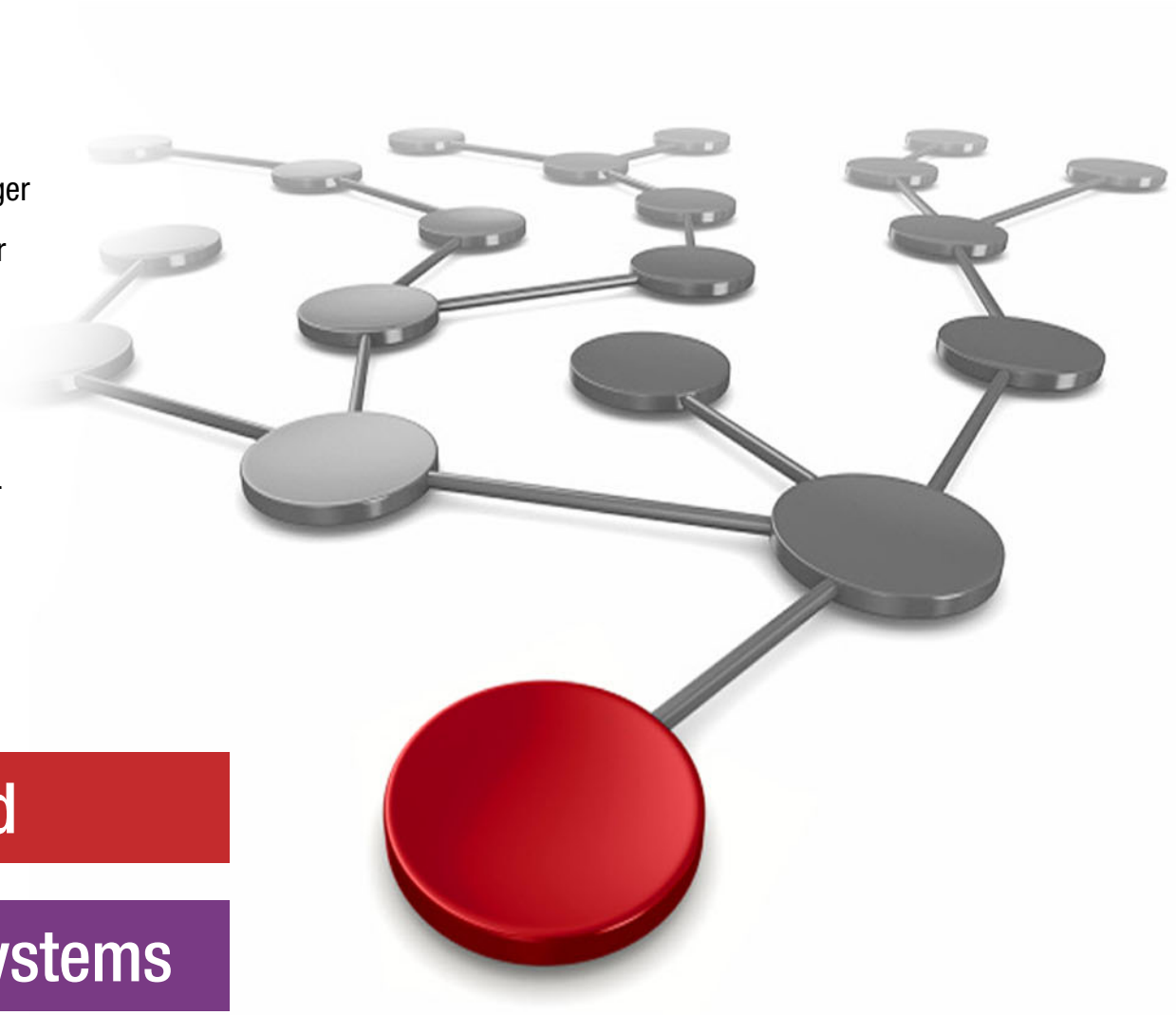
Redbooks
ibm.com/redbooks

Deploying SAP Software in Red Hat OpenShift on IBM Power Systems

Dino Quintero
Anastasiia Biliak
Christoph Gremminger
Thorsten Hesemeyer
Sabine Jaeschke
Sahitya K Jain
Jochen Röhrig
Andreas Schauberer



Cloud

Power Systems

IBM

Redpaper

IBM Redbooks

Deploying SAP Software in Red Hat OpenShift on IBM Power Systems

April 2021

REDP-5619-00

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (April 2021)

This edition applies to:

SAP HANA Platform Edition 2.0 SPS04 or higher
SAP S/4HANA 1909 or higher
SAP NetWeaver 7.5 or higher
Red Hat OpenShift Container Platform 4.5 or higher
Red Hat Enterprise Linux V8 or higher

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AIX®	IBM Watson®	POWER9™
Db2®	IBM Z®	PowerVM®
IBM®	POWER®	Redbooks®
IBM Garage™	POWER8®	Redbooks (logo)  ®

The following terms are trademarks of other companies:

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Ansible, OpenShift, Red Hat, RHCE, are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redpaper publication documents how to containerize and deploy SAP software into Red Hat OpenShift 4 Kubernetes clusters on IBM Power Systems by using predefined Red Hat Ansible scripts, different configurations, and theoretical knowledge, and it documents the findings through sample scenarios.

This paper documents the following topics:

- Running SAP S/4HANA, SAP HANA, and SAP NetWeaver on-premises software in containers that are deployed in Red Hat OpenShift 4 on IBM Power Systems hardware.
- Existing SAP systems running on IBM Power Systems can be repackaged at customer sites into containers that use predefined Red Hat Ansible scripts.
- These containers can be deployed multiple times into Red Hat OpenShift 4 Kubernetes clusters on IBM Power Systems.

The target audiences for this paper are Chief Information Officers (CIOs) that are interested in containerized solutions of SAP Enterprise Resource Planning (ERP) systems, developers that need containerized environments, and system administrators that provide and manage the infrastructure with underpinning automation.

This paper complements the documentation that is available at IBM Knowledge Center, and it aligns with the educational materials that are provided by IBM Garage™ for Systems Education.

Authors

This paper was produced in close collaboration with the IBM SAP International Competence Center (ISICC) in Walldorf, SAP Headquarters in Germany, and IBM Redbooks®.

Dino Quintero is an IT Management Consultant and an IBM Level 3 Senior Certified IT Specialist with IBM Redbooks in Poughkeepsie, New York. He has 24 years of experience with IBM Power Systems technologies and solutions. Dino shares his technical computing passion and expertise by leading teams developing technical content in the areas of enterprise continuous availability, enterprise systems management, high-performance computing, cloud computing, artificial intelligence (AI) (including machine and deep learning), and cognitive solutions. He also is a Certified Open Group Distinguished IT Specialist. Dino holds a Master of Computing Information Systems degree and a Bachelor of Science degree in Computer Science from Marist College.

Anastasiia Biliak is a software developer for SAP on Power Systems who joined IBM in 2019. She has 4 years of experience as a software developer in various industries. She holds a Bachelor of Science degree in Computer Science from Hochschule Niederrhein, University of Applied Sciences. Anastasiia has experience in developing back-end APIs, and has experience in front-end object-oriented design and analysis.

Christoph Gremminger is a Project Manager for the SAP on Power Systems Development Team in St. Leon-Rot, Germany. He has 23 years of experience with IBM, and uses cross-functional knowledge from various job roles to run successful projects.

Thorsten Hesemeyer is an IT Specialist working for Technical Field Enablement for SAP on Power Systems in St. Leon-Rot, Germany. He is an LPIC-3 certified Linux expert with 30 years of onsite customer experience. His main areas of expertise are data center migrations, server virtualization, and container orchestration with Red Hat products for many IBM customers. Thorsten holds a Diploma in Physics degree from Ruhr-University Bochum.

Sabine Jaeschke is a software developer for SAP on IBM Z® Development in Germany. She has 15 years of experience in adjusting SAP Software Provisioning Manager for specific IBM Db2® on Z customer needs. She has worked at IBM for more than 33 years. Her areas of expertise include container image building, databases, and SAP systems. She has written extensively on building and deploying container images.

Sahitya K. Jain is an Advisory Software Engineer who works for SAP platform support with IBM System Labs. He has over 13 years of experience in working with Power Systems servers. He has worked on functional verification testing for Virtual I/O Server (VIOS) and IBM AIX®. In his current role, he supports Power Systems customers running SAP applications, such as SAP NetWeaver or SAP HANA. Sahitya holds a Bachelor of Engineering (Computer Science) degree from Visvesvaraya Technological University, Belagavi, India.

Jochen Röhrig is a Senior Software Engineer with the joint IBM/SAP platform team for SAP on Power Systems at SAP in Walldorf, Germany. Having worked on enabling SAP software on traditional IBM systems in the past, he is focusing on emerging topics like running SAP systems on Red Hat OpenShift, using IBM Watson® services in Advanced Business Application Programming (ABAP), or connecting SAP systems to IBM Blockchain. Having worked for IBM for 20+ years, Jochen has 20+ years of experience in Linux and 16+ years of experience in SAP on IBM platforms. He holds a German and a French master's degree in computer science, and a Ph.D. in computer science from the Saarland University, Saabrücken, Germany. He is a Red Hat Certified Engineer (RHCE, 2004) and holds certificates LPIC-1 (2006) and LPIC-2 (2008) from the Linux Professional Institute. His areas of expertise include emerging technologies like cloud computing, containerization, and AI and blockchain, and traditional topics like software development, open source software, operating systems, parallel computing, and SAP on IBM platforms.

Andreas Schauberer is a Senior Software Engineer working for the IBM Systems Lab in Germany. He has 15 years of experience with the IBM POWER® platform, and with AIX and Linux on Power Systems for SAP applications. In earlier years, he worked in different software engineering roles on IBM high availability (HA) software for SAP applications. In his current role, he leads the IBM development team that is responsible for SAP NetWeaver and S/4HANA Foundation on the IBM PowerLinux platform. Andreas holds a German degree of “Diplom Informatiker (FH)” from Fachhochschule Giessen.

Thanks to the following people for their contributions to this project:

Wade Wallace
IBM Redbooks, Austin Center

Wolfgang Reichert, IBM Distinguished Engineer, CTO for SAP on IBM Systems
IBM Germany

Chongshi Zhang, Software Engineer, Red Hat OpenShift on IBM Power Systems
IBM Austin

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an IBM Redbooks residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

`ibm.com/redbooks/residencies.html`

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  `ibm.com/redbooks`

- Send your comments in an email to:

  `redbooks@us.ibm.com`

- Mail your comments to:

  IBM Corporation, IBM Redbooks
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Introduction

This chapter provides an overview of the scope of this paper.

This chapter contains the following topics:

1.1 Introduction

This paper provides a summary of a feasibility study that was run by IBM with the support of the Red Hat SAP team and the SAP LinuxLab team. The solution takes advantage of the enterprise class SAP S/4HANA intelligent Enterprise Resource Planning (ERP) system, reliable and secure IBM Power Systems technology, and the enterprise-grade container platform Red Hat OpenShift. Industry automation standards orchestrate the components end-to-end, and in single-step workflows. The target audiences are Chief Information Officers (CIOs) that are interested in containerized solutions of SAP ERP systems, developers with the need for containerized environments, and system administrators providing and managing the infrastructure with underpinning automation.

The first edition of this paper focused on functions and was targeted at test and non-production use only. The solution uses dedicated software product versions, basic configuration options such as SAP - Systems as Standard System (Primary Application Server (PAS) + Advanced Business Application Programming (ABAP) SAP Central Services (ASCS) + SAP HANA Database), related system resources, NFS storage attachment, and a Red Hat OpenShift cluster minimum configuration. Non-functional characteristics such as high availability (HA), vertical and horizontal scaling, and using alternative storage concepts can be the scope of future extensions.

This paper explains concepts, all the components that are used (Figure 1-1), and the structure of the solution. The paper provides usage guidance for the accompanying open source automation scripts.

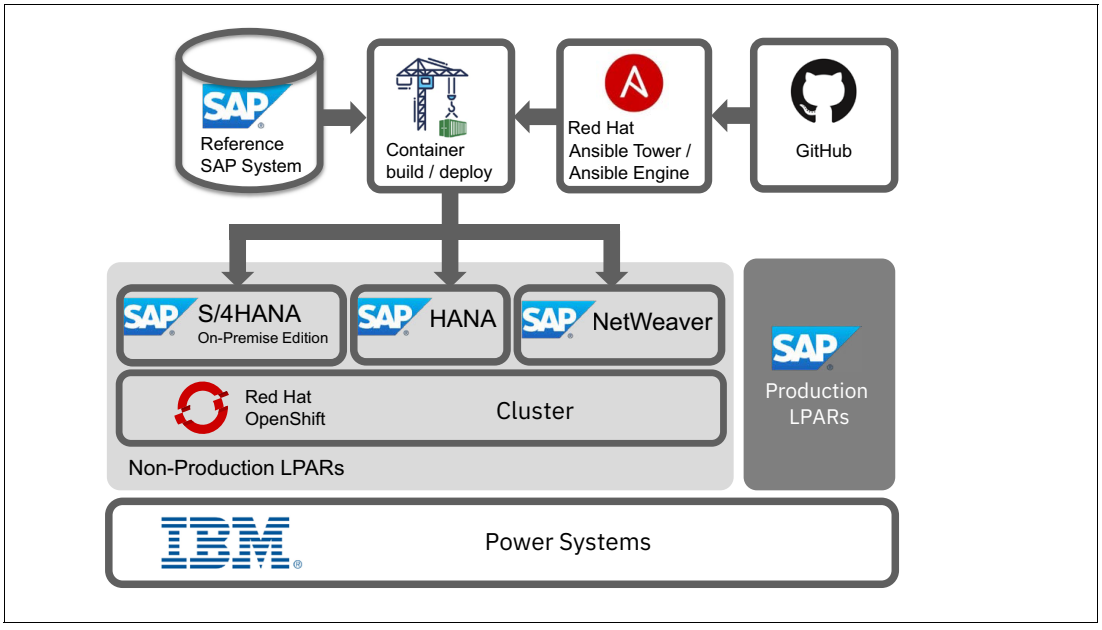


Figure 1-1 Solution component overview

1.2 Use cases and value proposition

The following use cases are supported by the current solution.

Note: In this current state, SAP on IBM Power Systems with Red Hat OpenShift covers a feasibility study, and it targets test and other non-production landscapes. The created deliverables are not supported by SAP or an agreed-to road map for official support in its current state (for more information, see SAP Note 1122387 - *Linux: SAP Support in virtualized environments*).

- Explore and run an SAP standard configuration that consists of SAP HANA, S/4HANA, or SAP NetWeaver on-premises editions for container deployment.
- Shift and migrate an on-premises SAP standard configuration to Red Hat OpenShift Container Platform automatically within the IBM Power platform.
- Rapid provisioning of SAP HANA, S/4HANA, or SAP NetWeaver test and non-production container instances.
- GUI and command-line interface (CLI) automation options allow for end-to-end automation and individual step executions.
- Co-existence with SAP production systems, for example, on IBM Power Systems logical partitions (LPARs).

Based on the implementation, the solution offers the following advantages:

- A virtualization alternative to hypervisors like VMware or Kernel-based Virtual Machine (KVM) based on the emerging market for container concepts.
- Extended resource options that are delivered by the Red Hat OpenShift layer on IBM PowerVM® LPARs.
- SAP HANA, S/4HANA, or SAP NetWeaver on-premises editions for Red Hat OpenShift Container Platform.
- Red Hat OpenShift as an enterprise version of open source Kubernetes.
- Expert knowledge that is encapsulated and combined in automation scripts.
- Running a container instance within seconds based on the overlay file system on the NFS server (write-on-change concept).
- Open-source nature that allows for immediate use and community contributions.
- Enterprise class ecosystem combining the strength of IBM Power Systems, Red Hat OpenShift Container Platform, SAP Business Suite Products, and industry standards for automation.

1.3 Solution design overview

Understanding the solution design requires you to learn about various aspects to accomplish optimal concept mappings from an on-premises instance to a container instance, such as inter-communication and operations. The design has the following characteristics:

- SAP system mapping into a container image (Service Distribution):
  - Two types of containers: one for the SAP HANA database, and that is composed of the ASCS and the PAS (the dialog instance (DI). Depending on the start parameters, ASCS or PAS are instanced at run time.
  - Persistent data is stored in a centrally accessible NFS share, which is outside of your Red Hat OpenShift cluster.
- Red Hat OpenShift feature mappings (Service Operation - lifecycle management):
  - GitHub, Build Server, and Red Hat Ansible Tower are infrastructure services that you use to automatically create and deploy the container images to Red Hat OpenShift Image Registry.
  - Container instances are created from Red Hat OpenShift Image Registry. To keep this example simple, we use the *all in one* runtime approach, which means that all container instances belonging to one SAP System are started automatically in a single Kubernetes pod.
  - Stopping and restarting container instances is managed by Red Hat OpenShift standard features.
- Component interaction model at run time (Service Interaction):
  - Inter-container instance communication and a Container-NFS share data exchange are based on TCP/IP.
  - User access from the outside world is provided by SSH forwarding. The SAP GUI uses the helper node to access the application server in the PAS container.

Figure 1-2 shows the solution design overview.

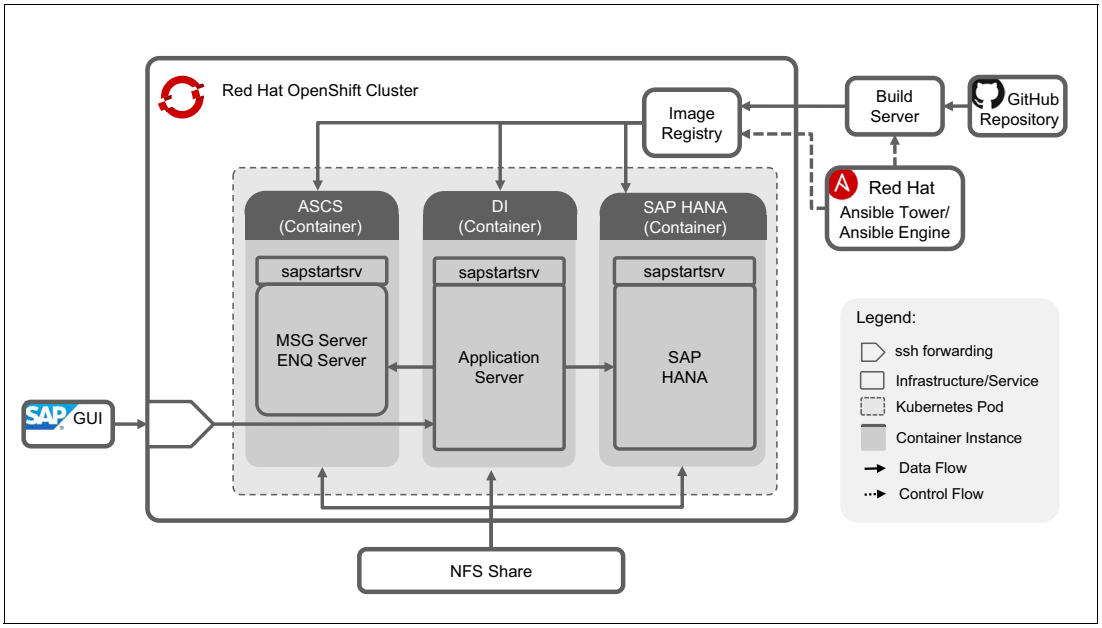


Figure 1-2 Solution design overview

1.4 Functional restrictions

In the current state of the solution, the following functional restrictions apply to the SAP system running inside a Red Hat OpenShift container:

- The SAP Host Agent is not installed in the containers, so the following SAP landscape management products cannot manage the SAP system in the container:
  - SAP Management Console
  - SAP Landscape Management
  - SAP Solution Manager
- The SAP Solution Manager Diagnostics Agent is not installed in the containers, so the SAP Solution Manager cannot manage this SAP system.
- The SAProuter is installed but not started in the ASCS container, so the SAP GUI cannot use SAP Central Services to connect to the application server. Instead, the SAP GUI must connect directly to the application server. It is possible to install the SAProuter outside of the cluster on the helper node in to route the traffic to the SAP System inside the container.
- The SAP Web dispatcher is installed but not started in the ASCS container, so the web GUI must connect directly to the application server instance in the container. It is possible to install the SAP Web dispatcher outside of the cluster on the helper node to route the traffic to the SAP System inside the container.

