

Confidential Computing with SUSE Linux Enterprise Base Container Images Using the IBM Hyper Protect Platform

SUSE Linux Enterprise Base Container Images
SUSE Linux Enterprise Server on IBM Z and LinuxONE

Mike Friesenegger, Solutions Architect (SUSE)

Confidential Computing with SUSE Linux Enterprise Base Container Images Using the IBM Hyper Protect Platform

Date: 2025-07-10

Summary

Deploy a workload built with SUSE Linux Enterprise Base Container Images into a hybrid confidential computing environment using IBM Hyper Protect Virtual Servers.

Disclaimer


Documents published as part of the series SUSE Technical Reference Documentation have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.


Contents

1	Introduction	4
2	Prerequisites	5
3	Technical overview	7
4	Preparation	11
5	Building, publishing, and defining	27
6	Deployment	37
7	Managing workloads	44
8	Summary	48
9	Legal notice	49
10	GNU Free Documentation License	50

1 Introduction

1.1 Motivation

Confidential computing focuses on enabling you to secure your data in use. This is accomplished by performing computations in a hardware-based, [trusted execution environment](https://en.wikipedia.org/wiki/Trusted_execution_environment) (https://en.wikipedia.org/wiki/Trusted_execution_environment) . This technology can be deployed in your data centers, in public and private clouds, and even at edge locations. With confidential computing, your workload data is protected no matter where it is running.

SUSE and IBM work together to deliver advanced technical capabilities, like confidential computing. IBM Z® and LinuxONE systems provide key hardware capabilities for the trusted execution environment. [SUSE Linux Enterprise Server on IBM Z and LinuxONE](https://www.suse.com/products/systemz/) (<https://www.suse.com/products/systemz/>)  (SLES) is designed to deliver performance, security, reliability, and efficiency for your mission-critical workloads on IBM Z® and LinuxONE systems.

Container technologies enable enterprises to achieve unprecedented agility, resilience, and scale. Enterprises still need to protect sensitive workload data. Thus, leveraging confidential computing for containerized workloads is essential.

In this guide, you learn how to deploy a containerized confidential computing workload to an IBM Z® and LinuxONE trusted execution environment using a SUSE Linux Enterprise Base Container Images (SLE BCI) and the IBM Hyper Protect Platform.

1.2 Scope

In this guide you:

- learn about the IBM Hyper Protect Platform architecture for on-premises and cloud deployments
- prepare an on-premises or cloud environment for a confidential container workload
- build a confidential container workload with SUSE Linux Enterprise Base Container Images
- deploy the confidential container workload
- verify the confidential container workload

1.3 Audience

This guide can help architects, platform engineers, developers, and operations teams to understand the requirements and processes for deploying containerized workloads into a confidential computing environment.

To be successful with this guide, you should have basic knowledge of container images, Docker Compose, and confidential computing concepts (such as attestation).

1.4 Acknowledgements


Contributions to the development of this guide by the following individuals is appreciated:

- Nicolas Mäding, Senior Product Manager - IBM HyperProtect Platform, IBM
- Dirk Herrendörfer, Architect - HPS Secure Execution on Linux, IBM
- Terry Smith, Director of Global Partner Solutions, SUSE

1.5 Revision history

- 2025-07-10: Add how to use with hardware accelerated crypto devices
- 2025-05-05: Remove deprecated IBM Log Analysis and replace with IBM Cloud Logs
- 2025-02-12: Remove spaces in command to define and start the slebci-paynow-website VM
- 2024-08-17: Fix copy/paste error in rsyslog server.conf
- 2024-01-11: Updated for Node.js 20
- 2023-07-05: Initial publication

2 Prerequisites

You are encouraged to start your journey with [Confidential computing with LinuxONE \(https://cloud.ibm.com/docs/vpc?topic=vpc-about-se#about-hyper-protect-virtual-servers-for-vpc\)](https://cloud.ibm.com/docs/vpc?topic=vpc-about-se#about-hyper-protect-virtual-servers-for-vpc)  using IBM Cloud Hyper Protect Virtual Servers for Virtual Private Cloud (VPC).

The infrastructure for the trusted execution environment needed for HPVS is already set up and available as an easy-to-use service in IBM Cloud. If you want to use IBM Cloud Hyper Protect Virtual Servers for VPC, then all you need is an IBM Cloud [Pay-As-You-Go account \(https://cloud.ibm.com/docs/account?topic=account-accounts#paygo\)](https://cloud.ibm.com/docs/account?topic=account-accounts#paygo).

On-premises confidential computing deployments use [IBM Hyper Protect Virtual Servers \(https://www.ibm.com/products/hyper-protect-virtual-servers\)](https://www.ibm.com/products/hyper-protect-virtual-servers). This is the same technology used in IBM Cloud Hyper Protect Virtual Servers for VPC, but you will need to prepare the required infrastructure. The following high level infrastructure prerequisites are needed to use IBM Hyper Protect Virtual Servers:

- An IBM Z® or LinuxONE system
 - IBM z16 (all models)
 - IBM z15 (all models)
 - IBM LinuxONE 4
 - IBM LinuxONE III
- Feature Code 115 Secure Execution for Linux
- Logical partition (LPAR) running SUSE Linux Enterprise Server on IBM Z and LinuxONE 15 SP5
- If hardware accelerated crypto passthrough will be used
 - IBM z16 or IBM LinuxONE 4 (https://www.ibm.com/docs/en/hpvs/2.1.x?topic=servers-crypto-passthrough#hpcr_prerequisites) is required
 - The Host Key Document (HKD) for the IBM Z or IBM LinuxONE server which was obtained in an earlier step.
 - One or more https://www.ibm.com/docs/en/linux-on-systems?topic=execution-crypto-express-adapters#lxse_ap_pass [supported and properly configured IBM Crypto Express adapters]
SLES will provide the IBM Secure Execution enabled Kernel-based Virtual Machine (KVM) host.
- IBM Hyper Protect Virtual Servers 2.1.x

A [trial program](https://www.ibm.com/docs/en/hpvs/2.1.x?topic=trial-program) (<https://www.ibm.com/docs/en/hpvs/2.1.x?topic=trial-program>) for IBM Hyper Protect Virtual Servers and Crypto Express Network API for Secure Execution Enclaves is available from IBM.

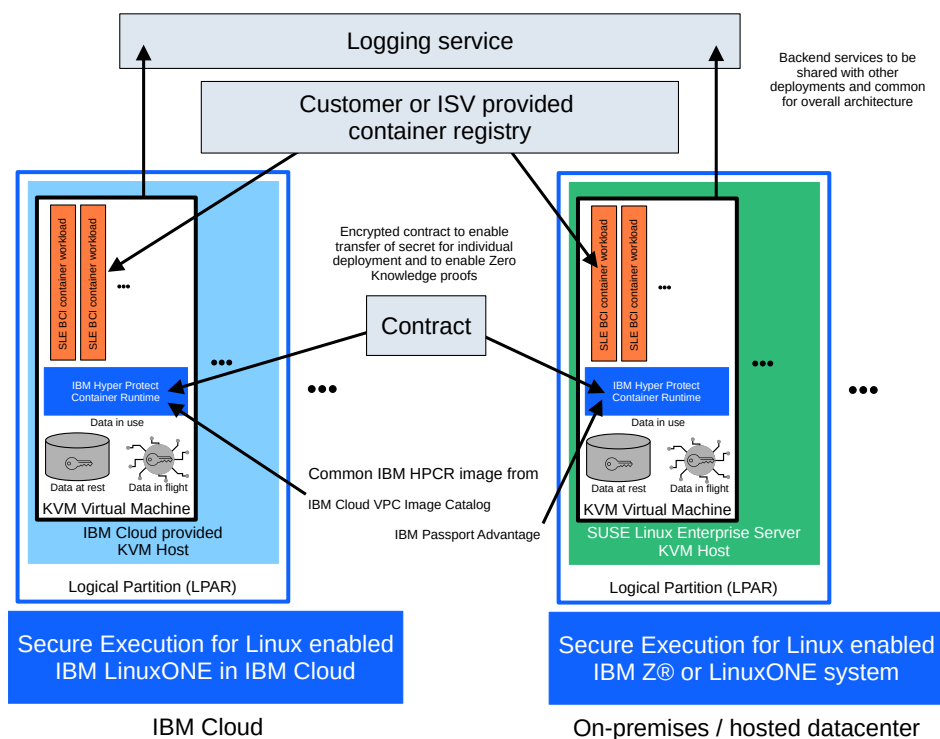
Detailed system requirements can be found in the [IBM Hyper Protect Virtual Servers 2.1.x documentation](https://www.ibm.com/docs/en/hpvs/2.1.x?topic=servers-system-requirements) (<https://www.ibm.com/docs/en/hpvs/2.1.x?topic=servers-system-requirements>).



Note

This guide was developed with the noted software versions. You are encouraged to use the latest releases to take advantage of various updates and security patches.

