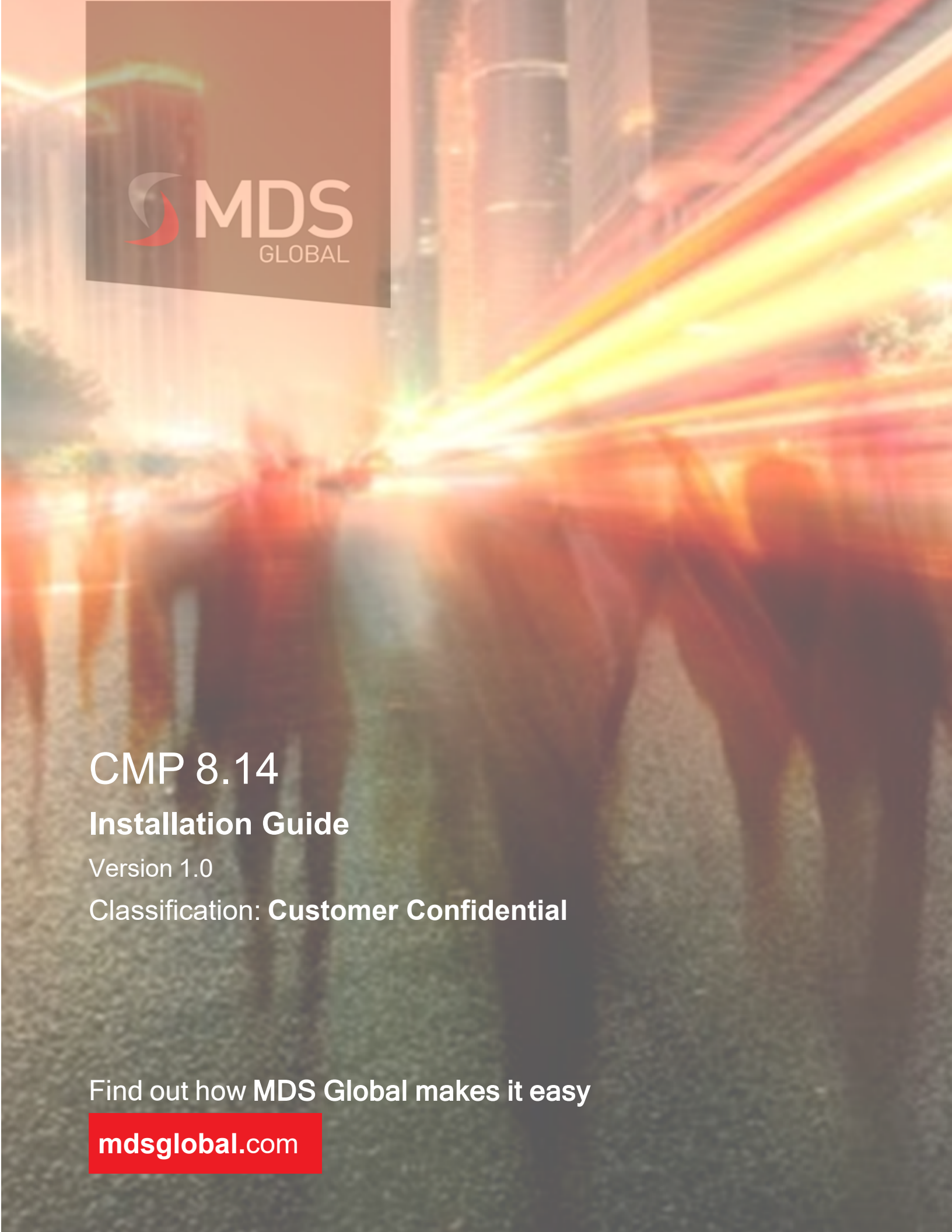


CMP 8.14

Installation Guide

Version 1.0

Classification: **Customer Confidential**

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

mdsglobal.com

Copyright

Confidentiality

Disclaimer

Trademarks

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

Version Control

Version	Issue Date	Author	Comments
Version 1.0	06 March 2024	MDS	CMP 8.14 Release - High Availability Prerequisites topic updated to include the shared storage directory for the Artemis messaging server, and the private key for decryption password updated in the SABRE Server Encryption topic.

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 Linux 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 Linux, networking and database administration 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.
- Ansible experience is beneficial.

2.0 About CMP Installation

CMP is delivered as a set of components that are RPM 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 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.

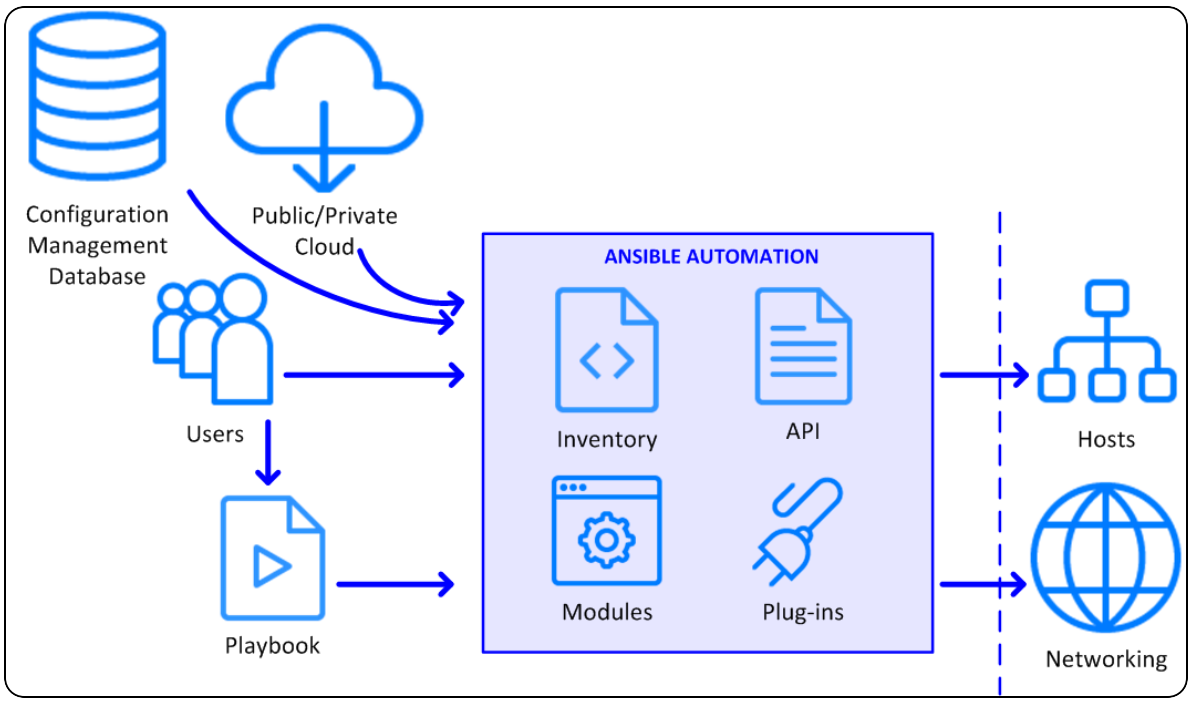
YUM (Yellowdog Updater, Modified) is an open-source command-line package-management utility for computers using the RPM Package Manager. YUM Repositories are warehouses of RPM package files that can hold RPM packages locally or remotely.

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 automate 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.

> The inventory file is created and modified using the Inventory Configuration Tool.

A number of host groups have been predefined for CMP; see the *Groups* table in *Prepare the Inventory File* for more information.

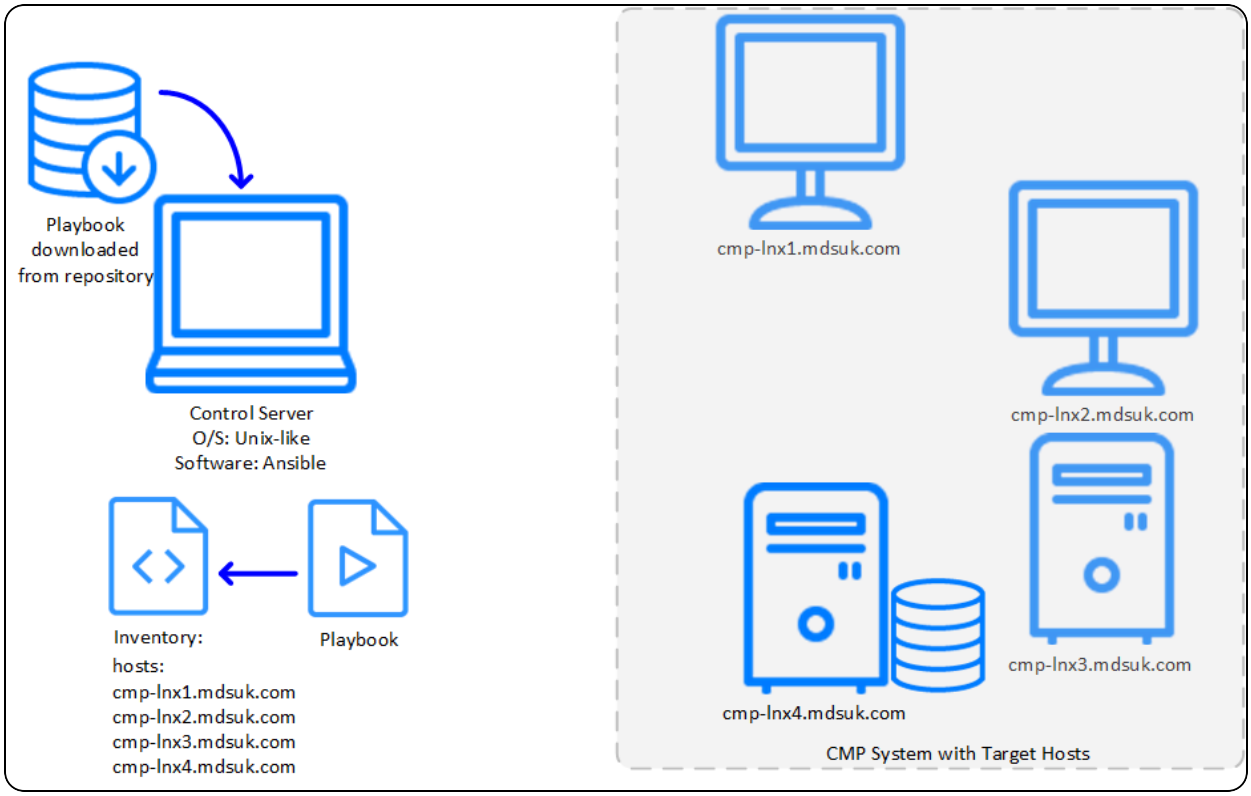
CMP provides a YAML inventory file template (`template.yaml`) that you must copy and edit to suit your deployment.

