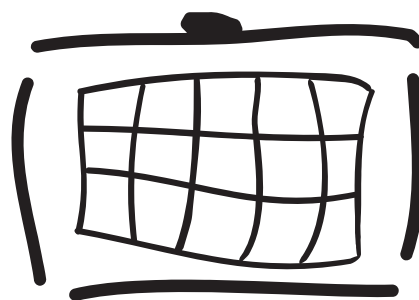


# Deployment and configuration guide for Radiator VNF Manager

Radiator VNF Flex

Copyright © 2022-2023 Radiator Software Oy.



# Radiator

## Table of contents

- [Table of contents](#)
- [Overview](#)
- [Radiator VNF Manager deployment](#)
  - [Prerequisites](#)
  - [Deploying Radiator VNF Manager](#)
- [Radiator VNF Manager set up](#)
  - [Setting up Radiator VNF Manager](#)
- [Radiator VNF Manager configuration](#)
  - [Radiator VNF Manager credentials file](#)
    - [Configuration options](#)
  - [Radiator VNF Manager configuration file](#)
    - [Configuration options](#)
    - [Cloud](#)
    - [Hosts](#)
      - [General host section](#)
      - [Host section](#)
  - [Radiator VNF Manager configuration permissions file](#)
    - [Configuration options](#)
  - [Radiator VNF Manager configuration restrictions](#)
    - [Cloud name](#)
    - [Host name](#)
    - [Radiator instance name](#)
    - [Name prefix](#)
    - [Configuration option name\\_prefix](#)
- [Maintenance tasks and other information](#)
  - [Radiator VNF host naming](#)
  - [Changing the Ansible vault password](#)
  - [Radiator VNF Manager upgrade process](#)
  - [Using own certificates in the Radiator VNF Manager](#)
  - [Updating Radiator VNF hosts](#)
  - [Radiator VNF Manager destroy process](#)
- [Appendix A: Radiator VNF Manager credentials file example](#)
- [Appendix B: Radiator VNF Manager configuration file example](#)
- [Appendix C: Radiator VNF Manager configuration permissions file example](#)

This document describes the deployment and configuration of the Radiator VNF Manager.

## Overview

Radiator VNF Flex is the management and orchestration solution for Radiator AAA that complements Radiator carrier portfolio by introducing the Radiator VNF Manager.

Single Radiator VNF Manager can be used to deploy and manage multiple different Radiator VNF hosts and configurations for fixed line, WiFi, mobile, VoWiFi, M2M, IoT and other AAA use cases.

Radiator VNF Manager provides both GUI and CLI interface for operation. Regardless which interface is used, the Radiator VNF Manager is configured with configuration files that are uploaded to the Radiator VNF Manager together with possible Radiator configuration files and any other needed files. Radiator VNF Manager keeps track of all the uploaded configuration files and performed operations with the help of Ansible and internal Git repositories.

There are several terms used in Radiator VNF Flex that have a specific meaning. Refer to the below table for a quick overview of most common ones.

<b>Radiator VNF Flex term</b>	<b>Meaning</b>
Cloud, cloud name, cloud ID	Logical group of Radiator VNF hosts managed by the Radiator VNF Manager
Radiator VNF host	OpenStack virtual machine
Radiator host	Radiator VNF host running Radiator
Radiator instance	Radiator process on Radiator host
Radiator VNF Manager credentials file	See <a href="#">Radiator VNF Manager credentials file</a>
Radiator VNF Manager configuration file	See <a href="#">Radiator VNF Manager configuration file</a>
Radiator VNF Manager configuration permissions file	See <a href="#">Radiator VNF Manager configuration permissions file</a>
<i>general</i> host section in the Radiator VNF Manager configuration file	A part in the Radiator VNF Manager configuration file containing information each Radiator VNF host in the cloud will have
<i>host</i> section in the Radiator VNF Manager configuration file	A part in the Radiator VNF Manager configuration file containing information specific to the Radiator VNF host

### *Common terms of Radiator VNF Flex*

Radiator VNF Flex deliverables include the Radiator VNF Manager Image, the passwords and SSH key used to authenticate to the Radiator VNF Manager GUI and CLI, and the Ansible vault password. Radiator VNF Manager GUI comes with default users *vnfmuser* and *vnfmadmin*, the command line with default user *vnfmadmin*. For the GUI access both users have a password, for the command line only SSH key authentication is supported. All deliverables are needed for setting up the Radiator VNF Manager after it has been deployed to the OpenStack.

# Radiator VNF Manager deployment

## Prerequisites

Radiator VNF Manager runs on top of OpenStack. Radiator VNF Manager and Radiator VNF hosts can be in the same or different OpenStack project (tenant), or even in different OpenStack environments providing the Radiator VNF Manager is able to access the OpenStack API of the OpenStack where Radiator VNF hosts are deployed to and later on access the hosts with SSH. OpenStack API is used to deploy and destroy the Radiator VNF hosts, managing the hosts is done via SSH and Ansible playbooks.

Radiator VNF Manager will create OpenStack security groups for the Radiator VNF hosts if configured so, but for example flavors and networking (internal and/or external networks, routers, gateway, DNS name servers etc.) must be configured to the OpenStack before Radiator VNF Manager can use them. Radiator VNF Manager requires at minimum 4 CPUs, 8G RAM and 80G disk which is suitable at least for testing environment, but on production environment the sizing depends on the use case, number of Radiator VNF hosts and amount of clouds being managed. Always consult Radiator Software about the flavor needs to have a working setup.

The Radiator VNF Manager GUI uses HTTPS, so HTTPS port TCP/443 must be allowed in the OpenStack security group attached to the Radiator VNF Manager. Setting up the Radiator VNF Manager also requires SSH port TCP/22 access. If the Radiator VNF Manager acts as a proxy for the Radiator VNF hosts, then TCP/3128 port also needs to be open on the Radiator VNF Manager OpenStack security groups.

Protocol	Port	Usage	Notes
TCP	22	SSH	mandatory
TCP	443	HTTPS	mandatory
TCP	3128	proxy	optional

*Ports and protocols used by the Radiator VNF Manager*

## Deploying Radiator VNF Manager

1. Prepare OpenStack networking and flavor as defined in [Prerequisites](#).
2. Import Radiator VNF Manager Image to OpenStack images with OpenStack tools.
3. Create security group for Radiator VNF Manager usage that allows at minimum HTTPS and SSH access with OpenStack tools.
4. Create Radiator VNF Manager OpenStack instance from the image with suitable flavor, network and the security group with OpenStack tools.
  1. Radiator VNF Manager uses *cloud-init*. If needed, it is possible to include changes to Radiator VNF Manager via cloud-init during deployment, for example to networks or nameservers.
5. Optionally attach floating IP to the Radiator VNF Manager with OpenStack tools.