3 Technical overview



The [IBM Hyper Protect Platform](https://www.ibm.com/downloads/cas/GPVMWPM3) (<https://www.ibm.com/downloads/cas/GPVMWPM3>) architecture is illustrated in this diagram at a high level. On the left are the components that comprise the confidential computing environment in IBM Cloud, on the right are those for on-premises or

hosted datacenter scenarios. The architectural similarities facilitate a hybrid deployment model that gives you the flexibility to easily target the location of your workload depending on business requirements.

The components of the IBM Hyper Protect Platform include:

- [Secure Execution for Linux](https://www.ibm.com/docs/en/linux-on-systems?topic=virtualization-introducing-secure-execution-linux) (<https://www.ibm.com/docs/en/linux-on-systems?topic=virtualization-introducing-secure-execution-linux>) ↗

This z/Architecture® security technology is introduced with IBM z15™ and LinuxONE III. With Secure Execution for Linux, no hardware administrator, KVM code, or KVM administrator can access the data in a KVM virtual machine that was started as an IBM Secure Execution guest.

- [Logical partition](https://www.ibm.com/docs/en/zos/2.4.0?topic=configuration-logical-partitions) (<https://www.ibm.com/docs/en/zos/2.4.0?topic=configuration-logical-partitions>) ↗ (LPAR)

Multiple LPARs can share the resources of a single, physical system.

The KVM host runs in an LPAR, which, in practice, is equivalent to an independent server running its own operating system.



Note

- IBM HPVS for VPC use an IBM provided KVM host.
- SUSE Linux Enterprise Server on IBM Z and LinuxONE can be used for on-premises and hosted data center deployments as a supported KVM host for IBM Hyper Protect Virtual Servers 2.1.x.
- KVM virtual machine
A KVM virtual machine can be started as a Secure Execution guest where memory protection is enforced, preventing unauthorized access to in-memory data.
Multiple KVM virtual machines can be started as Secure Execution guests. If not started as a Secure Execution guest, the virtual machine memory protection is not enforced.
- IBM Hyper Protect Container Runtime (HPCR)

The IBM Hyper Protect Container Runtime (HPCR) provides the environment for containers to be started and run confidentially. The HPCR is a KVM virtual machine QCOW2 disk image file specifically built to start as a Secure Execution guest. The HPCR requires input from a contract to pull and start containerized workloads from a customer or ISV provided container registry.



Note

- The HPCR virtual machine protects data-at-rest by encrypting the root disk and a separate data disk for container workload persistent storage.
- HPCR requires a contract, which is a definition of the workload configuration in YAML format. The HPCR virtual machine will immediately stop when a contract is missing or invalid. The contract is critical and is presented in more detail later in this document.

- Container images

Workload container images can be built on SUSE Linux Enterprise Base Container Images (SLE BCI).

You can run multiple workload containers on a single HPCR Secure Execution guest.

- Logging service

A logging service is required. The HPCR virtual machine will immediately stop if logging is not defined in the contract.

3.1 Workflow

The workflow to prepare, deploy, and manage a confidential container workload in IBM Cloud or on-premises is shown below.

Cloud deployment	On-premises deployment
Prepare for IBM Cloud Hyper Protect Virtual Servers for VPC	Prepare for IBM Hyper Protect Services

Cloud deployment	On-premises deployment
<ul style="list-style-type: none"> • Section 4.1.1, “Creating a Virtual Private Cloud (VPC)” • Section 4.1.2, “Setting up a logging service for HPVS for VPC provisioning” 	<ul style="list-style-type: none"> • Section 4.2.1, “Verifying required hardware and enabling IBM Secure Execution technology” • Section 4.2.2, “Installing and configuring an LPAR with KVM included in SLES for IBM Z and LinuxONE” • Section 4.2.3, “Enabling Secure Execution capabilities on the KVM host” • Section 4.2.4, “Downloading the HPCR image” • Section 4.2.5, “Setting up an rsyslog logging service for HPVS log information” • Section 4.2.6, “Prepare IBM Crypto Express adapters and domains to be passed into a containerized workload”



Build, publish, and define a confidential container workload for the IBM Hyper Protect Platform

- [Section 5.1, “Building a container image”](#)
- [Section 5.2, “Publishing the container image to a registry”](#)
- [Section 5.3, “Defining the contract”](#)
- [Section 5.3.1, “Adding IBM Crypto Express adapters and domains to the contract”](#)



Deploy a confidential container workload using the IBM Hyper Protect Platform

- [Section 6.1, “Cloud deployment”](#)

- [Section 6.2, “On-premises deployment”](#)
- [Section 6.2.1, “Adding IBM Crypto Express adapters and domains to the deployment”](#)



Manage the confidential container workload after deployment

- [Section 7.1, “Enabling and verifying attestation”](#)

4 Preparation

Before you can deploy your confidential container workloads, you must prepare your environment.

Follow the steps detailed in [Section 4.1, “Cloud preparation”](#) or [Section 4.2, “On-premises preparation”](#) depending on the infrastructure environment you intend to use.

4.1 Cloud preparation

The IBM Cloud Web Console is used in this guide, but it is also possible to perform the same actions using the IBM Cloud CLI or Terraform.

4.1.1 Creating a Virtual Private Cloud (VPC)

1. Log in to the [IBM Cloud Web Console \(https://cloud.ibm.com/\)](https://cloud.ibm.com/) [↗](#).
2. In the upper right part of the Web console (between *Manage* and *Help*), select the correct account from the list.
3. Select *Navigation Menu* > *VPC Infrastructure* > *VPCs*.
4. Select the *Region* from the list.
5. Click *Create*.
6. Edit *Name* in the *Details* section (for example, vpc-myvpc).

Do not change default selections for other items.


7. In the *Subnets* section, click *Edit* and rename the dynamically named subnets.
For example, three, regional zones could be named: sn-myvpc-01, sn-myvpc-02, and sn-myvpc-03.
8. Click *Create virtual private cloud*.
9. Select the *Default security group* for the newly created VPC.
 - a. Select *Rules*.
 - b. Click *Create* in the *Inbound rules* section.
 - c. Under *Create inbound rule*:
 - i. Set *Port min* to 8443.
 - ii. Set *Port max* to 8443.
 - iii. Avoid changing any other default settings.
 - iv. Click *Create*.
10. Select *Network > Floating IPs*.
 - a. Confirm the correct *Region* is selected.
 - b. Click *Reserve*.
 - i. Confirm the *Region*.
 - ii. Click *Edit* if you want to select a different *Zone*.
 - iii. Enter the *Floating IP name* (for example, myvpc-paynow-floatip).
 - iv. Click *Reserve*.
11. Make a note of the location (zone) within the region and the floating IP address that was assigned.



Important


Do not close the browser window or tab.

4.1.2 Setting up a logging service for HPVS for VPC provisioning

1. In a new browser window or tab, log in to the [IBM Cloud Web Console \(https://cloud.ibm.com/\)](https://cloud.ibm.com/) .
2. Select *Navigation Menu* > *Observability* > *Logging*.
3. Click *Logging* > *Instances* > *Create*.
4. Complete the following in the Cloud Logs screen:
 - a. In *Select a location*, select the same region that is used for your VPC.
 - b. Under *Select a pricing plan*, select a plan.
 - c. Under *Configure your resource*, set *Service name* (for example, Cloud Logs).
 - d. Click *I have read and agree to the following license agreements*.
 - e. Click *Create*.
 - f. Wait until the provisioning is complete and the Cloud Logs page shows *Active*.
5. Click *Storage* on the side menu in the Cloud Logs page.
 - a. Connect a *Data Bucket* and a *Metrics Bucket*.



Note

Use [Configuring buckets for long term storage and search \(https://cloud.ibm.com/docs/cloud-logs?topic=cloud-logs-about-bucket\)](https://cloud.ibm.com/docs/cloud-logs?topic=cloud-logs-about-bucket)  as a guide for creating and connecting IBM Cloud Object Storage to store your IBM Cloud Logs data for long term storage and search.

6. Click *Overview* on the side menu in the Cloud Logs page.
 - a. Click *Manage* for *Logs Routing*.
 - b. Click *Set target* for the location chosen above and select the Cloud Logs name that was created.
7. Select *Manage* > *Access (IAM)* in the drop-down box in the middle of the IBM Cloud page.



Note

Use [Granting access to IBM Cloud Logs \(https://cloud.ibm.com/docs/cloud-logs?topic=cloud-logs-iam-assign-access&interface=ui\)](https://cloud.ibm.com/docs/cloud-logs?topic=cloud-logs-iam-assign-access&interface=ui) to assign access to IBM Cloud Logs. This guide uses an API key assigned to a service ID as the Access groups method used to assign access to IBM Cloud Logs.

- a. Make a note of the API key that is created which will be used later.
8. Select *Navigation Menu > Observability > Logging*.
9. Click *Logging > Instances*.
10. Select the Cloud Logs instance created.
11. Click *Overview* on the side menu in the Cloud Logs page.
 - a. Verify that the *Logs data store*, *Logs to metrics storage* and *Logs Routing* show a green checkmark with the word "Configured".
12. Click *Endpoints* on the side menu in the Cloud Logs page.
 - a. Make a note of the *Public ingress endpoint* field which will be used later.
13. Click *Dashboard* in the Cloud Logs page which will open a new browser window or tab.