For more information, see Prepare the Inventory File.

2. 3 CMP Automatic Deployment Process

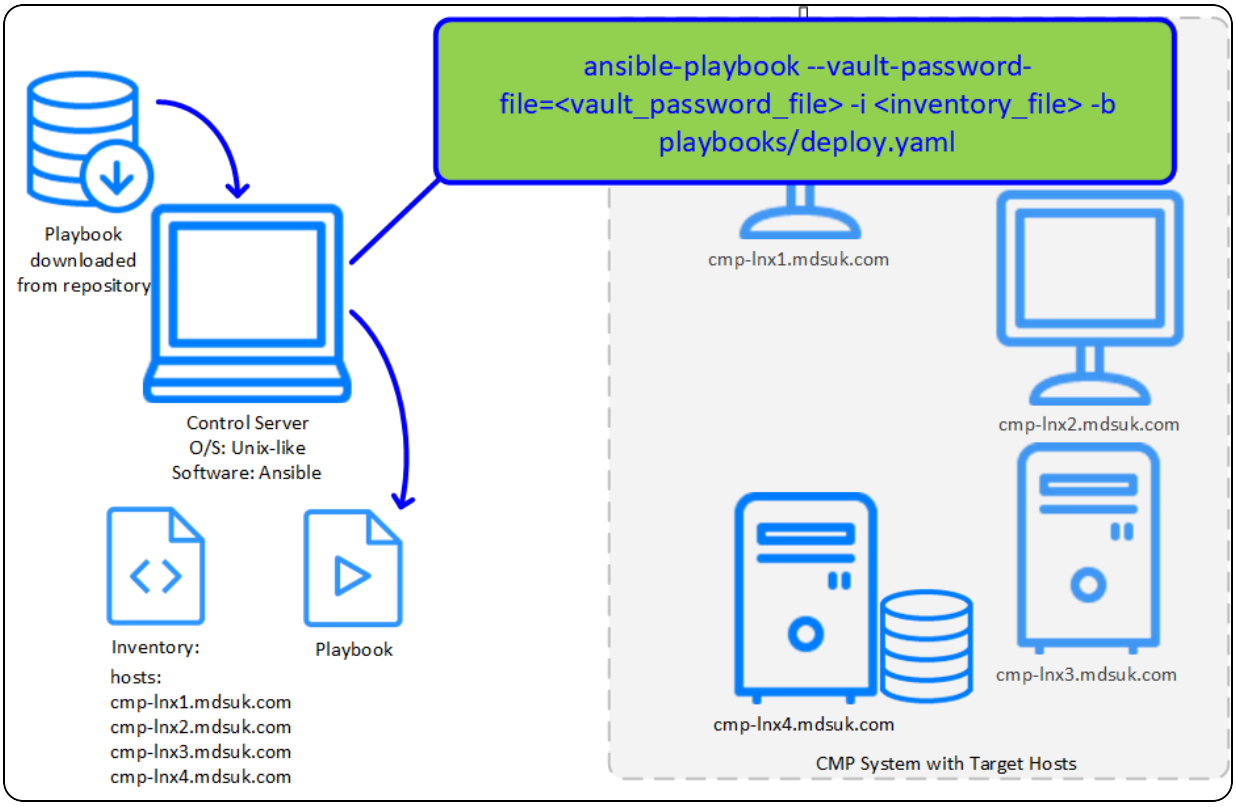
The Ansible deployment process used to deploy CMP is as follows:

Step 1



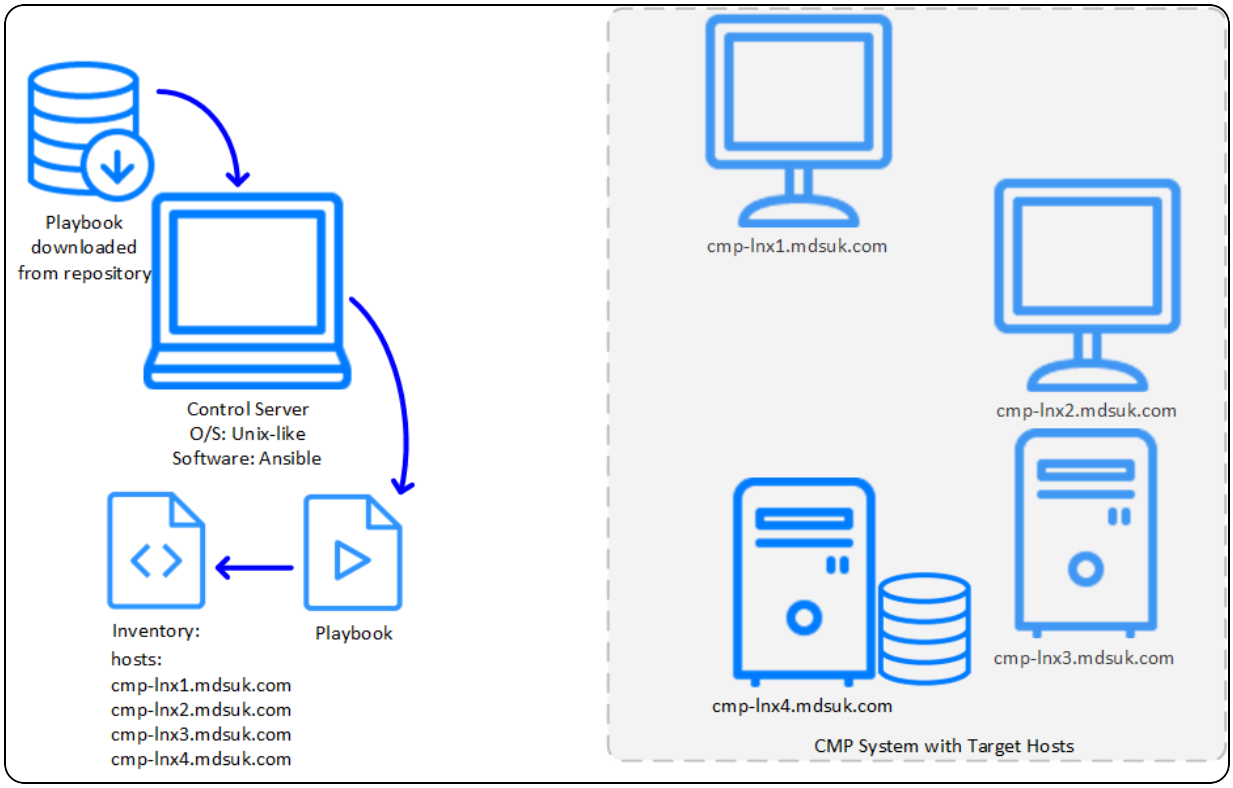
Download and unzip the Ansible Playbook package to the control server (controller node) and prepare the inventory file using the Inventory Configuration Tool.