## Radiator VNF Manager set up

The Radiator VNF Manager needs to be set up after it has been deployed to the OpenStack. The set up configures the Radiator VNF Manager GUI to be available in the specified address, creates the self-signed certificates used by default on the GUI side and starts all needed services. The set up can only be done from the command line and it requires Ansible vault password.

### Setting up Radiator VNF Manager

1. SSH to the Radiator VNF Manager <IP> as *vnfmadmin* user and with provided SSH key.
2. Run script `/home/vnfmadmin/bin/setup-vnfm.sh <IP>` to set up the Radiator VNF Manager.
  1. The script will ask the Ansible vault password when it starts, possibly multiple times. Provide the delivered Ansible vault password each time.
  2. The script will create certificates used by the GUI and start all needed services. Wait for the script to finish.
3. Browse to `https://<IP>`.
  1. Select *Run jobs* and login to the Radiator VNF Manager jobs interface either as a basic user *vnfmuser* or as an admin user *vnfmadmin* with provided respective password.
  2. Select *Documents and guides* to see available Radiator VNF Flex documentation.
  3. Select *License information* to see the Radiator licenses and Radiator VNF Flex Lifecycle Policy.

## Radiator VNF Manager configuration

Configuring the Radiator VNF Manager can be divided into two parts. The first part is the designing phase, when the use case with all the requirements and needed resources is identified. The use case may for example be fixed line RADIUS authentication and accounting, and the identified resource needs could be among all other things the number of Radiator instances to handle all the traffic.

The second part of the Radiator VNF Manager configuration is to translate the designing decisions into such a format that the Radiator VNF Manager is able to perform according to the design. This is done by creating configuration files.

There are several configuration files that are needed. The mandatory configuration files are the ***Radiator VNF Manager credentials file*** and the ***Radiator VNF Manager configuration file***. Optionally also the *Radiator VNF Manager configuration permissions file* can be used. For the Radiator VNF hosts managed by the Radiator VNF Manager, there might be a need to create for example the Radiator configurations with possible additional hooks and log configurations, certificate files, etc.

The configuration files described in this document are the Radiator VNF Manager credentials file, the Radiator VNF Manager configuration file and the Radiator VNF Manager configuration permissions file. For the Radiator configuration refer to the [Configuring Radiator](#) chapter in the Radiator reference manual.

### Radiator VNF Manager credentials file

The Radiator VNF Manager credentials file is a file that tells the Radiator VNF Manager what clouds it can manage with what credentials. The credentials available in the file are used by the openstacksdk library to access the OpenStack API and they are in plain text on the Radiator VNF Manager disk. The credentials needed are from the OpenStack where the Radiator VNF hosts should be located. It is recommended to create dedicated credentials for the Radiator VNF Manager usage, preferably the credentials should not expire nor require periodic password changes. The dedicated credentials should be able to use the OpenStack API, but there is no need for admin rights.

Example of the Radiator VNF Manager credentials file is available on the Radiator VNF Manager disk in `/var/lib/radiatorvnf/flex-ansible/credentials-example.yaml` and in [Radiator VNF Manager credentials file example](#).

The Radiator VNF Manager credentials file is a yaml file, so it must not contain any tab characters, only spaces are accepted. This document lists the most common configuration options the Radiator VNF Manager credentials file can have, but any other needed option can be added. If the option name starts with an `auth_` it is considered as part of authentication section, otherwise the option is considered to be part of the main section. The Radiator VNF Manager credentials file is used to create the OpenStack clouds.yaml file `/etc/openstack/clouds.yaml` and it can contain information for multiple clouds, regardless if the clouds reside in the same or different OpenStack projects.

### Configuration options

- `cloud_name`

```
cloud:
  <cloud_name>:
    region_name: RegionOne
```

This maps to the Radiator VNF Manager configuration file and to the cloud name that the file has. There must exist a Radiator VNF Manager credentials file which contains matching cloud\_name from the Radiator VNF Manager configuration file. Cloud name must be unique within the Radiator VNF Manager, it must not contain spaces, hyphens or preceding numbers and it must match to the cloud name used in Radiator VNF Manager configuration file. Cloud name can be for example: mycloud, M2C, MyCloud, my\_cloud

- *auth\_auth\_url*

OpenStack API authentication URL with version of the OpenStack hosting the Radiator VNF hosts, for example: <https://openstack.example.com:5000/v3>

- *auth\_username*

User name in the OpenStack hosting the Radiator VNF hosts, for example: userid. User must be able to use the OpenStack API, but admin rights are not needed.

- *auth\_password*

User password in the OpenStack hosting the Radiator VNF hosts in plain text, for example: MyPW1!!ss

- *auth\_project\_name*

Name of the project from the OpenStack hosting the Radiator VNF hosts, for example: os\_project

- *auth\_project\_domain\_name*

Name of the project domain from the OpenStack hosting the Radiator VNF hosts, for example: default

- *auth\_user\_domain\_name*

Name of the domain name from the OpenStack hosting the Radiator VNF hosts, for example: default

- *region\_name*

Name of the region on the OpenStack hosting the Radiator VNF hosts, for example: regionOne

- *interface*

Interface name on the OpenStack hosting the Radiator VNF hosts, for example: public. The interface is optional.

- *identity\_api\_version*

OpenStack identity API version of the OpenStack hosting the Radiator VNF hosts, for example: 3. The identity\_api\_version is optional.

**NOTE:** The version information must also be on the auth\_auth\_url, for example: v3.

- *cacert*

Certificate bundle for intermediate and root certificates used by the OpenStack hosting the Radiator VNF hosts, for example: `/var/lib/radiatorvnf/incoming/openstack-ca`. The certificate bundle file can be uploaded to Radiator VNF Manager either via GUI or CLI, see chapter *10 Upload configuration package* in [User guide for Radiator VNF Manager](#) or chapter *Upload configuration package* in [Advanced user guide for Radiator VNF Manager](#). The uploaded file will stay in the `/var/lib/radiatorvnf/incoming/` directory. The *cacert* is optional.

- *verify*

By default the `openstacksdk` library has SSL verification enabled. If there is no possibility for getting the intermediate and root certificates for *cacert* bundle, then the verification can be disabled, although this is not advisable. Possible values for the *verify* are `yes/no/true/false`. The *verify* is optional and the default value is **yes**.

