SUSE Public Cloud Infrastructure Setup Guide

SUSE Linux Enterprise Server, SUSE Manager, SUSE Cloud, SUSE OpenStack Cloud

Robert Schweikert, Public Cloud Architect, SUSE

This guide describes the setup of the required infrastructure for a SUSE public cloud in generic terms that scale from small to large cloud installations. The implementation of this setup is a requirement to obtain SUSE Certified Cloud Provider status. Additionally this setup can also be implemented in a private cloud setup.

Publication Date: 6 June 2016

Contents

1 Introduction

Use of public cloud resources is one of the fastest growing areas of the IT industry. In many cases Infrastructure as a Service (IaaS) is a leading use case of public cloud. The public cloud brings with it a “start and use” expectation. This poses a challenge for an Enterprise Linux distribution such as SUSE Linux Enterprise to provide access to the update repositories without formal registration of the OS installation.

In a traditional data center a SUSE customer will set up a new machine (physical or virtual) and configure the system to connect to a local SMT server, be managed by SUSE Manager, or connect to the SUSE Customer Center (SCC), for SUSE Linux Enterprise 12 based releases, or to the Novell Customer Center (NCC), for SUSE Linux Enterprise 11 based releases to obtain updates. In the public cloud the data center workflow is not practical.

Generating registration entitlements for every instance for use with SCC/NCC and providing these to the user would not meet the “fire up and use” expectation. Automated generation of entitlements and injection into instances, while possible, would provide registration with SCC/NCC. But it may generate additional cost for the customer as packages would need to be downloaded from outside the cloud provider`s network for update purposes. Additionally a direct registration of instances with SCC/NCC would put the burden of verification concerning eligibility into the SCC/NCC code base. For NCC this is not possible and for SCC this is undesirable. Therefore, in a public cloud setting it is important that the update infrastructure be operated inside the public cloud environment.

This guide describes the setup of the required infrastructure in generic terms that scale from small to large cloud installations. The implementation of this setup is a requirement to obtain SUSE Certified Cloud Provider status. Additionally this setup can also be implemented in a private cloud setup.

2 High Level Overview

The update infrastructure consists of 2 major components:

- Region Servers
- SMT Servers

Both run as VMs within the cloud framework. All services run on SUSE Linux Enterprise Server 12 SP1 or higher. These systems may be registered directly to SCC or may be managed using SUSE Manager. In addition to having the base repository the systems must also have the Public Cloud Module repository configured.

2.1 Region Server(s)

The Region Server uses information provided in a configuration file, `/etc/regionService/regionData.cfg` by default, to associate SMT server information with a particular region of the cloud framework. If the cloud service offered has only one region it is still useful to use the region server to allow for easy future expansion. The goal of the overall architecture is to always deliver updates for SUSE Linux Enterprise guests region local. In general, IP addresses for guests are provided by a DHCP service that is part of the cloud framework, and various CIDR blocks are used in various regions. This region association of IP addresses, IPv4 and/or IPv6, is part of the cloud framework configuration and is captured in the configuration file for the region server. For cases where the association of CIDR blocks to specific regions is more dynamic the Region Server REST API also accepts a `regionHint` argument that the guest may send to obtain information about the SMT servers in the guest's region.

SUSE Linux Enterprise Server guest instances connect to a Region Server to receive a list of SMT Servers available in the region in which the guest instance was launched. The information provided to the client is in XML format and is sufficient for the client to automatically register with the region local SMT Server.

Generally, multiple Region Server instances should be operated in the cloud environment to ensure availability of the Region Service if any Region Server is too distant (high latency), down, or otherwise unavailable. Information about the Region Servers in the cloud framework is encoded in the guest images by including the region server's public certificate in the `/var/lib/regionService/certs/` directory. The name of the certificate files is used to attempt to connect to the Region Server via HTTPS.

2.2 SMT Server(s)

The SMT (Subscription Management Tool) Server serves as cache for the package repositories obtained from SCC (SUSE Customer Center). The SMT Server itself is registered with SCC, or managed via SUSE Manager, as it would be in a traditional data center.

Given the data provided by a Region Server, the client proceeds through a “regular” automated registration process. This registration process is identical to the process an administrator in a data center would complete when registering a new system against an SMT Server operated in a traditional data center.

3 Detailed Setup Guide

Although the Region Server is the first service used by a client, its setup and configuration is dependent on the setup of SMT Servers. Therefore, the setup guide will describe the setup in reverse order as compared to the previous section.

3.1 General Setup

Before any of the servers are set up and configured some general preparations should be completed. Access restriction to the servers is a multi-level process as described in more detail later. Generally cloud frameworks provide a feature often called “Security Groups”. This is a firewalling mechanism that the cloud framework network infrastructure provides.