1.5 Paper overview

This paper use two resources to combine static and dynamic information channels effectively. There are reference links that provide more resources for readers according to their level of interest:

- This IBM Redpaper publication (static conceptual information about the solution)
- GitHub blog and repository (dynamic technical details about the solution, including automation scripts)

The following chapters in this paper reflect the logical flow of the IBM Power Systems with SAP software that is deployed in Red Hat OpenShift solution. The starting point is the infrastructure setup guidance for a Red Hat OpenShift cluster, which is followed by the on-premises SAP reference system that is converted into a containerized solution that is then deployed and operated on the established Red Hat OpenShift environment.

Note: Documented information regarding supported environments, configurations, and sizing guides are accurate at the time of writing. Because of the agile nature of Red Hat OpenShift, elements and aspects can change with subsequent Red Hat OpenShift V4 updates.

When major changes are required, a revised edition of this IBM Redpaper publication might be published. However, you should check official resources (release notes, online documentation, and so on) for any changes to what is presented in this paper.

2

Setting up the Red Hat OpenShift infrastructure

This chapter describes how to set up the Red Hat OpenShift infrastructure.

This chapter contains the following topics:

2.1 Introduction

This chapter describes the installation of Red Hat OpenShift on IBM Power Systems hardware.

2.2 Requirements for the Red Hat OpenShift cluster

This section describes the requirements for Red Hat OpenShift.

2.2.1 Software

Red Hat OpenShift Container Platform V4 is used for the SAP workload that is described in this paper. Quality assurance is performed with Red Hat OpenShift Container Platform V4.5.18. The Kubernetes release in Red Hat OpenShift is V1.18.3.

For more information, see the following resources:

- Red Hat OpenShift Container Platform
- Red Hat OpenShift Container Platform 4.6 release notes - IBM Power Systems
- Red Hat OpenShift Container Platform Lifecycle Policy

Red Hat OpenShift V4 is included with Red Hat Enterprise Linux CoreOS, which offers a fully immutable, lightweight, and container-optimized Linux operating system distribution. Only Red Hat Enterprise Linux CoreOS can be used on IBM Power Systems for all master and worker logical partitions (LPARs).

2.2.2 Hardware

Only IBM Power Systems with a PowerVM hypervisor and Little Endian support can be used for the SAP workload that is described in this paper. All IBM POWER8® and IBM POWER9™ processor-based scale-out and Enterprise models can be used.

2.3 Size nodes for SAP workloads

The LPARs must be sized to meet the minimum resource requirements that are shown in Table 2-1 before you start SAP deployments on the cluster.

Table 2-1 LPAR minimum resource requirements for SAP workloads

LPAR	Operating system	vCPU	Memory	Storage
Helper node (1 LPAR)	Red Hat Enterprise Linux 8x	4	64 GB	120 GB + 880 GB NFS share (see Figure 2-1)
Bootstrap (1 LPAR)	Red Hat Enterprise Linux CoreOS	2	32 GB	120 GB

LPAR	Operating system	vCPU	Memory	Storage
Master (3 LPARs)	Red Hat Enterprise Linux CoreOS	2	32 GB	120 GB
Worker (2 LPARs)	Red Hat Enterprise Linux CoreOS	4	256 GB	500 GB

The NFS share sizing for the helper node is based on the planned SAP HANA deployments, as shown in Figure 2-1.

$$\sum_{i=1}^{tns} hs_i + enc_i * ews_i$$

Figure 2-1 Sizing for the helper node

- tns: The total number of SAP systems for which images will be created (for example, SAP HANA data will be stored on the NFS server).
- hs_i: The SAP HANA size of the SAP system *i* at the time of image creation.
- enc_i: The expected maximum number of simultaneously running container instances of SAP system *i*.
- ews_i: The expected total write size for one container instance of SAP system *i* during the container lifetime.

2.4 Red Hat OpenShift software subscription

To install Red Hat OpenShift on IBM Power Systems, first download the `pull-secret` file from Install OpenShift on Power with user-provisioned infrastructure.

Note: The installer needs this `pull-secret` file for the installation.

Your cluster is automatically registered with a 60-day evaluation subscription that does *not* include support. To receive support for your cluster, you must edit the subscription settings in the Cluster Details page in the Red Hat OpenShift Cluster Manager.

2.5 Red Hat OpenShift setup with PowerVM

This section shows how to set up and run a PowerVM server that is managed by a Hardware Management Console (HMC).

The playbook that is described in this section sets up a helper node that has all the infrastructure and services to install Red Hat OpenShift V4. This playbook also installs a Red Hat OpenShift V4 cluster with three master nodes and two worker nodes. After you run the playbook, you are ready to log in to the Red Hat OpenShift cluster.

This chapter assumes the following items (see Figure 2-2):

- You are on a network with access to the internet.
- The network that you are on does *not* have DHCP (or you can block your existing DHCP from responding to the MAC addresses that is used for the Red Hat OpenShift LPARs).
- The helper node acts as a load balancer, DHCP, TFTP, DNS, HTTP, and NFS server for the Red Hat OpenShift cluster.

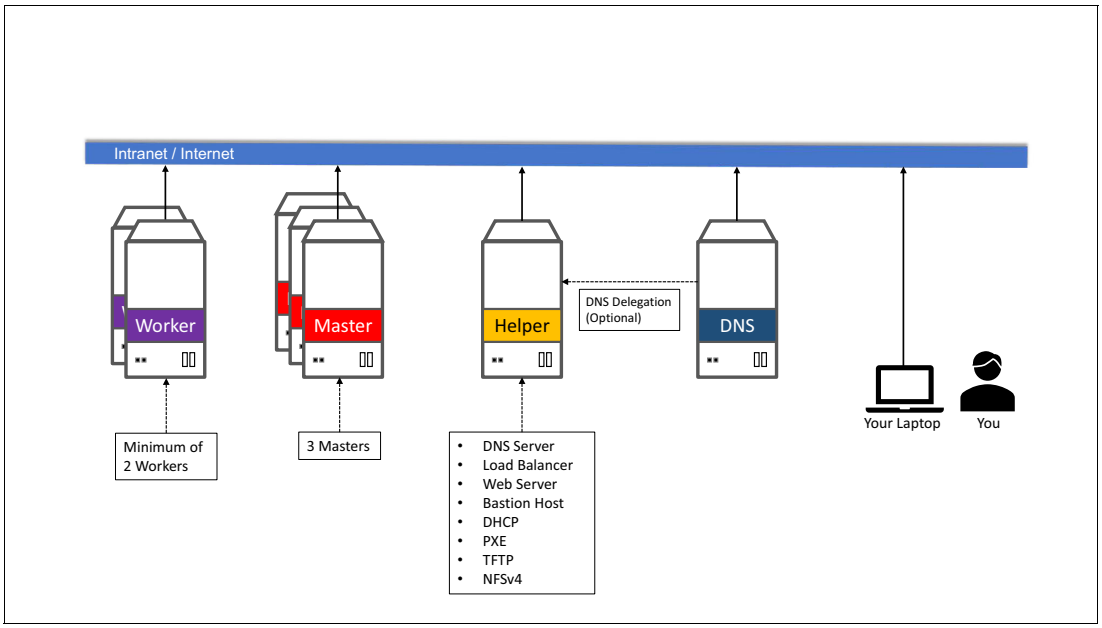


Figure 2-2 Network configuration and assumptions

You can delegate the DNS to the ocp4-helpernode if you do not want to use it as your main DNS server. You must delegate *$CLUSTERID.$DOMAIN* to this helper node.

For example, if you want a *$CLUSTERID* of `ocp4`, and you have a *$DOMAIN* of `example.com`, then you delegate `ocp4.example.com` to this ocp4-helpernode.

2.5.1 Creating the helper node (ocp4-helpernode)

To create the helper node, complete the following steps:

1. Create the helper LPAR by using the HMC GUI or the HMC **`mksyscfg`** command. If you decide to use the command, use SSH to access your HMC host and open the command-line interface (CLI). The steps in this section are specific to CLI.
2. Configure the LPAR with the following parameters:
   - Four vCPUs (`desired_procs`)
   - 64 GB of RAM (`desired_mem`)
   - 120 GB HD (OS) + 880 GB HD (NFS)

```
$ mksyscfg -r lpar -m <managed_system> -i name=ocp4-helper,
profile_name=default_profile, lpar_env=aixlinux, shared_proc_pool_util_auth=1,
min_mem=8192, desired_mem=65536, max_mem=65536, proc_mode=shared,
min_proc_units=0.2, desired_proc_units=0.4, max_proc_units=4.0, min_procs=1,
desired_procs=4, max_procs=16, sharing_mode=uncap, uncap_weight=128,
max_virtual_slots=64, boot_mode=norm, conn_monitoring=1
```

3. Attach the LPAR to the appropriate network and add storage (use the HMC GUI or the HMC **chsyscfg** command) after successfully creating the LPAR.
4. Go to Red Hat Enterprise Linux V8 and follow the instructions there to install Red Hat Enterprise Linux V8 in to the PowerVM LPAR.
5. After the helper LPAR is running, configure it with the correct network configurations based on your network:
   - IP address: `<helper_ip>`
   - Netmask: `<netmask>`
   - Default gateway: `<default_gateway>`
   - DNS server: `<default_DNS>`

2.5.2 Creating cluster nodes

This section creates six LPARs by using the HMC GUI or the HMC **mksyscfg** command.

Bootstrap

Complete the following steps:

1. Create one bootstrap LPAR with the following configuration parameters:
   - Two vCPUs (`desired_procs`).
   - 32 GB of RAM (`desired_mem`).
   - 120 GB HD (operating system).

   ```
   $ mksyscfg -r lpar -m <managed_system> -i name=ocp4-bootstrap,
   profile_name=default_profile, lpar_env=aixlinux, shared_proc_pool_util_auth=1,
   min_mem=8192, desired_mem=32768, max_mem=32768, proc_mode=shared,
   min_proc_units=0.2, desired_proc_units=0.2, max_proc_units=4.0, min_procs=1,
   desired_procs=2, max_procs=4, sharing_mode=uncap, uncap_weight=128,
   max_virtual_slots=64, boot_mode=norm, conn_monitoring=1
   ```
2. Attach the LPAR to the appropriate network and add storage (use the HMC GUI or the HMC **chsyscfg** command) after successfully creating the LPAR.
3. Go to Red Hat Enterprise Linux V8 and follow the instructions there to install Red Hat Enterprise Linux V8 in to the PowerVM LPAR.

The operating system is replaced later by the Red Hat OpenShift installer with a Red Hat Enterprise Linux CoreOS.

Master LPARs

Complete the following steps:

1. Create three master LPARs with the following configuration parameters:
   - Two vCPUs (`desired_procs`)
   - 32 GB of RAM (`desired_mem`)
   - 120 GB HD (operating system)

```
$ for i in master{0..2}
do
  mksyscfg -r lpar -m <managed_system> -i name="ocp4-${i}",
profile_name=default_profile, lpar_env=aixlinux, shared_proc_pool_util_auth=1,
min_mem=16384, desired_mem=32768, max_mem=32768, proc_mode=shared,
min_proc_units=0.2, desired_proc_units=0.2, max_proc_units=4.0, min_procs=2,
desired_procs=2, max_procs=2, sharing_mode=uncap, uncap_weight=128,
max_virtual_slots=64, boot_mode=norm, conn_monitoring=1
done
```

2. Attach the LPARs to the appropriate network and add storage (use the HMC GUI or the HMC **chsyscfg** command) after successfully creating the LPAR.
3. Go to Red Hat Enterprise Linux V8 and follow the instructions there to install Red Hat Enterprise Linux V8 in to the PowerVM LPAR.

The operating systems are replaced later by the Red Hat OpenShift installer with a Red Hat Enterprise Linux CoreOS.

Worker LPARs

Complete the following steps:

1. Create two worker LPARs with the following configuration parameters:
   - 4 vCPUs (desired_procs), more depending on the workload
   - 256 GB of RAM (desired_mem), more depending on the workload
   - 500 GB HD (OS), more depending on the workload

```
$ for i in worker{0..1}
do
  mksyscfg -r lpar -m <managed_system> -i name="ocp4-${i}",
profile_name=default_profile, lpar_env=aixlinux, shared_proc_pool_util_auth=1,
min_mem=16384, desired_mem=262144, max_mem=262144, proc_mode=shared,
min_proc_units=0.2, desired_proc_units=0.8, max_proc_units=4.0, min_procs=1,
desired_procs=4, max_procs=16, sharing_mode=uncap, uncap_weight=128,
max_virtual_slots=64, boot_mode=norm, conn_monitoring=1
done
```

2. Attach the LPARs to the appropriate network and add storage (use the HMC GUI or the HMC **chsyscfg** command) after successfully creating the LPAR.
3. Go to Red Hat Enterprise Linux V8 and follow the instructions there to install Red Hat Enterprise Linux V8 in to the PowerVM LPAR.

The operating systems are replaced later by the Red Hat OpenShift installer with a Red Hat Enterprise Linux CoreOS.

2.5.3 Obtaining the MAC address of the LPAR from the HMC

To obtain the MAC address, run the following command:

```
$ for i in <managed_systems>
do
  lshwres -m $i -r virtualio --rsubtype eth --level lpar -F lpar_name,mac_addr
done
```

If you are using single-root input/output virtualization (SR-IOV), run the following command instead:

```
$ for i in <managed_systems>
do
  lshwres -m $i -r sriov --rsubtype logport --level eth -F lpar_name,mac_addr
done
```

2.5.4 Preparing the helper node

Complete the following steps:

1. After the helper node operating system is installed, log in to it by running the following command:

   ```
   $ ssh root@<helper_ip>
   ```

   Note: For Red Hat Enterprise Linux V8, enable `rhel-8-for-ppc64le-baseos-rpms`, `rhel-8-for-ppc64le-appstream-rpms`, and `ansible-2.9-for-rhel-8-ppc64le-rpms`.

2. Perform the following software installations:

   a. Install Extra Packages for Enterprise Linux (EPEL) by running the following command:

      ```
      $ yum -y install
      https://dl.fedoraproject.org/pub/epel/epel-release-latest-$(rpm -E
      %rhel).noarch.rpm
      ```

   b. Install Ansible and Git by running the following command:

      ```
      $ yum -y install ansible git
      ```

   c. Install Firefox and X11 forwarding libs by running the following command:

      ```
      $ yum -y install firefox xorg-x11-xauth dbus-x11
      ```

2.5.5 Setting SELinux to permissive (if SELINUX=disabled)

Change `SELINUX=disabled` to `SELINUX=permissive` by running the following command. The Red Hat OpenShift installation fails if SELinux is disabled on the helper node:

```
$ vi /etc/selinux/config # change "SELINUX=disabled" to "SELINUX=permissive"
$ setenforce Permissive
$ vi /etc/default/grub  # change "selinux=0" to "selinux=1"
$ grub2-mkconfig
$ reboot
$ getenforce
```

2.5.6 Downloading the Red Hat OpenShift pull-secret

Complete the following steps:

1. Create a place to store your `pull-secret` by running the following command:

   ```
   $ mkdir -p ~/.openshift
   ```

2. Go to try.openshift.com and select **Run on Power**. Download your `pull-secret` and save it under `~/.openshift/pull-secret` by running the following command:

   ```
   $ ls -1 ~/.openshift/pull-secret
   /root/.openshift/pull-secret
   ```

Note: Do not manually download the Red Hat OpenShift client or installer packages from the website. The required packages are downloaded automatically by the playbook.

2.5.7 Creating the user SSH keys on the helper node

You can use the **ssh-keygen** tool to create the user's SSH public key (change *user@sample.com* to the user's email address) by running the following command:

```
$ ssh-keygen -t rsa -b 4096 -N '' -C "<user@sample.com>"
$ eval "$(ssh-agent -s)"
$ ssh-add ~/.ssh/id_rsa
$ ls -1 ~/.ssh/id_rsa
/root/.ssh/id_rsa
```

2.5.8 Authorizing password-less SSH for the helper node user on the HMC

Complete the following steps:

1. Log in to the HMC as <hmc_user>.
2. Authorize password-less SSH by running the **mkauthkeys** command and by using the public SSH key from the root user of the helper node:

   ```
   hmc_user@hmc_hostname:~> mkauthkeys -a "ssh-rsa
   <secret_content_of_/root/.ssh/id_rsa.pub> <user@sample.com>"
   ```

2.5.9 Checking password-less SSH for the helper node user on the HMC

Log in to the helper node as root by running the following command:

```
$ ssh hmc_user@hmc_hostname lshwres -m <managed_system> -r virtualio --rsubtype
eth --level lpar -F lpar_name,mac_addr
ocp4-helper,664A9A48690B
ocp4-bootstrap,664A9EC9CE0B
ocp4-master0,664A91C9280B
ocp4-master1,664A927A570B
ocp4-master2,664A9838420B
ocp4-worker0,664A97C5BB0B
ocp4-worker1,664A949F5F0B
```

2.5.10 Downloading all playbooks for the Red Hat OpenShift installation

You can download the playbooks by running the following commands:

- **$ git clone https://github.com/ocp-power-automation/ocp4-upi-powervm-hmc.git**
- **$ cd ocp4-upi-powervm-hmc/**
- **$ git submodule update --init --recursive --remote**

2.5.11 Creating the installation variable file vars-powervm.yaml in the ocp4-upi-powervm-hmc directory

Run the following commands:

- **$ cp examples/vars-powervm.yaml**
- **$ vi vars-powervm.yaml**

Attention: Update all <values> that are marked with *less than* and *greater than* characters in the vars-powervm.yaml file, as shown in Example 2-1.