**NOTE:** If the *verify* has been defined as disabled in the credentials configuration, it will override *cacert* definition.

## Radiator VNF Manager configuration file

The Radiator VNF Manager configuration file is a file that describes the Radiator VNF hosts and all the information related to them to the Radiator VNF Manager. The information includes for example in which cloud the Radiator VNF hosts are located to, if Radiator should be installed to the host and what is the Radiator instance name and configuration file name for it. Example of the Radiator VNF Manager configuration file is available on the Radiator VNF Manager disk in `/var/lib/radiatorvnf/flex-ansible/cloudinfo-example.yaml` and in [Radiator VNF Manager configuration file example](#).

The Radiator VNF Manager configuration file is a `yaml` file, so it must not contain any tab characters, only spaces are accepted. The Radiator VNF Manager configuration file can contain information for multiple clouds, and if the Radiator VNF Manager is managing several clouds all the clouds should be described in a single Radiator VNF Manager configuration file.

The cloud section in the Radiator VNF Manager configuration file can be divided to multiple *host* sections. Every cloud must have a *general* host section, which contains the bulk of the configured information, and *host* section per each Radiator VNF host. If the configuration option is only listed in the *general* host section, it will be used for every Radiator VNF host of the cloud. This means that there is no need to list for example *networks* for each Radiator VNF host if all the Radiator VNF hosts in the cloud use the same network. Simply define the *networks* on the *general* host section and omit the option from the *host* sections. On the other hand, if all the other Radiator VNF hosts in the cloud use the same network except one Radiator VNF host has different configuration, define the common network on the *general* host section and define again the *networks* on the specific host that has different network in use. Defining the configuration option on the host section will override that *general* definition.

This document lists all the configuration options the Radiator VNF Manager configuration file can have.

## Configuration options

### Cloud



- *cloud\_name*

```
cloud:
  <cloud_name>:
    hosts:
      general:
        name_prefix: ""
```

This maps to the Radiator VNF Manager credentials file and to the cloud name that the file has. There must exist a Radiator VNF Manager configuration file which contains matching *cloud\_name* from the Radiator VNF Manager credentials file. Cloud name must be unique within the Radiator VNF Manager, it must not contain spaces, hyphens or preceding numbers and it must match to the cloud name used in the Radiator VNF Manager credentials file. Cloud name can be for example: mycloud, M2C, MyCloud, my\_cloud

## Hosts

### General host section

- *name\_prefix*

A string that is prefixed to each Radiator VNF hosts' name. See more information on the Radiator VNF hosts naming convention from [Radiator VNF host naming](#). Option must exist at least in the *general* host section for each cloud, but it can be specified per host also. Default value is "".

- *base\_image*

The name of the image available in OpenStack that should be used when creating the Radiator VNF hosts, for example: Almalinux-9. Option must exist at least in the *general* host section for each cloud, but it can be specified per host also.

- *flavor*

The OpenStack flavor that the Radiator VNF hosts should use, for example: small. Option must exist at least in the *general* host section for each cloud, but it can be specified per host also.

- *radiator\_release*

The Radiator release that is installed to the Radiator host, for example: 4.27-1. By default the Radiator VNF Manager has deb, el8 rpm and el9 rpm packages available, but it is possible to upload a specific installation package version to the Radiator VNF Manager and use also that, see chapter *10 Upload configuration package* in [User guide for Radiator VNF Manager](#) or chapter *Upload configuration package* in [Advanced user guide for Radiator VNF Manager](#). Option must exist at least in the *general* host section for each cloud, but it can be specified per host also.

- *radiator\_radius\_utilxs\_release*

The Radiator Radius::UtilXS addon library release that is installed to the Radiator host, for example: 2.3-2. By default the Radiator VNF Manager has Radius::UtilXS packages for Ubuntu 20.04, Ubuntu 22.04,

el8 rpm and el9 rpm available, but it is possible to upload a specific installation package version to the Radiator VNF Manager and use also that, see chapter *10 Upload configuration package* in [User guide for Radiator VNF Manager](#) or chapter *Upload configuration package* in [Advanced user guide for Radiator VNF Manager](#). Option must exist at least in the *general* host section for each cloud, but it can be specified per host also.

- *networks*

A list of networks that the Radiator VNF host should use, for example:

```
networks:
- testing_net
- public
```

If any of the Radiator VNF hosts uses floating IPs, define those on specific host section. See *floating\_public\_ip*. Option must exist at least in the *general* host section for each cloud, but it can be specified per host also.

- *auto\_public\_ip*

Define if automatic public IP should be assigned to the Radiator VNF host, for example: no. Possible values are yes/no/true/false. Setting this to yes/true requires that there exists an external network from where the public IP can be obtained. If automatic public IP is assigned to the Radiator VNF host, it is also automatically cleared out when the Radiator VNF host is destroyed. Option must exist at least in the *general* host section for each cloud, but it can be specified per host also.

- *use\_private\_ip\_connections*

Use the private network interface of each Radiator VNF host, if it has one, as the host's IP in the Ansible inventory and when connecting to the Radiator VNF host via SSH. This can be useful if the Radiator VNF Manager is running in a cloud that can communicate to the Radiator VNF hosts over the private network. For example: yes. Possible values are yes/no/true/false.

**NOTE:** This option is cloud specific, so it must exist in the *general* host section for each cloud. Host specific definition may exist but it is not taken into account.

- *proxy\_in\_use*

The Radiator VNF Manager can act as a proxy for the Radiator VNF hosts. For example: no. Possible values are yes/no/true/false.

**NOTE:** This option is cloud specific, so it must exist in the *general* host section for each cloud. Host specific definition may exist but it is not taken into account.

- *radiator\_installed*

Control if the Radiator and Radius::UtilXS are installed on the Radiator VNF host, for example: yes. When set to yes/true, also the Radiator configurations are deployed to the Radiator VNF hosts when the Radiator VNF hosts are set up. When set to no/false, no Radiator software or configurations go to the Radiator VNF host. This is useful if the Radiator VNF host should be running for example database

server only. Option must exist at least in the *general* host section for each cloud, but it can be specified per host also.