Use this mechanism to set up two security groups: one used by the SMT Servers and one used for the Region Service. The security group (firewall rules) for the SMT Server (suggested name: smt-server) needs to allow incoming traffic on ports 22, 80, and 443 from all addresses (`0.0.0.0/0`; `::/0`). All other ports should be blocked. The security group (firewall rules) for the Region Service (suggested name: region-server) needs to allow incoming traffic on ports 22 and 443 from all addresses (`0.0.0.0/0`; `::/0`). If you so desire you can move SSH to run on a different port and configure your firewall rules accordingly.

Allocate static IP addresses in the cloud framework. For each region there should be at least two SMT Servers. Additionally there should be at least two Region Servers. Depending on the footprint of the cloud environment more Region Servers, with instances running in different regions, may be desired. Thus, allocate at least two static IPs per region plus one or more static IPs in each region where a Region Server will be set up.

The final preparatory step is to generate SSH key pairs for the servers. Most cloud frameworks provide a way to generate keys or to import keys. The SSH key does not need to be built into the image as in general it will be injected by the cloud framework on instance launch. It is recommended to use different keys for the SMT Server and the Region Server.

```
ssh-keygen -t rsa -f smt
ssh-keygen -t rsa -f regionsrv
```

3.2 SMT Server Setup

The SMT Server is a VM inside the cloud environment. For each region at least two SMT Servers should be configured. The number of SMT Servers in a region depends on the bandwidth within the data center and the number of expected simultaneous users. As a reference, SUSE operates two SMT Servers per region in AWS and can satisfy throughput needs for registered clients. Thus, it is unlikely that more than two SMT Servers are needed in your setup. The setup of an SMT Server inside a virtual machine is no different than the setup of an SMT Server on a physical machine. Therefore you can refer to the https://www.suse.com/documentation/sles-12/book_smt/data/book_smt.html SMT documentation for guidance. Alternatively you can build an image with kiwi; a template used to build SUSE-operated SMT Servers is attached in appendix A.

The SMT Server functions as a cache for all packages SUSE releases as updates for all enabled products. Therefore, it potentially requires a large amount of disk space. It is recommended that a 1TB virtual disk (or larger) be created inside the cloud environment and attached to the SMT Server VM. This storage device will be used to store the repositories. In addition it is recommended to store the database on an attached virtual disk; a size of 40 GB or more is suggested for the DB storage device.

Having the storage of the repositories on a separate device preserves the data should the SMT server VM need to be rebuilt. This improves the recovery time since the repositories do not need to be repopulated from SCC. Keeping the DB on a separate device ensures that the existing registrations do not get lost if the SMT VM needs to be rebuilt.

After the installation of the SMT Server package (*smt*) the directory structure required by SMT is set up in `/srv/www/htdocs/repo`. We want to use this directory as the mount point for the external virtual disk that is supposed to hold the repositories. The following process outlines the steps necessary to set the external device up to hold the repositories:

1. In the cloud framework, either via the customary web-UI, or the available command line tools create a virtual disk of 1 TB or larger.
2. After the virtual disk creation is complete attach the device to the running SMT instance.
3. When the disk is attached log in to the SMT Server as root, or become root after logging in.
4. Use **`fdisk -l`** to list the available block devices.

5. Create a partition table (using YaST, gparted, or fdisk) and create one partition on the device.

6. Create a file system on the newly created partition; XFS is the recommended file system for this storage device.

7. Copy the content of the SMT directory to a “safe” place:

```
mkdir /tmp/smtData; rsync -av /srv/www/htdocs/repo/ /tmp/smtData
```

8. Mount the storage partition:

```
mount /dev/sd? /srv/www/htdocs/repo/
```

9. Restore the content of the repo directory:

```
rsync -av /tmp/smtData/ /srv/www/htdocs/repo; rm -rf /tmp/smtData
```

Note that `/srv/www/htdocs/repo` and its content must be owned by the `smt` user and have `www` group ownership. The procedure for placing the DB data onto a separate device is the same.

1. In the cloud framework, either via the customary web-UI, or the available command line tools create a virtual disk of 40 GB.

2. After the virtual disk creation is complete attach the device to the running SMT instance.

3. When the disk is attached log in to the SMT Server as root, or become root after logging in.

4. Use **`fdisk -l`** to list the available block devices.

5. Create a partition table (using YaST, gparted, or fdisk) and create one partition on the device.

6. Create a file system on the newly created partition; XFS is the recommended file system for this storage device.

7. Copy the content of the DB directory to a “safe” place:

```
mkdir /tmp/dbData; rsync -av /var/lib/mysql/ /tmp/dbData
```

8. Mount the storage partition:

```
mount /dev/sd? /var/lib/mysql
```

9. Restore the content of the repo directory:

```
rsync -av /tmp/dbData/ /var/lib/mysql; rm -rf /tmp/dbData
```

The process of mounting the external devices upon instance start-up is cloud framework specific. Thus it depends on the cloud framework whether you want to create entries for the mount points in `/etc/fstab`.

