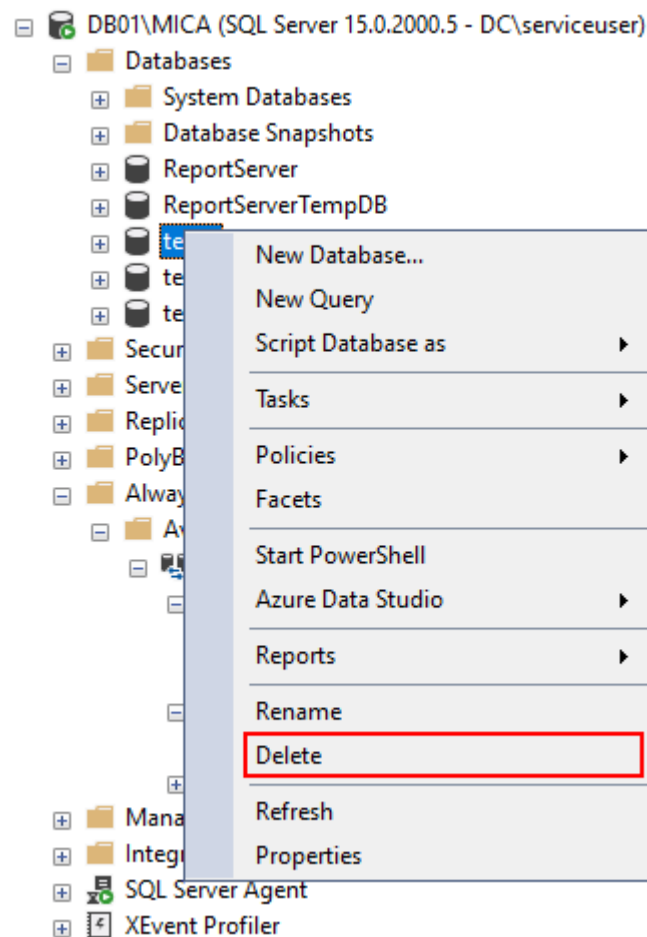


Figure 45: image20211008143106749

!!! note The secondary database might be in a restoring state. This is not a problem as it is a temporary database:

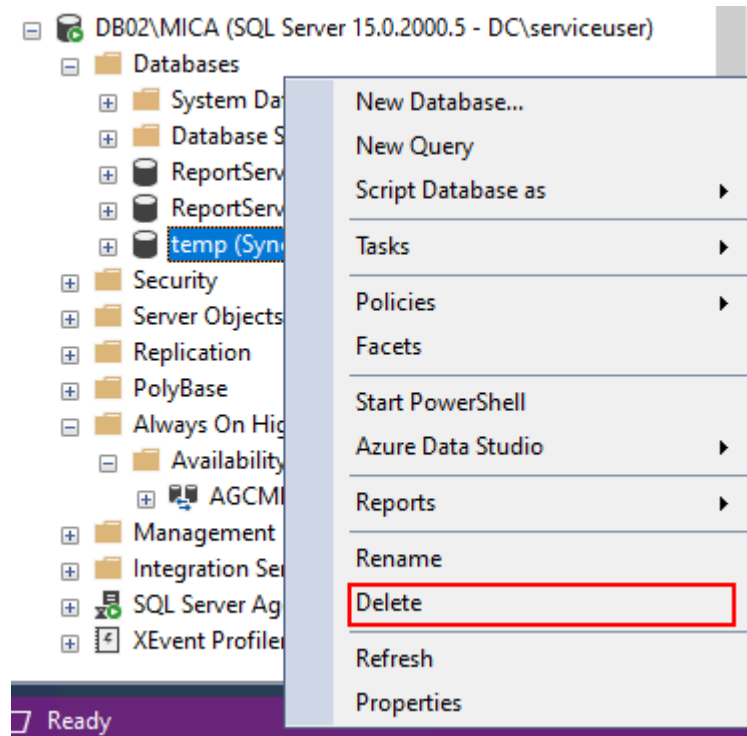


Figure 46: image20211008143211260

!!! info You need to repeat these steps for the other two Availability Groups.

For more information about SQL Server Always On for Availability Groups please refer to [SQL Server AlwaysOn Availability Groups](#)

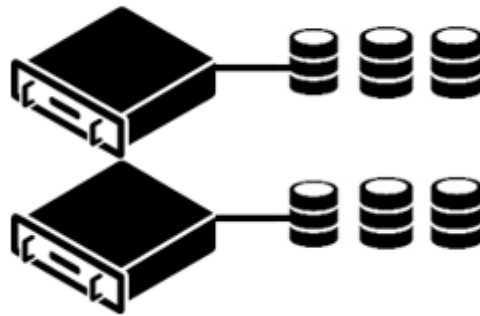


Figure 47: SQL Server Always On Availability Groups

19.10 Microsoft Advanced Firewall Configuration

The following inbound ports must be allowed in the database servers for proper operation:

Table 14: Advanced Firewall Configuration

Port Number	Purpose
1433	SQL Server Database Engine



19.11 SQL Server Installation Advanced Topics

In this section, we present some guidelines and considerations for optimizing SQL Server at the installation phase.

- [Breaking Up SQL Server Databases into Multiple Files](#)
- [Database Filegroups and Data Files](#)
- [TempDB Database Configuration](#)
- [Storage and RAID Levels](#)
- [Recommendations for SQL Server Installation Options](#)

19.12 Installing to an Azure managed instance

Critical Manufacturing can use an Azure SQL Managed Instance as the database. This will require having the files located in the actual instance, together with the proper configuration in the installation process.

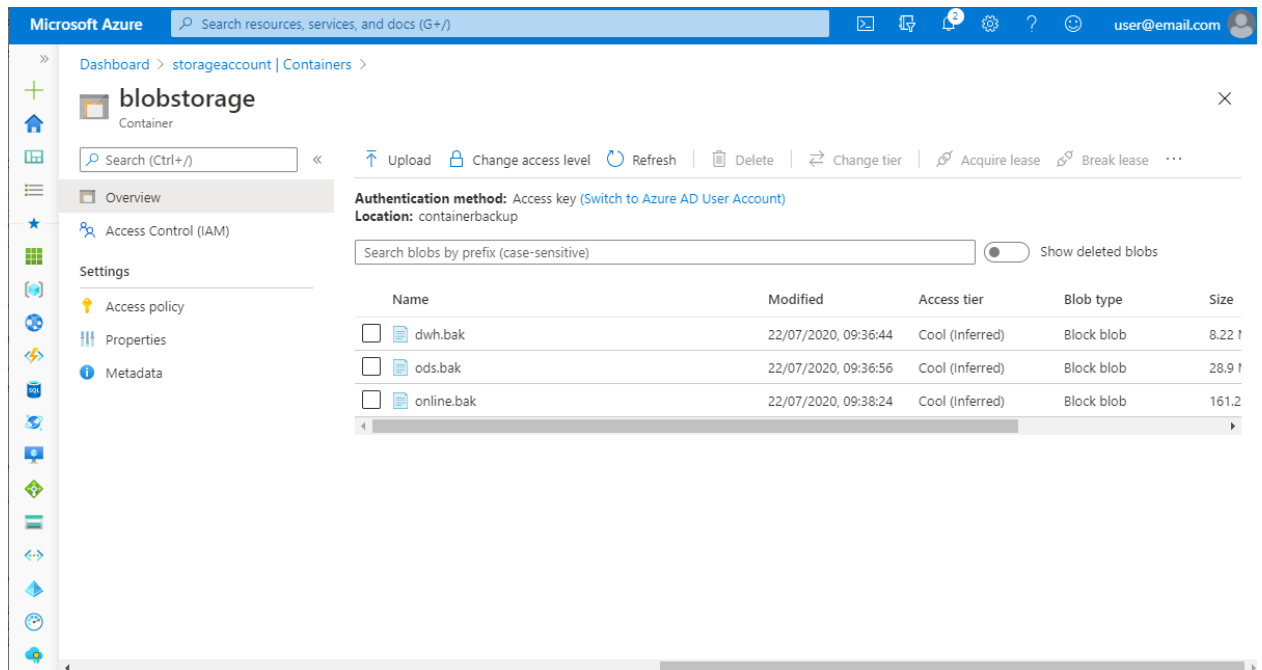


Figure 48: azureblobstorage



20 Always On Availability Groups

The Always On Availability Groups feature is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Always On Availability Groups maximize the availability of a set of user databases for an enterprise. An availability group supports a failover environment for a discrete set of user databases, known as availability databases, that fail over together. An availability group supports a set of read-write primary databases and one to eight sets of corresponding secondary databases. Optionally, secondary databases can be made available for read-only access and/or some backup operations.

20.1 Always On on Availability Groups versus on Failover Cluster Instances

An availability group is composed by a primary availability replica and one to four secondary replicas that synchronize via log-based data movement for data protection without the need for shared storage. This reduces the overall cost of the solution.

An availability group listener on the primary node responds to connection requests to the virtual network name, and redirects each request to the appropriate SQL Server instance. In the event of a failover, the system does not transfer ownership of shared physical resources to another node. WSFC assists SQL Server in the reconfiguration of a secondary replica on another SQL Server instance to become the availability group's primary replica. The availability group virtual network name resource is moved to that instance to complete the failover process.

20.2 Always On Terms

Table 15: Always On Terms

Name	Definition
Availability group	A container for a set of databases, availability databases, that fail over together.
Availability database	A database that belongs to an availability group. For each availability database, the availability group maintains a single read-write copy (the primary database) and one to eight read-only copies (secondary databases).
Primary database	The read-write copy of an availability database.
Secondary database	A read-only copy of an availability database.
Availability replica	An instantiation of an availability group that is hosted by a specific instance of SQL Server and maintains a local copy of each availability database that belongs to the availability group. Two types of availability replicas exist: a single primary replica and one to eight secondary replicas.
Primary replica	The availability replica that makes the primary databases available for read-write connections from clients and also sends transaction log records for each primary database to every secondary replica.



Name	Definition
Secondary replica	An availability replica that maintains a secondary copy of each availability database and serves as a potential failover target for the availability group. Optionally, a secondary replica can support read-only access to secondary databases and can support creating backups on secondary databases.
Availability group listener	A server name to which clients can connect in order to access a database in a primary or secondary replica of an Always On availability group. Availability group listeners direct incoming connections to the primary replica or to a read-only secondary replica.

21 Breaking Up SQL Server Databases into Multiple Files

File access is fundamental and an essential consideration to an OLTP system. If your system is busy (it gets a high number of transactions), or you expect it to be in the future, put in a little planning to prevent waiting for the disk. The next topics will address file placement, as well as log files and adding data.

For additional information, please refer to the following links:

- [Optimizing TempDB Performance](#)
- [Capacity Planning for TempDB](#)
- [Place Data and Log Files on Separate Drives](#)
- [How to Optimally Use SSDs Without Burning Them Out](#)



22 Database Filegroups and Data Files

Every SQL Server database contains a data file (.MDF) and a transaction log file (.LDF); however, you can add additional files to the database. These files are called secondary files (.NDF) and will also be used to store data rows or indexes. The files are assigned to filegroups in the same way that files are assigned to folders in a file system. When assigning several files to the same filegroup, data is distributed evenly between all of them, in a round-robin way.

When Critical Manufacturing is installed, it creates six data files:

1. `Primary`
2. `MainTableDat_1`
3. `MainTableIdx_1`
4. `HstTableDat_1`
5. `HstTableIdx_1`
6. The transaction log file

Additionally, it creates the necessary TempDB data and log files.

For Critical Manufacturing database, we strongly recommend that you isolate the transaction log file(s) into a separate LUN. Ideally, this LUN will be completely separated from the data files - even at disk level (most SAN vendors provide a way to set aside a few disks and isolate them to provide a write-optimized, mirrored disk resource like a RAID 10 LUN). With SQL Server's synchronous, write-log first transaction log behavior, the transaction log write latency is a critical potential bottleneck. You can avoid this by using proper SAN and database data file architecture.

SQL Server writes each transaction to the log before writing it to the data file. If both data and log are on the same disk, this means a lot of jumping back and forth between the two files and extra time taken up. Nonetheless, if the log is on its own disk, it has a dedicated read/write head to write the transactions, uninterrupted by data file writes.

It is also important to keep your log files separated, from a disaster recovery perspective. In case the disk that houses your data files fails, you will be glad to have the transaction log accessible in another disk. As a result, you are able to back up the tail of the log and be able to recover to the point of failure.

By definition, Critical Manufacturing uses one data file per filegroup.



23 Recommendations for SQL Server Installation Options

This section describes recommendations based on best practices for a Critical Manufacturing typical installation and workload. Some parameters might have to be adjusted during the lifetime of the application or if this recommendation is updated.

23.1 Install Analysis Services in Multidimensional and Data Mining Mode

Analysis Services provides online analytical processing (OLAP) and data mining functionality for business intelligence applications. When installing SQL Server, the Analysis Services must be installed in *Multidimensional mode*. This is one of three server modes in which Analysis Services runs in.

23.2 Provision Storage For The Operating System and for SQL Server

The following setup is recommended for each SQL Server Instance:

- Critical Manufacturing databases data files - **G:**
- Critical Manufacturing databases log files (including tempdb Log) - **L:**
- Critical Manufacturing databases backup disk - **H:**
- SQL Server Instance tempdb data file(s) - **T:**

In addition:

- Transaction log files (**L:**) should not be in the same disks as data files (**G:**) due to their random vs sequential way of writing in log files and data files.
- Use **RAID 10** (better write performance and reliability). Always make sure write cache is enabled (if applicable).
- TempDB should ideally be in SSD disks. If not possible, make sure it is using different spindles rather than the database files.

23.3 Operating System Configuration

The following configurations are recommended:

- Configure the Windows page file - we typically create a 2GB size page file on the system drive. Page file size can be found in the system properties of Windows Server.
- Set anti-virus exclusions - You need to configure exclusions for all SQL Server files per Microsoft's guidelines: <https://support.microsoft.com/en-us/kb/309422>.
- If you have additional tools that restrict the creation/modification of files, exclusions should also be set.
- Make sure server "Power Options" are set to "High Performance".
- Format the drives with 64K allocation blocks.

This only applies to drives holding SQL Server database and log files (including tempdb). Your C drive / system drive should be separate and 4K block size is appropriate for that logical drive.

23.4 Service Accounts and Permission Granting

Make sure to grant the 'Perform Volume Maintenance Tasks' and 'Page locks in memory' rights to the account that will be used for the SQL Server service (the engine, not the agent). This will enable the instant file initialization (IFI). Please refer to the link <https://msdn.microsoft.com/en-us/library/ms175935.aspx> for more information.



23.5 SQL Server Installation and Configuration

The following configurations are recommended:

- Make sure the TCP/IP Protocol is enabled.
- Configure this in the SQL Server Configuration Manager under "SQL Server Network Configuration." Enabling the TCP/IP protocol will only take effect after the SQL Server instance is restarted.
- Test Instant File Initialization (IFI)
- Create an empty database. Grow the data file by 5GB. If it does not complete immediately, then IFI is not working (revisit the previous step where it was granted.) If you have verified IFI is working, go ahead and drop the empty database.

23.6 TempDB Configuration

By default, the TempDB files are placed on the same drive as the SQL Server binaries. Even if the user chooses a custom install, TempDB still goes on the same drive as the other data files, and that is not advisable. Alternatively, the TempDB data files should be on their own dedicated drive.

23.7 Move TempDB to its own drive

In this example, we put the data file on the T drive and the log file on the L drive. (Important: directory paths must exist beforehand).

```
use master;
GO
alter database tempdb modify file (name='tempdev', filename='T:\MSSQL\DATA\tempDB.mdf', size = 1MB);
GO
alter database tempdb modify file (name='templog', filename='L:\MSSQL\LOGS\templog.ldf', size = 1MB);
GO
```

After this code runs, restart the SQL Server. That will create the new TempDB file on the new drive. Manually delete the old TempDB file on the original drive, since SQL Server does not delete it itself.

Grow that file and add additional data files. Now that TempDB is on the right drive, expand it to the full size according to your preferences and then create additional TempDB files.

```
USE [master];
GO
alter database tempdb modify file (name='tempdev', size = 2GB, FILEGROWTH = 100MB);
GO
```

The current guidance from Microsoft in KB 2154845 is to use the same number of tempdb files as the number of logical processors up to 8 logical CPUs. Do not add more, unless you observe you have contention.

The code to create one additional TempDB data file can be seen below - you can modify this in order to have more files:

```
USE [master];
GO
ALTER DATABASE [tempdb] ADD FILE (NAME = N'tempdev2', FILENAME = N'T:\MSSQL\DATA\tempdev2.ndf' , SIZE =
    2GB , FILEGROWTH = 100MB)
GO
```

The data file creation should only take a couple of seconds - if it takes more than ten seconds, then instant file initialization is not configured correctly (Revisit the section where IFI is enabled). On a general note: Autogrowing files by at least 100MB for the transaction log is acceptable, but this value may need to be higher to provide enough space to avoid autogrowing again quickly. The best option is to avoid autogrowing in the first place, by correctly sizing the files.



23.8 Configuration of SQL Server Max Degree of Parallelism

Set this value to the number of physical cores in a single NUMA node (processor) socket on your hardware or less.

Example to set the Max Degree of Parallelism to 8:

```
USE [master];
GO
ALTER DATABASE [tempdb] ADD FILE (NAME = N'tempdev2', FILENAME = N'T:\MSSQL\DATA\tempdev2.ndf' , SIZE =
    2GB , FILEGROWTH = 100MB)
GO
```

23.9 Configuration of SQL Server Cost Threshold for Parallelism

```
USE CriticalManufacturing;
GO
EXEC sp_configure 'show advanced options', 1;
GO
RECONFIGURE WITH OVERRIDE;
GO
EXEC sp_configure 'max degree of parallelism', 8;
GO
RECONFIGURE WITH OVERRIDE;
GO
```

23.10 Configuration of SQL Server Max Memory

By default, SQL Server's maximum memory is 2147483647, i.e. no memory limit. A limit should be defined so that paging to disk is less likely. It is advisable to leave 4GB or 10% of total memory free (whichever is larger on your instance to begin with) and adjust as needed.

Example for a server with 32GB of memory:

```
EXEC sys.sp_configure 'max server memory (MB)', '29491';
GO
RECONFIGURE WITH OVERRIDE;
GO
```

To double-check if you are paging to disk, go to *Task Manager*, open the *Performance* tab, and look at the free memory metric for Windows 2008. If it is under 200, you are in danger of swapping to disk.

23.11 SQL Server Maintenance Setup

Configure and schedule regular maintenance for all of the following:

- Full (and possibly differential) backups;
- Log backups (every 15 mins);
- CheckDB;
- Index maintenance;

We recommend using free scripts from Ola Hallengren to create customized SQL Server Agent: <http://ola.hallengren.com/>.



23.12 Set Compatibility level

The compatibility level set by Critical Manufacturing MES to its databases is equal to the maximum compatibility level supported by the oldest version of Microsoft SQL Server supported on each version.

!!! info For version 11, the oldest supported SQL Server version is SQL Server 2019 Standard Edition. The maximum database compatibility level support by this version is **150**, thus the compatibility level used by Critical Manufacturing MES v11 databases is **150**.

For more information, see <https://learn.microsoft.com/en-us/sql/t-sql/statements/alter-database-transact-sql-compatibility-level?view=sql-server-ver16>.

23.13 Other SQL Server Settings (to check after Critical Manufacturing has been installed)

Enable the option *optimize for ad hoc workloads* at the instance level.

```
EXEC sys.sp_configure 'optimize for ad hoc workloads', 1
GO
RECONFIGURE WITH OVERRIDE;
GO
```

Enable *backup compression default*

```
EXEC sp_configure 'backup compression default', '1';
GO
RECONFIGURE WITH OVERRIDE;
GO
```

Enable *Page_Verify* for a high level data-file integrity in our databases.

```
/*This will script out the command for you, check it and execute the output
*/
SELECT 'ALTER DATABASE ' + QUOTENAME(s.name) + ' SET PAGE_VERIFY CHECKSUM WITH NO_WAIT;'
FROM sys.databases AS s
WHERE s.page_verify_option_desc <> 'CHECKSUM';
GO
```

This script just creates the TSQL for your change: you still need to copy it and execute it in another window. When you configure your full backups, use the *With Checksum* option to check the checksums each time a full backup is run.

23.14 Manually Enable Backup Jobs

The SQL Server installation creates the backup jobs but does not enable them by default. In order for the jobs to run, they must be enabled manually.

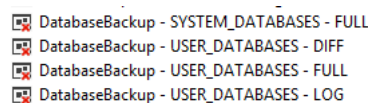


Figure 49: SQL Server Backup Jobs

For more information, see [Database Backup and Restore](#) section of the Operations Guide.



23.15 Optional Settings

Enable SQL Trace Flags. Trace flags are used to temporarily set specific server characteristics or to switch off a particular behavior. To find out more on how to enable SQL Trace flags on startup, visit <https://msdn.microsoft.com/en-us/library/ms190737.aspx>.

The trace flags indicated below are the most useful in this context:

- **1117** - Causes files in a filegroup to grow at the same time. Useful with tempdb.
- **1118** - Removes the use of mixed extents. Most often used to help with tempdb contention.
- **2371** - Statistics are recomputed less rarely as tables grow larger.
- **3226** - Stops logging all of your backup success entries to the error log.
- **4199** - To get new query optimizer hot fixes that produce different query plans.
- **8048** - Enables SOFT NUMA when SQL's NUMA-awareness code does not work on larger systems.



24 Storage and RAID Levels

The RAID levels have a big impact on both performance and availability. The most common RAID levels that you will encounter are:

- **RAID 0** (sometimes called *disk striping*). This RAID level spreads all the data across all the available disks. You often see this RAID level used in different database benchmarks. **RAID 0** provides good performance, but you should never use it on a production server because one disk failure will result in data loss.
- **RAID 1** (sometimes called *disk mirroring*). With **RAID 1**, data is mirrored on the disks. Read and write performance is good, but the overall disk capacity is cut in half. **RAID 1** is often used for SQL Server log files. It can sustain one disk failure.
- **RAID 5** (sometimes called *disk striping with parity*). **RAID 5** stripes data across multiple disks and uses a disk for data redundancy. It is often used for data files. This RAID level provides good read performance and can sustain the failure of one disk. However, it is known for slower write performance.
- **RAID 10** (sometimes called *disk mirroring with striping*). **RAID 10** combines the performance of striping with the protection of mirroring. **RAID 10** provides the highest levels of performance and availability out of the different RAID levels. **RAID 10** uses twice as many disks as **RAID 5**, but it can sustain multiple disk failures. A **RAID 10** array can sustain failures for as many as half of the disks in the set. **RAID 10** is good for both data and log files.

For Critical Manufacturing databases, we strongly recommend to use:

- **RAID 5** for the ODS and Data Warehouse databases.
- **RAID 10** or **RAID 5** for the Online database.
- Solid State Disks (SSD) or **RAID 10** for TempDB database.



25 TempDB Database Configuration

TempDB is critical to performance as it is used by several users and system actions such as cursors, temp tables, hash table for sorts, re-indexing, and so on. It is recommended that you handle TempDB before you separate your OLTP data and log files.

TempDB is typically one of the most active databases on a production SQL Server instance, so it is highly recommended that a separate LUN for the TempDB is used. The TempDB data and log files should be placed on different physical drives, apart from your production database data and log files. Because TempDB is so active, it is also a good idea to make sure the drives are protected with SSD or striped with RAID 10.

The Microsoft SQL Server Customer Advisory Team (SQLCAT) has recommended that TempDB should have one data file for each CPU core. However, this recommendation is best suited for very heavy workloads. It is more commonly recommended that TempDB have a 1:2 or 1:4 ratio of data files to CPU cores. As with most performance recommendations, this is a general guideline. The requirements for your system will vary. If you are unsure of how many data files to use for TempDB, a common recommendation is to start with four data files. Typically, one log file is enough for TempDB. (For more in-depth TempDB recommendations, see the resources listed in the *Breaking Up SQL Server Databases into Multiple Files* section.)