- *security\_groups*

OpenStack security groups that should be applied to the Radiator VNF hosts. Can contain existing OpenStack security group names and definitions of new OpenStack security groups that will be created. Multiple groups can be defined. For example:

```
security_groups:
  - name: "Existing security group name"
  - name: "New Radiator VNF Flex RADIUS security group"
    protocol: udp
    min_port: 1812
    max_port: 1813
    remote_ips: 0.0.0.0/0
```

- *key\_name*

Define existing OpenStack key that should be applied to all Radiator VNF hosts. By default the Radiator VNF Manager will create a SSH key that is used with all the Radiator VNF hosts managed by the Radiator VNF Manager regardless of the cloud the hosts are located in, but in addition it is possible to apply also the key from OpenStack to the Radiator VNF hosts. For example: `openstack_key`. Option can optionally exist in the *general* host section for each cloud, but it can be specified per host also. Default value is `""`.

- *availability\_zone*

OpenStack availability zone where the Radiator VNF host should be located to, for example: `zone1`. Option can optionally exist in the *general* host section for each cloud, but it can be specified per host also. Default value is `""`.

- *default\_repository\_mirror*

It is possible to define a default repository mirror that is used by the Radiator VNF hosts when packages are installed from the Internet. This makes it possible to restrict the URLs that the Radiator VNF hosts or the Radiator VNF Manager can access. For example: <https://ftp.funet.fi/pub/mirrors/almalinux.org/>. Option can optionally exist in the *general* host section for each cloud, but it can be specified per host also. Default value is `""`.

**NOTE:** Radiator VNF Manager does not contain package repositories. Depending on the image used for the Radiator VNF hosts it is possible that most of the packages required for successfully running Radiator must be installed from for example AlmaLinux public repositories.

- *vnfm\_to\_hosts\_net*

If the Radiator VNF hosts have multiple private network interfaces for different purposes, for example one for the Radiator VNF Manager usage and other for RADIUS traffic, define which network the Radiator VNF Manager should use when connecting to the Radiator VNF hosts. The network defined

here should also be present on the *networks* option either on *general* or host specific level. If the network specified here is not available on each Radiator VNF host, connection to the Radiator VNF hosts from the Radiator VNF Manager will fail. For example: *secondary\_net*.

**NOTE:** This option is cloud specific, so it must exist in the *general* host section for each cloud. Host specific definition may exist but it is not taken into account.

### Host section

- *host\_name*

```
cloud:
  <cloud_name>:
    hosts:
      general:
        name_prefix: ""
        ...
      <host_name>:
        name_prefix: "radhost"
        ...
      <host_name>:
        radiator_installed: no
```

Host name defines the Radiator VNF host that is created to the OpenStack. The host name must be unique within the clouds managed by a single Radiator VNF Manager and it must not contain spaces, hyphens or preceding numbers. There can be multiple host names defined per cloud, each defining multiple options that may or may not already be defined on the *general* host section. If the option is defined on the *general* host section, the host section definition will overwrite it. Host name can be for example: *myhost*, *Host2*, *rad\_hostA*

- *radiator\_instances*

If the Radiator VNF host is a Radiator host, define the Radiator instances that will be running on the Radiator host. Each Radiator VNF host can have multiple Radiator instances. Radiator instance name must be unique within a cloud. Define also *radiator\_installed* as *yes/true* either on the *general* host section or on the host section, otherwise the Radiator instances cannot be created. For each Radiator instance also Radiator configuration must be defined.

- *radiator\_configuration*

Each Radiator instance must have a Radiator configuration defined. The Radiator configuration should be the main configuration file with a name in the format of *radiator\_<Radiator instance name>.conf*. The Radiator configuration file name is used to create the named Radiator service *radiator@<Radiator instance name>* on the Radiator VNF host instead of using the standard *radiator.service*. It is good practise to have the Radiator instance name same as the Radiator service name, that is the Radiator instance name matches the Radiator configuration file name. This will make it easier to see which Radiator configuration file is used by which Radiator instance if the Radiator host has multiple Radiator instances. For example:

```
radiator_instances:  
  rad1:  
    radiator_configuration:  
      - radiator_rad1.conf  
  LB:  
    radiator_configuration:  
      - radiator_lb.conf  
  devel:  
    radiator_configuration:  
      - radiator_devel.conf
```

The Radiator configuration files must be uploaded to the Radiator VNF Manager before setting up the Radiator VNF hosts, see chapter *10 Upload configuration package* in [User guide for Radiator VNF Manager](#) or chapter *Upload configuration package* in [Advanced user guide for Radiator VNF Manager](#). This is because the setup expects to find the Radiator configuration files defined in the `radiator_configuration` option. See chapter *12 Setup OpenStack hosts* in [User guide for Radiator VNF Manager](#) or chapter *Setup OpenStack hosts* in [Advanced user guide for Radiator VNF Manager](#) how to run the setup.

- *floating\_public\_ip*

Fixed floating IP that the Radiator VNF host should have. The floating IP must be already created and available in the OpenStack. This option requires the info to which network the floating IP should be mapped to, so networks must be defined on the host level again. For example:

```
floating_public_ip: 1.2.3.4  
networks:  
  - private_net
```

**NOTE:** Radiator VNF host configuration cannot contain both enabled `auto_public_ip` and `floating_public_ip`.

- *enhancements*

Configuration option *enhancements* provides possibility to modify the Radiator VNF host's internal firewall, install additional software, restart services and upload files to the Radiator VNF hosts with specified permissions and ownership.

- *allow\_fw\_ports*

Specify the port and protocol that is allowed on the Radiator VNF host firewall, for example: 8080/TCP. To allow multiple ports this option can be defined multiple times. This operation is run before any other enhancement, so it is possible to open ports for for example 3rd party software that is installed in later stages.

- *deny\_fw\_ports*

Specify the port and protocol that is removed from the Radiator VNF host firewall, for example: 8080/TCP. To deny multiple ports this option can be defined multiple times.

- *installable*

This is the deb/rpm file to be uploaded to the Radiator VNF host for installation and then automatically installed, for example: file.rpm. The installable file must be first uploaded to the Radiator VNF Manager, see chapter *10 Upload configuration package* in [User guide for Radiator VNF Manager](#) or chapter *Upload configuration package* in [Advanced user guide for Radiator VNF Manager](#).

- *servicename*

This is a systemd service available on the Radiator VNF host that should be enabled and restarted, for example: filed. This operation is guaranteed to run last of all the enhancement actions, so it is possible to first install some 3rd party software to the Radiator VNF host and then enable and restart the service the 3rd party software brings.

- *configuration\_source*

*configuration\_source* is a file name of a file that has been uploaded to the Radiator VNF Manager and that should be uploaded to the Radiator VNF host. This can be:

1. a single file or
2. a package of files. Supported package types are .zip, .tar, .tar.gz, .tar.bz2, .tar.xz, .tar.zst and .gz, .bz2, .xz, or .zst files that contain a .tar archive.

