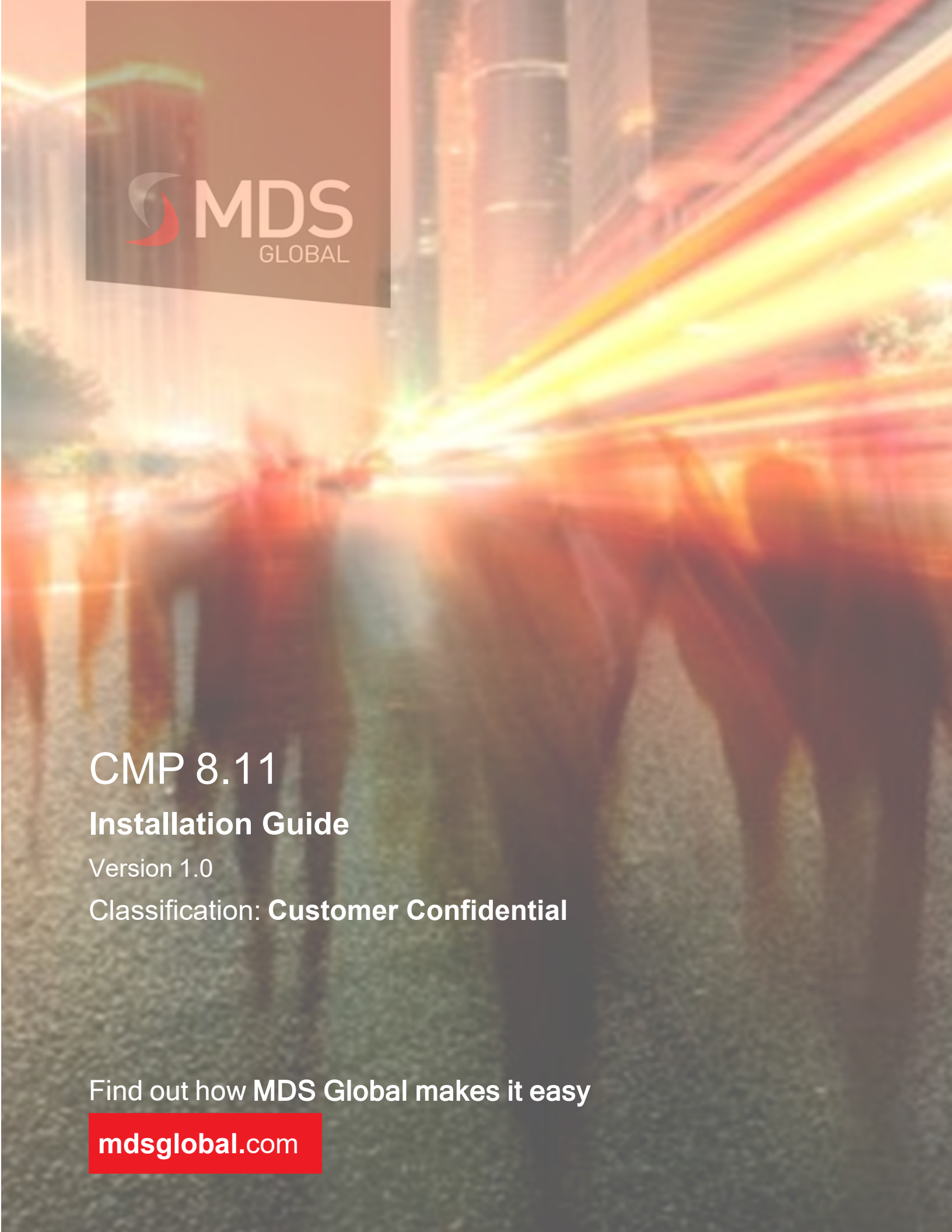


CMP 8.11

Installation Guide

Version 1.0

Classification: **Customer Confidential**

Find out how **MDS Global makes it easy**

mdsglobal.com

Copyright

© MDS Global 2023

THE CONTENTS OF THIS DOCUMENT ARE THE COPYRIGHT OF MDS GLOBAL LTD. ALL RIGHTS RESERVED. THIS DOCUMENT OR PARTS THEREOF MAY NOT BE REPRODUCED IN ANY FORM WITHOUT THE WRITTEN PERMISSION OF MDS GLOBAL.

Confidentiality

This document contains information that is proprietary to MDS Global and is confidential. The original recipient of this document may duplicate this document in whole or in part for internal distribution only, provided that this entire notice appears in all copies. This document and its contents may not otherwise be reproduced, distributed or disclosed. The recipient agrees to make every effort to prevent the unauthorised use, distribution or disclosure of the proprietary information contained in this document.

Disclaimer

No representation or warranty is contained in, made or given by this document or the information contained within it and no warranty or representation is made or to be implied that the information contained in this document is complete, up to date, accurate or fit for the purpose for which this document is supplied. In no event shall MDS Global be liable for incidental or consequential damages or loss in connection with, or arising from its use, whether MDS Global was made aware of the probability of such damages or loss arising or not.

Trademarks

The grey and red symbol above is an unregistered trademark of MDS Global Ltd. Other trademarks referred to within this document are the property of their respective trademark holders.

Contact Details

Please visit www.mdsglobal.com for further information on MDS Global products, solutions and services.

ISO 22301 standard is applicable to MDS Global Business Operations.

Table of Contents

Table of Contents ..... **ii**
Version Control iv
Terms Used in this Document ..... **v**
1.0 Assumptions **1**
2.0 About CMP Installation **2**
2. 1 RPM packages 2
2. 1. 1 Yum Repository 2
2. 2 Ansible 2
2. 3 CMP Automatic Deployment Process 4
Step 1 5
Step 2 6
Step 3 7
Step 4 8
2. 4 Encrypting Sensitive Information 8
2. 5 JVM Heap Size Allocations 10
2. 6 High Availability Stack Installation 10
2.0 Health Check Service **12**
3.0 Prerequisites **14**
3. 1 High Availability Prerequisites 16
3. 1. 1 High Availability for Reports 18
3. 2 SSL Certificates 18
3. 2. 1 Shared System Certificate Storage 20
3. 3 SABRE Server Encryption 20
3. 4 Network Communication 21
3. 4. 1 Intra Stack Communication 21
3.4.1.1 High Availability Intra Stack Communication 23
3. 4. 2 External Access 24
4.0 Pre-Installation Tasks **25**
5.0 Installation Tasks **26**
5. 1 Encrypt Sensitive Inputs in the Inventory File 26
5. 1. 1 Store Properties in an Encrypted Ansible Vault File 27
5. 1. 2 Encrypting Individual Properties 28
5. 2 Prepare the Inventory File 29
5. 2. 1 About the Inventory File 29
5. 2. 2 CMP Inventory Template 32
5. 2. 3 Installation Configuration Tool 35
5.2.3.1 Preparing an Inventory File with the Installation Configuration tool 36
5.2.3.2 Working with an Existing Inventory File 43
5. 2. 4 CMP Global Properties 44

5. 2. 5 Administration Console Properties 47
5. 2. 6 AgentView Interfaces Layer Properties 48
5.2.6.1 Additional Settings 49
5. 2. 7 AgentView Properties 50
5. 2. 8 Bulk Action Console Properties 52
5. 2. 9 Configuration Centre (Business Configuration) Properties 54
5. 2. 10 Database Properties 55
5. 2. 11 JBoss Server Properties 58
5. 2. 12 JBoss Web Server Properties 60
5. 2. 13 REST Web Services Properties 61
5. 2. 14 Pentaho Server Properties 61
5. 2. 15 Role Extender Properties 62
5. 2. 16 SABRE Server Properties 63
5. 2. 17 Shared Services Properties 65
5. 2. 18 SAM Services Properties 66
5. 2. 19 SOAP Web Services Properties 67
5. 2. 20 WSO2 Identity Server Properties 68
5. 2. 21 Advanced Property Options 68
5. 2. 22 Context Sensitive Help Properties 73
5. 2. 23 ActiveMQ (Data Export Only) Properties 73
5. 2. 24 Artemis Properties 74
5. 2. 25 SPaRC Engine Properties 76
5. 3 Deploy CMP 77
5. 3. 1 Common Ansible Command Line Parameters 78
5. 3. 2 Summary File 81
5.3.2.1 Example Summary File 81
6.0 Upgrading an Existing Installation 84
6. 1 Patch Updating 85
7.0 Troubleshooting 86
8.0 Post-Installation Checks and Tasks 91
8.0 High Availability Deployment Post-Installation Tasks 92
8. 1 Database 92
8. 1. 1 Replication 92
8. 1. 2 Archive Logging 92
8. 2 AgentView 93
9.0 Uninstall CMP 94
10.0 Appendix 95
10. 1 Appendix A: Default Ports Allocation 95
10. 2 Appendix B: Inventory Example 95
10. 2. 1 CMP Multiple Host Inventory Example 95

Version Control

Version	Issue Date	Author	Comments
Version 1.0	26 May 2023	MDS	CMP 8.11 Release - No changes since the last release.

Terms Used in this Document

For definitions and explanations of the terms, abbreviations and acronyms used in this document, please see the *CMP Glossary* document.

1.0 Assumptions

The installation and deployment instructions provided in this document are based on a number of assumptions:

- **IMPORTANT**: Those performing the installation have ***at least*** Basic Red Hat Administration knowledge, an appreciation of networking fundamentals and an understanding of Ansible automatic deployment.

  Ensure you are familiar with the concepts in "About CMP Installation" on page 2.

- For production or production grade installations, that although the database installation is automated, a suitably qualified PostgreSQL Database Administrator is associated with the configuration, installation and ongoing operation of CMP.
- A person performing a High Availability installation of CMP requires a higher level of expertise because such installations are technically more complex.
- Servers are available with the capacity and configuration as defined in conjunction with MDS Global to meet processing requirements of the installation.

2.0 About CMP Installation

CMP is delivered as a set of components that are RPM (Red Hat Package Manager) packages and it is installed and deployed by running an Ansible playbook.

The installation repository for CMP is available on the CMP host in the cloud (`https://vault.mdsglobal.dev/`), accessible via HTTPS with the credentials provided by MDS Global, according to the customer's contract. It includes the following:

- YUM and Maven repositories, which store the RPM packages and database package, as well as the Ansible Playbook package used for deployments

You need download only the Ansible playbook package to the control server. The RPM and database packages are deployed automatically when you run the playbook.

2. 1 RPM packages

The RPM Package Manager (RPM), originally called the Red Hat Package Manager, is a program for installing, uninstalling, and managing software packages in Linux. The "RPM" portion of this term comes from the fact that `.rpm` is the default extension for files used by the program.

Because CMP components are delivered as RPM packages, this dictates where in the operating system each file is deployed. It is not possible to deploy components anywhere else on the host file system.

In addition, the RPM-based deployment does not allow more than one instance of a component to a host. For example, you cannot install multiple instances of the AgentView Interfaces Layer on the same host.

For more information, see https://rpm.org/.

2. 1. 1 Yum Repository

CMP RPM packages are stored in YUM and Maven repositories.

YUM (Yellowdog Updater, Modified) is an open-source command-line package-management utility for computers using the RPM Package Manager. YUM Repositories are warehouses of RPM package files that can hold RPM packages locally or remotely.

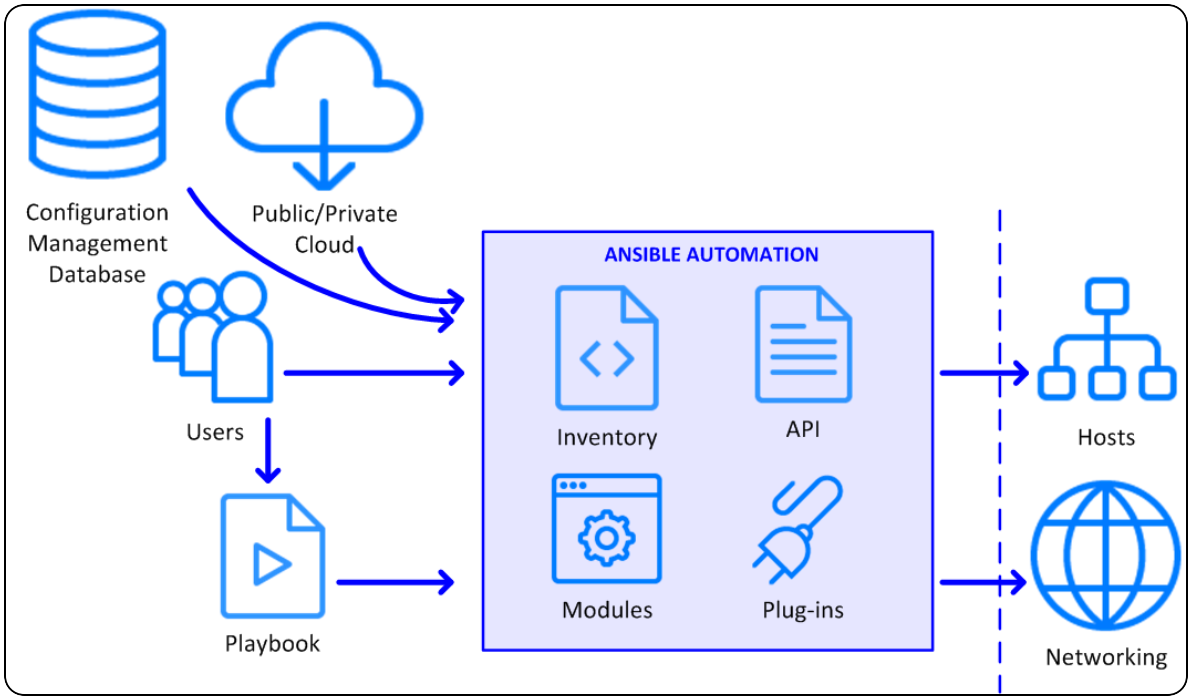
2. 2 Ansible

Ansible is a configuration management tool. Although it can be run in "ad hoc" mode to run commands manually like any command line interface tool, it is usually used to auto-

mate the manual tasks of configuration and installation, such as shutting down and restarting hosts, installing packages and components, and deploying configurations.

Ansible is agent-less and communicates with hosts using SSH (Secure Shell) for Unix/Linux and WinRM for Windows installations. For CMP purposes, SSH is used.

For more information see https://docs.ansible.com/.



Ansible Architecture

CMP deployment uses Ansible automatic installation, which involves the following concepts:

Control Server

For CMP purposes a control server is a server that runs Linux and has Ansible installed. You use it to run commands and *playbooks*.

Playbooks

A playbook is an ordered list of tasks, which are run sequentially. Playbooks can declare configurations, but they can also orchestrate steps of any manual ordered process, even as different steps must move back and forth between machines in particular sequence.

For more information, see https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html

Playbooks are written in YAML (YAML Ain't Markup Language), which is a human-readable data serialisation language that is simple to use and commonly used in configuration files. CMP provides playbooks for the installation and deployment of CMP components.

For more information, see https://yaml.org/.

Each *task* in the playbook is a unit of action and correlates to a single command that you would run in ad hoc mode. Tasks invoke one or more modules, which are the units of code executed by Ansible. Each module has a particular use, such as file management, package management and remote execution. Ansible provides an extensive list of modules for most common tasks and you can also write your own. CMP provides modules for the installation and deployment of CMP components.

For more information, see https://-docs.ansible.com/ansible/latest/modules/modules_by_category.html.

Inventory File

An inventory file is a list of the target hosts (or *managed nodes* in Ansible terminology). That is, they are the machines on which you want to install the CMP components. The inventory file uses variables (properties) to describe the desired state of the target hosts. The hosts can be grouped and nested, usually according to their role or function, such as database or web server, or load balancer, for example. A number of host groups have been predefined for CMP; see the *Groups* table in *Prepare the Inventory File* for more information.

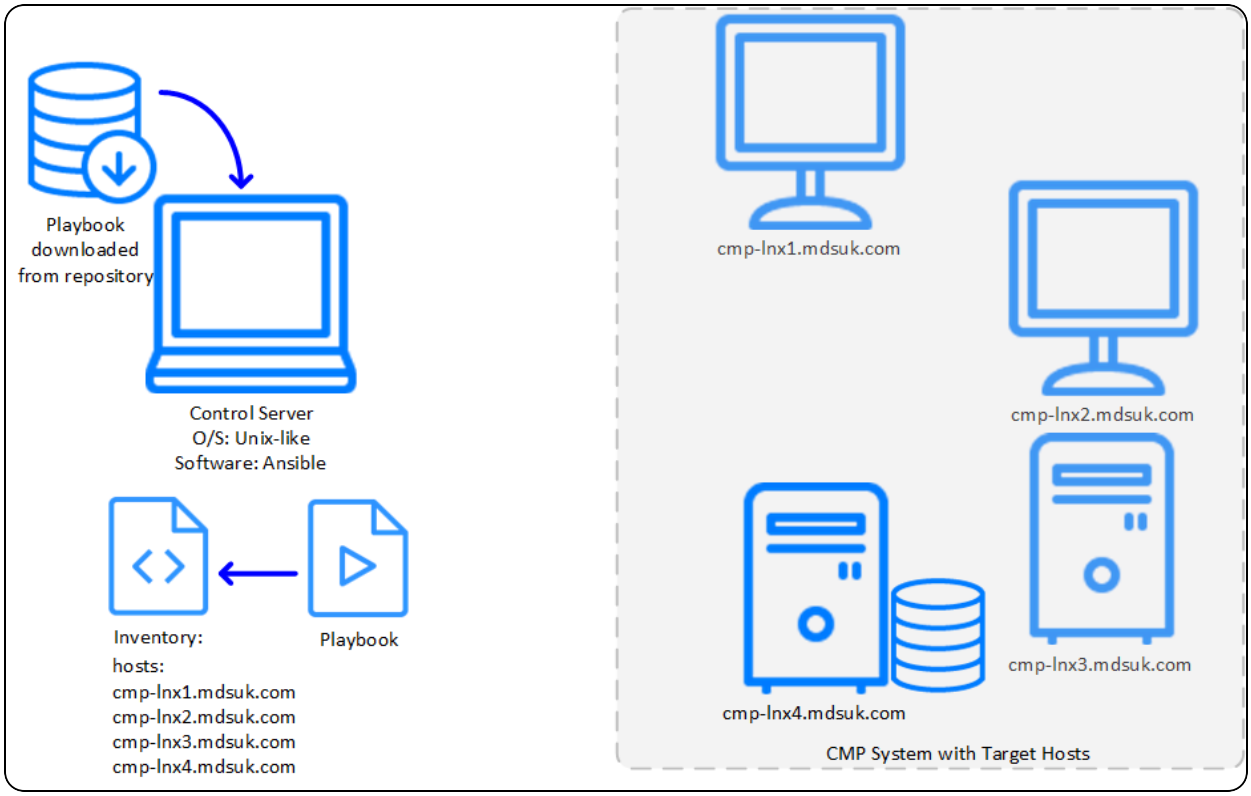
CMP provides a YAML inventory file template (`template.yaml`) that you must copy and edit to suit your deployment.

For more information, see "Prepare the Inventory File" on page 29

2. 3 CMP Automatic Deployment Process

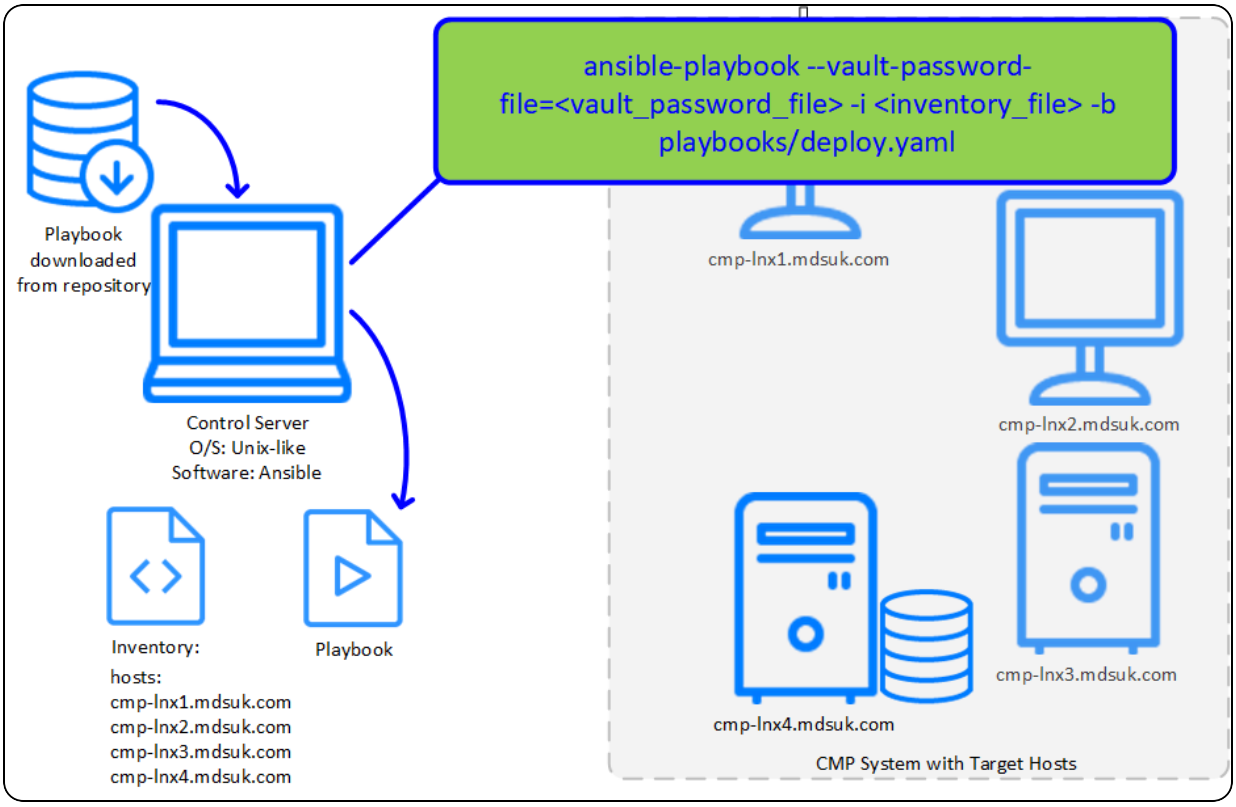
The Ansible deployment process used to deploy CMP is as follows:

Step 1



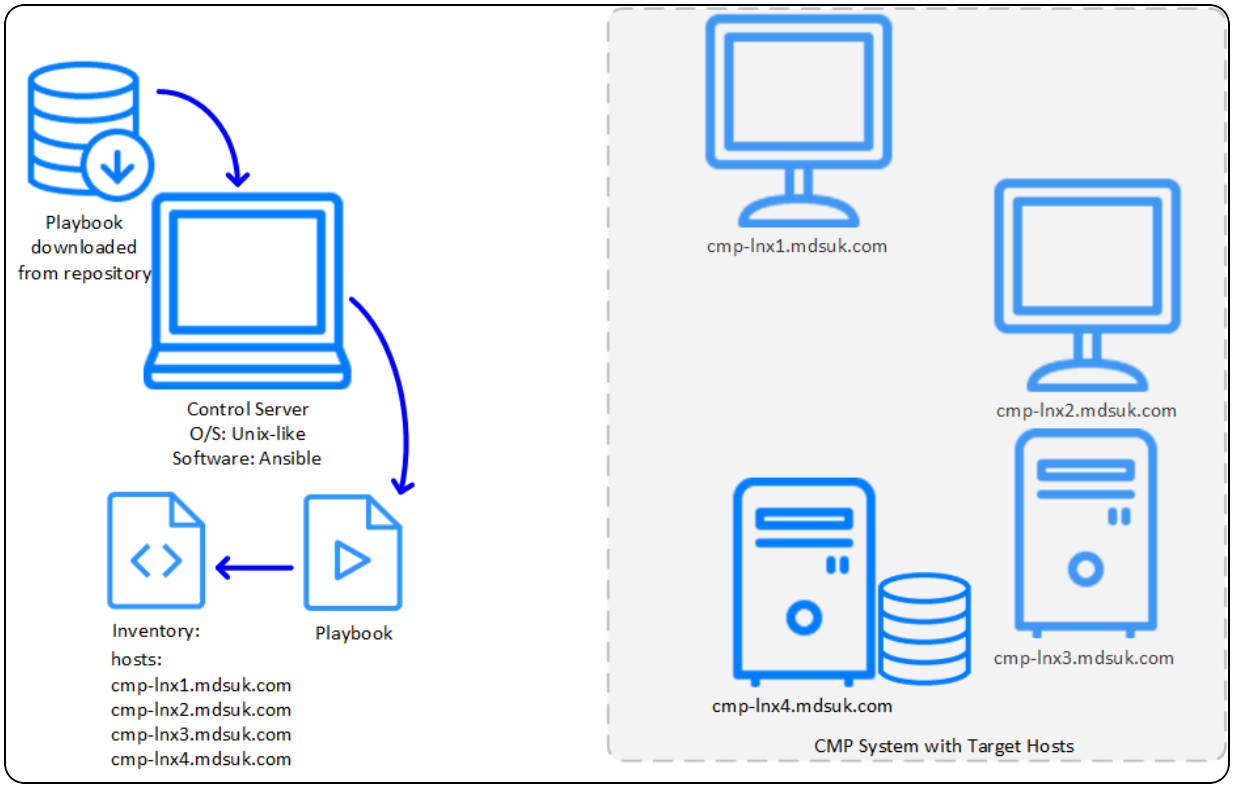
Download and unzip the Ansible Playbook package to the control server (controller node) and prepare the inventory file by supplying the host names and configuration properties

Step 2



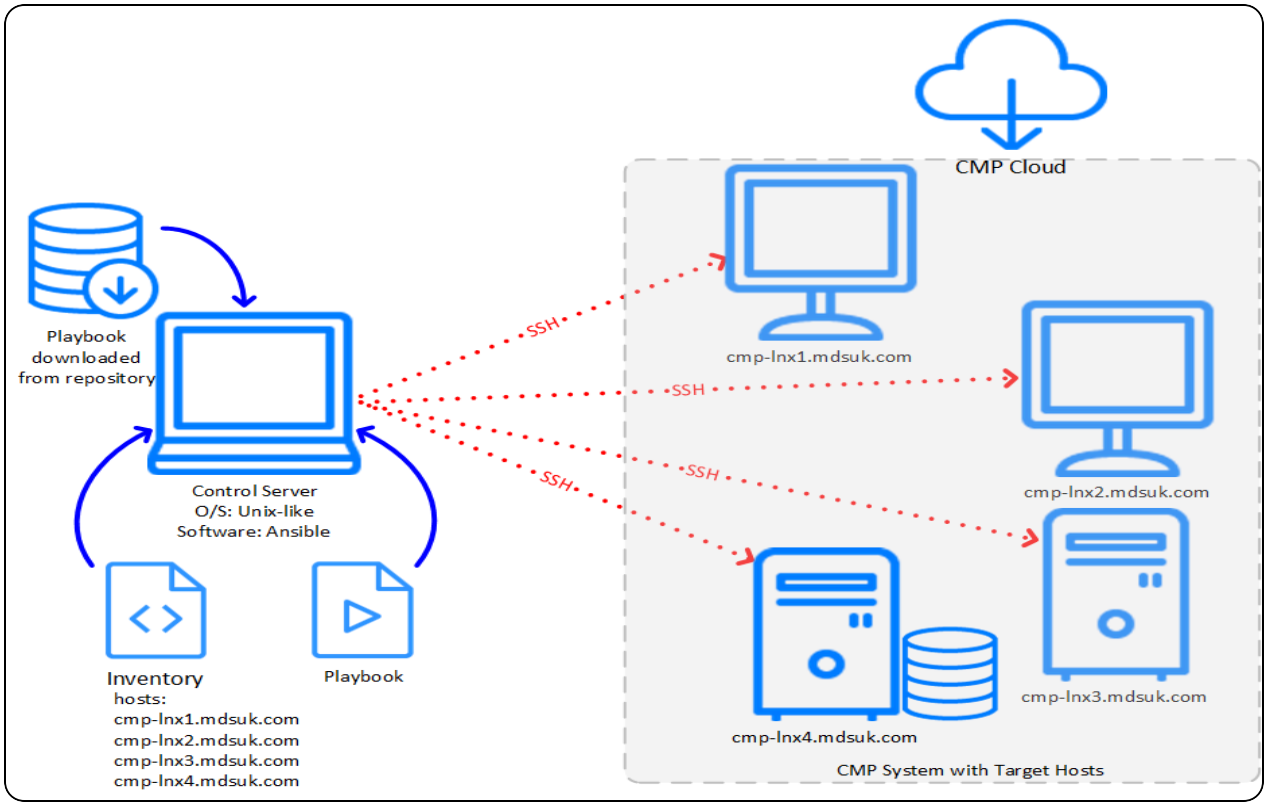
Run the command to execute the deployment playbook.

Step 3



As Ansible runs step by step through each task in the playbook, it encounters deployment instructions for groups of hosts. It consults the inventory to determine the list of specific hosts and what configuration is required for each.

Step 4



Having identified the hosts, the control server communicates the configuration and deployment instructions to the hosts via SSH. The hosts access the CMP installation repository in the cloud and deploy the required packages and configuration.

2. 4 Encrypting Sensitive Information

Ansible Vault is an Ansible feature that enables you to keep sensitive data such as passwords or keys in encrypted form. The sensitive data can be stored in *vault* files, where the whole file is encrypted, or individual strings and values can be encrypted directly in the inventory files. To encrypt files and properties, you use the command-line tool `ansible-vault`.

The `ansible-vault` program uses a password to encrypt files and properties. When `ansible-vault` is executed, it prompts the user to type in the password. Alternatively, a user can create a plain text file where the vault password is stored and then use the `--vault-password-file` parameter that points to the text file with the password to suppress the password prompt.

When the encrypted files or properties are used in the property hierarchy, one of the following command line flags should be used when running the `ansible-playbook` program: `--ask-vault-pass` or `--vault-password-file`.

The `--ask-vault-password` parameter forces the `ansible-playbook` program to prompt the user to enter the vault password and like `ansible-vault` above, the `--vault-password-file` property points to the plain text file where the vault password is stored.

2. 5 JVM Heap Size Allocations

It is possible to adjust the allocated JVM heap size for individual CMP applications using the `<application>_allocated_heap_size` property, for example the `jws_allocated_heap_size` property for the JBoss Web Server.

However, when choosing the minimum and maximum heap sizes for a CMP module, it is essential not to oversubscribe the available physical memory on any particular host.

You must ensure that the sum total of the maximum heap sizes for all of the CMP modules on a particular host does not exceed the memory available to CMP on the host. You can calculate available memory by subtracting the memory required for all non-CMP processes (for example, the memory requirement for the OS) from the total available memory on the host itself.

2. 6 High Availability Stack Installation

Most production-type deployments require that all CMP components are highly available[1] .

This means that:

- More than one instance of each component is configured.
- Multiple instances are clustered; that is, they can transfer state in the real time between instances, or the load balancers are configured to use a single instance at the time.
- These multiple instances of the software are load balanced, that is, the client (or other component) communicates with the load balancer instead of the specific instance of the application. The load balancer can distribute the load between available instances of the application in question and it can redirect the load to other instances in the stack in the case of the failure.

  The above point applies to only clustered components.

Important:

Before you install a high availability deployment, read the section High Availability in the *CMP Technical Architecture Overview*, which describes the requirements and responsibilities of those who will install and maintain a high

[1] High availability is a characteristic of a system, which aims to ensure an agreed level of operational performance, usually uptime, for a higher than normal period.

availability system. These requirements and responsibilities must be met for a successful deployment.

In addition, note that configuring the replication and the load balancing is not part of the CMP deployment and must configured using relevant third party software and hardware as a separate step to the installation itself.

The CMP stack can be installed in different configurations:

- Simple installation, when each application component has only one instance.
- Highly available, when all the components of the stack are highly available
- Mixed, when some application components are highly available and some are represented by the single instance.

CMP provides a global health check service for use alongside an external load balancer if required; see "Health Check Service" on the next page.

2.0 Health Check Service

CMP provides a global health check service for use alongside an external load balancer if required. The service first checks the `cmpModuleMaintenance` table for a valid entry before then calling the module specific health check URI. This allows for modules to be placed into maintenance mode. This can be used by the load balancer and allows any HA deployment to place modules into maintenance mode so that they appear unavailable on the load balancer when upgrading.

The global health check service is deployed alongside the following modules:

- agent-view
- agent-view-interfaces-layer
- agent-view-servlet
- bulk-action-console
- configuration-centre
- published-interfaces-layer
- rest-ws
- role-extender
- sabre-console
- sabre-server
- soap-ws
- wso2is
- wso2is-login

It is deployed on the same port (default of 21215) across all of the hosts on which the CMP application is installed.

The global health service can be accessed through the following URI:

`https://${hostName}:21215/healthCheck/${moduleName},`

where `${hostName}` is the host on which the module is deployed and `${moduleName}`is the name of the specific module. For example:

```
https://alpha23.int-dev.mdsuk.com:21215/healthCheck/agent-view
```

The service returns a response such as:

```
{ "status" : "UP" },
```

where "status" can be one of "UP", "DOWN", "MAINTENANCE", or "UNKNOWN", for example:

The status of "UNKNOWN" will only be returned for a module that does not have an application specific health check defined.

The global health service is configured in the Global Properties in the inventory. See "CMP Global Properties" on page 44 and "Installation Configuration Tool" on page 35 for more information.

3.0 Prerequisites

Before starting deployment ensure that the following are in place:

- A control server[1] is available that runs Unix, is connected to the network and has Ansible version 2.9 or later and a version of the unzip utility installed.
- Red Hat Enterprise Linux is installed and updated on each target host.

  For more information, see *Third Party Software Versions* in the *CMP Technical Architecture Guide*.
- The time on all servers being used is synchronised and set up using Network Time Protocol.
- The timezone on all the servers is set to the timezone in which users expect to schedule activities and to be used to trigger processing of events. This is usually the local timezone of the location where CMP is being used.
- The intended target hosts and the control server are connected to a network.
- Each intended target host can be accessed by their external hostname which is also known to the host in question; that is, the name that will be used by the end users and support staff to access the target hosts from outside of the CMP stack. You can achieve this by:
  - Adding the respective hostname/IP address pairs to the DNS server.
  - Adding the respective hostname/IP address pairs to the `local hosts` file on the control server and all target hosts.
- The target servers can communicate with one another over the ports required for the intra stack communications (see the table in "Network Communication" on page 21). If there are firewalls between the target servers other than the standard Linux firewall (iptables or firewalld), the communications described in the intra stack communications table must be allowed.
- The control server can communicate with the target host via SSH.
- The user account that will be used by Ansible is created on each target host. This account must have:
  - Either password access or the client public key added to list of the authorized keys for the user in question, allowing key-based authentication for the user.

    Ensure that you can log into the target hosts over SSH from the control server with the password or the private key prior to starting deployment.

    Ensure that the corresponding server keys are added to the `known_hosts`

[1] A computer on a network used to access and manage devices. For Ansible purposes, a control server runs Linux/Unix and has Ansible installed.

file on the control server.

- sudo[1] access (with or without a password).

- The customer has requested access to the MDS Global repository. The request must include the public IP addresses (IP address range) of the control server and the target hosts. MDS Global will whitelist the provided address(-es), allowing connectivity to the repository and will provide set of credentials that must be used both to download the installation scripts and to enter in the inventory file.

  For more information on the inventory file, see "Prepare the Inventory File" on page 29.

- For Production systems it is recommended to create separate filesystems for the database instance otherwise the database tablespace files will be stored in the “root” Linux filesystem. By default, the following locations are used to store the database data:
  - /pg_wal
  - /pg_logs
  - /pg_temp
  - /var/lib/pgsql
  - /pg_data/mdscmp/pg_index_data
  - /pg_data/mdscmp/pg_small_data
  - /pg_data/mdscmp/pg_large_data
  - /pg_data/mdscmp/pg_archive_data
  - /pg_archivelogs

  The last two filesystems above are ideally suited to slower storage devices and do not need the cost associated with fast disk / SSD.

  RedHat administration knowledge is needed to perform the creation of filesystems.
  PostgreSQL Database Administration knowledge is needed to correctly configure, manage and maintain the database for a production system.

- The required SSL certificates have been generated.

  For more information, see "SSL Certificates" on page 18 .

- Security certificates for third parties are stored in the correct subdirectory of the default system-wide trust store.

[1] A program for Unix-like computer operating systems that allows users to run programs with the security privileges of another user, by default the superuser.

This is because Enterprise Linux uses Shared Systems Certificates storage. See "Shared System Certificate Storage" on page 20.

- One of the following:
  - The target hosts are either registered with Red Hat, which allows management of the YUM repositories via Subscription Manager, with an active attached subscription for the YUM respositories required for each host.
  - OR, the target system has access to the hosted YUM repositories with JBoss and JBoss Web Server RPM packages, in the supported versions as per *Third Party Software Versions* in the *CMP Technical Architecture Guide*:
    - All target hosts require access to the Red Hat Enterprise Linux Server product subscription.
    - JBoss target hosts also require access to the Red Hat Enterprise Application Platform product subscription.
    - JBoss Web Server target hosts also require access to a product subscription for JBoss Web Server.

      See the note below:

Important

The deployment process relies on access to the RedHat YUM repositories (for example, jb-eap-7.2-for-rhel-7-server-rpms/jws-5-for-rhel-7-server-rpms) or their equivalents.
Access to the JBoss Enterprise Application Platform repository is only required on the server that hosts JBoss and related application components (jboss Ansible inventory group of hosts). Similarly, access to the jws repository is only required on the server that hosts JBoss Web Server and related application components (jws Ansible inventory group of hosts). Corresponding subscriptions must be active on the servers belonging to the respective Ansible inventory group.

All target hosts require access to the 'Red Hat Enterprise Linux Server' product subscription or equivalent hosted YUM repository.

3. 1 High Availability Prerequisites

- Before commencing the HA stack deployment, an instance of the Load Balancer must be configured for each highly available software component of the CMP stack (except Database).
- The load balancer URLs must be predefined because this information needs to be added to the inventory file either manually or via the Installation Configuration tool (recommended).

- When using a load balancer with CMP, the chosen make and model must support the following:
  - WebSockets - required for the correct functioning of AgentView/Webswing.
  - Health Checks - to determine the status of a service.
  - Varying Timeouts - timeouts can vary from short (seconds) to long (hours), needed to support long session timeouts with AgentView.
  - Session Affinity - for services such as AgentView and Administration Console that have interactive user interfaces requiring a connection to a unique instance for the session.

By default, Ansible expects the shared storage to be mounted as follows:

- Shared store should be provided and mounted to the all nodes that host an Identity Server instance to the `/opt/cmp-storage/wso2is` directory.
- Shared storage for the SABRE server (batch server) should be mounted to `/var/m-dsglobal/sabre-server` directory on all instances that host SABRE Server.
- Shared storage for the Report Server (Pentaho server) should be mounted to `/var/pentaho-solutions` directory on all instances that host Report Server. (See "High Availability for Reports" on the next page).

For the load balancer it is necessary to create the virtual services before an HA CMP installation takes place because URL access is checked during the installation process. The following table contains an example set of load balancer services:

Service Name	Protocol	Port / Healthcheck	Timeouts	Affinity
AgentView / Pentaho	HTTPS	Port 7443	28,800 seconds (8 hours)	Auto generated session Cookie
REST	HTTPS	Port 9000	30 seconds	None
SOAP	HTTPS	Port 8443	30 seconds	None
Config Centre	HTTPS	Port 8443	30 seconds	Client IP
WSO2	HTTPS	Port 9443	30 seconds	Client IP
Role Extender	HTTPS	Port 8081	30 seconds	None
SABRE Console	HTTPS	Port 31212	30 seconds	Client IP
ActiveMQ (Local to SABRE)	TCP	Port 61616	30 seconds	None
Bulk Action	HTTPS	Port 9009	30 seconds	Client IP

For the load balancer services DNS names must be configured in the CMP inventory. As described above these are needed BEFORE the installation of CMP is started.

Note:

If host and path rules are used in the load balancer to direct traffic to the correct services, only DNS subdomain configurations are supported and not path rules.

For example the following are valid:

```
wso2is.cmp8.com

soap-services.cmp8.com
```

However, the following are invalid:

```
www.cmp8.com/wso2is

www.cmp8.com/soap-services
```

This is because certain services such as WSO2IS have in-built redirects from the root context(/) that override any paths that have been configured as part of the load balancer rules.

3. 1. 1 High Availability for Reports

Recommendations:

- Report Server (the Pentaho server) should be deployed to all AgentView hosts. This must be manually configured in the Inventory Configuration tool.
- Report Server should make use of shared storage as described above. This reduces the confusion/complexity of report output being stored on multiple hosts
- Run Report Server in an Active/Active state so that end users do not need to consider switching on/off schedules between active/passive nodes as needed.

3. 2 SSL Certificates

By default, all CMP components communicate over HTTPS protocol. Before commencing the deployment, the required SSL certificate or certificates should be obtained from the Certification Authority.

The certificates must meet the following criteria:

- The certificate common name must match the target host name; the name that the other components and the users will use to access it.
- The certificate expiration date must be in the future.
- The certificate must be signed by the recognised or trusted Certification Authority.

If the last criterion is not met, the playbooks will still be able to establish trust by including the certificate in question into their trusted certificate list. However, if the first two criteria are not met, CMP will not function properly.

If only the last criterion is not met, end users will be notified by their browsers that certificate is not trusted, but they will be able to use the applications.

In most deployment scenarios, the components of the CMP stack will be deployed to a number of hosts. To meet the first criteria, in this case, a certificate must be obtained for each host. Alternatively, you can obtain a *wildcard* certificate.

The wildcard certificate's common name includes a wildcard and is usually produced to secure the whole domain of hosts.

For example, the certificate with the common name `demo.test.com` can only be used for the demo.test.com host. However, a wildcard certificate with the common name `*.test.com` can be used for any host in the test.com domain, for example test.com, demo.test.com or demo1.test.com.

Users have the option not to produce the SSL certificates issued by the Certificate Authority. In this case, the self-signed certificates will be generated by the deployment.

The use of the self-signed certificates is not recommended for production environments.

If you want to use a *real* or previously generated self-signed certificate, you must include two properties into the inventory file:

```
ssl_certificates.private_key
```

```
ssl_certificates.certificate
```

This is a dictionary property (hierarchical). It is as follows:

```
ssl_certificates:
```

```
private_key: <path to private key>
```

```
certificate: <path to certificate>
```

The property value should be a valid absolute path at the control server, where the private key and certificate (in PEM format) are stored.

Note that this is the path on the *control server*, that is, the server where the deployment is running from.

When a single (wildcard) certificate is used for all servers the `ssl_certificate` property can be set at the global level (`all->vars` section). However, if different host-specific certificates will be used, they should be defined at the corresponding group level.

When the `ssl_certificate` property is not defined, the Ansible playbook will generate a self-signed certificate for each of the concerned services.

3. 2. 1 Shared System Certificate Storage

Enterprise Linux uses Shared System Certificates storage, which allows NSS (Network Security Services), GnuTLS (GNU Transport Layer Security), OpenSSL (Open Secure Sockets Layer), and Java to share a default source - or *trust store* - for retrieving system certificate anchors[1] and blacklist information. This includes the SSL certificates that CMP uses for authentication with the third party software with which it is integrated.

The consolidated system-wide trust store is located in the following directories:

- `/etc/pki/ca-trust/`
- `/usr/share/pki/ca-trust-source/`

The CMP Ansible installer calls `update-ca-trust`[2]. Briefly, this command searches for certificates and trust settings in the subdirectories of the directories above and creates an extract of consolidated configuration files, which is output to the following directory: `/etc/pki/ca-trust/extracted`, where compatible applications can read the files.

To accommodate legacy applications that might expect certificates and trust configuration in a fixed location, contained in files with particular path and name, the classic file names are changed to symbolic links in the output. The symbolic links refer to dynamically created and consolidated output stored in the trust store directory hierarchy.

So you must select the correct subdirectory for adding files because the subdirectory defines how certificates therein will be trusted or distrusted, and which file formats are read. Follow the methodology described in this Linux Security Guide topic.

For more information about the `update-ca-trust`[3] command, consult the Linux Manual Page.

3. 3 SABRE Server Encryption

CMP is capable of encryption of all outgoing files and decryption of all incoming files using PGP encryption following the OpenPGP standard (RFC 4880) for encrypting and decrypting data. PGP is an asymmetric algorithm meaning it requires a key pair (public and private) to support encryption and decryption. Payloads are encrypted using the

[1] In cryptographic systems with hierarchical structure, a trust anchor is an authoritative entity for which trust is assumed and not derived. In X.509 architecture, for example, a root certificate is a trust anchor from which the whole chain of trust is derived.
[2] The Linux command used to manage a consolidated and dynamic configuration feature of Certificate Authority (CA) certificates and associated trust.
[3] The Linux command used to manage a consolidated and dynamic configuration feature of Certificate Authority (CA) certificates and associated trust.

public key and can only be decrypted using the matching private key along with a pass-phrase. There are many tools available which implement the OpenPGP standard and allow for the generation of keypairs and encryption/decryption of data using these keys. GNU Privacy Guard (GnuPG or GPG) is one such freely available and commonly used tool that you may wish to use.