Step 2



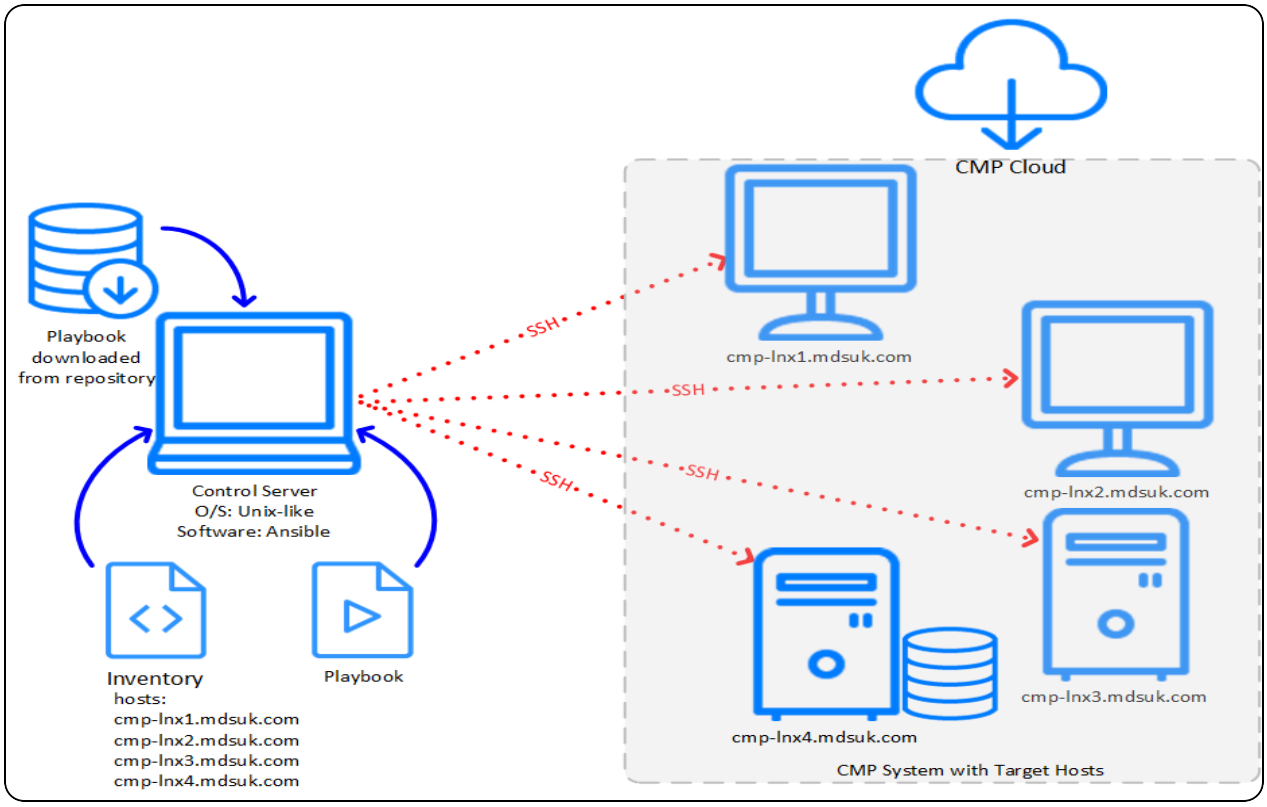
Run the command to execute the deployment playbook.

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.

Although Ansible supports both encrypted vault files to specifically hold sensitive data or encryption of individual fields directly in the inventory file, the CMP installation uses the encryption of individual fields method. If a vault password is specified when saving the inventory file from the Inventory Configuration Tool then all data sensitive fields will be encrypted.

When sensitive date is encrypted, 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 Memory Allocations

It is possible to adjust the allocated JVM heap size for individual CMP applications in some cases, in the expandable *Advanced Properties* section in the Installation Configuration Tool. These values should only be changed from the defaults after discussion with MDS Global or based on server sizing details provided by MDS Global.

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 instances don't hold any state so can be used indiscriminately.
- The connectivity into each component is load balanced over all instances of that component, with the load balancer using the "Health Check Service" on page 12 to determine if each instance should receive traffic.

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

[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.

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.

2. 7 Health Check Service

CMP provides a global health check service which it is mandatory for any external load balancer to use to determine the health of instances of CMP modules. The service checks the `cmpModuleMaintenance` table to determine whether the module is configured to be online, and then checks whether the module is running correctly and able to handle traffic. 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.

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 "Installation Configuration Tool" on page 29 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 7.x with Ansible Core 2.14 or Ansible Core 2.13 (both supported in CMP 8.15), python39-jmespath, and a version of the unzip utility installed.

  Note:

  If different Ansible and Ansible Core versions (higher or lower) are already installed on the control server they should be removed before installing the version above. If not readily available from the systems standard repositories, the correct Ansible version can be installed from an MDS Global hosted repository as follows:

  1. Remove any Ansible package exclusion in place by editing /etc/dnf/dnf.conf and removing any line stating exclude=ansible*

  2. Add the relevant MDS Global repository: sudo dnf config-manager --add-repo https://<username>:<password>@vault.mdsglobal.dev/repository/repos/cmp-prereq.repo replacing username and password with the repository credentials supplied by MDS Global for CMP installation

  3. Edit /etc/yum/repos.d/cmp-prereq.repo to replace <username> and <password> with the repository credentials supplied by MDS Global for CMP installation

  4. Install Ansible and Ansible Core: ***sudo dnf install ansible-7.2.0***

  5. Exclude updates for ansible packages by editing /etc/dnf/dnf.conf" and adding the following line exclude=ansible*

- The control server is locked to the above Ansible version so that is not inadvertently updated by Linux package updates (i.e. add a line with exclude=ansible* at the end of the /etc/dnf/dnf.conf file on the server).
- 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.

[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.

- 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 in the Inventory Configuration Tool for the installation, 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 22). If there are firewalls between the target servers other than the default Linux firewall (firewalld), the communications described in the intra stack communications table must be allowed.

  It is essential that on each host the external hostname resolves to the local IP address of the server that the server itself understands is its own address, and not to an external address which is routed to the server.
- 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 authorised 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` file on the control server.
  - sudo[1] access 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.
- 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:

[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.

- /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 19.

- Any security certificates required to explicitly trust a system that will be connected to from the CMP product itself, a customer specific adapter deployed with CMP, or any other process running on the server are installed in a subdirectory of the default systemwide trust store and not in the Java trust store, as the content of the Java trust store can be deleted as part of the CMP installation process.

  See "Shared System Certificate Storage" on page 21 for more.

- For Red Hat Linux installations only:
  - 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 repositories 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, load balancing 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 via the Installation Configuration Tool.
- 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.
  - HTTP 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 storage for the Artemis messaging server (ActiveMQ server) and its broker persistence directory, should be mounted to `/var/mdsglobal/amq-broker-/data` directory on all instances that host Artemis Server.
- Shared storage for the SABRE server (batch server) should be mounted to `/var/mdsglobal/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 page 19).

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:

The ports used are the standard ones in the Inventory Configuration. If they are changed as part of the installation then the ports accessed by the load balancer would need to be updated accordingly.

Service Name	Protocol	Default Port	Health Check Name	Timeouts	Affinity
AgentView / Pentaho	HTTP/HTTPS	Port 7443	/healthCheck/agent-view	28,800 seconds (8 hours)	Auto generated session Cookie
WSO2 Identity Server	HTTP/HTTPS	Port 9443	/healthCheck/wso2is	30 seconds	Client IP
Role Extender	HTTP/HTTPS	Port 8081	/healthCheck/role-extender	30 seconds	None
SOAP Web Services	HTTP/HTTPS	Port 8443		30 seconds	None
REST Web Services	HTTP/HTTPS	Port 9000	/healthCheck/rest-ws	30 seconds	None
Business Configuration	HTTP/HTTPS	Port 8443	/healthCheck/configuration-centre	30 seconds	Client IP
Bulk Action Console	HTTP/HTTPS	Port 9009	/healthCheck/bulk-action-console	30 seconds	Client IP
CMP Administration Console	HTTP/HTTPS	Port 31212	/healthCheck/sabre-console	30 seconds	Client IP
Artemis / ActiveMQ Broker	TCP/SSL	Port 61616	/healthCheck/apache-activemq-broker	30 seconds	None
Artemis / ActiveMQ Console	HTTP/HTTPS	Port 8161	/healthCheck/apache-activemq-console	30 seconds	Client IP
Shared Services	HTTP/HTTPS	Port 21218	/healthCheck/shared-services-rest	30 seconds	None
SAM Services	HTTP/HTTPS	Port 21220	/healthCheck/sam-services	30 seconds	None
Voucher Management Services	HTTP/HTTPS	Port 21223	/healthCheck/voucher-management-services	30 seconds	None

Service Name	Protocol	Default Port	Health Check Name	Timeouts	Affinity
Context Sensitive Help	HTTP/HTTPS	Port 21221	/healthCheck/context-sensitive-help	30 seconds	None
SPaRC Engine	HTTP/HTTPS	Port 21222	/healthCheck/sparc-engine	30 seconds	None

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 Installation 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, although the inventory can be configured to use HTTP if required. 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 provide information about the certificates via the Inventory Configuration Tool.

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. 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 can be obtained from MDS Global Professional Services prior to installation.

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 the default Linux firewalld 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. 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**
CMP Database	CMP PostgreSQL Database server.
JBoss/Wildfly Enterprise Application Platform	JBoss server instances. Required for AgentView Interface Layer, Published Interfaces Layer, Business Configuration and CMP Web Services – SOAP groups.
JBoss Web Server (JWS)/Tomcat	JBoss Web Server instances. Required for AgentView group.
Pentaho Reporting Server	The Report Server is an instance of a Pentaho server from Hitachi Vantara.
Identity Server	Identity server. CMP uses WSO2 Identity Server for this purpose.
Shared Services	Shared Services application instances.
SAM Services	SAM Services application instances.
Voucher Management Services	Voucher Management Services application instances.
Role Extender	Role extender application instances.
AgentView	AgentView application instances.
Bulk Action Console	The Bulk Action console application instances.
CMP Web Services - SOAP	SOAP Web Services application instances.
CMP Web Services - REST	REST Web Services application instances.
Business Configuration	CMP Business Configuration application instances.
AgentView Interface Layer	AgentView Interfaces Layer application instances.
Published Interfaces Layer	Published Interfaces Layer application instances.
SABRE Server	SABRE Server application instances.
CMP Administration Console	CMP Administration Console application instances.
Context Sensitive Help	Context Sensitive Help application instances.
Artemis	Artemis / ActiveMQ application instances.
SPaRC Engine	SPaRC Engine application instances.