By default, TempDB files are placed on the same drive as the SQL Server binaries. Even if the user chooses a custom install, the TempDB is still placed in the same drive as the other data files. Instead, the TempDB data files should be stored on their own dedicated drive. This must be corrected by first moving the TempDB to its own separate drive. In the example below, we put the data file on the T: drive and the log file on the L: drive. Note that the directory paths must already exist before running the command.

```
USE [master]
GO
ALTER DATABASE [tempdb] MODIFY FILE (NAME='tempdev', FILENAME='T:\MSSQL\DATA\tempDB.mdf', SIZE = 1mb)
GO
ALTER DATABASE [tempdb] MODIFY FILE (NAME='templog', FILENAME='L:\MSSQL\LOGS\templog.LDF', SIZE = 1mb)
GO
```

In the example above, we only set a 1mb file size because SQL Server behaves uncommonly. Even though we are instructing it to use a different drive letter, it will look for this amount of free space on the current TempDB drive. If the SQL Server had been installed on the server's C: drive, for example, and we had tried to create a 10GB TempDB file on a T: drive, that SQL command would have failed if 10GB of free space were not on the C: drive.

After the command above is run successfully, it is necessary to restart the SQL Server instance. That will create the new TempDB file on the new drive. The old TempDB file will have to be manually deleted from the original drive, because SQL Server will not do it itself.

Now that TempDB is on the right drive, expand it to the full size according to your preferences and then create additional TempDB files using the rule explained above (related with the number of processor cores). If you have got a quad-socket, quad-core box, i.e. 16 cores, it is recommended to use 4 to 8 TempDB files.

To create one additional TempDB data file it is necessary to run the command below:

```
USE [master]
GO
ALTER DATABASE [tempdb] ADD FILE (NAME = 'tempdev2', FILENAME='T:\MSSQL\DATA\tempdev2.ndf', SIZE =
10GB, FILEGROWTH = 0)
GO
```

It is important to highlight the fact that file growth has not been enabled. You should proactively create the TempDB files at their full sizes in order to avoid drive fragmentation. If you have a dual-cpu quad-core server (8 cores total) and an 80GB array for TempDB data, you should create eight 10GB files for TempDB. As a result, each file will be contiguous. If you create them as smaller files and let them grow automatically, then the disk will be fragmented because the files will be growing at random times. In addition, you could end up with differently sized TempDB files if one of them happened to grow faster than the rest. That is why we strongly recommend that all the TempDB files are pre-grown beforehand and then get them at exactly the right size.



26 Additional required components

In this section you have instructions on setup and configuration of the additional required components that must be running for a successful installation and operation of Critical Manufacturing MES.

- [ClickHouse](#)
- [Kafka](#)
- [RabbitMQ](#)
- [S3](#)



27 ClickHouse Connection Setup

In this section you can see information on specific settings that must be set in your system to allow connection and operation with ClickHouse:

Table 16: ClickHouse connection setup

Item	Description
ClickHouse Username	The ClickHouse User name
ClickHouse Password	The ClickHouse User password

See more information on [\[\[installation-guide-accountsandsecurity\]\]](#) for specific information on accounts and security settings for ClickHouse.

27.1 Deployment

Critical Manufacturing MES uses ClickHouse for data persistence. For the MES system to operate correctly, deploy **ClickHouse-Server** with the appropriate version matching the definitions listed in [\[\[system-requirements-additionalcomponents\]\]](#). You can use the provided ClickHouse templates available here: [ClickHouse Templates](#).

!!! warning These templates are available for guidance purposes only. Critical Manufacturing recommends consulting an expert for proper ClickHouse setup.

Furthermore, at least one user must exist with the permissions listed in [\[\[installation-guide-accountsandsecurity\]\]](#).

If you are using mTLS as the authentication method, a secret is required in the namespace where ClickHouse will be installed. This secret should contain the following key-value pairs:

Key	Value
<code>root.crt</code>	(Cluster Certificate Authority)
<code>client.crt</code>	(User Public Key)
<code>client.key</code>	(User Private Key)

The name of this secret should be passed to `values.yaml`.

!!! warning If you are using SASL Plain as the authentication method, it is necessary to pass the username and password to `values.yaml` instead of user certificates.

27.1.1 Fully managed ClickHouse service

It is also possible to use a fully managed ClickHouse service such as ClickHouse Cloud. If this is the setup you wish to deploy in your system, please contact ClickHouse on [ClickHouse Support](#) for detailed instructions on setting up ClickHouse and integrating with Confluent Cloud.



28 Kafka Connection Setup

In this section you can see information on what you must prepare to configure the connection to Kafka.

28.1 Authentication Methods

There are different required settings depending on the authentication method selected. Both are listed in the tables below:

28.1.1 Using Mutual TLS authentication

Table 18: Kafka connection setup using Mutual TLS authentication

Item	Description	Planning and installation guide	Checked
Cluster Certificate Authority	The Kafka cluster certificate authority	This will be used to connect to Kafka when using mutual TLS	[[installation-guide-accountsandsecurity]]
User public key	The kafka user public key	This will be used to connect to Kafka when using mutual TLS	[[installation-guide-accountsandsecurity]]
User private key	The kafka user private key	This will be used to connect to Kafka when using mutual TLS	[[installation-guide-accountsandsecurity]]

28.1.2 Using SASL SSL Plain

Table 19: Kafka connection setup using SASL SSL Plain

Item	Description	Planning and installation guide	Checked
Kafka Username	The Kafka User name	This will be used to connect to Kafka when using SASL SSL Plain	[[installation-guide-accountsandsecurity]]
Kafka Password	The Kafka User password	This will be used to connect to Kafka when using SASL SSL Plain	[[installation-guide-accountsandsecurity]]

28.2 Deployment

Critical Manufacturing MES uses Kafka for event streaming. To ensure proper functionality, Kafka version 3.7.0 or higher is required.

As stated above, Critical Manufacturing MES provides two forms of authentication to communicate with Kafka:

- Mutual TLS (client certificates)
- SASL Plain (username and password).

!!! info For an adequate operation connection setup for Kafka under Critical Manufacturing MES, the user must have the permissions described in [[installation-guide-accountsandsecurity]].



When running MES on OpenShift, we recommend using Red Hat Streams for Apache Kafka version 3.7.0 or higher. Please contact Red Hat at [Red Hat Support](#) for detailed instructions on setting up AMQ Streams.

It is also possible to use Confluent Cloud. Please contact Confluent at [Confluent Support](#) for detailed instructions on setting up Kafka.



29 RabbitMQ Connection Setup

In this section you can see information on specific settings that must be set in your system to allow connection and operation with RabbitMQ:

From the official [RabbitMQ](#) website:

RabbitMQ is a reliable and mature messaging and streaming broker, which is easy to deploy on cloud environments, on-premises, and on your local machine. It is currently used by millions worldwide.

Table 20: RabbitMQ connection setup

Item	Description
Host	The fully qualified domain name of the machine where RabbitMQ will be running
Port	The port where RabbitMQ will be running
Use Client Certificates	Whether to use client-signed digital certificates for authenticated requests
RabbitMQ Username	The RabbitMQ User name
RabbitMQ Password	The RabbitMQ User password
Certificate PEM	Certificate containing the public key for authentication
Key PEM	Key used for authentication
CA PEM	Certificate authority in PEM (Privacy Enhanced Mail) format



30 S3 Connection Setup

In this section you can see information on specific settings that must be set in your system to allow connection and operation with Amazon Web Services Simple Storage Service (S3):

From the official [Amazon S3](#) website:

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can use Amazon S3 to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides management features so that you can optimize, organize, and configure access to your data to meet your specific business, organizational, and compliance requirements.

Table 21: S3 connection setup

Item	Description
Address	The fully qualified domain name of the machine where S3 will be running (in <code>address:port</code> format)
Bucket name	Name of the bucket used for storage in S3
AccessKey Id	Access Key Id used to sign the requests sent to S3
Secret Access Key	Secret Access Key used to sign the requests sent to S3
Use Path Style	Whether to use virtual-hosted style or path-style requests to S3



31 Application Servers

This guide will walk you through the steps to successfully plan and install an application server to host the optional components used by Critical Manufacturing MES.

31.1 Planning for disaster recovery and high-availability

Critical Manufacturing recommends adopting the Windows Server Failover Clustering solution to provide high-availability of the roles required for Critical Manufacturing normal operation, including a scenario where Connect IoT is present and requires high-availability.

31.2 Application Server pre-requisites

Table 22: Application Server pre-requisites

Name	Requisite	Supported Versions	Checked
Operating System	Microsoft Windows Server	2016 or above	
Other software requirements	Microsoft Powershell Shell	5.1	
	.NET Framework	4.8	
	.NET	6	

!!! note Required by the Critical Manufacturing Deployment Framework to install or upgrade the solution

The deployment software will also install the following components automatically (if not present):

Table 23: Application Server components

Name	Feature Code (Get-WindowsFeature cmdlet)
Web Server (IIS)	Web-Server
Dynamic Content Compression	Web-Dyn-Compression
Basic Authentication	Web-Basic-Auth
Windows Authentication	Web-Windows-Auth
.NET Extensibility 4.8	Web-Net-Ext45
ASPNET 4.8	Web-Asp-Net45
WebSocket Protocol	Web-WebSockets
Management Tools	Web-Mgmt-Tools
ASPNET 4.8	NET-Framework-45-ASPNET
IIS Management Scripts and Tools	Web-Scripting-Tools
Management Service	Web-Mgmt-Service



31.3 .Net Framework Installation

Critical Manufacturing requires that all Application Servers have Microsoft .Net Framework 4.8 installed. .Net Framework is Critical Manufacturing's application runtime for optional components that are located outside of the main container stack.

This section contains detailed instructions on how to install .Net Framework.

- Get .Net Framework 4.8 installation package by downloading it from this [link](#).
- Install .Net Framework 4.8 downloaded in the previous step.
- After installing .Net 4.8, it might be necessary to register ASPNET applications with Internet Information Services (IIS). Please refer to this [article](#).



32 Application Clients

See the System Requirements section to learn more about this topic.

- [Client Devices](#)



33 File Shares

Information on file shares and volume configurations is available on the Customer Portal support website [here](#).



34 Email

This guide will walk you through the steps required to plan and prepare an email infrastructure to be used by Critical Manufacturing MES.

34.1 Email configuration

To configure the email infrastructure in Critical Manufacturing MES, the following information needs to be collected on the email service provider:

- SMTP Server Address
- SMTP Server User Name
- SMTP Server Password

!!! note Critical Manufacturing also supports using transport security (SSL) when sending emails. In this case, besides providing the https address in this setting, you need to set EnableSSL to ON.

When sending an email, Critical Manufacturing MES will identify itself through the information in these settings:

- Support Email Address
- Support Email From Name

All these settings will be requested in the installation wizard.



35 Installation

Installing Critical Manufacturing MES is a streamlined process carried out directly through the [Critical Manufacturing DevOps Center](#), available from the [Critical Manufacturing Customer Portal](#). This environment provides a centralized, guided experience designed to help you deploy, configure, and maintain your MES landscape with confidence.

In the section below, you'll find a step-by-step overview of how to perform a full installation of the core MES platform. Additional details on optional modules and complementary components are also provided, allowing you to tailor your installation to the needs of your operation.

!!! info If you're looking for broader information on prerequisites, architecture, or infrastructure-related topics, be sure to explore the comprehensive [DevOps Center documentation](#), which offers deeper insights into setup, best practices, and system administration.

In this guide you are going to create an MES Customer Environment in an existing Customer Infrastructure with an Infrastructure Agent already deployed and connected to the DevOps Center. Let's consider the Infrastructure Agent with version 11.0.4. This example is configured for an OpenShift v4.18 cluster.

1. Load the **Environments section** in the main page of the Customer Infrastructure and select **Create**. This opens a transaction wizard.

The screenshot shows the 'Customer Infrastructure' management interface. The 'Environments' section is expanded, and the 'Create' button is highlighted with a red box. Below the button is a table of existing environments.

NAME	DESCRIPTION	STATUS	TYPE	SITE	DEPLOYMENT PACKAGE	MODIFIED ON
MES_10_1_OS test		DeploymentSucceeded	Development	CMF - PT	MES 10.1.0	07/26/2023 07:04 PM
ProductDevQA		DeploymentSucceeded	Development	CMF - PT	MES 10.1.0	07/27/2023 10:03 PM
ProductDevQAApps		NotDeployed	Development	CMF - PT	MES 10.1.0	07/31/2023 01:46 PM

Figure 50: create_env_button

2. Set a **Name**, a **Type** and a **Site**. Select **Create**.



+ Create Customer Environment

GENERAL DATA

General Data

Name: ProductDevQA

Description:

* Type: Development

* Site: CMF - PT

Comments:

Cancel Create

Figure 51: create_action

3. The Customer Environment is now created in the system and ready to be used. The current page should now be its installation view where it is possible to start configuring. The first step group is named **Target** and is related with the configurations' base package and version, opt-in features and the deployment target.

1. In the first step, named **Package**, you can define the base package and version along with other metadata. Set the following values and select **Next**:

- **Deployment Package** - base package to be used. Since you are installing a Critical Manufacturing MES 11.0.4, set the value to **MES 11.0.4**.
- **Configuration Level** - different levels which will have an impact in how many configurations will be asked and are possible to configure. Set to **Advanced** so that you are able to configure everything that is currently possible.
- **License** - license for the Critical Manufacturing MES installation. Select an available license.

Package

* Deployment Package: MES 11.0.4

Configuration Level:

- Basic
- Intermediate
- Advanced

* License: CMF - PT_Critical Manufacturing_Development_v11.0.0_ED20250125

< Back Next >

Figure 52: step_target_package



2. Now configure the base package with opt-in features. These can vary based on the license modules and on the base package and version. The available ones for a Critical Manufacturing MES 11.0.4 are the following:

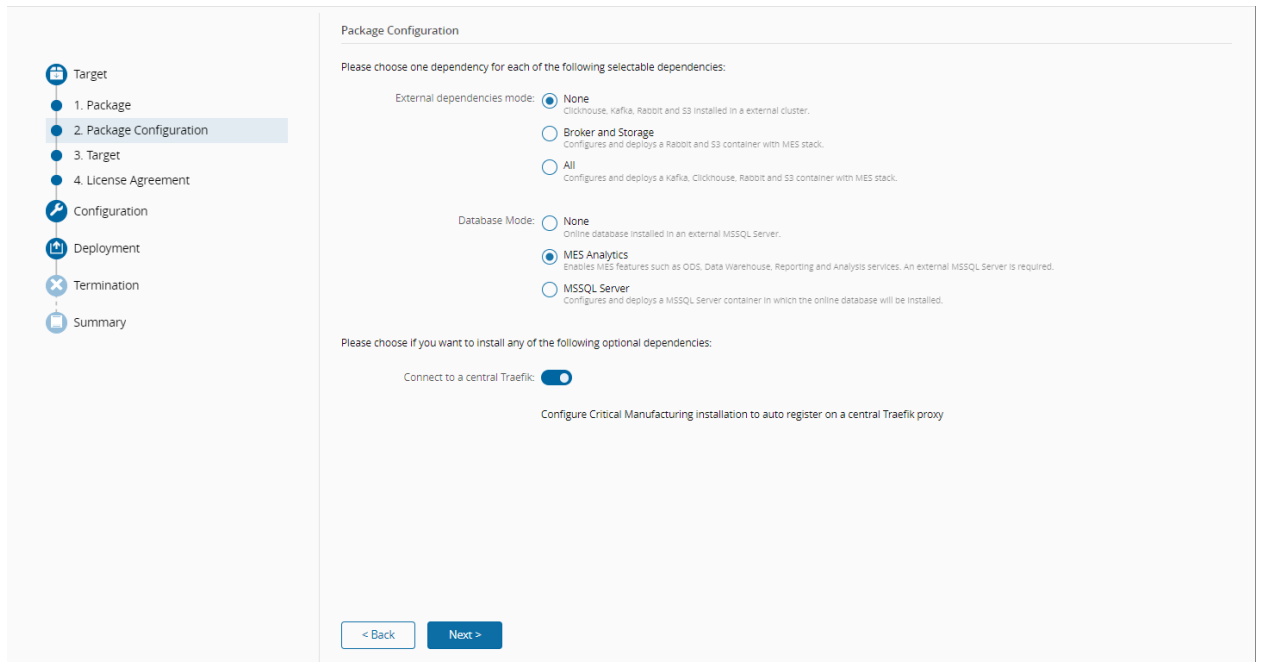


Figure 53: step_target_package_configuration

- **External dependencies mode** - different modes for ClickHouse, Kafka, RabbitMQ, and S3. These may be:
 - **None** - ClickHouse, Kafka, RabbitMQ, and S3 are not installed within the MES stack. These components must be available externally to the MES stack.
 - **Broker and Storage** - ClickHouse and Kafka are not installed within the MES stack, but RabbitMQ and S3 are installed within MES stack (not recommended for Staging or Production environments).
 - **All** - ClickHouse, Kafka, RabbitMQ, and S3 are installed within the MES stack (not supported in Staging or Production environments).
- **Database Mode** - different modes for the database installation. These may be:
 - **None** - base mode where only the Online database is installed in an external MSSQL Server.
 - **MES Analytics** - enhanced version of the previous mode where more features are available but an external MSSQL Server is still required. These features are: Operational Data Store (ODS), Data Warehouse (DWH), Reporting and Analysis Services.
 - **MSSQL Server** - similar to the None mode but instead of relying on an external MSSQL Server, a container is deployed along with the rest of the stack, which runs a containerized version of MSSQL Server.
- **Connect to a central Traefik** - configures the stack's Traefik reverse proxy to not be exposed and instead connect to the Traefik that is included in the Infrastructure Agent. This is very useful as many Customer Environments can be deployed and their only entrypoint is the Infrastructure Agent's Traefik reverse proxy, which means less open inbound ports and less networking configurations. When creating a Customer Environment in a Customer Infrastructure with an Infrastructure Agent, this option will be selected by default.

To exemplify the feature, keep the **Database mode** set to **MES Analytics**, the **External dependencies mode** set to **None** and the opt-in feature **Connect to a central Traefik** set to **true** since you have an Infrastructure Agent configured and running.

Set the values as shown in the example above and select **Next**.

3. Set the Target to **OpenShift Remote** as you are using OpenShift as your remote Deployment Target. Select **Next**.
4. Read and Understand all the licenses that are displayed and select **Next**, which advances to the next step group, **Configuration**.



4. By now, the base application and its features are set. Before starting the deployment process, you need to configure the base application and features with the required data. The following steps address the available configurations for each different part of the system. Bear in mind that the Configuration Level and License have an impact on how many steps and parameters are shown along with the available Critical Manufacturing MES features.

!!! note For configuration purposes, the character double quotes (") is not allowed to be inserted on input fields.

1. **General Data** - general information regarding the system.

• **Details**

- **System Name** - name of the system. Also used to set the database(s) name(s).
- **Tenant Name** - tenant name of the system. Must not contain blank spaces.

• **Access Information**

- **Application Public HTTP Address** - public HTTP address to access the environment. Remember that when using Domain Name System (DNS) providers or Transport Layer Security (TLS) via certificates, such as in the case of an Infrastructure Agent, this field must comply to their specifications, for example, **if configuring a wildcard certificate** for *.mydomain.com or using a Cloudflare for that same domain, **this field must be set to <mysubdomain>.mydomain.com**. Do not specify the HTTP Port, since the Agent's Traefik already has the ports 80 and 433 configured and these are the ones that are going to be used. For OpenShift, we should use its native routing mechanism which generally follows the cluster's certificate and domain as well instead of relying on DNS and TLS validation at the Infrastructure Agent or another level.
- **Application Public HTTP TLS Enabled** - set to **true** if this environment has TLS enabled. This is just for internal usage, the actual TLS configuration must be set via DNS providers or certificates.

• **Install Information**

- **Package to Install** - root package to be installed by the Environment Manager. If empty, it will use the default base package for the version. This can be set to override and install a different package, for example, a customization package.
- **Installation Data Path** - the path that the MSSQL Server can use to access the Installation Data volume. When using an external MSSQL Server, this volume must be a shared location between it and the containers.
- **Deployment Mode** - The deployment mode can be Transactional (if the installation fails, the database is guaranteed to be in a stable state and the installation can be retried) or it can be Non-Transactional (if the installation fails, the database is left in an intermediate state and it needs to be restored before retrying, **requiring downtime**. It is much faster than the Transactional mode).

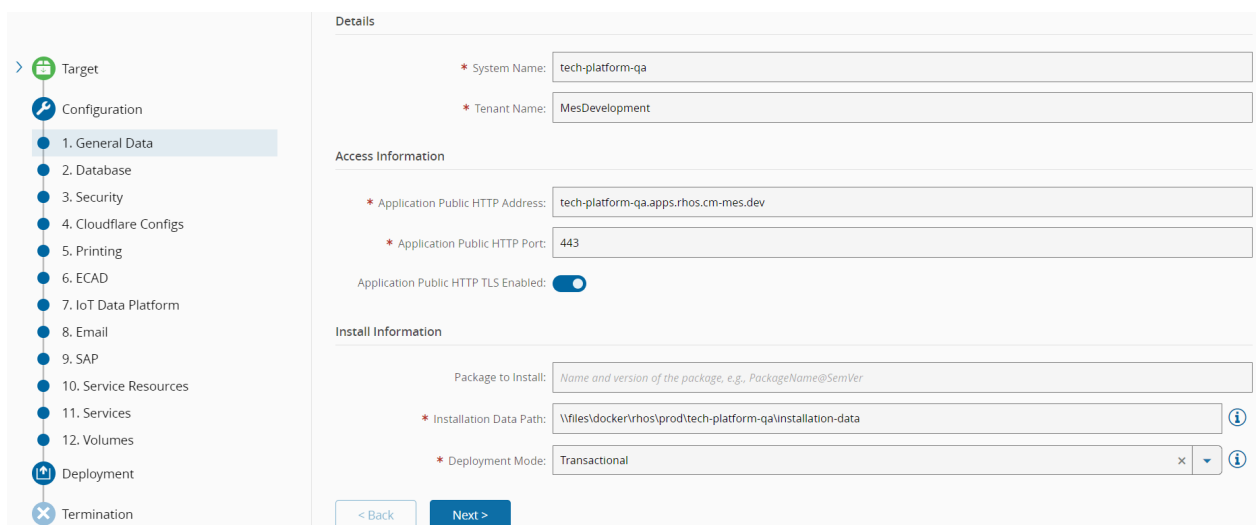


Figure 54: step_configuration_general_data



2. **Database** - information regarding database(s) connection(s).

- **Online** - online database information.
 - **Address** - database address. *Applies to Database Modes: None and MES Analytics.*
 - **Username** - database SA user. *Applies to Database Modes: None and MES Analytics.*
 - **Password** - database SA user password. **Generated when in database mode MSSQL Server.** *Applies to Database Modes: None, MES Analytics and MSSQL Server.*
 - **File Location** - location where the database files will be persisted. *Applies to Database Modes: None and MES Analytics.*
 - **Database Always On Enabled** - whether to enabled Always On mode. *Applies to Database Modes: None and MES Analytics.* **Feature not available with MSSQL Server database mode.**
 - **Port** - port to expose the database for remote access. *Applies to Database Modes: MSSQLServer.*
- **Online Data Store (ODS)** - ODS database information. *Applies to Database Modes: MES Analytics.*
 - **Address** - database address. If left blank, will default to the Online database value.
 - **Username** - database SA user. If left blank, will default to the Online database value.
 - **Password** - database SA user password. If left blank, will default to the Online database value.
 - **File Location** - location where database files will be persisted. If left blank, will default to the Online database value.
- **Data WareHouse (DWH)** - DWH database information. *Applies to Database Modes: MES Analytics.*
 - **Address** - database address. If left blank, will default to the Online database value.
 - **Username** - database SA user. If left blank, will default to the Online database value.
 - **Password** - database SA user password. If left blank, will default to the Online database value.
 - **File Location** - location where database files will be persisted. If left blank, will default to the Online database value.
- **Analysis Services (AS)** - AS database information. *Applies to Database Modes: MES Analytics.*
 - **Address** - database address.
 - **Username** - Windows authentication user.
 - **Password** - Windows authentication user password.