At the protocol level several functions of SMT for the client registration use HTTPS, that is these are SSL dependent. Therefore, certificate handling is important. In general the assumption is that an SMT Server uses a self signed certificate. If you choose to follow the path of using a self signed certificate you should use the integrated certificate creation step included in the SMT setup process in YaST. This process will be followed later after some more preliminary settings are complete. However, at your choice you may use a certificate signed by a public CA. In this case you need to handle certificate placement manually. In cases where SUSE operates an infrastructure self signed certificates are used.

If you choose to use a certificate that is signed by a public CA and is thus acceptable by using the known trust chain, it is still necessary that the `/srv/www/htdocs/smt.crt` file exists. This file, in a self signed certificate setup, represents the root CA and is imported into the trust chain on the guest that is registering with the SMT server. The registration automation downloads this file and registration will fail if it does not exist. Thus, even for certificates signed by a public CA the root CA must be made available via `/srv/www/htdocs/smt.crt`.

The next step for the SMT server configuration is to set up registration sharing. Registration sharing is enabled by the installation of the *smt-ha* package and is used to configure the SMT servers in an active HA configuration. Registration sharing can be configured between two or more SMT Servers. The certificate setup has an implication on the configuration of registration sharing between multiple SMT Servers as well. Registration sharing is configured in the SMT configuration file `/etc/smt.conf`. In the `[LOCAL]` section add the following configuration entries:

```
#
# This string is used to verify that any sender trying to share a
# registration is allowed to do so. Provide a comma separated list of
# names or IP addresses.
acceptRegistrationSharing=
```

```
#
# This string is used to set the host names and or IP addresses of sibling
# SMT servers to which the registration data should be sent. For multiple
# siblings provide a comma separated list.
shareRegistrations=
#
# This string provides information for SSL verification of the siblings.
# Certificates for the siblings should reside in the given directory.
# If not defined siblings are assumed to have the same CA as this server
siblingCertDir=
```

When a client system registers with the SMT server the SMT implementation will send a "share registration request" to all the servers configured with the *shareRegistrations* option. The value is a comma separated list of IP addresses or host names. Note that the entries must match the way the certificate was set up as the "share registration request" is sent to the other SMT server(s) via HTTPS. So if you encoded the FQDN of the sibling SMT servers in the certificate this name must be used as a value of the *shareRegistration* configuration option. If the certificate of the sibling server(s) is not imported into the default trust chain you may set the *siblingCertDir* option to point the verification code to the directory where the sibling server certificates are located.

When a "share registration request" is received by an SMT Server the list of IP addresses and/or host names (comma separated string) provided with the `acceptRegistrationSharing` setting is consulted to verify that the sending system is authorized to share a registration. This share request is not propagated to any SMT Servers that may be listed in the `shareRegistration` option. Only registration requests received from clients using the **`suse_register`** or **`SUSEConnect`** commands are shared with SMT Servers listed with the *shareRegistrations* setting.

Sharing of registration information is important to provide failover capabilities for the update infrastructure.

Before proceeding with the configuration for repository mirroring, access control to the SMT server(s) needs to be configured. The SMT Server in a public cloud is by definition public, or it could not be accessed by the instances of SUSE Linux Enterprise that run within the cloud framework. Therefore, access to the SMT Server must be restricted to eligible SUSE Linux Enterprise installations. Eligible SUSE Linux Enterprise installations are defined to be those for which the cloud provider can account and report accurate usage hours to SUSE. The level of access control behind the security group setting described earlier (which is considered level 1) occurs on two levels:

- HTTP and HTTPS traffic access
- repository access verification

Depending on the offering of SUSE Linux Enterprise in the cloud framework repository, access verification may not be necessary.

Access to the server on the protocol level needs to be restricted to IP addresses (IPv4 and/or IPv6) that are handed out by the cloud framework DHCP server(s). This process can be automated using the utility provided by the `python-serviceAccessConfig` package. The service provided with this package generates ACL information for Apache. More details about the configuration of this service are provided in the *Server Access Control* section below. You may also consult the man page, `man serviceAccessConfig` for details about the configuration. The file `/etc/smt.d/nu_server.conf` can be used as a target for the ACL generation to control access to the SMT functionality. However a higher level Apache file can be used to block access from outside IP addresses to Apache altogether.

