Advanced user guide for Radiator VNF Manager
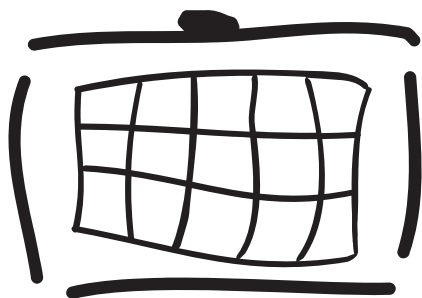
Radiator VNF Flex

Copyright © 2022-2023 Radiator Software Oy.

Table of contents

This document describes the available command-line actions for Radiator VNF Manager.

Running Ansible playbooks

Ansible playbooks are repeatable and re-usable collection of tasks that are used to create the Radiator VNF Flex. Ansible playbooks are run with *ansible-playbook* command. The command has several options available, but this document only lists the ones that are used when running Radiator VNF Flex commands:

- *--ask-vault-password, --ask-vault-pass*

  Ansible vault password is asked.

- *-e, --extra-vars*

  Set additional variables as key=value to be used in the playbooks.

- *-i, --inventory*

  Specify directory from where inventory is read.

- *-l, --limit*

  Limit the host selection.

- *-v, --verbose*

  Verbose mode can be used when troubleshooting, causes Ansible to print more debug messages. Adding multiple -v will increase the verbosity, but reasonable level to start is -vvv, connection debugging might require -vvvv.

Commands to manage Radiator VNF hosts

> **NOTE:** The command-line commands require the Ansible vault password, which is queried from the user in the beginning of the operation. Some playbooks require the Ansible vault password on multiple tasks so the Ansible vault password might be queried several times.
>
> However, regardless of how many times the vault password is queried, all queries will happen at the beginning of the playbook run.

Upload OpenStack credentials

Create Radiator VNF Manager credentials file called *credentials.yaml* to /var/lib/radiatorvnf/incoming/ directory. See chapter *Radiator VNF Manager credentials file* in
[Deployment and configuration guide for Radiator VNF Manager](#) for more information about the file.

To upload OpenStack credentials to the Radiator VNF Manager, run the command:

```
ansible-playbook /var/lib/radiatorvnf/flex-
ansible/openstack_credentials_upload.yml --ask-vault-password
```

The command expects to find the credentials.yaml file from /var/lib/radiatorvnf/incoming/ directory, reads it and creates an OpenStack clouds.yaml file */etc/openstack/clouds.yaml* which is then used by rest of the operations.

Upload OpenStack image

Copy the qcow2 image to be uploaded to the Radiator VNF Manager disk.

To upload the OpenStack image from Radiator VNF Manager to the OpenStack, run the command:

```
ansible-playbook /var/lib/radiatorvnf/flex-ansible/openstack_image_upload.yml -e
"imagefile=<full path to qcow2 image file> imagename=<image name in cloud>
cloudid=<cloud>" --ask-vault-password
```

Command parameters

- *imagefile*

  Full path to the OpenStack qcow2 image file on Radiator VNF Manager disk.

- *imagename*

  Name of the OpenStack image that will be be created.

Import image from Radiator VNF Manager to OpenStack

Radiator VNF Manager may contain a cloud image created by the operating system vendor for initial testing purposes.

To import the OpenStack image possibly available in the Radiator VNF Manager, run the command:

```
ansible-playbook /var/lib/radiatorvnf/flex-ansible/openstack_image_import.yml -e
"imagefile=<full path to qcow2 image file> imagename=<image name in cloud>
cloudid=<cloud>" --ask-vault-password
```

The available cloud image filename is provided when running the command without needed information:

```
ansible-playbook /var/lib/radiatorvnf/flex-ansible/openstack_image_import.yml --
ask-vault-password
```

The command will fail without the needed information, but the output of the command shows the available cloud images:

```
TASK [Command line parameter check]
*********************************************************************************
*********************************************************************
fatal: [localhost]: FAILED! => changed=false
  msg: 'Command line is missing mandatory parameters! cloudid, imagefile and imagename
must be defined. Available imagefiles are: /var/lib/radiatorvnf/flex-
ansible/files/cloudimages/AlmaLinux-9-GenericCloud-9.0-20221102.x86_64.qcow2'
```

Command parameters

- *imagefile*

  Full path to the OpenStack qcow2 image file on Radiator VNF Manager disk.