Example:

The screenshot shows a configuration form with two main sections: 'Online' and 'Online Data Store (ODS)'.
The 'Online' section includes:
- 'Address': VM-DEV-DB02.cmf.criticalmanufacturing.com\ONLINE
- 'Username': ExampleDbAdminUser
- 'Password': masked with dots and a lock icon
- 'File Location': E:\DATABASES
- 'Database Always On Enabled': a toggle switch that is turned on.
The 'Online Data Store (ODS)' section includes:
- 'Address': Microsoft SQL Online Data storage database server address, e.g., SQLSERVERINSTANCE
- 'Username': Microsoft SQL Online Data storage database username
- 'Password': masked with dots and a lock icon
- 'File Location': Microsoft SQL Online Data storage database file location

Figure 55: step_configuration_database_1



Data Warehouse (DWH)

Address:

Username:

Password:

File Location:

Analysis Services (AS)

* Address:

* Username:

* Password:

Figure 56: step_configuration_database_2

3. **Dependencies** - information regarding external MES dependencies.

- **ClickHouse** - ClickHouse information. *Applies to External dependencies mode: None/Broker and Storage*
 - **Address** - The hostname or IP address of the ClickHouse server.
 - **HTTP Port** - The HTTP port used by ClickHouse.
 - **TCP Port** - The TCP port used by ClickHouse.
 - **Username** - The username for ClickHouse authentication.
 - **Password** - The password for ClickHouse authentication.
 - **Validate certificates** - Toggle to enable or disable server certificate validation.
- **Kafka** - Kafka information. *Applies to External dependencies mode: None/Broker and Storage*
 - **Bootstrap Servers** - The Kafka bootstrap servers.
 - **Authentication Method** - The authentication method used by Kafka (None, mTLS, SASL_SSL Plain)
 - **Ssl Certificate Authority** - The certificate authority (CA) file for validating the Kafka server's certificate. *Applies to Authentication Method: mTLS and SASL_SSL Plain.*
 - **Ssl Certificate** - The public key certificate used for client authentication against Kafka. *Applies to Authentication Method: mTLS.*
 - **Ssl Key** - The private key certificate used for client authentication against Kafka. *Applies to Authentication Method: mTLS.*
 - **Validate certificates** - Toggle to enable or disable server certificate validation. *Applies to Authentication Method: mTLS.*
 - **Kafka Username** - The username for Kafka authentication. *Applies to Authentication Method: SASL_SSL Plain.*
 - **Kafka Password** - The password for Kafka authentication. *Applies to Authentication Method: SASL_SSL Plain.*
- **RabbitMQ** - RabbitMQ information. *Applies to External dependencies mode: None*
 - **Host** - The hostname or IP address of the RabbitMQ server.
 - **Port** - The port used by RabbitMQ.
 - **Virtual Host** - The RabbitMQ virtual host name.
 - **Username** - The username for RabbitMQ authentication.
 - **Password** - The password for RabbitMQ authentication.
 - **Use TLS** - Toggle to enable or disable TLS for RabbitMQ communication.
 - **Ssl Certificate** - The public key used for client authentication against RabbitMQ. *Applies to Use TLS: true.*
 - **Ssl Key** - The private key used for client authentication against RabbitMQ. *Applies to Use TLS: true.*
 - **Ssl Certificate Authority** - The certificate authority (CA) file for validating the RabbitMQ server's certificate. *Applies to Use TLS: true.*



- **Validate Certificate(s)** - Toggle to enable or disable server certificate validation. *Applies to Use TLS: true.*
- **External Storage (S3-compatible)** - S3 information. *Applies to External dependencies mode: None*
 - **Address** - The hostname or IP address of the S3-compatible storage service.
 - **Bucket Name** - The name of the S3 bucket to be used.
 - **AccessKey Id** - The access key ID for authenticating with the S3-compatible storage.
 - **Secret Access Key** - The secret access key for authenticating with the S3-compatible storage.
 - **Use Path Style** - Toggle to enable or disable path-style access for S3-compatible storage.

Example:

The screenshot shows two sections of a configuration form: ClickHouse and Kafka. The ClickHouse section includes fields for Address (clickhouse.apps.rhos.cm-mes.dev), HTTP Port (30443), TCP Port (30440), Username (default), and Password. It also has a 'Validate Certificates' toggle set to 'on'. The Kafka section includes fields for Bootstrap Servers (infra-kafka-kafka-tls-bootstrap-infra-kafka.apps.rhos.cm-mes.dev:443), Authentication Method (mTLS), Ssl Certificate Authority (kafkacapem), Ssl Certificate (kafkacertificatepem), and Ssl Key (kafkakeypem). It also has a 'Validate Certificates' toggle set to 'on'.

Figure 57: step_configuration_dependencies_1

The screenshot shows two sections of a configuration form: Rabbit MQ and External Storage (S3-compatible). The Rabbit MQ section includes fields for Host (vm-imp.cmf.criticalmanufacturing.com), Port (5672), Virtual Host (qa1), Username (CMFUser), and Password. It also has a 'Use TLS' toggle set to 'on' and fields for Ssl Certificate (RabbitCertificate), Ssl Key (RabbitKey), and Ssl Certificate Authority (RabbitCA). The 'Validate Certificate(s)' toggle is set to 'off'. The External Storage (S3-compatible) section includes fields for Address (http://vmdsdev01.cmf.criticalmanufacturing.com:9123), Bucket Name (testlibgenerator), AccessKey Id, and Secret Access Key. The 'Use Path Style' toggle is set to 'on'.

Figure 58: step_configuration_dependencies_2

4. **Security** - information regarding the Security Portal.

- **Domain**



- **Client Id** - System's Auth Client Id. Defaults to **MES** and cannot be changed.
- **Active Directory**
 - **Enable** - whether to enable the Active Directory authentication strategy.
 - **Domain** - default domain where user information is stored.
 - **Address** - AD address to connect.
 - **Base DN Address** - base search query.
 - **Username** - user to use for searching.
 - **Password** - user password to use for searching.
 - **Use SSL** - whether to use SSL.
 - **Port** - AD port to connect.

Example:

Active Directory

Enable:

* Domain:

* Address:

* Base DN Address:

* Username:

* Password:

Use SSL:

Port:

Figure 59: step_configuration_security_ad

- **WebAuthn**
 - **Enable** - whether to enable the WebAuthn authentication strategy.
 - **Open ID Connect**
 - **Enable** - whether to enable the OpenID Connect authentication strategy.
 - **Display Name** - The display name of the strategy in the Security Portal. Defaults to **OpenID**.
 - **Client ID** - ID of an existing OpenID provider's auth client.
 - **Metadata URL** - URL of the OpenID provider metadata.
 - **Session Options**
 - **Session Duration** - the duration while a session is still valid for authentication.
 - **Show Remain Signed In** - whether the option to remain signed in via a session strategy is shown to the user after a login.
 - **CORS (Cross-Origin Resources Sharing)**
 - **Allowed Origins** - When set, will configure some client containers, such as the UI, Help and Security Portal, with the domains specified here, effectively blocking cross-domain requests by the browser.
5. **Reporting Services** - Reporting Services access information. *Applies to Database Modes: None and MES Analytics.*
- **Reporting Services**
 - **Web Portal URL** - URL of the MSSQL Reporting Services Web Portal.
 - **Web Service URL** - URL of the MSSQL Reporting Services Web Service.
 - **Username** - user with read and write access for the MSSQL Reporting Services.
 - **Password** - user password.



Reporting Services

* Web Portal URL:	<input type="text" value="http://vm-dev-db02.cmf.criticalmanufacturing.com/Reports"/>
* Web Service URL:	<input type="text" value="http://vm-dev-db02.cmf.criticalmanufacturing.com/ReportServer"/>
* Username:	<input type="text" value="CMFExample-user"/>
* Password:	<input type="password" value="....."/>

Figure 60: step_configuration_reports

6. **Cloudflare Configs** - Cloudflare configuration to be used to create a subdomain for the current Customer Environment. Since we're not using Cloudflare to configure DNS, skip this step.
7. **Printing** - access information of the printing component service or to use a CUPS server. Currently, the service only works in Windows and it is not deployed in a containerized stack. **It must be installed using the Deployment Framework in a Windows machine.**
 - **Access Information**
 - **Use CUPS** - Set to true if using a CUPS server.
 - **Printing Service URL** - URL to a printing service running in a Windows server. *Available if Use CUPS is set to false.*
 - **CUPS URL** - URL of a CUPS server. *Available if Use CUPS is set to true.*

Access Information

Use CUPS:	<input checked="" type="checkbox"/>
Printing Service URL:	<input type="text" value="http://vm-prod-ci.cmf.criticalmanufacturing.com:5124"/>

Figure 61: step_configuration_printing

8. **ECAD** - access information of the ECAD component service. Currently, this service only works in Windows and it is not deployed in a containerized stack. **It must be installed using the Deployment Framework in a Windows machine.**
 - **Access Information**
 - **HTTP Address** - HTTP address to connect to the ECAD service.
 - **HTTP Port** - HTTP port to connect to the ECAD service.

Access Information

HTTP Address:	<input type="text" value="VM-PROD-CI.cmf.criticalmanufacturing.com"/>
HTTP Port:	<input type="text" value="49167"/>

Figure 62: step_configuration_ecad

9. **IoT Data Platform** - configurations for the IoT Data Platform:
 - **Event Strategy**
 - **Event Handler Strategy** - either Kafka itself or an SQL Queue.
10. **Email** - email server information for the system used by several features.
 - **Email Server**
 - **From** - email address to be used to send emails.
 - **Address** - email server address.



- **Port** - email server port.
- **TLS Enabled** - whether the email server is configured with TLS or not.
- **Username** - email server user.
- **Password** - password of the email server user.
- **Support Email Address** - email address of the local support team to send emails to.

Email Server

From: test.support@criticalmanufacturing.com

Address: criticalmanufacturing-com.mail.protection.outlook.com

Port: 25

TLS enabled:

Username: navigo.support

Password:

Support Email Address: test.support@criticalmanufacturing.com

Figure 63: step_configuration_email

11. **SAP** - ERP SAP connection information:

• **Access Information**

- **Enabled** - whether the connection to a SAP system is enabled.
- **Address** - address of the SAP host.
- **System Number** - SAP system number.
- **Service Name** - name of the gateway in SAP.
- **Program ID** - SAP Program ID.
- **Username** - user to connect to the SAP system.
- **Password** - user password to connect to the SAP system.
- **Client Number** - SAP Client Number.
- **Connect License** - theobald ERP connect license.

Example:

Access Information

Enabled:

Address: vm-sap.cmf.criticalmanufacturing.com

System Number: 00

Service Name: sapgw00

Program ID: ERPTTEST

Username: developer

Password:

Client Number: 001

Connect License: 050HX2DEFP

Figure 64: step_configuration_sap

12. **Service Resources** - Definition of resources used by stack. You can define the **memory** (GB/GBi) and **CPU** (number of virtual cores) needed to deploy the specific container and maximum available to it. Also, you can specify the number



of **replicas** to be deployed. Note that, in the upper right corner, there is a button to restore these configurations to their default values, as defined in the deployment package manifest, depicted in the image below.

Figure 65: step_configuration_serviceresources

13. **Services** - generic stack-wide configurations that vary depending on the base deployment package and deployment target.

- **Environment Manager**

- **Stop Installation on External Components Validation Failure** - Stops installation if validation against external components fails.

- **DNS** - allows to set custom DNS domains to use for resolving host names. **It's recommended to use Fully Qualified Domain Names (FQDN) everywhere** instead of short names and configuring this setting since it will impact the performance of hostname resolving and may even lead to unexpected issues.

- **Container Image Registry Override** - this setting can be used to override the image registry used to pull container images from. Useful in the case that a private registry is preferred.

- **Custom Certificates** - allows adding new certificates to the running containers. It is possible to add more than one certificate, and all of them will be injected into the containers that accept this feature. Before starting the deployment, a secret for each certificate must be created with the content of the certificate (same logic as external secrets). It is necessary to insert the name of the created secrets in this field. **If there are proxies performing SSL Inspection on the network traffic, the respective certificate should also be added through this feature.**

Figure 66: step_configuration_services_cacerts

14. **Volumes** - configuration for each required volumes. The entries here depend on the deployment package, database mode and opt-in features.



- **Cube** - repository for cube models. *Applies to Database Modes: MES Analytics*
- **Dagster** - repository for Dagster Data files. *Applies to Database Modes: MES Analytics*
- **ML Platform Agent** - repository to store the ML models binaries that are deployed and in use.
- **ML Platform Training** - repository to store the ML models' configurations and other associated data such as CSV datasets, transformed data and binaries.
- **Redis Data Folder** - repository for the Redis Data files. Recommended to be a local high-performance disk.
- **MSSQL Server Data** - volume to persist the database files to. Currently, MSSQL Server does not support restoring a database when persisting the data to a Windows directory. Ensure that the environment is running on Linux. *Applies to Database Modes: MSSQL Server.* Recommended to be a local high-performance disk.
- **Grafana Folder** - repository to persist the Grafana's data.
- **Installation Data** - shared location between the MSSQL Server and the Environment Manager. **When using an external database, the shared path must point to the same location as the Installation Data Path setting.**
- **Connect IoT Repository Share** - repository for Connect IoT compressed files.
- **Rabbit Data Folder** - repository for RabbitMQ Data files. *Applies to External dependencies mode: All/Broker and Storage*
- **Rabbit Log Folder** - repository for RabbitMQ Log files. *Applies to External dependencies mode: All/Broker and Storage*
- **Storage Data Folder** - repository for Storage Data files. *Applies to External dependencies mode: All/Broker and Storage*
- **Kafka Data Folder** - repository for Kafka Data files. *Applies to External dependencies mode: All.* Recommended to be a local high-performance disk.
- **ClickHouse Data Folder** - repository for ClickHouse Data files. *Applies to External dependencies mode: All.*
- **ClickHouse Log Folder** - repository for ClickHouse Data files. *Applies to External dependencies mode: All.*
- **Documents Folder** - location where the Critical Manufacturing MES documents and attachments are persisted to. Advised to be a shared location, so that when having more replicas of the Critical Manufacturing MES host, the containers maintain data consistency.
- **Logs Folder** - location to persist logs as files. This is an optional volume, you can set this volume to the type *None* in order to not use it.

Each volume can be configured with a different volume type. These types are Deployment Package specific. For more information on each type and their configurations, see [Kubernetes Volumes Configuration](#) documentation. Also, check the requirements for each volume [System Requirements](#).

For local volumes, it's recommended to use dynamic provisioning so that all local paths are handled by the cluster and not by you. For more information, see [Local Volumes with dynamic provisioning](#).

5. Selecting **Next** will trigger the deployment process. It is automatic and you are provided with feedback to follow during the installation.



1 Starting deployment to target OpenShift Remote...
 2 Connecting to the cluster...
 3 Connection established!

Figure 67: step_deploy_connect_to_cluster

```

1629 2023-07-27 21:43:55.0694 | Info | Integration Entry processed: 6a314fb4-2d01-4500-ade8-cf8abf299702-UI/1.CheckOut_Employee_Step.xml |
1630 2023-07-27 21:43:55.0694 | Info | Integration Entry processed: 3a172944-b81b-4eed-9f01-31da1f088005-UI/2.Check_Out_Employees_Wizard.xml |
1631
1632 2023-07-27 21:43:56.0774 | Info | Integration Entry processed: fd41089b-18d9-4b40-9516-0a98635bd606-UI/2.CheckOut_Employee_Wizard.xml |
1633
1634 2023-07-27 21:43:57.0862 | Info | Integration Entry processed: ef0833f2-2e0f-47b1-90e8-5e6c59e0b827-UI/Cmf_Inner_OEE_Page.xml |
1635 2023-07-27 21:43:57.0862 | Info | Integration Entry processed: 37ed641f-27ae-40d3-9808-dde10f249808-UI/Cmf_Overall_Equipment_Effectiveness.xml |
1636
1637 2023-07-27 21:43:58.1487 | Info | Integration Entry processed: 21f4b07e-7571-4b8e-bf72-7c6cca8dce88-UI/Cmf_Process_KPI.xml |
1638 2023-07-27 21:43:58.1487 | Info | Integration Entry processed: 3bd70b67-5670-498c-b24c-497c2ce768c5-UI/Cmf_Resources_KPI.xml |
1639
1640 2023-07-27 21:44:00.1644 | Info | Integration Entry processed:
89ca7c3b-1a27-4c31-94e3-2544df5726ab-ProcessRules/01-SetupAppFrameworkSystemSecurity.cs |
1641
1642 2023-07-27 21:44:07.2825 | Info | Integration Entry processed:
0a2d630d-2bf4-4a71-a9d2-230110de352d-assets/ie3JobSupervisorAutomationController.xml |
1643
1644 2023-07-27 21:44:08.2887 | Info | Integration Entry processed: 36c1603e-ad55-4099-8232-c5107080cb57-DEEs/EnableConnectIoTSite.cs |
1645
1646 2023-07-27 21:44:13.3276 | Info | Integration Entry processed: aeffa29c-f9dc-4c4c-8b66-cf8c17adc814-MasterData/Calendar.xlsx |
1647
1648 2023-07-27 21:47:22.0986 | Info | Integration Entry processed: 0bc5180d-7092-4048-914a-e964e44cd0e8-MasterData/MasterDataSample.xlsx |
1649
1650 2023-07-27 21:47:54.7693 | Info | Integration Entry processed: c73a2795-6cf1-4cdd-a2f5-9f9c843572a4-DEEs/SyncUsersWithAD.cs |
1651
1652 2023-07-27 21:47:55.7886 | Info | Integration Entry processed: 4d8a0fd-3a99-4ed5-89b2-e039beb2cb5e-DEEs/AddEmailToTestUsers.cs |
1653
1654 2023-07-27 21:47:56.7899 | Info | Integration Entries processed |
1655 2023-07-27 21:47:56.7920 | Info | Installation completed. |
1656 2023-07-27 21:47:56.7948 | Info | There are no packages to cache |
1657 2023-07-27 21:47:56.7948 | Info | Log file location: "/var/log/cmf/envmanager" |
1658 Installation finished. Please check the environment @ https://devqa.apps.vmrhosdsclt1.cmf.criticalmanufacturing.com:443/
  
```

< Back Next >

Figure 68: step_deploy_check_installation

Name: ProductDevQA
 Description:
 Status: DeploymentSucceeded
 Target: OpenShift Remote
 Administrator Username: admin
 Administrator Password: iv5d2R9Q08Af
 Environment: https://devqa.apps.vmrhosdsclt1.cmf.criticalmanufacturing.com:443/

Figure 69: step_summary



If everything is correct, the last screen should present you with the deployment Summary, which includes information such as the outcome, admin credentials to use to access the environment, and the URL.

Accessing this URL should present the Critical Manufacturing MES to be used. In this case, you must log in and because this is the first login, the credentials in the Summary must be used to access the system and the password must be reset after logging in:

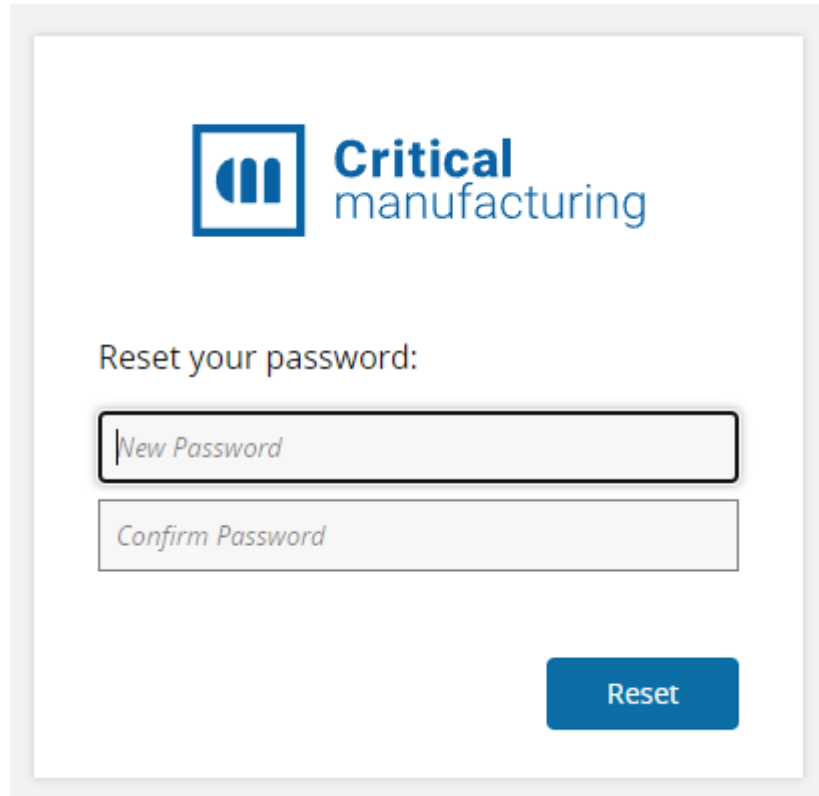


Figure 70: access_mes_reset_password_on_first_login

After resetting the password, you'll be redirected to the Home Page which will show there are no Apps installed:

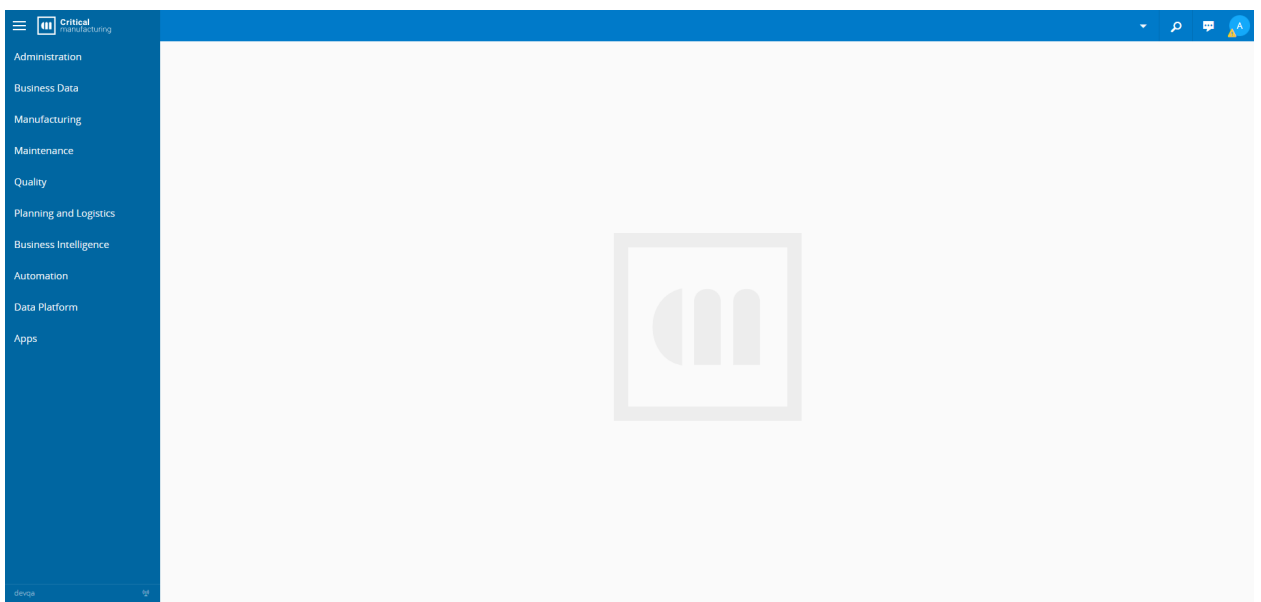


Figure 71: home_page



!!! info At this stage, only Administrator users have access to the MES. To allow other users to access it, you need to assign them the MES **OAuth** Role in the security page:

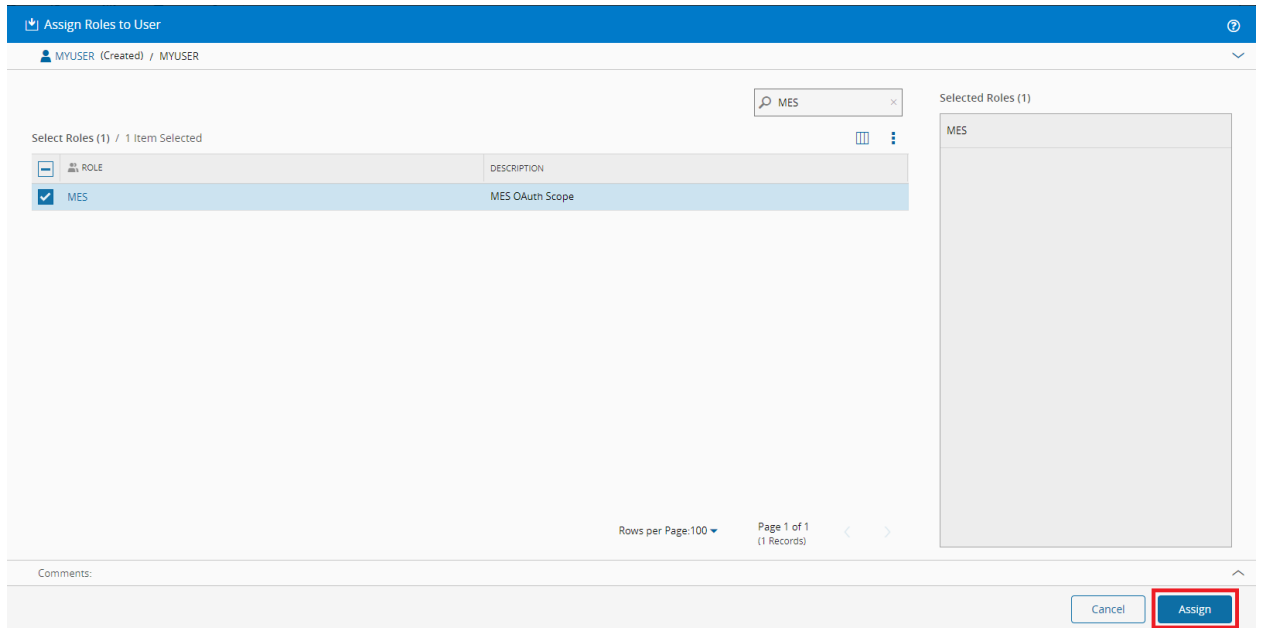


Figure 72: assign_mes_oauth_role

35.1 Optional Component Installation

The Critical Manufacturing MES installation procedure is supported by an installation wizard that is described step-by-step in this section. Depending on whether the installation is performed with or without Internet access, and on the chosen installation package, the setup's interface may present changes. Therefore, you may notice slight differences between the screenshots included in this guide and the version being installed.