Important

Do not close the browser window or tab.

4.2 On-premises preparation

To simplify the preparation, the information and steps provided in following sections are consolidated from multiple documentation sources. Links to each documentation source are provided.

4.2.1 Verifying required hardware and enabling IBM Secure Execution technology


1. Review the IBM Hyper Protect Virtual Servers 2.1.x Hardware requirements (<https://www.ibm.com/docs/en/hpvs/2.1.x?topic=servers-system-requirements#hardware-requirements>) .
2. Confirm that the Feature Code 115 Secure Execution for Linux (<https://www.ibm.com/docs/en/hpvs/2.1.x?topic=servers-system-requirements#additional-hardware-requirements-for-the-kvm-host>)  has been ordered and installed.
3. Use the machine serial number (<https://www.ibm.com/docs/en/linux-on-systems?topic=tasks-find-machine-serial>)  to obtain the host key document (<https://www.ibm.com/docs/en/linux-on-systems?topic=execution-obtain-host-key-document>)  from IBM Resource Link.
4. Ensure the host key document (<https://www.ibm.com/docs/en/linux-on-systems?topic=execution-verify-host-key-document>)  is genuine and provided by IBM.
5. Import (<https://www.ibm.com/docs/en/linux-on-systems?topic=tasks-key-bundles>)  the host key document to complete the hardware enablement of the IBM Secure Execution technology.

4.2.2 Installing and configuring an LPAR with KVM included in SLES for IBM Z and LinuxONE



Note

This section provides installation configuration and guidance but **not** a detailed set of steps.

1. Use the recommendation in Additional hardware requirements for the KVM host (<https://www.ibm.com/docs/en/hpvs/2.1.x?topic=servers-system-requirements#additional-hardware-requirements-for-the-kvm-host>)  to define an LPAR.

The following `lsblk` output is an example of the disk storage devices, sizes, lvm details, file system types and mount points used to test HPVS 2.1.x.

NAME	SIZE	FSTYPE	MOUNTPOINTS
sde	500G	mpath_member	
└─sde1	500G	LVM2_member	

└─3600507638081855cd80000000000004a	500G		
└─3600507638081855cd80000000000004a-part1	500G	LVM2_member	
└─vmsvg-vms	500G	xfs	/var/lib/libvirt/ images
sdj	500G	mpath_member	
└─sdj1	500G	LVM2_member	
└─3600507638081855cd80000000000004a	500G		
└─3600507638081855cd80000000000004a-part1	500G	LVM2_member	
└─vmsvg-vms	500G	xfs	/var/lib/libvirt/ images
dasda	6.9G		
└─dasda1	102M	ext2	/boot/zipl
└─dasda2	6.8G	LVM2_member	
└─system-root	13.6G	btrfs	/
dasdb	6.9G		
└─dasdb1	6.9G	LVM2_member	
└─system-root	13.6G	btrfs	/

At least one network interface must be defined. The following is the `lszdev qeth` output showing one OSA network interface has been defined.

TYPE	ID	ON	PERS	NAMES
qeth	0.0.0800:0.0.0801:0.0.0802	yes	yes	eth0

2. Use the [Installation on IBM Z and LinuxONE \(https://documentation.suse.com/sles/15-SP4/single-html/SLES-deployment/#cha-zseries\)](https://documentation.suse.com/sles/15-SP4/single-html/SLES-deployment/#cha-zseries) documentation to install the latest version of SUSE Linux Enterprise Server into the LPAR.

The following is a list of recommended patterns to select during the installation.

Name	Summary
apparmor	AppArmor
base	Minimal Base System
enhanced_base	Enhanced Base System
hwcrypto	System z HW crypto support
kvm_server	KVM Host Server
kvm_tools	KVM Virtualization Host and tools
x11_yast	YaST User Interfaces
yast2_basis	YaST Base Utilities

- a. Register and fully patch the SLES installation.

3. Enable and start the KVM host

```
systemctl enable --now libvirtd.service
```




Important

A networking choice should be considered at this time.

The [KVM Host Networking Configuration Choices \(https://www.ibm.com/docs/en/linux-on-systems?topic=recommendations-kvm-host-networking-configuration-choices\)](https://www.ibm.com/docs/en/linux-on-systems?topic=recommendations-kvm-host-networking-configuration-choices) in the IBM documentation provides a detailed list of networking choices and pros and cons.

- Using a Linux bridge with NAT for KVM guests
- Using a Linux bridge (without NAT) for KVM guests
- Using an Open vSwitch bridge with KVM guests
- Using the MacVTap driver with KVM guests

Confirm prerequisites like [OSA interface traffic forwarding \(https://www.ibm.com/docs/en/linux-on-systems?topic=choices-osa-interface-traffic-forwarding\)](https://www.ibm.com/docs/en/linux-on-systems?topic=choices-osa-interface-traffic-forwarding) are met for your choice of networking.

This getting started guide will use the [MacVTap driver \(https://www.ibm.com/docs/en/linux-on-systems?topic=choices-using-macvtap-driver\)](https://www.ibm.com/docs/en/linux-on-systems?topic=choices-using-macvtap-driver) in Bridge mode with the `eth0` interface. No prerequisites are required using MacVTap but it is important to review the [MacVTap isolation / limitations](#) section.

An external DHCP server is required to provide networking information for the VMs using MacVTAP.

4.2.3 Enabling Secure Execution capabilities on the KVM host

1. Enable [IBM Secure Execution \(https://www.ibm.com/docs/en/linux-on-systems?topic=tasks-enable-kvm-host\)](https://www.ibm.com/docs/en/linux-on-systems?topic=tasks-enable-kvm-host)

- a. Append `prot_virt=1` to `GRUB_CMDLINE_LINUX_DEFAULT` in `/etc/default/grub`.

The following is an example of the `GRUB_CMDLINE_LINUX_DEFAULT` line with `prot_virt=1` appended.

```
GRUB_CMDLINE_LINUX_DEFAULT="hvc_iucv=8 TERM=dumb mitigations=auto  
security=apparmor cio_ignore=all,!ipldev,!condev prot_virt=1"
```

- b. Recreate `grub.cfg`.

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

c. Reboot.

2. Verify IBM Secure Execution is enabled with `virt-host-validate`. Verify the line has `PASS` in the output.

```
QEMU: Checking for secure guest support : PASS
```

4.2.4 Downloading the HPCR image

1. Make a directory on the KVM host to store the HPCR image.

```
mkdir /opt/hpvs-hpcr
```

2. Download the latest version of the HPCR image from IBM Passport Advantage (https://www-01.ibm.com/software/passportadvantage/pao_customer.html) to the directory created in the previous step.



Note

Verify if an IBM HPCR fix pack is available from IBM Fix Central (<https://www.ibm.com/support/fixcentral>).

3. Verify the integrity and decompress the download following steps 5 - 6 of the procedure (<https://www.ibm.com/docs/en/hpvs/2.1.x?topic=servers-downloading-hyper-protect-container-runtime-image#procedure>).
4. Make a note of location and file name of `qcow2` HPCR image. For example, `ibm-hyper-protect-container-runtime-23.3.0.qcow2` is the file name which is located in directory `/opt/hpvs-hpcr/IBM-HPVS-OnPrem-v2.1.4-EN-Trial/images`.

4.2.5 Setting up an `rsyslog` logging service for HPVS log information

With the release of SUSE Linux Enterprise Server 15 SP5, Minimal VM images for IBM Z and LinuxONE systems can be downloaded from [suse.com](https://www.suse.com/download/sles/) (<https://www.suse.com/download/sles/>). Two KVM `qcow2` image file options are available: *kvm* and *Cloud*. The file name of the `qcow2` image files will include either *kvm* or *Cloud* for differentiation.

- The *kvm* image file will prompt during firstboot for information to customize items like host name and networking information.
- The *Cloud* image file uses `cloud-init` to customize the host name and networking information.

You can choose to manually install SUSE Linux Enterprise Server with the installation media or use a pre-built SLES image file from SUSE.



Note

This guide will provide the steps to use the *Cloud* image to deploy an `rsyslog` kvm virtual machine on the same KVM host where the confidential computing workload will be run.

1. Make a directory to store the disk image and other files for the `rsyslog` VM.

```
mkdir /var/lib/libvirt/images/rsyslog
```

2. Download the `SLES 15 SP5 Minimal VM Cloud qcow2` image file at [suse.com \(https://www.suse.com/download/sles/\)](https://www.suse.com/download/sles/). Place the file in the `/var/lib/libvirt/images/rsyslog` directory. Name the file `rsyslog.qcow2`.

3. Create the following required `cloud-init` files.

- a. `vi /var/lib/libvirt/images/rsyslog/meta-data`

```
local-hostname: rsyslog
```

- b. `vi /var/lib/libvirt/images/rsyslog/user-data`

```
#cloud-config
ssh_authorized_keys:
  - <public ssh key 1>
  - <public ssh key 2>
```

⇒ where `<public ssh key 1>` and if needed `<public ssh key 2>` will be added to the `authorized_keys` file for the `sles` user.