When CMP is installed encryption is configured for all files and default public and private keys are installed. Those keys reside on each SABRE host in the following locations:

- Public key used for encrypting files: `/etc/mdsglobal/sabre/gpg/pubring.gpg`
- Private key used to decrypt files: `/etc/mdsglobal/sabre/gpg/secring.gpg`

The password to use in conjunction with the private key for decryption is `notsosecretpassword`.

Important

The default keys are designed for test purposes only. Before processing sensitive data through CMP or using CMP in production it is the responsibility of operators of CMP to update the properties in the Administration Console to use keys that have been handled with appropriate security since generation. For enhanced security different key pairs should be configured for each CMP module rather than relying on a single system-wide setting as per the default deployment. When changing key pairs, different file names should be configured rather than overwriting the default key files above, as the default files will be overwritten by a CMP upgrade.

3. 4 Network Communication

3. 4. 1 Intra Stack Communication

It is important that components of the CMP stack can communicate with one another. The Intra Stack Communications table describes which components require access to the other components, over which port. If there are firewalls (other than standard Linux firewalld or iptables firewalls, as these are updated by the CMP installation) used between target hosts, the communication over the specified ports must be allowed by the firewall. The table shows default ports as described in "Appendix A: Default Ports Allocation" on page 95. If any of the ports are changed in the inventory file, the new value must be used when configuring firewalls.

Access must be allowed for each server in the corresponding Ansible Source hosts group.

CMP Ansible Host Groups	
Group	**Description**
db	Database server. Currently, the deployment playbook supports only single database server deployments
wso2is	Identity server. CMP uses WSO2 Identity Server for this purpose
rolext	Role extender application
jboss	JBoss server instances. Required for aview_iface, pil, soap and configuration_centre groups.
jws	JBoss Web Server instances. Required for agent_view group
aview_iface	Agent View Interfaces Layer. If the host is listed in this group it must also l be listed in the jboss group
pil	Published Interfaces Layer application
soap	SOAP Web Services application
restws	REST Web Services application
agent_view	Agent View application instances
sabre_server	SABRE Server application instances
sabre_console	Administration Console application instances
configuration_centre	CMP Business Configuration
pentaho	The Report Server is an instance of a Pentaho server
bulk_action_console	The Bulk Action console

Intra Stack Communications		
Source Hosts Groups	**Destination Host Groups**	**Port Numbers**
wso2is jboss rolext sabre_server sabre_console restws pentaho bulk_action_console	db	5432
aview_iface agent_view pil rolext sabre_console restws configuration_centre bulk_action_console	wso2is	9443
aview_iface pil	rolext	8081
sabre_server sabre_console bulk_action_console	sabre_console sabre_server	27300 27400 22

3.4.1.1 High Availability Intra Stack Communication

Communication between components in a High Availability deployment is more complicated than described in the table above. Services will communicate via the load balancer and connectivity to and from the load balancers is required, for example you must ensure the necessary ports are open to allow communication. Determining the exact network connectivity requirements in a High Availability deployment is the responsibility of the person performing the installation, based on the load balancer and shared storage configuration they have set up. See also "Assumptions" on page 1 and "High Availability Prerequisites" on page 16

3. 4. 2 External Access

To provide user access to CMP, communication over the following ports must be allowed from the client networks. If the `use_ssl` property is set to `true i`n the inventory file, the HTTPS ports listed need to be allowed, otherwise the HTTP ports must be allowed:

Target Host Groups	HTTP Port	HTTPS Port
jws (hosting AgentView, WSO2 Login)	7080	7443
jboss (hosting Business Configuration, SOAP Web Services, etc.)	8080	8443
wso2is (hosting WSO2 and Role Extender)	9763	9443
sabre_console	31212	31212
sabre_server	21212	21212
rewstws	9000	9000
bulk action console	9009	9009

The table again shows default ports as described in "Appendix A: Default Ports Allocation" on page 95. If any of the ports are changed in the inventory file then access needs to be allowed to the newly specified ports instead.

In addition to the SSH access described earlier, the control server also needs to have the above “External Access” to the target hosts.

4.0 Pre-Installation Tasks

Before you install CMP:

- Ensure that the prerequisites are in place. (See "Prerequisites" on page 14)
- Ensure that you are familiar with CMP Ansible automatic deployment concepts.

  See "About CMP Installation" on page 2.
- In the case of a CMP version upgrade, ensure that any associated third-party upgrade steps, for example for the PostgreSQL database, are completed as described in the release notes of the software version in question.
- Make a copy of the current Virtual Machine(s) (or snapshot) prior to installation/upgrade. This allows for the environment to be restored in the event that a rollback is required.
- Download the CMP Ansible installation files from the CMP installation repository in the cloud to the control server:

  The files are in a zip package that stores the Ansible deployment playbooks, inventories and other files:

  1. Download the Ansible Playbook package using the '`wget`' command and passing in your credentials as supplied by MDS Global. For example:

```
wget --user=<user.name> --ask-password https://vault.mds-
global.dev/maven/ga/latest/com/mdsglobal/cmp/ansible/cmp-ansible-playbook-pkg.zip
```

  1. Unzip the downloaded package. For example:

     ```
     unzip cmp-ansible-playbook-pkg.zip
     ```

     This step should be carried out for each installation/upgrade event to ensure the person executing the event has the latest copy of the installation files.

Important: Deploying a previous specific version other than the latest publicly available version must be agreed with MDS Global to ensure that the installation is covered as part of the contracted support and that the correct version of the Ansible Playbook scripts is provided.

5.0 Installation Tasks

To install CMP, you must:

1. Encrypt Sensitive Inputs for the Inventory File using Ansible Vault
2. Prepare an Inventory file
3. Deploy CMP
4. Upgrading an Existing Installation

Important

It is strongly recommended that you use the Installation Configuration tool provided by MDS Global for easier, quicker preparation of the inventory file. See "Installation Configuration Tool" on page 35.

If you intend to use the tool, the information provided in the following sections is for background information only:

- "Encrypt Sensitive Inputs in the Inventory File" below
- "About the Inventory File" on page 29
- "CMP Inventory Template " on page 32

However, if you have strong reasons for preparing the inventory file manually, the sections listed here are required reading. Manually preparing the inventory should be attempted only by persons experienced in YAML, who fully understand the consequences of the actions they take when preparing the file.

5. 1 Encrypt Sensitive Inputs in the Inventory File

Ansible Vault can be used to encrypt properties to be populated in the inventory file, as described in "Encrypting Sensitive Information" on page 8. Whether properties should be encrypted in the inventory file depends on the individual deployment and how and where the file is to be stored. For example, if the inventory file is very securely stored with access to very few authorized users, you may not need to encrypt any information. It also depends on the property itself, for example you may choose to encrypt passwords but not server names. In addition, not all passwords are external. The only external password is the `wso2is_admin_password` property. The rest of the credentials are either only used by installer or by the applications for internal communications. For example, any password property with `_jdbc_` in the middle is internal and used by the application to authenticate against the database.

To encrypt properties either create an Ansible Vault file, store sensitive data in it and reference it from the inventory file or individually encrypt sensitive strings for inclusion directly in the inventory file:

5. 1. 1 Store Properties in an Encrypted Ansible Vault File

To create or edit existing vault files (encrypted files), you can run the following command(s):

- `ansible-vault create --vault-password-file=~/.vaultpwd <vault_file>`

  Ansible retrieves the password from the password file and opens the new file in the chosen editor. Once the file is created, it is saved as an encrypted file.

  > The editor is defined by the `$EDITOR` environment variable.

- `ansible-vault create --ask-vault-pass <vault_file>`

  Ansible prompts the user for the vault password. If it is correct, it opens the new file in the chosen editor. Once the file is created, it is saved as an encrypted file.

- `ansible-vault edit --vault-password-file=~/.vaultpwd <vault_file>`

  Ansible retrieves the password from the password file, decrypts the encrypted file and opens a temporary file in the chosen editor .Once the file is edited, it is re-encrypted and saved as an encrypted file.

- `ansible-vault edit --ask-vault-pass <vault_file>`

  Ansible prompts the user for the vault password. If it is correct, it decrypts the encrypted file and opens a temporary file in the chosen editor. Once the file is edited, it is re-encrypted and saved as an encrypted file.

The decrypted ansible vault file is a standard YAML file which is very like the standard Ansible vars file, for example:

```
---
# vault file for sabre-server

enc_sabre_user: sabre
enc_sabre_password: sabre
```

Once the vault file is saved and you quit the editor, `ansible-vault` encrypts the content of the file, for example:

```
$ANSIBLE_VAULT;1.1;AES256
336336663437633530636534623064623264396338643930666332623239313334396435396339363862386130356663393234636334356462333639626365330a363862313930303636623230663162643764346239376265353437323766633238363937326664363330383066623238353935626333383565663937313764300a353134636135653635623665333632376636636365653332303238306539643035623238376333613131303663336234636538363932393438303935306161383438666532333431363738343166356363311315633430383366373539656561396261626139613536393764353138306135643765343233383138653061616532346534386162636665616337343536323131346461353665346235343539376131666437373137336165636337623030363433653433376130662636439323533
```

The variables in the vault file can be referenced directly in the plays. Ansible will determine if there are any encrypted files available in the directories where the vars files are expected.

5. 1. 2 Encrypting Individual Properties

To encrypt an individual property run the following command (again using either `--vault-password-file` or the `--ask-vault-pass` parameter to feed the password to the program):

```
ansible-vault encrypt_string --vault-password-file=~/.vaultpwd
<value_to_encrypt>
```

When the string that you're about to encrypt contains dollar sign, at sign, or exclamation mark characters ($, @'or !), you must either wrap the string in single quotes or run the `ansible-vault` command without providing the value to encrypt as part of the command. In the latter case, the `ansible-vault command` will get the value that it should encrypt from the standard input. You can finish input by pressing **Ctrl-D** keys twice.

In either case the `ansible-vault` program prints the encrypted value onto the screen. The value will look like the following:

```
!vault |
          $ANSIBLE_VAULT;1.1;AES256
          35653138653431393766363733373932613365643639623930666439313234326237326263613838
          62323232393662623061373136363864363434616666633630a65386131303561626233333383832
          30666136666661303336362646535363661303030323538326133333032373736353131643939366 3
          30316435326336643 00a6463626464346337633665623662393761356233363037386434623866 65
          3738
```

To place the value to the inventory or the variables file you must keep indentation of the encrypted lines the same. So, if the above encrypted value has to be set as a value of the `jdbc_password` property it should be in the YAML inventory file as follows:

```
jdbc_password: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          35653138653431393766363733373932613365643639623930666439313234326237326263613838
          62323232393662623061373136363864363434616666633630a65386131303561626233333383832
          30666136666661303336362646535363661303030323538326133333032373736353131643939366 3
          30316435326336643 00a6463626464346337633665623662393761356233363037386434623866 65
          3738
```

Important

There is a single space between a colon character that separates the value from the property name as well as a single space between the `!vault` string and the pipe (|) character that is used to define a multiline property in YAML. Then every subsequent line in the multiline property value must have the same indentation shifted few characters to the right of the property name indentation.

For more information, see https://docs.ansible.com/ansible/latest/user_guide/vault.html.

The variable files must be stored in the correct location as required by Ansible. For more information see:

https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html#variable-precedence-where-should-i-put-a-variable

5. 2 Prepare the Inventory File

Before deploying the CMP components, you must prepare an inventory file that lists the CMP hosts and describes the target state of the deployment. The CMP installation repository includes a template for an inventory file that you can copy and then edit by supplying the hostnames of the CMP hosts and values for variables that set the properties for the hosts. See CMP Inventory Template for more information.

In order to install the highly available CMP stack (either fully or partially), more than one host must be specified for a component that requires high availability. Make sure that hosts listed in the JBoss and JWS groups in the inventory match the hosts of all the components that are deployed to these application servers.

5. 2. 1 About the Inventory File

An inventory file is a text file in the YAML format that has the structure as shown in the following example excerpt:

```
---
all:
  vars:
    company_name: MODEL
    branding_code: MODEL
  children:
    rolext:
```

```
hosts:
  server.example.demo.com:
vars:
  # Role extender server port
  server_port: 8081

  # Should use HTTPS listener
  use_ssl: false

  # Name of the WSO2 is user that can validate token
  introspect_username: introspect

  # Above user's password
  introspect_password: password
```

Important

Each section and each parameter in the inventory file MUST have the correct indentation. Use only spaces to create indentations - tabs are not allowed.

Rather than preparing the inventory file manually, you can use the CMP Installation Configuration tool. See "Installation Configuration Tool" on page 35.

There are two global sections under the root of the inventory file (`all`):

- `vars`, which defines the global variables that can be accessed by all hosts used for deployment.
- `children`, which are groups of hosts grouped by function or *role*, such as a database server or an AgentView client. Variables defined under a group are only accessible by the hosts in the group.

  Variables defined in the inventory file are visible within the scope where they are defined. Global variables, defined under `all`, can be accessed by any host in any

group. Group variables are defined within a group and their scope is the hosts within that group, for example the variables for `jboss` in the example above.

The CMP inventory file contains a number of host groups as described in the CMP Groups table. Details of all the properties that can be defined for each can be viewed by clicking on the group name. All the global variables that can be defined are described in "CMP Global Properties" on page 44.

CMP Groups	
Group	**Description**
db	Database server. Currently, the deployment playbook supports only single database server deployments
wso2is	Identity server. CMP uses WSO2 Identity Server for this purpose
rolext	Role extender application
jboss	JBoss server instances. Required for aview_iface, pil, soap and configuration_centre groups.
jws	JBoss Web Server instances. Required for agent_view group
aview_iface	Agent View Interfaces Layer. If the host is listed in this group it must also l be listed in the jboss group
pil	Published Interfaces Layer application
soap	SOAP Web Services application
restws	REST Web Services application
agent_view	Agent View application instances
sabre_server	SABRE Server application instances
sabre_console	Administration Console application instances
configuration_centre	CMP Business Configuration
pentaho	The Report Server is an instance of a Pentaho server.
bulk_action_console	The Bulk Action console application instances

Important

Two groups in the inventory file - jboss and jws - define the underlying platform rather than CMP itself. They are set up separately so that the applications that are hosted on the platform can be deployed to the different physical servers.

Any hosts, used in the following application groups must be listed in the jboss group as well:

- aview_iface
- pil
- soap
- configuration_centre

Also, any hosts used in the following application groups must be also listed in the jws group:

- agent_view

The detailed information about all permissible properties in the inventory file is for use only for an advanced installation. These properties should be changed only by experienced users who understand the consequences or following direction by MDS Global support. A CMP installation usually requires only the placeholders in the inventory file to be replaced as described in "CMP Inventory Template " below and no other values to be changed.

5. 2. 2 CMP Inventory Template

CMP provides an inventory template file, `template.yaml`, which you can copy and edit to suit your CMP deployment environment. It can be adapted to both single host and multiple host environments. See "Appendix B: Inventory Example" on page 95 for examples.

The template inventory file includes placeholders for the property values, placed between angle brackets **< >** as follows:

```
pil:
  hosts:
    <jboss_hostname>:
soapws:
  hosts:
    <jboss_hostname>:
db:
  hosts:
    <database_server_hostname>:
  vars:
    # Database owner credentials, default user 'deploy'. Password should be
    encrypted with ansible-vault
    db_owner: <database_owner_username>
    db_owner_password: <database_owner_password>
```

Section of the CMP inventory template file showing placeholders

To prepare the inventory template:

1. Locate `template.yaml` in the `examples` directory in the unzipped playbook directory that was downloaded as part of the pre-installation steps. Copy the file and open the copy in your editor of choice.
2. Complete the inventory file by providing values for the placeholders. See the table "Template Placeholders" on the next page for information.
3. If you want to encrypt sensitive information such as keys and passwords, follow the instructions for encrypting single properties in "Encrypt Sensitive Inputs in the Inventory File" on page 26.
4. Properties in the template that do NOT have placeholder values have default values provided, for example default port numbers. Some properties may be commented out. You can amend these properties. For more information, see the properties tables below.

   Amending properties may have unintended consequences. Only persons experienced in CMP and your particular deployment should do so. If in doubt, please contact MDS Global.

5. If CMP is deployed to a single host, the following parameters should be set to the same value, which is the name of the target host:
   - database_server_hostname
   - jboss_hostname
   - wso2is_hostname
   - role_extender_hostname
   - jws_hostname
   - sabre_console_hostname
   - sabre_server_hostname
   - restws_hostname
   - pentaho_hostname
   - bulk_action_console_hostname

Important

When developing the inventory file, be aware that the same port number must not be used for more than one service on a single target host. If the same port is used, deployment may not detect it correctly. As a result, deployment will complete successfully but the application will not be installed correctly.

If two or more components that can use the same default ports are to be installed onto the same target host, the port of one these components must be modified to avoid potential port conflict.

The template placeholders that must be replaced for an installation are listed and described in the following table:

Template Placeholders	
Name	**Description**
database_server_hostname	The name of the host where the database is going to be installed
database_schema_name	The database schema name
ga_repository_username	Username that has access to the repository vault.m-dsglobal.dev as supplied by MDS Global
ga_repository_password	The vault.mdsglobal.dev repository username password as supplied by MDS Global
proxy_server_url	Proxy Server URL. If no proxy is required, comment out this property in the inventory file by using a hashtag (#). Do not leave the property blank.
no_proxy_list	Comma separated list of the names and/or IP addresses, access to which should NOT be going through the proxy server. If no proxy is required, comment out this property in the inventory file by using a hashtag (#). Do not leave the property blank. You must also comment out this property if a proxy *is* required but there are not any sites that can be accessed without the proxy.
jboss_hostname	The name of the host where Jboss is going to be installed
database_owner_username	The database owner username
database_owner_password	The password of the database owner user
wso2is_hostname	The fully qualified name of the host where WSO2IS is going to be installed
wso2_jdbc_app_password	The password of the owner of the WSO2 database (used internally within CMP only)
wso2is_admin_admin	WSO2 IS admin user
wso2is_admin_password	WSO2 IS admin user password
role_extender_hostname	The fully qualified name of the host where Role Extender is going to be installed
role_extender_password	Role Extender database user password
jws_hostname	The fully qualified name of the host where JBoss Web Server is going to be installed
jboss_password	JBoss database user password
sabre_console_hostname	The fully qualified name of the host where the Administration Console is going to be installed
sabre_console_password	Administration Console database user password
sabre_server_hostname	The fully qualified name of the host where SABRE Server is going to be installed
sabre_server_password	SABRE Server database user password
restws_hostname	The fully qualified name of the host where REST Web Services is going to be installed
restws_password	REST Web Services database user password
bulk_action_console_hostname	The fully qualified name of the host where Bulk Action UI is to be installed.

Template Placeholders	
Name	**Description**
bulk_action_console_jdbc_password	Database password Bulk Action console
bulk_action_console_remote_host	Name or IP address where bulk action console will be transferring files to/from over the SFTP protocol
bulk_action_console_remoteusername	SFTP username to access the remote host, specified above
bulk_action_console_remote_password	SFTP password to access the remote host, specified above
bulk_action_console_valid_email_domains	A comma separated list of the email domains where Bulk Action Console is allowed to send notifications
pentaho_hostname	The fully qualified name of the host the Report Server is to be installed.
pentaho_server_admin_password	The password for the Pentaho admin user
pentaho_server_jbdc_password	The password for the users that Pentaho server uses in the PostgreSQL database
pentaho_jdbc_app_password	he password for the users that Pentaho server uses in the PostgreSQL database, for the hardcoded database user.

Note 1: As described in "SSL Certificates" on page 18, self-signed SSL certificates are generated unless provided. If trusted 'real' certificates are to be used, the `ssl_certificate` property specified in *SSL Certificates* must be completed.

Note 2: All the database usernames defined above are created in the database by the installation and the components against which they are specified are then configured by the installation to use the username in question. The passwords defined against these usernames are therefore internal to the CMP application and do not need to be known by users of the application.

Note 3: When setting properties for the `db` (database) group, consider what is best to set for the `db_data_population` property for your deployment. For more information, see a description of the property options in "Database Properties" on page 55.

5. 2. 3 Installation Configuration Tool

If you are not experienced in writing YAML, you may find it easier, quicker and less error-prone to use the Installation Configuration tool provided by CMP. The Installation Configuration tool is a web-based tool that enables you to provide values for the properties in the inventory template file and produces:

- An inventory file - `<filename>.yaml`.
- A environment state file - `<yaml filename>.state.json`.

The environment state file is a JSON file that is a snapshot of the inventory file that is being produced. It is created when you create an inventory file and then uploaded to the

tool when you upload that inventory file. The environment state file is used for comparison purposes to detect whether:

- The inventory file has been amended outside the Installation Configuration tool.
- The environment model embedded in the tool has changed, for example the default value of a parameter has changed.

In either case, the affected value will be highlighted when you open your inventory file.

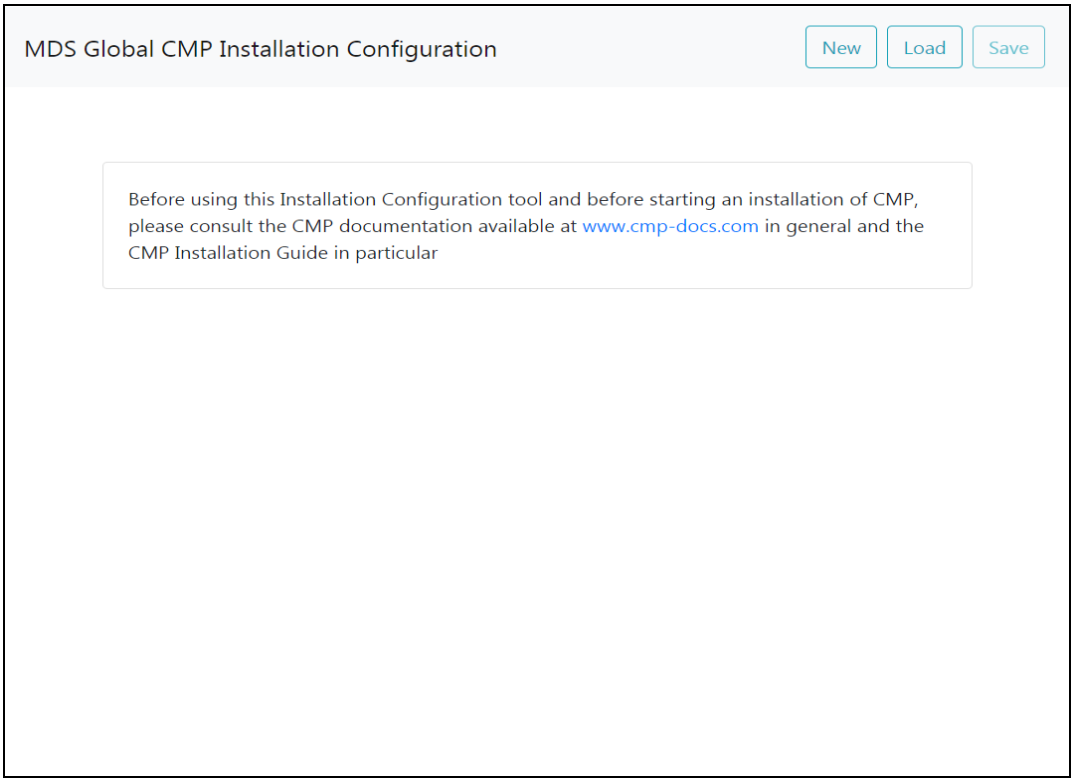
5.2.3.1 Preparing an Inventory File with the Installation Configuration tool

To prepare an inventory file using the Installation Configuration tool, follow these steps:

1. Access the tool via the URL: https://vault.mdsglobal.dev.

   The tool is password protected. Use the credentials provided for the software installation.