!!! note Random errors may occur during the Critical Manufacturing MES installation process if it is initialized using a blocked ISO file. This occurs when Windows Attachment Manager marks ISO files as blocked (more information [here](#)).

Before mounting and starting the installation, execute the following procedure to unblock your ISO file:

1. Open the folder containing the ISO file on Windows Explorer.
2. Right-click on the file and select the **Properties** option.
3. Select the **Unblock** option, if available.
4. Select the **Apply** button and then the **OK** button.

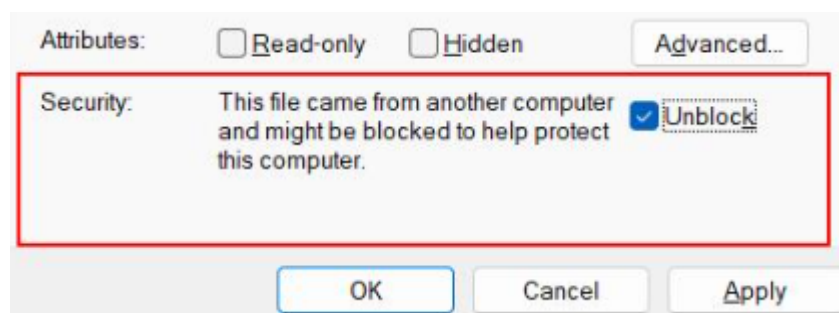


Figure 73: iso_properties



The image below shows the first screen of the setup wizard:

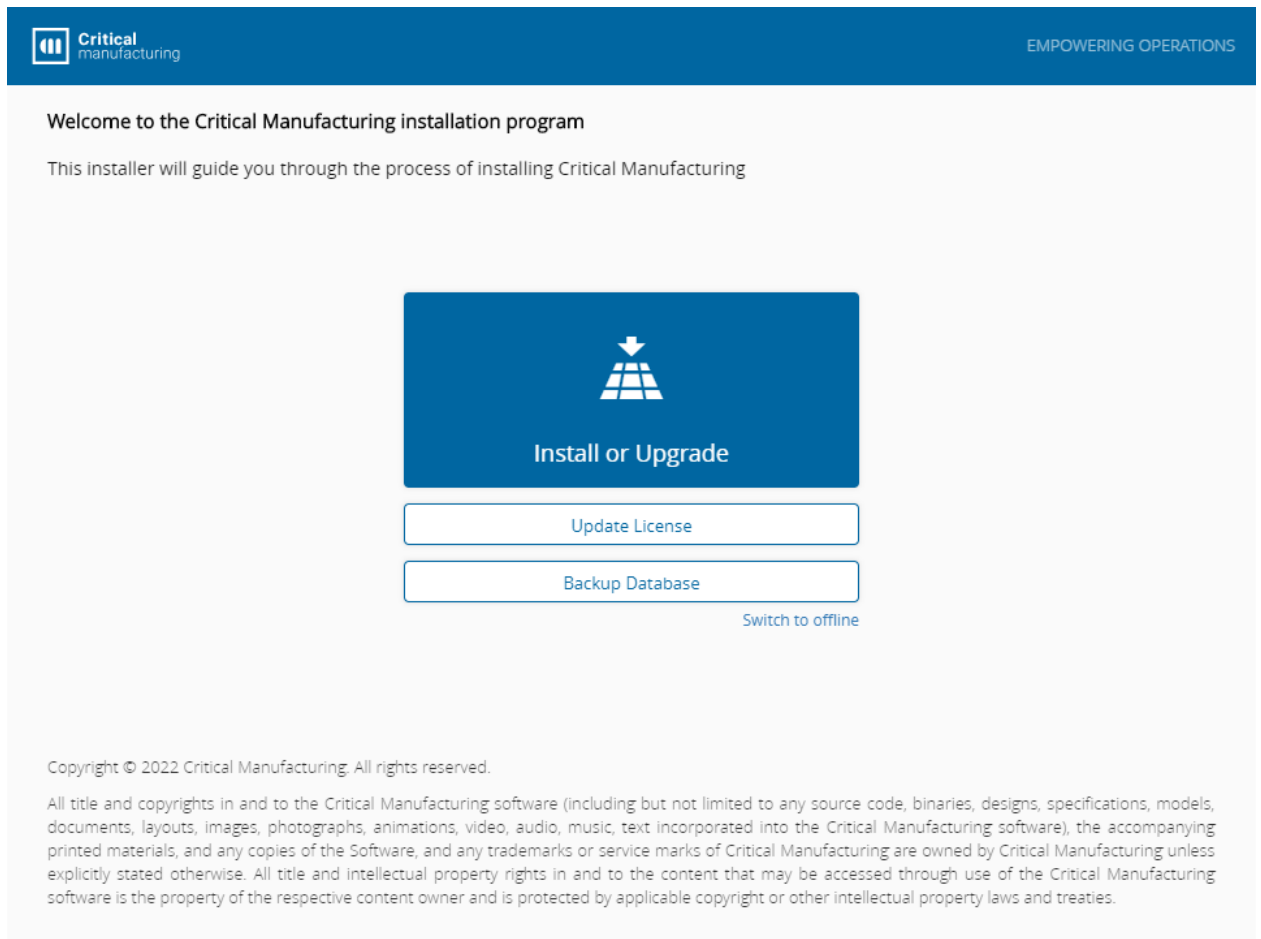


Figure 74: Installation - Welcome screen

All screens of the setup wizard are divided in three areas:

1. The top area displays all the steps of the installation, and the current step is highlighted. It also displays the name of the user who is logged in to the Customer Portal or **Offline**. If the setup process is launched in **Online** mode without previous valid user authentication for the current user in the Customer Portal, a separate browser tab window is automatically opened for proper user authentication.
2. The middle area shows the information setup step and the text boxes to be filled out with the required setup information.
3. The bottom area is the navigation area. In this navigation area, you can go to the next screen or return to the previous one. It is also possible to cancel the installation, thus aborting the setup. The **Install** button is only enabled when all the configurations are filled out, and the setup is then ready to start.

Completing the text boxes may be mandatory (flagged by { style="margin-top: 2px; color: #b83128;" }). Throughout the steps, the existing groups may have an associated **Validate** button which if selected, will check if the entered value is within the expected ranges.

See the example below:



Critical Manufacturing EMPOWERING OPERATIONS OFFLINE

✓ The values are valid

REGISTRATION — 13 PRESENTATION — 14 IOT DATA PLATFORM — 15 PRODUCT LICENSE — 16 SUMMARY — 17 COMPLETE INSTALLATION

⌵ SETTINGS

* Binding Port:

* Binding Host:

* Presentation Timeout:

Figure 75: Installation - Field Validation

35.1.1 Main Installation Process

If you need to backup existing databases, visit [Database Backup and Restore](#) for a specific guide for this process.

The first step of the installation process is the **License Agreement**. Select **Install or Upgrade** to get started.

35.1.2 License Agreement

In this step, you need to read and agree to the Critical Manufacturing License Agreement.



Critical Manufacturing EMPOWERING OPERATIONS
OFFLINE

Critical Manufacturing Installation

① LICENSE AGREEMENT — ② PACKAGE SOURCES — ③ PACKAGE SELECTION — ④ IMPORT INSTALLATION FILE — ⑤ SUMMARY — ⑥

SOFTWARE LICENSE AGREEMENT FOR CRITICAL MANUFACTURING S.A. (CRITICAL)

1. DEFINITIONS

1.1 "CRITICAL" shall mean Critical Manufacturing S.A., a company incorporated in Portugal.

1.2 "Licensed Software" shall mean the software identified on Appendix A which must be attached to this MSLA. The term Licensed Software is understood to specifically include any and all Licensed Software documentation but specifically does not include open-source components.

1.3 "Third-Party Software" shall mean software developed and owned by an entity other than Critical Manufacturing S.A., which is used as an integral part of the Licensed Software.

1.4 Test system shall mean an installed instance of the Licensed Software which is strictly used for testing clean installations or upgrades prior to implementing into a licensed production environment.

2. OWNERSHIP

2.1 The foregoing license gives Licensee a limited permit to use the Licensed Software. CRITICAL retains all rights, title and interest, including all copyright and intellectual property rights, in and to, the Licensed Software. All rights not specifically granted in this MSLA are reserved by CRITICAL.

2.2 Third Party Software licenses shall be owned by and licensed by the entity holding rights to said software separate from this MSLA.

3. LICENSE GRANTS

3.1 CRITICAL grants Licensee a non-exclusive, perpetual, non-transferable, without the right to grant sublicenses, limited license to use an object code copy of the Licensed Software based on the quantity, description, and limitations in Appendix A exclusively for Licensee's internal business purposes.

I agree to the License Agreement

Cancel < Back Next >

Figure 76: Installation - License Agreement review

It is necessary to accept the License Agreement to continue with the installation, otherwise the wizard will display an error message:



Critical Manufacturing EMPOWERING OPERATIONS
OFFLINE

Critical Manufacturing Installation

1 LICENSE AGREEMENT — 2 PACKAGE SOURCES — 3 PACKAGE SELECTION — 4 IMPORT INSTALLATION FILE — 5 SUMMARY — 6

herein. Failure to pay this one-time fee will result in shut down of the Licensed Software for which damages Critical shall not be liable.

7.2 Licensee is responsible for any taxes applicable to the license fee.

8. GENERAL

8.1 All disputes arising out of or in connection with the present contract shall be finally settled under the Rules of Arbitration of the International Chamber of Commerce by one or more arbitrators appointed in accordance with the said Rules. The place of arbitration shall be Austin, Texas, USA and the law applicable to the matters under dispute in the arbitration shall be USA substantive law. The language of the proceedings shall be English.

8.2 Future licenses may be purchased under the terms of this MSLA provided this agreement is referenced on the license quotation. Accordingly, this agreement may incorporate multiple attachments identified as Appendix A. Upon signature, each one shall be considered an original and together shall be considered the same agreement.

8.3 This MSLA and Appendix A contain the complete agreement between the parties with respect to the subject matter hereof, and supersedes all prior or contemporaneous agreements or understandings, whether oral or written. Licensee agrees that any varying or additional terms contained in any purchase order or other written notification or document issued by Licensee in relation to the Licensed Software licensed hereunder shall be of no effect. The failure or delay of CRITICAL to exercise any of its rights under this MSLA or upon any breach of this MSLA shall not be deemed a waiver of those rights or of the breach.

8.4 If any provision of this Agreement shall be held by a court of competent jurisdiction to be contrary to law, that provision will be enforced to the maximum extent permissible, and the remaining provisions of this Agreement will remain in full force and effect.

8.5 All questions concerning this MSLA shall be directed to: Critical Manufacturing S.A., Rua Eng. Frederico Ulrich, n. 2650, 4470-605 Moreira da Maia, Portugal.

You must accept the license agreement to continue the installation

Cancel < Back Next >

Figure 77: Installation - License Agreement acceptance

To accept the License Agreement select **I agree to the License Agreement**. Then, select **Next** to continue.

35.1.3 Package Sources

In the **Package Sources** step, you can add a location to search for installation packages. You can also add multiple package sources, and dependencies are resolved using the provided source order. If the setup is running in **Online** mode, the default process will add the local packages folder, and the server, to the list.

The package source can either be a **Server** or a **FileSystem**. The **Server** should be an NPM compatible endpoint. The **FileSystem** should be a folder accessible by the user running the setup, when the installation is started from the installation media, or the user running the master agent windows service is using a master only or master/slave configuration.

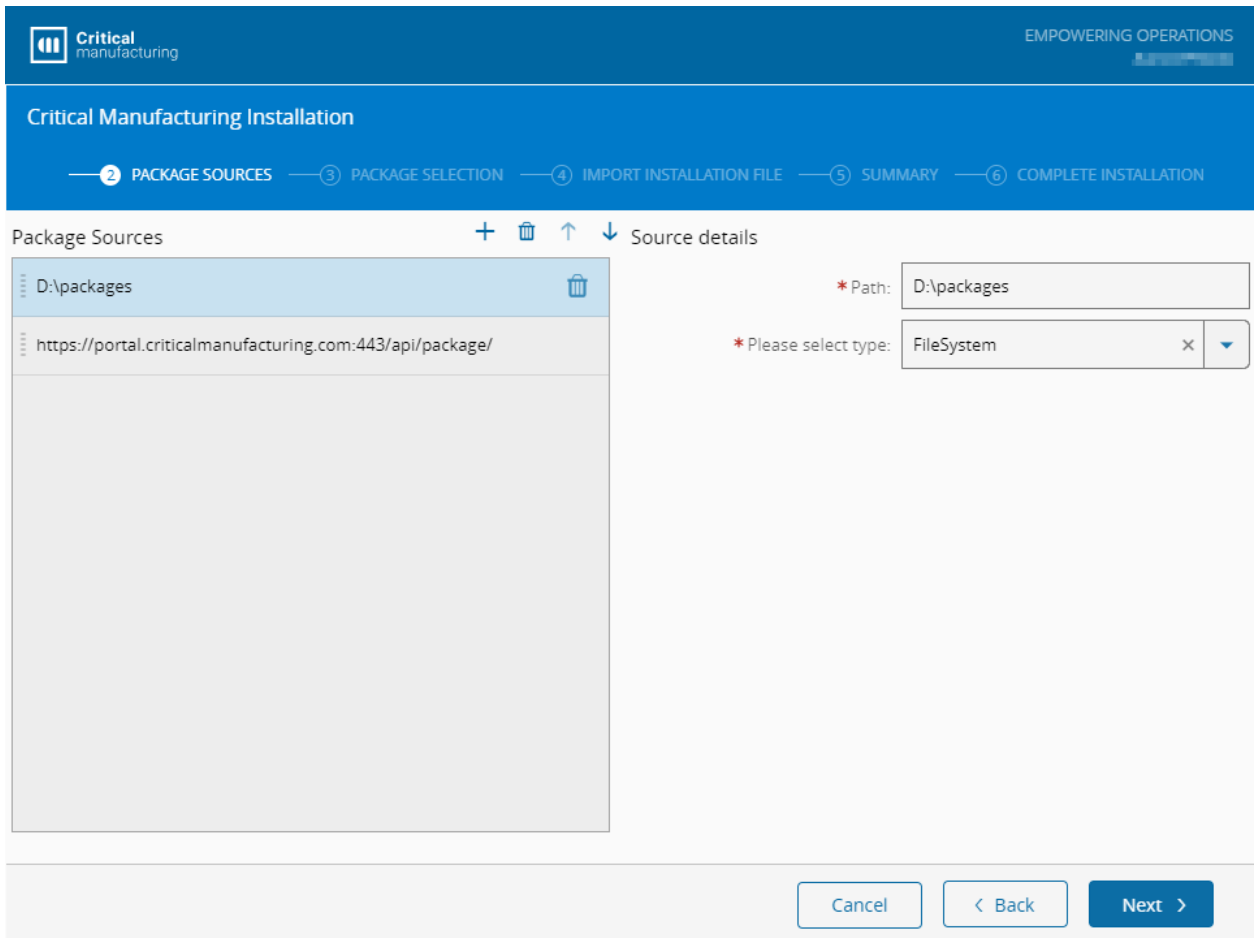


Figure 78: Installation - Package Sources

Select **Next** to continue.

35.1.4 Package Selection

In the **Package Selection** screen, you must select the product and the version to be installed. The available packages are:

- **Cmf.ConnectIoT.Packages:** Connect IoT packages to upload to a supported Package Repository.
- **Cmf.DataPlatform.Consumers:** Consumers to use with Critical Manufacturing IoT Data Platform. - remove
- **Cmf.ECADService.Server:** Critical Manufacturing MES ECAD (Electronic Computer-Aided Design) server to run as a service.
- **Cmf.PrintingService.Server:** Critical Manufacturing MES Printing Service.

For Critical Manufacturing Connect IoT packages there is only one available option. The product names, versions, and installation options can vary according to the installation packages available on the installation media.



Critical
manufacturing

EMPOWERING OPERATIONS
OFFLINE

Critical Manufacturing Installation

— 2 PACKAGE SOURCES — 3 PACKAGE SELECTION — 4 IMPORT INSTALLATION FILE — 5 SUMMARY — 6 COMPLETE INSTALLATION

Select the packages you want to install ↻

* Package:
Cmf.ECADService.Server × ▾

* Version:
11.0.0 × ▾

PACKAGE	VERSION
Cmf ECADService Server	11.0.0

Cancel < Back Next >

Figure 79: Installation - Package Selection

Choose the package you wish to install and select **Next** to continue.

ECAD Service

Before installing the **ECAD Service**, ensure that the Visual C++ Redistributable Packages for Visual Studio 2013 are installed. Afterwards, the **ECAD Service** can be installed using several different configurations, as shown in the image.



Critical Manufacturing EMPOWERING OPERATIONS OFFLINE

Critical Manufacturing Installation

PACKAGE SELECTION — 4 IMPORT INSTALLATION FILE — 5 ECAD SERVICE INSTALLATION — 6 SUMMARY — 7 COMPLETE INSTALLATION

INSTALLATION PATH

* Root Installation Directory:

CONFIGURATION

* System Name:

* Tenant Name:

ONLINE DATABASE

* Database Server:

Figure 80: ECAD Configuration.1

!!! info The **Import Installation File** step allows you to load a file with the configuration of the installation. It will automatically fill out the information existing in the selected file.

Configure the **Online Database** and **Services User Account** settings:



Critical Manufacturing EMPOWERING OPERATIONS
OFFLINE

Critical Manufacturing Installation

PACKAGE SELECTION — 4 IMPORT INSTALLATION FILE — 5 ECAD SERVICE INSTALLATION — 6 SUMMARY — 7 COMPLETE INSTALLATION

ONLINE DATABASE

* Database Server:

* Database User:

* Database User Password:

SERVICES USER ACCOUNT

* User Account:

* User Password:

ECAD SERVICE

Figure 81: ECAD Configuration.2

Configure the settings for the **ECAD Service**. The **PCBI Floating Service Address** should use the default **IPv4** address and the **PCBI Floating Service Port** can use any available port.



Critical Manufacturing EMPOWERING OPERATIONS OFFLINE

Critical Manufacturing Installation

PACKAGE SELECTION — 4 IMPORT INSTALLATION FILE — 5 ECAD SERVICE INSTALLATION — 6 SUMMARY — 7 COMPLETE INSTALLATION

ECAD SERVICE

* ECAD Service Address: localhost

* ECAD Service Port: 44304

ECAD Service Documents Path: *Example: C:\share\documents\permanent*

Validate

PCBI FLOATING SERVICE

PCBI Floating Service Address: 192.168.1.70

PCBI Floating Service Port: 5555

PCBI Floating Service License: *Example: C:\Environments\MES\Licences\PCBIFloatingServer.lic*

Validate

Cancel < Back Next >

Figure 82: ECAD Configuration.3

The current way to get the **PCBI Floating Service License** is by requesting a license file using the **Server ID**.

If you do not have the **PCBI Floating Service License** you can keep this field empty and follow the manual steps below (after the setup finishes).

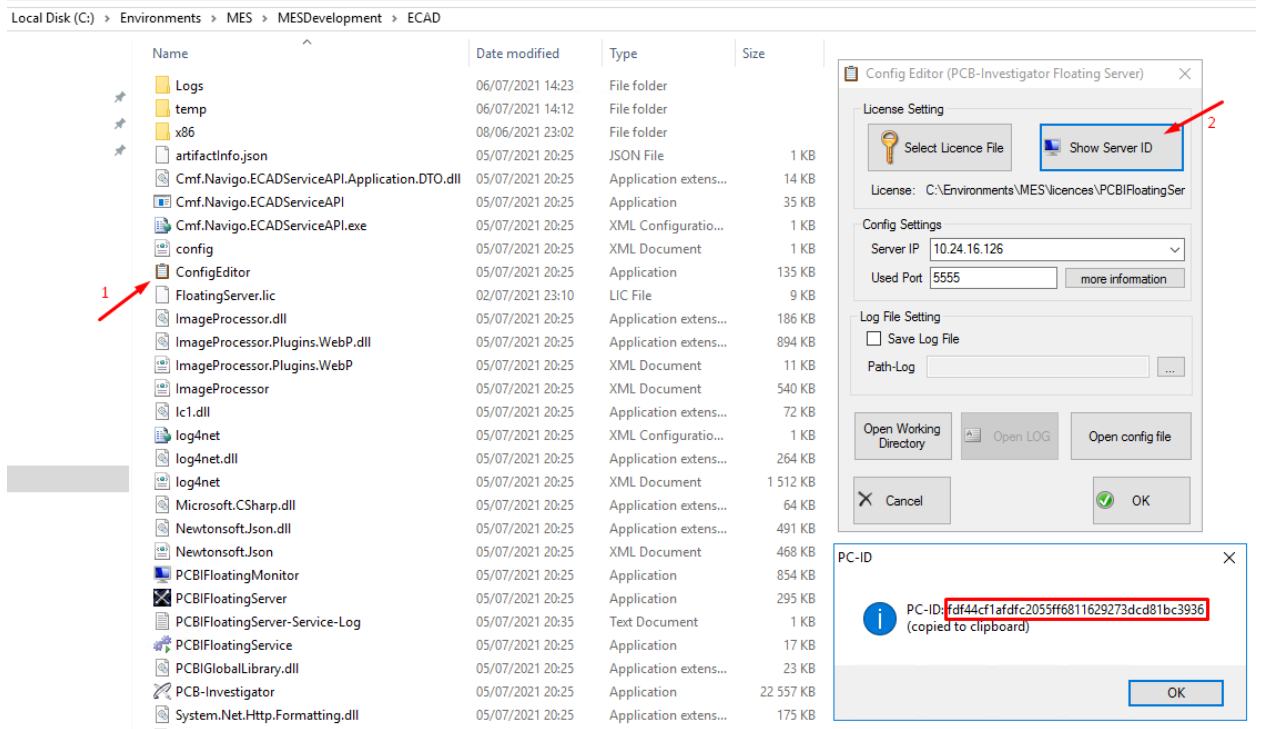
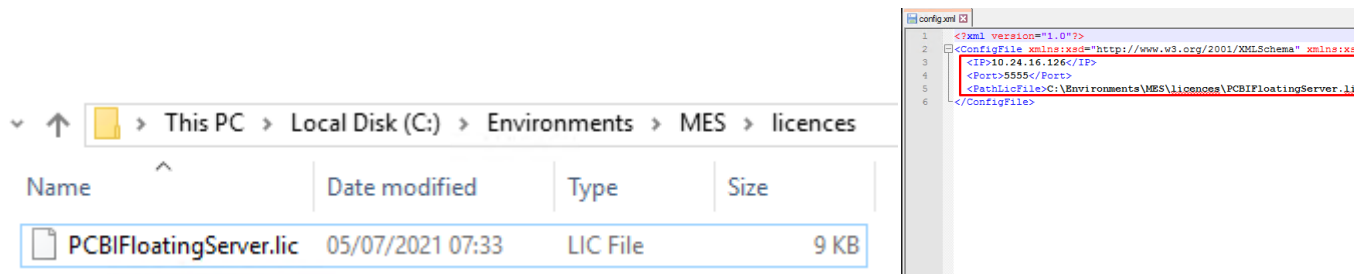


Figure 83: ECAD Configuration - PCBI Floating Service.2

Contact the System Administrator by sending the **Server ID** to generate the license.