*configuration\_source* must be used together with *target\_with\_permissions*, it is not a standalone option. In the second case, there is an option *permissions\_file* that can be used with the package. If *permissions\_file* is used, the file specified on the *permissions\_file* option must be included in the package.

- *target\_with\_permissions*

*target\_with\_permissions* is an option that defines to where and with what permissions the *configuration\_source* should be uploaded on the Radiator VNF host. *target\_with\_permissions* and *configuration\_source* must be used together. *target\_with\_permissions* by itself is not enough, at minimum it requires the *configuration\_target* to be defined. For example:

```
configuration_source: conf.cfg
target_with_permissions:
  - configuration_target: /etc/sss/s.conf
```

- *configuration\_target*

*configuration\_target* is either:

1. The full path, including the file name, of the uploaded file on the Radiator VNF host.
2. The path where the configuration source package should be unpacked. For example:

```
configuration_source: large_file.zip
target_with_permissions:
  - configuration_target: /usr/
```

The trailing `/` in the `configuration_target` is significant and indicates whether a single file is to be replaced/inserted (case 1.), or if the archive is to be extracted into the described directory (with trailing `/`) "as is" (case 2.). The `configuration_target` location must exist, the operation will fail if the location is not available.

- *owner*

File ownership, for example: `radiator`. This is an optional parameter.

- *group*

File group, for example: `radiator`. This is an optional parameter.

- *mode*

File permission in octal. Note quotes and leading zero, they are mandatory for *mode*. For example: `'0555'`. This is an optional parameter.

- *permissions\_file*

When the `configuration_source` is not a single file but a package, the `permissions_file` can be used to define file ownership and permissions for multiple files uploaded in a package, for example: `permissionfile.yaml`. The `permissions_file` must be included in the package defined in `configuration_source`. See [Radiator VNF Manager configuration permissions file](#) for more.

- *repo\_installable*

Define single package that should be installed from existing repository, for example: `perl-JSON-PP`. For multiple packages this option can be defined multiple times. This is guaranteed to be run after `installable` and `configuration_source`, so it is possible to create a new repository source before using this option.

## Radiator VNF Manager configuration permissions file

The main idea how the Radiator VNF Manager configuration permissions file and the related *enhancements* parts is implemented is to provide a way to create a file tree of custom files to the Radiator VNF host specific location. For example consider the following setup of multiple configuration files, hooks and an additional Perl module:

```
radiator
├── conf.d
│   ├── README.txt
│   └── acctlog-formats.conf
```

```
├── authlog-formats.conf
└── rr1
    ├── RadiatorModule.pm
    ├── hook1.pl
    ├── hook2.pl
    ├── hook3.pl
    └── options.conf
```

It is possible to define each file as *configuration\_source* and define matching *target\_with\_permissions* section for it, but it is a lot simpler to just define *permissions\_file* and create the Radiator VNF Manager configuration permissions file listing each file with wanted permissions, package all the files together including the Radiator VNF Manager configuration permissions file for example as a zip or .tar.gz package, and upload that to */etc/*. When the package is extracted the files automatically go to the intended places and the Radiator VNF Manager configuration permissions file is processed so the files get correct ownership and permissions.

The Radiator VNF Manager configuration permissions file file name is defined in the *permissions\_file* option. The *configuration\_target* option defines the path where the package listed in *configuration\_source* is copied and unpacked. The Radiator VNF Manager configuration permissions file is then processed, resulting in the unpacked files to get the specified ownership and permissions. As a last step, the used Radiator VNF Manager configuration permissions file is cleared from the Radiator VNF host. For example, the Radiator VNF Manager configuration permissions file for the above file tree could be following:

```
- file: /etc/radiator/conf.d/acctlog-formats.conf
  owner: root
  group: radiator
  mode: '0644'
- file: /etc/radiator/conf.d/authlog-formats.conf
  owner: root
  group: radiator
  mode: '0644'
- file: /etc/radiator/rr1/RadiatorModule.pm
  owner: root
  group: radiator
  mode: '0644'
- file: /etc/radiator/rr1/hook1.pl
  owner: root
  group: radiator
  mode: '0644'
- file: /etc/radiator/rr1/hook2.pl
  owner: root
  group: radiator
  mode: '0644'
- file: /etc/radiator/rr1/hook3.pl
  owner: root
  group: radiator
  mode: '0644'
- file: /etc/radiator/rr1/options.conf
  owner: radiator
  group: radiator
  mode: '0640'
```



Only file information is mandatory in the Radiator VNF Manager configuration permissions file, other information is optional. However, there is no need to specify a file unless some other information is also specified. If file is not specified in the Radiator VNF Manager configuration permissions file, the uploaded file will get default permissions and ownership of the target directory system.

**NOTE:** The operation processing the unpacking and the Radiator VNF Manager configuration permissions file has sudo privileges, so it can overwrite existing files. Be careful not to replace any system files as that could lead to a broken Radiator VNF host. In case of such event, destroy the Radiator VNF host, upload correct files to Radiator VNF Manager, create the Radiator VNF host and set it up again. See chapters *30 Destroy hosts*, *10 Upload configuration package*, *11 Create OpenStack hosts* and *12 Setup OpenStack hosts* in [User guide for Radiator VNF Manager](#) or chapters *Destroy hosts*, *Upload configuration package*, *Create OpenStack hosts* and *Setup OpenStack hosts* in [Advanced user guide for Radiator VNF Manager](#).

Example of the Radiator VNF Manager configuration permissions file is available on the Radiator VNF Manager disk in `/var/lib/radiatorvnf/flex-ansible/cloudinfo_permission_file-example.yaml` and in [Radiator VNF Manager configuration permissions file example](#).

The Radiator VNF Manager configuration permissions file is a yaml file, so it must not contain any tab characters, only spaces are accepted.

**NOTE:** Include the Radiator VNF Manager configuration permissions file to the package listed in `configuration_source`.

## Configuration options

- *file*

Full path and name of the file in the Radiator VNF host environment, for example: `/etc/letsencrypt/renewal-hooks/post/renewal.sh`. Mandatory parameter.

- *owner*

File ownership in the Radiator VNF host environment, for example: `radiator`. The owner must exist in the Radiator VNF host. Optional parameter.

- *group*

File group in the Radiator VNF host environment, for example: `radiator`. The group must exist in the Radiator VNF host. Optional parameter.

- *mode*

File permissions in the Radiator VNF host environment, for example: `'0660'`. Note the syntax, permissions are given as octals so they need to start with preceding zero (0) and quotes are needed to ensure the proper handling of the permission. Optional parameter.