Intra Stack Communications			
Source Hosts Groups	**Destination Host Groups**	**Port Numbers**	**Protocol**
JBoss/Wildfly JBoss Web Server (JWS)/Tomcat Identity Server Pentaho Reporting Server Shared Services SAM Services Voucher Management Services Role Extender Bulk Action Console CMP Web Services - REST SABRE Server CMP Administration Console SPaRC Engine Health Check Service	CMP Database	5432	JDBC
JBoss/Wildfly JBoss Web Server (JWS)/Tomcat Role Extender CMP Web Services - REST Bulk Action Console SABRE Server CMP Administration Console	Identity Server	9443	HTTPS
JBoss/Wildfly JBoss Web Server (JWS)/Tomcat SABRE Server CMP Administration Console	Role Extender	8081	HTTPS
CMP Administration Console SABRE Server	SABRE Server CMP Administration Console	27300 27400	HTTPS
JBoss/Wildfly JBoss Web Server (JWS)/Tomcat Identity Server	Health Check Service	21215	HTTP

Intra Stack Communications			
Source Hosts Groups	Destination Host Groups	Port Numbers	Protocol
Pentaho Reporting Server			
Shared Services			
SAM Services			
Voucher Management Services			
Role Extender			
Bulk Action Console			
CMP Web Services - REST			
SABRE Server			
CMP Administration Console			
Artemis / ActiveMQ			
SPaRC Engine			

The Port Numbers indicated in the table above are defaults, and will be different if changed in the inventory configuration.

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.

Even in a High Availability deployment with traffic going between components via a Load balancer, direct intra stack communication needs to be allowed as follows:

Source Host Groups	Destination Load Balancer
AgentView AgentView Interface Layer Bulk Action Console JBoss/Wildfly JBoss Web Server (JWS)/Tomcat Role Extender REST Web Services CMP Administration Console	Identify Server WSO2
AgentView Pentaho Reporting Server WSO2 IS Login	JBoss Web Server (JWS)/Tomcat
Business Configuration SOAP Web Services	JBoss/Wildfly

3. 4. 2 External Access

To provide user access to CMP, communication over the following ports must be allowed from the client networks. If `use_ssl` is selected in the Inventory Configuration Tool, the HTTPS ports listed need to be allowed, otherwise the HTTP ports must be allowed:

Target Host Groups	HTTP Port	HTTPS Port
JBoss/Wildfly (hosting AgentView, WSO2 Login SSO)	7080	7443
JBoss Web Server (JWS)/Tomcat (hosting Business Configuration, SOAP etc.)	8080	8443
Identity Server	9763	9443
Bulk Action Console	9009	9009
CMP Web Services - REST	9000	9000
SABRE Server	21212	21212
CMP Administration Console	31212	31212
Artemis / ActiveMQ	8161	8161

External access will also be required for all incoming and outgoing communication from Sabre Jobs that has been deployed as part of the customer specific CMP installation, for example, event records, sending email, reading from external queues. The network requirements in this area need to be determined as part of the CMP implementation project and are outside the scope of this document.

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 13)
- 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 `curl` command for the release version being installed using the credentials a supplied by MDS Global:

     where:

     <username> is the username in credentials supplied by MDS Global

     <cmp version> is the CMP version to be installed, for example, 8.10 or 8.12.

  2. Unzip the downloaded package. For example:

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

Important: The latest Ansible Playbook for the release being installed must always be downloaded prior to running the installation process. It is critical that the playbook version corresponds to the version of CMP being installed.

5.0 Installation Tasks

To install CMP, you must:

1. Prepare the Inventory File using the Inventory Configuration Tool
2. Deploy CMP
3. Upgrade an Existing Installation

Important

The Inventory File used for CMP installation must be generated from the release specific version of the online Inventory Configuration Tool. MDS Global does not support installations using manually created or manually modified Inventory Files due to the complications that this causes in trying to identify any syntax or other issues with the content.

5. 1 Installation Configuration Tool

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 only required for advanced use of the installer and can be ignored unless MDS Global provides specific instruction to the contrary.

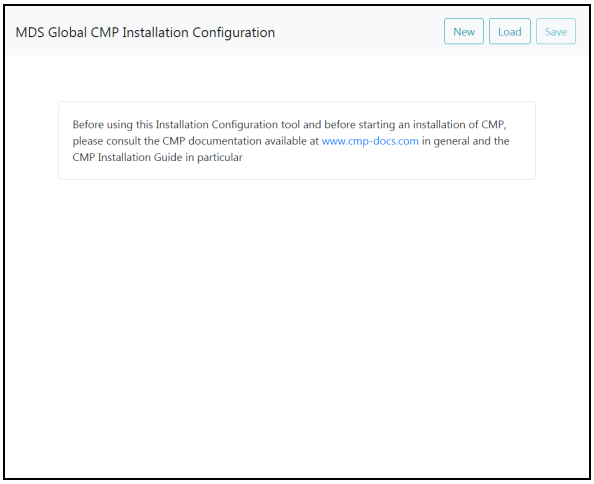
5. 1. 1 Preparing an Inventory File with the Installation Configuration Tool

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

1. Access the tool via the URL: https://inventory-config.cmp-docs.com.

   The tool is password protected. Use the same cmp-docs.com Google account used to access the CMP Documentation set.

   The tool opens in the browser.



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

   The **New Inventory** dialog opens.

3. Select the CMP release version for which to create an inventory file from the Inventory Version drop-down list, and then select *Create*.

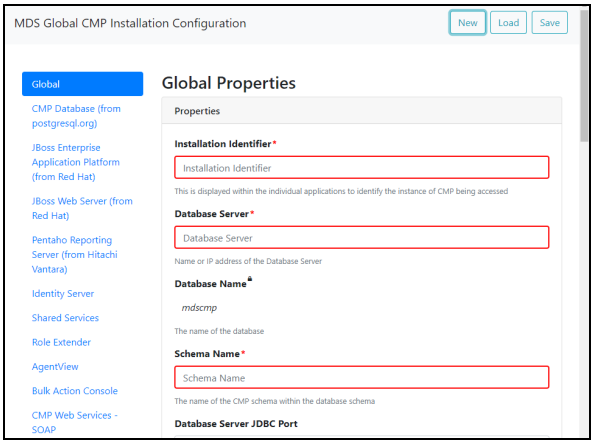
   The MDS Global CMP Installation Configuration tool opens. On the left is a navigation pane with links to configure each of the components of CMP, for example **CMP Database**, **JBoss Enterprise** or **AgentView**. The list of components reflects the table of CMP Groups.

   On the right are the properties, for which you can enter values.

4. Enter the host name on which the selected CMP component will be installed (selected from the Software drop-down list to the right of the Host field).

   You must specify at least one host per CMP component.

5. To remove a host, click the corresponding **Delete** icon.
6. Select Global from the left-hand menu to update **Global Properties**.



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

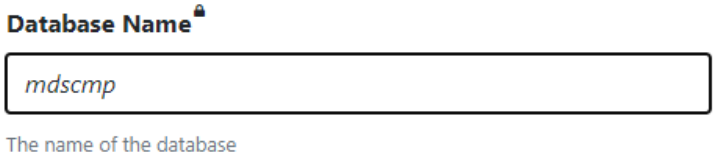
The colours of the fields are also a guide:

Mandatory property fields are highlighted in red:

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

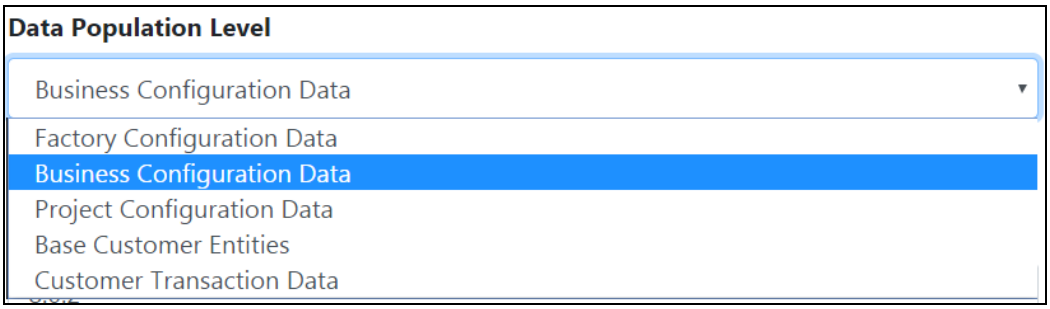
A property field for which you have entered a value is highlighted in green:

A hardcoded value is highlighted in black:

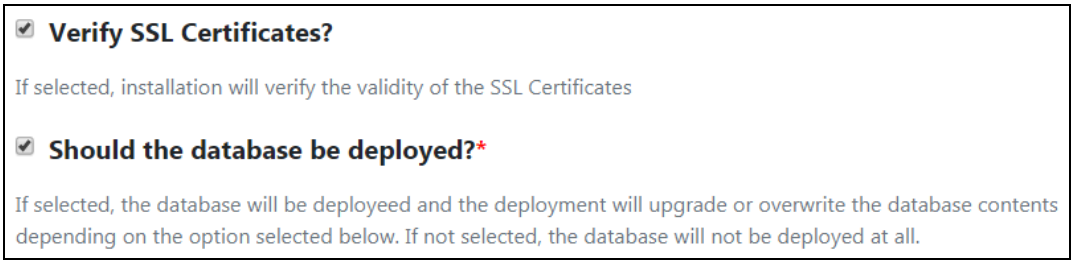


7. Enter the property values for the component.
   Enter a value for a property as follows:
   - For a text value, such as a hostname, database name or URL, enter the text in the field.