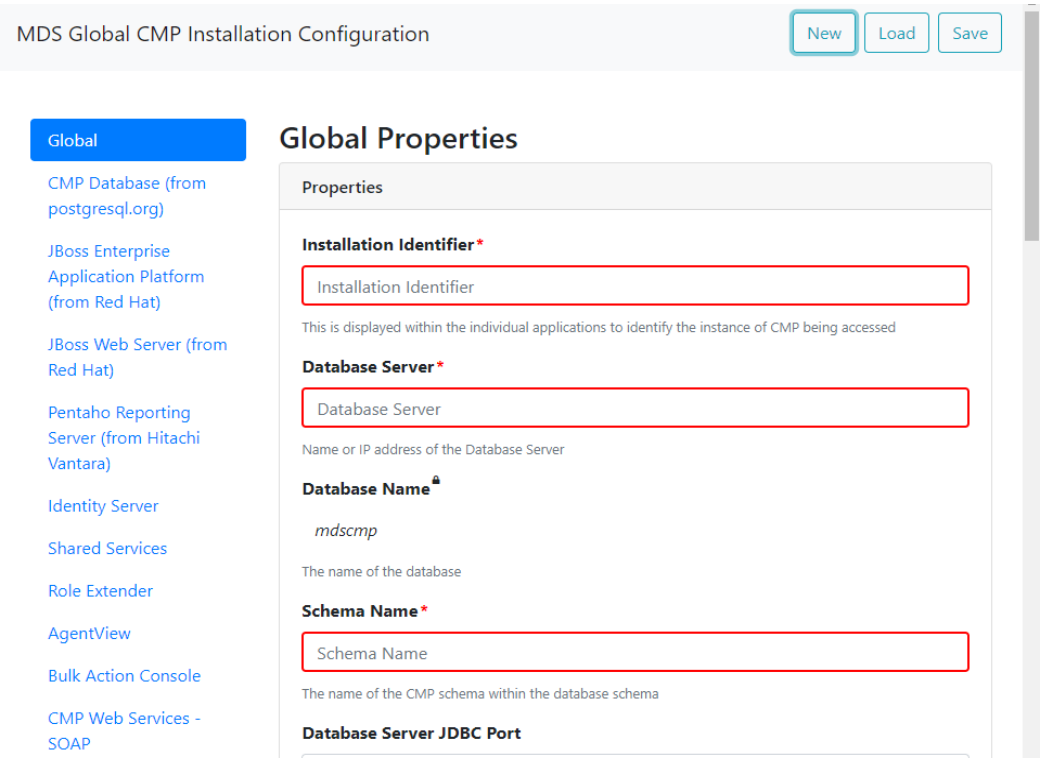
   The tool opens in the browser.



2. To create a new inventory file, click **New**.

   The tool opens on the **Global Properties** screen. On the right is a navigation pane with links to configure each of the components of CMP, for example **CMP Database**, **JBoss Enterprise** or **AgentView**. The list of components reflects the table of "CMP Groups" on page 31.

On the left are the properties, for which you can enter values

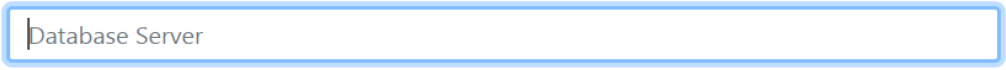


For a description of the Global properties and the properties for each component, see the relevant topic in this section. The explanatory onscreen text will also help you. Default property values are pre-filled.

The colours of the fields are also a guide:

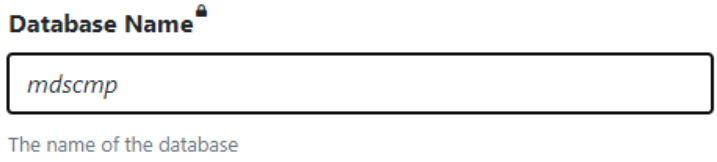
- Mandatory property fields are highlighted in red:

- A property field that has the focus is highlighted in blue:



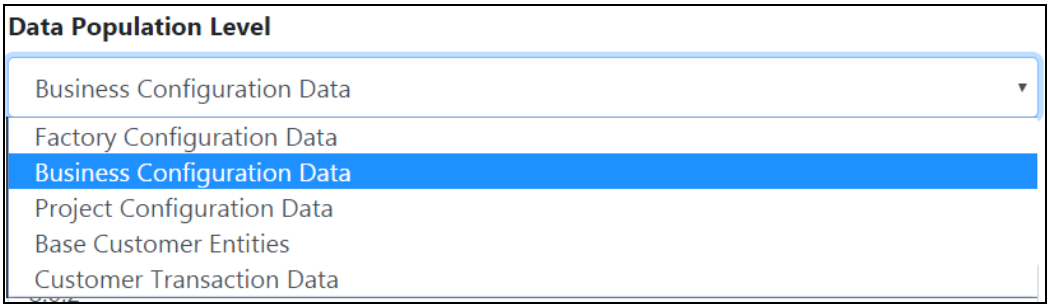
- A property field for which you have entered a value is highlighted in green:

A hardcoded value is highlighted in black:

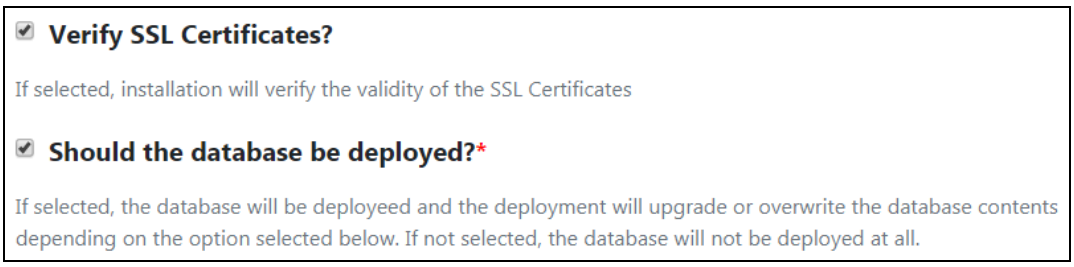


3. Enter a value for a property as follows:
   - For a text value, such as a hostname, database name or URL, enter the text in the field.

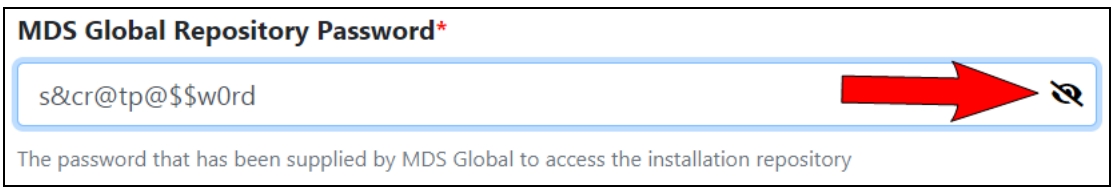
   - To choose from a set of predefined (enum) values, select from a drop-down list.



   - To set a property to true/false or enable/disable, select or deselect a check-box.

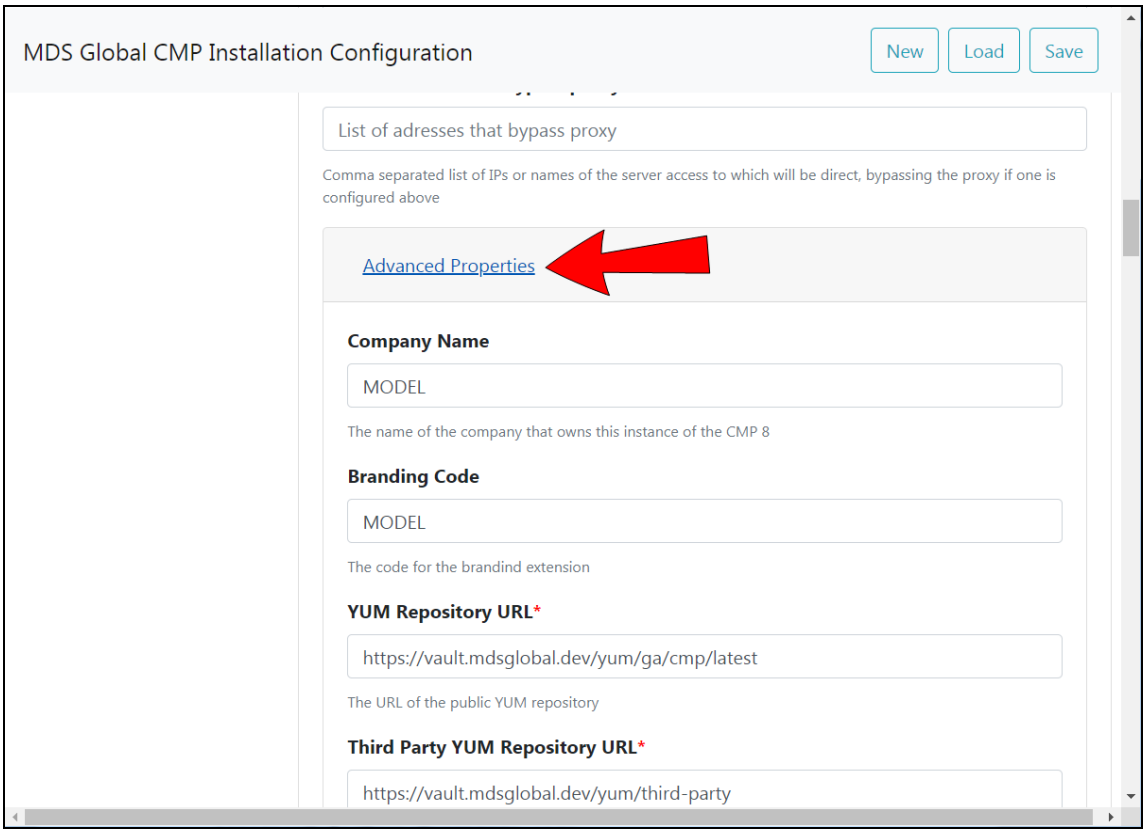


   - When entering a masked property, such as a password, you can toggle the mask/unmask icon to reveal and check your entry.



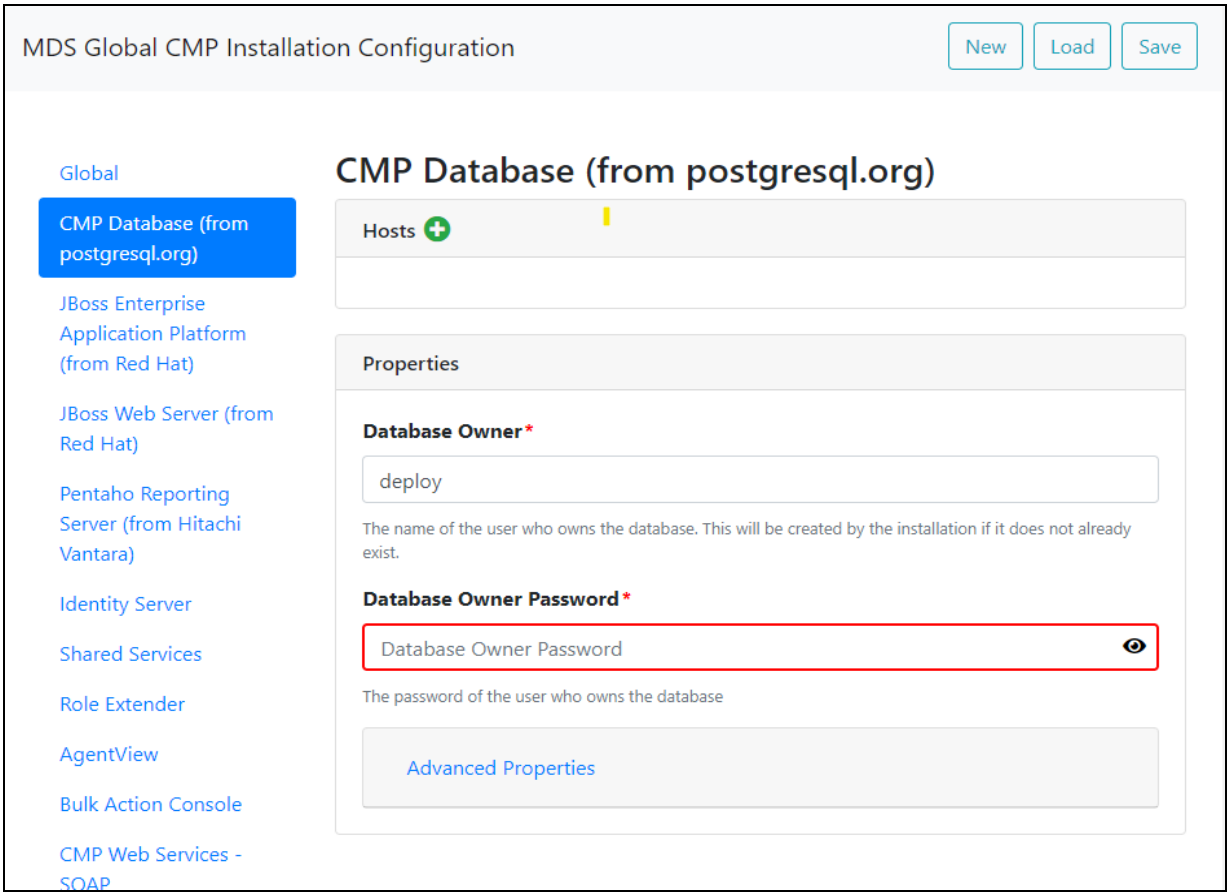
4. Continue to update **Global Properties**.

5. Click the **Advanced Properties** link at the bottom of the **Global Properties** screen to expand the **Advanced Properties** list if you want to update the advanced properties.



For more information on advanced properties, see "Advanced Property Options" on page 68

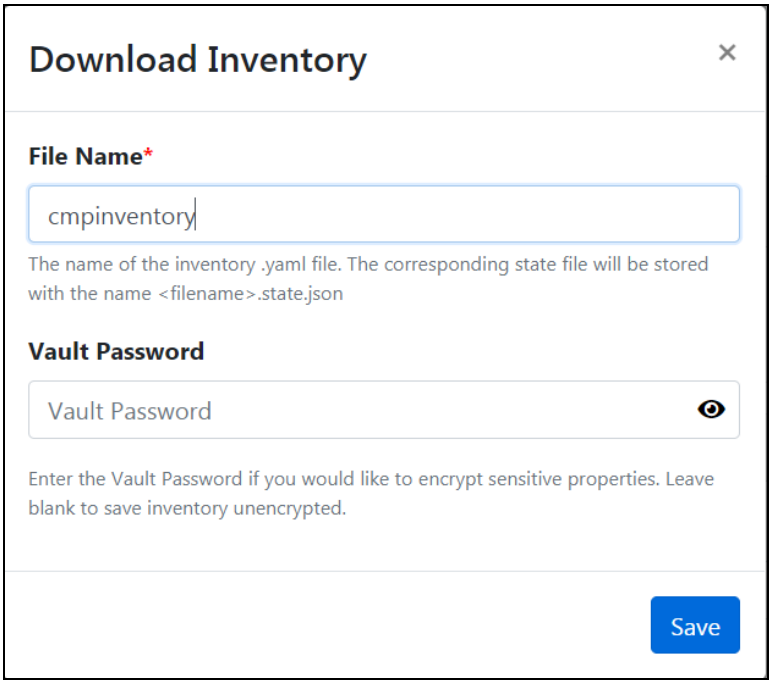
6. Once you have completed the **Global Properties** screen, click a component link in the pane of the left, for example **CMP Database**, to add hosts for the component and to provide property values that will apply to the hosts.



7. To specify the hosts on which this CMP component will be installed, click the **Add** icon.

   You must specify at least one host per CMP component.
8. Enter the host name.
9. To remove a host, click the corresponding **Delete** icon.
10. Enter the property values for the component.
11. Repeat the steps to enter values for all CMP components until you everything is configured.
12. Click **Save**.

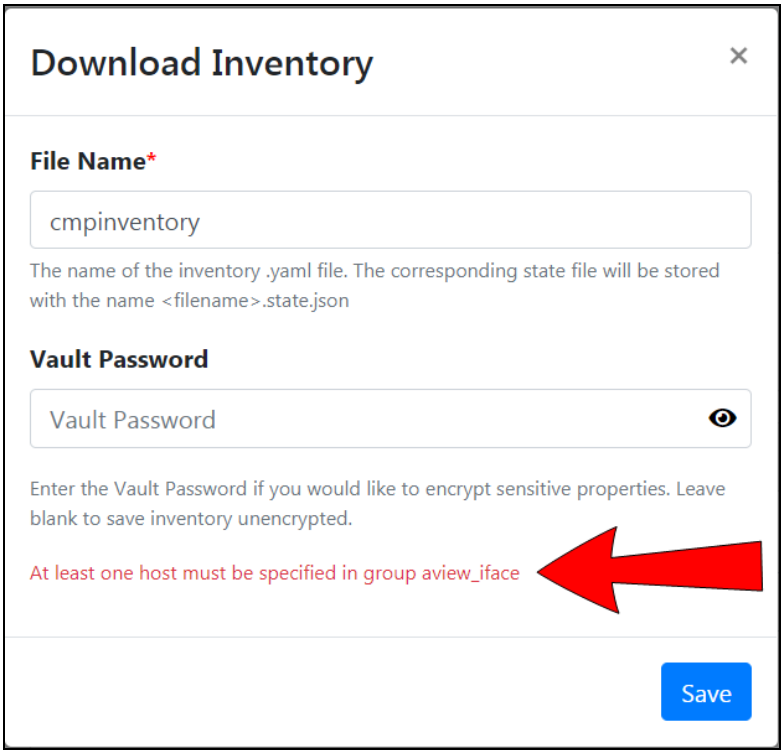
    The **Download Inventory** window is displayed.
13. Enter a name for the inventory file.



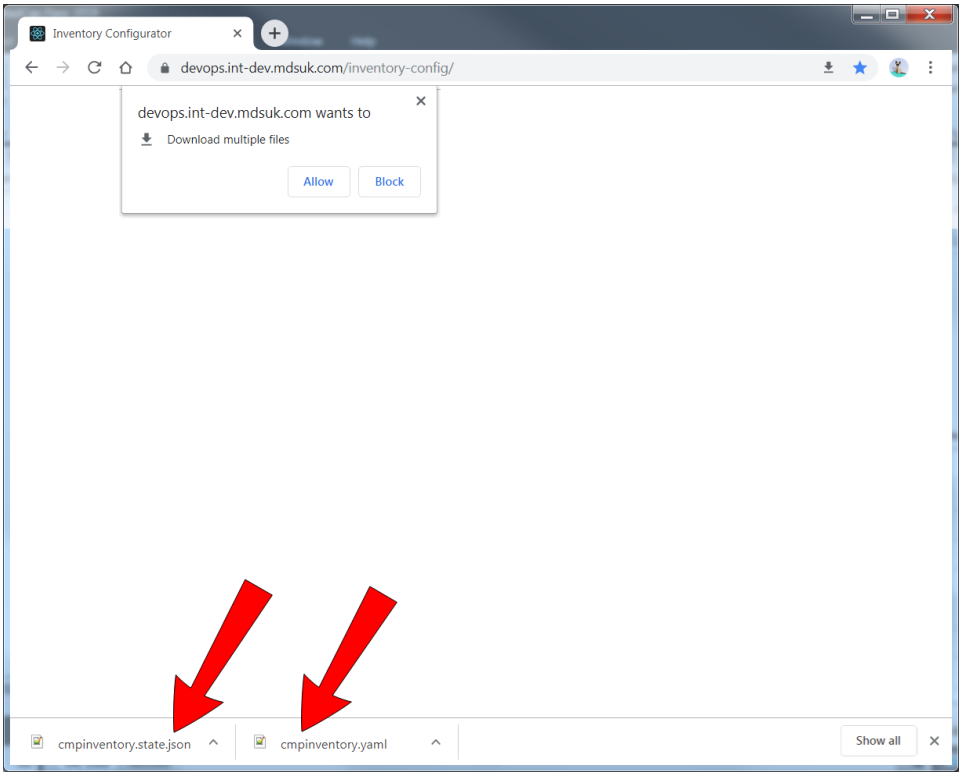
The inventory file will be saved as `<filename>.yaml`. The corresponding environment state file will be saved as `<filename>.state.json`.

14. To encrypt properties in the file, enter the vault file password.
15. Click **Save**.

    The file is validated. An error message will alert you to any issues.



16. Fix any issues and save the file.
17. The files are saved to your Downloads folder.



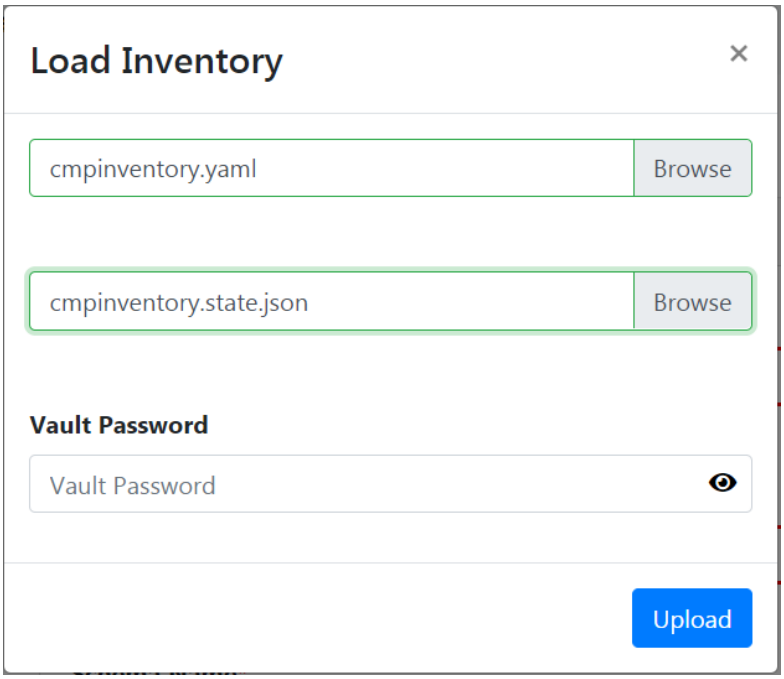
18. The final step is to copy the inventory file to the Ansible control server and use it as part of the installation process, in the same manner as the manually generated file following the instructions in "Deploy CMP" on page 77.