Note

This configuration requires an external DHCP server to provide the static networking information to the `rsyslog` server based on MAC address. Additional cloud examples for networking and other configurations can be found at the following resources:

- cloud-init configuration examples (<https://documentation.suse.com/pt-br/sles/15-SP4/html/SLES-all/article-minimal-vm.html#sec-cloud-init-config-examples>) ↗
- Configuration with cloud-init (<https://en.opensuse.org/Portal:MicroOS/cloud-init>) ↗
- Cloud config examples (<https://cloudinit.readthedocs.io/en/latest/reference/examples.html#cloud-config-examples>) ↗

4. Generate a MAC address for the `rsyslog` VM.

a. Create the `/root/bin/macgen.py` python script.

```
#!/usr/bin/python3
# macgen.py script to generate a MAC address for guests on KVM

import random

def randomMAC():
    mac = [ 0x52, 0x54, 0x00,
            random.randint(0x00, 0x7f),
            random.randint(0x00, 0xff),
            random.randint(0x00, 0xff) ]
    return ':'.join(map(lambda x: "%02x" % x, mac))

print(randomMAC())
```

b. Make the `macgen.py` script executable.

```
chmod +x /root/bin/macgen.py
```

c. Run `macgen.py` and note the output. Here is an example when run.

```
# macgen.py
```

52:54:00:5b:15:df

5. Use the generated MAC address to define a host entry with a static IP address in your DHCP server.
6. Define and start the `rsyslog` VM with the following command.

```
virt-install \
--name=rsyslog \
--osinfo sle15sp5 \
--memory=2048 \
--vcpus=2 \
--clock offset=utc \
--events on_poweroff=destroy,on_reboot=restart,on_crash=destroy \
--disk /var/lib/libvirt/images/rsyslog/rsyslog.qcow2,\
driver.iommu=on,target.bus=virtio,boot.order=1 \
--network type=direct,source=eth0,source.mode=bridge,model.type=virtio,\
driver.name=vhost,mac.address=<macaddress> \
--console pty,target.type=sclp,target.port=0 \
--audio id=1,type=none \
--memballoon none \
--panic s390 \
--cloud-init user-data=/var/lib/libvirt/images/rsyslog/user-data,\
meta-data=/var/lib/libvirt/images/rsyslog/meta-data
```

⇒ where `<macaddress>` was noted above.

The VM console will be displayed and the VM will boot. The VM IP address is properly configured when the IP address is displayed next to `eth0`. For example:

```
Welcome to SUSE Linux Enterprise Server 15 SP5 (s390x) - Kernel
5.14.21-150500.53.2-default (ttysclp0).

eth0: 10.161.159.11 fe80::5054:ff:feb6:ad3e

rsyslog login:
```



Note

Press `Ctrl +]` keys to disconnect from the VM console.

7. Log in to the VM via SSH as the `sles` user from another system using the private portion of the SSH key defined above.
8. Register and fully patch SLES.

- a. Reboot if necessary.

9. Install the rsyslog packages.

```
zypper in rsyslog rsyslog-module-gtls
```

10. Generate self-signed certificates for encrypted communication from the HPCR instance to the rsyslog server.

- a. mkdir /certs && cd /certs to create and change into the directory.

- b. Create the following OpenSSL configuration files.

/certs/ca.cnf

```
[ req ]
default_bits = 2048
default_md = sha256
prompt = no
encrypt_key = no
distinguished_name = dn

[ dn ]
C = US
O = rsyslog CA
CN = ca.rsyslog
```

/certs/rsyslog.cnf

```
[ req ]
default_bits = 2048
default_md = sha256
prompt = no
encrypt_key = no
distinguished_name = dn

[ server ]
subjectAltName = IP:<ip address>
extendedKeyUsage = serverAuth

[ dn ]
CN = <ip address>
```

⇒ where <ip address> is the IP address of the rsyslog VM.

/certs/slebcy-paynow-website.cnf

```
[ req ]
default_bits = 2048
default_md = sha256
prompt = no
encrypt_key = no
distinguished_name = dn

[ server ]
subjectAltName = IP:<ip address>
extendedKeyUsage = serverAuth

[ dn ]
CN = <ip address>
```

⇒ where <ip address> is the IP address of the HPCR VM that will run the PayNow Web site application.

- c. Generate the key, certificate signing request and certificate for the certificate authority.

```
openssl genrsa -out ca-key.pem 4096
openssl req -config ca.cnf -key ca-key.pem -new -out ca-request.pem
openssl x509 -signkey ca-key.pem -in ca-request.pem -req -days 3650 \
-out ca-cert.pem
```

- d. Generate the key, certificate signing request and CA signed certificate for the rsyslog server.

```
openssl genrsa -out rsyslog-key.pem 4096
openssl req -config rsyslog.cnf -key rsyslog-key.pem -new \
-out rsyslog-request.pem
openssl x509 -req -in rsyslog-request.pem -days 1000 -CA ca-cert.pem \
-CAkey ca-key.pem -CAcreateserial -extfile rsyslog.cnf -extensions server \
-out rsyslog-cert.pem
```

- e. Generate the key, certificate signing request and CA signed certificate for the HPCR VM that will run the PayNow Web site.

```
openssl genrsa -out slebci-paynow-website-key.pem 4096
openssl req -config slebci-paynow-website.cnf \
-key slebci-paynow-website-key.pem -new -out slebci-paynow-website-request.pem
openssl x509 -req -in slebci-paynow-website-request.pem -days 1000 \
-CA ca-cert.pem -CAkey ca-key.pem -CAcreateserial \
-out slebci-paynow-website-cert.pem
```

11. Create the `rsyslog` configuration to capture the logging information from the HPCR VM that will run the PayNow Web site.

```
vi /etc/rsyslog.d/server.conf
```

```
# output to journal
module(load="omjournal")
template(name="journal" type="list") {
# can add other metadata here
property(outname="PRIORITY" name="pri")
property(outname="SYSLOG_FACILITY" name="syslogfacility")
property(outname="SYSLOG_IDENTIFIER" name="app-name")
property(outname="HOSTNAME" name="hostname")
property(outname="MESSAGE" name="msg")
}

ruleset(name="journal-output") {
# action(type="omjournal" template="journal")
action(type="omfile" dirCreateMode="0766" FileCreateMode="0644" File="/var/log/
hpcr.log")
}

# make gtls driver the default and set certificate files
$DefaultNetstreamDriver "gtls"
$DefaultNetstreamDriverCAFile /certs/ca-cert.pem
$DefaultNetstreamDriverCertFile /certs/rsyslog-cert.pem
$DefaultNetstreamDriverKeyFile /certs/rsyslog-key.pem

# load TCP listener
module(
load="imtcp"
StreamDriver.Name="gtls"
StreamDriver.Mode="1"
StreamDriver.Authmode="x509/certvalid"
)

# start up listener at port 6514
input(
type="imtcp"
port="6514"
ruleset="journal-output"
)
```


⇒ where logging information will be written to `/var/log/hpcr.log`. Comment out this line and uncomment the line above to write log information to the journal.

12. Enable, start and verify that the `rsyslog` service is running.

```
systemctl enable --now rsyslog
systemctl status rsyslog
```

4.2.6 Prepare IBM Crypto Express adapters and domains to be passed into a containerized workload

The Crypto Passthrough capability enables Hardware Security Modules (HSM) to be available to the IBM Hyper Protect Container Runtime (HPCR) where the containerized workload will run. A [video \(https://youtu.be/f7iQ-1dE9Ag?feature=shared\)](https://youtu.be/f7iQ-1dE9Ag?feature=shared) is available which explains crypto passthrough with a demonstration application built on a SUSE Linux Enterprise Base Container image.

To enable virtual cryptographic adapters to be passed through, specific domains also referred to as AP queues must have an association request created.



Important

The domain associations that are created following these steps do not persist after a reboot of the KVM host. For information about creating a persistent association, see [Preparing pass-through devices for cryptographic adapter resources \(https://www.ibm.com/docs/en/linux-on-systems?topic=through-ap-queues\)](https://www.ibm.com/docs/en/linux-on-systems?topic=through-ap-queues).

1. Use `lszcrypt -V` to review the Crypto Express cards and domains that are available to be associated.



Note

The card `00` and the domain `0032` are used in this documentation.

CARD.DOM	TYPE	MODE	STATUS	REQUESTS	PENDING	HWTYPE	QDEPTH	FUNCTIONS
DRIVER								

00	CEX8P	EP11-Coproc	online	3	0	14	08	-----XN-F-
	cex4card							
00.0032	CEX8P	EP11-Coproc	online	3	0	14	08	-----XN-F-
	cex4queue							

2. Make note of the Master Key Verification Pattern (mkvp) which will be used in following sections.

```
cat /sys/devices/ap/card00/00.0032/mkvp
```

3. Release the domains from the direct control by the host operating system.

```
echo 0x0 > /sys/bus/ap/apmask
echo 0x0 > /sys/bus/ap/aqmask
```



Note

This releases all domains from host control. For information about handling domains more selectively, see [Freeing AP queues for KVM guests \(https://www.ibm.com/docs/en/linuxonibm/com.ibm.linux.z.lkdd/lkdd_t_crypto_wrk_mask.html\)](https://www.ibm.com/docs/en/linuxonibm/com.ibm.linux.z.lkdd/lkdd_t_crypto_wrk_mask.html).