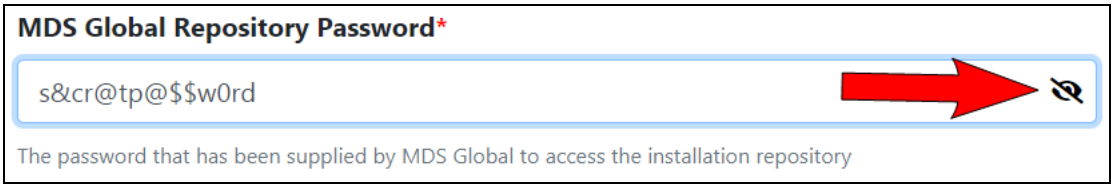
- To choose from a set of predefined (enum) values, select from a drop-down list.



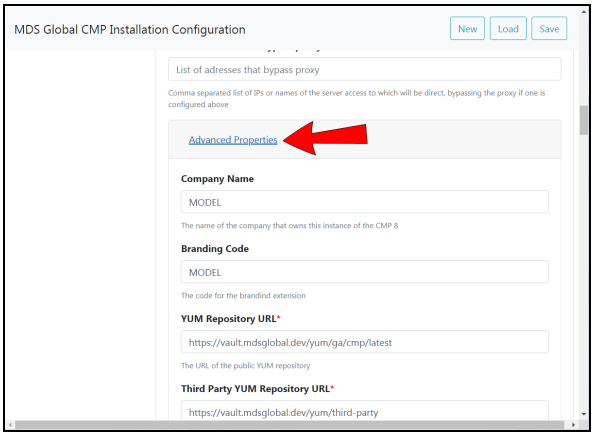
- To set a property to true/false or enable/disable, select or deselect a check-box.



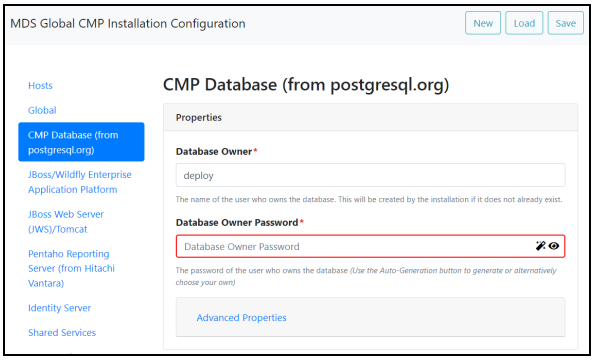
- When entering a masked property, such as a password, you can toggle the mask/unmask icon to reveal and check your entry.



8. 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.

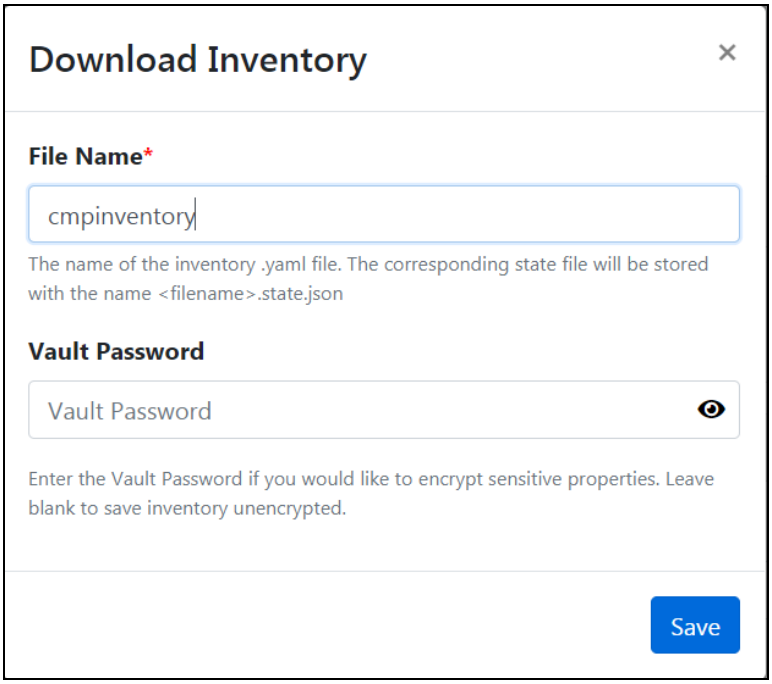


9. Once you have completed the **Global Properties** screen, click a component link in the pane of the left, for example **CMP Database**, to add the relevant properties.



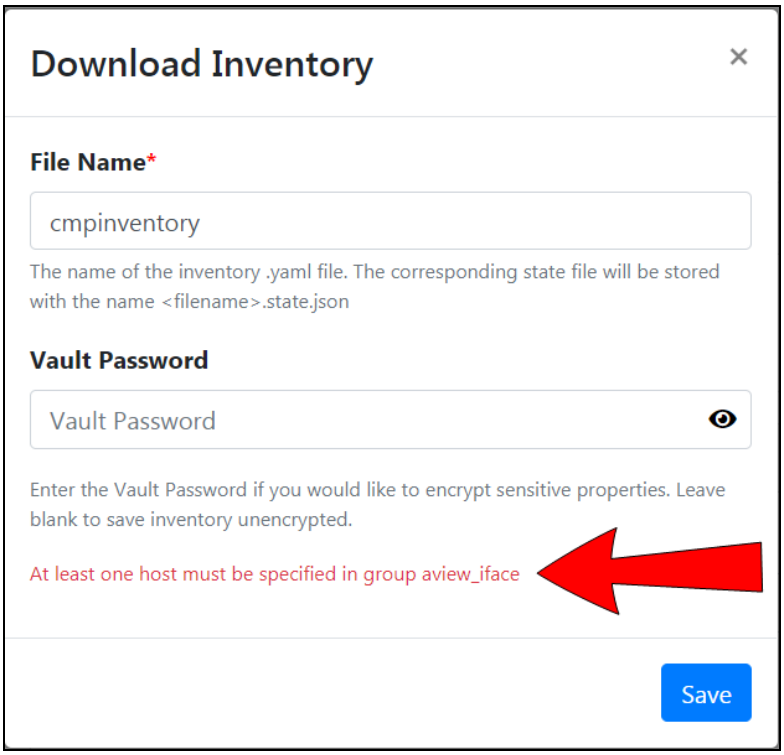
10. Repeat the steps to enter values for all CMP components until everything is configured.
11. Select AgentView from the left-hand menu. Select Advanced Properties to configure the branding. The standard CMP font and branding is deployed by default but this can be overwritten by changing the default font, colour, background images, and so on, and/or by uploading customer specific branding.
12. Click **Save**.

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

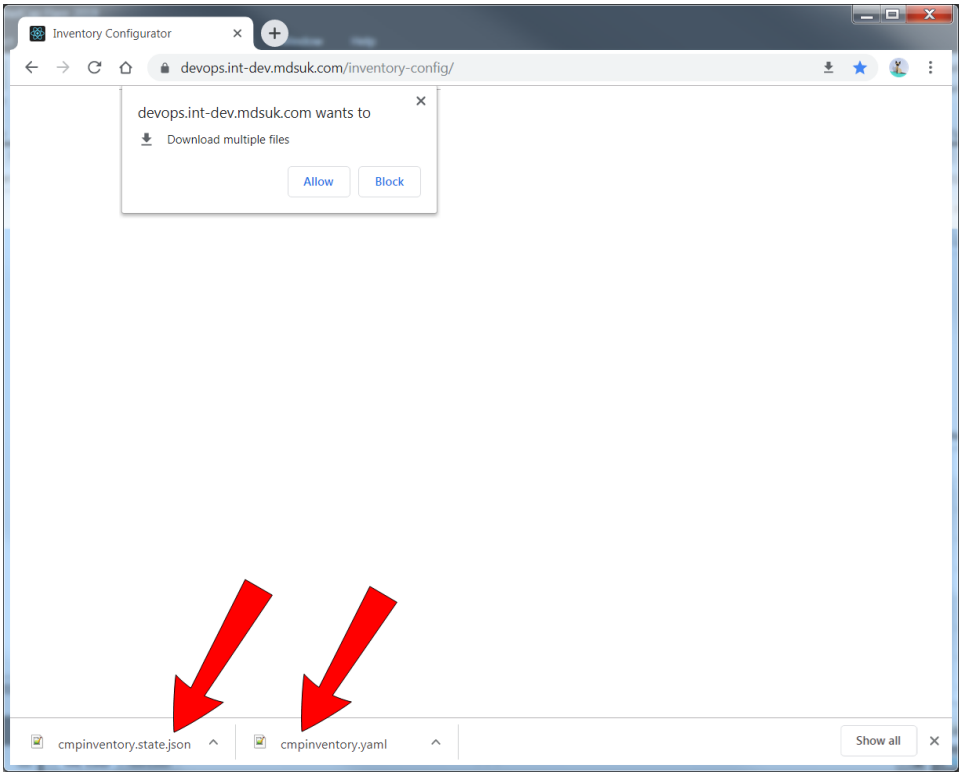


    The inventory file will be saved as `<filename>.yaml`. The corresponding environment state file will be saved as `<filename>.state.json`.
14. To encrypt properties in the file, enter the vault file password.