5.2.3.2 Working with an Existing Inventory File

To upload and edit an inventory file that was created previously with the tool, follow these steps:

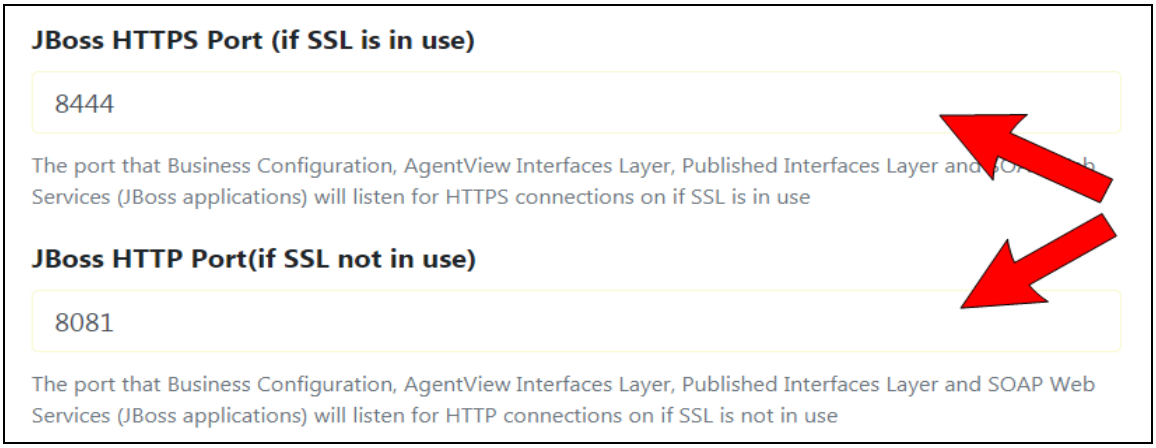
1. Open the Installation Configuration tool and click **Load**.

   The **Load Inventory** window is displayed.
2. To **Select an Inventory File**, click the corresponding **Browse** button.
3. Navigate to the inventory file and select it.
4. Select the corresponding environment state file.



5. If the file is encrypted, enter the **Vault Password**.
6. Click **Upload**.

   The file is uploaded and the tool uses the environment state file to detect whether the inventory file has been changed outside of the tool or whether it differs from the model embedded in the tool. Property values that are different are highlighted in yellow:



7. Edit the inventory by entering property values as described in "Preparing an Inventory File with the Installation Configuration tool" on page 36
8. Save your changes.

5. 2. 4 CMP Global Properties

The following table lists the global properties (variables) that can be defined for CMP when manually configuring the inventory file.

When using the Installation Configuration tool described in "Installation Configuration Tool" on page 35, some properties are either hidden or their range of permitted values is restricted to avoid misconfiguration of the system.

Some values are hardcoded to facilitate configuration:

The name of the database used by the CMP components = `MDSCMP`

The database type = `postgres`

CMP Global Properties			
Property	**Default Value**	**Description**	**Mandatory?**
company_name	MODEL	Company name	Optional
branding_code	MODEL	Branding code	Optional
installation_id	N/A	An installation identifier displayed within individual applications to identify the instance of CMP being accessed	Mandatory
jdbc_server	N/A	Database server name or IP address	Mandatory
jdbc_server_port	5432	Database server port	Optional
jdbc_schema_name	N/A	The name of the database schema used by the CMP components	Mandatory
delivery_state	alpha	YUM repository delivery state suffix. This property is used for internal CMP deployments only.	Optional
yum_repository_url	N/A	The URL of the CMP YUM repository	Mandatory
yum_repository_username	N/A	The CMP repository username.	Optional
yum_repository_password	N/A	The CMP repository password.	Optional
third_party_yum_repository_url	N/A	The URL of the YUM repository with the third party components.	Mandatory
maven_repository_url	N/A	Maven repository URL	Mandatory
should_deploy_db	true	Flag indicating if the database should be deployed using Liquibase or just current properties applied to it	Optional
verify_ssl_certificates	true	If set to false, YUM won't check the validity of the SSL certificate	Optional
ssl_certificates	N/A	This property is a dictionary of three elements: "private_key", "certificate" and "ca". Each element	Optional

CMP Global Properties			
Property	**Default Value**	**Description**	**Mandatory?**
		is the path to the corresponding PEM file. If specified at the global level and `<service>_should_use_ssl` is `true`, the deployment playbook will use the specified files to enable the HTTPS listener. If the property is omitted while `<service>_should_use_ssl` is still set to `true`, the self-signed certificate will be generated for the hostname where the specific service is deployed. The property can be overridden at the group/host level.	
ansible_connection	ssh	Type of connection that Ansible will use to access the remote system	Optional
ansible_ssh_user	ansible	Ansible SSH username	Optional
http_proxy		The address of the HTTP proxy	Optional
https_proxy		The address of the HTTPS proxy	Optional
no_proxy		The comma separated list of hosts for which a proxy should not be used	Optional
use_redhat_subscription	true	If set to `true`, the playbooks will try to enable JBoss and JWS repositories using Red Hat Subscription Manager. In this case, the playbooks expect that all target hosts are registered with Red Hat and have the correct subscriptions enabled. If set to `false`, the target hosts must be manually configured by the customer to access the customer managed YUM repository(-ies), which provide the EAP7 and JWS5 rpm packages.	Optional
SMTP Mail Server Settings		**This email server will be used by various modules to send emails such as password resets and notifications from Sabre**	
smtp_host		Email server SMTP hostname. This email server will be used by various modules to send emails such as password resets and notifications from Administration Console.	Optional
smtp_port	25	The port number of the email server	Optional
smtp_username		The username for authentication	Optional
smtp_password		The password for authentication	Optional
smtp_auth	false	Do your server settings support SMTP authentication?	Optional
smtp_starttls	false	Do your server setting support the STARTTLS command?	Optional

CMP Global Properties			
Property	**Default Value**	**Description**	**Mandatory?**
smtp_from		The "from" email address	Optional
Common Application Properties		**These are properties that are not specific to a single application.**	
sam_name		Defines the Synchronous Allowance Management implementation that will be deployed for this CMP installation. The options are: `none`, `offline`, `lifecycle`, or `openet` (None, Offline SAM, Lifecycle SAM or Openet SAM in the Installation Configuration Tool). Depending on what is selected, the appropriate modules for that OCS will be installed along with SABRE Server, REST, AgentView Interfaces Layer and Business Configuration (Configuration Centre).	Mandatory
internal_web-service_password		The password of the user used when calling SOAP and REST web services from other applications within the CMP product suite (only used internally within CMP)	Mandatory
log_collector_sftp_password		The password of the "logcollector" user that the CMP Administration Console will use to retrieve log files for display.	Mandatory
Health Check Service Properties		**These are properties which relate to the health check service that is deployed alongside each web application for use by an external load balancer if required.**	
use_ssl	true	Should SSL be used for the Health Check Service?	Optional
jdbc_password	N/A	The password for the user that the Health Check Service will use to connect to the database (only used internally within CMP)	Mandatory
http_port	21215	The port that the Health Check Service will listen for HTTP(S) connections on. This will be common across all hosts to which the health check service is deployed.	Optional

5. 2. 5 Administration Console Properties

The following properties could be modified when deploying the Administration Console:

The JDBC username used by the Administration Console to connect to the database is hardcoded to facilitate configuration.

Property	Default Value	Description
sabre_console_port	31212	The HTTP(S) port to which the Administration Console is bound
sabre_console_jdbc_password	N/A	The JDBC password used by the Administration Console to connect to the database
sabre_console_use_ssl	false	Flag, indicating if the HTTPS protocol should be used by the Administration Console
ssl_certificates	N/A	This property is a dictionary of three elements: "private_key", "certificate" and "ca". Each element is the path to the corresponding PEM file. If specified and `sabre_console_use_ssl` is `true,` the deployment play will use the specified files to enable the HTTPS listener. If the property is omitted while `sabre_console_use_ssl` is `true,` , the self-signed certificate will be generated for the hostname to which the Administration Console is being deployed.
sabre_console_lb_url	N/A	The URL of the Load Balancer in format `<protocol>://<lb_hostname>(:<lb_port>)` that serves the Administration Console application. It must be defined for the HA installation; i.e. when there is more than one Administration Console server is defined. The `port` is optional and only must be specified when the non-standard port is used.
sabre_console_allocated_heap_size	2 GB	The size of the Administration Console heap memory. Please ensure there is enough memory on the host to accommodate allocated heap sizes.

5. 2. 6 AgentView Interfaces Layer Properties

The following properties can be modified when deploying AgentView:

Property	Default Value	Description
settings_ address_lookup_ url	N/A	The URL of the address lookup provider
settings_ address_lookup_ provider	QA67	Address lookup provider name
additional_set- tings	N/A	Additional settings that should be injected to the AgentView Database. It is the array of the dictionaries, for details refer to the Additional Settings table.

5.2.6.1 Additional Settings

These additional AgentView Interfaces Layer properties must only be used if instructed by MDS Global support. The list of properties will be provided when necessary. In most deployment cases the properties must be removed or commented out.

Property	Description
group	The property group
property	The property name
value	The property value

For example:

```
additional_settings:
  - group: CURRENCY
    property: ISOCURRENCYCODE
    value: GBP
  - group: QUERYLIMITS
    property: CNTRCTSUB
    value: 5
```

5. 2. 7 AgentView Properties

Property	Default Value	Description
agent_view_b_url	N/A	The URL of the LoadBalancer in format `<protocol>://<lb_hostname>(:<lb_port>)` that serves AgentView (WebSwing) application. It must be defined for the HA installation, i.e. when more than one AgentView server is defined. The `port` is optional and must be specified only when the non-standard port is used.
redirect_payment_rest_password	NA	The password for the user that is used to update the payment records based on the response from the payment provider (only used internally within CMP). This password must adhere to the defined WSO2 Identity Server password policy. The password will not be validated against the policy when saving the inventory, but the installation will fail if the password policy is violated. The default policy states the password should contain a digit[0-9], a lower case letter[a-z], an upper case letter[A-Z] and one of '!@#$%&' , but this may be changed post installation.
Advanced Properties		
agent_view_font	/usr/share/fonts/msttcorefonts/arial.ttf	The font to display in the AgentView application. Customize the physical font used by the application by specifying the absolute file path to the file (TTF only) containing the font.

5. 2. 8 Bulk Action Console Properties

The following properties can be modified when deploying the Bulk Action UI:

The username used to connect to the database is hardcoded to facilitate configuration.

Property	Default Value	Description	Mandatory?
hosts	N/A	The hostname of the Bulk Action console	Mandatory
bulk_action_console_server_port	9009	Bulk Action Console Application port	Mandatory
bulk_action_console_use_ssl:	true	Should Bulk Action Console use HTTPS Listenet?	Mandatory
bulk_action_console_jdbc_password	N/A	Database password	Mandatory
file_transfer:	N/A	Parameters of file transfer between Bulk Action Console and Bulk Action Job	Optional
type	SFTP	Protocol to use for transfer	Mandatory
host	localhost	Bulk Action Job Server Host	Mandatory
port	22	Standard SFTP port	Mandatory
username	N/A	Username to access server remotely	Mandatory
password	N/A	Password to access server remotely	Mandatory
directory	bulkaction/pending_transform/	Directory where the Bulk Action Console drop files into	Mandatory
reports_directory	bulkaction/reports/	Directory where the Bulk Action Job stores reports	Mandatory
valid_email_domains:	N/A	Comma separated list of the email domains where Bulk Action Console can send communications	Mandatory
bulk_action_console_keystore_password	changeme	The optional keystore password, default value is changeme	Optional
bulk_action_console_lb_url	N/A	The URL of the LoadBalancer in format `<protocol>://<lb_hostname>(:<lb_port>)` that serves Bulk Action Console application. It must be defined for the HA installation; i.e. when there is more than one Bulk Action Console	Required for HA deployment

Property	Default Value	Description	Mandatory?
		server is defined. The `port` is optional and only must be specified when the non-standard port is used.	
bulk_action_allocated_heap_size	2 GB	The size of the Bulk Action Console heap memory. Please ensure there is enough memory on the host to accommodate allocated heap sizes.	

5. 2. 9 Configuration Centre (Business Configuration) Properties

Property	Default Value	Description
configuration_centre_lb_url	N/A	The URL of the Load Balancer in format `<protocol>://<lb_hostname>(:<lb_port>)` that serves the Business Configuration Login application. It must be defined for the HA installation, i.e. when more than one Business Configuration server is defined. The `port` is optional and must be specified only when the non-standard port is used.

5. 2. 10 Database Properties

Property	Default Value	Description
db_owner	N/A	The database owner username
db_ owner_ password	N/A	The password of the database owner user
should_ upgrade_ db	true	Flag indicating whether the database upgrade should be performed. From version 8.0.3, the CMP deployment playbook supports the option to perform either: • A clean database deployment when the database will be recreated and as a result of the deployment, existing data will be deleted. • A database upgrade when the database structure and dictionary data will be upgraded to the new version, retaining existing data. **Important**: Do not change this property from the default value of `true` unless you fully understand the consequences of this action. Under NO circumstances should this be done in a production environment.
upgrade_ db_from_ version	N/A	The current version of the CMP database (e.g. 8.0.2) in case of the database upgrade. The version stored in the upgrade_db_from_ version property must be less than the version of CMP that is being deployed. In addition, the version must be one that is supported for upgrade from for the release as indicated in the release notes.
db_data_ population	Business_Configuration	The string identifying what data should be populated to the database at the time of the initial deployment. This property accepts the following values: • Factory_Configuration Data configuration that is required for the basic operation of CMP and is maintained as part of future software upgrades. • Business_Configuration Data configuration that is very infrequently changed in a CMP installation. Example data is provided that will likely form a good foundation for the configuration of CMP. This data will not be overwritten or updated as part of future software upgrades.

Property	Default Value	Description
		• Project_Configuration Data configuration that will always be specific to each CMP installation and will frequently change. Example data is provided that could be used as a foundation for the configuration of CMP but is more likely of use for demonstration and test purposes only. This data will not be overwritten or updated as part of future software upgrades. The data population options are hierarchical and build on each other. For example, if you select Factory_Configuration, Business and Project configuration are automatically and mandatorily populated too. This property can be omitted, in which case, the default value is used.

Two dictionary properties control database behaviour:

Group	Description
pg	PostgreSQL related properties
os	Operating system related properties

When CMP is deployed for the first time, the PostgreSQL and OS property settings for the database are automatically derived from the CPU and memory information supplied in the following database properties:

Property	Default Value	Description
db_system_cpu	4CPU	Number of CPUs on the system which hosts the database. This is used to determine how the Postgres database is configured.
db_system_mem	8GB Memory	The amount of memory on the system which hosts the database. This is used to determine how the Postgres database is configured.
db_system_dsk	SlowDisk	The type of disk installed on the system which hosts the database. This is used to determine how the Postgres database is configured.

When deploying CMP for the first time, if you need to alter these properties it is recommend that you configure them in the Installation Configuration tool in the page for the **CMP Database** host group under **Advanced Properties**.

Once CMP is installed, however, the system must be monitored and these settings adjusted as needed as the system matures, either manually or using automation tools.

5. 2. 11 JBoss Server Properties

The following properties can be modified when deploying a JBoss server:

The JDBC username for the default connection pool is hardcoded to facilitate configuration.

Property	Default Value	Description
jboss_jdbc_password	N/A	The password for the default connection pool
jboss_http_port	8080	The HTTP listener port
jboss_https_port	8443	The HTTPS listener port
jboss_management_http_port	9990	The management server HTTP listener port
jboss_management_https_port	9993	The management server HTTPS listener port
customer_name	MODEL	Customer Name
server_type	JBOSS	Server type: JBOSS or WEBSPHERE
primary_server	localhost	Primary server; defines an instance group
jboss_remoting_port	20000	JBoss remoting port, the port that is used to communicate with the CMP services layer
primary_server_port	20000	Primary server port. Defines an instance group.
jboss_node_id	node-	JBoss cluster node could be overridden at the host level as it must be unique in the clustered environment
jboss_dynamic_list_threshold	20	JBoss dynamic list threshold
jboss_heartbeat_interval	420000	JBoss heartbeat interval
jboss_large_tree_threshold	50000	JBoss large tree threshold
is_cmp_lite	true	Is the CMP running in the light mode?
metaspace_size	340M	The Java metaspace size, with the max_metaspace_size property, this property sets the boundaries for metaspace allocation by instructing the JVM to start garbage collection when metaspace allocation reaches the value of metaspace_size.
jboss_allocated_heap_size	4GB	The size of the JBoss EAP heap memory. Please ensure there is enough memory on the host to accomodate allocated heap sizes.
max_metaspace_size	N/A	With the metaspace_size property, this property sets the boundaries for metaspace allocation. When metaspace allocation reaches the value of the max_metaspace_size, an OutOfMemory error will be thrown.
ssl_certificates	N/A	This property is a dictionary of three elements: "private_key", "certificate" and "ca". Each element is the path to the corresponding PEM file. If specified and `jboss_use_ssl` is `true`, the

Property	Default Value	Description
		deployment play will use the specified files to enable the HTTPS listener. If the property is omitted while `jboss_use_ssl` is `true`, the self-signed certificate will be generated for the hostname to which JBoss is being deployed.
jboss_use_ssl	false	Flag indicating if the SSL listeners should be activated

5. 2. 12 JBoss Web Server Properties

The following properties can be modified when deploying JBoss Web Server:

Property	Default Value	Description
jws_http_port	8080	The HTTP(S) port to which the JBoss Web Server is bound
jws_https_port	7443	The HTTPS port to which the JBoss Web Server is bound
jws_use_ssl	true	Flag, indicating if the HTTPS protocol should be used by the JBoss Web Server
ssl_certificates	N/A	This property is a dictionary of three elements: "private_key", "certificate" and "ca". Each element is the path to the corresponding PEM file. If specified and `jws_use_ssl` is `true`, the deployment play will use the specified files to enable the HTTPS listener. If the property is omitted while `jws_use_ssl` is set to `true`, the self-signed certificate will be generated for the hostname to which JWS being deployed.
wso2is_login_lb_url	N/A	The URL of the LoadBalancer in the format `<protocol>://<lb_hostname>(:<lb_port>)` that serves WSO2 IS Login application. It must be defined for the HA installation; i.e. when there is more than one JWS server is defined. The `port` is optional and must be specified only when the non-standard port is used.
jws_allocated_heap_size	4 GB	The size of the JBoss Web Server heap memory. Please ensure there is enough memory on the host to accommodate allocated heap sizes.

5. 2. 13 REST Web Services Properties

The following properties can be modified when deploying REST Web Services:

The JDBC username that REST Web Services application uses to connect to a database is hardcoded to facilitate configuration.

Property	Default Value	Description
restws_ server_port	9000	The HTTP(S) port to which REST Web Services is bound
restws_jdbc_ password	N/A	The JDBC password that REST Web Services application uses to connect to a database
restws_use_ ssl	true	Flag, indicating if the HTTPS protocol should be used by the REST Web Services
ssl_cer- tificates	N/A	This property is a dictionary of three elements: "private_key", "cer- tificate" and "ca". Each element is the path to the corresponding PEM file. If specified and `restws_use_ssl` is `true`, the deploy- ment play will use the specified files to enable HTTPS listener. If the property is omitted and `restws_use_ssl` is `true`, the self- signed certificate will be generated for the hostname where SABRE Console is being deployed to.
restws_lb_url	N/A	The URL of the LoadBalancer in the format `<protocol>://<lb_ hostname>(:<lb_port>)` that serves REST WS application. It must be defined for the HA installation; i.e. when there is more than one REST WS server is defined. The `port` is optional and must be specified only when the non-standard port is used.
restws_alloc- ated_heap_ size	4 GB	The size of the REST Web Service heap memory. Please ensure there is enough memory on the host to accomodate allocated heap sizes.

5. 2. 14 Pentaho Server Properties

The following properties can be modified when deploying a Pentaho server (the Report Server):

The username for the CMP database used by the Pentaho Server to run reports is hard- coded to facilitate configuration.

Property	Default Value	Description	Mandatory?
hosts	N/A	The hostname of the Pentaho server	Mandatory
pentaho_jdbc_reporting_password	N/A	The password for the CMP database that is used by the Pentaho Server to run reports	Mandatory
pentaho_admin_password	N/A	The Pentaho Server admin user password	Mandatory
pentaho_jdbc_app_password	N/A	The password for the users that Pentaho server uses in the PostgreSQL database, for the hardcoded database user.	Mandatory
old_pentaho_admin_password	password	Optional old admin password. Will be used when the purpose of the Pentaho Server setup playbook execution is to change the admin password.	Optional
pentaho_lb_url	N/A	The URL of the LoadBalancer in format `://<lb_host-name>(:<lb_port>)` that serves the Pentaho application. It must be defined for the HA installation; that is, when more than one SOAP Web Services server is defined. Port is optional and must be specified only when the non-standard port is used.	Mandatory for a High Availability installation

5. 2. 15 Role Extender Properties

The following properties can be modified when deploying Role Extender Web Service:

The JDBC username used by the Role Extender to connect to a database is hardcoded to facilitate configuration.

Property	Default Value	Description
rolext_server_port	8080	The HTTP(S) port to which Role Extender Web Services is bound
rolext_jdbc_password	N/A	The JDBC password used by the Role Extender to connect to a database
rolext_should_use_ssl	true	Flag, indicating if the HTTPS protocol should be used by the Role Extender Web Services
ssl_certificates	N/A	This property is a dictionary of three elements: "private_key", "certificate" and "ca". Each element is the path to the corresponding PEM file. If specified and `rolext_should_use_ssl` is `true`, the deployment play will use the specified files to enable HTTPS listener. If the property is omitted while `rolext_should_use_ssl` is `true`, the self-signed certificate will be generated for the hostname where SABRE Console is being deployed to.
rolext_lb_hostname	N/A	The Fully Qualified Domain Name of the Load Balancer Host that serves Role Extender. Required for the HA installation.
rolext_lb_port	N/A	The port number used by the Load Balancer that serves Role Extender. Should be set only when non-standard port is used

5. 2. 16 SABRE Server Properties

The following properties could be modified when deploying SABRE Server:

The JDBC username used by the SABRE Server to connect to the database is hard-coded to facilitate configuration.