## Radiator VNF Manager configuration restrictions

The Radiator VNF Manager configuration files have some restrictions for example how hosts can be named. The restrictions are listed on each configuration option, but they are also collected to this chapter for easy high level view.

**Cloud name**

- Must be unique within the Radiator VNF Manager
- Cannot contain spaces or hyphens, start with numbers
- Must have a matching cloud name in the Radiator VNF Manager credentials file

**Host name**

- Must be unique within the Radiator VNF Manager
- Cannot contain spaces or hyphens, start with numbers

**Radiator instance name**

- Must be unique within the cloud

**Name prefix**

- Cannot contain spaces, start with numbers or hyphens

**Configuration option name\_prefix**

- Cannot contain spaces, start with numbers or hyphens

## Maintenance tasks and other information

### Radiator VNF host naming

The Radiator VNF hosts created by the Radiator VNF Manager have partially fixed hostnames. The hostname by default is in the format of `<cloud>-vnf-host-<host>`, where the `<cloud>` and `<host>` come from the Radiator VNF Manager configuration file uploaded to the Radiator VNF Manager. Both `<cloud>` and `<host>` can be freely configured providing they do not contain spaces, hyphens or preceding numbers. For example, if `<cloud>` is "fixedline" and `<host>` is "dbtest", the hostname would be `fixedline-vnf-host-dbtest`.

It is possible to prefix the default hostname with a configurable prefix to create hostname in the format of `<prefix>-<cloud>-vnf-host-<host>`. The prefix is not mandatory, but if it is defined it can not contain spaces or start with a hyphen or a number. For example, if `<prefix>` is "ABC-321", `<cloud>` is "fixedline" and `<host>` is "dbtest", the hostname would be `ABC-321-fixedline-vnf-host-dbtest`.

**NOTE:** The configurable prefix, cloud name and host name are all case sensitive. Pay special attention to the cloud name, as the cloud name defined in the Radiator VNF Manager configuration file must match to the cloud name in the Radiator VNF Manager credentials file. For example, uploading configuration for the cloud "VNFFlex" and credentials for "vnfflex" will cause error when trying to run operations against the cloud as the credentials are not found.

### Changing the Ansible vault password

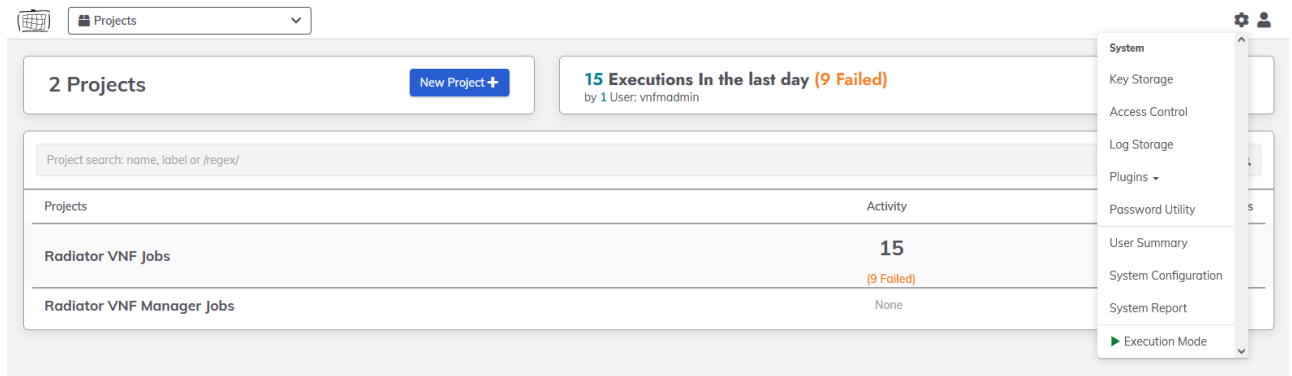
The Radiator VNF Manager uses Ansible vault to store sensitive information like self-signed certificates and SSH keys. Each Radiator VNF Manager Image has unique Ansible vault password that is created during building time. If needed, the Ansible vault password can be changed, although it is not recommended. The password needs to be changed both on command line and in GUI.

1. Run Ansible vault rekey on command line: `ansible-vault rekey --vault-id vnfm@prompt /var/lib/radiatorvnf/flex-ansible/group_vars/all/vault.yml`.
2. When prompted, give current Ansible vault password.
3. When prompted, give new Ansible vault password.
4. When prompted, repeat new Ansible vault password.

The Ansible vault password is now changed on the command line and the GUI jobs will fail until the new password is also updated on the GUI side.

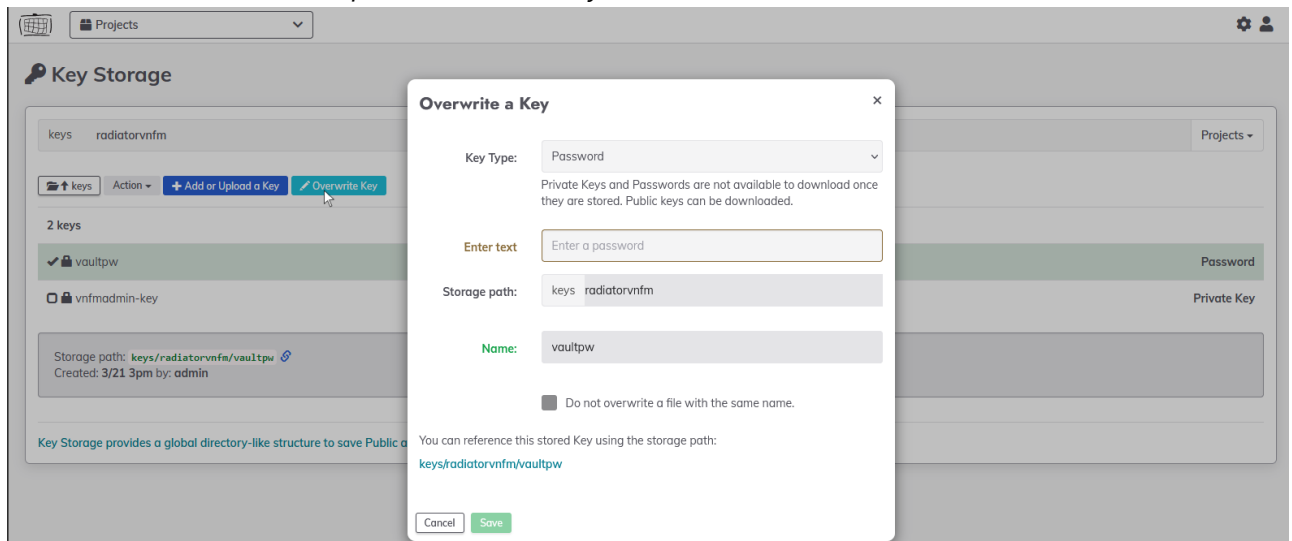
1. Login to the Radiator VNF Manager GUI as `vnfmadmin`.