4. Load the virtual function IO adapter ap kernel module.

```
modprobe vfio_ap
```

5. Execute the following commands to associate the domain as a vfio_ap controlled device.

```
uuid=$(uuidgen)
echo ${uuid} > /sys/devices/vfio_ap/matrix/mdev_supported_types/vfio_ap-passthrough/create
echo 0x00 > /sys/devices/vfio_ap/matrix/${uuid}/assign_adapter
echo 0x0032 > /sys/devices/vfio_ap/matrix/${uuid}/assign_domain
```

6. Verify the card and domain are controlled by vfio_ap.

CARD.DOM	TYPE	MODE	STATUS	REQUESTS	PENDING	HWTYP	QDEPTH	FUNCTIONS
00	CEX8P	EP11-Coproc	online	4	0	14	08	-----XN-F-
	cex4card							
00.0032	CEX8P	EP11-Coproc	assigned	-	-	14	08	-----XN-F-
	vfio_ap							

5 Building, publishing, and defining

In this section, you prepare a workload container image, publish the container image to a registry, and define the IBM Hyper Protect Services contract that will be used during deployment. The steps outlined here are the same for cloud and on-premises deployments.

You will use the [PayNow Node.js application \(https://cloud.ibm.com/docs/vpc?topic=vpc-financial-transaction-confidential-computing-on-hyper-protect-virtual-server-for-vpc\)](https://cloud.ibm.com/docs/vpc?topic=vpc-financial-transaction-confidential-computing-on-hyper-protect-virtual-server-for-vpc), which has been already built on a SLE BCI container image and published to a container registry. The steps are also available if you want to build the [PayNow Node.js application](#) on a SLE BCI container.

See [confidential computing in action \(https://mediacenter.ibm.com/media/Confidential+Computing+for+a+financial+transaction+using+Hyper+Protect+Virtual+Server+for+VPC/1_v3j2oo6\)](https://mediacenter.ibm.com/media/Confidential+Computing+for+a+financial+transaction+using+Hyper+Protect+Virtual+Server+for+VPC/1_v3j2oo6). In this demonstration, the [PayNow Node.js application](#) shows a financial transaction running without confidential computing and with confidential computing.

5.1 Building a container image

Building one or more application container images is a necessary step. The [PayNow Node.js application container image](#) has already been built on a SLE BCI container and published to a container registry. You can go to [Section 5.3, “Defining the contract”](#) to use the [PayNow Node.js application container image](#) or follow the steps below to build your own image of the [PayNow Node.js application](#).

Access to a Linux on s390x instance is required to build the [PayNow Node.js application container image](#). These steps will use [git](#) and [podman](#) to build and publish the OCI-compliant [PayNow Node.js application container image](#). An option is to use an IBM Cloud virtual server instance (VSI) for VPC with a SUSE Linux Enterprise Server [s390x stock virtual server image \(https://cloud.ibm.com/docs/vpc?topic=vpc-vsabout-images#vs-s390x-supported-os\)](#) but any Linux distribution on s390x with [git](#) and [podman](#) is sufficient.

The [SLE BCI Language Container Images \(https://registry.suse.com/repositories/bci-nodejs22?arch=s390x\)](https://registry.suse.com/repositories/bci-nodejs22?arch=s390x) for [Node.js development](#) is used as base for the [PayNow Node.js application container image](#).

1. Clone the Pay Now Web site sources.

```
git clone https://github.com/mfriesenegger/paynow-website.git
cd paynow-website
```

```
git checkout slebci-paynow-website
```

2. Build the PayNow Node.js application container image.

```
podman build -f ./Dockerfile -t slebci-paynow-website
```

3. Start the **Verify the PayNow Node.js** application container.

```
podman run -d --rm --name paynow-website -p 8443:8443 \
localhost/slebci-paynow-website
```

You will see similar output.

```
6628a61d4e8c427e065c00f140c114e57f0dbd8ddf861752140e43cc77f08d74
```

4. Verify the PayNow Node.js application container has started.

```
podman ps
```

You will see similar output.

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	
6628a61d4e8c	localhost/slebci-paynow-website:latest	/bin/sh -c npm st...	6
seconds ago	Up 5 seconds ago	0.0.0.0:8443->8443/tcp paynow-website	

5. Stop the running PayNow Node.js application container

```
podman stop paynow-website
```

You will see similar output.

```
6628a61d4e8c427e065c00f140c114e57f0dbd8ddf861752140e43cc77f08d74
```

5.2 Publishing the container image to a registry

Publishing one or more application container images is a necessary step. The PayNow Node.js application container image has already been built on a SLE BCI container and published to a container registry. You can go to [Section 5.3, "Defining the contract"](#) to use the PayNow Node.js application container image or follow the steps below to publish your own image of the PayNow Node.js application.

Access to the same Linux on s390x instance that built the `PayNow Node.js` application container image is required to publish the image. These steps will use `podman` to publish the OCI-compliant `PayNow Node.js` application container image. A properly configured GitHub account is required to publish the image to your [GitHub Container registry \(https://docs.github.com/en/packages/working-with-a-github-packages-registry/working-with-the-container-registry\)](https://docs.github.com/en/packages/working-with-a-github-packages-registry/working-with-the-container-registry).

1. Tag the `PayNow Node.js` application container image to your GitHub account.

```
podman tag localhost/slebci-paynow-website:latest \
ghcr.io/<github user>/slebci-paynow-website:latest
```



Note

Replace `<github user>` with your personal GitHub user name.

2. Log in to the GitHub container registry.

```
podman login ghcr.io
Username: <github user>
Password: <github token>
Login Succeeded!
```



Note

Replace `<github user>` with your personal GitHub user name.

Replace `<github token>` with your personal GitHub access token.

3. Push the `PayNow Node.js` application container image.

```
podman push ghcr.io/<github user>/slebci-paynow-website:latest
```

You will see similar output.

```
Getting image source signatures
Copying blob 0ccbad5913ff done
Copying blob b326be1c2597 done
Copying blob 33514e467d16 done
Copying blob bd6fa590b9f7 done
Copying blob b6d354fd1660 done
Copying config e71fdf2e7a done
Writing manifest to image destination
Storing signatures
```



Note

Replace `<github user>` with your personal GitHub user name.

4. Remove the `PayNow Node.js` application container image.

```
podman rmi ghcr.io/<github user>/slebcy-paynow-website:latest
```

You will see similar output.

```
Untagged: ghcr.io/mfriesenegger/slebcy-paynow-website:latest
```

5. Confirm that the `PayNow Node.js` application container image can be pulled from your GitHub Container Registry.

```
podman pull ghcr.io/<github user>/slebcy-paynow-website:latest
```

You will see similar output.

```
Trying to pull ghcr.io/mfriesenegger/slebcy-paynow-website:latest...
Getting image source signatures
Copying blob 433e70b2cea5 skipped: already exists
Copying blob b12c44df6add skipped: already exists
Copying blob cc2d0d7ba04c [-----] 0.0b / 0.0b
Copying blob a74519fc309a [-----] 0.0b / 0.0b
Copying blob ed8b5e1d0d7a [-----] 0.0b / 0.0b
Copying config e71fdf2e7a done
Writing manifest to image destination
Storing signatures
e71fdf2e7ad357830ece71bfc30367f52ab7afa74401d1506cefc091a62064b
```



Note

Replace `<github user>` with your personal GitHub user name.

6. Log out of ghcr.io.

```
podman logout ghcr.io
```

You will see similar output.

```
Removed login credentials for ghcr.io
```

5.3 Defining the contract

A contract is a required definition of the workload configuration in YAML format.

1. Create a directory and Docker Compose file on a local system.

```
mkdir slebci-paynow-website
cd slebci-paynow-website
vi docker-compose.yml
```

Update the digest of the [latest image \(https://github.com/mfriesenegger/paynow-website/pkgs/container/slebci-paynow-website\)](https://github.com/mfriesenegger/paynow-website/pkgs/container/slebci-paynow-website) in the following example if using the pre-built PayNow Node.js application container image.

```
services:
  paynow:
    image: ghcr.io/mfriesenegger/slebci-paynow-website@sha256:ecb229f68aef81ca4a2b7b5a9eb192081fa2a170e44d9e5a28180bb12682ce5d
    ports:
      - "8080:8080"
      - "8443:8443"
```



Important

Verify the [SHA256 digest \(https://github.com/users/mfriesenegger/packages/container/package/slebci-paynow-website\)](https://github.com/users/mfriesenegger/packages/container/package/slebci-paynow-website) of the prebuilt image.

Or use the following if the PayNow Node.js application container image was built by you and published in your GitHub Container registry.

```
services:
  paynow:
    image: ghcr.io/<github user>/slebci-paynow-website@<sha256 digest>
    ports:
      - "8080:8080"
      - "8443:8443"
```



Note

Replace `<github user>` with your personal GitHub user name.

Replace `<github digest>` with the digest information for the in published in your GitHub Container Registry.