Property	Default Value	Description
sabre_server_jdbc_password	N/A	The JDBC password used by the SABRE Server to connect to the database
hz_cluster_port	27300	A TCP port that is used by Hazelcast for the intracluster communication
hz_log4j_cluster_port	27400	A TCP port that is used by Hazelcast for the log4j intracluster communitation
hz_multicast_enabled	false	Flag stating if the multicast should be enable or disabled for the Hazelcast intracluster communication
hz_multicast_group	224.14.14.14	A multicast group used by Hazelcast if multicast is enabled in the cluster
sabre_management_port	21212	SABRE Server management port (HTTP)
sabre_allocated_heap_size	4 GB	The heap size that will be allocated by JVM to SABRE Server. Mandatory parameter in the inventory file. Please ensure there is enough memory on the host to accommodate allocated heap sizes.
sabre_server_use_ssl	true	The flag indicating if the SABRE Server management endpoint should use HTTPS protocol
sabre_admin_user	admin	An admin user to access the management endpoint
sabre_admin_password	admin	An admin user password that is required to access management endpoint.
additional_modules	N/A	The array of the names of the additional modules that should be installed. Each element in the array should be the name of the additional module RPM
additional_property_files	N/A	The array of the names of the additional properties files that should be deployed to the target system. The files that included into this array must exist in the `sabre-server/properties` directory and should be the plain text properties file without use of the Jinja template placeholders. The files will be deployed "as-is". If you do not want to deploy any additional properties files, this element must not be included into inventory file as an empty array causes the playbook execution failure
additional_property_templates	N/A	The array of the names of the additional properties Jinja templates that should be deployed to the target system. The files that included into this array must exist in the `sabre-server-/property_templates` directory. If you do not want to deploy

Property	Default Value	Description
		any additional properties templates, this element must not be included into inventory file as an empty array causes the play-book execution failure
bulkaction_ password	N/A	The password of the 'BULKACTION' user that the SABRE jobs will use when calling the CMP SOAP and REST web services
ssl_certificates	N/A	This property is a dictionary of three elements: "private_key", "certificate" and "ca". Each element is the path to the cor-responding PEM file. If specified and `sabre_server_should_use_ssl` is `true`, the deployment play will use the specified files to enable the HTTPS listener. If the property is omitted while `sabre_server_should_use_ssl` is `true`, the self-signed cer-tificate will be generated for the hostname to whichSABRE Server is being deployed.

5. 2. 17 Shared Services Properties

The following properties can be modified when deploying shared services

Property	Default Value	Description
shared services_jdbc_password	N/A	The password for the user that the Shared Services uses to connect to the database (only used internally within CMP)
shared services_port	21218	The port on which Shared Services will listen for connections
shared services_use_ssl	false	Flag that indicates whether the SSL listeners should be activated
shared_services_passwordt	N/A	The password that is used by other applications to call Shared Services (only used internally within CMP). If omitted a password will be generated.
sharedservices_lb_url	N/A	The URL of the Load Balancer that serves Shared Services REST application in the format: `://<lb_hostname>(:<lb_port>)` It must be defined for the HA installation, i.e. when there is more than one Shared Services server to be defined. `Port` is optional and must be specified only when the non-standard port is used
ssl_certificates	N/A	This property is a dictionary of elements: "private_key"and "certificate". Each element is the path to the corresponding PEM file. If specified and `sharedservices_use_ssl` is `true`, the deployment playbook will use the specified files to enable the HTTPS listener. If the property is omitted and `sharedservices_use_ssl` is `true`, the self-signed certificate will be generated for the hostname to where SABRE Console is being deployed.
sharedservices_allocated_heap_size	4 GB	The size of the Shared Service heap memory. Please ensure there is enough memory on the host to accommodate allocated heap sizes.

5. 2. 18 SAM Services Properties

The OCS Synchronous Provisioning Interface has been renamed Synchronous Allowance Management (SAM). SAM Services support configuration of the CMP module that supports multiple OCS implementations as well as CMP running in offline mode i.e. with no OCS integration.

Property	Default Value	Description
samservices_jdbc_password	N/A	The password for the user that the SAM Services uses to connect to the database (only used internally within CMP)
samservices_port	21220	The port on which SAM Services will listen for connections
samservices_use_ssl	false	Flag that indicates whether the SSL listeners should be activated
sam_services_password	N/A	The password that is used by other applications to call SAM Services (only used internally within CMP). If omitted a password will be generated.
ssl_certificates	N/A	This property is a dictionary of elements: "private_key"and "certificate". Each element is the path to the corresponding PEM file. If specified and `samservices_use_ssl` is `true`, the deployment playbook will use the specified files to enable the HTTPS listener. If the property is omitted and `samservices_use_ssl` is `true`, the self-signed certificate will be generated for the hostname to where SABRE Console is being deployed.
samservices_lb_url	N/A	The URL of the Load Balancer that serves SAM Services REST application in the format: `://<lb_hostname>(:<lb_port>)` It must be defined for the HA installation, i.e. when there is more than one SAM Services server to be defined.`Port` is optional and must be specified only when the non-standard port is used
Advanced Properties		
samservices_allocated_heap_size	2 GB	The size of the SAM Services heap memory. Minimum 1 GB. Please ensure there is enough memory on the host to accommodate allocated heap sizes.

5. 2. 19 SOAP Web Services Properties

Property	Default Value	Description
soapws_lb_url	N/A	The URL of the LoadBalancer in format `://<lb_hostname>(:<lb_port>)` that serves the SOAP Web Services application. It must be defined for the HA installation; that is, when more than one SOAP Web Services server is defined. Port is optional and must be specified only when the non-standard port is used.

5. 2. 20 WSO2 Identity Server Properties

The following properties could be modified when deploying WSO2 Identity Server

Property	Default Value	Description
well_known_context_secured	false	This flag indicates if the WSO2 IS discovery endpoint (https://<wso2_is_server>:9443/oauth2/oidcdiscovery/.well-known/openid-configuration) should require basic authentication to get access to.
should_use_external_database	true	This flag indicates if the WSO2 IS should use external database for setup. Recommended for production setup. If false, WSO2 will create an H2 database.
should_use_database_user_store	true	This flag indicates if the WSO2 IS should use the database as a primary user store. If false, it will create embedded LDAP server and will use it as a primary user store.
wso2is_admin_user	admin	The initial WSO2 IS admin user
wso2is_admin_password	admin	The initial WSO2 IS admin user password
wso2_jdbc_app_password	N/A	The password of the owner of the WSO2 database (used internally within CMP only)
ssl_certificates	N/A	This property is a dictionary of three elements: "private_key", "certificate" and "ca". Each element is the path to the corresponding PEM file. If the property is omitted, the self-signed certificate will be generated for the hostname to which WSO2r is being deployed.
wso2is_shared_storage_path	`/opt/cmp-storage/wso2is`	Path on the target server with WSO2 IS instance where the shared storage is mounted. It is required only in the HA installation scenario.
wso2is_lb_hostname	N/A	The Fully Qualified Domain Name of the Load Balancer Host that serves WSO2 IS. Required for the HA installation
wso2is_lb_port	N/A	The port number used by the Load Balancer that serves WSO2 IS. Only should be set when non-standard port is used.
wso2is_allocated_heap_size	4 GB	The size of the Identity Server heap memory. Please ensure there is enough memory on the host to accommodate allocated heap sizes.

5. 2. 21 Advanced Property Options

The following are advanced global properties that can be configured:

Adjusting the following parameters may have unintended consequences. Only persons experienced in CMP and your particular deployment should do so. If in doubt, please contact MDS Global.

Property	Default Value	Description
toolbox_account_managers_enabled	true	
toolbox_activation_date_enabled	true	
toolbox_add_agreement_bundle_links_enabled	true	
toolbox_add_apn_enabled	true	
toolbox_add_non_default_bundles_enabled	true	
toolbox_addcreditcard_for_secuiritisation_enabled	true	
toolbox_aged_debt_enabled	true	
toolbox_allow_access_to_tef_pins_enabled	true	
toolbox_authorise_on_capture_enabled	true	
toolbox_avs_csc_responses_enabled	true	
toolbox_bill_viewer_enabled	true	
toolbox_bulk_data_loader_enabled	true	
toolbox_by_pass_validate_card_details_against_third_party_enabled	true	
toolbox_cc_authorisation_details_enabled	true	
toolbox_cc_onceoff_payments_enabled	true	
toolbox_client_polling_enabled	true	
toolbox_clone_sub_and_maintain_service_price_enabled	true	
toolbox_closed_user_group_rating_enabled	true	
toolbox_computer_telephony_integration_enabled	true	
toolbox_configured_tasks_enabled	true	
toolbox_convergent_prepay_postpay_enabled	true	
toolbox_corporate_deletion_enabled	true	
toolbox_correspondence_mailmerge_enabled	true	
toolbox_country_code_enabled	true	
toolbox_create_address_enabled	true	
toolbox_create_invoice_query_amount_enabled	true	
toolbox_credit_limit_review_enabled	true	

Property	Default Value	Description
toolbox_customer_preferences_enabled	true	
toolbox_detailed_credit_card_details_enabled	true	
toolbox_diaryevent_duedate_indication_enabled	true	
toolbox_disable_bundles_enabled	true	
toolbox_disabled_recurring_card_payment_enabled	true	
toolbox_donor_led_porting_enabled	true	
toolbox_enable_add_remove_bundle_enabled	true	
toolbox_enhanced_call_unit_details_enabled	true	
toolbox_fup_and_capping_enabled	true	
toolbox_gprs_tariff_search_enabled	true	
toolbox_group_deletion_enabled	true	
toolbox_hardware_credit_limit_enabled	true	
toolbox_health_check_enabled	true	
toolbox_include_agents_phone_number_in_pci_redirect_payment_enabled	true	
toolbox_internal_external_account_segregation_enabled	true	
toolbox_invoice_external_reference_enabled	true	
toolbox_log_termination_cost_request_enabled	true	
toolbox_managed_card_details_enabled	true	
toolbox_managed_service_sessions_enabled	true	
toolbox_mandatory_category_code_enabled	true	
toolbox_nsa_command_enabled	true	
toolbox_nsa_response_enabled	true	
toolbox_online_payments_enabled	true	
toolbox_opticals_enabled	true	
toolbox_optimised_address_search_enabled	true	
toolbox_override_bank_account_validation_enabled	true	
toolbox_parent_child_subscription_link_enabled	false	Keep it false every-

Property	Default Value	Description
		where.
toolbox_pci_compliant_enabled	true	
toolbox_pci_compliant_one_off_transactions_enabled	true	
toolbox_post_paid_caps_enabled	true	
toolbox_prescribed_alert_enabled	true	
toolbox_query_email_address_enabled	true	
toolbox_query_invoice_query_amounts_enabled	true	
toolbox_quick_address_enabled	true	
toolbox_record_batch_payment_transaction_enabled	true	
toolbox_record_product_item_serial_numbers_enabled	true	
toolbox_redirect_payment_enabled	true	
toolbox_refunds_enabled	true	
toolbox_remove_email_address_enabled	true	
toolbox_sales_order_courier_interface_enabled	true	
toolbox_separate_payment_authorise_and_capture_ enabled	true	
toolbox_single_parent_subscription_for_account_link_ enabled	true	
toolbox_split_billing_enabled	true	
toolbox_subc_key_info_enabled	true	
toolbox_swms_order_interface_enabled	true	
toolbox_tariff_pricing_enabled	true	
toolbox_technology_funds_enabled	true	
toolbox_unbilled_calls_enabled	true	
toolbox_unbilled_unit_summary_enabled	true	
toolbox_vat_law_reverse_charge_enabled	true	
toolbox_warehousing_enabled	true	
toolbox_weblinks_enabled	true	

Property	Default Value	Description
toolbox_wifi_purchase_and_usage_enabled	true	

5. 2. 22 Context Sensitive Help Properties

AgentView and other consoles, such as Business Configuration and the Administration Console, have context-sensitive HTML5 Help that is installed as a CMP module.

Property	Default Value	Description
context_sensitive_help_port	21221	The port on which the Help will listen for connections
context_sensitive_help_use_ssl	false	Flag that indicates whether the SSL listeners should be activated
ssl_certificates	N/A	This property is a dictionary of elements: "private_key"and "certificate". Each element is the path to the corresponding PEM file. If specified and `context_sensitive_help_use_ssl` is `true`, the deployment playbook will use the specified files to enable the HTTPS listener. If the property is omitted and `context_sensitive_help_use_ssl` is `true`, the self-signed certificate will be generated for the hostname to where SABRE Console is being deployed.
context_sensitive_help_lb_url	N/A	The URL of the Load Balancer that serves the Help application in the format: `://<lb_hostname>(:<lb_port>)` It must be defined for the HA installation, i.e. when there is more than one Help server to be defined.`Port` is optional and must be specified only when the non-standard port is used

5. 2. 23 ActiveMQ (Data Export Only) Properties

Properties for the export of ActiveMQ data.

Property	Default Value	Description
amq_broker_persistence_directory	/opt/activemq/data	The location of an existing ActiveMQ installation's data directory on the file system. Messages and queues will be migrated from here into Artemis.

5. 2. 24 Artemis Properties

ActiveMQ Artemis message queues can be monitored in the Administration Console.

Property	Default Value	Description
artemis_heap_size_allocations:	2 GB	The size of the Artemis heap memory. Minimum 128 MB. Please ensure there is enough memory on the host to accommodate allocated heap sizes.
artemis_broker_use_ssl	false	Flag that indicates whether the SSL listeners should be activated
artemis_broker_password	N/A	The password of the Artemis Broker keystore when the Artemis Broker is set to use SSL.
artemis_broker_port	61616	Artemis Broker Port.
artemis_console_lb_url	N/A	The URL of the Load Balancer that serves the Artemis application in the format: `://<lb_hostname>:{<lb_port>}` It must be defined for the HA installation, i.e. when there is more than one Artemis server to be defined. `Port` mandatory and Protocol must be one of `tcp` or `ssl`.
artemis_broker_scheduler_support:	true	Whether the ActiveMQ Broker scheduler should be enabled.
artemis_broker_persistence_directory:	/opt/activemq/data	When not persisting messages to the CMP database, the Artemis Broker will store messages in this location on the file system.
artemis_console_enabled	true	Whether the Artemis console should be enabled.
artemis_console_use_ssl	N/A	Whether the Artemisconsole should use SSL
artemis_console_keystore_password	N/A	The password of the Artemis Console keystore when theArtemis Console is set to use SSL.
artemis_console_port	8161	Artemis Console Port
artemis_console_lb_url	N/A	The URL of the Load Balancer that serves the Artemis console in the format: `://<lb_hostname>:{<lb_port>}` It must be defined for the HA installation, i.e. when there is more than one

Property	Default Value	Description
		Artemis server to be defined.`Port` is optional and only must be specified when the non-standard port is used.
artemis_console_admin_users:	N/A	Users with read/write access to the Artemis Console. Format: `'<username>:<password>'`
artemis_console_readonly_users:	N/A	Users with read-only access to the ArtemisConsole. Format: `'<username>:<password>'`
ssl_certificates	N/A	This property is a dictionary of elements: "private_key"and "certificate". Each element is the path to the corresponding PEM file. If specified and `samservices_use_ssl` is `true`, the deployment playbook will use the specified files to enable the HTTPS listener. If the property is omitted and `samservices_use_ssl` is `true`, the self-signed certificate will be generated for the hostname to where SABRE Console is being deployed.

5. 2. 25 SPaRC Engine Properties

The following properties can be modified when deploying the pricing and rating engine:

Property	Default Value	Description
sparc_admin_user	N/A	The name of the SPaRC Engine administrator required to execute API calls to the server.
sparc_admin_password	N/A	The password of the SPaRC Engine administrator required to execute API calls to the server.
sparc_jdbc_password	N/A	The password for the user that the SPaRC Engine uses to connect to the database (only used internally within CMP).
sparc_port	21222	The SPaRC Engine port
sparc_lb_url	N/A	The URL of the Load Balancer that serves the SPaRC Engine application in the format: `://<lb_hostname>:{<lb_port>}` It must be defined for the HA installation, i.e. when there is more than one SPaRC Engine server to be defined.`Port`mandatory and Protocol must be one of `tcp` or `ssl`.
sparc_use_ssl	false	Flag indicating whether the SSL listeners should be activated
ssl_certificates	N/A	This property is a dictionary of three elements: "private_key", "certificate" and "ca". Each element is the path to the corresponding PEM file. If specified and `jboss_use_ssl` is `true`, the deployment play will use the specified files to enable the HTTPS listener. If the property is omitted while `jboss_use_ssl` is `true`, the self-signed certificate will be generated for the hostname to which JBoss is being deployed.
Advanced Properties		
sparc_hz_cluster_port	2770	Hazelcast Cluster Port
sparc_hz_multicast_enabled	False	Whether or not Hazelcast Multicast is enabled.
sparc_hz_multicast_group	N/A	The name of the Hazelcast Multicast group.
sparc_allocated_heap_size	8 GB/4GB	The size of the SPaRC Engine heap memory. Please ensure there is enough memory on the host to accommodate allocated heap sizes.

5. 3 Deploy CMP

To deploy the CMP stack to the target system, run the following command:

```
ansible-playbook --vault-password-file=<vault_password_file> -i
<inventory_file> -b playbooks/deploy.yaml
```

See "Common Ansible Command Line Parameters" below to get details about combinations of the command line parameters that can be used.

5. 3. 1 Common Ansible Command Line Parameters

The `ansible-playbook` is quite flexible, allowing users control the way it accesses the target environment. Depending on the deployment scenario, a combination of the command line parameters that are described in the following table can be used:

Parameter	Description	Example
-e '<string>'	Submit the Ansible play parameter to the play. Has the highest precedence over the parameters with the same name defined elsewhere	`-e 'jdbc_password=Password321'`
-b	Runs all the tasks in the play with elevated access (using sudo). This is mandatory parameter. It must always be used when running Ansible playbooks.	`-b`
-k	Prompt user for connection password. If specified, the user will be prompted to enter the password for the user specified in the `ansible_ssh_user` parameter	`-k`
-K	Prompt user for the sudo password. If specified, user will be prompted to enter the remote username password. Usually it is defined by the `ansible_ssh_user` parameter. Can be required when the user in question requires a password to gain the sudo access but the SSH private key is used to access the remote system.	`-K`
--ask-vault-pass	Prompt user for the vault password. If specified, the user will be prompted to enter the vault password.	`--ask-vault-pass`
--vault-password-file	Points Ansible command to the vault password file.	`--vault-password-file=~/.vaultpwd`
-i	Points Ansible to the inventory file.	`-i inventory/test.yaml`

Rather than deploying all CMP components in one go as described above, you can deploy individual components of the system using the playbooks files stored in the `playbooks` directory. The `deploy.yaml` playbook referenced in the overall installation above simply orchestrates these component-specific playbooks.

If the playbooks are executed individually; it is important to note that they have the following dependencies:

- Three components must be installed in order:
  1. Database (deploy-db.yaml)
  2. WSO2IS (deploy-wso2is.yaml)
  3. Role-extender (deploy-role-extender.yaml)

- JBoss must be deployed prior to the following: agent-view-installation-layer, published-interfaces-layer, SOAP WS and configuration-centre.
- JWS must be deployed prior to agent-view and wso2is-login.
- Once JBoss and JWS are deployed, then the dependant components and any remaining components can be deployed in any order.

This table describes the deployment playbooks and summarises the dependencies between them:

Playbook Name	Depends On	Description
deploy-db.yaml		Deploys postgres database
deploy-wso2is.yaml	deploy-db.yaml	Deploy WSO2 Identity Server
deploy-role-extender-.yaml	deploy-db.yaml deploy-wso2is.yaml	Deploy Role Extender Application
deploy-jboss.yaml	deploy-db.yaml	Deploy JBoss Application Server
deploy-jws.yaml		Deploy JBoss Web Server
deploy-aview-iface.yaml	deploy-db.yaml deploy-wso2is.yaml deploy-role-extender.yaml deploy-jboss.yaml	Deploy AgentView Interfaces Layer
deploy-wso2is-login.yaml	deploy-jws.yaml	Deploy the WSO2 IS custom login application
deploy-agent-view.yaml	deploy-jws.yaml deploy-wso2is.yaml deploy-aview-iface.yaml	Deploy AgentView
deploy-pil.yaml	deploy-db.yaml deploy-wso2is.yaml deploy-role-extender.yaml deploy-jboss.yaml	Deploy Published Interfaces Layer
deploy-soapws.yaml	deploy-db.yaml deploy-wso2is.yaml deploy-jboss.yaml	Deploy SOAP Web Services
deploy-sabre-server.yaml	deploy-db.yaml	Deploy Sabre Server Application
deploy-sabre-console.yaml	deploy-db.yaml deploy-wso2is.yaml	Deploy Sabre Console Application
deploy-rest-ws.yaml	deploy-db.yaml deploy-wso2is.yaml	Deploy REST Web Services
deploy-configuration-	deploy-db.yaml	Deploy Configuration

Playbook Name	Depends On	Description
centre.yaml	deploy-wso2is.yaml deploy-role-extender.yaml deploy-jboss.yaml	Centre
deploy-pentalo-server.yaml	deploy-db.yaml deploy-wso2is.yaml deploy-role-extender.yaml deploy-jboss.yaml	Deploy Pentalo Server

The individual playbooks can be executed independently, providing that the playbooks they depend upon have already been executed.

5. 3. 2 Summary File

Once the `deploy.yaml` script has successfully completed execution, a new file will be produced. The file outlines the summary that includes the URLs of the installed components as well as location of the component log files. The summary filename matches the name of the inventory file used (without the `.yaml` extension) with the suffix, `summary.txt`.

5.3.2.1 Example Summary File

```
CMP 8 is successfully deployed.

AgentView

==========

URL: https://server.example.demo.com:8443/agent-view

Logs:

Host: server.example.demo.com

Path: /var/log/webswing/

Sabre Console

==============

URL: https://server.example.demo.com:31212/

Logs:

Host: server.example.demo.com

Path: /var/log/sabre-console/
```

```
Sabre Server
==============
URL: https://server2.example.demo.com:21212/
Logs:
Host: server2.example.demo.com
Path: /var/log/sabre-server/
Configuration Centre
====================
URL: https://server2.example.demo.com:8443/config
SOAP WS
====================
URL: https://server2.example.demo.com:8443/ws
JBoss
======
Logs:
Host: server2.example.demo.com
Path: /var/log/eap7/server.log
REST WS
==============
URL: https://server.example.demo.com:9000/
Logs:
Host: server.example.demo.com
Path: /var/log/rest-ws/boot.log
WSO2 IS
=========
URL: https://server.example.demo.com:9443
Logs:
Host: server.example.demo.com
Path: /var/log/wso2carbon/wso2carbon.log
```

6.0 Upgrading an Existing Installation

Prerequisites:

- If CMP was originally deployed with PostgreSQL version 12, PostgreSQL must be upgraded to version 13 or the upgrade will fail.
- If CMP was originally deployed using a control server with Ansible 2.8 installed, that Ansible is upgraded to 2.9 or later.
- Complete pre-installation tasks.
- Review the *Installation Configuration Tool* section of this guide to identify new parameters, or changes to existing parameters, that you may need for your upgrade.
- Prepare the inventory file using the Installation Configuration Tool.