If only on-demand images of SUSE Linux Enterprise are offered in the cloud framework it is not necessary to implement a repository access verification mechanism. However, if BYOS (Bring Your Own Subscription) is offered within the cloud framework a repository access mechanism needs to be implemented. This is accomplished by adding

```
#
# This string is used to load a cloud specific verification module to verify
# the guest issuing the registration request is eligible to access the
# repositories. The value none indicates that no verification should
# take place.
cloudGuestVerify=PERL_PLUGIN_NAME
```

to the `/etc/smt.conf` file in the `[LOCAL]` section. The PERL_PLUGIN_NAME in the above example is a placeholder for a Perl module that needs to be implemented to verify that the guest is eligible to access the repositories. This plug-in may perform verification based on accessing a cloud framework API or by verifying metadata sent by the guest, or a combination thereof.

The verification module must implement two interfaces, *verifyGuest* and *verifySCCGuest*. The *verifyGuest* implementation is called when a client accesses the SMT Server via *suse_register* for the first time and the *verifySCCGuest* implementation is used when the client uses *SUSEConnect* for registration purposes.

suse_register is the implementation shipped with SUSE Linux Enterprise 11 to register a SUSE Linux Enterprise installation with NCC (Novell Customer Center) or SMT. **`SUSEConnect`** was developed to interface with SCC (SUSE Customer Center) and SMT and is shipped with SUSE Linux Enterprise 12.

The implementation of the access verification plug-in is placed in a file named to match the configuration option, PERL_PLUGIN_NAME.pm for the example above, that is placed in the `/srv/www/perl-lib/SMT/Client` directory. The following example shows a stub implementation of the verification module that also dumps the received data:

EXAMPLE 1: `/srv/www/perl-lib/SMT/Client/exampleVerify.pm`

```
use strict;
use warnings;

use Apache2::RequestRec ();
use Apache2::RequestIO ();

use Data::Dumper;

sub verifyGuest {

    my $self    = shift;
    my $r       = shift;
    my $regroot = shift;
    # Insert code to connect to cloud framework and verify the guest
    # return 1 for successful verification, undef for verification failure
    # $r       -> the request, i.e. an Apache request object
    #             http://perl.apache.org/docs/2.0/api/Apache2/RequestRec.html
    # $regroot -> HASHREF containing information sent by the client.
    open(my $DBGOUT, '>', '/tmp/verifyGuest.txt');
    print $DBGOUT "The client information\n\n";
    my $dumper = Data::Dumper->new([$regroot]);
    print $DBGOUT $dumper->Dump();
    return 1;
}

sub verifySCCGuest {

    my $self     = shift;
    my $r        = shift;
    my $clntData = shift;
    my $result   = shift;
    # Insert code to connect to cloud framework and verify the guest
    # return the result HASHREF for successful verification, undef for
    # verification failure
    # $r        -> the request, i.e an Apache request object
    #              http://perl.apache.org/docs/2.0/api/Apache2/RequestRec.html
    # $clntData -> data received from the client
```

```
    # $result   -> HASHREF of results of various previous operations
    open(my $DBGOUT, '>', '/tmp/sccVerifyGuest.txt');
    print $DBGOUT "The client information\n\n";
    my $dumper = Data::Dumper->new([$clntData]);
    print $DBGOUT $dumper->Dump();
    my $dd = Data::Dumper->new([$result]);
    print $DBGOUT $dd->Dump();
    return $result;
}

1;
```

A similar file is provided as part of the *smt-ha* package.

This completes the configuration of the access control of the SMT server. In summary the access control of the SMT server occurs on potentially three levels:

Level 1

Access control at the cloud framework level, via firewall opening ports 22, 80, and 443.

Level 2

Access control at the protocol level allowing only systems with IP addresses used by the cloud framework to access the server. This can be accomplished using the `serviceAccessConfig` tool.

Level 3

Repository access verification via SMT plug-in. This is only necessary if BYOS images are supported in the cloud framework.

The final modifications to the `etc/smt.conf` file prior to registering SMT with SCC and configuring the repositories are to set the authentication and the forwarding information. In `etc/smt.conf` set the `forwardRegistration` setting to `false` and set the `requiredAuthType` to `lazy`.

You are now ready to complete the higher level SMT setup and eventually synchronize the repository content from SCC. Use YaST to configure the remaining settings of SMT following the SMT setup documentation. As part of this process, if you are using a self signed certificate, the certificate creation workflow can be invoked.

The final step is to set up the repositories that should be mirrored. This can be completed with YaST or the **smt-repos** command. Mirror the repositories `Pool`, and `Update` for the base distributions SLES 11 and SLES 12, as well as the `Debuginfo` and `SDK` repositories. In addition both SLES 11 and SLES 12 offer specific module repositories that are considered to be part of SLES but are delivered as separate repositories. These should also be enabled for mirroring.