2. Create a base64 encoded TAR archive .

```
tar -czvf - docker-compose.yml | base64 -w0 > docker-compose.b64
```

3. Create the contract file.

```
vi slebci-paynow-website-contract.yml
```

a. Add the logging configuration to the contract.

Use the following [Section 4.1.2, "Setting up a logging service for HPVS for VPC provisioning"](#) for IBM Cloud.

```
env: |
  type: env
  logging:
    logDNA:
    logRouter:
      hostname: <hostname>
      iamApiKey: <apikey>
      port: 443
```

⇒ where `<hostname>` and `<apikey>` were noted earlier.

Use the following [Section 4.2.5, "Setting up an rsyslog logging service for HPVS log information"](#) for on-premises deployments.

```
env: |
  type: env
  logging:
    syslog:
      hostname: <ip address>
      port: 6514
      server: <ca-cert>
      cert: <client certificate>
      key: <client private key>
```

⇒ where `<ip address>` is the IP address of the rsyslog server.

⇒ where the output of the following command executed on the rsyslog server replaces each item.

<ca-cert>


```
IFS=$'\n'; echo -n \"; for line in $(cat /certs/ca-cert.pem);\ndo echo -n $line\"\\n\"; done; echo \"
```

<client certificate>

```
IFS=$'\n'; echo -n \"; for line in $(cat \\\n/certs/slebcipaynowwebsitecert.pem); do echo -n $line\"\\n\"; done; echo \"
```

<client private key>

```
IFS=$'\n'; echo -n \"; for line in $(cat \\\n/certs/slebcipaynowwebsitekey.pem); do echo -n $line\"\\n\"; done; echo \"
```



Note

The certificate and key output generated by the commands above will be copied and pasted as a single line for each item.

- b. Add the workload configuration to the contract.

Use the following if using the pre-built PayNow Node.js application container image.

```
workload: |\n  type: workload\n  compose:\n    archive: <archive>
```

⇒ where <archive> was created in the **Create a base64 encoded TAR archive** [\[base64-encoded-archive\]](#) step.

Or use the following if the PayNow Node.js application container image was built by you and published in your GitHub Container Registry.

```
workload: |\n  type: workload\n  auths:\n    ghcr.io:\n      username: <github user>\n      password: <github token>\n  compose:\n    archive: <archive>
```

⇒ where <github user> and <github token> are for your personal GitHub account.

⇒ where <archive> was created in the **Create a base64 encoded TAR archive** [\[base64-encoded-archive\]](#) step.

An example contract using the IBM Cloud logging service with the pre-built image is shown below.



Note

The base64 data for the archive item is truncated.

```
env: |
  type: env
  logging:
    logDNA:
      hostname: syslog-a.ca-tor.logging.cloud.ibm.com
      port: 6514
      ingestionKey: e2db535664e682b59101b742d59234da
  workload: |
    type: workload
    compose:
      archive: H4sIAAAAAAAAA+3UzW7bMAwHcJ/zFELucahP2gYG7LjjXkG2mNRoXQdWtrZvP6XBsCyHboeuW7H/
```

An example contract using the rsyslog logging service with the pre-built image is shown below.



Note

The certificate and key data for the rsyslog logging service is truncated.


The base64 data for the archive item is truncated.

```
env: |
  type: env
  logging:
    syslog:
      hostname: 10.161.159.11
      port: 6514
      server: "-----BEGIN CERTIFICATE-----
\nMIIE9TCCAt0CFGXTURlEXHPvfkrSMq6u80keo3NrMA0GC...0qex34LGN3kYw==\n-----END
CERTIFICATE-----\n"
      cert: "-----BEGIN CERTIFICATE-----\nMIIE3jCCAsYCFBYlycZ0oZP8fenP3RSx7/
PJDkjcMA0GCSq...M0yk3XJgr\nB5w=\n-----END CERTIFICATE-----\n"
      key: "-----BEGIN RSA PRIVATE KEY-----
\nMIIIKqIBAAKCAgEA1bSLJYv9309nMUZOB4Qo79N9vrZE...4WUbz3Fyw==\n-----END RSA PRIVATE
KEY-----\n"
```

```
workload: |
  type: workload
  compose:
    archive: H4sIAAAAAAAAAA+3UzW7bMAwHcJ/zFELucahP2gYG7LjjXkG2mNRoXQdWtrZvP6XBsCyHboeuW7H/
```

5.3.1 Adding IBM Crypto Express adapters and domains to the contract

Important

The IBM Crypto Passthrough documentation (https://www.ibm.com/docs/en/hpvs/2.2.x?topic=servers-crypto-passthrough#hpcr_sample_crypto_passthrough_contract)  provides options - secret and pvsecret - to pass the crypto device details within the contract. This demo is using the secret option, for simplification. The HPCR bootstrap will transparently execute the pvsecret using IBM validated and curated code. Using the pvsecret option requires additional effort by a relying party persona. Using pvsecret provides additional security because the attestation request and result are created and validated by the relying party rather than by IBM provided bootstrap code.

1. Add one or more *index* sections in the *crypto-pt* section to the *env* section in the contract.

```
env: |
  type: env
  crypto-pt:
    lock: true
    index-<num>:
      type: secret
      domain-id: "<card>.<domain>"
      secret: <base64 encoded secret>
      mkvp: <master key verification pattern>
```

⇒ where *lock*: is true.

⇒ where *<num>* in *index-<num>*: must start at 1 and must increase in sequence for each domain-id passed via the contract. For example, 1, 2, 3, and so on.

⇒ where *type*: is secret.

⇒ where *domain-id*: is the '*<card>.<domain>*' of the device being passed.

⇒ where *secret*: can be generated with `dd if=/dev/random bs=32 count=1 2> /dev/null | base64`.

⇒ where *mkvp*: was noted in the [\[note-mkvp\]](#) step.

2. Add one or more *HKD-<model>-<plant> <serial>* sections in the *host-attestation* section to the *env* section in the contract.

```
env: |
  type: env
  host-attestation:
    HKD-<model>-<plant><serial>:
      description:<Description of HKD>
      host-key-doc: <A base64 encoded Host Key Document>
```

⇒ where *<model>*, *<plant>* and *<serial>* in *HKD-<model>-<plant> <serial>*: follows the naming convention of the download HKD file.

⇒ where *_description*: is a description of the HKD.

⇒ where *host-key-doc*: can be generated with IFS=\$'\n'; echo -n \"; for line in \$(cat <path to downloaded HKD document> | base64); do echo -n \$line\"\\n\"; done; echo \"

An example of the *env* section of the contract with only the *crypto-pt* and *host-attestation* sections is provided below.



Note

The base64 encoded HKD for host-key-doc is truncated.

```
env: |
  type: env
  crypto-pt:
    lock: true
    index-1:
      type: secret
      domain-id: "00.0032"
      secret: A2tDsMCzwSxr09bCvxSzjtd02SkhKfL99KXtvIJKgHM=
      mkvp: 0xbb3cf1db618294d05e9fd4d51572fe2de8409044f96ea1519bf9417500000000
  host-attestation:
    HKD-3931-0271CC8:
      description: zt14
      host-key-doc:
        "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tDQpNSUlgSmPDQ0F3NmdBd0lCQWdJS1VVSg...NCi0tLS0tRU5EIENFUlRJRkdDQVRV
        \\n"
```


6 Deployment

The deployment steps differ, depending on whether you are deploying to cloud or to your on-premises environment. Proceed to either [Section 6.1, “Cloud deployment”](#) or [Section 6.2, “On-premises deployment”](#).

6.1 Cloud deployment

For this guide, you use the IBM Cloud Web Console to perform the deployment. It is also possible to perform a deployment using the IBM Cloud CLI or Terraform.

Follow these steps to deploy your confidential container workload on IBM Cloud Hyper Protect Virtual Servers for VPC.

1. Return to the <https://cloud.ibm.com/>  Web browser window or tab where you created the VPC.
2. Select *Navigation Menu > VPC Infrastructure > Virtual server instances*.
3. Confirm that the correct Region is selected.
4. Click *Create*.
5. Edit the following in *Virtual server for VPC*:
 - a. Change *Architecture* to IBM Z, LinuxONE.
 - b. Use the slider under *Confidential computing* to enable Run your workload with an OS and a profile protected by Secure Execution.
 - c. Verify that the *Zone* in *Location* matches the location (zone) within the region that you noted earlier.
 - d. Set the *Name* of the instance to slebci-paynow-website.
 - e. Click *Import user data* in *User data (optional)* to select slebci-paynow-website-contract.yml from the local system.
 - f. Click *Create virtual server*.

6. Select *Network > Floating IPs* and complete the following:
 - a. Click *Actions...* to the right of the myvpc-paynow-floatip entry.
 - b. Select *Bind*.
 - c. Select slebci-paynow-website in *Resource to bind*.
 - d. Click *Bind*.
7. Select *Compute > Virtual server instances* and complete the following:
 - a. Click *Actions...* to the right of the slebci-paynow-website entry.
 - b. Select *Open serial console*.
 - c. Watch for the following within the serial console as the instance is being started:

```
VSI has started successfully.
```

6.2 On-premises deployment

An on-premises deployment will be prepared and started via the CLI on the KVM host.