When you get your license, place the file inside a folder and make sure that the configuration properties of **ECAD** are duly set. These configurations can be found inside the **ECAD** folder named **config.xml** and **Cmf.Navigo.ECADServiceAPI.exe.config**.



Restart the **PCBI Floating Server** and **Critical Manufacturing ECAD** services.

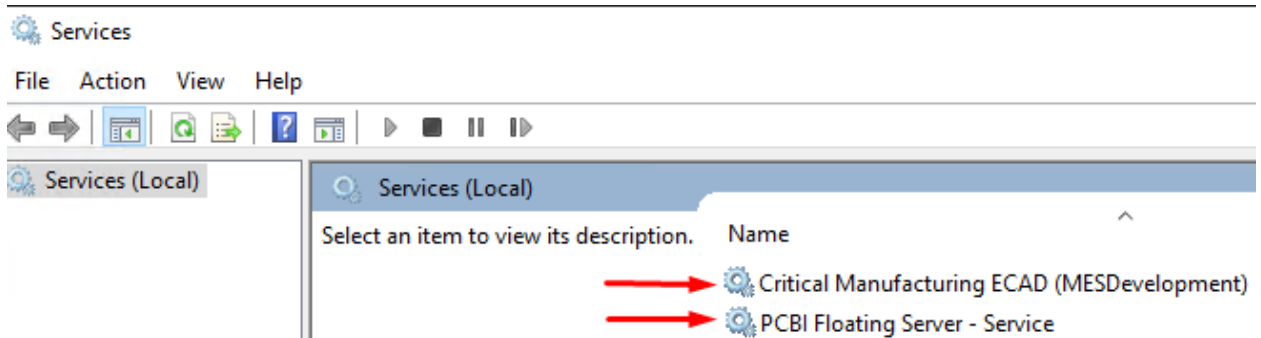


Figure 84: ECAD Configuration.6

!!! note If you have multiple environments in the same machine, all environments should point to the same **PCBI Floating Service** and only one **PCBI Floating Service** can be started.



Confirm the settings and select **Next**, where you will be shown the summary of the component you are about to install. You have the option to force a reinstallation if you have previously installed a version of this component in your system.

PACKAGE	VERSION
Cmf ECADService	11.0.0
Cmf ECADService Server	11.0.0

Force reinstallation of the currently installed version:

Cancel < Back Next >

Figure 85: ECAD Configuration Summary

Select **Next** to complete the installation process configuration.

In the **Complete Installation** screen, you can export all the current installation configuration data (including licenses) to a file. To export the current configuration data, select **Export** and then choose a location and a file name.

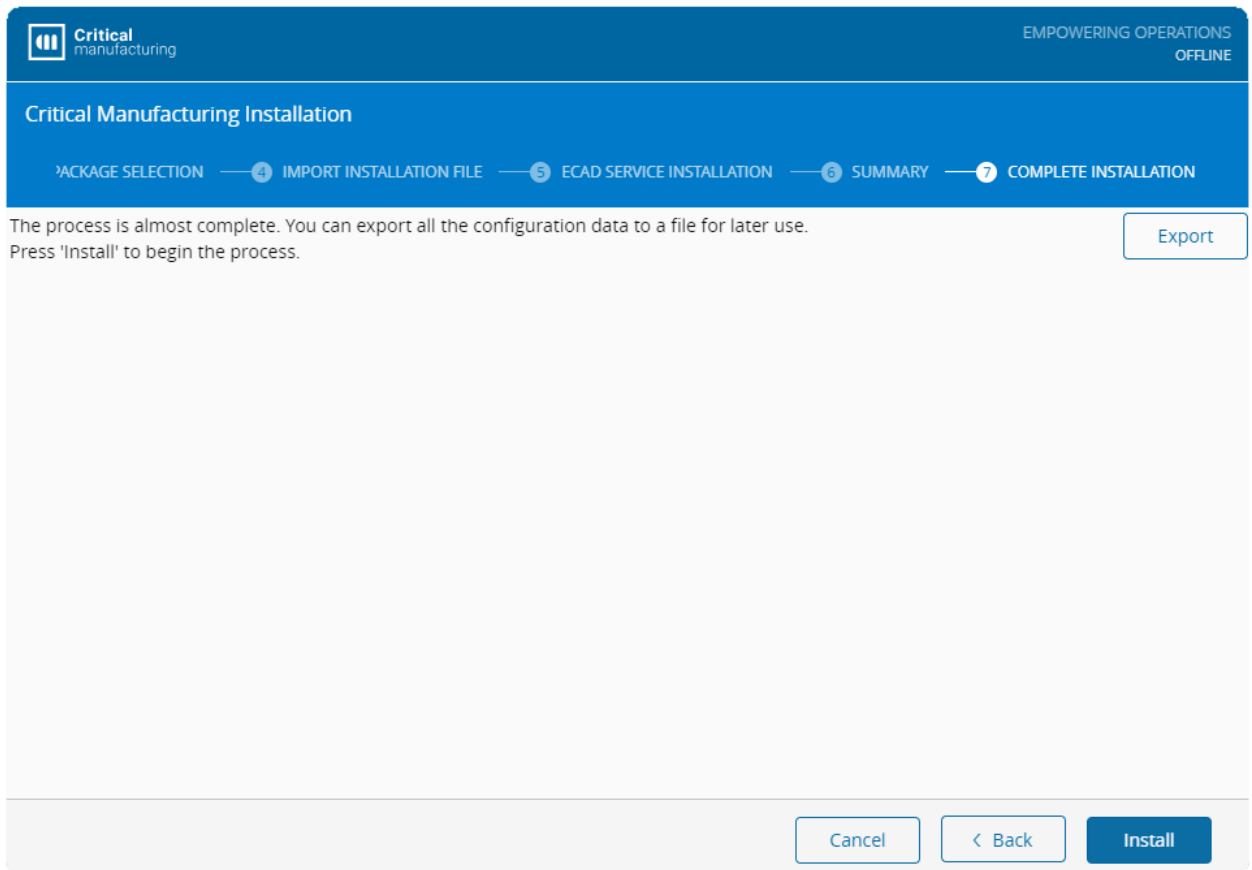


Figure 86: ECAD Installation Export

Select **Install** to start the installation process.

Printing Service

The Critical Manufacturing MES Printing Service is a standalone service that enables containerized environments to perform printing jobs using printers configured in a Windows machine.

To achieve this, the Printing Service must be installed in the Windows machine that has access to the printers (which may need permissions to be used by the user running the service). Additionally, when deploying the environment using DevOps Center, the user must select the option to use the external printing service and configure the URL to the Windows machine running the service.

!!! info This service is standalone and not coupled to a specific MES installation, which means that multiple MES installations can share the same Printing Service.

The Printing Service is installed using the same setup process as the traditional installation and selecting the **Cmf.PrintingService.Server** option in the Package Selection screen.



Critical Manufacturing EMPOWERING OPERATIONS OFFLINE

Critical Manufacturing Installation

— ③ PACKAGE SELECTION — ④ IMPORT INSTALLATION FILE — ⑤ ECAD SERVICE INSTALLATION — ⑥ SUMMARY — ⑦ COMPLETE INST

Select the packages you want to install ↻

* Package:
Cmf.PrintingService.Server x ▾

* Version:
11.0.0 x ▾

PACKAGE	VERSION
Cmf PrintingService Server	11.0.0

Cancel < Back Next >

Figure 87: Printing Selection

!!! info The **Import Installation File** step allows you to load a file with the configuration of the installation. It will automatically fill out the information existing in the selected file.

After selecting the Printing Service package, the user must fill (or import from a parameters file) the following information:

- **Root Installation Directory** - The Directory where the Printing Service will be installed.
- **Services User Account** - The user account that will be used to run the service.
- **Port** - The port where the service will be exposed.



Critical manufacturing EMPOWERING OPERATIONS
OFFLINE

Critical Manufacturing Installation

AGE SELECTION — 4 IMPORT INSTALLATION FILE — **5 PRINTING SERVICE INSTALLATION** — 6 SUMMARY — 7 COMPLETE INSTALLATION

INSTALLATION PATH

* Root Installation Directory: C:\Program Files\CriticalManufacturing

Validate

SERVICES USER ACCOUNT

* User Account: {{UserAccount}}

* User Password:

Validate

PRINTING SERVICE

* Port: 5124

Validate

Cancel < Back **Next >**

Figure 88: Printing Installation

Confirm the settings and select **Next**, where you will be shown the summary of the component you are about to install. You have the option to force a reinstallation if you have previously installed a version of this component in your system.



Critical Manufacturing EMPOWERING OPERATIONS OFFLINE

Critical Manufacturing Installation

AGE SELECTION — 4 IMPORT INSTALLATION FILE — 5 PRINTING SERVICE INSTALLATION — 6 SUMMARY — 7 COMPLETE INSTALLATION

PACKAGE	VERSION
Cmf PrintingService Server	11.0.0

Force reinstallation of the currently installed version:

[Cancel](#) [< Back](#) [Next >](#)

Figure 89: Printing Installation Summary

In the **Complete Installation** screen, you can export all the current installation configuration data (including licenses) to a file. To export the current configuration data, select **Export** and then choose a location and a file name.

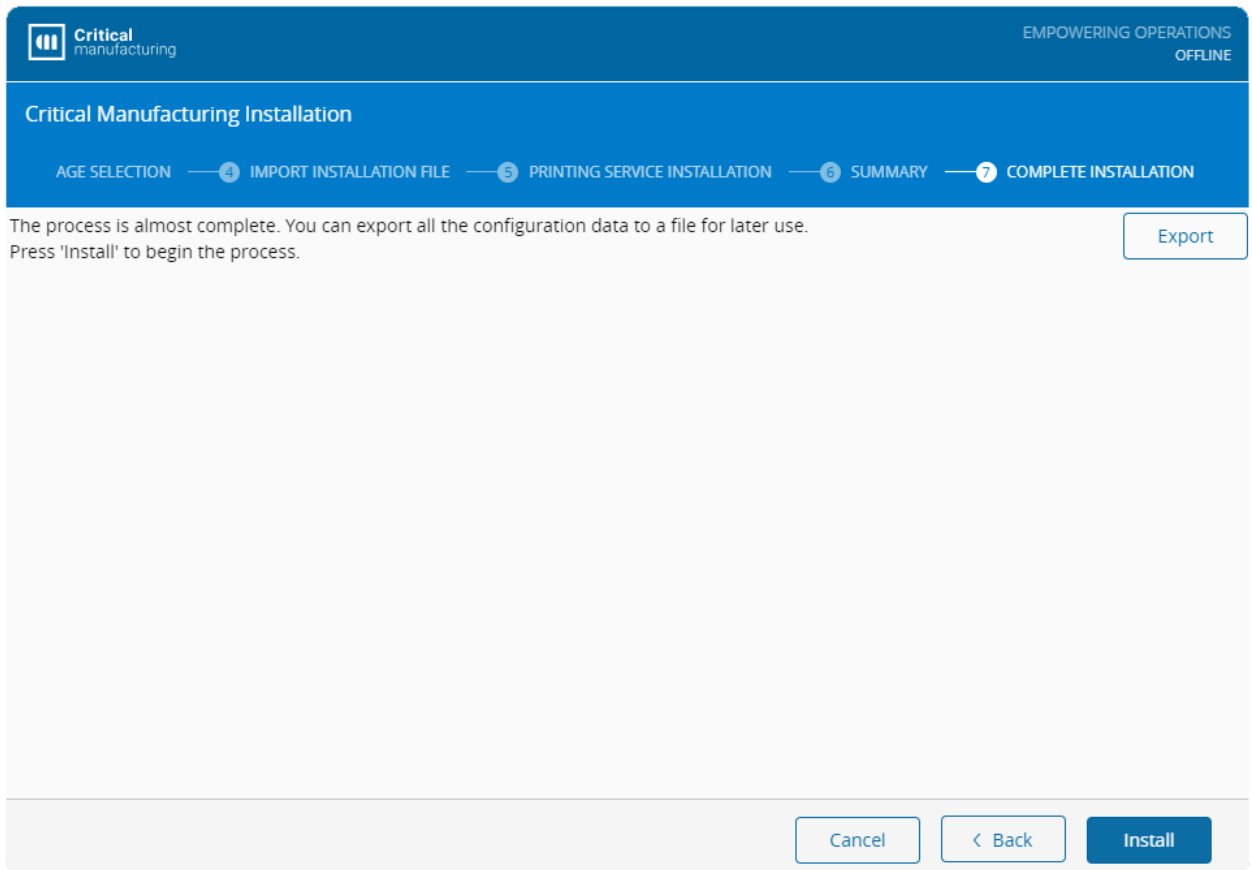


Figure 90: Printing Installation Export

Select **Install** to start the installation process.

35.1.5 Update Product License

The Product License can be updated using the Critical Manufacturing setup program in three ways:

- Setup - Online
- Setup - Offline
- Command Console

For the **Setup - Online**, follow the steps below:

1. Mount the Critical Manufacturing MES ISO.
2. Run the **Setup.exe**:

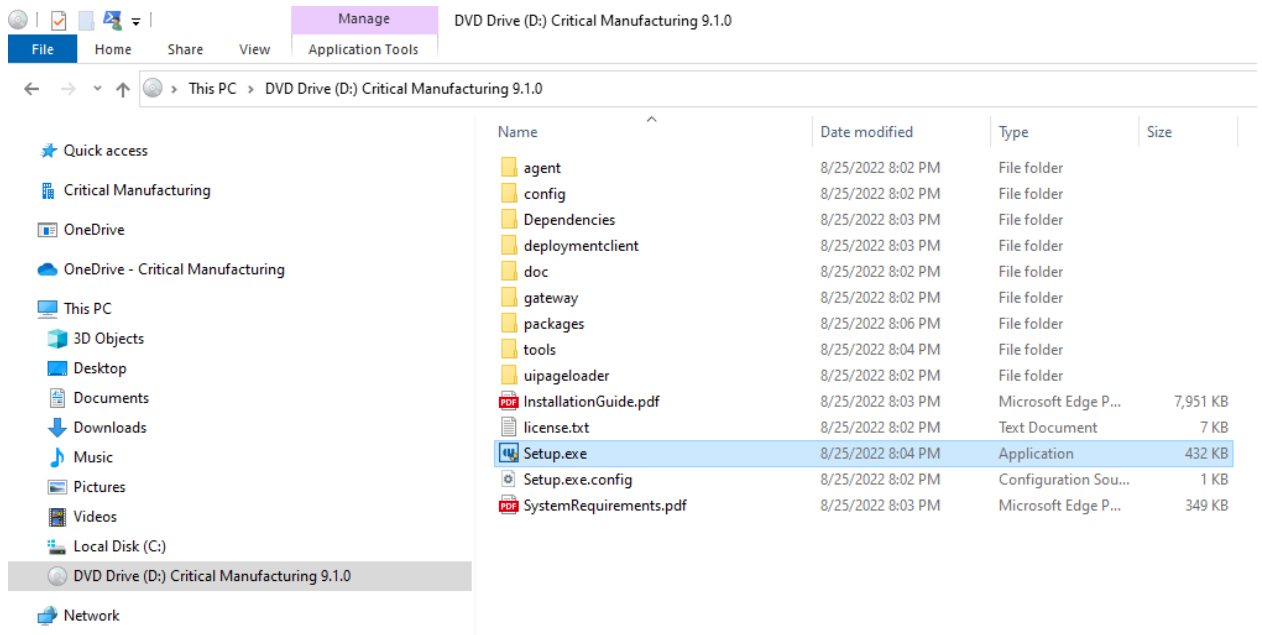


Figure 91: installation_setup

3. Select **Update License**:

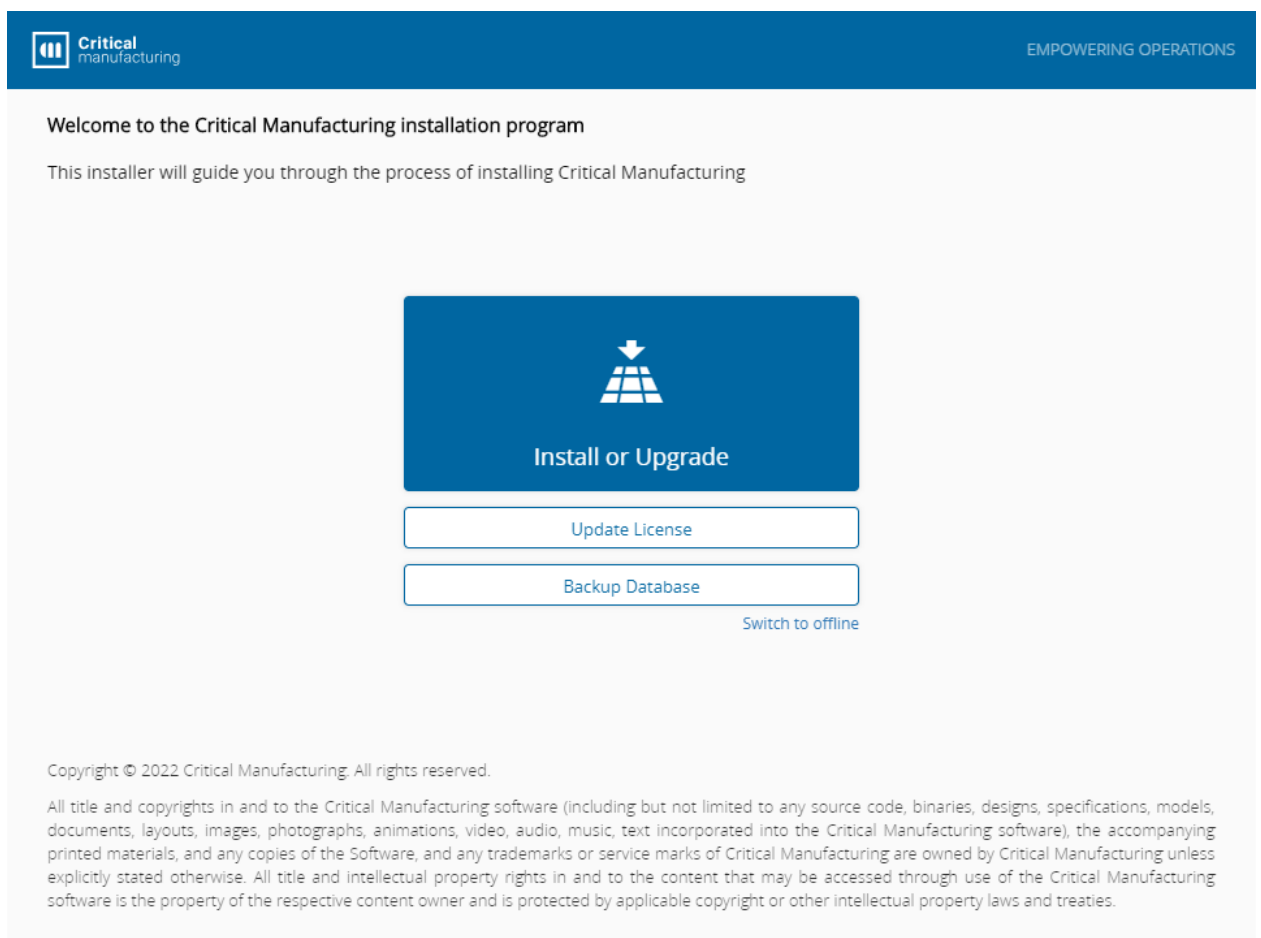


Figure 92: installation_update_license

4. You will be redirected to the Critical Manufacturing Customer Portal and will need to log in with a User that has access to



Critical Manufacturing Licenses:

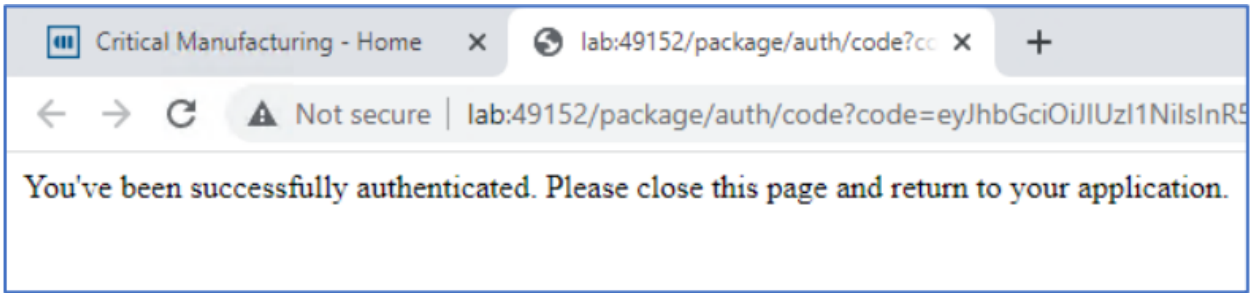


Figure 93: installation_authentication1

5. Import the `.json` parameters file used with the original installation:

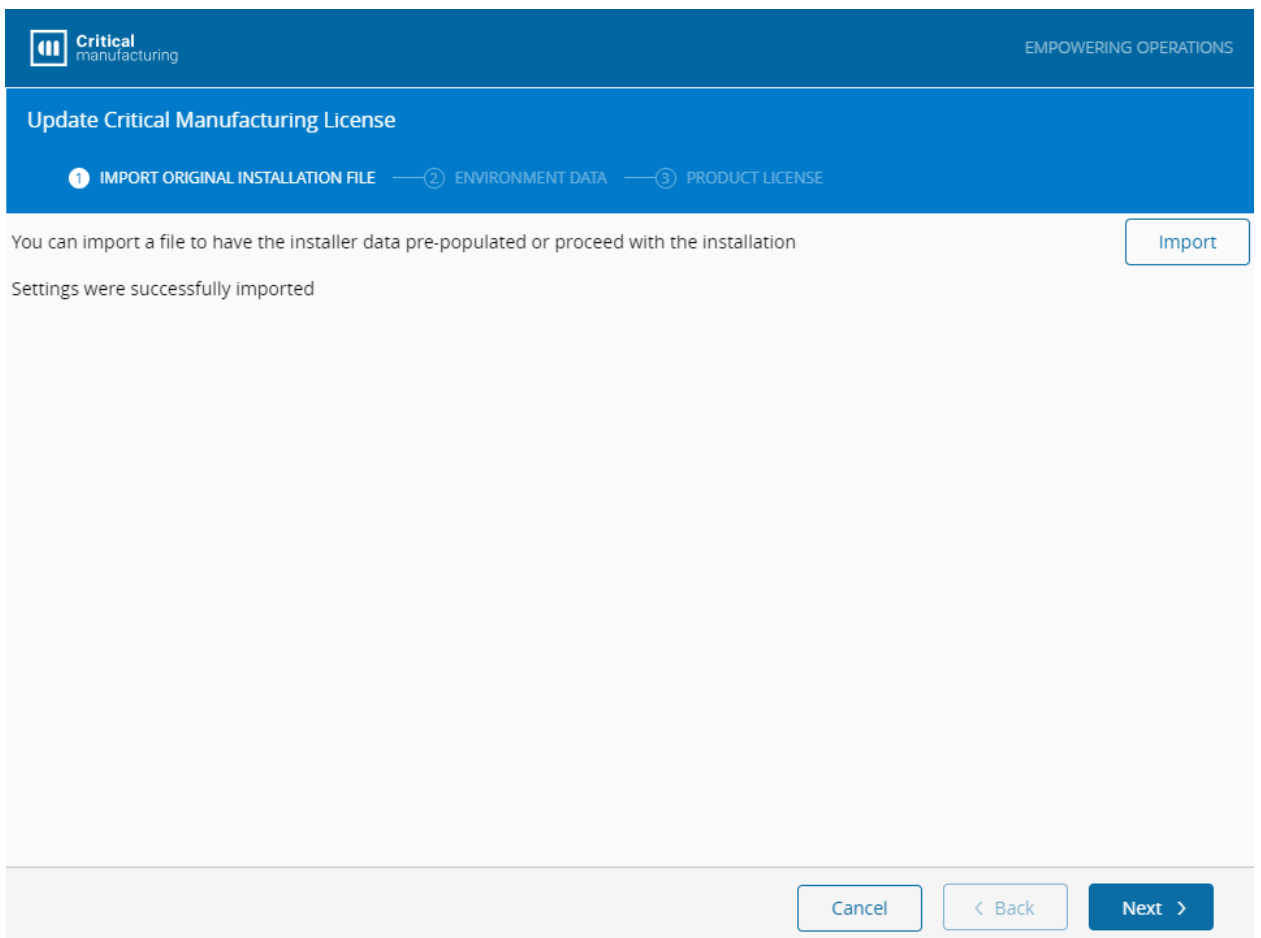


Figure 94: installation_import_parameter1

6. Select **Next** to continue.

7. The **Environment Data** screen will contain the **System Name** and the connection to the **Online DataBase**:



Critical manufacturing EMPOWERING OPERATIONS

Update Critical Manufacturing License

① IMPORT ORIGINAL INSTALLATION FILE — ② ENVIRONMENT DATA — ③ PRODUCT LICENSE

SYSTEM DATA

* System Name:

ONLINE DATABASE

* Database Server:

* Database User:

* Database User Password:

Figure 95: installation_import_parameter2

!!! note You can also set the parameters manually if you know the settings.