You only need to mirror the repositories for the architecture of interest to you. In most cases this will be only x86_64. There is generally no advantage in a cloud environment to run 32-bit instances as the underlying hardware is generally 64-bit capable and 64-bit instances are fully capable of running 32-bit applications. If you need to mirror more than one architecture you may need to increase the size of the storage that holds the packages.

The final step in the SMT setup is to mirror the repositories from SCC using the *smt-mirror* command as root. This will download all the packages for the configured repositories and takes some time. Usually the synchronization will complete overnight.

3.3 SMT Server Monitoring

The SMT servers should be monitored for health status. In addition to the general statistics

- CPU Load/Utilization
- Memory Usage
- Disk I/O
- Kernel Health
- File system space
- Uptime

the following SMT Server specific system functions and files should be monitored:

- /etc/apache2/conf.d/nu_server.conf - monitor for presence and changes
- /etc/serviceaccess/srvAccess.cfg - monitor for presence
- /etc/regionService/regionData.cfg - monitor for presence
- /srv/www/htdocs/smt.crt - monitor for presence and changes
- mariadb - process running

- serviceAccessConfig - process running
- apache - process running
- Mount points - DB and repositories

3.4 Region Server Setup

The function of the Region Server is to provide information about the SMT Servers in a given region to the connecting guest. The Region Server runs as a Python script using the Flask framework in Apache. The Region Server is provided with the `cloud-regionsrv` package. As with the SMT server the region server should be access restricted to the IP address ranges for the cloud framework. More on this access control in the section *Section 3.6, "Server Access Control"*.

The service itself uses two configuration files: one file, `/etc/regionService/regionInfo.cfg` is used to configure the service, the other, `/etc/regionService/regionData.cfg` provides the data the Region Server will provide to the connecting client; both files are in the ini format.

The `regionInfo.cfg` file is used to configure the location of the log file and the location of the `regionData.cfg` file with the `logFile` and `regionConfig` options, respectively. The default settings for the configuration file are shown below and should suffice for most installations.

```
[server]
logFile = /var/log/regionService/regionInfo.log
regionConfig = /etc/regionService/regionData.cfg
```

The default `regionData.cfg` file provides a template for the configuration file. This file can be maintained manually or be auto-generated, depending on your setup for IP address allocation within your cloud framework. The `regionData.cfg` file needs to contain one configuration section per region. Often regions in a given cloud setup are named by geographic location, such as "us-east-1" for the Amazon cloud setup in the eastern geography of the United States. This location indicator is also often available through querying of metadata in the guest image. It is therefore recommended that the section name in the `regionData.cfg` file match the names of the region names available through the metadata of the framework. In cases where IP addresses are not stable in their association with regions the region metadata can be queried in the guest and passed to the Region Server as a hint to obtain the SMT Server information for the given region. The hint is processed with string matching and thus having section names match the configured region names is important. The server implementation has no option of name mapping. For each region all options in the section must be configured.

The section options are as follows:

public-ips

The value for this option is a comma separated list of IP ranges in CIDR format, for example:

```
public-ips = 62.135.16.0/18,56.56.130.0/16
```

These are the ranges the DHCP server in the given region is configured to use. If the access configuration utility **`python-serviceAccessConfig`** is used and pointed at this file it will use the `public-ips` entry to generate the ACL.

smt-server-ip

The value for this option is a comma separated list of the SMT Server IP addresses in the region being configured.

smt-server-name

The value for this option sets the host name of the SMT Server that was encoded into the certificate during the setup of the SMT Server. If only one value is supplied it will be used for all IP addresses provided by the `smt-server-ip` setting. If more than one value is supplied the number of names must match the number of IP addresses given with the `smt-server-ip` option. The order of the names and IP addresses must match as well.

smt-fingerprint

The value for this option is the fingerprint of the root CA created during the SMT Server setup. On the SMT Server, the `/srv/www/htdocs` includes the `smt.crt` file, which is the root CA transferred to the client and verified prior to accepting the repositories from the SMT Server. Obtain the fingerprint with the command:

```
openssl x509 -in smt.crt -noout -fingerprint | /usr/bin/cut -d= -f2
```

Use this fingerprint for the `smt-fingerprint` value. As with the `smt-server-name` supplying one value is sufficient if all SMT Servers have the same root CA. If each server has its own CA supply a comma separated list. The order must match the order of the IP addresses or certificate acceptance will fail and the guest cannot register with the SMT Server.