15. Click **Save**.
    The file is validated. An error message will alert you to any issues.



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



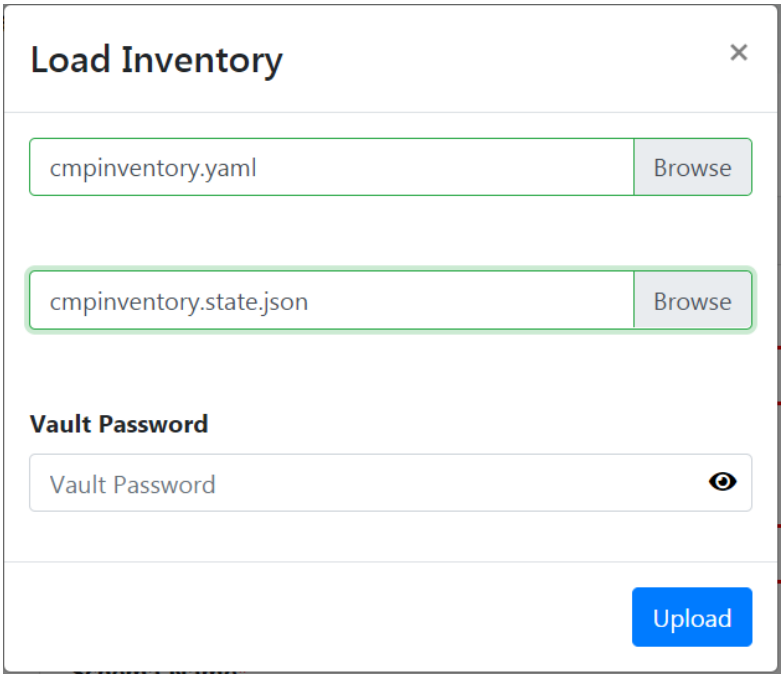
17. The final step is to copy the inventory file to the Ansible control server and use it as part of the installation process, following the instructions in "Deploy CMP" on page 37.

5. 1. 2 Working with an Existing Inventory File

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

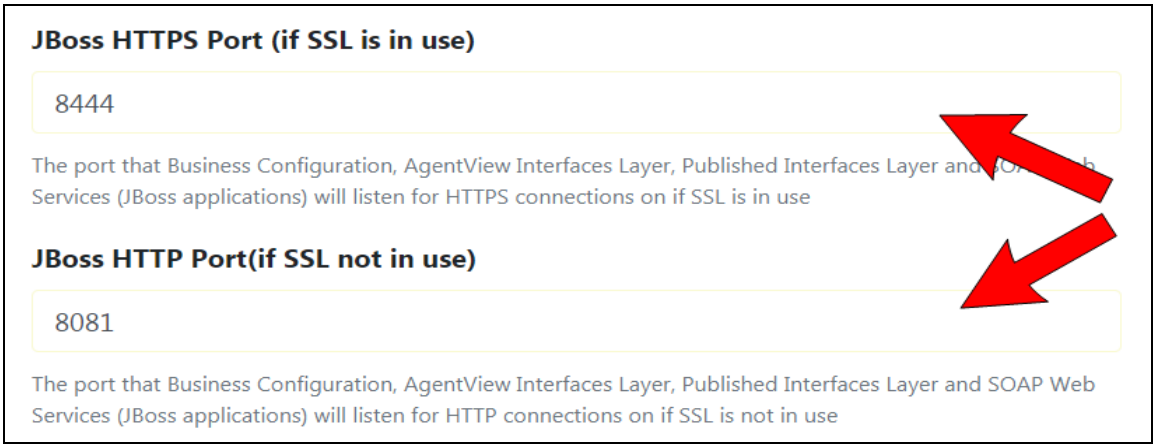
1. Open the Installation Configuration Tool and click **Load**.

   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 29
8. Save your changes.

5. 2 Deploy CMP

To deploy the CMP stack to the target system, run the following command from the directory that the Ansible Playbook has unzipped from:

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

where <inventory file> is the file that has been generated by the Installation Configuration Tool.

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

5. 2. 1 Common Ansible Command Line Parameters

The `ansible-playbook` is quite flexible, allowing users to 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
-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.

Important: Execution of lower level playbooks individually rather than using the orchestration in `deploy.yaml` can result in system misconfiguration if not done correctly. Zero downtime upgrade of a High Availability installation is also not possible by executing playbooks individually.Therefore only the use of `deploy.yaml` is supported for production environments unless MDS Global gives explicit instruction to execute a specific lower level playbook.

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 the postgres database
deploy-wso2is.yaml	deploy-db.yaml	Deploys the WSO2 Identity Server
deploy-role-extender.yaml	deploy-db.yaml deploy-wso2is.yaml	Deploys the Role Extender Application
deploy-health-check.yaml	deploy-db.yaml	Deploys the Health Check Service
deploy-sam-services.yaml	deploy-db.yaml	Deploys the SAM Services
deploy-voucher-management-services.yaml	deploy-db.yaml	Deploys the Voucher Management Services
deploy-shared-services.yaml	deploy-db.yaml	Deploys the Shared Services
deploy-jboss.yaml contains deployment roles for: agent-view-interfaces-layer pil configuration-centre soap-ws	deploy-db.yaml deploy-wso2is.yaml deploy-role-extender.yaml	Deploys the JBoss/Wildfly Application Server along with AgentView Interfaces Layer, Published Interfaces Layer, Configuration Centre and SOAP Web Services
deploy-jws.yaml contains deployments roles for: wso2is-login agent-view	deploy-db.yaml deploy-wso2is.yaml deploy-role-extender.yaml deploy-jboss.yaml	Deploys the JBoss/Tomcat Web Server along with WSO2 Identity Server Custom Login

Playbook Name	Depends On	Description
		application for SSO and AgentView
deploy-rest-ws.yaml	deploy-db.yaml deploy-wso2is.yaml deploy-role-extender.yaml	Deploy REST Web Services
deploy-bulk-action-console.yaml	deploy-db.yaml deploy-wso2is.yaml deploy-role-extender.yaml	Deploys the Bulk Action Console Application
deploy-context-sensitive-help.yaml		Deploys the Context Sensitive Help Application
deploy-pentaho-server.yaml	deploy-db.yaml deploy-wso2is.yaml deploy-role-extender deploy-jws.yaml	Deploys the Pentaho Reporting Server
deploy-artemis.yaml		Deploys the Artemis / ActiveMQ Application
deploy-sparc-engine.yaml	deploy-db.yaml deploy-artemis.yaml	Deploys the SPaRC Engine Application
deploy-sabre-server.yaml	deploy-db.yaml deploy-wso2is.yaml deploy-role-extender.yaml deploy-artemis.yaml	Deploy the Sabre Server Application
deploy-sabre-console.yaml	deploy-db.yaml deploy-wso2is.yaml deploy-role-extender.yaml deploy-artemis.yaml deploy-sabre-server.yaml	Deploys the Administration Console Application

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

5. 2. 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`.

5.2.2.1 Example Summary File

```
The CMP System is successfully installed at Version: 8.12.0


Agent View
==========
Note: Below is the specified load balancer URL and this will only
work when the Load Balancer is correctly cofigured
URL: https://cmp.demo.mdsglobal.dev/agent-view


Host: cmp-int.demo.mdsglobal.dev
Logs:
Path: /var/log/webswing/ (Webswing)
/var/log/jws5      (Tomcat)


Bulk Action Console
===================
Note: Below is the specified load balancer URL and this will only
work when the Load Balancer is correctly cofigured
URL: https://bulk.cmp.demo.mdsglobal.dev/


Host: cmp-int.demo.mdsglobal.dev
Logs:
```

Path: /var/log/bulk-action-console/

Sabre Console

==========

Note: Below is the specified load balancer URL and this will only work when the Load Balancer is correctly cofigured

URL: https://admin.cmp.demo.mdsglobal.dev/

Host: cmp-int.demo.mdsglobal.dev

Logs:

Path: /var/log/sabre-console/

Sabre Server

==========

URL: https://cmp-int.demo.mdsglobal.dev:21212/

Host: cmp-int.demo.mdsglobal.dev

Logs:

Path: /var/log/sabre-server/

JBoss

======

Host: cmp-int.demo.mdsglobal.dev

Logs:

Path: /var/log/eap7/server.log

Configuration Centre

==========

Note: Below is the specified load balancer URL and this will only work when the Load Balancer is correctly cofigured

URL: https://soap.cmp.demo.mdsglobal.dev/config

SOAP WS

==========

Note: Thi is only the root of the URL and needs to be combined with relvant endpoints to make API calls

Note: Below is the specified load balancer URL and this will only work when the Load Balancer is correctly cofigured

URL: https://soap.cmp.demo.mdsglobal.dev/

REST WS

==========

Note: Thi is only the root of the URL and needs to be combined with relvant endpoints to make API calls

Note: Below is the specified load balancer URL and this will only work when the Load Balancer is correctly cofigured

URL: https://rest.cmp.demo.mdsglobal.dev/

Host: cmp-int.demo.mdsglobal.dev

Logs:

Path: /var/log/rest-ws/boot.log

Pentaho Server

==========

URL: https://cmp-int.demo.mdsglobal.dev:7443/pentaho

Host: cmp-int.demo.mdsglobal.dev

Logs:

Path: /var/log/jws5/pentaho.log

WSO2 IS

=========

Note: Below is the specified load balancer URL and this will only work when the Load Balancer is correctly configured

```
URL: https://login.cmp.demo.mdsglobal.dev:443
Host: cmp-int.demo.mdsglobal.dev
Logs:
Path: /var/log/wso2is/wso2carbon.log

Artemis Console
=========
URL: https://cmp-int.demo.mdsglobal.dev:8161
Host: cmp-int.demo.mdsglobal.dev
Logs:
Path: /var/log/artemis/artemis.log

Artemis Broker
=========
URL: ssl://cmp-int.demo.mdsglobal.dev:61616
Host: cmp-int.demo.mdsglobal.dev
Logs:
Path: /var/log/artemis/artemis.log
```

6.0 Upgrading an Existing Installation

Prerequisites:

- Check in the release notes of the new CMP version that an upgrade path from the currently installed version exists. If there is no direct upgrade path then it will be necessary to perform a series of upgrades following defined upgrade paths, for example 8.10 - 8.12 is not supported so an 8.10 - 8.11 upgrade would need to be performed, followed by an 8.11 - 8.12 upgrade.
- If the release notes for the target release state that PostgreSQL or Linux major versions need to be upgraded, then complete those upgrades prior to starting the installation.
- Ensure that the Ansible version on the control server is aligned to the requirements of the target CMP release.
- Complete the pre-installation tasks as described in the release note of the CMP version to be upgraded to.
- 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.
- Update the original inventory file used for the last installation or upgrade of the target system using the Installation Configuration tool online at https://inventory-config.cmp-docs.com:

  1. Open the Installation Configuration Tool and select the CMP version to be upgraded to

  2. Click on load and select the original inventory file used for the last installation or upgrade of the target system

  3. Acknowledge the warning that the CMP version to be installed by the inventory file is being updated

  4. Make any changes required to the properties to be used for the installation paying particular attention to update values for new mandatory properties added since the previous release version

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

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

The `playbooks/deploy.yaml` file 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 (except where adding additional hosts). 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.

6. 1 Certificate Updates

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. 2 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 (although unless explicitly stated in the patch release