- 8. Select **Next** to continue.
- 9. Select the license you want to use:



Critical manufacturing EMPOWERING OPERATIONS

Update Critical Manufacturing License

1 IMPORT ORIGINAL INSTALLATION FILE — 2 ENVIRONMENT DATA — 3 PRODUCT LICENSE

Select the License to use for this installation

NAME	SITE	VERSION	EXPIRATION DATE	ACTIVATIONS	MODULES
DEVELOPMENT					
CMF - Training_Developr	CMF - Training	8.0.0	02/24/2023	55	29
CMF - Training_Developr	CMF - Training	9.0.0	02/24/2023	106	29

Rows per Page: 25 Page 1 of 1 (2 Records)

Cancel < Back Update

Figure 96: installation_license_select

10. Proceed with the **Update**:

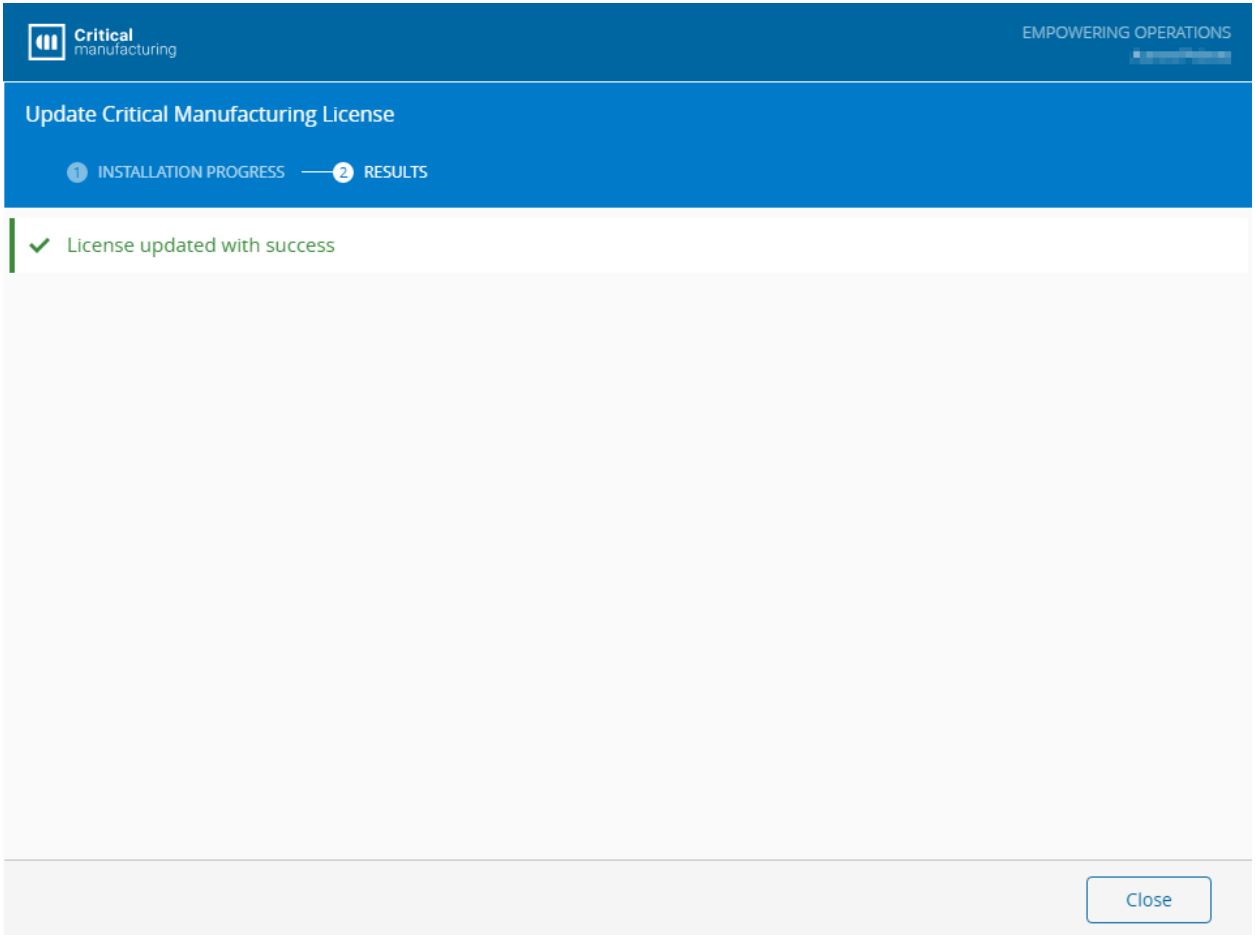


Figure 97: installation_license_success

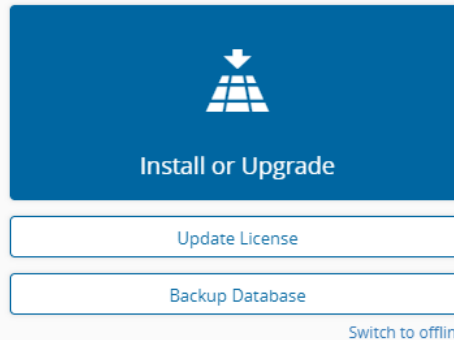
For the **Setup - Offline**, follow the steps below:

1. Before selecting **Update License**, select the **Switch to offline** option:



Welcome to the Critical Manufacturing installation program

This installer will guide you through the process of installing Critical Manufacturing



The screen displays three main options in a vertical stack: a large blue button with a white downward arrow icon and the text "Install or Upgrade"; a white button with a blue border and the text "Update License"; and another white button with a blue border and the text "Backup Database". Below these buttons, the text "Switch to offline" is visible.

Copyright © 2022 Critical Manufacturing. All rights reserved.

All title and copyrights in and to the Critical Manufacturing software (including but not limited to any source code, binaries, designs, specifications, models, documents, layouts, images, photographs, animations, video, audio, music, text incorporated into the Critical Manufacturing software), the accompanying printed materials, and any copies of the Software, and any trademarks or service marks of Critical Manufacturing are owned by Critical Manufacturing unless explicitly stated otherwise. All title and intellectual property rights in and to the content that may be accessed through use of the Critical Manufacturing software is the property of the respective content owner and is protected by applicable copyright or other intellectual property laws and treaties.

Figure 98: Installation - Welcome screen

2. After importing the `.json` parameters file used with the original installation or manually setting the **Environment Data**, an activation code is provided:



Critical manufacturing EMPOWERING OPERATIONS
OFFLINE

Update Critical Manufacturing License

1 IMPORT ORIGINAL INSTALLATION FILE — 2 ENVIRONMENT DATA — 3 PRODUCT LICENSE

Use the activation code shown below in Critical Manufacturing Portal to retrieve your product license. [Go to portal](#)

```
gIvTz3oWZxrgX5MBVfeE+8FG61OQhc0UH5Djjr6JdjkgT/RDR/WjheJ9A6znFVTq4EFb1fKBNEZcFGSMwU1u7q+i0oqAXRnTF2qF4EjtGXWmtocR6D1tgNQagzxZPMA
4LrCZG0U5kq0VcYme6BH8JhqjQK1ZmMQ1DKfP9ACY1hVwB+LUVSp4NWoeUVo8wNwKVoJAft1BaoIAL8DQaKjJjgbUj2e83jykEJvinPO2NUjJC/WNAQGT5TjTyQEGFK8
1oHrPDQ/DHsf2VseNg1TiLJLE+Bt+DbyM3/XLpgRCqSbZiIRVeBiAQUeS1r21R+GZ/jDmHWHoGFPQzwZQ54coFuw==4AWyNLN0WJ6Pm0AqhvTFh1gFxf6tntBBDDyMh1wrJ
hquQ19IifMnoGTOMPGC3uq2P8uFog+PJK21V3EY71zqFB3ps/6sfyHYGZm1p39uG6+sxgKK6xAa5Eb26AGfRChEMhAx391yRzZTIp9La6NKYwtu2PdNeARihWYtBYg+3
YnZM+r10U1J9I04eJnS5eRMnNLG9ghQbrYDVS1V/qr9+QuQ8RkRqAxNw7n4wjOb1DA7tsn4mgAYLYryvDjFTWBHmu4Rap326bukqSFA07HbuBqvKhdbcfX5303BPBMU
PibkDqrF218T8dMG5X1oTVDesfse0+jBOj/0ngov6n7++RN1cYcVcHINA+M2u97MLxDO0h/8r1JVtOUynb/2kL1C741k2M01jEM2b1jsvY26HAaY46VHWU5vPOIf/6W
conuEx74XRy7dsUa/ht6nw2P6XI82FINGiftOhxBWM7U6An1KR9NmLuN/aGJeT4VcSXVnWUEX1AQDQbBfTrg1aRFunhDAlzUhg024SJS5kbV2fSHTE9RHdyQEgWITr1
D0LQ+CbJz13VRenY7rsXYC2TBrzAB+vjevECJ5NEIbShi85ewMhYT0nL4c2OjVHjYBNZnnb3QqEU2Dc3/TuKrxNFZ2LLaeY2g2KJzrxGohz9Xga7FEXe7/Bs+4/n1xeaD
N57STdXELAsxwkE403JH1n3TI2hHsTQ3BT62GMfhYuyppqhGAX+shfw644z29I40kmFEa+39YlyfyN1bAbR3EpxUUKvTpAyP+QvwScWvdKwaBFy1FmYhQTEaZMXVgzbbqi
```

[Download](#) [Copy](#)

Place your license code below:

[Upload](#) [Paste](#)

[Cancel](#) [← Back](#) [Update](#)

Figure 99: installation_activation_code

3. Copy the activation code.
4. With a different device, log in to the Critical Manufacturing Customer Portal with a User that has access to the required license.
5. Open the **Licenses** menu and select **My Licenses**:

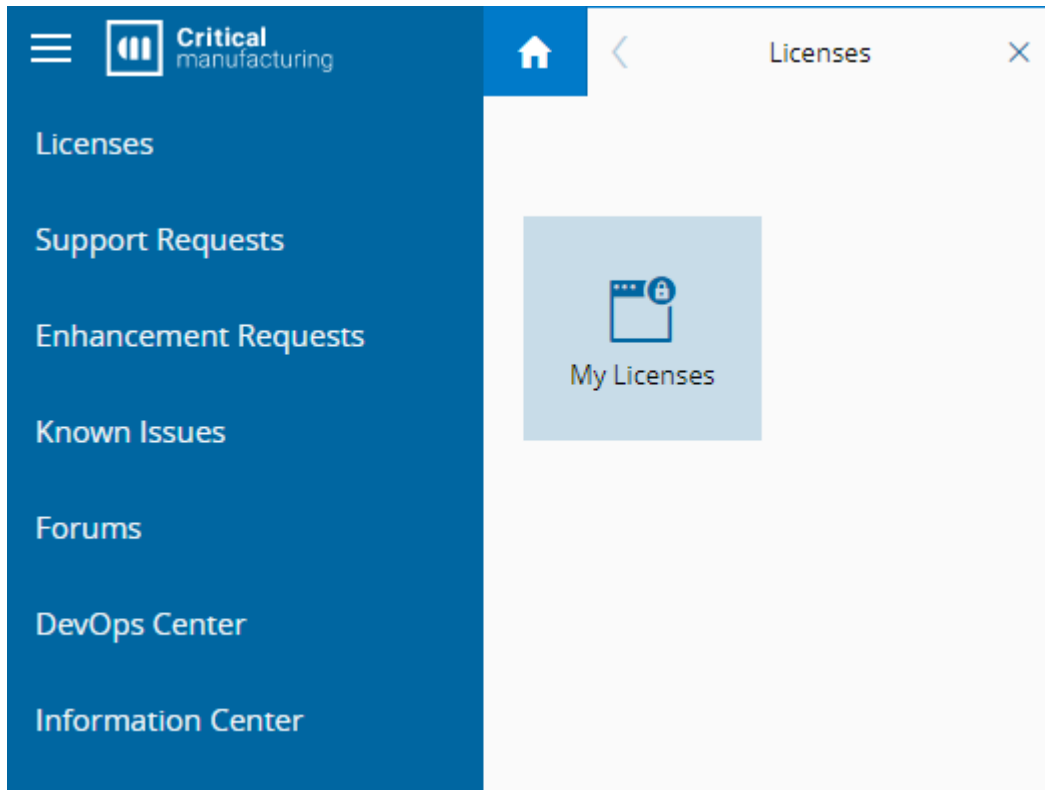


Figure 100: Installation - Activation - Step 2

6. From here you can:

- Select the **Activate License** in the top ribbon:

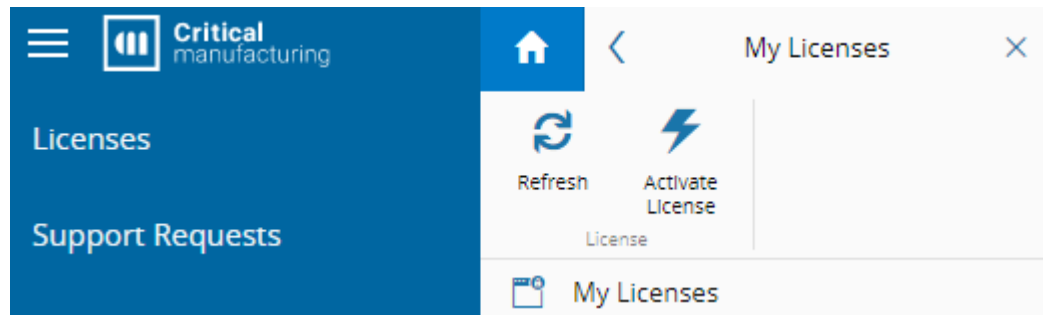


Figure 101: Installation - Activation - Step 3

- Paste the **Activation Code** from the Setup - Offline update license operation:



Activate License

1 ACTIVATION CODE — 2 SELECT LICENSE

Paste Activation Code:

```

UXkMzr/cGkJE/3MgARFz79iOuxH1op+a+P5HkocfuZfb0Gmfon23zSF5vktbGDyerKxvDz7awSDJp7uIMj+e5yE+uH/2P8hcoeI2F8eQ3x9B0b1YA+hQH4bWVZ/czi07Na+e22FRkQjz1lNxrDWhpNkrehAGQw2kc7Juvmlz78n42ShjHIT+GoPovUyU1lpe2amouTmfyoK1S8TBQ
B324drnqunC5GzeExoH8IBJ8HtgAVeMtdUHK8DSzmm1gXJDrbdSmk5Nbr;IpiH9Q8m3pzUoXOLBbXmW9iLoLq80bb14qkcDuDg9FFkq;KXIDi4Yo5U64n9PC61niaeSQ==tW0b99cRiJiIveDvqqi02r89Pr51aePv18r16HaCL9YF+UVF3q+JEQm2/+9DuY2oKx97a8F2oPwRzq
VWuCbLxmfaU0eg7F9iC0mWm1Wk/Fpr5W45612IVp/P8W4y2z;I0F4Cgg5C0Nex2LV2L8eokQVgVW4aPyD96X16Q0B9a8d8AAVQc408XQ603868W8CTUlwyc1150ymBn9PRLD048Ibej86DnyV8nrf00C82dR145VWAV9ag5j8n584U0BmWJ14FFI7IAE380Q5e
B324drnqunC5GzeExoH8IBJ8HtgAVeMtdUHK8DSzmm1gXJDrbdSmk5Nbr;IpiH9Q8m3pzUoXOLBbXmW9iLoLq80bb14qkcDuDg9FFkq;KXIDi4Yo5U64n9PC61niaeSQ==tW0b99cRiJiIveDvqqi02r89Pr51aePv18r16HaCL9YF+UVF3q+JEQm2/+9DuY2oKx97a8F2oPwRzq
Lw5YrD4TH8dgl1LwF+Emu85egYQ1GFc1qgMEFNAHRYeXo7p+8dqY86EYy0TM4dhy3EodOTk2V9KH9V93NOCademGyUG2b4thmSTED0/JpEE2j0Fw38U2m46AprAbxtduv7GAVeou/x1BAZ3VAdrc5Fgu7UVa1kSZFR58SD4y604gS1YF16H3PVHxIXDhp3+HTq5TKADyCY7uFE
G/zcFFzrY06V77X0pt1R131a6bDEEPrdYIumLEf812125dne9Up5VbQ7mltY3N1A4QMEuf5kx/bNCFPLerAyaZB1P5v4e+X3j9eAKJ508/7dvgqr

```

This code is supplied during the 'PRODUCT LICENSE' step of Critical Manufacturing MES installation process.

Upload Paste

Comments:

Cancel < Back Next >

Figure 102: Installation - Activation - Step 4

- Then select the required Environment License and **Activate** it:

Activate License

1 ACTIVATION CODE — 2 SELECT LICENSE

Select the license to use for this installation

	SITE	VERSION	TYPE	MODULES	EXPIRATION DATE	ACTIONS
DEVELOPMENT						
CMF - Training_Development_v9.0.0_ED20230224	CMF - Training	9.0.0	Development	29	02/24/2023	104

Rows per Page: 100 Page 1 of 1 (1 Records)

Comments:

Cancel < Back Activate

Figure 103: Installation - Activation - Step 5

7. A license code will be provided and you should **Download** or **Copy** it:



For the **Command Console**, follow the steps below:

!!! note This method requires that you know the **License Id** or the **License Name**. You also need to have a valid **json** file with the **Environment data** (original installation **json** file), and online access to the Critical Manufacturing Customer Portal.

1. Mount the Critical Manufacturing MES ISO.
2. Open a console at the mounted ISO root:

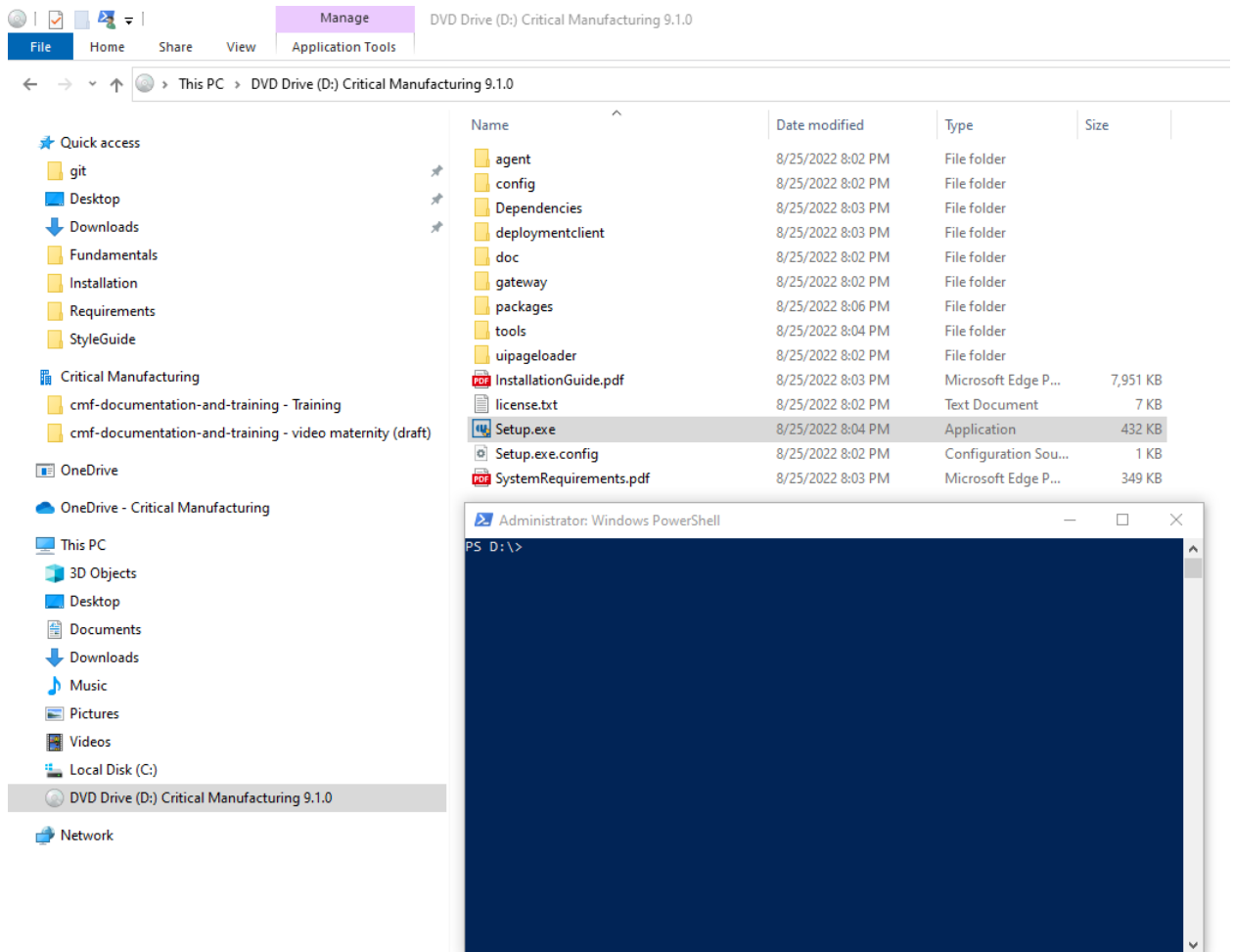


Figure 106: installation_console1

3. Run the **CmfDeploy.exe** and provide the **licenseId** and the **parameters**:

- Command for reference: `.\tools\CmfDeploy.exe installlicense --licenseId="LicenseName" --parameters="OriginalInstallationJsonFilePath"`
- Example for reference: `.\tools\CmfDeploy.exe installlicense --licenseId="CMF - CMFLAB\Development_v7" --parameters="C:\Users\Administrator\Downloads\installation 7.0.2.json"`