Example 2-1 Updating the values

```
---
############################################################
# Variables defined for use by ocp4-upi-powervm-playbooks
# pvm_hmc : The HMC host IP and user. It is used to run the HMC CLI remotely. The
helper must be able to run ssh to HMC without a password.
############################################################
pvm_hmc: <hmc_user>@<hmc_ip>

############################
# OCP4 helper node variables
# Docu:
https://github.com/RedHatOfficial/ocp4-helpernode/blob/master/docs/vars-doc.md
# pvmcec: The physical machine where the LPAR(node) is running on
# pvmlpar: The LPAR(node) name in HMC
### Note: pvmcec and pvmlpar are required for all cluster nodes that are defined
in this yaml file
disk: sda
helper:
  name: "<ocp4-helper_hostname>"
  ipaddr: "<helper_ip>"
dns:
  domain: "<sample.com>"
  clusterid: "ocp4"
  forwarder1: "<existing_dns_1_ip>"
  forwarder2: "<existing_dns_2_ip>"
dhcp:
  router: "<router_ip_c_net>.1"
  bcast: "<router_ip_c_net>.255"
  netmask: "255.255.255.0"
  poolstart: "<helper_ip>"
  poolend: "<worker2_ip>"
  ipid: "<router_ip_c_net>.0"
  netmaskid: "255.255.255.0"
bootstrap:
  name: "<ocp4-bootstrap_hostname>"
  ipaddr: "<bootstrap_ip>"
  macaddr: "<66:4a:9e:c9:ce:0b>"
  pvmcec: <managed_system>
  pvmlpar: ocp4-bootstrap
masters:
  - name: "<ocp4-master0_hostname>"
    ipaddr: "<master0_ip>"
    macaddr: "<66:4a:91:c9:28:0b>"
```

```
    pvmcec: <managed_system>
    pvmlpar: ocp4-master0
  - name: "<ocp4-master1_hostname>"
    ipaddr: "<master1_ip>"
    macaddr: "<66:4a:92:7a:57:0b>"
    pvmcec: <managed_system>
    pvmlpar: ocp4-master1
  - name: "<ocp4-master2_hostname>"
    ipaddr: "<master2_ip>"
    macaddr: "<66:4a:98:38:42:0b>"
    pvmcec: <managed_system>
    pvmlpar: ocp4-master2
workers:
  - name: "<ocp4-worker0_hostname>"
    ipaddr: "<worker0_ip>"
    macaddr: "<66:4a:97:c5:bb:0b>"
    pvmcec: <managed_system>
    pvmlpar: ocp4-worker0
  - name: "<ocp4-worker1_hostname>"
    ipaddr: "<worker1_ip>"
    macaddr: "<66:4a:94:9f:5f:0b>"
    pvmcec: <managed_system>
    pvmlpar: ocp4-worker1

############################
# OCP 4 release to install
# Before changing check if new download location exists:
# https://mirror.openshift.com/pub/openshift-v4/ppc64le/dependencies/rhcos/{{
ocp_release }}/latest/
# https://mirror.openshift.com/pub/openshift-v4/ppc64le/clients/ocp/stable-{{
ocp_release }}/
ocp_release: 4.6

############################################################
# The variables below should be changed only if needed.
############################################################

ssh_gen_key: false
ppc64le: true

setup_registry:
  deploy: false
  autosync_registry: true
  registry_image: docker.io/ppc64le/registry:2
  local_repo: "ocp4/openshift4"
  product_repo: "openshift-release-dev"
  release_name: "ocp-release"
  release_tag: "4.3.27-ppc64le"

chronyconfig:
   enabled: false

##############################
# URL path to OCP download site
ocp_base_url: "https://mirror.openshift.com/pub/openshift-v4/ppc64le"
```

```
######################
# RHCOS server for OCP
ocp_rhcos_base: "dependencies/rhcos/{{ ocp_release }}"
ocp_rhcos_tag: "latest"
######### OCP 4.5 #############
#ocp_bios: "{{ ocp_base_url}}/{{ ocp_rhcos_base }}/{{ ocp_rhcos_tag
}}/rhcos-metal.ppc64le.raw.gz"
#ocp_initramfs: "{{ ocp_base_url}}/{{ ocp_rhcos_base }}/{{ ocp_rhcos_tag
}}/rhcos-installer-initramfs.ppc64le.img"
#ocp_install_kernel: "{{ ocp_base_url}}/{{ ocp_rhcos_base }}/{{ ocp_rhcos_tag
}}/rhcos-installer-kernel-ppc64le"
######### OCP 4.6 #############
ocp_bios: "{{ ocp_base_url}}/{{ ocp_rhcos_base }}/{{ ocp_rhcos_tag
}}/rhcos-live-rootfs.ppc64le.img"
ocp_initramfs: "{{ ocp_base_url}}/{{ ocp_rhcos_base }}/{{ ocp_rhcos_tag
}}/rhcos-live-initramfs.ppc64le.img"
ocp_install_kernel: "{{ ocp_base_url}}/{{ ocp_rhcos_base }}/{{ ocp_rhcos_tag
}}/rhcos-live-kernel-ppc64le"
########################
# Client/install for OCP
ocp_client_base: "clients/ocp"
ocp_client_tag: "stable-{{ ocp_release }}"
########################
ocp_client: "{{ ocp_base_url}}/{{ ocp_client_base }}/{{ ocp_client_tag
}}/openshift-client-linux.tar.gz"
ocp_installer: "{{ ocp_base_url}}/{{ ocp_client_base }}/{{ ocp_client_tag
}}/openshift-install-linux.tar.gz"
helm_source: "https://get.helm.sh/helm-v3.2.4-linux-ppc64le.tar.gz"

# If "force_ocp_download: true" then download again all packages when calling
playbook again
force_ocp_download: false

# End OCP4 helper node variables
################################


#########################################################
# Variables used by ocp4-playbook
# Docu: https://github.com/ocp-power-automation/ocp4-playbooks
# pull_secret: pull-secret file for access OpenShift repo
# public_ssh_key: the public key for ssh to access the cluster nodes from helper
#########################################################
install_config:
  cluster_domain: "{{ dns.domain }}"
  cluster_id: "{{ dns.clusterid }}"
  pull_secret: '{{ lookup("file", "~/.openshift/pull-secret") | from_json |
to_json }}'
  public_ssh_key: "{{ lookup('file', '~/.ssh/id_rsa.pub') }}"

# workdir: is the working directory for openshift-install
workdir: "~/ocp4-pvm"
# storage_type: <Storage type used in the cluster. Eg: nfs (Note: Currently, NFS
provisioner is not configured by using this playbook.
```

```
#                This variable is only used for setting up image registry to
EmptyDir if storage_type is not nfs)>
storage_type:
# log_level: <Option --log-level in openshift-install commands. Default is 'info'>
log_level: debug
# release_image_override: '<This is set to
OPENSHIFT_INSTALL_RELEASE_IMAGE_OVERRIDE while creating ign files.
#                           If you are using internal artifactory then ensure that
you have added auth key to the pull_secret>'
release_image_override: ""
# node connection timeout
node_connection_timeout: 2700
# rhcos_kernel_options: <List of kernel options for RHCOS nodes eg:
["slub_max_order=0","loglevel=7"]>
rhcos_kernel_options: []
sysctl_tuned_options: false
enable_local_registry: "{{ setup_registry.deploy }}"
powervm_rmc: true

####################################################
# Set up the proxy server on helper node if set it to true
setup_squid_proxy: false

################################
# using a predefined proxy server
#proxy_url: "http://192.168.79.2:3128"
#no_proxy: "127.0.0.1,localhost,192.168.0.0/16"
proxy_url: ""
no_proxy: ""

#ocp_haproxy_vip: 9.47.89.173
ocp_haproxy_vip: ""
```

2.5.12 Running the playbook

Run the playbook to install the complete Red Hat OpenShift V4 cluster by running the following command:

```
$ ansible-playbook  -e @vars-powervm.yaml playbooks/main.yaml
```

2.5.13 Checking the installation progress

With your Notebook, go to the Red Hat OpenShift installation status page. If you use a Notebook with an X11 server and an SSH session with X11 forwarding to connect to the helper node, then you can also start Firefox from the helper node by running the following command:

```
$ firefox http://<helper_ip>:9000
```

You see that the bootstrap LPAR turns green, then the masters turn green, and then the bootstrap turns red. Next, all workers turn green.

Also, you can check all th cluster node LPAR statuses by going to the HMC partition list view.

Watch your Certificate Signing Requests (CSRs) (without stopping the playbook) in another shell session on the helper node by running the following command. They can take some time. You see all your node CSRs with the `AVAILABLE=False` status changing to `AVAILABLE=True`

```
$ watch oc get csr
```

2.5.14 Finishing the installation

Set the registry for your cluster. For proof of concepts (PoCs), you can use emptyDir as image registry storage by running the following command. To use persistent volumes (PVs) as image registry storage, see Configuring registry storage for IBM Power Systems.

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge
--patch '{"spec":{"storage":{"emptyDir":{}}, "managementState": "Managed"}}'
```

2.5.15 Deleting the bootstrap LPAR

Stop and delete the bootstrap LPAR.

2.5.16 Logging in to the web console

The Red Hat OpenShift V4 web console is running at the following website:

```
https://console-openshift-console.apps.{{ dns.clusterid }}.{{ dns.domain }}
```

For example, https://console-openshift-console.apps.ocp4.example.com, where you can log in by using the following credentials:

- Username: kubeadmin.
- Password: Output of **`cat /root/ocp4-pvm/auth/kubeadmin-password`**.

2.6 Postinstallation tasks

This section describes the postinstallation tasks.

2.6.1 Configuring an HTPasswd identity provider

The first task after logging in to the cluster is to configure an identity provider that is used to authenticate more users by using a user ID and password. You can find the documentation for this task at Red Hat OpenShift documentation.

The Identity Provider `HTPasswd` can be used for the first tests with Red Hat OpenShift by completing the following steps:

1. Create the `users.htpasswd` file on the helper node and add multiple users by running the following commands:
   - **`htpasswd -c -B -b users.htpasswd <userid1> <init_passwd>`**
   - **`htpasswd -B -b users.htpasswd <userid2> <init_passwd>`**
   - **`htpasswd -B -b users.htpasswd <userid3> <init_passwd>`**

2. Use the link that is provided by the Red Hat OpenShift web console cluster OAuth configuration to go to the configuration for the identity provider, as shown in Figure 2-3.

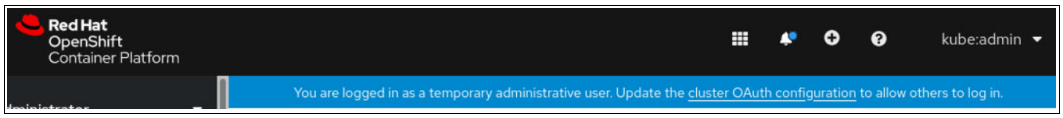


Figure 2-3 Red Hat OpenShift Container Platform Console

3. Click **Add** and select **HTPasswd** for your first tests with Red Hat OpenShift, as shown in Figure 2-4.

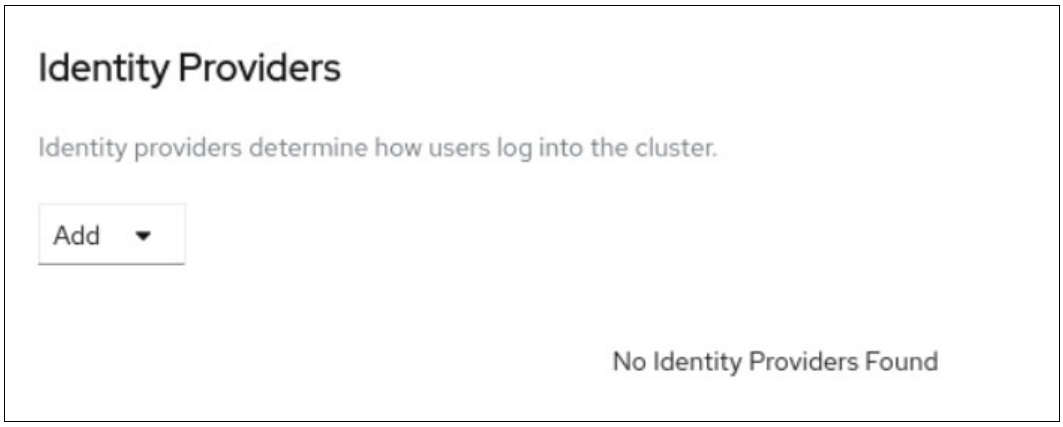


Figure 2-4 Red Hat OpenShift Container Platform Console: Identity Providers window

4. In the Add Identity Provider: HTPasswd window, click **Browse** to select the `users.htpasswd` file from the host where you started the browser. Then, click **Add** to activate the `users.htpasswd` file in the Red Hat OpenShift cluster, as shown in Figure 2-5.

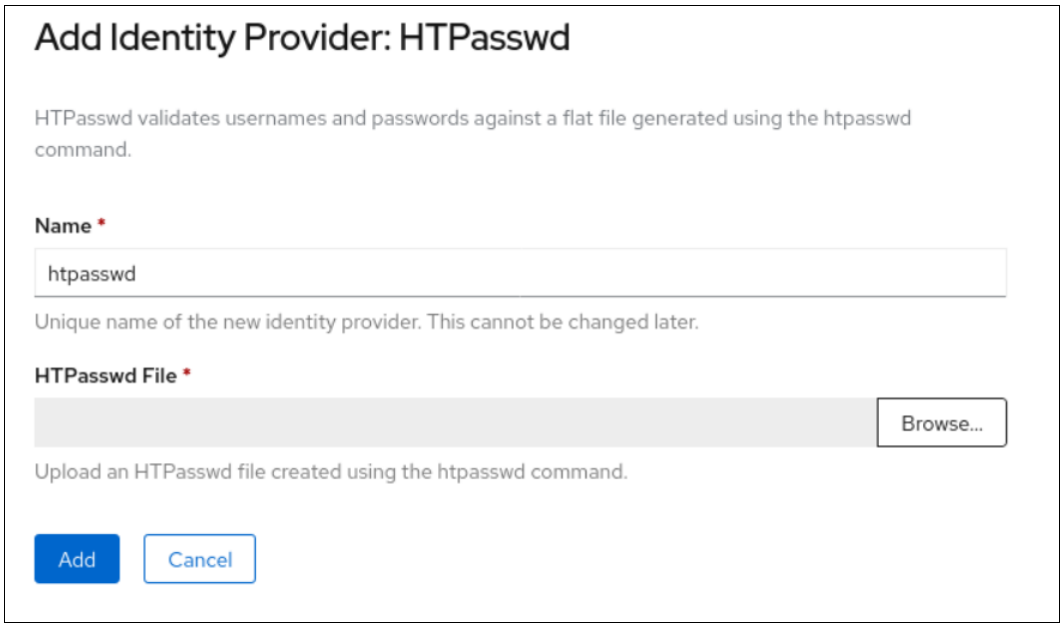


Figure 2-5 Add Identity Provider: HTPasswd window

2.6.2 Setting SELinux to disabled on all worker nodes

For the SAP workload, you must set SELinux to disabled.

The commands work only if all nodes in the cluster have the status `UPDATED=True` and `UPDATING=False`, as shown in Example 2-2.

Example 2-2 Checking the cluster configuration status

```
[root@ocp4-<helper_hostname> ~]# oc login ...
[root@ocp4-<helper_hostname> ~]# oc get machineconfigpool
NAME     CONFIG                                             UPDATED   UPDATING
DEGRADED   MACHINECOUNT   READYMACHINECOUNT   UPDATEDMACHINECOUNT
DEGRADEDMACHINECOUNT   AGE
master   rendered-master-2b61ea632c288521ff146bee1811dff1   True      False
False     3             3                   3                   0
17d
worker   rendered-worker-433ab42c91475db7366acd6ce9ce8385   True      False
False     2             2                   2                   0
17d
```

Complete the following steps:

1. Run the command on the helper node to disable SELinux, as shown in Example 2-3.

   Example 2-3 Disabling SELinux on the helper node

   ```
   [root@ocp4-<helper_hostname> ~]# oc create -f- <<_EOF_
   {'apiVersion': 'machineconfiguration.openshift.io/v1', \
   'kind': 'MachineConfig', \
   'metadata': {'labels': {'machineconfiguration.openshift.io/role': 'worker'}, \
   'name': '05-worker-kernelarg-selinuxoff'}, \
   'spec': {'config': {'ignition': {'version': '2.2.0'}}, \
   'kernelArguments': ['selinux=0']}}
   _EOF_
   ```

   The command triggers a restart of all worker nodes in sequence (one node after the other). The command finishes when all worker nodes in the cluster have the status `Ready`, as shown in Example 2-4.

   Example 2-4 Checking the status of the nodes in the cluster

   ```
   [root@ocp4-<helper_hostname> ~]# oc get nodes
   NAME                        STATUS   ROLES    AGE   VERSION
   ocp4-<master-0_hostname>    Ready    master   17d   v1.18.3+ac53d20
   ocp4-<master-1_hostname>    Ready    master   17d   v1.18.3+ac53d20
   ocp4-<master-2_hostname>    Ready    master   17d   v1.18.3+ac53d20
   ocp4-<worker-0_hostname>    Ready    worker   17d   v1.18.3+ac53d20
   ocp4-<worker-1_hostname>    Ready    worker   17d   v1.18.3+ac53d20
   ```

2. Check the SELinux setting by running the following command for all worker nodes:

   ```
   root@ocp4-<helper_hostname> ~]# ssh core@<worker_hostname> "getenforce"
   ```

   The output is:

   ```
   Disabled
   ```

For more information, see Adding kernel arguments to nodes.

2.6.3 Setting runtime limits on all worker nodes

With the SAP workload, you must set the limit for the number of processes running to `8192` and the limit for the container images to 30 GB.

The commands work only if all nodes in the cluster have the status `UPDATED=True` and `UPDATING=False`, as shown in Example 2-5.

Example 2-5 Checking the cluster configuration status

```
[root@ocp4-<helper_hostname> ~]# oc login ...
[root@ocp4-<helper_hostname> ~]# oc get machineconfigpool
NAME     CONFIG                                             UPDATED   UPDATING
DEGRADED   MACHINECOUNT   READYMACHINECOUNT   UPDATEDMACHINECOUNT
DEGRADEDMACHINECOUNT   AGE
master   rendered-master-2b61ea632c288521ff146bee1811dff1   True      False
False     3             3                   3                   0
17d
worker   rendered-worker-433ab42c91475db7366acd6ce9ce8385   True      False
False     2             2                   2                   0
17d
```

Complete the following steps:

1. Run the commands on the helper node to set the limits, as shown in Example 2-6.

   Example 2-6 Setting limits on the helper node

   ```
   [root@ocp4-<helper_hostname> ~]# oc label machineconfigpool worker \
   custom-crio=set-sap-config
   [root@ocp4-<helper_hostname> ~]# oc create -f- <<_EOF_
   {'apiVersion': 'machineconfiguration.openshift.io/v1', \
   'kind': 'ContainerRuntimeConfig', \
   'metadata': {'name': 'set-sap-config'}, \
   'spec': {'machineConfigPoolSelector': \
   {'matchLabels': {'custom-crio': 'set-sap-config'}}, \
   'containerRuntimeConfig': {'pidsLimit': 8192, 'overlaySize': '30G'}}}
   _EOF_
   ```

   The command triggers a restart of all worker nodes. This is done sequentially (one node after the other). The command finishes when all worker nodes in the cluster have the status `Ready`, as shown in Example 2-7.

   Example 2-7 Checking the status of the nodes in the cluster

   ```
   [root@ocp4-<helper_hostname> ~]# oc get nodes
   NAME                       STATUS   ROLES    AGE   VERSION
   ocp4-<master-0_hostname>    Ready    master   17d   v1.18.3+ac53d20
   ocp4-<master-1_hostname>    Ready    master   17d   v1.18.3+ac53d20
   ocp4-<master-2_hostname>    Ready    master   17d   v1.18.3+ac53d20
   ocp4-<worker-0_hostname>    Ready    worker   17d   v1.18.3+ac53d20
   ocp4-<worker-1_hostname>    Ready    worker   17d   v1.18.3+ac53d20
   ```

2. Check the `pids_limit` parameter by running the following command for all worker nodes:

   ```
   root@ocp4-<helper_hostname> ~]# ssh core@<worker_hostname> "crio config
   2>/dev/null | grep 'pids_limit'"
   ```

The output is:

```
pids_limit = 8192
```

For more information, see Adding kernel arguments to nodes.

2.6.4 Setting up an NFS server for database data and logs on the helper node

After the Red Hat OpenShift cluster is running, look for an NFS server that is configured on the helper node, as shown in Example 2-8.

Example 2-8 Checking the status of the NFS server

```
# systemctl status nfs-server
? nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; vendor
preset: disabled)
  Drop-In: /run/systemd/generator/nfs-server.service.d
           ??order-with-mounts.conf
   Active: active
```

The configuration file `/etc/exports` contains the following lines:

```
# cat /etc/exports
/export *(rw,sync,root_squash)
```

The disk space that is available for the NFS server to export files can be checked by running the following command:

```
# df -h /export

Filesystem               Size  Used Avail Use% Mounted on
/dev/mapper/rootvg-root  471G   13G  459G   3% /
```

If needed, you can increase the logical volume for `rootvg-root` or assign another data disk to the LPAR and mount it to `/export`. For more information, see Chapter 29, “Exporting NFS shares”, of the *Red Hat Enterprise Linux 8 System Design Guide*.

2.6.5 Releasing node resources by using garbage collection

The worker nodes do not automatically release the space that is used by terminated containers or old container images. The administrator must configure automatic garbage collection for containers and images for all worker nodes. For more information about this task, see Freeing node resources using garbage collection.

If you run out of disk space on a worker node, see Running out of space under /var/lib/containers/storage.

3

Automated installation of SAP S/4HANA and SAP HANA on IBM Power Systems with Red Hat Ansible

This chapter describes how to install SAP S/4HANA and SAP HANA on IBM Power Systems by using Red Hat Ansible Engine and Red Hat Ansible Tower. The goal is to automate the installation of SAP S/4HANA and SAP HANA.

This chapter contains the following topics:

3.1 Introduction

Red Hat Ansible is a powerful command-line interface (CLI) automation engine to automate IT tasks. Red Hat Ansible uses simple YAML syntax to describe a configuration, offers secure access to remote systems, and integrates with other solutions. For more information about Red Hat Ansible, see Ansible documentation.

Red Hat Ansible Tower is a web-based console that makes Red Hat Ansible adaptable for IT teams. Red Hat Ansible Tower helps IT teams to scale automation, roll out updates, build configurations, organize inventory management, and schedule jobs. Red Hat Ansible Tower comes with a web interface and a REST API that can be embedded into other IT processes and tools. The Red Hat Ansible Tower web-based user interface (UI) provides an overview dashboard of all job exit statuses, successful and failed playbook runs, statuses of host inventories, role-based access control (RBAC), and a permission system for playbooks. For more information, see Red Hat Ansible Tower.

Figure 3-1 shows two techniques to install SAP software:

- Install the SAP software by using the Red Hat Ansible CLI.
- Install the SAP software by using Red Hat Ansible Tower.

You can use either one of them depending on your skill set and purpose for use. To install SAP HANA, SAP S/4HANA, and SAP NetWeaver, you must install prerequisites that are specific to the Red Hat Enterprise Linux operating system for target systems like IBM Power Systems by running SAP installer and SAP product packages. These tasks are automated with Red Hat Ansible CLI or Red Hat Ansible Tower.

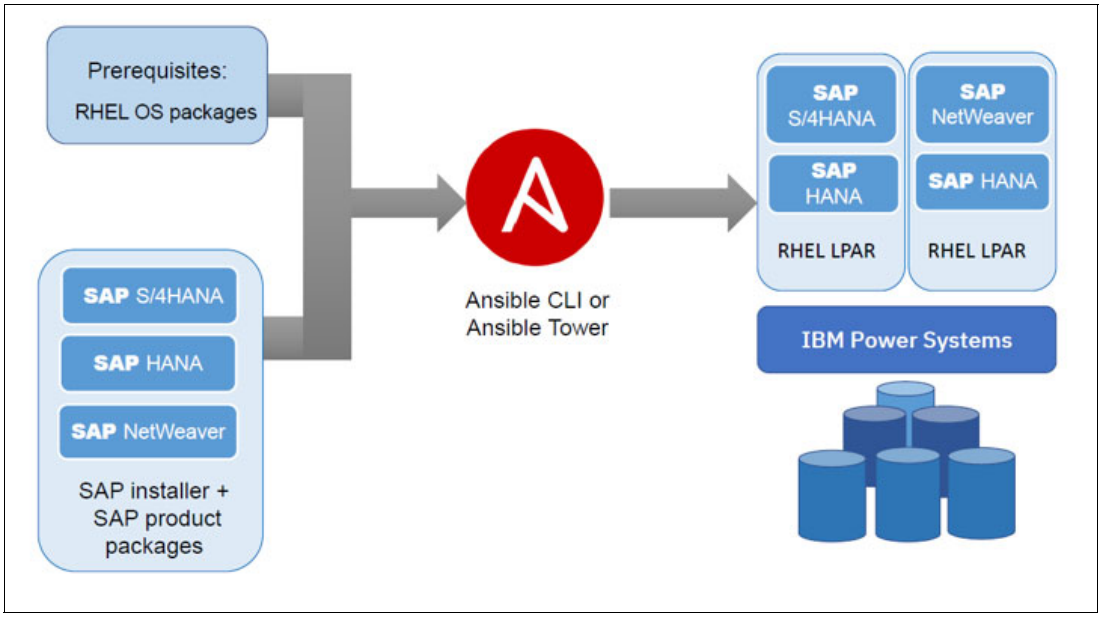


Figure 3-1 Red Hat Ansible CLI versus Red Hat Ansible Tower

For more information about how to use Red Hat Ansible automation to deploy SAP Solutions on other hardware architectures, see Automating your SAP HANA and S/4HANA by SAP deployments using Ansible.

3.2 Customer value

Deploying SAP software requires an SAP admin skill level. However, you can handle these complex deployment scenarios by using automated workflows that encapsulate this level of admin skill. Red Hat Ansible is a tool that can this job. You do not need to program extensive shell scripts because there are developed and reusable open source packages that are called Ansible Roles that you can obtain from the Ansible Galaxy community. Red Hat Ansible configuration files, which are called *playbooks*, contain instructions for the Red Hat Ansible Engine. The playbooks are written in YAML like key-value pairs, so you do not need to learn a new programming language. The configuration of prerequisites and specific configurations for SAP S/4HANA and SAP HANA are implemented and published by the Red Hat Enterprise Linux community.

Red Hat Ansible Tower is a web-based GUI solution to automate the installation of SAP S/4HANA and SAP HANA on IBM Power Systems. Red Hat Ansible Tower offers a graphical dashboard and a navigation menu to show your host status and configuration, go to your job runs and templates, show your projects, and more. Therefore, by using the Red Hat Ansible Tower visual UI, you can create a job template to automate the complete SAP software installation.

Red Hat Ansible runs commands by using SSH, so you do not need to install extra software on your server to implement authentication for client hosts.

Table 3-1 shows many of the features of Red Hat Ansible CLI or Red Hat Ansible Tower.

Table 3-1 Red Hat Ansible and Red Hat Ansible Tower features

Key value	Red Hat Ansible CLI	Red Hat Ansible Tower
Open-source automation tool.	Yes.	No.
Free for commercial use.	Yes, under the policies of General Public Access.	Yes, with a trial license.
Usability	Has a CLI for those users who are familiar with using CLI tools.	Has a web-based GUI that is easy to use and browse.
Easy to install and set up.	Yes.	Yes.
Helps to check whether your system is configured correctly before installing the prerequisites and SAP software.	Yes.	Yes.
Automates the installation of prerequisites and extra packages before installing the SAP software.	Yes.	Yes.
Automates multiple SAP software installations by running at least one script.	By command execution.	By clicking a button.
Helps scaling up by adding new hosts, building host groups, and adding multiple Red Hat Ansible nodes.	Yes.	Yes.

Key value	Red Hat Ansible CLI	Red Hat Ansible Tower
Shows each step within the running of the SAP software installation.	Yes.	Yes.
Manages access permissions for different roles.	By using the CLI.	By using the dashboard.
Monitors host statuses and job runs in real time.	Yes.	Yes.
Helps to automate, deploy, and monitor applications in complex environments.	Yes.	Yes.
Configures the sending of notifications about the automation status.	Yes.	Yes.

3.3 Use case: Unattended installation of SAP reference and test systems

This section describes the unattended installation of a single logical partition (LPAR) SAP Standard System. This Red Hat Ansible based installation of SAP S/4HANA or SAP NetWeaver and the SAP HANA database on a single server is used as an example for installing reference and test systems.

These SAP instances are installed as shown in Table 3-2.

Table 3-2 SAP installed instances

SAP Instance	Instance number	Instance name	SAPSID
SAP HANA database	20	HDB20	ABD
Advanced Business Application Programming (ABAP) SAP Central Services (ASCS) instance	21	ASCS21	AB1
Primary Application Server (PAS) instance	22	D22 or DVEBMGS22	AB1

The chosen SAP instance number and SAPSIDs are examples only, and may be customized by using variables.

The following values are used throughout this configuration and can be adapted to match your system characteristics:

- LPAR hostname: <*yourhostname*>
- Directory for installation files: `/data/installer`.
- Password: *XXpasswd1*

Two approaches are explained here: the Red Hat Ansible CLI, and the Red Hat Ansible Tower setup. If the CLI is sufficient, skip the Red Hat Ansible Tower description. If you are aiming for the Red Hat Ansible Tower setup, first start with the Red Hat Ansible CLI setup, verify that all settings are working, and then proceed with the Red Hat Ansible Tower setup.

3.4 Preconfiguring and setting up the environment

This section assumes that the following steps are taken:

1. Set up your LPAR on IBM Power Systems: POWER8 processor-based system or later (Little Endian and PPC64LE).
2. Install Red Hat Enterprise Linux V8 or later by following the instructions at Red Hat Enterprise Linux V8.
3. Check that your LPAR has available operating system software repositories.
4. Update all packages on your system.
5. Provide SSH root access on the target host to install SAP software.
6. Install the latest version of Python.
7. Download the SAP installer and the SAP product packages to the target host.
8. To avoid the restart handler error, check that SELinux is set to targeted and permissive.
9. The prerequisites checker `sap-netweaver-preconfigure` requires at least 20 GB of swap space that is configured for SAP NetWeaver and SAP S/4HANA installations.

Before you work with Red Hat Ansible, you must check that all machines and hosts are configured correctly.

Hint: If your root file system space is limited and a single large file system is mounted for the SAP application, you must link the various locations where the SAP application is stored to the single, large volume.

For example, if the large volume is mounted at `/data`, then create symbolic links from these directories to the new volume before starting the installation:

```
$ ls -ld /sapmnt /hana /usr/sap
lrwxrwxrwx 1 root root 10 May 6 14:31 /hana -> /data/hana
lrwxrwxrwx 1 root root 12 May 6 14:31 /sapmnt -> /data/sapmnt
lrwxrwxrwx 1 root root 13 May 6 14:30 /usr/sap -> /data/usr_sap
```

Note: The Community Roles come with a `playbook sap_hostagent/tasks/deploy_sar.yml` file that does not run remotely. With this change, the playbook can be started on a remote machine, which is essential, for example, when using Red Hat Ansible Tower. This situation can be fixed by adding the following line twice to `playbook remote_src: yes`:

```
- name: Copy SAR based SAPHOSTAGENT to the target host
  copy:
   ...
   remote_src: yes

- name: Copy SAPCAR tool to the target host
  copy:
   ...
   remote_src: yes
```

3.4.1 Repeating a playbook and uninstalling SAP

If there are typographical errors or other errors, a playbook run can fail.

The current set of playbooks does not use the resume option of the SAP installer. To start again after a failed attempt, complete the following steps:

1. Correct the errors or typographical errors in the variables and playbooks.
2. Uninstall all SAP instances that you intended to install.
3. Check that no SAP processes that contain your SAPSID are running on your target system (especially `sapstartsrv` processes).

You do not need to remove SAP related UNIX user or group accounts of an SAP instance because they can be reused without errors. Clean up your system so that the playbooks can start a full SAP installation from scratch.

3.5 Installing SAP software with Red Hat Ansible CLI

This section combines documented concepts, approaches, and experiences to describe how to install SAP by using the Red Hat Ansible CLI.

3.5.1 Getting started

To install Red Hat Ansible CLI, use a package installer like **`yum`** on your development environment. In our example, we install Red Hat Ansible CLI on Red Hat Enterprise Linux V8.1 by running the following command:

```
sudo yum install ansible
```

If you have a different platform than Red Hat Enterprise Linux, see Installing Ansible.

To implement your configuration for installing SAP software, define a playbook. Red Hat Ansible playbooks are configuration files that are written in YAML and contain all the information about target system requirements, tasks, variables, and so on. If you have a large system environment and you must automate many processes on multiple machines, divide your configuration into different files. This bundle of files is defined as Ansible Roles. They are reusable components and can be included inside any playbook. Ansible Roles are stored in their own repository that is called Ansible Galaxy.

For the example SAP software deployment, we use two sets of Ansible Roles:

- Red Hat Enterprise Linux System Roles for SAP to configure the system settings and install extra software according to the SAP Notes for Red Hat Enterprise Linux.
- Community Roles for SAP to deploy the software that is needed to run an SAP S/4HANA and SAP HANA database.

Ansible Galaxy CLI is used later to retrieve these two packages. Before you start writing playbooks, create a working directory where the playbooks will be stored. For example, Figure 3-2 on page 31 shows the project directory that stores the files and configuration files.

```
deploy_netweaver_and_hana_on_ibm_power_project/
         ├──group_vars/
         │   └──sapservers.yml
         ├──host_vars/
         │   └──<yourhostname>.yml
         ├──roles/
         │   └──(download described below)
         ├──hosts
         ├──requirements.yml
         └──sap-deploy.yml
```

Figure 3-2 Project directory to store files

3.5.2 Red Hat Ansible inventory

Red Hat Ansible manages hosts and host groups within the Ansible inventory. Based on the Ansible inventory, you can define which SAP software component is installed on which host group. An Ansible inventory file can have different formats depending on your inventory plug-ins. The most used formats are INI and YAML. For more information, see How to build your inventory.

The community roles that are mentioned use two hosts: one host for SAP S/4HANA, and another host for SAP HANA. In our setup, SAP S/4HANA and SAP HANA are installed on one machine that is named <*yourhostname*>, but the structure is kept if you want to divide the installation.

The Ansible inventory file hosts in the INI format with joined groups `hana` and `s4hana` points to the same hostname <*yourhostname*>, as shown in Example 3-1.

Example 3-1 Ansible inventory file

```
#inventory for servers
[sapservers:children]
hana
s4hana
[hana]
<yourhostname>
[s4hana]
<yourhostname>
```

To run a playbook, add the **`-i`** option and directory path to tell Red Hat Ansible where your inventory file is. To test whether all defined hosts are accessible to Red Hat Ansible, try to ping your machines by running the following command:

```
ansible all -i /path/to/your/inventory/file -m ping
```

This command displays a result for all host machines that are available for your SAP installation. If a host is not accessible from a remote machine, check your host credentials, SSH settings like SSH private key, and so on. If the SSH private key requires a passphrase, you must specify it in your inventory file. To avoid this complexity, use an SSH private key without the passphrase.

3.5.3 General installation definitions

SAP Host installation settings, SAP Domain, and other settings that are general to all hosts of the group `sapservers` are stored in a group variable file that is named `group_vars/sapservers.yml`.

SAP host agent software is installed on all hosts, so these SAP host agent settings are defined in the group variable file:

- SAP host agent installation type
- SAP host agent paths and file names

The SAP installer `sapinst` needs a host entry in `/etc/hosts` to resolve your hostname. You can either create a manual entry in order *<ip> <full qualified domain name> <short hostname>* and set `sap_preconfigure_modify_etc_hosts: false`. Alternatively, Red Hat Ansible can add the hostname entry if you set the variable `sap_preconfigure_modify_etc_hosts: true` and add the host DNS domain in variable *sap_domain*.

The group variable file `sapservers.yml` file in the *group_vars* directory is shown in Example 3-2.

Example 3-2 The sapservers.yml file

```
#Defined variables for sap_hostagent role
sap_hostagent_installation_type: "sar"
sap_hostagent_sar_local_path: "/data/installer/S4HANA1909FPS01"
sap_hostagent_sar_file_name: "SAPHOSTAGENT46_46-70002261.SAR"
sap_hostagent_sapcar_local_path: "/data/installer/S4HANA1909FPS01"
sap_hostagent_sapcar_file_name: "SAPCAR.EXE"

sap_hostagent_clean_tmp_directory: true
#Defined variables for sap_preconfigure role
sap_preconfigure_selinux_state: permissive
# If you need to modify your hostnames set up it as true
sap_preconfigure_modify_etc_hosts: false
# define the SAP domain name only if you set 'sap_preconfigure_modify_etc_hosts:
true'
#sap_domain: "subdomain.enterprise-domain-name.com"
```

To keep this demonstration simple and start a working setup quickly, passwords are added to the host variables without encryption.

Note: The file name for the **SAPCAR** tool cannot be `SAPCAR` because this file is also inside the host agent SAR file, and an error occurs when the file already exists while extracting this tool. Rename the SAPCAR tool to **SAPCAR.EXE** or **SAPCAR_ *<version>*.EXE** and specify this name as the value of the variable *sap_hostagent_sapcar_file_name*.

For more information, see the GitHub repository where the roles are implemented:

- GitHub redhat-sap/sap-hostagent
- GitHub linux-system-roles/sap-preconfigure

3.5.4 SAP HANA and SAP S/4HANA specific definitions

SAP HANA and SAP S/4HANA host-specific settings, such as software installation source paths, installer file name, and SAPSIDs, are stored in the Red Hat Ansible host variable file `host_vars/<yourhostname>.yml`.

To keep this demonstration simple, passwords are added to the host variables that are not encrypted.

Red Hat Ansible password vault: In sensitive environments, passwords can be managed by an encrypted Ansible-Vault. For more information, see Encrypting content with Ansible Vault and the description of the command-line tool in ansible-vault.

The host variable `<yourhostname>.yml` file in the `host_vars` directory is shown in Example 3-3.

Example 3-3 Host variable yml file: host_vars/<yourhostname>.yml

```
#Defined variables for SAP HANA deployment
# the following two lines must be changed in sync with two lines below:
# sap_hana_deployment_hana_sid = sap_s4hana_deployment_db_sid
# sap_hana_deployment_hana_instance_number =
sap_s4hana_deployment_hana_instance_nr
sap_hana_deployment_hana_sid: "ABD"
sap_hana_deployment_hana_instance_number: "20"
sap_hana_deployment_bundle_path: "/data/installer/S4HANA1909FPS01"
sap_hana_deployment_bundle_sar_file_name: "IMDB_SERVER20_047_0-80002046.SAR"
sap_hana_deployment_sapcar_path: "/data/installer/SAPCAR"
sap_hana_deployment_sapcar_file_name: "SAPCAR.EXE"
sap_hana_deployment_root_password: "XXpasswd1"
sap_hana_deployment_sapadm_password: "XXpasswd1"
sap_hana_deployment_hana_env_type: development
sap_hana_deployment_hana_mem_restrict: "n"
sap_hana_deployment_common_master_password: "XXpasswd3"
sap_hana_deployment_sidadm_password: "XXpasswd1"
sap_hana_deployment_hana_db_system_password: "XXpasswd2"
sap_hana_deployment_ase_user_password: "XXpasswd4"
sap_hana_deployment_apply_license: false

#Defined variables for S/4HANA deployment
sap_s4hana_deployment_product_id: "NW_ABAP_OneHost:S4HANA1909.CORE.HDB.ABAP"
sap_s4hana_deployment_sapcar_path: "/data/installer/SAPCAR"
sap_s4hana_deployment_sapcar_file_name: "SAPCAR.EXE"
sap_s4hana_deployment_swpm_path: "/data/installer/S4HANA1909FPS01"
sap_s4hana_deployment_swpm_sar_file_name: "SWPM20SP05_5-80003426.SAR"

sap_s4hana_deployment_sid: "AB1"
sap_s4hana_deployment_ascs_instance_nr: "21"
sap_s4hana_deployment_pas_instance_nr: "22"
sap_s4hana_deployment_db_host: "<yourhostname>"
# these two lines must be changed in sync with the sap_hana settings above:
sap_s4hana_deployment_db_sid: "ABD"
sap_s4hana_deployment_hana_instance_nr: "20"

sap_s4hana_deployment_db_schema_password: "XXpasswd"
```

```
sap_s4hana_deployment_db_schema_abap_password: "XXpasswd"
sap_s4hana_deployment_master_password: "XXpasswdM"
sap_s4hana_deployment_hana_systemdb_password: "xxPasswd"
sap_s4hana_deployment_hana_system_password: "xxSystemPsw"
sap_s4hana_deployment_parallel_jobs_nr: "30"
sap_s4hana_deployment_db_sidadm_password: "yourPasswd"

sap_s4hana_deployment_igs_path: "/data/installer/S4HANA1909FPS01"
sap_s4hana_deployment_igs_file_name: "igsexe_10-80003246.sar"
sap_s4hana_deployment_igs_helper_path: "/data/installer/S4HANA1909FPS01"
sap_s4hana_deployment_igs_helper_file_name: "igshelper_17-10010245.sar"
sap_s4hana_deployment_kernel_dependent_path: "/data/installer/S4HANA1909FPS01"
sap_s4hana_deployment_kernel_dependent_file_name: "SAPEXEDB_100-80004417.SAR"
sap_s4hana_deployment_kernel_independent_path: "/data/installer/S4HANA1909FPS01"
sap_s4hana_deployment_kernel_independent_file_name: "SAPEXE_100-80004418.SAR"
sap_s4hana_deployment_software_path: "/data/installer/S4HANA1909FPS01"
sap_s4hana_deployment_sapadm_password: "spAdmpass1"
sap_s4hana_deployment_sap_sidadm_password: "spAdmpass2"
```

These role variables are defined under:

- GitHub redhat-sap/sap-hana-deployment
- GitHub redhat-sap/sap-s4hana-deployment

3.5.5 Getting Community and System Roles from the Red Hat Ansible Galaxy requirements.yml file

According to SAP Note 2772999, there are prerequisites for installing SAP, such as packages and system settings. These prerequisites must be implemented before installing and running SAP systems. These prerequisites are implemented as Ansible Roles and can be used to configure all required changes on the Red Hat target server. For Red Hat Enterprise Linux V8.1, the following three Red Hat Enterprise Linux System Roles for SAP prerequisites must be applied:

- `sap-preconfigure`
- `sap-netweaver-preconfigure`
- `sap-hana-preconfigure`

Additionally, the following community roles for SAP software deployment are required:

- `sap-hostagent`
- `sap-s4hana-deployment`
- `sap-hana-deployment`

You install both package groups in one step. Therefore, your `requirements.yml` file contains two files, which are described in Automating your SAP HANA and S/4HANA by SAP deployments using Ansible - Part 2 and Automating your SAP HANA and S/4HANA by SAP deployments using Ansible - Part 3.

These roles are available in Red Hat Community repositories and in Ansible Galaxy. You can choose which source is defined in your playbook. For our example, we add all required Ansible Roles to the playbook `requirements.yml` file, as shown in Example 3-4 on page 35.

Example 3-4 Ansible Roles that are defined for the playbook requirements.yml file