note, a patch will not require any change to the inventory file that was used for the last 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 --ask-vault-password-file=<vault_password_file>
-i <inventory_file> -b playbooks/deploy.yaml -e patch_update=true
--tags installation
```

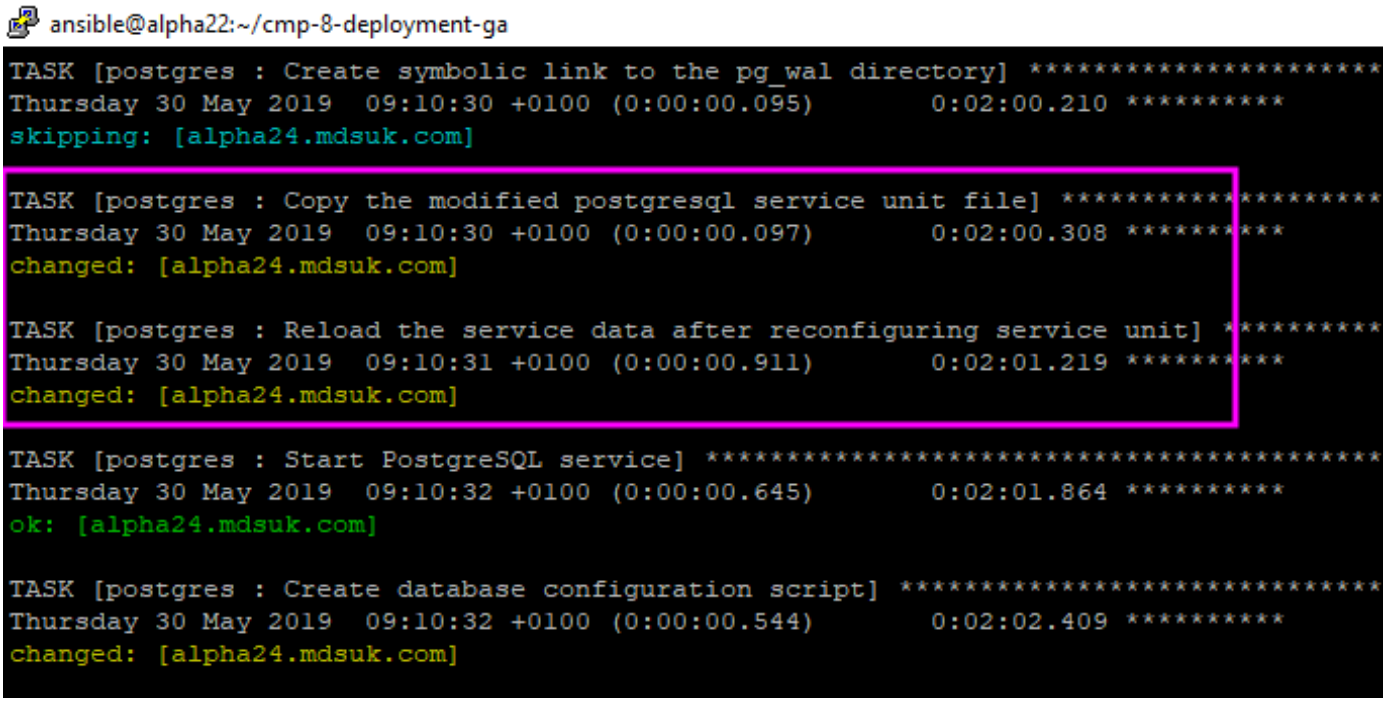
7.0 Troubleshooting

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:

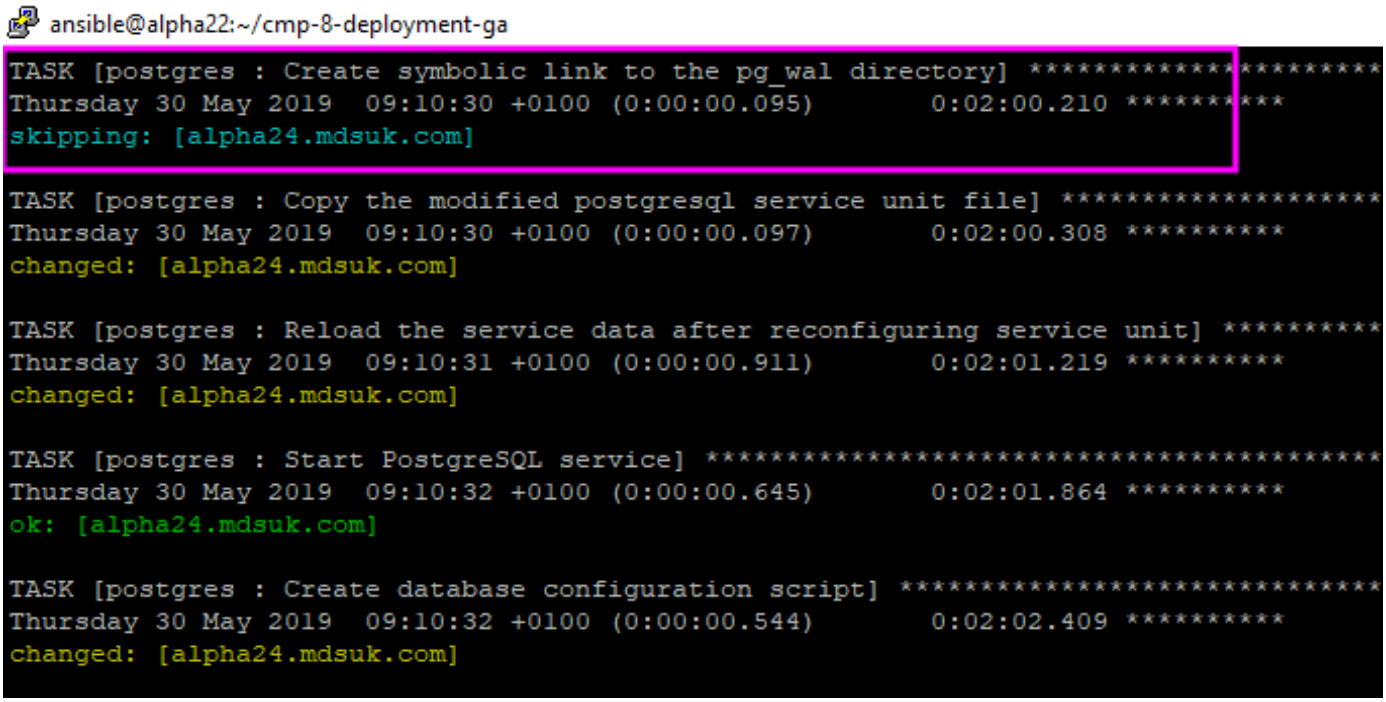
- Yellow

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



- Blue

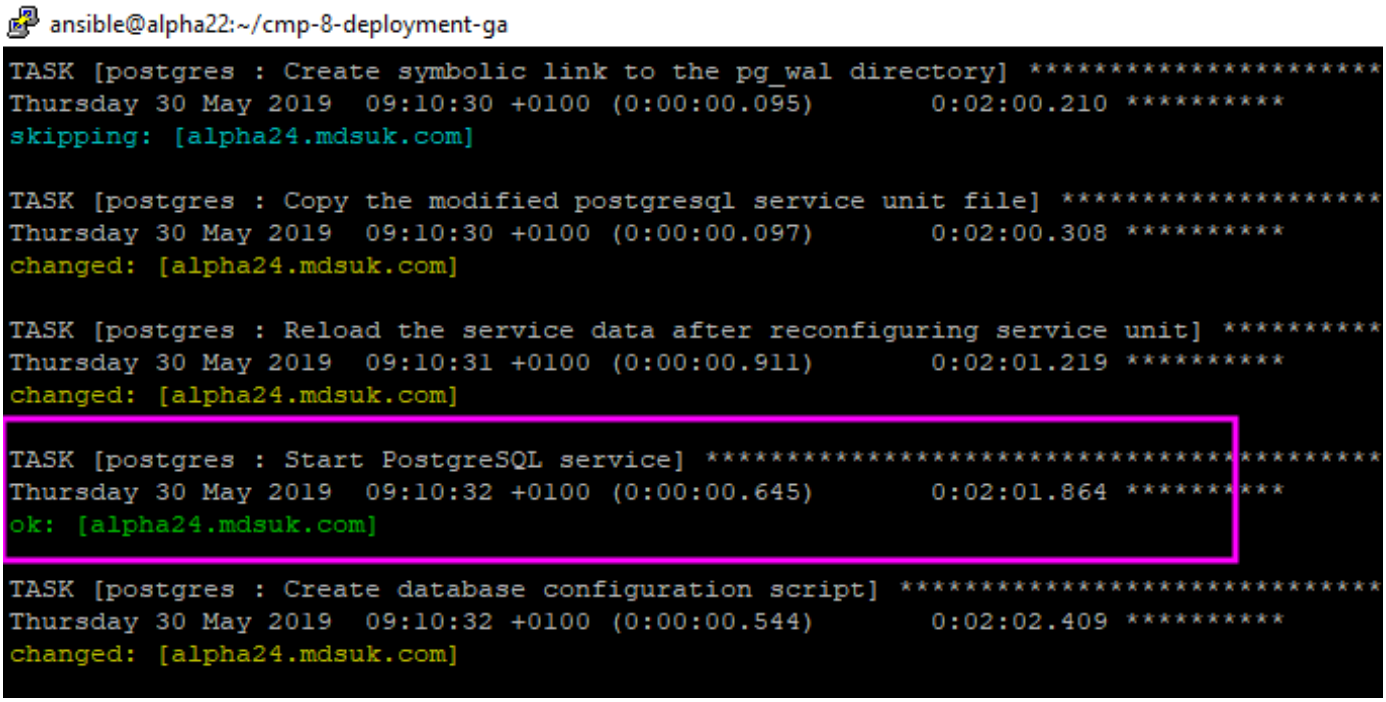
  Blue text indicates that the task was not executed because the condition attached to the file was false. For example:



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.

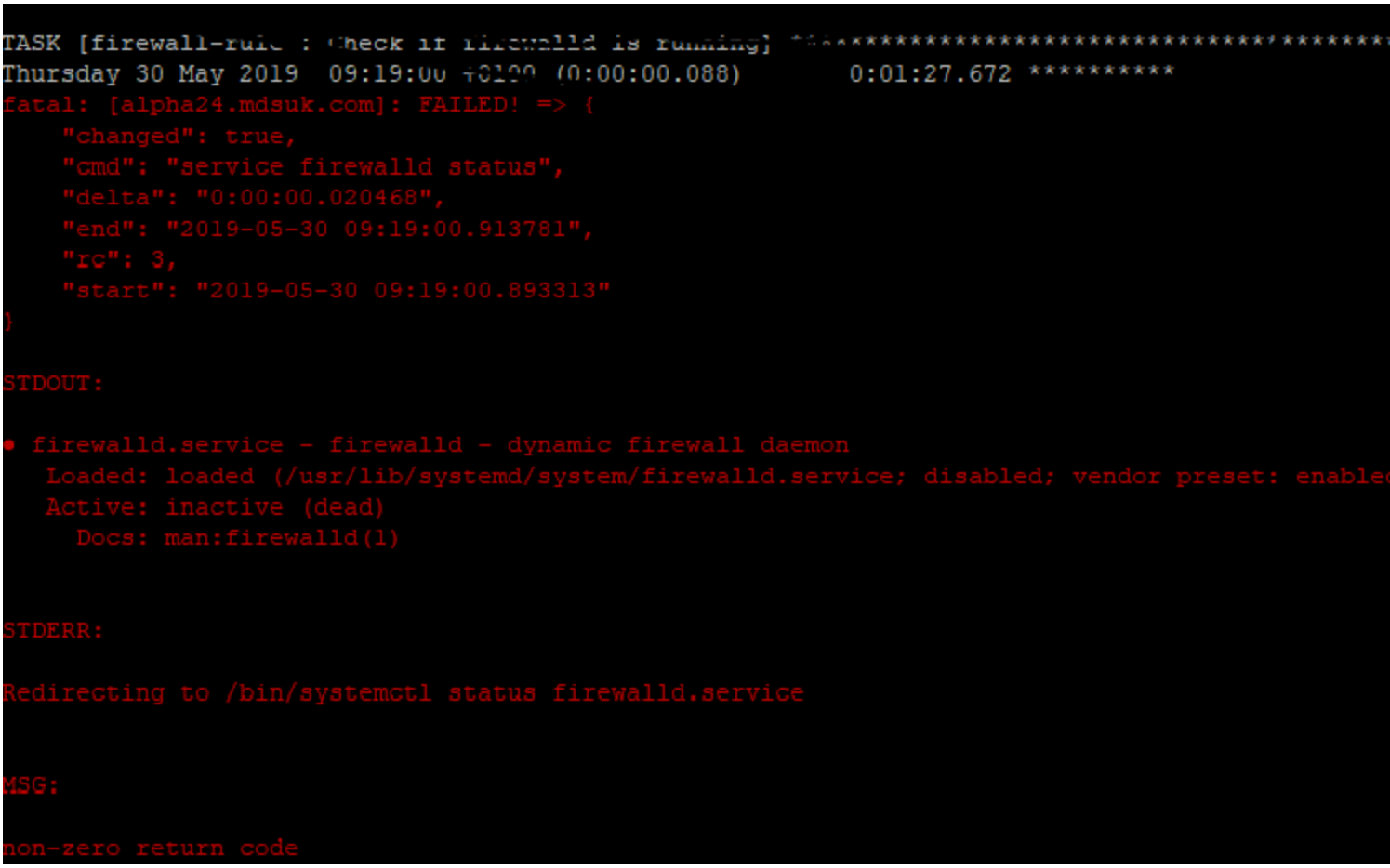
- Green

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



- Red

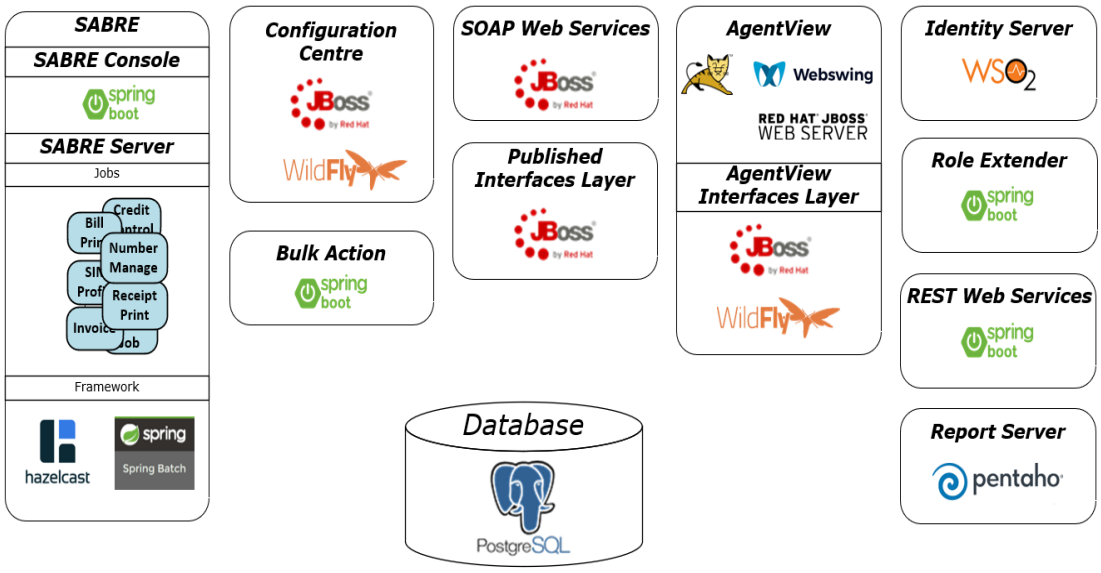
  Red text indicates that there was an error. For example:



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 41

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-13
WSO2 Identity Server	wso2is
Role Extender	role-extender
JBoss / Wildfly	eap7-standalone
JWS / Tomcat	jws5-tomcat
Health Check Service	health-check
SAM Services	sam-services
Voucher Management Services	voucher-management-services
Shared Services	shared-services
REST Web Services	rest-ws
Bulk Action Console	bulk-action-console
Context Sensitive Help	context-sensitive-help
Artemis / ActiveMQ	artemis
SPaRC Engine	sparc-engine
SABRE Server	sabre-server
CMP Administration Console	sabre-console

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

High Availability deployments rely upon Load Balancing, PostgreSQL replication and shared storage to be correctly configured before executing the installation process. These steps are not covered in this guide and will vary depending on the exact server landscape being deployed to.

Please consult MDS Global for assistance with planning and executing a High Availability deployment

9.0 Uninstall CMP

To uninstall the CMP stack run the following command:

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