2. Locate the settings icon on upper right corner and select *Key Storage*.



### *Radiator VNF Manager GUI settings*

3. Select *radiatorvnfm* -> *vaultpw* -> *Overwrite Key*



### *Radiator VNF Manager GUI Key Storage*

4. Insert the new password to *Enter text* field.
5. Press Save.

The Ansible vault password is now changed on the GUI.

## **Radiator VNF Manager upgrade process**

Upgrading process for the Radiator VNF Manager is needed for example when newer Radiator VNF Manager Image is available and it is taken into use. The upgrade process refers mainly to a situation where the Radiator VNF Manager has been in use and there are one or more Radiator VNF hosts that the Radiator VNF Manager has managed. If the Radiator VNF Manager hasn't been used, that is there never has been any Radiator VNF hosts that the Radiator VNF Manager has managed, it could be that the old Radiator VNF Manager can be just destroyed from the OpenStack and a new Radiator VNF Manager can be deployed from the new image. Even in such case it is recommended that an export package is taken from the old Radiator VNF Manager for backup before destroying the old Radiator VNF Manager and all the data inside it.

"The old Radiator VNF Manager" refers to the existing Radiator VNF Manager that is being replaced with new one. "The new Radiator VNF Manager" refers to the new Radiator VNF Manager that is being deployed.

1. On the old Radiator VNF Manager run job *01 Export Radiator VNF Manager data* from the [User guide for Radiator VNF Manager](#) or *Export Radiator VNF Manager data* from the [Advanced user guide for](#)

- [Radiator VNF Manager](#) to create export package.
2. Download the created export package from the old Radiator VNF Manager to some safe location.
3. Import the new Radiator VNF Manager Image to the OpenStack with OpenStack tools.
4. Launch the new Radiator VNF Manager OpenStack instance from the imported image with OpenStack tools, see more from [Radiator VNF Manager deployment](#).
5. Set up the new Radiator VNF Manager as instructed in [Radiator VNF Manager set up](#).
6. On the old Radiator VNF Manager run job *04 Commission Radiator VNF Manager* from the [User guide for Radiator VNF Manager](#) or *Commission Radiator VNF Manager* from the [Advanced user guide for Radiator VNF Manager](#) to commission the new Radiator VNF Manager IP address to the existing Radiator VNF hosts.
7. On the new Radiator VNF Manager run job *02 Import Radiator VNF Manager exported data* from the [User guide for Radiator VNF Manager](#) or *Import Radiator VNF Manager exported data* from the [Advanced user guide for Radiator VNF Manager](#) to import the export package from the safe location to the new Radiator VNF Manager.
8. Verify that the new Radiator VNF Manager is able to manage the Radiator VNF hosts. For example check that on the GUI the *Nodes* view shows the same information on the new Radiator VNF Manager as on the old Radiator VNF Manager or on the new Radiator VNF Manager run job *15 Restart Radiator instances* from the [User guide for Radiator VNF Manager](#) or *Restart Radiator instances* from the [Advanced user guide for Radiator VNF Manager](#) to see that connection really works to the Radiator VNF hosts.
9. On the new Radiator VNF Manager run job *05 Decommission Radiator VNF Manager* from the [User guide for Radiator VNF Manager](#) or *Decommission Radiator VNF Manager* from the [Advanced user guide for Radiator VNF Manager](#) to decommission the old Radiator VNF Manager IP address from the existing Radiator VNF hosts.
10. Shutdown and remove the old Radiator VNF Manager from the OpenStack with OpenStack tools.

## Using own certificates in the Radiator VNF Manager

See chapter *00 Import certificates to Radiator VNF Manager* on the [User guide for Radiator VNF Manager](#) or *Import certificates to Radiator VNF Manager* on the [Advanced user guide for Radiator VNF Manager](#)

## Updating Radiator VNF hosts

See chapter *23 Update hosts* on the [User guide for Radiator VNF Manager](#) or *Update hosts* on the [Advanced user guide for Radiator VNF Manager](#).

## Radiator VNF Manager destroy process

Destroy the Radiator VNF Manager from OpenStack with OpenStack tools. This operation will destroy the Radiator VNF Manager and all the data inside, but if the Radiator VNF Manager has Radiator VNF hosts it manages, they are left intact. It is recommended to run job *01 Export Radiator VNF Manager data* from the [User guide for Radiator VNF Manager](#) or *Export Radiator VNF Manager data* from the [Advanced user guide for Radiator VNF Manager](#) to create export package before destroying the Radiator VNF Manager.

## Appendix A: Radiator VNF Manager credentials file example

```
---
# credentials-example.yaml
# Credentials for Radiator VNF Manager openstacksdk usage
cloud:
  # This cloud name can be changed, it must be:
  # - Unique
  # - Contain no spaces or hyphens, or start with numbers
  # - Match to the cloud name used in the Radiator VNF Manager configuration file
vnfflex:
  region_name: RegionOne
  auth_auth_url: https://openstack.address.example.com:5000/v3
  auth_username: osuser
  auth_password: ospassword
  auth_project_name: osproject
  auth_project_domain_name: default
  auth_user_domain_name: default
  interface: public
  identity_api_version: 3
  cacert: /var/lib/radiatorvnf/incoming/openstack-ca
  verify: yes

vnfnon:
  region_name: regionone
  auth_auth_url: https://os.example.com:13000/v3
  auth_username: osuser
  auth_password: ospassword
  auth_project_name: project
  auth_project_domain_name: default
  auth_user_domain_name: default
  verify: no

# There can be multiple credentials for example for different projects or clouds
# which are all managed from the same Radiator VNF Manager
```