To upgrade an existing CMP installation, use the same `ansible-playbook` command to deploy a new installation:

```
ansible-playbook --vault-password-file=<vault_password_file> -i
<inventory_file> -b playbooks/deploy.yaml
```

The `playbooks/deploy.yaml` file and the individual playbooks that `deploy.yaml` calls can be executed multiple times. At each run the playbooks will:

- Deploy the latest CMP RPMs
- Recreate the necessary configuration file

It is safe to run the `deploy.yaml` playbook multiple times as long as the principal properties in the inventory file used during the execution are not modified. The principal properties of the inventory file are:

- Group hosts
- Ports allocated to each component
- Credentials used within the CMP components

The group hosts must not be modified under any circumstances. If there is a need to change a host in the group ,the CMP stack must currently be uninstalled first using the old inventory file. After that the hosts can be changed and the `deploy.yaml` playbook executed again with the modified inventory file.

Ports in theory can be changed, however the old ports must be excluded from the firewall configuration (if firewall is enabled on the target host).

Database credentials can be changed in the inventory file only to reflect manual changes. For example, if an administrator changes a database user password, the new password in question must be modified in the inventory file prior to the next upgrade.

The same applies to the application credentials.

In a similar way to running an upgrade, you can also run a deployment to ONLY update the SSL Certificates installed in the various backend systems. To do this, first make sure

the new certificates are in the correct location as defined in your inventory file under the `ssl_certificates` section.

Then use the same `ansible-playbook` command as previously but adding to the end of it `--tags ssl_configuration` as shown below:

```
ansible-playbook --vault-password-file=<vault_password_file> -i <inventory_
file> -b playbooks/deploy.yaml --tags ssl_configuration
```

6. 1 Patch Updating

There is an option to run a deployment to ONLY patch update the installed CMP components where necessary. This option uses the Ansible tags in order to skip running unnecessary tasks and run only those tasks required for updating the RPMs that may have been patched for the CMP system.

This method runs through the playbook, installs only those RPMs that have been patched/updated and performs a service restart, thereby reducing downtime by affecting only those updated components.

When you are ready to patch update, first ensure you read and adhere to the above notes on updating an existing installation and then use the same ansible-playbook command as previously but adding to the end of it `-e patch_update=true --tags installation` as shown in the following example:

```
ansible-playbook --vault-password-file=<vault_password_file> -i
<inventory_file> -b playbooks/deploy.yaml -e patch_update=true --
tags installation
```

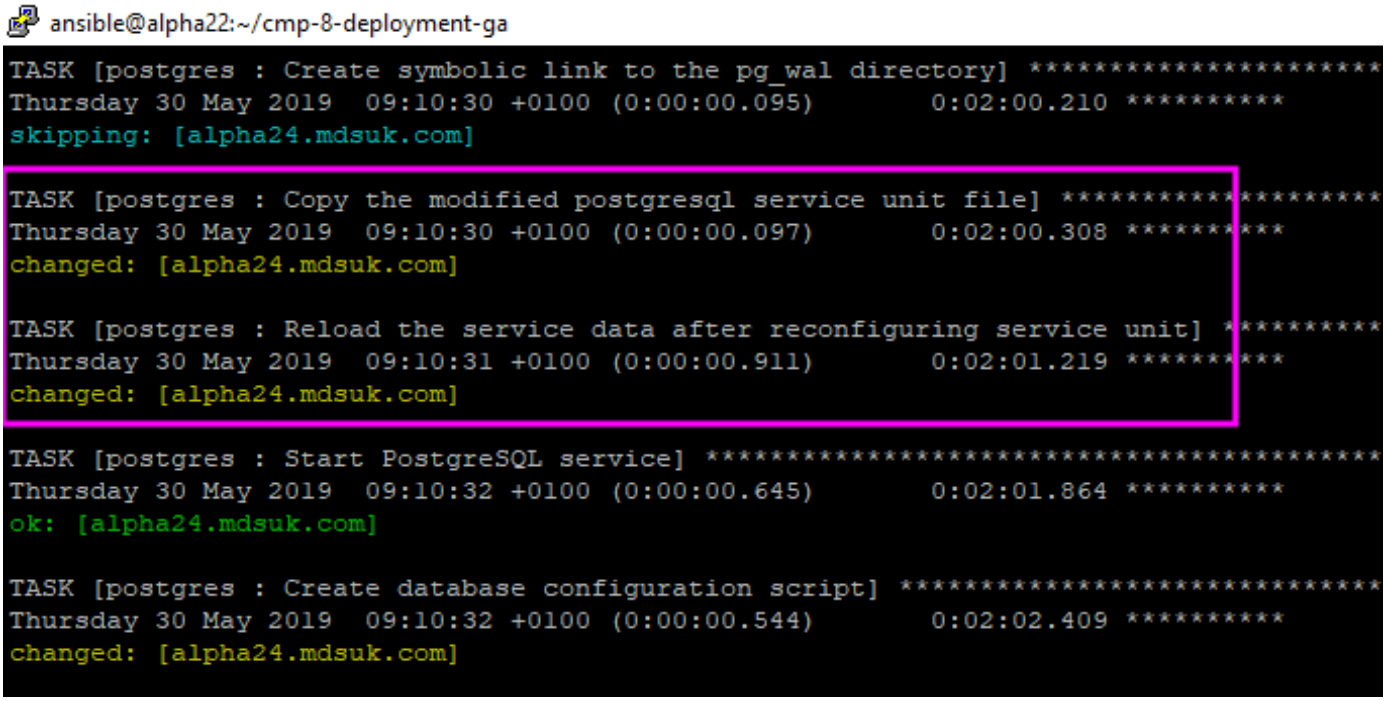
7.0 Troubleshooting

Typically, if anything is wrong with the host environment, the playbook stops running and the installation fails.

Ansible produces a status report that you can consult for troubleshooting. The lines in the status report are colour-coded as follows:

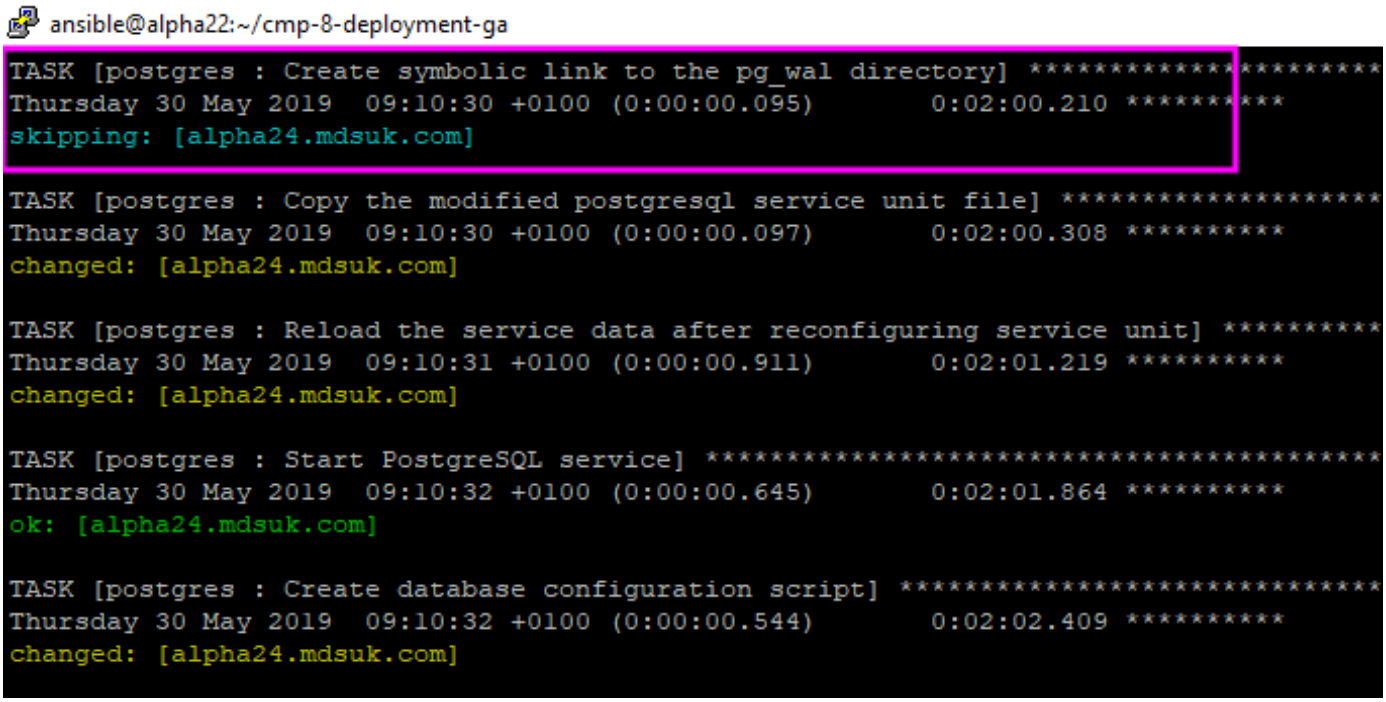
- Yellow

  Yellow text indicates that Ansible made changes to the target host. For example:



- Blue

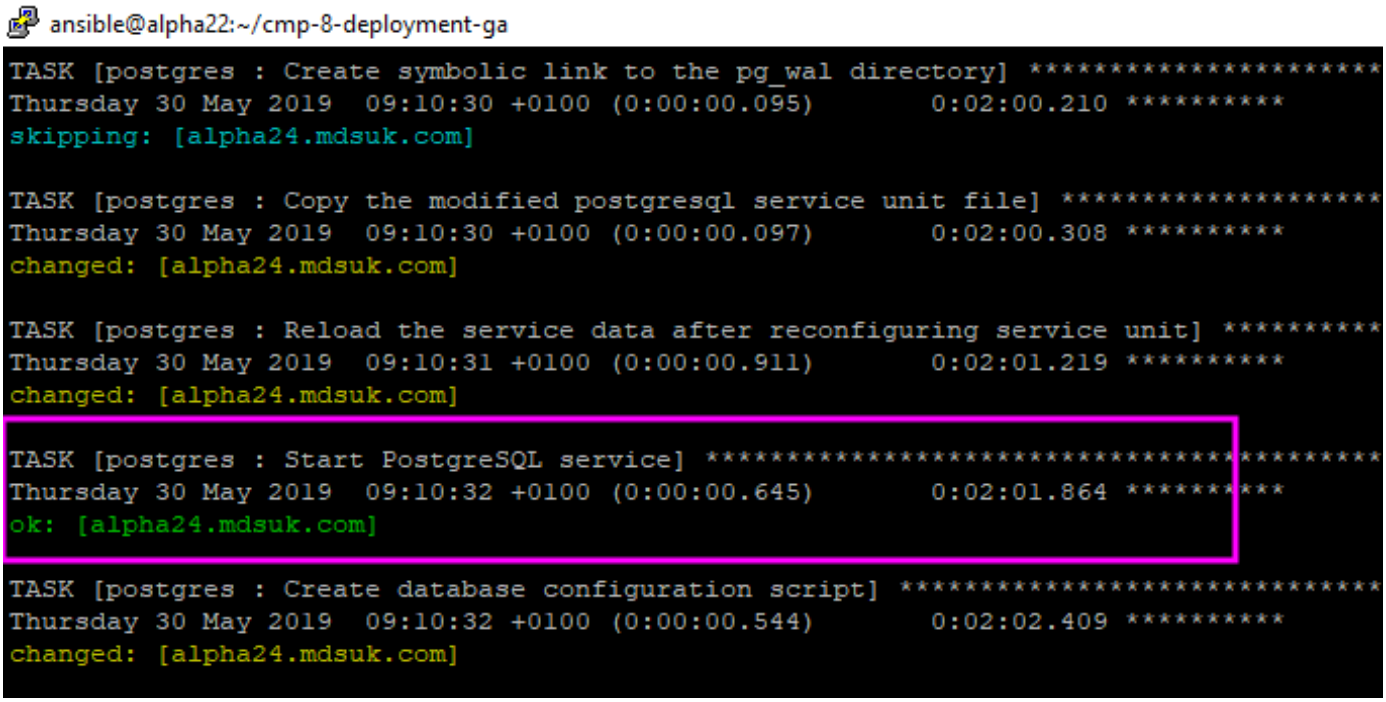
  Blue text indicates that the task was not executed because the condition attached to the file was false. For example:



If the task that caused the error is followed by a blue line '...ignoring', this means that task in question has the `ignore_errors` property set to `true`, so even in the case of an error, playbook execution will not be interrupted. This is normal execution for the CMP installation.

- Green

  Green text indicates that the task was executed but the state of the target host did not change. For example:



- Red

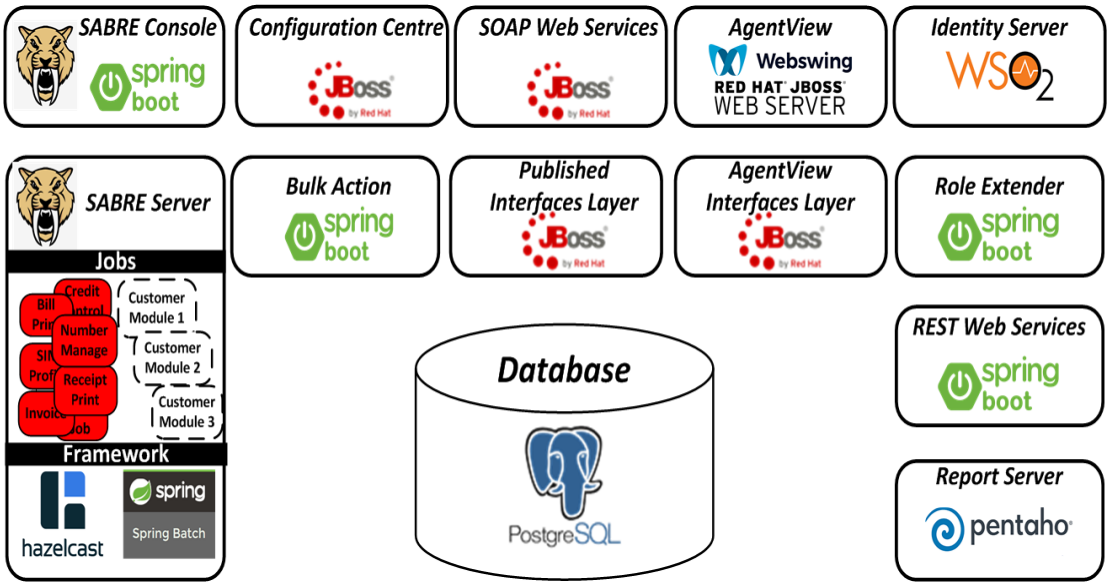
  Red text indicates that there was an error. For example:

```
TASK [firewall-rule : Check if firewalld is running] ***********************************
Thursday 30 May 2019  09:19:00 +0100 (0:00:00.088)       0:01:27.672 **********
fatal: [alpha24.mdsuk.com]: FAILED! => {
    "changed": true,
    "cmd": "service firewalld status",
    "delta": "0:00:00.020468",
    "end": "2019-05-30 09:19:00.913781",
    "rc": 3,
    "start": "2019-05-30 09:19:00.893313"
}

STDOUT:

● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enable
   Active: inactive (dead)
     Docs: man:firewalld(1)


STDERR:

Redirecting to /bin/systemctl status firewalld.service


MSG:

non-zero return code
```

Some tasks can fail by design; but this is normal as the error is ignored by the play.

Once the deployment has completed, if CMP does not work as expected, you need to go through the application logs. The location of the component logs is in the summary file produced by the Ansible play. For more information, see "Summary File" on page 81

As part of troubleshooting, it may be necessary to examine the CMP technical architecture to determine the components that each part of the system depends on. For example, if AgentView doesn't work then the AgentView Interfaces layer, WSO2 and the database are all potential sources of error depending on exactly where the application problem is seen. For more information, see the *CMP Technical Architecture Guide.*



Examining the technical architecture of CMP can help to troubleshoot an installation

8.0 Post-Installation Checks and Tasks

The deployment playbooks have built-in post deployment checks to confirm that all components are started. However, this does not guarantee that application is in a normal state.

It is recommended that post deployment you manually log into the each of the application that has a UI and confirm that it works as expected by performing a set of standard operations. If there are failures, check the logs as sometimes some of the services may need to be restarted.

CMP installs the following services:

Component	Service Name
PostgreSQL	postgresql-12
WSO2 IS	wso2is
JBoss	eap7-standalone
JWS	jws5-tomcat
Role Extender	role-extender
SABRE Server	sabre-server
Administration Console	sabre-console
REST WS	rest-ws

All CMP components are configured to start as `systemd` services. The standard `systemctl` command should be used to control the component services.

To check the status of a service - to see if you need to restart it, for example - run the following command:

```
sudo systemctl status <service_name>
```

To start a service run the following command:

```
sudo systemctl start <service_name>
```

To stop the service run the following command:

```
sudo systemctl start <service_name>
```

And to restart the service run the following command:

```
sudo systemctl restart <service_name>
```

8.0 High Availability Deployment Post-Installation Tasks

Note:

Configuring the replication and the load balancing is not part of the CMP deployment and needs to be configured using relevant third-party software and hardware as a separate step to the installation itself.

8. 1 Database

8. 1. 1 Replication

After deployment the primary PostgreSQL server should be configured and ready. The secondary server specified has the PostgreSQL software installed but the database is not configured. It is from this point that the user should configure the replication mechanism to suit the needs of their organisation.

The basic forms of replication are as follows:

- Shared Disk Failover
- Filesystem Replication
- Write Ahead Log Shipping
- Write Ahead Log Streaming
- Logical Replication
- Trigger-Based Master-Standby

The most common of replication is physical, using Write Ahead Log Streaming, which is the in-built PostgreSQL mechanism.

Any automatic failover mechanisms, synchronous/asynchronous configuration, network configuration and other implementation details are outside the scope of this document.

8. 1. 2 Archive Logging

In a Production system, the database should be able to recover from a point in time (PITR recovery), for which archive logging must be set up. Archive logging allows the database to save all the logging. Along with database backup, this allows you to recover the database to a particular point in time (or just the latest point possible). This feature must be enabled because it is not by default. The location of the logs also needs to be configured.

To enable archive logging, run the following within Postgres as an admin user:

```
ALTER SYSTEM SET wal_level TO 'hot_standby';
ALTER SYSTEM SET archive_mode TO 'ON';
ALTER SYSTEM SET hot_standby TO 'ON';
```

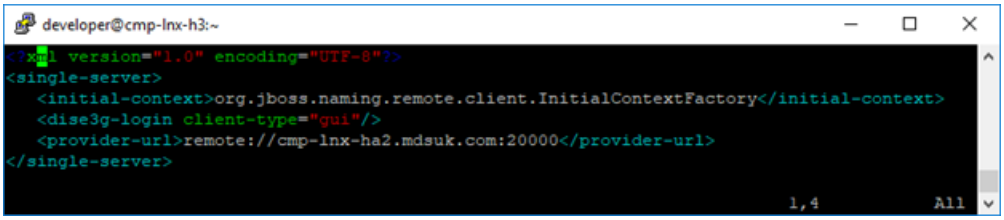
The following statement is an example of the command to copy the logs to an archive location and also compress them:

```
ALTER SYSTEM SET archive_command TO 'test ! -f /pg_archivelogs/%f
&& cp %p /pg_archivelogs/%f;gzip -5 /pg_archivelogs/%f ';
```

Modify the statement above to suit the requirements and/or installation of the system. After the above commands have been executed, restart the database and confirm that the archive logging is working as expected.

8. 2 AgentView

Make sure that all instances of the AgentView are configured to communicate with dedicated instance of the AgentView Interfaces Layer application. To do this open the `/etc/mdsglobal/agent-view/LoginComponent.xml` file in the editor and make sure that value of the `provider-url` tag points to a correct AgentView Interfaces Layer instance. For example:



In the most common scenario there will be a dedicated instance of the AgentView Interfaces Layer application to each instance of AgentView. However, there can be scenarios when one instance of the AgentView Interfaces Layer serves more than one instance of AgentView.