```
#From GitHub repository:
- src: https://github.com/linux-system-roles/sap-preconfigure.git
- src: https://github.com/linux-system-roles/sap-hana-preconfigure.git
- src: https://github.com/linux-system-roles/sap-netweaver-preconfigure.git
#From Ansible Galaxy repository:
- name: redhat_sap.sap_hostagent
- name: redhat_sap.sap_hana_deployment
- name: redhat_sap.sap_s4hana_deployment
```

Before the **ansible-galaxy** command can be run to download files from GitHub, check that the Git software is installed on your machine. If it is not installed, run the following command:

```
sudo yum install git
```

Now, install all the required roles in the directory roles by running the following Ansible Galaxy command:

```
ansible-galaxy install -r requirements.yml -p roles
```

3.5.6 SAP software deployment: The sap-deploy.yml file

Finally, create a playbook that is named `sap-deploy.yml`, which includes all the required system preparation and community software deployment rules, as shown in Example 3-5.

Example 3-5 The sap-deploy.yml playbook

```
---
- hosts: sapservers
roles:
- { role: redhat_sap.sap_hostagent }
- { role: sap-preconfigure }
- hosts: hana
roles:
- { role: sap-hana-preconfigure }
- { role: redhat_sap.sap_hana_deployment }
- hosts: s4hana
roles:
- { role: sap-netweaver-preconfigure }
- { role: redhat_sap.sap_s4hana_deployment }
```

To start the automation deployment, run the following command:

```
ansible-playbook -i hosts sap-deploy.yml
```

After the playbook finishes without errors, SAP hostagent, SAP HANA, and S/4HANA are installed on your host.

3.6 Installing SAP software with Red Hat Ansible Tower

This section describes how to install SAP with Red Hat Ansible Tower.

3.6.1 Starting with Red Hat Ansible Tower

To use Red Hat Ansible Tower for the unattended Ansible installation, you need the following requirements:

- A host with Red Hat Ansible Tower installed.
- A Red Hat Ansible Automation Platform license. You can try a trial license for 60 days at no cost.
- A Red Hat Ansible Tower user account with the following permissions:
  - Administrative permissions to edit your Red Hat Ansible Tower inventory and project (each permission must be created one at a time).
  - Permissions to create Red Hat Ansible Tower templates and credentials.
- A working SSH connection from Red Hat Ansible Tower Server to your SAP installation target system.

The following guidelines are used for configuring Red Hat Ansible Tower:

- Define your inventory by adding groups and hosts to the configuration if needed.
- Create or choose the credential for Ansible Tower to connect to and run Red Hat Ansible playbooks.
- Create or choose one project that will be used for your playbook to run the SAP software installation.
- Create a template that contains the playbook and installation parameters.

The following sections describe these steps in more detail.

3.6.2 Setting up a directory for Ansible roles

A local directory on the Red Hat Ansible Tower server is used as file storage instead of a Git source code management repository to achieve two goals:

- A single, combined repository for all Red Hat system and Community roles.
- You can customize roles to enable running them from the Red Hat Ansible Tower server.

To set up the directory, complete the following steps:

1. Create a target directory on the Red Hat Tower server for the Ansible roles by running the following command:

   ```
   REPOS=/var/lib/awx/projects
   sudo mkdir $REPOS/sap_installation
   ```

2. Assign directory ownership to your UNIX user account by running the following command:

   ```
   sudo chown $LOGNAME: $REPOS/sap_installation
   ```

3. Ensure that the new directory is accessible by assigning execute permissions to the parent directory by running the following command:

   ```
   sudo chmod o+x $REPOS
   ```

The new directory is now ready to be filled with content.

3.6.3 Preparing a custom repository

The installation target machine <yourhostname> can be used to prepare a storage space for all Ansible roles first, and then you can copy files to the Red Hat Ansible Tower server.

In our example, the roles directory structure is shown in Figure 3-3.

```
sap_installation/
        ├──roles/
        │   └──(download described below)
        └──sap-deploy.yml
```

Figure 3-3 Red Hat Ansible roles directory structure

To prepare the repository, complete the following steps:

1. Install the required software on to your Red Hat System <yourhostname> by running the following command:

   ```
   yum install -y git ansible
   ```

2. Create the working folder `sap_installation` and then change to it by running the following commands:

   ```
   mkdir ~/sap_installation
   cd ~/sap_installation
   ```

3. Create the `sap_deploy.yml` file under the working folder `sap_installation` with the contents shown in Example 3-6.

   Example 3-6 The sap_deploy.yml file

   ```
   ---
   - hosts: "{{ sap_hostagent_hostname }}"
     roles:
        - { role: sap-preconfigure }
        - { role: redhat_sap.sap_hostagent }
   - hosts: "{{ sap_hana_hostname }}"
     roles:
        - { role: sap-hana-preconfigure }
        - { role: redhat_sap.sap_hana_deployment }
   - hosts: "{{ sap_s4hana_hostname }}"
     roles:
        - { role: sap-netweaver-preconfigure }
        - { role: redhat_sap.sap_s4hana_deployment }
   ```

   The three host variables *sap_hostagent_hostname*, *sap_hana_hostname*, and *sap_s4hana_hostname* are defined in the Red Hat Tower GUI later.

4. Create a second file that is named `requirements.yml` with the contents that are shown in Example 3-7.

   Example 3-7 Contents for the second requirements.yml file

   ```
   #From GitHub repository:
   - src: https://github.com/linux-system-roles/sap-preconfigure.git
   - src: https://github.com/linux-system-roles/sap-hana-preconfigure.git
   - src: https://github.com/linux-system-roles/sap-netweaver-preconfigure.git
   #From Ansible Galaxy repository:
   ```

```
- name: redhat_sap.sap_hostagent
- name: redhat_sap.sap_hana_deployment
- name: redhat_sap.sap_s4hana_deployment
```

5. Download all roles by running the following command:

   ```
   ansible-galaxy install -r requirements.yml -p roles
   ```

   All roles are now together in a single storage location.

6. At the time of writing, the `deploy_sar.yml` file must be patched to include `remote_src: yes`. Check whether your version needs this change by running the following command:

   ```
   grep "remote_src:" -B2 roles/redhat_sap.sap_hostagent/tasks/deploy_sar.yml
   ```

   If the last command did not find the parameter (there is no output), then run this command:

   ```
   sed -i.bak 's/copy:/copy:\n   remote_src: yes/'
   roles/redhat_sap.sap_hostagent/tasks/deploy_sar.yml
   ```

   This command adds the required **`remote_src`** option to the **`copy`** commands, which is required to copy files when they start from the Red Hat Tower server.

7. The combined and modified Ansible rules set are now ready and can be copied to the Red Hat Ansible Tower server by running the following command:

   ```
   scp -r sap-deploy.yml roles
   <toweruser>@<towerserver>:/var/lib/awx/projects/sap_installation/
   ```

Note: The `requirements.yml` file is intentionally not copied to avoid accidentally overwriting customized rules by automatically downloading them again.

3.6.4 Setting up a project

Red Hat Ansible Tower uses projects as logical collections of one or more Red Hat Ansible Playbooks.

Using the Red Hat Tower web interface, create a new project, as shown in Figure 3-4. Select **Projects** in the left menu, and then click **+** at the upper right.

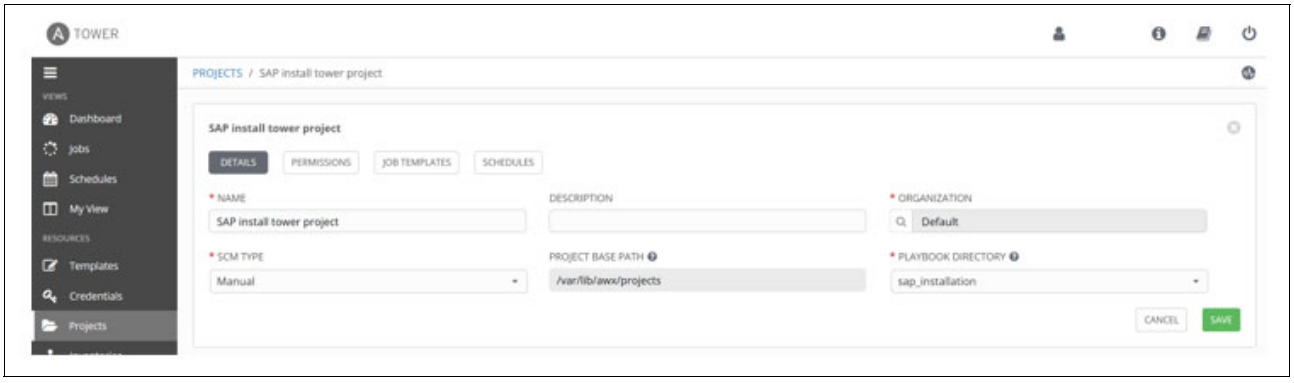


Figure 3-4 Red Hat Tower: Projects window

To set up a project, complete the following steps:

1. Define a project name.
2. Enter a project description.

3. Choose **Manual** as the scm type, which allows the local directory to be a repository source.
4. Leave the project base path as the default.
5. Select the `sap_installation` directory as the playbook directory from the drop-down list.
6. Click **Save**.

3.6.5 Setting up inventory

An inventory in Red Hat Ansible Tower contains hosts and groups like a Red Hat Ansible inventory from a CLI does, but it is extended for the Tower web interface with fields for organization, user permissions, and more. For more information, see Inventories.

To set up an inventory, complete the following steps:

1. Click **Inventories** in the left menu. If you do not have an inventory, click **+** to create one, as shown in Figure 3-5. Define the properties that you use.

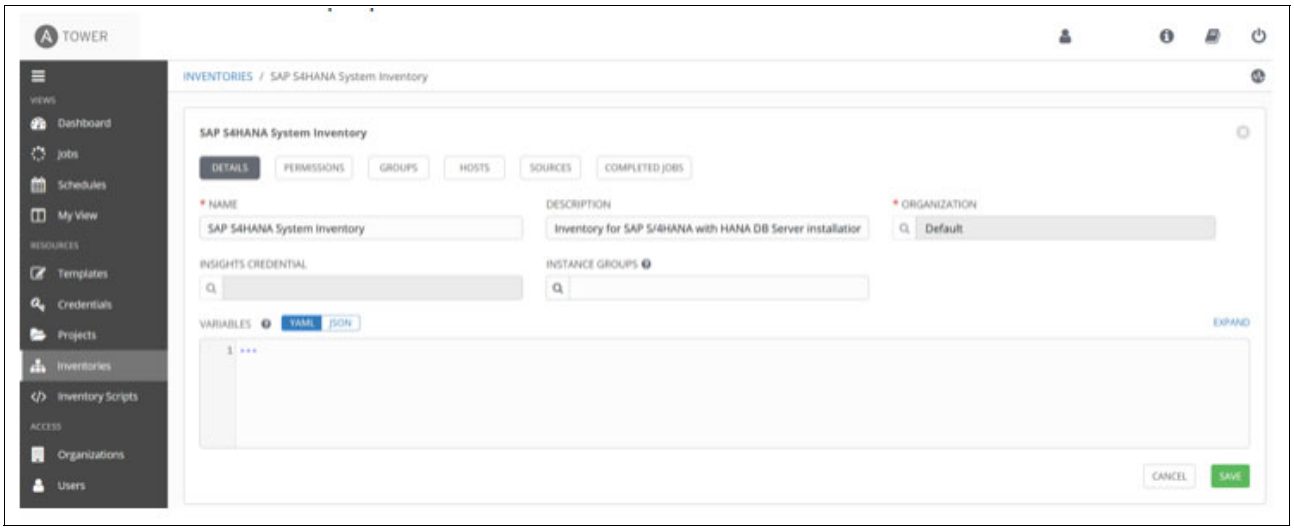


Figure 3-5 Red Hat Tower: Inventories window

The current setup description does not use groups because in a standard SAP setup all instances are installed on one host.

2. Add your host by clicking **+** in the **Hosts** tab, as shown in Figure 3-6.

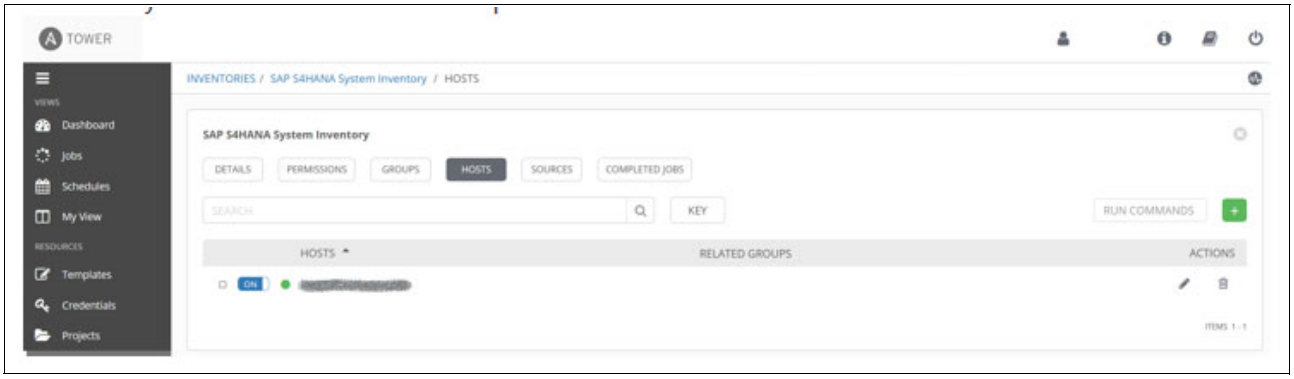


Figure 3-6 Red Hat Ansible Tower: Inventory hosts window

With Red Hat Ansible Tower, you can define variables at different locations. This setup uses extra variables that are defined in the template. Extra variables overwrite all values that are defined for the same variable at other locations. For more information about this topic, see the “Ansible Tower Variable Precedence Hierarchy (last listed wins)” table in Extra Variables.

For this reference system, all installation parameters are stored in template variables, as described in 3.6.7, “Defining a job template” on page 41.

After saving the new inventory, proceed by configuring permissions for users and team members. For more information about how to configure your inventory, see Inventories.

3.6.6 Setting up target host credentials

Red Hat Ansible Tower uses credentials for authentication and building connections to remote hosts when jobs are run on machines to install SAP software. You must set up your host credentials, such as username, password, and an existing SSH key.

If you do not have an SSH key, you can use the **`ssh-keygen`** tool to generate it on the target host and copy it to the Red Hat Ansible Tower credentials. Click **Credentials** in the left menu to see the window that displays all the available credentials (Figure 3-7).

Create a credential by clicking **+** at the upper right.

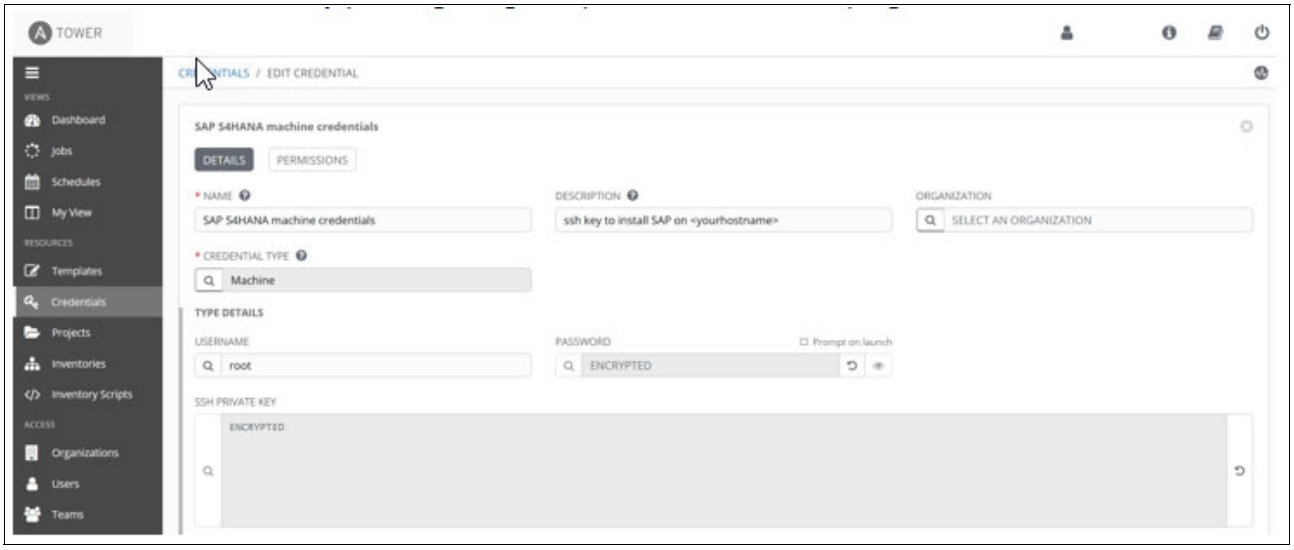


Figure 3-7 Red Hat Ansible Tower: Credentials window

To set the SAP application installation host credentials, complete the following steps:

1. Enter a hostname, for example, *yourhostname*.
2. Enter a description, for example, “SAP S/4HANA reference system”.
3. Use **Machine** as the credential type.
4. Enter **root** as the username for installation.
5. Enter the password to be used for the SSH authentication.
6. Enter the SSH private key and, if used, the passphrase for your key.

The SSH key is used when copying files by using the username and password to run the playbook on the target host. For more information, see Credentials.

3.6.7 Defining a job template

A job template defines the parameters that are used to run Ansible playbooks. For more information, see Job Templates.

To create a template, complete the following steps:

1. Select **Template** from the left menu and click + at the upper right, as shown in Figure 3-8.

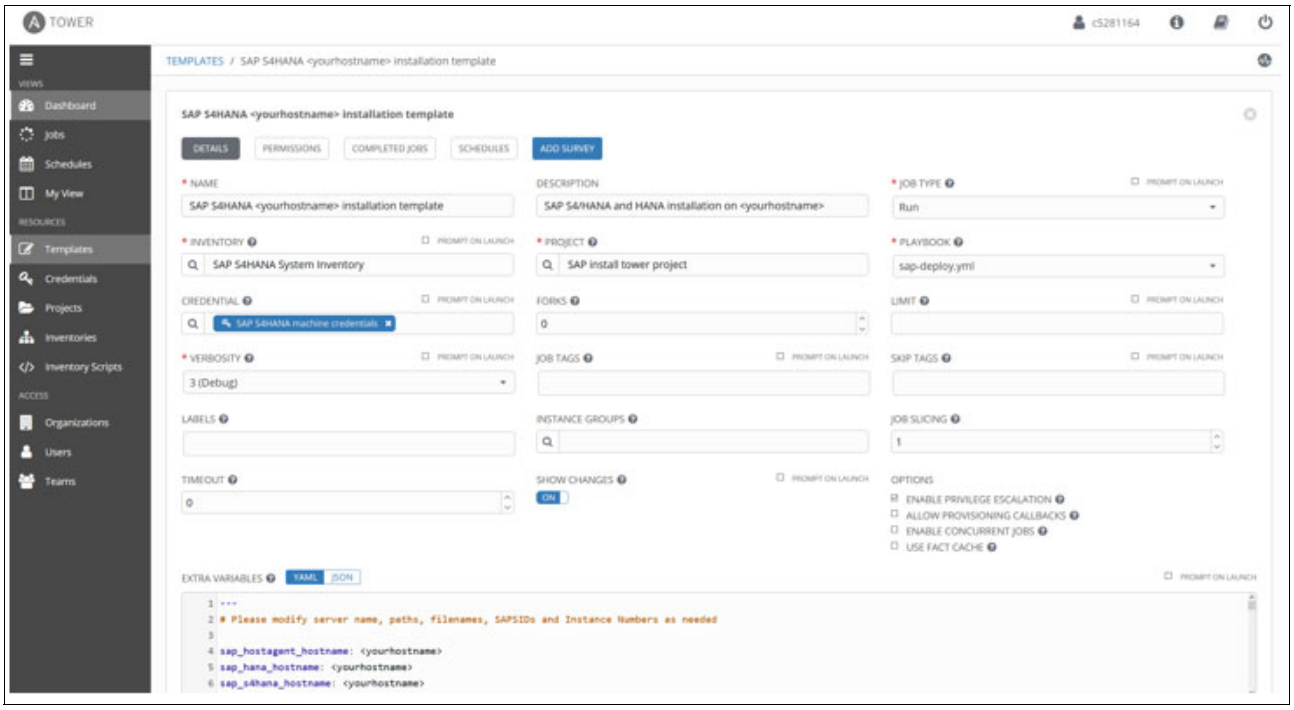


Figure 3-8 Red Hat Ansible Tower: Job template window

2. Complete the required and optional fields:
   - **Template Name**: SAP S4HANA *<yourhostname>*.
   - **Template Description**: SAP S4/HANA and HANA Installation on *<yourhostname>*.
   - Select **Run** as the **Job Type**.
   - Select **SAP S4HANA System Inventory** from the **Inventory** drop-down list.
   - Select **SAP install tower project** from the **Project** drop-down list.
   - Select **sap-deploy.yml** from the **Playbook** drop-down list.
   - Select **SAP S4HANA machine credentials** for your machine credentials.
   - A verbosity of **3 (Debug)** can be initially helpful and can be later set to **0 (Normal)**.
3. There are many more settings like permissions and notifications, that can be added if required. In the **Extra Variables** field, add the host variables as key-value pairs by using a YAML syntax.

   For demonstration purposes, we do not encrypt our sensitive content like usernames and passwords. In sensitive environments, passwords can be managed by an encrypted `ansible-vault` file. Ansible Vault provides an easy way to encrypt a string or file. For more information, see Encrypting content with Ansible Vault.

The job template variables for the SAP software installation are defined in Example 3-8.

Example 3-8 Job template variable definitions