## Appendix B: Radiator VNF Manager configuration file example

```

---
# cloudinfo-example.yaml
# Required information for the Radiator VNF Manager for OpenStack cloud deployment
cloud:
  # This cloud name can be changed, it must be:
  # - Unique
  # - Contain no spaces or hyphens, or start with numbers
  # - Match to the cloud name used in the Radiator VNF Manager credentials file
vnfflex:
  hosts:
    general:
      # "general" is a fixed name, do not change it
      # Generic information here, applies to all hosts
      # If host has same parameter, it overwrites the generic one
      # All parameters here are mandatory
      # Name prefix will prefix the hostname with specified value
      # Name prefix cannot contain spaces, start with hyphens or preceding numbers
      name_prefix: FLEX
      base_image: Almalinux-9
      flavor: small
      radiator_release: 4.27-1
      radiator_radius_utilxs_release: 2.3-2
      # Define networks either here or separately per host. If floating IPs are
      # needed, define those on specific host section
      networks:
        - testing_net
        - public
      # Should automatic public IP be assigned to the host?
      auto_public_ip: no
      # Use the private interface of each server, if it has one, as the host's IP in
      # the inventory.
      # This can be useful if you are running ansible inside a server in the cloud
      # and would rather communicate to your servers over the private network.
      # Since this parameter is not host specific, only the value under "general" in
      # taken into account
      use_private_ip_connections: yes
      # Should hosts traffic flow via Radiator VNF Manager?
      # NOTE: Radiator VNF Manager must have port TCP/3128 open
      proxy_in_use: no
      # Should Radiator be installed or not? If yes, also Radiator configurations are
      # deployed to host. If no, no Radiator software or configurations go to host
      radiator_installed: yes
      # Default security group, applied to all hosts. Multiple groups can be defined
      # Security group names must be unique
      # SSH and when needed proxy ports are automatically added to Radiator VNF Flex
      # default security group
      security_groups:
        - name: "Radiator VNF Flex RADIUS security group"
          protocol: udp
          min_port: 1812
          max_port: 1813
          remote_ips: 0.0.0.0/0
          # Apply already existing security group to all hosts
        - name: "Already existing security group"
      # Define default repository mirror.
      # Following are optional parameters, should only be included if they are
      # wanted/needed
      key_name: openstack_key
      availability_zone: ""
      # Can be used for example when firewall has list of allowed URLs
      # Mirror URL syntax is following:
      https://ftp.funet.fi/pub/mirrors/almalinux.org/
      default_repository_mirror: ""

```

```

# If you have multiple private network interfaces for different purposes, for
example one for VNF usage and other for traffic
# from Radiator hosts to internet, define which network VNF should use to
connect to the hosts
# NOTE: This parameter can only be set on general level, not on host level!
vnfm_to_hosts_net: ""

w1:
# host w1 name is editable with following rules:
# - it must be unique within the clouds managed by single Radiator VNF Manager
# - contain no spaces, hyphens or preceding numbers
# Specific information for w1, optional
auto_public_ip: yes
availability_zone: "az1"
# This host has three Radiator instances. It is recommended to name the
instance and configuration so they match,
# this makes it easier to see which configuration is used by which instance
# Radiator configuration name should not change, that can lead to unexpected
situation with tags and related configuration switch
# Radiator instance name must be unique within a cloud
radiator_instances:
  rad1:
    radiator_configuration:
      - radiator_rad1.conf
  rad2:
    radiator_configuration:
      - radiator_rad2.conf
  rad3:
    radiator_configuration:
      - radiator_rad3.conf
# Host should be connected to these networks
# Since auto_public_ip is defined, the provider/public network must be defined
also here
networks:
- test_net
- public
# Example how to define multiple security groups
security_groups:
- name: "Radiator VNF Flex security group 8080"
  protocol: tcp
  min_port: 8080
  max_port: 8088
  remote_ips: 10.10.0.0/0
- name: "Radiator VNF Flex security group icmp"
  protocol: icmp
  min_port: -1
  max_port: -1
  remote_ips: 20.20.0.0/0
# Apply already existing security group to this host only
- name: "Already existing security group2"

w2:
# Specific information for w2, optional
flavor: medium
# Fixed floating IP requires the info of which network to use
floating_public_ip: 1.2.3.4
networks:
- testing_net
radiator_instances:
  abc1:
    radiator_configuration:
      - radiator_abc1.conf
  abc2:
    radiator_configuration:
      - radiator_abc2.conf

vnfnon:
  host:

```



```

hosts:
  general:
    # Generic information here, applies to all hosts
    name_prefix: GEN
    base_image: Ubuntu2204
    key_name: user_key
    flavor: small
    radiator_release: 4.26-1
    radiator_radius_utilxs_release: 2.3-2
    availability_zone: ""
    auto_public_ip: yes
    networks:
      - packaging_net
      - public
    proxy_in_use: no
    radiator_installed: yes

  instanssil:
    flavor: m1.medium
    radiator_instances:
      c:
        radiator_configuration:
          - radiator_c.conf
      b:
        radiator_configuration:
          - radiator_b.conf

# Example of non-Radiator instance. This could be for example database instance
instanssi2:
  flavor: db
  radiator_installed: no
  # Additional files to be installed with their optional configuration files and
  # locations
  # The trailing / in the configuration_location is significant and indicates
  # whether a single file is to be replaced,
  # or if the archive is to be extracted into the described directory (with
  # trailing /) as is"
  # configuration_target location must exists
  enhancements:
    - allow_fw_ports: 888/TCP
    - deny_fw_ports: 123/UDP
    - installable: file1.rpm
    # This is guaranteed to be run last from all enhancements actions
    # Service is restarted and enabled
    servicename: file1-service
    configuration_source: conf.cfg
    target_with_permissions:
      - configuration_target: /etc/ssss/s.conf
        owner: vnfadmin
        group: vnfmrund
        # Note quotes, they are mandatory for mode
        mode: '0555'
    - installable: file2.rpm
      configuration_source: locd_content.zip
      target_with_permissions:
        - configuration_target: /
          permissions_file: permission_file-example.yaml
    - installable: file3.rpm
      configuration_source: abc_repository
      target_with_permissions:
        - configuration_target: /etc/yum.repos.d/new.repo
          owner: root
          group: radiator
          mode: "0600"
    # This is guaranteed to be run after installables and configuration_sources
    - repo_installable: perl-JSON-PP

```

## Appendix C: Radiator VNF Manager configuration permissions file example

```
# Example file to set the permissions for files uploaded via enhancements
# This file should be valid yaml
# For each file in the upload package that requires something else than
# the default permissions, the required permissions can be set here
# File must be full path to the file after it has been unpacked to the system
# Other possible information is owner, group and mode
- file: /usr/share/misc/file
  owner: almalinux
  group: root
  mode: '0600'
- file: /etc/yum.repos.d/newrepo.repo
  group: almalinux
  mode: '0666'
- file: /etc/letsencrypt/renewal-hooks/post/renewal.sh
  mode: '0755'
```