9.0 Uninstall CMP

To uninstall the CMP stack run the following command:

```
ansible-playbook --vault-password-file=<vault_password_file> -i
<inventory_file> -b playbooks/uninstall.yaml
```

10.0 Appendix

10. 1 Appendix A: Default Ports Allocation

The following table presents the default ports that are assigned to various components of the CMP stack:

Port	Port Number
Database server port	5432
WSO2 IS HTTP port	9763
WSO2 IS HTTPS port	9443
Role extender server port	8081
JBoss HTTP listener port	8080
JBoss HTTPS listener port	8443
JBoss Management HTTP port	9990
JBoss Management HTTP port	9993
JBoss Remoting port (used to communicate with CMP Services layer)	20000
Primary Server port	20000
JWS HTTP listener port	7080
JWS HTTPS listener port	7443
Administration Console port	31212
Hazelcast cluster port for SABRE Server	27300
Hazelcast log4j2 cluster port for SABRE Server	27400
SABRE Server management port	21212
REST Web Services port	9000
Bulk Action Console	9009

10. 2 Appendix B: Inventory Example

10. 2. 1 CMP Multiple Host Inventory Example

```
all:
 vars:
  installation_id: GAMMA-POSTST3
  jdbc_server: cmp-lnx1.mdsuk.com
  jdbc_database_name: mdscmp
  jdbc_schema_name: cmp
  jdbc_server_port: 5432
  yum_repository_username: test.username
```

```
  yum_repository_password: !<!vault> >-
   $ANSIBLE_VAULT;1.1;AES256
   623032656139356265623033313036366531623738336236376333333383-
965363323963306231
   626236313464656338363666376634346431343832326234 0a63383138323-
5336462613461353865
   316561376536386265646635646562346532353231373734313864343764 6-
165343831646338633 5
   656233373533363637 0a663261626435383438626234656236373831636 26-
6633539396632653937
   3430
  verify_ssl_certificates: true
  should_deploy_db: true
  should_upgrade_db: false
  db_data_population: Business_Configuration
  ansible_connection: ssh
  ansible_ssh_user: ansible
  #http_proxy: 'http://192.168.208.10:8080'
  #no_proxy: mdsuk.com
  mail_server:
   smtp_port: 25
   smtp_auth: false
   smtp_starttls: false
   smtp_from: smtp_user
  common_application_properties
   sam_name: lifecyle
   internal_webservice_password:!<!vault> >-
   $ANSIBLE_VAULT;1.1;AES256
   623032656139356265623033313036366531623738336235633836366376-
6343464313438323262 34
```

```
   6262363134646563383636663766343464313438323262340a63383138323-
533646261346135386
5

   3165613765363862656466356465623465323532313737343138643437646-
165343831646338633
5

   6562333735333563383636663766343464313438323262340236373831699-
456735393966326539
3

   3567

   log_collector_sftp_password:!<!vault> >-

   $ANSIBLE_VAULT;1.1;AES256

   6230326561393562656230333130363665316237383362363763333333383-
965363332396330623
1

   6262363134646563383636663766343464313438323262340a63383138323-
533646261346135386
5

   3165613765363862656466356465623465323532313737343138643437646-
165343831646338633
5

   6562333735333636370a663261626435383438626234656236373831699456-
733539396632653937

   3234

  health_check:

   use_ssl:

   jdbc_password:

   http_port:

  company_name: MODEL

  branding_code: MODEL

  yum_repository_url: 'https://vault.m-
dsglobal.dev/yum/ga/cmp/latest'

  third_party_yum_repository_url: 'https://vault.m-
dsglobal.dev/yum/third-party'

  maven_repository_url: 'https://vault.m-
dsglobal.dev/maven/ga/latest'

  third_party_yum_repository_username: '{{ yum_repository_user-
name }}'
```

```
  third_party_yum_repository_password: '{{ yum_repository_pass-
word }}'
  maven_username: '{{ yum_repository_username }}'
  maven_password: '{{ yum_repository_password }}'
  database: postgres
  use_redhat_subscription: true
  deploy_ntpd: false
  toolbox_activation_date_enabled: true
  toolbox_add_agreement_bundle_links_enabled: true
  toolbox_add_apn_enabled: true
  toolbox_add_non_default_bundles_enabled: true
  toolbox_addcreditcard_for_securitisation_enabled: true
  toolbox_aged_debt_enabled: true
  toolbox_allow_access_to_tef_pins_enabled: true
  toolbox_authorise_on_capture_enabled: true
  toolbox_avs_csc_responses_enabled: true
  toolbox_bill_viewer_enabled: true
  toolbox_bulk_data_loader_enabled: true
  toolbox_by_pass_validate_card_details_against_third_party_
enabled: true
  toolbox_cc_authorisation_details_enabled: true
  toolbox_cc_onceoff_payments_enabled: true
  toolbox_client_polling_enabled: true
  toolbox_clone_sub_and_manumberain_service_price_enabled: true
  toolbox_closed_user_group_rating_enabled: true
  toolbox_computer_telephony_numberegration_enabled: true
  toolbox_configured_tasks_enabled: true
  toolbox_convergent_prepay_postpay_enabled: true
  toolbox_corporate_deletion_enabled: true
  toolbox_correspondence_mailmerge_enabled: true
```

```
  toolbox_country_code_enabled: true
  toolbox_create_address_enabled: true
  toolbox_create_invoice_query_amount_enabled: true
  toolbox_credit_limit_review_enabled: true
  toolbox_customer_preferences_enabled: true
  toolbox_detailed_credit_card_details_enabled: true
  toolbox_diaryevent_duedate_indication_enabled: true
  toolbox_disable_bundles_enabled: true
  toolbox_disabled_recurring_card_payment_enabled: true
  toolbox_donor_led_porting_enabled: true
  toolbox_enable_add_remove_bundle_enabled: true
  toolbox_enhanced_call_unit_details_enabled: true
  toolbox_fup_and_capping_enabled: true
  toolbox_gprs_tariff_search_enabled: true
  toolbox_group_deletion_enabled: true
  toolbox_hardware_credit_limit_enabled: true
  toolbox_health_check_enabled: true
  toolbox_include_agents_phone_number_in_pci_redirect_payment_
enabled: true
  toolbox_numberernal_external_account_segregation_enabled: true
  toolbox_invoice_external_reference_enabled: true
  toolbox_log_termination_cost_request_enabled: true
  toolbox_managed_card_details_enabled: true
  toolbox_managed_service_sessions_enabled: true
  toolbox_mandatory_category_code_enabled: true
  toolbox_nsa_command_enabled: true
  toolbox_nsa_response_enabled: true
  toolbox_online_payments_enabled: true
  toolbox_opticals_enabled: true
  toolbox_optimised_address_search_enabled: true
```

```
  toolbox_override_bank_account_validation_enabled: true
  toolbox_parent_child_subscription_link_enabled: false
  toolbox_pci_compliant_enabled: true
  toolbox_pci_compliant_one_off_transactions_enabled: true
  toolbox_post_paid_caps_enabled: true
  toolbox_prescribed_alert_enabled: true
  toolbox_query_email_address_enabled: true
  toolbox_query_invoice_query_amounts_enabled: true
  toolbox_quick_address_enabled: true
  toolbox_record_batch_payment_transaction_enabled: true
  toolbox_record_product_item_serial_numbers_enabled: true
  toolbox_redirect_payment_enabled: true
  toolbox_refunds_enabled: true
  toolbox_remove_email_address_enabled: true
  toolbox_sales_order_courier_numbererface_enabled: true
  toolbox_separate_payment_authorise_and_capture_enabled: true
  toolbox_single_parent_subscription_for_account_link_enabled:
true
  toolbox_split_billing_enabled: true
  toolbox_subc_key_info_enabled: true
  toolbox_swms_order_numbererface_enabled: true
  toolbox_tariff_pricing_enabled: true
  toolbox_technology_funds_enabled: true
  toolbox_unbilled_calls_enabled: true
  toolbox_unbilled_unit_summary_enabled: true
  toolbox_vat_law_reverse_charge_enabled: true
  toolbox_warehousing_enabled: true
  toolbox_weblinks_enabled: true
  toolbox_wifi_purchase_and_usage_enabled: true
  toolbox_hardware_orders_enabled: false
```

```
 children:
  db:
   hosts:
    cmp-lnx1.mdsuk.com: {}
   vars:
    db_owner: deploy
    db_owner_password: !<!vault> >-
     $ANSIBLE_VAULT;1.1;AES256
     6364383866363661373639653939656132646536616364306638313537 3-
962343839393634303862
     6133336131643165663265323637396264333932303462620a383933636 -
36365646261353231613 7
     6135313035623933356438373539313734373039353236623133633164 3-
532386134303638626331
     3732643361613439350a313335326332343334386661646266356565323 -
561656361376639336339
     3637
    db_system_cpu: 4CPU
    db_system_mem: 8GBMemory
    db_system_dsk: SlowDisk
    os:
     min_performance: 100
     readahead: 4096
     min_preemption_granularity: 10000000
     scheduled_wakeup_granularity: 15000000
     dirty_ratio: 20
     dirty_background_ratio: 10
     swapiness: 10
     overcommit_memory: 2
     overcommit_ratio: 80
     huge_pages: 2500
```

```
  jboss:
   hosts:
    cmp-lnx2.mdsuk.com: {}
   vars:
    jboss_jdbc_username: jboss
    jboss_jdbc_password: !<!vault> >-
     $ANSIBLE_VAULT;1.1;AES256
     30396137653262303533313238336132656638303330366266373537666-
462663538343936353065
     66323965646361663033623861306566343062383963333 00a326633653-
73037383761326237373 8
     61666235616430346539616633613064383033376236613233643339616-
163316233376537393765
     65346634636663376 20a39356633643331633134633865323566303932 3-
039356365333161666635
     6661
    jboss_use_ssl: true
    jboss_https_port: 8443
    jboss_http_port: 8080
    jboss_management_http_port: 9990
    jboss_management_https_port: 9993
    jboss_remoting_port: 20000
    metaspace_size: 240M
    jboss_allocated_heap_size: 4096m
    max_metaspace_size: 340M
    jboss_dynamic_list_threshold: 20
    jboss_heartbeat_numbererval: 420000
    jboss_large_tree_threshold: 50000
    is_cmp_lite: true
  jws:
```

```
   hosts:
    cmp-lnx3.mdsuk.com: {}
   vars:
     jws_use_ssl: true
     jws_https_port: 7443
     jws_http_port: 7080
    jws_allocated_heap_size: 4096m
   pentaho:
    hosts:
     cmp-lnx3.mdsuk.com: {}
    vars:
     pentaho_admin_password: !<!vault> >-
      $ANSIBLE_VAULT;1.1;AES256
     326665343432316138353938393461373565383739316136663037363 96-
43462613934376233383 1
     623038353964356438636134646465313466663261366631 0a343937653-
764313165633732626665
     346563376362626661343336643664664396263346139373034333335396-
461643661353766383863
     313332383433623130 0a643166613364613331333539656230626362383-
831333534366462363561
     3435
    pentaho_jdbc_app_password: !<!vault> >-
     $ANSIBLE_VAULT;1.1;AES256
     356232323261356639383265373032396430393530303664356261393 86-
661396539666662306134
     646239633162396330343366353737366334393434323034 0a643865376-
666326466613766313062
     366433376166653232663434393333363135313737656137646337393 83-
365646539363031656664
     626663656637396665 0a363234653563363031613632643735653233363-
961653130646436636434
```

```
      3032
    pentaho_jdbc_reporting_password: '{{ pentaho_jdbc_app_pass-
word }}'
  wso2is:
   hosts:
    cmp-lnx3.mdsuk.com: {}
   vars:
    wso2is_admin_user: admin
    wso2is_admin_password: !<!vault> >-
     $ANSIBLE_VAULT;1.1;AES256
     61643035663161663737363233643837643237346539363135336339376-
36338346166363634663 2
     32626563623963333934336163303933663962393834343 10a336638303-
232313133313836323039
     38326335363130343832373231643534313636303239636531346161663-
138313437323362646665
     37633361303430393 00a393731643965376635353636376562633330623-
465323335366633643035
     3735
    wso2_jdbc_app_password: !<!vault> >-
     $ANSIBLE_VAULT;1.1;AES256
     61353234343239653338666263336566393339333264643239663864653-
636386465353262373239
     31393333643938316338366139633064373664396531346 50a623930646-
436663732396436646532
     30363163393038393836336630313765393239376135656534356532613-
862336631356238646465
     34616266353864383 60a313262343638333632326233333664383535343-
634336430373536316331
     6566
    wso2is_shared_storage_path: /opt/cmp-storage/wso2is
    well_known_context_secured: false
```

```
    should_use_external_database: true
    should_use_database_user_store: true
    wso2is_allocated_heap_size: 4096m
 shared_services:
   hosts:
    alpha22.int-dev.mdsuk.com:
   vars:
    sharedservices_use_ssl: true
    sharedservices_jdbc_password: !vault |
    $ANSIBLE_VAULT;1.1;AES256
    32343831643739633836626364313236326466333539323531333365333
7-6666326662336433663 5
    64393761663132666238623162353565316432343837643 20a3264336239-
34663035643632643365
    36323965613265313634353763386232373235333833343536316331306 5-
6664653932333764373 6
    36393361376337353 40a32393766376430343165373138313137333434 37-
62623732663566646334
    6338
    shared_services_password:  !vault |
    $ANSIBLE_VAULT;1.1;AES256
    32343831643739633836626364313236326466333539323531333365333
7-6666326662336433663 5
    64393761663132666238623162353565316432343837643 20a3264336239-
34663035643632643365
    36323965613265313634353763386232373235333833343536316331306 5-
6664653932333764373 6
    36393361376337353 40a32393766376430343165373138313137333434 37-
62623732663566646334
    6338
    sharedservices_port: 21218
    sharedservices_allocated_heap_size: 4096m
```

```
 sam_services:
   hosts:
    alpha22.int-dev.mdsuk.com:
   vars:
    samservices_use_ssl: true
    samservices_jdbc_password: !vault |
    $ANSIBLE_VAULT;1.1;AES256
    32343831643739633836626364313236326466323438316437396338366 2-
6666326662336433663 5
    64393761663132666238623162353565316432343837643 20a3264336239-
34663035643632643365
    36323965613265313634353763383234383164373963383662677331306 5-
66646539323337643736
    36393361376337353 40a32393766376430343165373138313137333434 37-
62623732663566646334
    6338
    sam_services_password:  !vault |
    $ANSIBLE_VAULT;1.1;AES256
    32343831643323438316437396338366266466333539323531333365333 7-
66663266623364336635
    64393761663132666238623162353565316432343837643 20a3264336239-
34663035643632643365
    36323965613265313634353763333234383164373963383662631633130 65-
66646539323337643736
    36393361376337353 40a32393766376430343234383164373963383662 67-
62623732663566646334
    6338
    samservices_port: 21220
    samservices_allocated_heap_size: 2048m
  rolext:
   hosts:
    cmp-lnx3.mdsuk.com: {}
```

```
    vars:
     rolext_jdbc_username: rolext
     rolext_jdbc_password: !<!vault> >-
      $ANSIBLE_VAULT;1.1;AES256
      643565366232613832663930623634613932633835653263626562396 43-
13263386332656439646 3
      343761323662383335636361303061373130363138346338 0a316338383-
632356332343961306235
      363865353832383762376538356661626531313237653130306266386 63-
734343230613732666366
      623233313936316132 0a623265376433376165666564623338373938613-
638306462306630393934
      6231
     rolext_server_port: 8081
     rolext_use_ssl: true
   agent_view:
    hosts:
     cmp-lnx3.mdsuk.com: {}
    vars: {}
             redirect_payment_rest_password: !<!vault> >-
      $ANSIBLE_VAULT;1.1;AES256
      623133306637303631303937383933613832383036663635353437653 03-
432613932336263626637
      393066343835626633653265333734323161326431393062 0a386562616-
238666131303935623663
      306562653435333564643663333036396633373263353138623438303 13-
732346333343333643537
      393936623162633164 0a646637303335343463353734323761326236396-
136663632666261323263
      6586
      agent_view_font: /usr/share/fonts/msttcorefonts/arial.ttf
   bulk_action_console:
```

```
   hosts:
    cmp-lnx3.mdsuk.com: {}
   vars:
    bulk_action_console_system_id: cmp-lnx-int
    bulk_action_console_jdbc_username: baconsole
    bulk_action_console_jdbc_password: !<!vault> >-
     $ANSIBLE_VAULT;1.1;AES256
     623133306637303631303937383933613832383036663635353437653 03-
43261393233626362663 7
     326539306135333731663266306264313131303930623 2640a623831666 -
663376433646461383363
     306562653435333564643663333036396633373263353138623438303 13-
732346333343333643537
     393936623162633 1640a646637303335343463353734323761326236396 -
136663632666261323263
     6532
    bulk_action_console_server_port: 9009
    bulk_action_console_use_ssl: true
    file_transfer:
     type: SFTP
     host: cmp-lnx2.mdsuk.com
     port: 22
     username: sabre
     password: !<!vault> >-
      $ANSIBLE_VAULT;1.1;AES256
      616665343061346537613261346530303463346666313336393734333 5-
383939623261383833616 2
      326539306135333731663266306264313131303930623 2640a623831 66-
666337643364646138336 3
      376233306665613439363861323264386533373336663663303135346 6-
653137633036393762613 1
```

```
      6538383539346537370a6362633636376635343036656465633135643
2-353138393365336262313 5
      3265
     directory: bulkaction/pending_transform/
     reports_directory: bulkaction/pending_transform/
    valid_email_domains: 'mdsuk.com,virtusa.com'
    bulk_action_allocated_heap_size: 2048m
  soapws:
   hosts:
    cmp-lnx2.mdsuk.com: {}
   vars: {}
  restws:
   hosts:
    cmp-lnx2.mdsuk.com: {}
   vars:
    restws_jdbc_username: restws
    restws_jdbc_password: !<!vault> >-
     $ANSIBLE_VAULT;1.1;AES256
     3139633534366439646433653736323833393533633933335623036326
63-33933838396465373932 35
     3437663333633330643434336161373137326136653566370a6161643
96-234613532376132346361
     6163333265303961393962396535636630356634653762383261366264
6-565326235643334383562
     3830663339333562350a6364656233633237386363373362343033316
13-633353964393765383237
     6165
    restws_server_port: 9000
    restws_use_ssl: true
    restws_allocated_heap_size: 4096m
  configuration_centre:
```

```
    hosts:
      cmp-lnx2.mdsuk.com: {}
    vars: {}
  aview_iface:
    hosts:
      cmp-lnx2.mdsuk.com: {}
    vars: {}
  pil:
    hosts:
      cmp-lnx2.mdsuk.com: {}
    vars: {}
  sabre_server:
    hosts:
      cmp-lnx2.mdsuk.com: {}
    vars:
      sabre_admin_user: admin
      sabre_admin_password: !<!vault> >-
        $ANSIBLE_VAULT;1.1;AES256
        66666364643062646563366535363161333763323731623030626534343-
834376465623462326639
        37623231336463376264376132636365613166663237356 60a316361343-
663931386461393438 31
        61343332323737323763626534343736333639386134623961613734646-
237633362383462396633
        396630313030663665 0a303262376134646233326438363234663330383-
264356465363662346637
        3839
      sabre_server_jdbc_username: cmpsabre
      sabre_server_jdbc_password: !<!vault> >-
        $ANSIBLE_VAULT;1.1;AES256
```

```
      6332636438653638383563646537663961653432306664376630666662 6-
137373431343265623061

      6661333635653163326132636138353639336437666135330a306339313 -
761376637323337653630

      6534323231333430363962616639343363303635346630363261386463 6-
537646164356462346464

      3332356566313836650a6637626362306131336233343639623133323 93-
335613634396465323836

      3365

     sabre_management_port: 21212

     sabre_server_use_ssl: true

     bulkaction_password: !<!vault> >-

    $ANSIBLE_VAULT;1.1;AES256

    6230326561393562656230333130363665316237383362363763333333383 -
9653633323963306231

    6262363134646563383636663766343464313438323262340a63383138323 -
5336462613461353865

    3332356566313836650a6637626362306131336233343639623133323933 3-
5613634396465323836

    6562333735333636370a6632616264353834386262346562363738316362 6-
6633539396632653937

    3427

     hz_cluster_port: 27300

     hz_log4j_cluster_port: 27400

     hz_multicast_enabled: false

     hz_multicast_group: 224.14.14.14

     sabre_allocated_heap_size: 4096m

     additional_modules:

      - sabre-addacs-transformation-adapter

      - sabre-analyser-extract

      - sabre-arucs-transformation-adapter

      - sabre-auddis-transformation-adapter
```

```
      - sabre-auddis-transformation-inbound-adapter
      - sabre-awacs-transformation-adapter
      - sabre-bacs-transformation-adapter
      - sabre-bill-pdf-collection-adapter
      - sabre-equifax-insight2001
      - sabre-experian-transformation-adapter
      - sabre-matrix-usage-transformation-daemon
      - sabre-offline-rating
      - sabre-openet-provisioning-adapter
      - sabre-openet-recurring-prepayment-adapter
      - sabre-openet-usage-adapter
      - sabre-recurring-bank-payments-bacs-adapter
      - sabre-simulator
  sabre_console:
   hosts:
    cmp-lnx3.mdsuk.com: {}
   vars:
    sabre_console_system_id: cmp-lnx-int
    sabre_console_jdbc_username: sabre
    sabre_console_jdbc_password: !<!vault> >-
     $ANSIBLE_VAULT;1.1;AES256
     376337343639646232626131316432623364373865343331643065663966-
166326336353064656461
     333431313834643165663862396339623330313439376137 0a613431396-
562386633633765386138
     323136376361373936343535396434333035653266343561303639336 13-
034393661636436393533
     303836363965323636 0a3038653134613832306133653336343431613 53-
339643333633936363035
     3864
    sabre_console_port: 31212
```

```
    sabre_console_use_ssl: true
    sabre_console_allocated_heap_size: 2048m
  context_sensitive_help:
   hosts:
    cmp-help.mdsuk.com: {}
   vars:
    context_sensitive_help_server_port: 21221
    context_sensitive_help_use_ssl: true
 activemq:
   hosts:
    amq_host: {}
   vars:
    amq_broker_persistence_directory: /opt/activemq/data
artemis:
   artemis_host:
    amq.mdsuk.com: {}
   vars:
    artemis_heap_size_allocations: 2048m
    artemis_broker_use_ssl: true
    artemis_broker_keystore_password: !<!vault> >-
     $ANSIBLE_VAULT;1.1;AES256
     376337343639646232626131316432623364373865343331643065663 96-
166326336353064656461
     333431313834643165663862396339623330313439376137 0a613431396-
562386633633765386138
     323136376361373936343535396434333035653266343561303639336 13-
034393661636436393533
     303836363965323636 0a303865313461383230613365333634343161353-
339643333633936363035
     3864
    artemis_broker_port: 8161
```

```
    artemis_broker_scheduler_support: true
    artemis_broker_persistence_directory: /opt/activemq/data
    artemis_console_enabled: true
    context_sensitive_help_use_ssl: true
    artemis_console_use_ssl: true
    artemis_console_keystore_password: !<!vault> >-
     $ANSIBLE_VAULT;1.1;AES256
     55897621456396462326261313164326233643738653433316430656639-
616632633633053064656461
     33343131383464316566386239633962333031343937613 70a613431396-
562386633633765386138
     32313637636137393634353539643433303565326634356130363933613-
034393661636436393533
     63396233303134393761947573653134613832306133653336343431613-
533396433336339363035
     9587
    artemis_console_port: 8161
    artemis_console_admin_users: joe.brown:admin33
    artemis_console_readonly_users: ann.green:anG560£
 sparc_engine:
   hosts:
    sparc_host: {}
   vars:
    sparc_admin_user: sparc
    sparc_admin_password: !<!vault> >-
     $ANSIBLE_VAULT;1.1;AES256
     30383636396532363 60a303865313461383230613365333634343161353-
339643333633936363035
     33343131383464316566386239633962333031343937613 70a613431396-
562386633633765386138
```

```
     376337343639646232626131316432623364373865343331643065663 96-
166326336353064656461

     323136376361373936343535396434333035653266343561303639336 13-
034393661636436393533

    sparc_jdbc_password: !<!vault> >-

     $ANSIBLE_VAULT;1.1;AES256

     376337343639646232626131316432623364373865343331643065663 96-
166326336353064656461

     323136376361373936343535396434333035653266343561303639336 13-
034393661636436393533

     333431313834643165663862396339623330313439376137 0a613431396-
562386633633765386138

     303836363965323636 0a303865313461383230613365333634343161353-
339643333633936363035

    sparc_port: 21222

    sparc_use_ssl: true

    sparc_lb_url: 'https://loadbalancer:2121'

    sparc_hz_cluster_port: 27700

    sparc_hz_multicast_enabled: true

    sparc_hz_multicast_group: mcgroup

    sparc_allocated_heap_size: 4g/8g
```