```
---
# Modify server name, paths, file names, SAPSIDs, and Instance Numbers as
needed

sap_hostagent_hostname: <yourhostname>
sap_hana_hostname: <yourhostname>
sap_s4hana_hostname: <yourhostname>

# SAP instance installation parameters
# the following two lines must be changed in sync with two lines below:
#    sap_hana_deployment_hana_sid = sap_s4hana_deployment_db_sid
#    sap_hana_deployment_hana_instance_number =
sap_s4hana_deployment_hana_instance_nr
sap_hana_deployment_hana_sid: "ABD"
sap_hana_deployment_hana_instance_number: "20"

# Variables required for `sap_preconfigure` role
sap_preconfigure_selinux_state: permissive
# If you need to modify your hostnames set up it as true
sap_preconfigure_modify_etc_hosts: false
# define the SAP domain name only if you set
'sap_preconfigure_modify_etc_hosts: true'
# sap_domain: "yoursubdomain.enterprise-domain-name.com"

#Common variables that are required for sap hostagent role
sap_hostagent_installation_type: "sar"
sap_hostagent_sar_local_path: "/data/installer/S4HANA1909FPS01"
sap_hostagent_sar_file_name: "SAPHOSTAGENT46_46-70002261.SAR"
sap_hostagent_sapcar_local_path: "/data/installer/S4HANA1909FPS01"
sap_hostagent_sapcar_file_name: "SAPCAR.EXE"
sap_hostagent_clean_tmp_directory: true

#Defining specific variables to be used for SAP HANA database deployment
sap_hana_deployment_bundle_path: "/data/installer/S4HANA1909FPS01"
sap_hana_deployment_bundle_sar_file_name: "IMDB_SERVER20_047_0-80002046.SAR"
sap_hana_deployment_sapcar_path: "/data/installer/SAPCAR"
sap_hana_deployment_sapcar_file_name: "SAPCAR.EXE"
sap_hana_deployment_root_password: "XXpasswd1"
sap_hana_deployment_sapadm_password: "XXpasswd1"
sap_hana_deployment_hana_env_type: development
sap_hana_deployment_hana_mem_restrict: "n"
sap_hana_deployment_common_master_password: "XXpasswd3"
sap_hana_deployment_sidadm_password: "XXpasswd1"
sap_hana_deployment_hana_db_system_password: "XXpasswd2"
sap_hana_deployment_ase_user_password: "XXpasswd4"
sap_hana_deployment_apply_license: false

#Variables to be used for S/4HANA deployment

sap_s4hana_deployment_product_id: "NW_ABAP_OneHost:S4HANA1909.CORE.HDB.ABAP"
sap_s4hana_deployment_sapcar_path: "/data/installer/SAPCAR"
sap_s4hana_deployment_sapcar_file_name: "SAPCAR.EXE"
```

```
sap_s4hana_deployment_swpm_path: "/data/installer/S4HANA1909FPS01"
sap_s4hana_deployment_swpm_sar_file_name: "SWPM20SP05_5-80003426.SAR"

sap_s4hana_deployment_sid: "AB1"
sap_s4hana_deployment_ascs_instance_nr: "21"
sap_s4hana_deployment_pas_instance_nr: "22"
sap_s4hana_deployment_db_host: "<yourhostname>"
# The following two lines must be changed in sync with two lines below:
#    sap_hana_deployment_hana_sid = sap_s4hana_deployment_db_sid
#    sap_hana_deployment_hana_instance_number =
sap_s4hana_deployment_hana_instance_nr
sap_s4hana_deployment_db_sid: "ABD"
sap_s4hana_deployment_hana_instance_nr: "20"

sap_s4hana_deployment_db_schema_password: "XXpasswd"
sap_s4hana_deployment_db_schema_abap_password: "XXpasswd"
sap_s4hana_deployment_master_password: "XXpasswdM"
sap_s4hana_deployment_hana_systemdb_password: "xxPasswd"
sap_s4hana_deployment_hana_system_password: "xxSystemPsw"
sap_s4hana_deployment_parallel_jobs_nr: "30"
sap_s4hana_deployment_db_sidadm_password: "yourPasswd"
sap_s4hana_deployment_igs_path: "/data/installer/S4HANA1909FPS01"
sap_s4hana_deployment_igs_file_name: "igsexe_10-80003246.sar"
sap_s4hana_deployment_igs_helper_path: "/data/installer/S4HANA1909FPS01"
sap_s4hana_deployment_igs_helper_file_name: "igshelper_17-10010245.sar"
sap_s4hana_deployment_kernel_dependent_path: "/data/installer/S4HANA1909FPS01"
sap_s4hana_deployment_kernel_dependent_file_name: "SAPEXEDB_100-80004417.SAR"
sap_s4hana_deployment_kernel_independent_path:
"/data/installer/S4HANA1909FPS01"
sap_s4hana_deployment_kernel_independent_file_name: "SAPEXE_100-80004418.SAR"
sap_s4hana_deployment_software_path: "/data/installer/S4HANA1909FPS01"
sap_s4hana_deployment_sapadm_password: "spAdmpass1"
sap_s4hana_deployment_sap_sidadm_password: "spAdmpass2"
```

4. As a best practice, check all variables in advance to prevent a time-consuming error investigation if running with the template fails.

Attention: When you check variable content, pay close attention to verifying and modifying these settings:

- Hostname: The *<youhostname>* variable should match your target hostname.
- File name and paths: Depending on the SAP installation software package, the software version and local storage location on your target machine are likely to change.
- `SAPCAR`: Verify that a copy that is named `SAPCAR.EXE` of the `SAPCAR` file is stored in the `sapcar_path` directory.
- SAP SIDs and instance numbers: Matched to your needs.
- Passwords: Change the example passwords.

For more information about variable precedence in Red Hat Ansible Tower (if you already used them in Red Hat Ansible CLI), see Extra Variables.

5. After you finish configuring the job template, click **Save**. Click **Launch** to run the job, as shown in Figure 3-9.

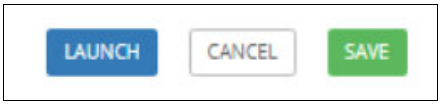


Figure 3-9 Running the job

If the job run is successful, a green color is shown. The **Completed Jobs** view shows the list of all job templates that complete. From this tab, you can also see the job status and detailed information. The **Templates** view provides the list of all defined job templates. If you click the rocket icon, you can restart the job run, as shown in Figure 3-10.

Figure 3-10 Templates view

For more information about using Red Hat Ansible Tower to run playbooks, see Job Template.

3.7 Conclusion

Red Hat Ansible CLI and Red Hat Ansible Tower can speed up your productivity by helping you manage complex processes with job automation and scheduling. Red Hat Ansible Tower provides a dashboard to view every job run and status. You have a GUI and many features to support the automation process. You can also customize Red Hat Ansible Tower for your needs. If you are familiar with CLI tools, then Red Hat Ansible CLI is a perfect solution to use in infrastructure workflows. You can easily integrate Red Hat Ansible Engine with other building tools for continuous integration and deployment of systems.

4

Building and deploying container images with scripts

This chapter contains details about how to build and deploy container images with scripts.

This chapter contains the following topics:

4.1 Introduction

To run an SAP system in a Red Hat OpenShift environment, you must build images from your existing SAP NetWeaver or SAP S/4HANA reference system. The reference system must be a central system, which means that all instances including the SAP HANA database must run on the same host. Distributed or high availability (HA) systems are not supported.

During the build phase, three different images (Init, SAP AppServer, and SAP HANA) are created, as shown in Figure 4-1.

Note: The build logical partition (LPAR) should be different from the cluster helper node. All actions that are described in Figure 4-1 are performed on the build machine, not on the helper node.

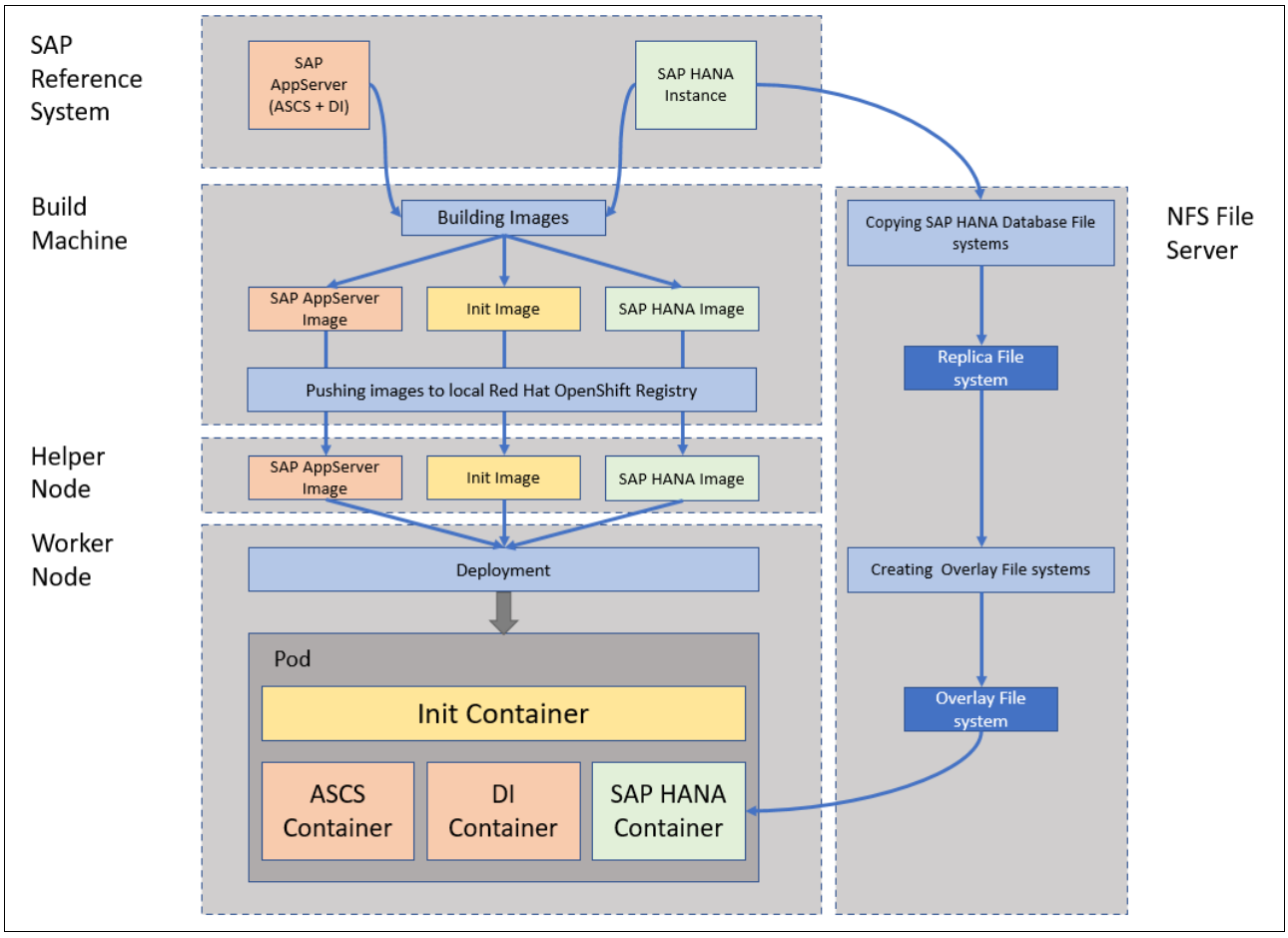


Figure 4-1 Building the reference images

The following list explains Figure 4-1 on page 46 from top to bottom:

- SAP Reference System

  Chapter 3, “Automated installation of SAP S/4HANA and SAP HANA on IBM Power Systems with Red Hat Ansible” on page 25 describes the installation process for the reference system. It contains the SAP AppServer and the HANA database.

- Build Machine

  These three images are built from the reference system:

  - Image: Init
  - Image: SAP AppServer
  - Image: SAP HANA

  The images are created on your build machine. These steps are described in 4.1.1, “The init image” on page 47 to 4.7, “Testing images locally” on page 53.

- Helper Node

  The images are pushed to the Red Hat OpenShift registry that is hosted on the helper node. This step is described in 4.8, “Pushing the images to the Red Hat OpenShift registry” on page 55.

- Worker node

  Four containers are deployed on Red Hat OpenShift in a worker node:

  - Container: init (temporary)
  - Container: SAP HANA
  - Container: Advanced Business Application Programming (ABAP) SAP Central Services (ASCS)
  - Container: Dialog instance (DI)

  The required steps are explained in 4.9, “Deploying container images by using scripts” on page 56.

- NFS File Server (on the right side of Figure 4-1 on page 46)

  The SAP HANA database data and logs are stored on this NFS share. The database copy is described in 4.5, “Building the images by using the scripts from the repository” on page 52.

4.1.1 The init image

The init image is used during the initialization of the deployment in Red Hat OpenShift. It is reference-system-independent and creates the environment setup for the application containers (ASCS container, DI container, and SAP HANA container).

4.1.2 The SAP AppServer image

The SAP AppServer image contains the following directory trees of the original SAP system from which the image is created:

```
/usr/sap/<NWS4-SID>
/usr/sap/trans (only the directory structure without any content)
<sapmnt>/<NWS4-SID> file systems
```

<NWS4-SID> is the image that is used for starting both the ASCS and the DI container.

Note: The SAP HostAgent is not included.

4.1.3 SAP HANA image

The SAP HANA image contains the files of the SAP HANA database instance of the original SAP system. It does *not* contain the `/data/<HDB-SID>` and `/log/<HDB-SID>` directories of the original database (where `<HDB-SID>` is the system ID of the original database).

During the build phase of the images, these two directories must be copied to the replica file system on the NFS server. To make sure that every pod uses its own SAP HANA database content, an overlay file system is created during the deployment.

4.2 Requirements for the build system

This section describes which requirements must be fulfilled before building and deploying images on your build system.

4.2.1 File system for the image build environment

During an image build, files are copied from the original host to the build system. To store this data and the generated images, you need a file system with at least 500 GB. In this section, we assume that this file system is mounted at the `/data` directory.

Two subtrees must be moved from the `root/` file system to the `/data` file system because they are heavily used during the image build process, which might lead to a 100% filled `root/` file system.

As the `root` user, move the `/var/lib/containers` subtree from the `root /` file system to the `/data` file system by running the following commands:

- **`$ mkdir -p /data/var/lib`**
- **`$ mv /var/lib/containers /data/var/lib/containers`**
- **`$ ln -s /data/var/lib/containers /var/lib/containers`**

As the `root` user, move `/var/tmp` subtree from the `root /` file system to the `/data` file system by running the following commands:

- **`$ mkdir -p /data/var/`**
- **`$ mv /var/tmp /data/var/tmp`**
- **`$ ln -s /data/var/tmp /var/tmp`**

4.2.2 Software requirements

For more information about the software requirements, see this GitHub repository.

4.3 Cloning the containerization-for-sap-s4hana code repository

To clone the containerization-for-sap-s4hana code repository, complete the following steps:

1. Log in to your build system.
2. Create a directory under which the `containerization-for-sap-s4hana` code repository will be cloned by running the following command:

   ```
   $ mkdir -p containerization-for-sap-s4hana
   ```

3. Clone the containerization-for-sap-s4hana code repository into your local Git directory by running the following commands:
   - **`$ cd containerization-for-sap-s4hana`**
   - **`$ git clone https://github.com/IBM/containerization-for-sap-s4hana.git`**
   - **`$ cd containerization-for-sap-s4hana`**

4.3.1 Setting up SSH

During the build process, multiple ssh connections are established to the host on which the reference SAP system is installed and to the NFS server. To avoid having to enter the SSH key passphrase or login credentials on each SSH connection start, run the build under a ssh-agent (see ssh-agent - How to configure the forwarding protocol) session or use a passphrase-less SSH key (see Passwordless SSH using public-private key pairs).

4.4 Setting up the Red Hat OpenShift environment for building and deploying

This section describes how to set up Red Hat OpenShift for building and deploying.

4.4.1 Creating a user ID

Create a user ID as described at Configuring an HTPasswd identity provider.

4.4.2 Creating a project by using the Red Hat OpenShift Console

To create a project by using the Red Hat OpenShift Console, complete the following steps:

1. Log in to your Red Hat OpenShift Console.
2. Check that you are in the **Administrator** tab, as shown in Figure 4-2.
3. Click **Projects**.
4. Click **Create Project**.

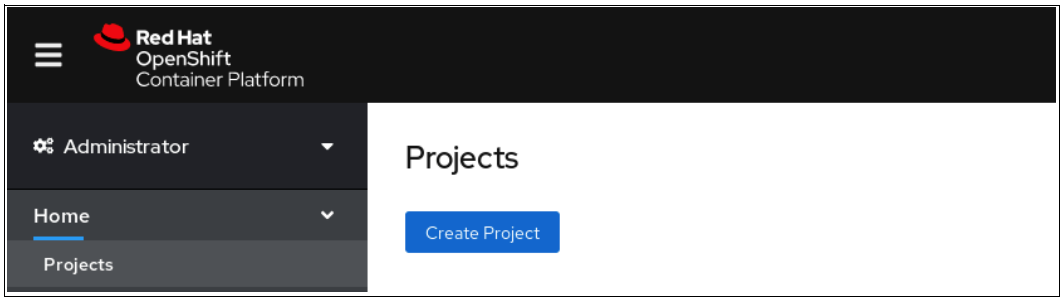


Figure 4-2 Red Hat OpenShift Container Platform: Administrator window

5. Enter a meaningful name for your project and click **Create**, as shown in Figure 4-3.

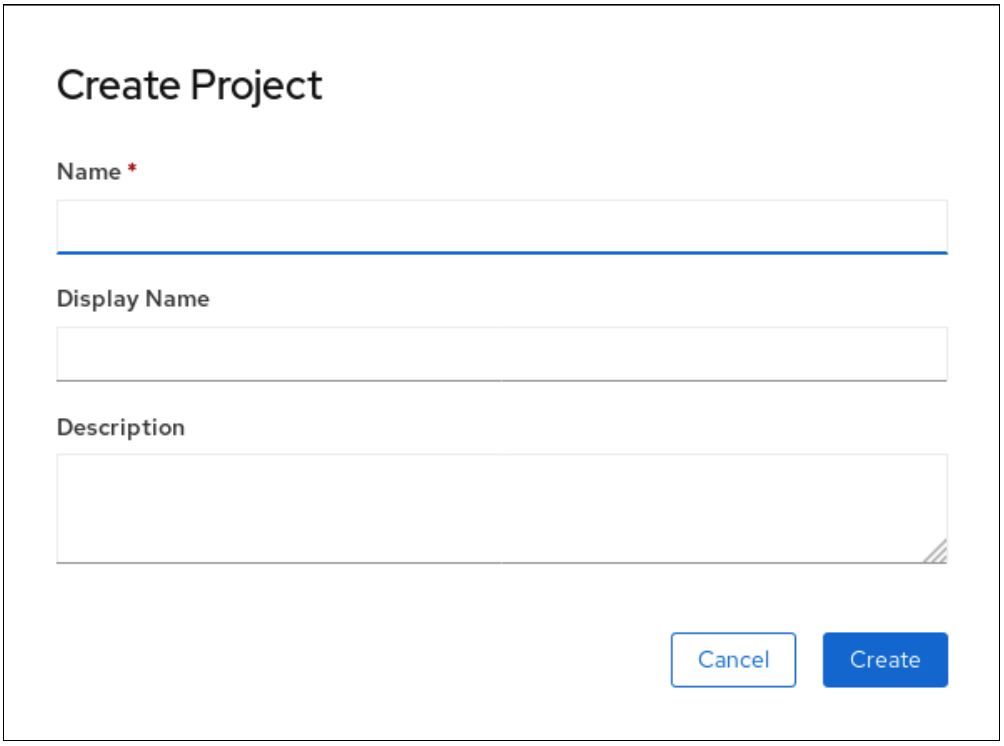


Figure 4-3 Red Hat OpenShift Container Platform: Create Project window

4.4.3 Creating a project by using the Red Hat OpenShift command-line interface

If you do not prefer to use the Red Hat OpenShift Console for creating a project, you can create it by using the command-line interface (CLI) command and running the following command:

```
$ oc new-project <your-project>
```

To switch between existing projects, run the following command:

```
$ oc project <your-project>
```

4.4.4 Retrieving login tokens from the Red Hat OpenShift Console

To retrieve login tokens, complete the following steps:

1. Log in to your Red Hat OpenShift Console.
2. Click your username in the upper right.
3. Click **Copy Login Command**.
4. Log in again with your credentials.
5. Click **Display Token**. Figure 4-4 shows the token details.

Figure 4-4 Token details

Copy the **`oc login --token=`**… command. You can use this command to log in to the Red Hat OpenShift cluster instead of providing a user and a password. Paste the full command and run it on your system, as shown in Example 4-1.

Example 4-1 Running the command with the retrieved token

```
$ oc login --token=Gttn56_1lViGdBoYFqb8NqvqrhVFvmrwRlosCjqG4IM
--server=https://api.ocp4-d07c.soos.ibm.corp:6443
Logged in to "https://api.ocp4-d07c.soos.ibm.corp:6443" as "jaeschke" using the
token provided.