The following shows an example of a completed section for a region in a cloud setup.

```
[nor-north]
public-ips = 62.135.16.0/18,56.56.130.0/16
smt-server-ip = 62.153.16.20,56.56.130.253
```

```
smt-server-name = smt-nor.supertuxcloud.com
smt-fingerprint = 9D:B9:88:DB:87:52:00:55:F0:FF:5D:5C:66:60:D3:E0:5C:D4:FB:79
```

In the example above both SMT Servers share the same certificate. If this were not the case another value for the `smt-server-name` and for the `smt-fingerprint` options would need to be configured.

```
[mid-north]
public-ips = 62.135.16.0/18,56.56.130.0/16
smt-server-ip = 62.153.16.20,56.56.130.253
smt-server-name = smt-mid-a.supertuxcloud.com, smt-mid-b.supertuxcloud.com
smt-fingerprint = 9D:B9:88:DB:87:52:00:55:F0:FF:5D:5C:66:60:D3:E0:5C:D4:FB:79
```

In this example the servers share the same certificate, but have different names. The certificate in this case would contain a wild card.

The Region Server reads the `regionData.cfg` file as configured in the `regionInfo.cfg` file at start-up and creates a hash table from the information provided in the `regionData.cfg` file. Depending on the number of IP address ranges this may result in a relatively large requirement of memory. For estimation purposes one can use a requirement of 20 MB per octet (254 IP addresses). The Region Server also creates a secondary hash table that relates the region names to the SMT Server information. The secondary hash table is consulted if the Region Server receives a region hint from the client.

The Region Server provides the `regionInfo` REST API, that is to obtain SMT information the client image will access the Region Server via: https://IP_ADDRESS_OF_REGION_SERVER/regionInfo

The knowledge of the IP addresses of the Region Servers and the certificates for the Region Servers are built into the guest image. It is recommended to create a package for the Region Service client; for more details see the section concerning the guest image.

As mentioned previously, for environments where the IP address assignment per region is not stable or accessible via API the option exists to let the client pass a region hint to the Region Server. If the region hint is passed the Region Server will first try to provide SMT information to the client based on the region name given in the hint. If this fails the Region Server will fall back to using the IP address. A request using the region hint option provides the region hint as an argument with the URL: https://IP_ADDRESS_OF_REGION_SERVER/regionInfo?regionHint=REGION_NAME

In this case *REGION_NAME* must match a name of one of the sections in the `regionData.cfg` file as indicated previously.

Instead of using IP addresses directly for the Region Servers in the client it is also possible to use name resolution via DNS. One potential advantage of using DNS for the Region Server is that no new package would need to be released if a Region Server needs to move out of a given region and into a new region where a new IP address would need to be allocated. However, this implies that an entire cloud region would be shut down, which is a very unlikely scenario. The Region Server package (`cloud-regionsrv`) provides a convenient executable to generate the server certificate. After the region server instance is booted run:

```
genRegionServerCert -c COUNTRY -d DEPARTMENT --host IP_ADDRESS_OR_HOSTNAME -l
 LOCATION -o ORGANIZATION -s STATE
```

This will generate the server certificate and place the public cert into `/root/regionServCert/`. This certificate needs to be included in the SUSE Linux Enterprise guest image. The details about the Region Server certificate are described in the guest image creation section. The cert generation script will restart the Apache Web server.

With the configuration in place and the certificate generated the Region Server setup is complete.

3.5 Region Server Monitoring

The Region servers should be monitored for health status. In addition to the general statistics

- CPU Load/Utilization
- Memory Usage
- Disk I/O
- Kernel Health
- File system space
- Uptime

the following SMT Server specific system functions and files should be monitored:

- /etc/apache2/vhosts.d/regionsrv_vhost.conf - monitor for presence
- /etc/serviceaccess/srvAccess.cfg - monitor for presence
- /etc/regionService/regionData.cfg - monitor for presence

- serviceAccessConfig - process running
- apache - process running

3.6 Server Access Control

Both the Region Server and the SMT server should be configured to have ACL level control to allow access to the services only from the IP ranges that are used by the DHCP servers of the cloud framework. This configuration can be set up manually or it can be automated using the `python-serviceAccessConfig` tool.

For both the SMT server and the Region Server, access control has two layers: the firewall rule and the ACL rules as described earlier. The SMT server may also implement a third layer. The `python-serviceAccessConfig` generates ACL rules for Apache and can be pointed to the `regionData.cfg` file to collect all the public IP addresses used by the cloud framework. This set of CIDR blocks is used to generate the access control rules for Apache. Therefore, it is recommended that the same `/etc/regionService/regionData.cfg` exist on SMT and Region servers.

The `serviceAccessConfig` process monitors the `/etc/regionServer/regionData.cfg` file and generates new access rules when changes are detected. It is therefore possible to fully automate the access rule generation by implementing a generator for `regionData.cfg` that may parse the DHCP rules and then push out a new file to all servers when the rules change, for example.