4. You will be redirected to the Customer Portal and will need to log in with a User that has access to the provided License. What follows is a successful output example for this operation:



```
PS E:\> .\tools\CmfDeploy.exe installlicense --licenseId="CMF - CMFLAB_Development_v7.0.0_ED20230901" --parameters="C:\Users\Administrator\Downloads\Installation 7.0.2.json"
Logging directory not specified, using 'C:\Users\Administrator\AppData\Local\CMF\DF\Log'.
Starting to install Cmf.Deployment.UpdateLicense...
Version 1.0.0
Activating License 'CMF - CMFLAB_Development_v7.0.0_ED20230901'...
License activated successfully.
Execution plan:
Cmf.Deployment.UpdateLicense : 1.0.0 : Install
Executing powershell script install_prerequisites.ps1
VERBOSE: Installing installation prerequisites
VERBOSE: Installing installation prerequisites
VERBOSE: Module cMsmq already exists.
VERBOSE: Module cMsmq already exists.
VERBOSE: Module MSI already exists.
VERBOSE: Module MSI already exists.
```

Figure 107: installation_console2

```
VERBOSE: OR 11.0.0 dependencies are deployed
Entering execute for Cmf.Deployment.UpdateLicense@1.0.0
CriticalManufacturing:Checked
Leaving execute for Cmf.Deployment.UpdateLicense@1.0.0
Entering complete for Cmf.Deployment.UpdateLicense@1.0.0
Leaving complete for Cmf.Deployment.UpdateLicense@1.0.0
Installation completed.
PS E:\>
```

Figure 108: installation_console3



36 Post Installation

After installing Critical Manufacturing MES, it is necessary to perform the following steps:

- [Enterprise Resource Planning Configuration](#) (optional, only for the usage of the ERP Integration functionality)
- [Report Server Security](#)
- [Connect IoT Installation](#), in [Connect IoT Installation](#)
- [Manually set TRUSTWORTHY Database Property on Always On systems](#)



37 Connect IoT Introduction

- [Connect IoT Installation](#)
- [Automation Manager Configuration File](#)



38 IoT Runtime Components Configuration

The configuration file is a JSON structured file and it will be used by all components (Manager, Monitor, Controller, Driver). It allows the use of some tokens to be inserted depending on the context, those tokens are explained in the following table:

Table 24: Connect IoT Configuration Tokens

Name	Type	Usage	Description
application	Application	<code>\${application}</code>	The name of the current application
pid	Application	<code>\${pid}</code>	The process id of the current application
component	Application	<code>\${componentId}</code>	Name of the component running (<code>AutomationManager</code> , <code>AutomationMonitor</code> , <code>AutomationController</code> , <code>DriverCsv</code> , etc)
entityName	Application	<code>\${entityName}</code>	The name of the context entity for the current application
tmp	OperatingS	<code>\${tmp}</code>	User temporary directory
pwd	OperatingS	<code>\${pwd}</code>	Running script location
cwd	OperatingS	<code>\${cwd}</code>	Working directory (where the command is being executed)

38.1 Base Structure

Table 25: Connect IoT Base Structure

Name	Type	Default Value	Description
id	String		Identifier of the process
monitorHost	String	<code>"localhost"</code>	Address of the local monitor listener address
monitorPort	Integer	<code>0</code>	Port of local monitor listener. 0 for automatic port assignment
cache	String		Local path where the packages will be downloaded and executed
monitorApplicator	String	<code>"\${pwd}\\monitor"</code>	Complete path of the Monitor process
repository	Repository		Repository regarding packages handling
system	System		System access settings
storage	Storage		Section to configure persistency location
logging	Logging		Logging mechanism settings
monitor	monitor		Monitor process specific settings
controller	controller		Controller(s) process(es) specific settings
driver	driver		Driver(s) process(es) specific settings



38.2 Repository Structure

The repository represents the way the Monitor process handles the location where the packages are located (remote) and stored (local).

Table 26: Connect IoT Repository Structure

Name	Type	Possible Values	Default Value	Description
type	String	Npm Directory System		Type of repository - Npm - uses an NPM server - Directory - uses a local directory with the packages and a json-based database descriptor - System - downloads the packages from MES (retrieves settings from the system structure detailed below)
settings	Object			Repository settings (depending on the type)

38.2.1 Type Npm Settings

Table 27: Npm specific settings

Name	Type	Default Value	Description
url	String		URL where the NPM server is located (including port)
token	String		Optional Bearer token used for authentication

38.2.2 Type Directory Settings

Table 28: Directory specific settings

Name	Type	Default Value	Description
path	String		Directory where the packages are located. Must be accessible by process and currently, UNC's are not supported

38.2.3 Directory Examples

```

"repository": {
  "type": "Npm",
  "settings": {
    "url": "YourNpmAddress",
    "token": "YourNpmToken"
  }
}

```



```

"repository": {
  "type": "Directory",
  "settings": {
    "path": "[UserSelectedDirectory]\\MyDirectoryRepository"
  }
}

```

38.3 Storage Structure

Table 29: Connect IoT Storage Structure

Name	Type	Possible Values	Default Value	Description
type	String	Directory	Directory	Type of persistency location
settings	Object			Storage type settings
settings	String			Base path where the persistency data will be stored. OperatingSystem tokens are supported
settings	Integer string		"30d"	Number of seconds to store the data. 0 means forever. Supports string with a trailing token indicating the time unit: s - seconds m - minutes h - hours d - days w - weeks (7 days) m - months (30 days)

38.3.1 Storage Structure Examples

```

"storage": {
  "type": "Directory",
  "settings": {
    "path": "[UserSelectedDirectory]/Persistency",
    "retentionTime": 3600
  }
}

```

```

"storage": {
  "type": "Directory",
  "settings": {
    "path": "[UserSelectedDirectory]/Persistency",
    "retentionTime": "10d"
  }
}

```

38.4 System Structure



Table 30: Connect IoT System Structure

Name	Type	Default Value	Description
tenantName	String		System Tenant name to use by the components
address	String	"localhost"	
port	Integer		
timeout	Integer		
useSsl	Boolean	false	
isLoadBalancingEnabled	Boolean	false	Whether the configuration pointing to load balancing
authentication	Authentication		Object

38.4.1 Authentication Structure

Table 31: Authentication structure

Name	Type	Possible Values	Default Value	Description
type	String	Password SecurityPortal		Authentication type to use
settings	Object			Authentication settings (depending on the type)

38.4.2 Authentication type Password settings

Table 32: Authentication type Password structure

Name	Type	Default Value	Description
domain	String		User domain
username	String		User name
password	String		User password

38.4.3 Authentication type SecurityPortal settings

Table 33: Authentication type SecurityPortal structure

Name	Type	Default Value	Description
clientId	String		Id of the client accessing to (MES)



Name	Type	Default Value	Description
accessToken	String		Security Portal generated long term Access Token. To be set on non-interactive scenarios with Security Portal
openIdConfig	String		URL where the OpenId endpoint can be accessed. Example: http://YourEnvironment/tenant/YourTenant/.well-known/openid-configuration

38.4.4 Examples

```
"system": {
  "tenantName": "YourTenant",
  "address": "YourEnvironment",
  "port": 8093,
  "timeout": 60000,
  "useSsl": false,
  "isLoadBalancingEnabled": false,
  "authentication": {
    "_type": "Password",
    "settings": {
      "domain": "YourDomain",
      "username": "administrator",
      "password": "1234567890"
    }
  }
}
```

```
"system": {
  "tenantName": "YourTenant",
  "address": "YourEnvironment",
  "port": 8093,
  "timeout": 60000,
  "useSsl": false,
  "isLoadBalancingEnabled": false,
  "authentication": {
    "type": "SecurityPortal",
    "settings": {
      "clientId": "MES",
      "accessToken": "user created access token",
      "openIdConfiguration":
        "http://YourEnvironment/tenant/YourTenant/.well-known/openid-configuration"
    }
  }
}
```

38.5 Logging Structure

The logging configuration allows several different loggers to be configured at same time, meaning that for each type of logger, different options can be configured.



Table 34: Connect IoT Logging Structure

Name	Type	Possible Values	Default Value	Description
id	String			Identifier of the logger, if none is configured, only one of the same type is allowed
type	String	Console File Http OTLP		Type of the transport
options	Object			Transport options, different for each type of transport, see each transport information below
apply	Array of String	AutomationMonitor AutomationController AutomationManager DriverSecsGem DriverOpcUA DriverMqtt DriverBle DriverCsvFile DriverRawFile DriverKeyboardWedge DriverOib DriverOpcDA DriverSerial DriverTcpIp		A wildcard can configure all or filter by name, example: <code>'Driver*'</code>

38.5.1 Common Transport Options

These options apply to all types:

Table 35: Connect IoT Common Transport Options

Name	Type	Possible Values	Default Value	Description
level	String	debug info warn error	"info"	Minimum level of messages that this transport should log. If <code>info</code> level is defined, all entries of level <code>info</code> , <code>warn</code> and <code>error</code> will be logged and <code>debug</code> entries will be ignored.
label	String		"\${application}"	Label to append in the beginning of the log entry (if used in the format)
format	String		"\${log.time}"	Format of the line of the log



Name	Type	Possible Values	Default Value	Description
times	Boolean		true	Flag indicating if we should prepend output with timestamps. If function is specified, its return value will be used instead of timestamps
times	String		"YYYY-MM-DD HH:mm:ss.SSS"	Format of the timestamp in the log entry. Please refer to https://github.com/taylorhakes/fecha#formatting-tokens for full format information
maxLength	Integer		0	Max length of the message entry to log (<=0 to ignore). If entry is larger than the value, will log (70% of the maxLength) of the start of the message and the last (30% of maxLength) characters. This setting is useful to keep huge communication logs in a more controllable state.
specifyLevel	Boolean		false	Only log the specified verbosity level. With this setting set to true the levels greater than the level will be ignored.
isEnabled	Boolean		true	Is the Transport enabled

38.5.2 Console Transport Options

This transport log all the messages to the application console.

Table 36: Connect IoT Console Transport Options

Name	Type	Default Value	Description
colorize	Boolean	true	Flag indicating if we should colorize output.
colorizeMessage	Boolean	true	Is the colorization to apply to the full message or only the verbosity.

38.5.3 Example

```
{
  "type": "Console",
  "options": {
    "level": "debug",
    "prettyPrint": true,
    "colorizeMessage": true
  },
  "applications": [ "*" ]
}
```

38.5.4 File Transport Options

As the name states, this transport log all the messages to the file system.



Table 37: Connect IoT File Transport Options

Name	Type	Default Value	Description
frequency	String		A string representing the frequency of rotation. This is useful if you want to have timed rotations, as opposed to rotations that happen at specific moments in time. Valid values are ' \#m ' or ' \#h ' (e.g., '5m' or '3h'). Leaving this null relies on datePattern for the rotation times.
datePattern	String	"YYYY-MM-DD"	A string representing the moment.js date format to be used for rotating. The meta characters used in this string will dictate the frequency of the file rotation. For example, if your datePattern is simply ' HH ' you will end up with 24 log files that are picked up and appended to every day.
filename	String	"LogFile\"	Filename to be used to log to. This filename can include the <code>{date}</code> placeholder which will include the formatted datePattern at that point in the filename <code>dirname</code> String <code>""</code> The directory name to save log files to <code>maxSize</code> String <code>"10m"</code> Maximum size of the file after which it will rotate. This can be a number of bytes, or units of kb, mb, and gb. If using the units, add 'k', 'm', or 'g' as the suffix. The units need to directly follow the number. <code>maxFiles</code> String <code>"30d"</code> Maximum number of logs to keep. If not set, no logs will be removed. This can be a number of files or number of days. If using days, add 'd' as the suffix <code>option</code> Object <code>{ flags: 'a', mode: 0o777 }</code> An object resembling https://nodejs.org/api/fs.html#fs_fs_createwritestream_path_options indicating additional options that should be passed to the file stream <code>auditFile</code> String <code>"\${dirname}/.-audit.json"</code>

38.5.5 File Transport Options Example

```
{
  "id": "MyFileLogger01",
  "type": "File",
  "options": {
    "filename": "${applicationName}_${date}.log",
    "dirname": "${tmp}/YourManager01/Logs/${applicationName}",
    "level": "debug",
    "timestampFormat": "HH! [] (images\\.png){ width=20px }ss.SSSSS",
    "maxSize": "10m",
    "maxFiles": 5,
    "maxLength": 5000,
    "specificLevelLock": false
  },
  "applications": [ "AutomationMonitor", "AutomationManager" ]
}
```

38.5.6 HTTP Transport Options

The HTTP transport is a generic way to log, query, and stream logs from an arbitrary HTTP endpoint, preferably **winston**. It takes options that are passed to the node.js http or https request:



Table 38: Connect IoT HTTP Transport Options

Name	Type	Default Value	Description
host	String	"localhost"	Remote host of the HTTP logging endpoint.
port	Integer	80 or 443	Remote port of the HTTP logging endpoint.
path	String	"/"	Remote URI of the HTTP logging endpoint.
auth	Object	None	An object representing the username and password for HTTP Basic Auth.
ssl	Boolean	false	Value indicating if we should use HTTPS

38.5.7 HTTP Transport Options Example

```
{
  "id": "MyHttp01",
  "type": "Http",
  "options": {
    "host": "localhost",
    "port": "80",
    "path": "logger",
  },
  "applications": [ "AutomationController" ]
}
```

38.5.8 OTLP Transport Options

The OTLP transport is a transport to broadcast logs using the open telemetry standard. It requires the endpoint to be configured:

Table 39: Connect IoT OTLP Transport Options

Name	Type	Default Value	Description
isEnabled	Boolean		Flag to enable the transport.
level	String		Logging Level.
endpoint	String		An internal endpoint that receives and processes telemetry signals sent according to the OTLP (open telemetry protocol) format.

38.5.9 OTLP Transport Options Example

```
{
  "type": "OTLP",
  "options": {
    "isEnabled": true,
    "level": "debug",
    "endpoint": "https://mytelemetrystack.com/telemetry/http",
  },
}
```



```
"applications": [ "*" ]
},
```

38.6 Monitor Structure

The optional monitor section is specifically for the monitor process and are not used by the other applications.

Table 40: Connect IoT Monitor Structure

Name	Type	Default Value	Description
notifyBeforeK	Boolea	true	Flag indicating if a notification to the processes before killing them (allowing them to properly cleanup any resources, disconnect from devices, etc), or simply kill them.
killNotificator	Integer	10000	Number of milliseconds to wait for the "about to be killed" process reply that it has finished the cleanup an is ready to be killed.
retryAttempts	Integer	30	When calls protected by retry mechanism fail, number of executions before failing
sleepBetween	integer	1000	When calls protected by retry mechanism fail, time to wait between retries
processComn	SslCo:		Configuration to allow inter-process communications to be using SSL (monitor <-> controller(s) and monitor <-> driver(s))

38.6.1 Monitor Structure Example

```
"monitor": {
  "notifyBeforeKill": "true",
  "killNotificationTimeout": 30000,
  "retryAttempts": 45,
  "sleepBetweenAttempts": 1000,
  "processCommunication": {
    "useSsl": true,
    "privateKey": "YourCertificatesRepository\\key.pem",
    "certificate": "YourCertificatesRepository\\cert.pem",
    "certificateAuthority": "YourCertificatesRepository\\ca-cert.pem"
  }
}
```

38.7 Controller Structure

The optional controller section is specifically for the controller(s) process(es) and are not used by the other applications.

Table 41: Connect IoT Controller Structure

Name	Type	Default Value	Description
retryAttempts	Integer	30	When calls protected by retry mechanism fail, number of executions before failing



Name	Type	Default	
		Value	Description
sleepBetweenAttempts	integer	1000	When calls protected by retry mechanism fail, time to wait between retries
profilerSessionsLocation	string		Location where the profiler sessions will be stored

38.7.1 Controller Structure Example

```
"controller": {
  "retryAttempts": 45,
  "sleepBetweenAttempts": 1000,
  "profilerSessionsLocation": "YourControllerRepository\\tempController"
}
```

38.8 Driver Structure

The optional driver section is specifically for the driver(s) process(es) and are not used by the other applications.

Table 42: Connect IoT Driver Structure

Name	Type	Default	
		Value	Description
retryAttempts	Integer	30	When calls protected by retry mechanism fail, number of executions before failing
sleepBetweenAttempts	integer	1000	When calls protected by retry mechanism fail, time to wait between retries
processCommunication	SslConfig		Configuration to allow inter-process communications to be using SSL (driver<-> controller)

38.8.1 Driver Structure Example

```
"driver": {
  "retryAttempts": 45,
  "sleepBetweenAttempts": 1000,
  "processCommunication": {
    "useSsl": true,
    "privateKey": "YourCertificatesRepository\\key.pem",
    "certificate": "YourCertificatesRepository\\cert.pem",
    "certificateAuthority": "YourCertificatesRepository\\ca-cert.pem"
  }
}
```

38.9 SslConfig structure

When communication between processes require SSL use these settings.



Table 43: Connect IoT SslConfig Structure

Name	Type	Value	Description
useSsl	Box	false	Use SSL communication on component communication
rejectUnauthorized	Box	false	If not false a server automatically reject clients with invalid certificates. Allows bypass of error: "Connection with monitor error: unable to verify the first certificate" on a completely unsecured way when set to false! also, if we want to ignore environment variable "NODE_TLS_REJECT_UNAUTHORIZED=" we may set rejectUnauthorized: true On self-signed certificates, we can pass the correct CA (certificate authority) certificate with certificateAuthority option
privateKey	string		PEM encoded SSL private key Value or file full path containing the key value
certificate	string		PEM encoded SSL certificate Value or file full path containing the certificate value
certificateAuthority	string		PEM encoded SSL custom certificate authority (CA) Value or file full path containing the certificate value

38.9.1 SslConfig Structure Example

```
{
  "useSsl": true,
  "privateKey": "YourCertificatesRepository\\key.pem",
  "certificate": "YourCertificatesRepository\\cert.pem",
  "certificateAuthority": "YourCertificatesRepository\\ca-cert.pem"
}
```

38.10 Full Example

The following JSON document illustrates a possible configuration:

```
{
  "id": "MyManager",
  "cache": "[MyCacheLocation]/Cache",
  "hostName": "localhost",
  "monitorApplication": "${pwd}/monitor.js",
  "repository": {
    "type": "System"
  },
  "system": {
    "tenantName": "MyTenant",
    "address": "MyAddress",
    "port": 80,
    "timeout": 120000,
    "useSsl": false,
    "isLoadBalancingEnabled": false,
    "authentication": {
      "type": "SecurityPortal",
      "settings": {
        "clientId": "MES",

```



```
    "accessToken": "user access token",
    "openIdConfiguration":
      "http://MyAddress/SecurityPortal/tenant/MyTenant/.well-known/openid-configuration"
  }
}
},
"isMinimal": false,
"storage": {
  "type": "Directory",
  "settings": {
    "path": "[MyPersistencyDirectory]/Persistency",
    "retentionTime": "30d"
  }
},
"logging": [
  {
    "type": "Console",
    "options": {
      "level": "debug",
      "prettyPrint": true,
      "colorizeMessage": true,
      "isEnabled": true
    },
    "applications": [
      "*"
    ]
  },
  {
    "type": "OTLP",
    "options": {
      "level": "info",
      "isEnabled": false
    },
    "applications": [
      "*"
    ]
  },
  {
    "id": "Controllers",
    "type": "File",
    "options": {
      "filename": "${applicationName}_${date}.log",
      "dirname": "[MyLogsDirectory]/Instances/${entityNameNormalized}/${componentId}",
      "level": "debug",
      "timestampFormat": "HH! [] (images\\.png){ width=20px }ss.SSSSS",
      "maxSize": "10m",
      "maxFiles": "30d",
      "isEnabled": true
    },
    "applications": [
      "AutomationController"
    ]
  }
],
{
```



```
"id": "Drivers",
"type": "File",
"options": {
  "filename": "${applicationName}_${date}.log",
  "dirname": "[MyLogsDirectory]/Instances/${entityNameNormalized}/${componentId}",
  "level": "debug",
  "timestampFormat": "HH! [] (images\\.png){ width=20px }ss.SSSSS",
  "maxSize": "10m",
  "maxFiles": "30d",
  "isEnabled": true
},
"applications": [
  "Driver*"
]
},
{
  "id": "ManagerAndMonitor",
  "type": "File",
  "options": {
    "filename": "${applicationName}_${date}.log",
    "dirname": "[MyLogsDirectory]/Instances/Manager/${applicationName}",
    "level": "info",
    "timestampFormat": "HH! [] (images\\.png){ width=20px }ss.SSSSS",
    "maxSize": "10m",
    "maxFiles": "30d",
    "isEnabled": true
  },
  "applications": [
    "AutomationMonitor",
    "AutomationManager"
  ]
},
{
  "id": "Jobs",
  "type": "File",
  "options": {
    "filename": "Job_${jobId}_${date}.log",
    "dirname": "${temp}/ConnectIoT/Jobs",
    "auditFile": "${temp}/ConnectIoT/Jobs/.audit.json",
    "level": "info",
    "timestampFormat": "HH! [] (images\\.png){ width=20px }ss.SSSSS",
    "maxSize": "10m",
    "maxFiles": "30d",
    "isEnabled": true
  },
  "applications": [
    "AutomationJob"
  ]
}
]
```



39 Connect IoT Installation

The **Connect IoT Installation** requires post installation steps that are described in this document.

!!! note While this method provides a direct installation method for Connect IoT, Critical Manufacturing recommends using the automatic deployment method, which you can read about in [\[\[user-guide-automation-manager-configure-deployment\]\]](#).

39.1 Package Repository

As described in the System Requirements, Connect IoT requires a Package Repository to store all binaries and respective versions.

Currently, two types of repositories are supported (**NPM** and **Directory**), each with their advantages and disadvantages.

Table 44: Package Repository types

Type	Advantages	Disadvantages
NPM	Authentication with roles Web-Server-Based Well defined Api Commercial solutions with support	Must install/configure/maintain server Commercial solutions/support is not free Internet connection may be required Updating packages requires unpublish + publish (development + hotfixes)
Director	Free Easy to prepare Easy to retrieve/change packages No internet/ports required	Must be available in all IoT computers (mount) Authentication based on ACL Anyone with permissions can simply delete everything

39.2 Deploy Connect IoT Packages

39.2.1 Package Selection

Run the setup wizard and select the package **Cmf.ConnectIoT.Packages** from the dropdown and select **Next**.



Critical Manufacturing EMPOWERING OPERATIONS
OFFLINE

Critical Manufacturing Installation

— 2 PACKAGE SOURCES — 3 PACKAGE SELECTION — 4 IMPORT INSTALLATION FILE — 5 SUMMARY — 6 COMPLETE INSTALLATION

Select the packages you want to install

* Package:

* Version:

PACKAGE	VERSION
ConnectIoT Packages	9.1.0

Figure 109: Deploy Connect IoT Packages - Package Selection

If you have any previously exported configuration file, you can import it. Nevertheless, continue to the **Next** step.

39.2.2 Connect IoT Configuration

In the **NPM Server Repository** group, fill out the fields as described below:

- **Is Enabled** - specify if the packages should be published to an NPM Repository.
- **Address** - full address (including port) of the server (must support NPM api).
- **Tag** - tag to mark the packages.
- **Registry User** - username with publish permissions.
- **Registry User Password** - password of the user indicated in the previous field.
- **Registry User Email** - email to associate to the user that will publish the packages.

In the **Directory Repository** group, define the following options:

- **Is Enabled** - specify if the packages should be published to a Directory Repository.
- **Location** - directory full path (if the directory does not exist, it will be created).