You have access to the following projects and can switch between them with 'oc
project <projectname>':

    jaeschke-soos
  * jaeschke-th1-thd
    jaeschke-thh-hdb

Using project "jaeschke-th1-thd".
```

4.4.5 Obtaining the anyuid Security Context Constraint for your project

Using a CLI, log in to your Red Hat OpenShift Cluster as the cluster administrator, and add the privileged mode to your project by running the following command:

```
$ oc adm policy add-scc-to-group anyuid "system:serviceaccounts:<your-project>"
```

4.4.6 Creating the service account

The SAP HANA container mounts the SAP HANA data and log directories by using NFS. To allow NFS mounting, you must use a service account with the corresponding security context constraints (scc).

To create the service account, complete the following steps:

1. Run the following command:

   ```
   $ tools/ocp-service-account-gen
   ```

   This command generates the following YAML file:

   ```
   $ <ocp-project-name>-service-account.yaml
   ```

2. Create the service account by running the following command:

   ```
   $ oc apply -f <ocp-project-name>-service-account.yaml
   ```

3. Add the required scc to the service account by running the following command:

   ```
   $ oc adm add-scc-to-user hostmount-anyuid \
   system:serviceaccount:<your-project>:<your-project>-sa
   ```

4.4.7 Enabling the default route to the internal Red Hat OpenShift registry

To push images to the internal Red Hat OpenShift registry, you must enable the default route to the registry. For more information, see Enable the Image Registry default route with the Custom Resource Definition.

4.5 Building the images by using the scripts from the repository

Instead of using the Ansible scripts or Ansible Tower, as described in Chapter 5, “Building and deploying container images with Red Hat Ansible” on page 59, you can also use the scripts directly. However, as a best practice, use the automated build and deployment process to build the images and deploy them to Red Hat OpenShift by running the following command from the root directory of your repository clone:

```
$ tools/containerize -a
```

For more information, see Containerization by IBM for SAP S/4HANA with Red Hat OpenShift.

4.6 Deploying with Red Hat OpenShift CLI

This section shows how to deploy with Red Hat OpenShift CLI.

4.6.1 Creating a deployment configuration file

To deploy the images to a Red Hat OpenShift worker node, create a deployment configuration file.

For more information about how you can create a deployment configuration file `<deployment-config-file>` that suits your SAP system setup, see Containerization by IBM for SAP S/4HANA with Red Hat OpenShift.

4.6.2 Starting the deployment

After you create a deployment configuration file, you can now deploy your images by completing the following steps:

1. Log in to your Red Hat OpenShift CLI interface.
2. Check that your previously created `<deployment-config-file>` is accessible, and then run the following command:

   ```
   $ oc apply -f <deployment-config-file>
   ```

   For example:

   ```
   $ oc apply -f jaeschke-soos-deployment-th1-thd.yaml

   service/soos-th1-np created
   deployment.apps/soos-th1 created
   ```

For more information about how to verify whether the SAP system was correctly started, see Containerization by IBM for SAP S/4HANA with Red Hat OpenShift.

4.7 Testing images locally

Before you push the images to your Red Hat OpenShift registry, test them locally. To do so, set up a configuration file, as described at Containerization by IBM for SAP S/4HANA with Red Hat OpenShift.

4.7.1 Testing the SAP HANA image

This section shows how to test the SAP HANA image.

Exporting the replica file system on the NFS server

During the image build phase, a replica file system of the original SAP HANA database is created on the NFS server, either with the Ansible scripts or during the running of the manual steps.

Before using the image locally, you must create an overlay file system by running the following command on your build machine:

```
$ tools/containerize -o
```

The command emits the unique ID (uuid) of the freshly created file system that is used in the next step:

```
<uuid>-<ocp-user-name>-<ocp-project-name>-<HDB-host>-<HDB-SID>
```

Starting the container

You can start the container by running a script that is in `git-repository`:

```
$ tools/container-local -a start -f hdb -u <overlay-uuid>
```

The <*overlay-uuid*> is the unique ID that is obtained during the creation of the replica file system.

The `container-local` script mounts both `/data/<HDB-SID>` and `/log/<HDB-SID>` on local directories and exposes the local directories to the SAP HANA container.

In addition, the `<HDB-SID>-HDB` directory is created in the working directory to hold the `soos-env` file, which is needed during the start of the container.

The script returns the name <*container-name*> of the started container.

Connecting to the container

To connect to the container, log in to it by running the following command:

```
$ podman exec -it <container-name> bash
```

The container name is returned by the `container-local` script or can be gathered by displaying the running containers by running the following command:

```
$ podman ps --filter 'ancestor=localhost/soos-<hdb-sid>:latest' --format
'{{.Names}}'
```

You are now logged in to your container. To check for messages, view the contents of the `/var/log/messages` file.

To check the status of the SAP HANA database, run the **`HDB info`** command as the `<hdb-sid>adm` user.

Stopping the container

Stop the SAP HANA database running within the container before you stop the container itself. You can now stop the container by running the following commands:

- **`$ podman stop <container-name>`**
- **`$ podman rm <container-name>`**

4.7.2 Testing the SAP AppServer image

The easiest way to test the SAP AppServer image is to start an ASCS Container.

Starting the container

You can start the container by running a script that is provided in the `git-repository`:

```
$ tools/container-local -a start -f nws4 -i ascs
```

The script returns the name of the started container.

Connecting to the container

You can now log in to the container by running the following command:

```
$ podman exec -it <container-name> bash
```

The container name is returned by the `container-local` script. You can also view it by displaying the running containers by running the following command:

```
$ podman ps --filter 'ancestor=localhost/soos-<nws4-sid>:latest' --format
'{{.Names}}'
```

You are now logged on to your container. To check for messages, see the `/var/log/messages` file.

To check whether your ASCS instance is running, switch to the `<nws4-sid>adm` user and call **`sapcontrol`**, as shown in Example 4-2.

Example 4-2 Checking the ASCS instance status

```
$ su - <nws4-sid>adm
$ sapcontrol -nr <instNo> -function GetProcessList

24.09.2020 09:16:18
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
msg_server, MessageServer, GREEN, Running, 2020 09 24 09:15:43, 0:00:35, 610
enq_server, Enqueue Server 2, GREEN, Running, 2020 09 24 09:15:43, 0:00:35, 611
```

Note: The SAP host agent is not part of the images.

Stopping the container

You can stop the container by running the following commands:

- **`$ podman stop <container-name>`**
- **`$ podman rm <container-name>`**

4.8 Pushing the images to the Red Hat OpenShift registry

After you test the images locally, you can make them available to your Red Hat OpenShift cluster by pushing the three images to the cluster registry.

To push the images, complete the following steps:

1. Log in to your build system and run the following command from the root repository of your directory clone to push the three images to the local Red Hat OpenShift cluster registry. The push process can take a few minutes.

   ```
   $ tools/containerize -p
   ```

2. Verify whether the images are available in the Red Hat OpenShift cluster registry by running the following command:

   ```
   $ oc get imagestream.image.openshift.io
   NAME             IMAGE REPOSITORY
   soos-init        default-route-openshift-image-registry.apps....
   soos-th1         default-route-openshift-image-registry.apps....
   soos-thd         default-route-openshift-image-registry.apps....
   ```

4.9 Deploying container images by using scripts

After you build the three different images (SAP AppServer, SAP HANA, and Init) and push them to the local registry of your Red Hat OpenShift cluster, you can deploy them.

4.9.1 Introduction

The first time that you deploy the images, they are pulled from the Red Hat OpenShift cluster registry to one of your worker nodes. You can check the progress of the deployment by running the **`oc describe`** command.

The Init container

First, the Init container runs. Init containers are special containers that run before the other containers (App containers) start. The Init container runs a shell script that reads the environment variables that are specified in the deployment configuration file and creates the environment files for the different containers.

We differentiate five kinds of environment variables:

Prefix SOOS_GLOBAL	Used for all containers.
Prefix SOOS_NWS4	Used for both the ASCS and the DI containers.
Prefix SOOS_ASCS	Used for the ASCS container only.
Prefix SOOS_DI	Used for the DI container only.
Prefix SOOS_HDB	Used for the HDB container only.

The environment files are created in three different working directories:

`/envdir-ascs`	Mounted to the ASCS container.
`/envdir-di`	Mounted to the DI container.
`/envdir-hdb`	Mounted to the HDB container.

Important: Do not change the names of the working directories in your deployment configuration file.

App containers

The App containers (ASCS, DI, and SAP HANA containers) are started in parallel when the running of the Init container finishes.

SAP HANA container

During the startup of the SAP HANA container, the overlay file systems from the NFS server are mounted under the container's file system.

The SAP HANA DB instance directory `/usr/sap/<HDB-SID>/HDB<HDB-instNo>` is generated during container startup by running the **`hdblcm`** command. Then, the SAP HANA DB starts.

ASCS container

During the startup of the ASCS container, first the ASCS instance `exe` directory `/usr/sap/<NWS4-SID>/ASCS<ASCS-InstNo>/exe`, is created. Then, the SAP service is created. Finally, the ASCS instance starts.

Dialog instance container

During the startup of the DI container, the DI instance `exe` directory `/usr/sap/<NWS4-SID>/DI<DI-InstNo>/exe` is created, and then the SAP service starts.

The container waits until you can access the SAP HANA DB instance by running the **R3trans -d** command, and then the container starts the DI.

For more information about how to operate the containers, see Chapter 6, “Operating the containers” on page 67.

5

Building and deploying container images with Red Hat Ansible

This chapter provides how to build and deploy container images with Red Hat Ansible.

This chapter contains the following topics:

5.1 Requirements for Red Hat Ansible

This section describes the Red Hat Ansible requirements.

5.1.1 Directory for the image build environment

During the image build process, files are copied from the original host to the build system. To store this data and the generated images, you need a file system with at least 500 GB. We assume in the following chapter that this file system is mounted at the `/data` directory.

Two subtrees must be moved from the `root /` file system to the `/data` file system because they are used heavily during the image build process, which might lead to 100% of the `root /` file system being used, which is unwanted.

To move the `/var/lib/containers` subtree from the `root /` file system to the `/data` file system, run the following commands as the `root` user:

- **`$ mkdir -p /data/var/lib`**
- **`$ mv /var/lib/containers /data/var/lib/containers`**
- **`$ ln -s /data/var/lib/containers /var/lib/containers`**

To move the `/var/tmp` subtree from the `root /` file system to the `/data` file system, run the following commands as the `root` user:

- **`$ mkdir -p /data/var/`**
- **`$ mv /var/tmp /data/var/tmp`**
- **`$ ln -s /data/var/tmp /var/tmp`**

5.1.2 Cloning the containerization-for-sap-s4hana code repository

To clone the containerization-for-sap-s4hana code repository, complete the following steps:

1. Log in to your build system.
2. Create a directory under which the `containerization-for-sap-s4hana` code repository will be cloned by running the following command:

   ```
   $ mkdir -p containerization-for-sap-s4hana
   ```

3. Clone the containerization-for-sap-s4hana code repository into your local Git directory by running the following commands:
   - **`$ cd containerization-for-sap-s4hana`**
   - **`$ git clone https://github.com/IBM/containerization-for-sap-s4hana.git`**
   - **`$ cd containerization-for-sap-s4hana`**

5.1.3 Setting up ssh

During the build process, multiple ssh connections are established to the host on which the reference SAP system is installed and to the NFS server. To avoid having to enter the SSH key passphrase or login credentials on each SSH connection start, run the build under a ssh-agent (see ssh-agent - How to configure the forwarding protocol) session or use a passphrase-less SSH key (see Passwordless SSH using public-private key pairs).

5.1.4 Providing an IP route from the build server to the helper node

You can perform most of the actions that are described in the following sections on your development machine. To log in to the Red Hat OpenShift cluster and connect to the local registry of the cluster, add the following lines to the `/etc/hosts` file of your development machine:

```
<helper-node-ip> api.<ocp-cluster-domain>
oauth-openshift.apps.<ocp-cluster-domain>
default-route-openshift-image-registry.apps.<ocp-cluster-domain>
```

5.2 Building with Red Hat Ansible

This section describes how to build SAP HANA and SAP S/4HANA container images before running them on the Red Hat OpenShift Container Platform. In this example, we build three images: Init, SAP AppServer SID, and SAP HANA SID, as described in Chapter 4, “Building and deploying container images with scripts” on page 45, by using scripts. The Ansible command-line interface (CLI) helps to automate the building process of all three images. Before starting with Ansible CLI, make sure that all required software packages are installed.

In your cloned GitHub repository, there is a directory that is named `ansible` that has the following structure:

```
|__ansible
   |__roles
   |__tasks
   |__vars
   |__ocp-deployment.yml
```

- The directory that is called `roles` has reusable Ansible playbooks that will be included in the `ocp-deployment.yml` playbook to deploy SAP HANA and SAP S/4HANA. Each role includes a set of related tasks to organize them more efficiently. There are roles for checking general and OpenShift prerequisites; copying SAP HANA to the NFS server; building images, pushing images, and creating an SAP HANA overlay share; and starting deployment.
- The `tasks` directory has files that are reused more than once in playbooks. There are tasks like prerequisites for Red Hat Enterprise Linux 8.x, log in as a user in the OpenShift cluster, log in as admin in the OpenShift cluster, and installing the GNU GCC-compiler and GNU Make utilities and other packages that are needed for the Paramiko SSH client. Defined roles include task files within playbooks. You can extend tasks by defining one to customize your system requirements.
- The `vars` directory is for extra variables and contains a file with default variables, which are used in all playbooks. You can name it `<your-extra-vars>.yml` and specify your variables as key-value pairs. The variables will be included in roles and used multiple times. They are referenced by using the Jinja2 syntax as double curly braces.
- The `ocp-deployment.yml` file is a main playbook that contains one play with included roles.

Roles have the following directory structure:

```
|__roles
       |__os-prerequisites
       |__ocp-prerequisites
       |__copy-hdb-nfs
       |__build-images
       |__push-images
       |__create-overlay-share
       |__deploy-images
```

Each role contains the tasks/main.yml file, where our list of tasks that the role runs are defined. The roles have different functions:

- The `os-prerequisites` role installs those packages as Pod Manager tool (`podman`), `git`, `python3`, `python3-devel`, and Paramiko, and includes tasks for Red Hat Enterprise Linux 8.x to install more requirements. The role checks the connection to the Red Hat OpenShift cluster and to the default route to the image registry, and it verifies whether the local OpenShift client tool exists. The role also verifies the NFS server connections and generates a `config.yaml` file from a file template, and then the script verifies whether all input variables in the `config.yaml` file are valid.
- The `ocp-prerequisites` role ensures that the prerequisites are met for image pushing and deployment on the Red Hat OpenShift Container Platform. The role verifies and then sets up a new project, and then checks whether the default route to the internal registry of the Red Hat OpenShift cluster is enabled. It also sets up permissions to run containers in the defined project and generates a file for a service account.
- The `copy-hdb-nfs` role creates a snapshot copy of your SAP HANA data and log directories on the NFS server. Check that your SAP HANA is stopped before running this role. Before running this role, you might need to copy the SSH key of the NFS server to your build host by running the following command:

  ```
  ssh-copy-id -i ~/.ssh/<nfs_rsa_key>.pub <user_name>@<build_host_name>
  ```

- The `build-images` role runs the image build process for your SAP HANA and SAP S/4HANA instances. The three images will be built and stored in the local `podman` registry on the build machine.

The `vars` directory has a file with variables that can contain sensitive content like IP addresses, passwords, and usernames. Therefore, use the Ansible Vault utility to protect your content by encrypting it. To keep sensitive information hidden in a playbook when using verbose output, add the `no_log` attribute to a playbook at the beginning. We do not show how to use Ansible Vault because of its complexity. For more information about Ansible Vault, see Encrypting content with Ansible Vault.

Roles and tasks make playbooks reusable to avoid duplication of source code. The main playbook `ocp-deployment.yml` includes all roles for building images, and it has the following directory structure:

```
---
- hosts:
  roles:
  - os-prerequisites
  - ocp-prerequisites
  - copy-hdb-nfs
  - build-images
```

Before running this playbook, you must set up your inventory. In the `ansible` directory, define the file hosts and add the name of your remote machine. Then in the `ansible` directory, create the `host_vars` directory and define a file name the same as your remote machine in the hosts; it looks like `<your_build_server>.yml`, and you add the remote username and SSH key:

```
---
ansible_user: root
ansible_ssh_private_key_file: ~/.ssh/<your_rsa_key>
```

In the `<your_build_server>.yml` file, you can define other configuration parameters that are needed to connect to your remote host. After this task is done, the `ansible` directory is organized as follows:

```
|__ansible
         |__hosts
         |__host_vars/<your_build_server>.yml
         |__roles
         |__tasks
         |__vars
         |__ocp-deployment.yml
```

Run the `ocp-deployment.yml` playbook by passing variables at the CLI by using the **`-e`** option for extra variables. Run your Ansible playbook by running the following command:

```
ansible-playbook -i hosts -e @vars/ocp-extra-vars.yml ocp-deployment.yml
```

After running the `ocp-deployment.yml` playbook, the prerequisites are installed and three images are created: Init, SAP AppServer SID, and SAP HANA SID.

5.3 Deploying with Red Hat Ansible

Section 5.2, “Building with Red Hat Ansible” on page 61 describes how to build SAP HANA and SAP S/4HANA images. This section describes how to deploy the three images into the Red Hat OpenShift cluster. For our deployment, we use these roles:

- The `push-images` role runs the script that pushes the three images from the local registry to your Red Hat OpenShift cluster.
- The `create-overlay-share` role creates an SAP HANA DB overlay share on the NFS server. This overlay share is used by the SAP HANA container.
- The `deploy-images` role generates a deployment file that contains all information about your setup and environment for containers running in your Red Hat OpenShift cluster. The complete `ocp-deployment.yml` has all the roles, and it has the following directory structure:

  ```
  ---
  - hosts:
    roles:
    - os-prerequisites
    - ocp-prerequisites
    - copy-hdb-nfs
    - build-images
    - push-images
    - create-overlay-share
    - deploy-images
  ```

Comment out the roles that already were used. Run the playbook by using these roles as `push-images`, `create-overlay-share`, and `deploy-images`, and add the **-e** option for extra variables as follows:

```
ansible-playbook -i hosts -e @vars/ocp-extra-vars.yml ocp-deployment.yml
```

5.4 Building and deploying with Red Hat Ansible Tower

To start with Red Hat Ansible Tower, see 3.6.1, “Starting with Red Hat Ansible Tower” on page 36. To build SAP HANA and SAP S/4HANA images with Red Hat Ansible Tower, you must configure your inventory by adding a build host and credentials for an SSH connection, set up your project and job template, and define extra variables.

To build and deploy with Red Hat Ansible Tower, complete the following steps:

1. You must have a project that will be used in a job template for building and deploying images, so you must either define one or choose an existing project directly in the job template.

   To set up a new project, log in to the Red Hat Ansible Tower web GUI with Administrator user authority and click **Projects** in the left menu. You see a list of available projects. To get a new project, click the + at the upper right and complete the required fields:

   a. Define a project name.

   b. Add a description.

   c. Select an organization. For this example, you can use **Default**.

   d. For the SCM TYPE, copy the URL link of the GitHub repository where the Ansible playbooks are stored.

   e. Input the scm branch to check out source code. For this example, you can use `master`.

   f. Select the SCM UPDATE OPTIONS check boxes, such as **CLEAN**, **DELETE ON UPDATE**, and **UPDATE REVISION ON LAUNCH**.

   You do not need credentials for an open-source GitHub repository because the provided URL where all scripts are stored is public, and you can copy the URL into the **SCM URL** field of the Projects template, as shown in Figure 5-1.

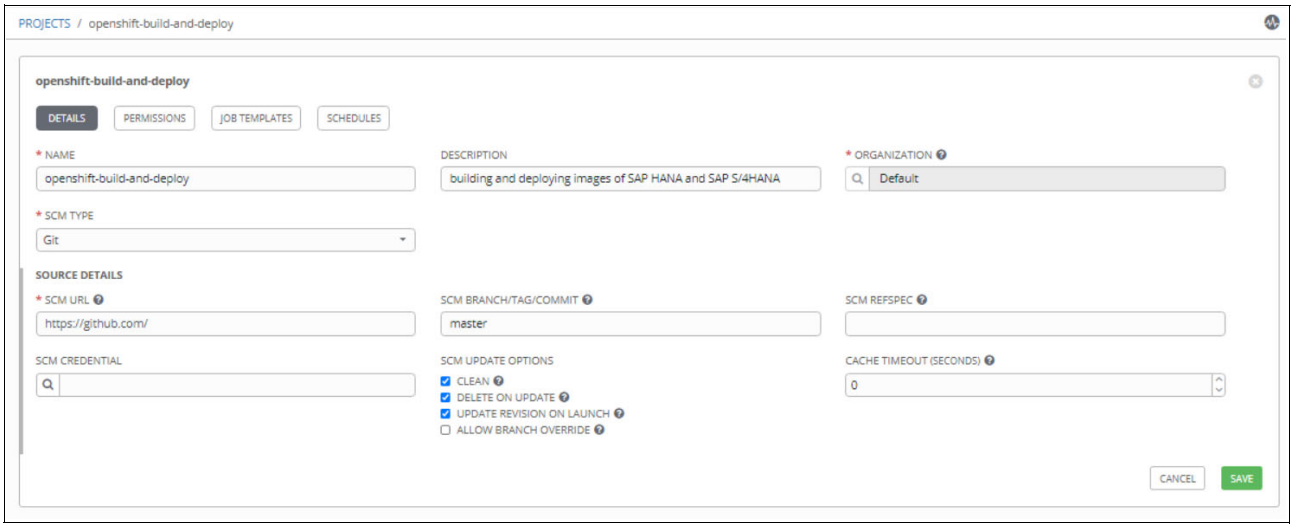


Figure 5-1 GitHub project details view

2. Next, click Templates in the left menu (Figure 5-1 on page 64) and click the **+** at the upper right, as shown in Figure 5-2.

Figure 5-2 Credentials window

3. A new job template opens where you can complete required and optional fields, as described in 3.6.7, “Defining a job template” on page 41. Before completing the job template, check whether you have a defined inventory, as described in 3.6.5, “Setting up inventory” on page 39 and 3.6.6, “Setting up target host credentials” on page 40. In the **Extra Variables** field, add the specified variables from the file in the `vars` directory. In the **Playbook** field, select the playbook that is defined for Red Hat Ansible Tower deployment. It is also inside the GitHub `ansible/` directory.
4. Save the job template for building and deployment, and then start the job. If the job run is successful, it has a green status, which means that the building and deployment of the SAP HANA and SAP S/4HANA images successfully completed.

6

Operating the containers

This chapter describes how to operate and manage the containers.

This chapter contains the following topics:

6.1 Checking the status of containerized SAP instances

This section provides information about checking the status of your containerized SAP instances.

6.2 Checking the status of the pod

After you apply your deployment, check whether all containers are running. You must wait for the different containers to complete their logins to see whether the pod is running.

You can check the status of your pod by running the following command:

```
tools/ocp-pod-status
```

The output looks like the following string:

```
Status of Pod soos-<nws4-sid>: Running
```

If the status of the pod is `Running`, the Pod is running. In all other cases, the containers might still be in the startup phase or an error occurred.

For more information about how to check the status of your SAP system in your Red Hat OpenShift cluster, see Containerization by IBM for SAP S/4HANA with Red Hat OpenShift.

6.3 Accessing containers

Your SAP system is running in one pod but in different containers. To get access to the shell of one of your containers, run the following command:

```
tools/ocp-container-login
usage: ocp-container-login [-h]
                           [-v {critical,error,warning,info,debug,notset}]
                           [-w] [-g <logfile-dir>] [-c <config-file>]
                           [-f <flavor>] [-i <nws4_instance_type>]
```

If you want to log in to your SAP HANA container, run the following command:

```
tools/ocp-container-login -f hdb
```

You are now logged on to your SAP HANA container.

Note: Red Hat OpenShift terminates an interactive connection to the container automatically after a period of inactivity.

6.4 Connecting with SAP GUI to your containerized SAP system

To connect to your containerized SAP system, you must create an SSH port forwarding tunnel from the machine on which your SAP GUI is running to the worker node on which the pod is running.

To get the ssh command, running the following command:

```
tools/ocp-port-forwarding
```

Use this command from the machine on which your SAP GUI runs to establish port forwarding, as shown in Example 6-1.

Example 6-1 Establishing the SSH port forwarding tunnel

```
$ ssh -L 3200:56.21.50.60:31200 jaeschke@lsv3064.ibm.com
Password:
Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

Last failed login: Fri Sep 25 07:12:23 UTC 2020 from 56.76.112.114 on ssh:notty
There were 2 failed login attempts since the last successful login.
Last login: Thu Sep 24 10:32:41 2020 from 56.76.112.114
```

Note: If your SAP GUI is running on Windows, do not use the Power Shell for establishing the SSH port forwarding tunnel, but instead use tools like MobaXterm or Cygwin.

Create a connection in your SAP GUI with the following parameters:

System ID `<nws4-sid>`
Instance Number `<instno>`
Application Server `<build-machine-name>`

- `<nws4-sid>` is the SAP system ID of your reference SAP NetWeaver or SAP S/4HANA system.
- `<instno>` in general corresponds to the instance number of the dialog instance (DI) of your reference system. It might differ if the required port on the build machine is taken by another application.
- `<build-machine-name>` is the name of your build machine.

6.5 Restarting the SAP workload

If you want to restart the SAP system, you can either log in to the containers and restart the instances by using SAP tools, or you can restart the pod containing the SAP system by completing the following steps:

1. Log in to the Red Hat OpenShift Console.
2. Check that you are in the Administrator view, as shown in Figure 6-1.
3. Select **Workloads** → **Pods**.
4. Select the project that contains the pod that you want to restart.

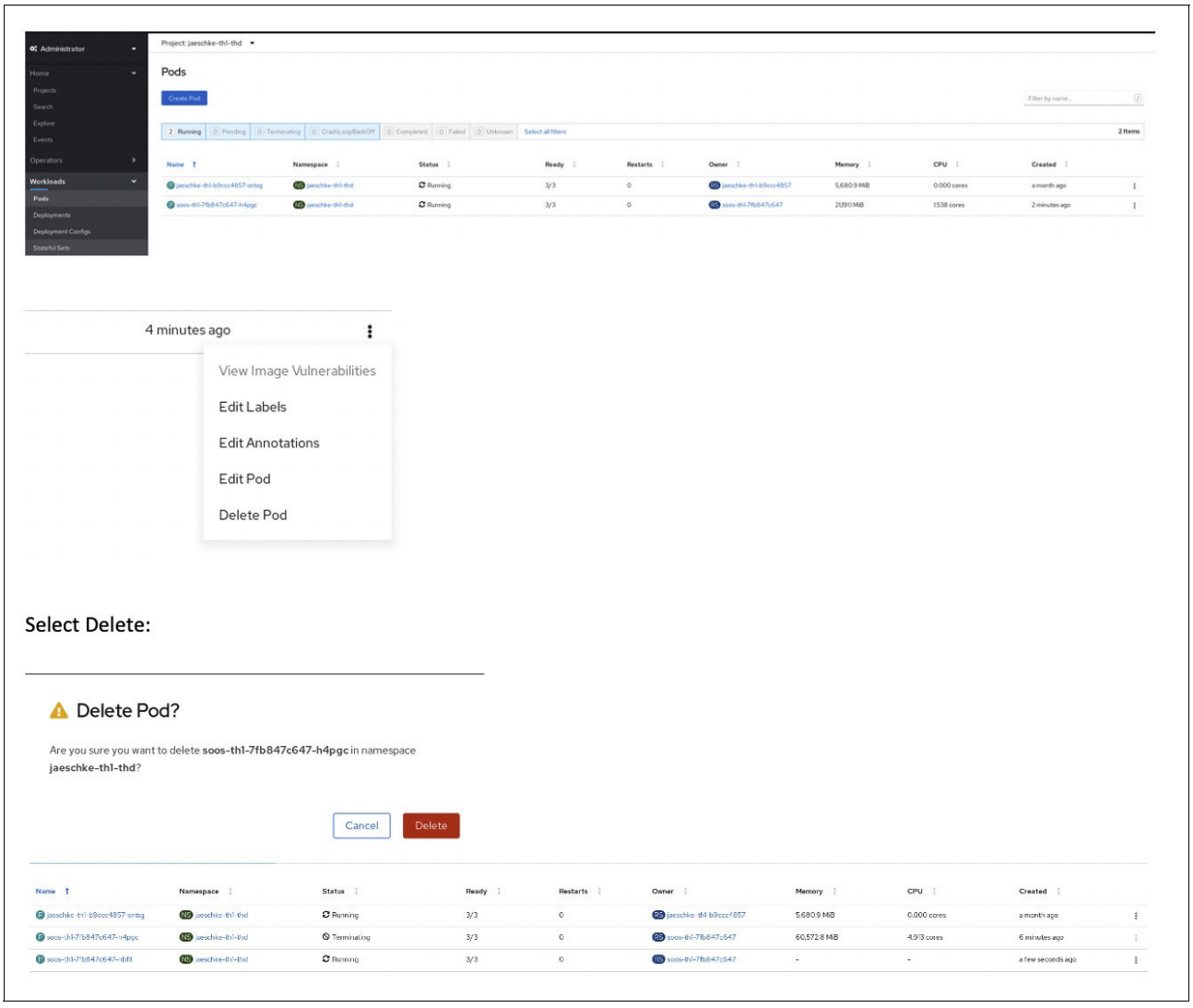


Figure 6-1 Red Hat OpenShift Console: Administrator view window

Taking a closer look to the pod list, you recognize that a new pod is automatically started at the same time the old one is terminating, as shown in Figure 6-1.

Here is what happens to the SAP system inside the pod when you stop the pod: There are changes to the SAP DIs. For example, changes in the profiles do not persist. Considering all the changes that you made to the SAP HANA database content are stored in the overlay file system, they are persistent if you do not tear down the overlay file system.

6.6 Deleting the SAP workload

If you want to stop the SAP system and prevent it from restarting, you can easily scale down the number of pods to zero by completing the following steps:

1. Select the Developer view, as shown in Figure 6-2.

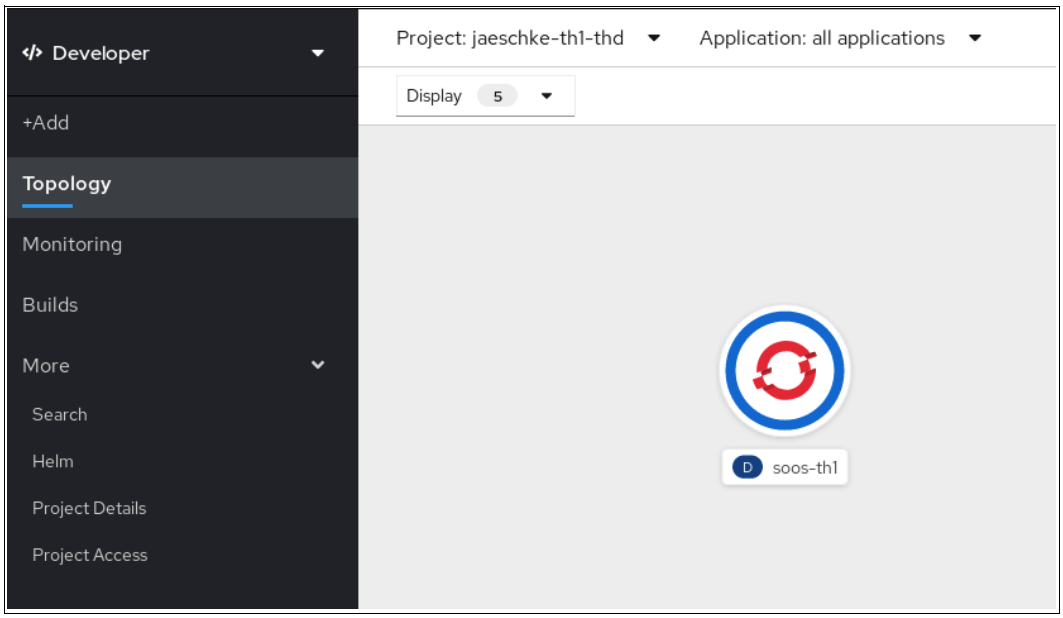


Figure 6-2 Developer view window

2. Click the pod, as shown in Figure 6-3.

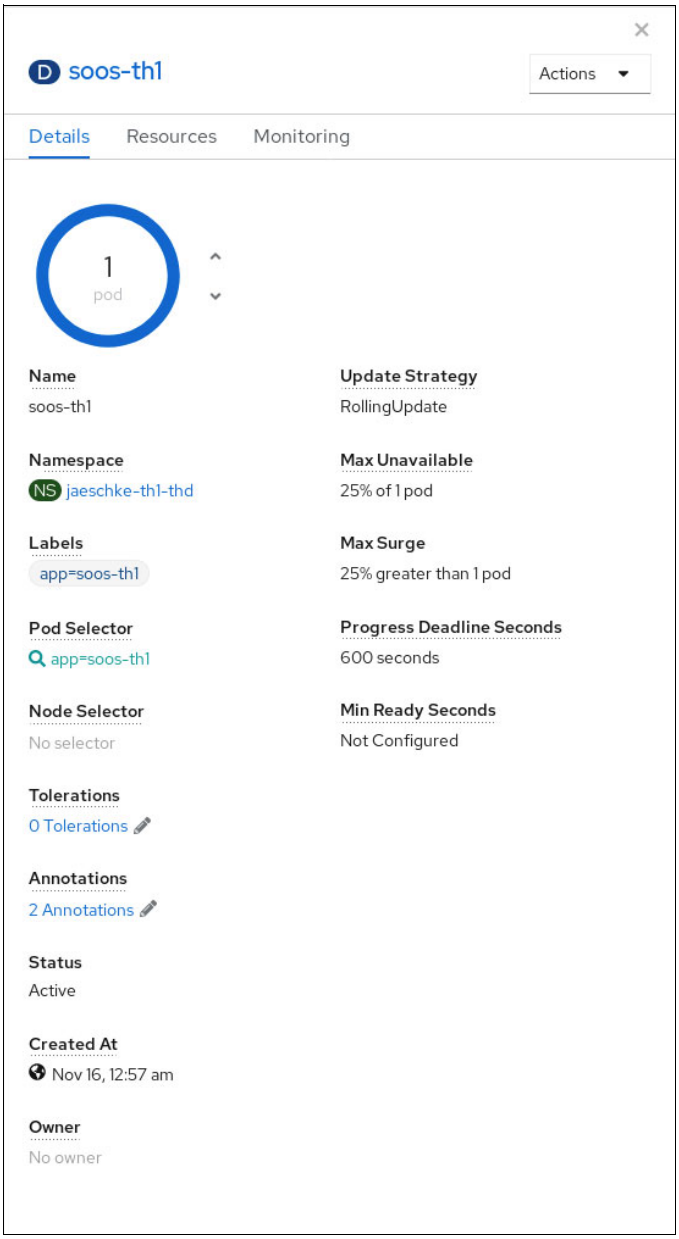


Figure 6-3 Selected pod

3. To scale down the number of running pods to zero, click the down arrow near the number of pods. The pod stops, and no restart is initiated.

   To restart this pod, scale the number of pods to 1. The pod automatically starts again.

Related publications

The publications that are listed in this section are considered suitable for a more detailed description of the topics that are covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide more information about the topics in this document. Some publications that are referenced in this list might be available in softcopy only.

- *Red Hat OpenShift V4.3 on IBM Power Systems Reference Guide*, REDP-5599
- *Red Hat OpenShift V4.X and IBM Cloud Pak on IBM Power Systems Volume 2*, SG24-8486
- *Software Defined Data Center with Red Hat Cloud and Open Source IT Operations Management*, SG24-8473

You can search for, view, download, or order these documents and other Redbooks, Redpapers, web docs, drafts, and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- Ansible Galaxy Repository

  https://galaxy.ansible.com/redhat_sap

- Automating the Installation of SAP S/4HANA and SAP HANA on IBM Power Systems using Red Hat Ansible

  https://blogs.sap.com/2020/11/03/automating-the-installation-of-sap-s-4hana-and-sap-hana-on-ibm-power-systems-using-red-hat-ansible/

- Building and deploying with Red Hat Ansible

  https://github.ibm.com/SAP-OpenShift/containerization-for-sap-s4hana/tree/master/ansible

- Community Roles

  https://github.com/redhat-sap

- Containerization by IBM for SAP S/4HANA with Red Hat OpenShift

  https://github.com/ibm/containerization-for-sap-s4hana

- Installing Red Hat Ansible

  https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

- Red Hat Ansible Tower docs

  https://docs.ansible.com/ansible-tower/latest/html/quickstart/create_job.html

- Red Hat Enterprise Linux System Roles

  https://github.com/linux-system-roles/
- Red Hat OpenShift Container Platform

  https://www.openshift.com/products/container-platform
- Red Hat OpenShift Container Platform 4.6 release notes - IBM Power Systems

  https://docs.openshift.com/container-platform/4.6/release_notes/ocp-4-6-release-notes.html#ocp-4-6-ibm-power
- Red Hat OpenShift Container Platform Life Cycle Policy

  https://access.redhat.com/support/policy/updates/openshift
- SAP Certified and Supported SAP HANA Hardware Directory - IBM Power Systems

  https://www.sap.com/dmc/exp/2014-09-02-hana-hardware/enEN/power-systems.html
- SAP NetWeaver and SAP S/4HANA on Red Hat OpenShift Container Platform

  https://github.ibm.com/SAP-OpenShift/containerization-for-sap-s4hana
- SAP Note 765424 - Linux: Released IBM Hardware - POWER based servers

  https://launchpad.support.sap.com/#/notes/765424
- Supported Linux distributions and virtualization options for POWER8 and POWER9 Linux on Power Systems

  https://www.ibm.com/support/knowledgecenter/en/linuxonibm/liaam/liaamdistros.htm
- Tower Licensing, Updates, and Support

  https://docs.ansible.com/ansible-tower/latest/html/installandreference/updates_support.html
- Using Red Hat Ansible

  https://docs.ansible.com/ansible/latest/user_guide/index.html

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

REDP-5619-00

ISBN 0738459585

Printed in U.S.A.