The `serviceAccessConfig` process is configured with the `/etc/cloudServiceAccess/srvAccess.cfg` configuration file; consult the man page for detailed information.

3.7 Guest Image

The content of the guest image, that is the image that provides the basis for the instances CSP customers use is customizable to the desires of the cloud framework provider.

For an image to be considered supportable the image must have at least the so-called `Minimal Pattern` installed. In addition to this a cloud image generally requires some initialization code that handles SSH key injection, account creation, and other housekeeping tasks. This initialization code can be *cloud-init*, an open source solution found in the Public Cloud Module repository, or some other initialization implementation.

The package `cloud-regionsrv-client` provides the necessary executable to handle automated guest image registration with the update infrastructure. The client code is configured with the `/etc/regionserverclnt.cfg` file. The configuration file has two sections with the options shown below:

```
[server]
api = regionInfo
certLocation = /var/lib/regionService/certs
regionsrv = COMMA_SEP_LIST_OF_CLOUD_SPECIFIC_REGION_SERVER

[instance]
dataProvider = none
instanceArgs = none
```

The `[server]` section and all options are mandatory.

api

The value of the api determines the API the client should call. In a "standard" setup this will be set to `regionInfo` (as shown in the example above).

certLocation

The value of this option indicates the location of the server certificates for the Region Server(s). These are the public certs that were generated with the **`genRegionServerCert`** command shown earlier during Region Server setup. All certificates from the region servers are collected in the specified location.

regionsrv

The value is a comma separated list of the names or IP addresses of the Region Servers. The names listed here must match with the certificate names, without the .pem extension. The client code will randomize the list to distribute the access load across the Region Servers and will then try to obtain the SMT information in the order the list randomization produced. Generally it is expected that SMT information is provided by the first region server contacted.

The `[instance]` configuration section is optional.

dataProvider

Specifies a command that generates data passed to the SMT Server. For verification purposes it may be necessary to collect information from the instance and pass it to the SMT Server. This command must produce data in the format expected by the repository access verification plug-in discussed in *Section 2.2, "SMT Server(s)"*.

instanceArgs

The value of this option specifies the name of a plug-in to load that will provide the information for the `regionHint` passed to the Region Server. The name provided must match the file name without the `.py` extension.

The `instanceArgs` plug-in needs to be implemented in Python and be located in the `python-sitelib` path in the `cloudregister` directory. The plug-in must implement the `generateRegionSrvArgs` function. No arguments are passed to the function and the return value is expected to be a string that will match a section in the `regionData.cfg` file. The return value of the `generateRegionSrvArgs` is expected to be a string and provides the value for the `regionHint` argument the region server accepts.

The format of the guest image is cloud framework dependent and dictated by the hypervisor used by the cloud framework. KIWI (https://doc.opensuse.org/projects/kiwi/doc/), the open source, SUSE sponsored image build tool can produce images in almost any format required by the known hypervisors. KIWI can also produce so-called OEM images that can be used for bare metal deployment. Images can also be created with SUSE Studio (https://susestudio.com). Other means of image creation are possible and feasible; ultimately it is the CSP's responsibility to create images that boot and perform within the CSP's cloud framework.