1. Make a directory to store the disk image and other files for the HPCR.
2. Copy the HPCR image file from the location noted in a previous step [\[note-hpcr-image-location\]](#) to `/var/lib/libvirt/images/slebci-paynow-website`.
3. Create the cloud configuration files and init-disk ISO.

- a. Create the meta-data file in the directory created above.

For example:

```
vi /var/lib/libvirt/images/slebci-paynow-website/meta-data
```

Add the following line to the file:

```
local-hostname: slebci-paynow-website
```

- b. Create the vendor-data file in the same directory.

For example:

```
vi /var/lib/libvirt/images/slebci-paynow-website/vendor-data
```

Add the following contents to the vendor-data file:

```
#cloud-config
users:
- default
```

- c. Copy the slebci-paynow-website-contract.yml contract file from the local system to the /var/lib/libvirt/images/slebci-paynow-website/user-data file on the KVM host.
- d. Create the init-disk ISO image.

```
mkisofs \
  -output /var/lib/libvirt/images/slebci-paynow-website/init-disk \
  -volid cidata -joliet -rock \
  /var/lib/libvirt/images/slebci-paynow-website/user-data \
  /var/lib/libvirt/images/slebci-paynow-website/meta-data \
  /var/lib/libvirt/images/slebci-paynow-website/vendor-data
```

4. Generate a MAC address for the HPCR VM with macgen.py.

```
macgen.py
```

You should see something like:

```
# macgen.py
52:54:00:02:77:72
```

5. Use the generated MAC address to define a host entry with a static IP address in your DHCP server.
6. Define and start the slebci-paynow-website VM.

```
virt-install \
  --name=slebci-paynow-website \
  --osinfo ubuntu20.04 \
  --memory=3815 \
  --vcpus vcpu.placement=static,vcpus=2 \
  --boot hd \
  --cpu mode=host-model,check=partial \
  --clock offset=utc \
```

```
--events on_poweroff=destroy,on_reboot=restart,on_crash=destroy \
--disk /var/lib/libvirt/images/slebci-paynow-website/\
ibm-hyper-protect-container-runtime-23.3.0.qcow2,driver.iommu=on,\
target.bus=virtio,\
boot.order=1 \
--disk /var/lib/libvirt/images/slebci-paynow-website/\
slebci-paynow-website-data.qcow2,size=10,format=qcow2,cache=none,\
driver.discard=ignore,driver.iommu=on,target.bus=virtio \
--disk /var/lib/libvirt/images/slebci-paynow-website/init-disk,\
format=raw,cache=none,driver.discard=ignore,driver.iommu=on,\
target.bus=virtio \
--controller type=pci,index=0,model=pci-root \
--network type=direct,source=eth0,source.mode=bridge,\
model.type=virtio,driver.name=vhost,driver.iommu=on,\
mac.address=<macaddress> \
--console pty,target.type=sclp,target.port=0 \
--audio id=1,type=none \
--memballoon none \
--panic s390 \
--channel none \
--rng none
```

⇒ where <macaddress> in this command is replaced with the MAC address you noted.

7. The VM console will be displayed and the HPCR VM will boot.

Watch for the following in the console as the instance is being started:

```
VSI has started successfully.
```



Note

Press Ctrl +] keys to disconnect from the VM console.

6.2.1 Adding IBM Crypto Express adapters and domains to the deployment



Note

The IBM Hyper Protect Container Runtime (HPCR) version used at the time of this writing is ibm-hyper-protect-container-runtime-25.1.0.

1. Follow all of the steps in the previous section returning here when you get to the [\[define-start-vm\]](#) step.
2. Determine and make note of the name of the crypto domain node device.

```
virsh nodedev-list | grep $(echo ${uuid} | tr - _)
```

3. Define and start the [slebcy-paynow-website](#) VM.

```
virt-install \
--name=slebcy-paynow-website \
--osinfo ubuntu20.04 \
--memory=3815 \
--vcpus vcpu.placement=static,vcpus=2 \
--boot hd \
--cpu mode=host-model,check=partial \
--clock offset=utc \
--events on_poweroff=destroy,on_reboot=restart,on_crash=destroy \
--disk /var/lib/libvirt/images/slebcy-paynow-website/\
ibm-hyper-protect-container-runtime-23.3.0.qcow2,driver.iommu=on,\
target.bus=virtio,\
boot.order=1 \
--disk /var/lib/libvirt/images/slebcy-paynow-website/\
slebcy-paynow-website-data.qcow2,size=10,format=qcow2,cache=none,\
driver.discard=ignore,driver.iommu=on,target.bus=virtio \
--disk /var/lib/libvirt/images/slebcy-paynow-website/init-disk,\
format=raw,cache=none,driver.discard=ignore,driver.iommu=on,\
target.bus=virtio \
--controller type=pci,index=0,model=pci-root \
--network type=direct,source=eth0,source.mode=bridge,\
model.type=virtio,driver.name=vhost,driver.iommu=on,\
mac.address=<macaddress> \
--console pty,target.type=sclp,target.port=0 \
--audio id=1,type=none \
--memballoon none \
--panic s390 \
--channel none \
--rng none \
--hostdev <mdev-matrix>
```

⇒ where [<macaddress>](#) in this command is replaced with the MAC address you noted.

⇒ where [<mdev-matrix>](#) in this command is replaced with the crypto domain node device name you noted.

4. The VM console will be displayed and the HPCR VM will boot.

Watch for the following in the console as the instance is being started:

```
VSI has started successfully.
```



Note

Press Ctrl +] keys to disconnect from the VM console.

6.3 Verifying the workload is running

6.3.1 Accessing the PayNow application via its Web UI

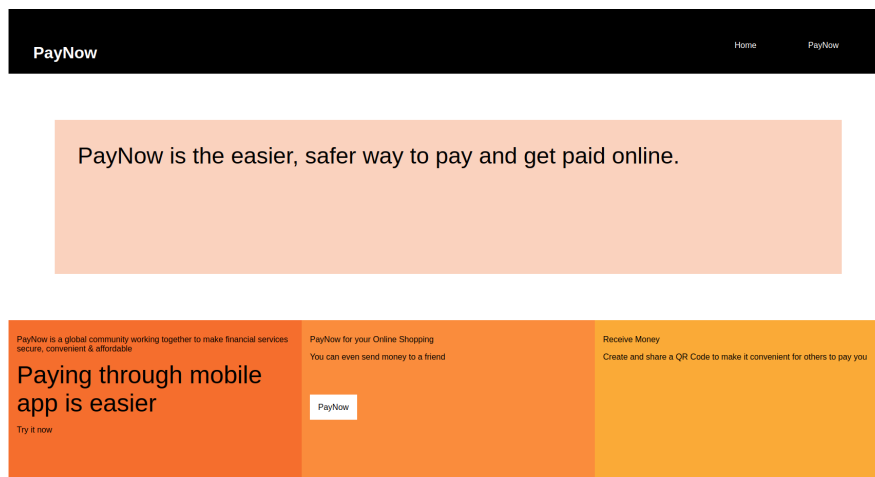
1. Open a new Web browser window or tab.
2. Enter `https://<ip address>:8443`, where `<ip address>` is one of the following:
 - the floating IP address noted earlier for the VPC deployment in IBM Cloud.
 - the IP address assigned to the HPCR VM running on the KVM host.



Note

Your Web browser may display a warning about an insecure connection and an invalid certificate because the application uses a self-signed certificate.

3. You should see that the PayNow application is running.



4. Click the *PayNow* button.
5. Enter example personally identifiable information (PII) and credit card information.

6.3.2 Reviewing the logs

1. Access the log information.
 - Use the following to review the logs of the IBM Cloud Hyper Protect Virtual Servers for VPC instance.
 - a. Return to the <https://cloud.ibm.com/> window or tab where you created the logging service.
 - b. Verify that log information is displayed.



Tip

Do the following to minimize the log output displayed.

- i. Select the Navigation menu in the upper left of the window.
 - ii. Select *Explore logs > Logs*.
 - iii. Click *Columns* to manage the displayed columns.
 - iv. Set the *In Use columns* to *Timestamp*, *_HOSTNAME* and *MESSAGE*.
- Use the following to review the logs of the on-premises IBM Hyper Protect Virtual Servers Container Runtime VM.
 - a. Log in to the rsyslog server with ssh.
 - b. Display the log file on the command line.

```
less /var/log/hpcr.log
```

2. Review the log information and find the entries:

- a. where the `slebcipaynow-website` container image is being pulled from the container registry.
- b. where the `slebcipaynow-website` container image being started.

For example:

```
slebcipaynow-website hpcr-container info Container compose-paynow-1 Started
slebcipaynow-website hpcr-container debug Docker compose result:
slebcipaynow-website hpcr-container info CONTAINER ID IMAGE
COMMAND CREATED STATUS
PORTS
NAMES
slebcipaynow-website hpcr-container info cab3993b8eb6 ghcr.io/mfriesenegger/
slebcipaynow-website "/bin/sh -c 'npm sta..." 4 seconds ago Up Less than
a second 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 0.0.0.0:8443->8443/
tcp, :::8443->8443/tcp compose-paynow-1
slebcipaynow-website hpcr-container debug Container service completed
successfully
```



Note

Date and time stamps for entries have been removed from the logging output examples shown here.