Critical Manufacturing EMPOWERING OPERATIONS
OFFLINE

Critical Manufacturing Installation

— 5 CONNECTIOT CONFIGURATION — 6 MES CONFIGURATIONS — 7 CONNECTIOT REPOSITORY SETTINGS — 8 CONNECTIOT MANAGER

✕ ☰

▼ NPM SERVER REPOSITORY

Is Enabled:

Address:

Tag:

Registry User:

Registry User Password:

Registry User Email:

▼ DIRECTORY REPOSITORY

Is Enabled:

Location:

Figure 110: Deploy Connect IoT Packages - Connect IoT Configurations

39.2.3 MES Configurations

Define the configurations for the MES system:

- **Tenant Name** - tenant name used to connect to the MES system.
- **HostName or IP Address** - address where the MES system is installed.
- **Port** - port address for the MES system.
- **Use SSL** - whether SSL will be used.

In the **User Account** group, fill out the details for the user that will access the system:

- **User Account** - user account to access the MES system.
- **User Password** - user password to access the MES system.



Critical manufacturing EMPOWERING OPERATIONS
OFFLINE

Critical Manufacturing Installation

— 6 MES CONFIGURATIONS — 7 CONNECTIOT REPOSITORY SETTINGS — 8 CONNECTIOT MANAGERS CONFIGURATIONS — 9 SUMMARY

MES CONFIGURATIONS

Tenant Name: CriticalManufacturing

HostName or IP Address: localhost

Port: 8083

Use SSL:

USER ACCOUNT

User Account: DOMAIN\connectuser

User Password:

[Cancel](#) [< Back](#) [Next >](#)

Figure 111: Deploy Connect IoT Packages - MES Configurations

Select **Next** to continue.

39.2.4 Connect IoT Repository Settings

In the **Repository Settings** group, define whether to use Configure Repository Settings, as well as the type of Repository to use:

- **Npm**
- **Directory**

In the **Temporary Files** group, select whether the temporary files should be removed.

Select **Next** to continue.



Critical manufacturing EMPOWERING OPERATIONS
OFFLINE

Critical Manufacturing Installation

⑦ CONNECTIOT REPOSITORY SETTINGS — ⑧ CONNECTIOT MANAGERS CONFIGURATIONS — ⑨ SUMMARY — ⑩ COMPLETE INSTALLATION

✕ ↻

REPOSITORY SETTINGS

Configure Repository Settings:

Repository Type: ✕ ▼

TEMPORARY FILES

Remove Temporary Files:

Cancel < Back Next >

Figure 112: Deploy Connect IoT Packages - Connect IoT Repository Settings

39.2.5 Connect IoT Managers Configurations

In the **Manager Selection** group, insert the Automation Manager IDs (one entry per line), and define the base directory where the Managers will be running.

Finally, define whether the Manager should be installed as a service.

Below, in the **Automation Manager User Settings** group, define the user credentials (username and password) to be used to run the Automation Manager.



Critical Manufacturing EMPOWERING OPERATIONS OFFLINE

Critical Manufacturing Installation

7 CONNECTIOT REPOSITORY SETTINGS — 8 **CONNECTIOT MANAGERS CONFIGURATIONS** — 9 SUMMARY — 10 COMPLETE INSTALLATION

MANAGER SELECTION

Automation Managers IDs:

Managers Base Directory:

Install Service:

AUTOMATION MANAGER USER SETTINGS

Username:

Password:

Figure 113: Deploy Connect IoT Packages - Connect IoT Managers Configurations

Select **Next** to review the installation summary, and then continue with the installation process until the end.

39.3 Manually Deploy Packages

Sometimes, there is the need to manually deploy packages - example: Customization packages, so it is important to understand how to accomplish this task.

39.3.1 Directory Repository

It is fairly simple to deploy a package or even a set of packages:

1. Copy the file(s) into the directory you are using as the Directory Repository (the one used in the installation of the previous section).
2. Execute the `.rebuildDatabase.ps1` PowerShell script that was created during the installation:



```
Administrator: Windows PowerShell
PS C:\repositoryLocation> .\.\rebuildDatabase.ps1
** Connect IoT Directory Repository **
** Rebuilding database of deployed packages **
Found 24 files in C:\repositoryLocation
Found 'cmf.lbos', version '7.2.0-20200414.4'
Found '@criticalmanufacturing/connect-iot-common', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-controller', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-controller-engine', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-controller-engine-core-tasks', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-controller-engine-filedrivers-tasks', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-controller-engine-mes-tasks', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-controller-engine-oib-tasks', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-controller-engine-secsgem-tasks', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-driver', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-driver-ble', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-driver-csv', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-driver-fileraw', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-driver-keyboardwedge', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-driver-mqtt', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-driver-oib', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-driver-opcda', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-driver-opcua', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-driver-secsgem', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-driver-serial', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-driver-tcpip', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-manager', version '7.2.0-202004146'
Found '@criticalmanufacturing/connect-iot-monitor', version '7.2.0-202004146'
Found '@criticalmanufacturing/messagebus-client', version '7.2.0-20200218.2'

** Finished **
Terminating executing within 5 seconds
PS C:\repositoryLocation> _
```

Figure 114: Rebuild database Powershell script

3. The database is now updated will all the available packages.

!!! note The script fully updates the content of the directory, so you can delete/add/update the packages and run the script.

39.3.2 NPM Repository

If the NPM server is configured with authentication roles for publishing, make sure you log in first:

```
npm login --registry=<url>
```

Then, and for each of the packages you intend to publish, run the command:

```
npm publish <package> --registry=<url> --tag=<tag>
```



```
Administrator: Command Prompt
C:\repositoryLocation>npm publish criticalmanufacturing-connect-iot-driver-opcua-7.2.0-202004146.tgz --registry=http://localhost:4873 --tag=dev
npm notice
npm notice package: @criticalmanufacturing/connect-iot-driver-opcua@7.2.0-202004146
npm notice === Tarball Contents ===
npm notice 6.2kB package.json
npm notice 8.0kB npm-shrinkwrap.json
npm notice 12.1kB README.md
npm notice 1.6kB certificates/client_selfsigned_cert_2048.pem
npm notice 1.7kB certificates/private_key.pem
npm notice 136.4kB data/gif-anime.gif
npm notice 17.6kB data/image.bmp
npm notice 26.7kB data/tiger.jpg
npm notice 41.2kB data/tux.png
npm notice 43.0kB nodesets/1.02/FTNIR.NodeSet2.xml
npm notice 449.7kB nodesets/1.02/Opc.Ua.Adi.NodeSet2.xml
npm notice 89.5kB nodesets/1.02/Opc.Ua.Di.NodeSet2.xml
npm notice 485.3kB nodesets/Opc.Ua.Adi.NodeSet2.xml
npm notice 105.6kB nodesets/Opc.Ua.Di.NodeSet2.xml
npm notice 250.0kB nodesets/Opc.Ua.Gds.NodeSet2.xml
npm notice 29.2kB nodesets/Opc.Ua.NodeSet2.Part8.xml
npm notice 2.9MB nodesets/Opc.Ua.NodeSet2.xml
npm notice 1.7MB nodesets/Opc.Ua.NodeSet2Old.xml
npm notice 17.8kB nodesets/UANodeSet.xsd
npm notice 13.3MB src/index.js
npm notice 4.7kB test_fixtures/fixture_empty_nodeset2.xml
npm notice 70.0kB test_fixtures/fixture_simple_statemachine_nodeset2.xml
npm notice 16.3kB test_fixtures/fixture_nodeset_objects_with_some_methods.xml
npm notice 213.5kB test_fixtures/mini.Node.Set2.xml
npm notice 863B test_fixtures/minimalist_nodeset_with_models_more_complex.xml
npm notice 346B test_fixtures/minimalist_nodeset_with_models.xml
npm notice 4.4kB test_fixtures/nodeset_issue543.xml
npm notice 1.8kB test_fixtures/nodeset_with_analog_items.xml
npm notice === Tarball Details ===
npm notice name: @criticalmanufacturing/connect-iot-driver-opcua
npm notice version: 7.2.0-202004146
npm notice package size: 2.8 MB
npm notice unpacked size: 20.0 MB
npm notice shasum: f8d67a6cf34ff50efac266fe732fa368ff74e6d6
npm notice integrity: sha512-P4eGYvx1MYnK1[...].JCLYrPxWG/R94g==
npm notice total files: 28
npm notice
+ @criticalmanufacturing/connect-iot-driver-opcua@7.2.0-202004146
C:\repositoryLocation>
```

Figure 115: Running npm commands

39.4 Install Automation Manager

As of version 7.2, it is possible to download, from the [Automation Manager](#) entity page, a zip compressed file fully prepared to be used, so, installing the Automation Manager has never been easier.

After downloading the file using the respective wizard, simply extract all the contents into the destination directory.

The final step is to install the automation as a Windows Service. For this feature, execute the PowerShell script that is located in `scripts\InstallService.ps1` as an administrator (requirement to create Windows Services), and answer the questions.

```
Administrator: Windows PowerShell
PS C:\IoT\JSManager01\scripts> .\InstallService.ps1
Installing this Manager as a Windows Service...
Identified Manager as being 'JSManager01'. Preparing service configuration...
Installing Windows Service with ID 'ConnectIoT (JSManager01)'
User to run the service (leave empty for Local System): CMF\jpsantos
Enter password for user 'CMF\jpsantos': *****
User changed!
Done!
PS C:\IoT\JSManager01\scripts>
```

Figure 116: Install Automation Manager

!!! info .Net Framework 3.5 or higher is required for a correct installation of the Automation Manager.



39.5 Troubleshooting

39.5.1 "Unable to verify the first certificate"

39.5.2 "Unable to get local issuer certificate"

When you get any of these errors, this means you are connecting to an SSL enabled host and most likely, the server certificate was issued by a non trusted certificate authority.

To fix this issue, you need to create a text file with the entire chain of certificates (in `.pem` format), which is needed to allow the server certificate to be validated.

If you only have certificates in `.pfx` format, you can use OpenSSL to convert them:

```
openssl pkcs12 -in file.pfx -out file.pem -nodes
```

The structure of the final file is expected to be something like this:

```
-----BEGIN CERTIFICATE-----
bGUgQ28xEDA0BgnVBAsMB3R1Y2hvcHMxCzAJBgNVBAMAMNhMSAwHgYJKoZIhvcN
AQkBFhFjZXJ0c0BlcGFtcGx1LmNvbTAeFw0xOTA1MTcxMDQ5NTRaFw0ONjEwMDEx
...
oEGp4U7q1UGmGfmXKiT/gsxJB6bbD6k01SVdE+706WLg1vN4cLj1jvIr00jhWt41
sJtjAyB64zRvS5Ic7Vidv6UDMM=
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MDQ5NTRaMIGBMQswCQYDVQQGEwJVUzELMAkGA1UECAwCTUExDzANBgNVBACMBkKJv
c3RvbjETMBEGA1UECgwKRXhhbXBsZSBDbzEQMA4GA1UECwwHdGVjaG9wczELMAkG
...
knyZpJnYVid5NUVmjWNS1K/S6eA/Ka9LxFUhjRt0MMcXP91YHp6+rgsigZt5c3V
aBM3yGsn3YQnttUNp2dQwBgNpH9=
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
BwmQEGpD6ECCjPXZLH38KLCpSLzzQEz0tulvDtRIG1EPLsd1WMLnwbQPG/TJ2M5
GpezB90Lqfi6P00bEI03/3dAewgn4xhW6aZHf6yWgtM7gdQIJpRXK2T8ZGwv1LMz
...
KAjgC4ttj4D1Jfljq61DlegE/0BbAtqNiQ3er9NB/wor3ET5e00/12Ly1eotiHd5
/rkm+ENnvbNwk4w+LzHXfejtsAh=
-----END CERTIFICATE-----
```

Then, set the location of this file into the `NODE_EXTRA_CA_CERTS` environment variable:

```
set NODE_EXTRA_CA_CERTS=c:\certificates\extra_ca_certificates.txt
```

As an alternative, you can disable the certificate integrity check. This will not disable security, but will use the certificate exactly as received. However, this option should **only** be used in a development environment.

```
set NODE_TLS_REJECT_UNAUTHORIZED=0
```

!!! note The environment variables should be defined as a persistent wide definition. The commands explained in this section are temporary and will only be valid during the session of the command window.



40 Theobald ERPConnect License

In addition, it is also necessary to license the Theobald ERPConnect component by obtaining a valid license key which must be encrypted using the tool *String Encrypter* available in the installation CD in the folder `packages\Cmf.ERPLicenseEncrypter.6.1.0.zip`. The encrypted string must be set in the configuration file of each application server under the entry **ERPConnectorLic** in the **SapConnector** section of the `.config` file together with the remaining ERP configuration entries, as listed in the next section.

40.1 ERP Application Configuration Entry Files

The table below describes the ERP application configuration entries of the SapConnector section `.config` of the Critical Manufacturing application.

Table 45: ERP Application Configuration Entry Files

Key	Description	Example
ERPMgmtActive	Enable or disable ERP integration - this flag will determine if the Critical Manufacturing will be able to send and receive information from SAP (true, false)	<code>true</code>
ERPHost	The SAP server name	<code>/H/saprouter.mycompany.com/H/m2bsrv03</code>
ERPSystemNumb	The SAP system number	<code>0</code>
ERPService	The name of the gateway in SAP, which is usually a string that results from the concatenation of the constant <code>sapgw</code> with the system number (i.e: <code>sapgw00</code>)	<code>sapgw00</code>
ERPProgramID	The Registered Server Program name (this must match the Registered Server Program defined in SAP in transaction SM59)	<code>ZERPCONNECT</code>
ERPUser	The user name used to connect to the SAP system	<code>user001</code>
ERPPwd	The user password used to connect to the SAP system	<code>12345678</code>
ERPClient	The SAP client (usually a number)	<code>800</code>
ERPLanguage	A two letter string that defines the language to be used for the SAP integration (e.g.: EN for English or DE for German)	<code>EN</code>
ERPConnectorLic	ERP connector license string - this string must be encrypted from the original license string and using the tool Script Encrypter included in the <code>Software\cmNavigo Tools</code> directory	<code>Encrypted license key text</code>

40.2 Configuring ERP Integration

Critical Manufacturing ships with a generic ERP Integration solution that is capable of working with different ERPs. This configuration is based on three tables listed below, filled with sample configuration data for illustration purposes:



40.2.1 IntegrationSystem (Lookup Table)

Used to define the different systems to be integrated.

Table 46: IntegrationSystem Lookup Table

Value	Description
MES	MES
ERP	ERP

40.2.2 IntegrationHandler (Generic Table)

Used to register the assembly to be used for the integration.

Table 47: IntegrationHandler Generic Table

Name	AssemblyQualifiedName
SapIntegrationHanc	Cmf.Foundation.Integration.SapConnector.SapIntegrationPackage, Cmf.Foundation.Integration.SapConnector, Version=4.2.0.0, Culture=neutral, PublicKeyToken=6bbf07329f6aa8df

40.2.3 IntegrationHandlerResolution (Smart Table)

Used to define the assembly to be used for a particular integration between two systems.

Table 48: IntegrationHandlerResolution Smart Table

FromSystem	ToSystem	MessageType	HandlerType
MES	ERP		SapIntegrationHandler
ERP	MES		SapIntegrationHandler

More information in the [IntegrationHandlerResolution](#) Smart Table.



41 Manually Set TRUSTWORTHY Database Property on Always On Systems

For customers installing Critical Manufacturing on Always On availability groups it is necessary to manually set the trustworthy flag on the secondary replicas. For the Online, ODS and DWH databases follow this procedure:

1. Failover to the node currently holding the secondary replica.
2. Run the following command statement in SQL Management Studio:

```
ALTER DATABASE <Database Name> SET TRUSTWORTHY ON;
```



42 Report Server Security

This section describes the SQL Server Reporting Services' security configuration from the point of view of SQL Reporting Services' server. Critical Manufacturing provides an integrated security management GUI which allows the security configuration to be managed from a central location.

!!! note To access reports from Critical Manufacturing, a domain group must be configured in the report server and the same domain group must be created as a role in Critical Manufacturing. All users that belong to that role will have access to the reports.

!!! note The user who is running the Critical Manufacturing Windows service must have administration privileges on the SQL Server Reporting Services server(s).

!!! note Due to security restrictions on application servers, it is advisable to open the Report Server Manager in a client computer and not on the server.



43 Accessing SQL Server Reporting Services

To configure the security settings for Critical Manufacturing's reports, it is necessary to access SQL Server Reporting Services™ by using the browser and providing an URL address in the form:

[http://\[server_name\]/Reports\\[instance_name\]](http://[server_name]/Reports\[instance_name])

- The [\[server_name\]](#) must be replaced by the name of the server where the SQL Server Reporting Services is installed.
- The [\[instance_name\]](#) must be replaced by the name of the instance where the SQL Server Reporting Services is running.

In case that the server name is [dbserver](#) and the SQL Server Reporting Services instance name is [online](#), the URL address to access the SQL Server Reporting Services is:

http://dbserver/Reports_ONLINE

!!! note Due to security restrictions and browser configurations, adding the Report Server manager URL to the browser trusted sites' list, could be required.



44 Assigning or modifying a User/Group to a System Role

Follow the steps below to assign or modify a user/group to a system role:

1. Click on **Site Settings**, available in the top right corner.
2. In the left tab, click on **Security**. The roles for each user and group are displayed.
3. To add a new role, click on the **New Role Assignment**, and then enter the Group or User Name and specify the Roles to be assigned.
4. To edit one user or group role, click on the user or group **Edit** link.
5. To delete one or more roles, tick the checkbox and click on **Delete**.



45 Defining Role-based security for the Reports Homepage

Follow the steps below to define the roles for the home page:

1. In the homepage, click on the **Properties** tab.
2. To add a new role, click on the **New Role Assignment**, and then enter *Group* or *User Name* and specify the Roles to be assigned.
3. To edit one user or group role, click on the user or group **Edit** link.
4. To delete one or more roles, tick the checkbox and click on **Delete**.



46 Defining Security by Folder

By default, each folder inherits the security definitions from the parent folder. It is possible to change this configuration by following the steps described below:

1. Click on **Folder Settings**.
2. Click on **Edit Item Security** and perform the desired changes.

It is also possible to customize each folder with different security settings. In order to do that, follow the steps below:

1. Click on the folder to change the roles.
2. Click on the **Properties** tab.
3. Click on **Edit Item Security** and perform the desired changes.

!!! note Click on **Revert to Parent Security** to inherit the father's folder roles again.



47 Critical Manufacturing Upgrade Procedure

The Critical Manufacturing Upgrade procedure is heavily dependent on the level of customization of the system and must always be validated and rehearsed in a staging system. When upgrading between major versions, it's expected that some downtime is required. When upgrading between minor versions of the system it might be possible to upgrade without downtime depending on the customer requirements. While the database is being upgraded incoming requests might block waiting for the upgrade to complete and this might have impact on the system normal operation. When applying an update, it's not expected any downtime.

Information on the upgrade procedure for the Critical Manufacturing MES is available on the [Developer Portal](#).

47.1 Upgrade an MES Customer Environment

Upgrading a Customer Environment is a simple process in the [DevOps Center](#). Having an existing Customer Environment, the process relies on creating a new version and changing what is necessary before deploying. What changes in an upgrade is completely your choice.

By creating a new version, changing configurations and finishing by deploying, a new stack will be created in the cluster with the newly applied changes.

To exemplify, you should continue from the [\[installation-guide-installation#step-1-create-an-environment|Create Environment\]](#) section of the Guide and upgrade that environment to a Critical Manufacturing MES v8.3.3. This will create a new environment and the actual system will also upgrade the database to v8.3.3 during the installation.

1. Start by selecting **New Version** in the main page of the Customer Environment.

New Version button on the Customer Environment main page

2. In the wizard, nothing has to be done. However, it is possible to change the **Description** and opt-in to clear the parameters. Be careful as this last operation clears all configurations for the new version. Select **Create** and the new version is created in the system.

3. The next step is to change Critical Manufacturing MES from v8.3.2 to v8.3.3. In the installation page, change the **Deployment Package** to [MES 8.3.3](#).

Changing the Deployment Package to MES 8.3.3 in the installation page

4. The version is now changed from v8.3.2 to v8.3.3. This is enough for the upgrade to happen after triggering the deployment process. However, if using a customization package and if the upgrade also comprehends a change to that package, it is necessary to change it. In that case, update the value of **Package to Install** in the **General Data** step.

Updating the Package to Install field in the General Data step

!!! warning If consuming local packages for Critical Manufacturing MES Environment Manager, such as a customization package, do not forget to add those to the location that the Boot Packages Folder points to.

5. In case the upgrade also comprehends changing any configuration or enabling a new feature, you can now do that. What you have to do is select **Next** and wait until the execution jumps to the Deployment step, and then wait until the deployment and installation finishes. You are advised to turn on Critical Manufacturing MES Maintenance Mode before upgrading to ensure that if the new version has integration entries to be processed, these are not handled by the old Critical Manufacturing MES that is still running as this could result in errors or data inconsistency.

To turn on the Maintenance Mode, go to Critical Manufacturing MES, select **Administration** followed by **System Monitoring**. On that page, there is a button named **Set Maintenance Mode**. Select it.

Set Maintenance Mode button in the System Monitoring page

Selecting **Active** will turn on the Maintenance Mode. A notification is then displayed in the bottom-left corner:

Notification confirming that Maintenance Mode is active



6. After starting the Deployment, the process is the same; it is automatic and feedback is presented.

Deployment progress and status for the new MES version

After a short time, the Deployment should finish with success (upgrades take longer than clean installs). This indicates that the new stack was successfully deployed and Critical Manufacturing MES was upgraded.

7. Lastly, you have to deactivate the Maintenance Mode. Once again, go to Critical Manufacturing MES, select **Administration** followed by **System Monitoring**. Select **Set Maintenance Mode** and then select **Deactivate**.

Deactivate option for Maintenance Mode in System Monitoring

After deactivation, you are presented with a notification regarding the new version that is available:

Notification indicating that a new MES version is available after upgrade

Selecting the new version will refresh Critical Manufacturing MES, which will load any new changes that came from the upgrade.



48 Uninstall

To uninstall the Critical Manufacturing software, follow the steps below:

1. Follow the steps on the Customer Portal support website [here](#) to uninstall/terminate a MES Environment.
2. Manually remove all the Windows-specific Services. To know specifically which services need to be removed, go to [System Architecture](#), scroll down to the **Optional Components** and see each corresponding service.
3. Remove the databases
 - **Online**
 - **ODS**
 - **Data Warehouse**
 - **Analysis Services**
4. Remove the MES Reports from SQL Server Reporting Services.
5. Open the SQL Server Agent, go to **Jobs** and click on **Job Activity Monitor** to remove all related jobs.
6. Go to **Linked Servers** (in the instance, go to **Server Objects** and **Linked Servers**) and remove all related linked servers.



Legal Information

Disclaimer

The information contained in this document represents the current view of Critical Manufacturing on the issues discussed as of the date of publication. Because Critical Manufacturing must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Critical Manufacturing, and Critical Manufacturing cannot guarantee the accuracy of any information presented after the date of publication. This document is for informational purposes only. Critical Manufacturing makes no warranties, express, implied or statutory, as to the information herein contained.

Confidentiality Notice

All materials and information included herein are being provided by Critical Manufacturing to its Customer solely for Customer internal use and for its business purposes. Critical Manufacturing retains all rights, titles, interests in and copyrights to the materials and information herein. The materials and information contained herein constitute confidential information of Critical Manufacturing and the Customer must not disclose or transfer by any means any of these materials or information, whether total or partial, to any third party without the prior explicit written consent by Critical Manufacturing.

Copyright Information

All title and copyrights in and to the Software (including but not limited to any source code, binaries, designs, specifications, models, documents, layouts, images, photographs, animations, video, audio, music, text incorporated into the Software), the accompanying printed materials, and any copies of the Software, and any trademarks or service marks of Critical Manufacturing are owned by Critical Manufacturing unless explicitly stated otherwise. All title and intellectual property rights in and to the content that may be accessed through use of the Software is the property of the respective content owner and is protected by applicable copyright or other intellectual property laws and treaties.

Trademark Information

Critical Manufacturing is a registered trademark of Critical Manufacturing.

All other trademarks are property of their respective owners.