3.8 Appendix A

```
<?xml version="1.0" encoding="utf-8"?>
<image schemaversion="6.2" name="SMTServer" displayname="SMTServer">
    <description type="system">
        <author>Public Cloud Team</author>
        <contact></contact>
        <specification>SUSE Linux Enterprise Server 12 SPX SMT Server image</
specification>
    </description>
    <preferences>
        <type image="oem" boot="oemboot/suse-SLES12" filesystem="ext4"
 boottimeout="1" bootloader="grub2">
```

```
            <size unit="M">8192</size>
            <oemconfig>
                <oem-swap>false</oem-swap>
            </oemconfig>
        </type>
        <version>0.1.4</version>
        <packagemanager>zypper</packagemanager>
        <rpm-check-signatures>false</rpm-check-signatures>
        <locale>en_US</locale>
        <keytable>us.map.gz</keytable>
        <hwclock>utc</hwclock>
        <timezone>utc</timezone>
    </preferences>
    <users group="root">
        <user password="linux" pwdformat="plain" home="/root" name="root"/>
    </users>
    <!-- Repository definition goes here -->
    <packages type="image">
        <!-- jeos server -->
        <package name="patterns-sles-Minimal"/>
        <package name="dhcp-client"/>
        <package name="fontconfig"/>
        <package name="fonts-config"/>
        <package name="grub2"/>
        <package name="iproute2"/>
        <package name="iputils"/>
        <package name="lvm2"/>
        <package name="openssh"/>
        <package name="parted"/>
        <package name="psmisc"/>
        <package name="rsync"/>
        <package name="syslinux"/>
        <package name="systemd"/>
        <package name="systemd-sysvinit"/>
        <package name="sudo"/>
        <package name="tar"/>
        <package name="vim"/>
        <package name="which"/>
        <!-- end jeos server -->
        <!-- basic functionality -->
        <package name="at"/>
        <package name="attr"/>
        <package name="audit"/>
        <package name="autofs"/>
        <package name="bc"/>
```

```
        <package name="binutils"/>
        <package name="blktrace"/>
        <package name="command-not-found"/>
        <package name="crash"/>
        <package name="cryptconfig"/>
        <package name="curl"/>
        <!-- Authentication functionality -->
        <package name="cyrus-sasl"/>
        <package name="cyrus-sasl-digestmd5"/>
        <package name="cyrus-sasl-gssapi"/>
        <package name="cyrus-sasl-plain"/>
        <package name="cyrus-sasl-saslauthd"/>
        <!-- Authentication functionality end -->
        <package name="deltarpm"/>
        <package name="dos2unix"/>
        <package name="dosfstools"/>
        <package name="ethtool"/>
        <package name="expect"/>
        <package name="fping"/>
        <package name="glibc-i18ndata"/>
        <package name="haveged"/>
        <package name="icmpinfo"/>
        <package name="irqbalance"/>
        <package name="kernel-default"/>
        <package name="klogd"/>
        <package name="ksh"/>
        <package name="libnl1"/>
        <!-- netlink protocol support -->
        <package name="libnettle4"/>
        <!-- used by gpg -->
        <package name="lockdev"/>
        <package name="man"/>
        <package name="man-pages"/>
        <package name="mozilla-nss-certs"/>
        <package name="netcat-openbsd"/>
        <package name="nfsidmap"/>
        <package name="nscd"/>
        <package name="ntp"/>
        <package name="openldap2-client"/>
        <package name="opie"/>
        <package name="pam-modules"/>
        <package name="polkit-default-privs"/>
        <package name="prctl"/>
        <package name="procinfo"/>
        <package name="quota"/>
```

```
        <package name="recode"/>
        <package name="rsh"/>
        <package name="screen"/>
        <package name="strace"/>
        <package name="supportutils"/>
        <package name="SUSEConnect"/>
        <package name="SuSEfirewall2"/>
        <package name="suse-build-key"/>
        <package name="tcpd"/>
        <package name="tcpdump"/>
        <package name="tcsh"/>
        <package name="telnet"/>
        <package name="terminfo"/>
        <package name="vlock"/>
        <package name="wget"/>
        <package name="x86info"/>
        <package name="xfsprogs"/>
        <package name="xinetd"/>
        <package name="zip"/>
        <package name="zsh"/>
        <!-- packages needed for resolution in OBS -->
        <package name="acl"/>
        <package name="fipscheck"/>
        <package name="ncurses-utils"/>
        <package name="sg3_utils"/>
        <package name="pkg-config"/>
        <package name="elfutils"/>
        <!-- end packages needed for resolution in OBS -->
        <!-- end basic functionality -->
        <!-- user configuration tools -->
        <package name="libyui-ncurses-pkg7"/>
        <package name="yast2"/>
        <package name="yast2-ntp-client"/>
        <package name="yast2-pam"/>
        <package name="yast2-perl-bindings"/>
        <package name="yast2-pkg-bindings"/>
        <package name="yast2-registration"/>
        <package name="yast2-schema"/>
        <package name="yast2-security"/>
        <package name="yast2-sudo"/>
        <package name="yast2-support"/>
        <package name="yast2-sysconfig"/>
        <package name="yast2-update"/>
        <package name="yast2-users"/>
        <package name="yast2-xml"/>
```

```
        <package name="yast2-ycp-ui-bindings"/>
        <!-- end user configuration tools -->
        <!-- framework specific packages -->
        <!-- instance initialization -->
        <!-- SMT server and framework specific -->
        <!-- End SMT server and framework specific -->
        <!-- end framework specific packages -->
        <!-- SMT server specific -->
        <package name="apache2"/>
        <package name="apache2-mod_perl"/>
        <package name="apache2-prefork"/>
        <package name="apache2-utils"/>
        <package name="python-serviceAccessConfig"/>
        <package name="smt"/>
        <package name="smt-ha"/>
        <package name="smt-support"/>
        <!-- end SMT server specific -->
        <!-- Infrastructure server monitoring -->
    </packages>
    <packages type="bootstrap">
        <!-- products -->
        <package name="sles-release"/>
        <package name="sles-release-POOL"/>
        <package name="filesystem"/>
        <package name="glibc-locale"/>
    </packages>
</image>
```

GNU Free Documentation License

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 non-commercially. Secondarily, 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 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.