- c. for PayNow application API calls.

For example:

```
slebcipaynow-website compose-paynow-1 info GET /api/v1/transactions
slebcipaynow-website compose-paynow-1 info POST /api/v1/transactions
slebcipaynow-website compose-paynow-1 info GET /api/v1/transactions
```

7 Managing workloads

This section focuses on steps in the [Section 3.1, “Workflow”](#) that introduce additional ways to interact with the confidential containerized workload.

7.1 Enabling and verifying attestation

The IBM Hyper Protect Platform container runtime generates attestation data as it starts. The attestation data creation process is the same in [IBM Cloud](https://cloud.ibm.com/docs/vpc?topic=vpc-about-attestation) (<https://cloud.ibm.com/docs/vpc?topic=vpc-about-attestation>) and [on-premises](https://www.ibm.com/docs/en/hpvs/2.1.x?topic=servers-attestation) (<https://www.ibm.com/docs/en/hpvs/2.1.x?topic=servers-attestation>) storing several attestation-related files in `/var/hyperprotect` on the container runtime. The attestation record (also called [attestation document](https://cloud.ibm.com/docs/vpc?topic=vpc-about-attestation#attestation_doc) (https://cloud.ibm.com/docs/vpc?topic=vpc-about-attestation#attestation_doc)) contains checksums, which are used to verify the integrity of the environment where a workload starts.

The recommended method to access the attestation document is via the workload. The PayNow Node.js application (<https://cloud.ibm.com/docs/vpc?topic=vpc-financial-transaction-confidential-computing-on-hyper-protect-virtual-server-for-vpc>) application developers added API calls that provide an [encrypted](https://github.com/ibm-hyper-protect/paynow-website/blob/main/app/app.js#L42) (<https://github.com/ibm-hyper-protect/paynow-website/blob/main/app/app.js#L42>) or an [unencrypted](https://github.com/ibm-hyper-protect/paynow-website/blob/main/app/app.js#L47) (<https://github.com/ibm-hyper-protect/paynow-website/blob/main/app/app.js#L47>) attestation document.

Follow the steps below to access the encrypted attestation document and verify the attestation data of a deployed PayNow Node.js application.

1. Enable the workload to access the attestation data.

- a. Add `volumes` section to your Docker Compose file.

For example, if you are using the pre-built `PayNow Node.js` application container image, update your `docker-compose.yml` as follows:

```
services:
  paynow:
    image: ghcr.io/mfriesenegger/slebcy-paynow-website@sha256:ecb229f68aef81ca4a2b7b5a9eb192081fa2a170e44d9e5a28180bb12682ce5d
    ports:
      - "8080:8080"
      - "8443:8443"
    volumes:
      - "/var/hyperprotect:/var/hyperprotect/:ro"
```

- b. Recreate the base64 encoded TAR archive.

- c. Update the `archive` item in the workload contract with the new base64 data.

2. Use a private/public key pair for attestation document encryption.

- a. In the directory where the Docker Compose file exists, the following commands will create the private/public key pair for attestation encryption.

```
openssl genpkey -aes256 -algorithm RSA -pkeyopt rsa_keygen_bits:2048 \
-out attest-private-key.pem
openssl pkey -in attest-private-key.pem -out attest-public-key.pem -pubout
```



Note

You will be asked to define a passphrase when creating the private key and to provide the same passphrase when creating public portion of the key.

- b. Add `attestationPublicKey` to the contract file (the relevant portion is shown below).

```
workload: |
  type: workload
  compose:
    archive: H4sIAAAAAAAAAA+3UzW7bMAwHcJ/
    zFELucahP2gYG7LjjXkG2mNRoXQdWtrZvP6XBsCyHboeuW7H/
    attestationPublicKey: <attest-pubkey>
```

⇒ where the output of the following command replaces `<attest-pubkey>`.

```
IFS=$'\n'; echo -n \""; for line in $(cat attest-public-key.pem);\
do echo -n $line"\n"; done; echo \"
```

3. Deploy the workload either in IBM Cloud or on-premises.
4. Open your Web browser to <https://<ip address>:8443//api/v1/attestation> to access the encrypted attestation document.
Replace `<ip address>` with the floating IP address noted earlier for the VPC deployment in IBM Cloud or the IP address assigned to the HPCR VM running on the KVM host.
5. Save the encrypted attestation document to *se-checksums.txt.enc*.
The file can be saved on the local system where the Docker Compose file is located.
6. Use the following script to decrypt the encrypted attestation document.

```
#!/bin/bash
#
# Example script to decrypt attestation document.
```

```
#
# Usage:
#   ./decrypt-attestation.sh <rsa-priv-key.pem> [file]
#
# Token Format:
#   hyper-protect-basic.<ENC_AES_KEY_BASE64>.<ENC_MESSAGE_BASE64>

RSA_PRIV_KEY="$1"
if [ -z "$RSA_PRIV_KEY" ]; then
    echo "Usage: $0 <rsa-priv-key.pem>"
    exit 1
fi
INPUT_FILE="${2:-se-checksums.txt.enc}"
TMP_DIR="$(mktemp -d)"
#trap 'rm -r $TMP_DIR' EXIT

PASSWORD_ENC="${TMP_DIR}/password_enc"
MESSAGE_ENC="${TMP_DIR}/message_enc"

# extract encrypted AES key and encrypted message
cut -d. -f 2 "$INPUT_FILE" | base64 -d > "$PASSWORD_ENC"
cut -d. -f 3 "$INPUT_FILE" | base64 -d > "$MESSAGE_ENC"

# decrypt password
PASSWORD=$(openssl rsautl -decrypt -inkey "$RSA_PRIV_KEY" -in "$PASSWORD_ENC")

# decrypt message
echo -n "$PASSWORD" | openssl aes-256-cbc -d -pbkdf2 -in "$MESSAGE_ENC" -pass stdin
--out se-checksums.txt
```

7. Verify the integrity of workload contract.

- a. Generate an SHA256 checksum of the local contract file.

```
sha256sum slebci-paynow-website-contract.yml
```

- b. Compare the output of the above command with the user-data line in the *se-checksums.txt* attestation document.

If the checksums match, then the contract used to start the workload is same as its source.



Tip

A similar verification process can be done for the other items listed in the attestation document.

8 Summary

Containerized confidential computing enables you to protect your workload data no matter where it is running. This guide featured the PayNow Web site application. It was built on SUSE Linux Enterprise Base Container Images and you securely deployed it on the IBM Hyper Protect Platform, either in the cloud or on-premises.


Continue your learning journey with the following additional resources:

- Accelerate Application Development with Open, Secure Containers with [SUSE Linux Enterprise Base Container Images](https://www.suse.com/products/base-container-images/) (<https://www.suse.com/products/base-container-images/>) .
- Build a trusted PayNow Web site container on SUSE Linux Enterprise Base Container Images using [IBM Hyper Protect Platform Secure Build](https://cloud.ibm.com/docs/vpc?topic=vpc-about-hpsb) (<https://cloud.ibm.com/docs/vpc?topic=vpc-about-hpsb>) .
- Deploy multiple containers using the [play](https://www.ibm.com/docs/en/hpvs/2.1.x?topic=notes-whats-new-in-version-215#deploying-multiple-containers) (<https://www.ibm.com/docs/en/hpvs/2.1.x?topic=notes-whats-new-in-version-215#deploying-multiple-containers>)  subsection, which explains a new HPCR capability that was recently added for [IBM Cloud](https://cloud.ibm.com/docs/vpc?topic=vpc-about-contract_se#hpcr_contract_play) (https://cloud.ibm.com/docs/vpc?topic=vpc-about-contract_se#hpcr_contract_play)  and on-premises (https://www.ibm.com/docs/en/hpvs/2.1.x?topic=servers-about-contract#hpcr_contract_play)  deployments.
- Confidential computing with [LinuxONE](https://cloud.ibm.com/docs/vpc?topic=vpc-about-se) (<https://cloud.ibm.com/docs/vpc?topic=vpc-about-se>)  in IBM Cloud.
- [IBM Hyper Protect Virtual Servers 2.1.x](https://community.ibm.com/community/user/ibmz-and-linuxone/blogs/nicolas-mding/2022/10/06/ibm-hyper-protect-virtual-servers-v21) (<https://community.ibm.com/community/user/ibmz-and-linuxone/blogs/nicolas-mding/2022/10/06/ibm-hyper-protect-virtual-servers-v21>) .

9 Legal notice

Copyright © 2006–2025 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled "GNU Free Documentation License".

SUSE, the SUSE logo and YaST are registered trademarks of SUSE LLC in the United States and other countries. For SUSE trademarks, see <https://www.suse.com/company/legal/> .

Linux is a registered trademark of Linus Torvalds. All other names or trademarks mentioned in this document may be trademarks or registered trademarks of their respective owners.

Documents published as part of the series SUSE Technical Reference Documentation have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

10 GNU Free Documentation License

Copyright © 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition. The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.

- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all

Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2

```
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “ with... Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.