- *imagename*

  Name of the OpenStack image that will be be created.

Upload configuration file

Create Radiator VNF Manager configuration file called *cloudinfo.yaml* to /var/lib/radiatorvnf/incoming/ directory. See chapter *Radiator VNF Manager configuration file* in
Deployment and configuration guide for Radiator VNF Manager for more information about the file.

To upload the configuration to the Radiator VNF Manager, run the command:

```
ansible-playbook /var/lib/radiatorvnf/flex-
ansible/radiatorvnfm_configuration_upload.yml --ask-vault-password
```

The command expects to find the cloudinfo.yaml file from /var/lib/radiatorvnf/incoming/ directory, reads it and populates the Radiator VNF Manager internal information so it can be used by rest of the operations.

Upload configuration package

Create Radiator VNF Manager configuration file called *cloudinfo.yaml* to /var/lib/radiatorvnf/incoming/ directory and a package containing all the files referred to in the Radiator VNF Manager configuration file. See chapter *Radiator VNF Manager configuration file* in
Deployment and configuration guide for Radiator VNF Manager for more information about the file.

To upload the configuration and package containing all needed files, run the command:

```
ansible-playbook /var/lib/radiatorvnf/flex-
ansible/radiatorvnfm_configuration_upload.yml -e "packagename=/tmp/package.tar.gz
packagedimport=true" --ask-vault-password
```

The command expects to find the cloudinfo.yaml file from /var/lib/radiatorvnf/incoming/ directory, reads it and populates the Radiator VNF Manager internal information so it can be used by rest of the operations. The imported package full path has to be provided as command parameter. Suitable package types are .zip, .tar, .tar.gz, .tar.bz2, .tar.xz, .tar.zst and .gz, .bz2, .xz, or .zst files that contains a .tar archive. The package is unpacked and removed, and it's contents are moved to /var/lib/radiatorvnf/incoming/ directory for further processing by other operations.

Command parameters

- *packagename*

  Full path to the configuration upload package on Radiator VNF Manager disk.

- *packagedimport*

  Must be defined true when packaged import is done.

Create OpenStack hosts

To create the Radiator VNF hosts to the OpenStack, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud>
/var/lib/radiatorvnf/flex-ansible/openstack_create_hosts.yml --ask-vault-password
```

The command creates the Radiator VNF hosts to OpenStack according to the uploaded Radiator VNF Manager configuration file.

> **NOTE:** Do not remove hosts from the Radiator VNF Manager configuration file unless they have been destroyed via Radiator VNF Manager. Each configuration file run adds new information or updates existing information, but does not remove information.

Setup OpenStack hosts

To set up the Radiator VNF hosts in the OpenStack, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud>
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_setup_hosts.yml --ask-vault-
password
```

The command installs Radiator and Radiator Radius::UtilXS library and deploys the Radiator configuration to the Radiator VNF host if the *radiator_installed* was true. In addition, any *enhancements* defined in the configuration are processed. See chapter *Radiator VNF Manager configuration file* in Deployment and configuration guide for Radiator VNF Manager for more information about the options.

If the *radiator_installed* is true but there are no Radiator configuration files available, this operation will fail. To upload the needed Radiator configuration files run Upload configuration package command.

Start Radiator instances

To start some Radiator instances in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud> -l "
<instance1, instance2>" /var/lib/radiatorvnf/flex-
ansible/radiatorvnfm_radiator_start.yml --ask-vault-password
```

To start all Radiator instances in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud>
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_start.yml --ask-vault-
password
```

To start all Radiator instances in all the clouds, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_start.yml --ask-vault-
password
```

Radiator instance name is for example *radiator@example*.

This command will start the selected Radiator instances.

Stop Radiator instances

To stop some Radiator instances in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud> -l "
<instance1, instance2>" /var/lib/radiatorvnf/flex-
ansible/radiatorvnfm_radiator_stop.yml --ask-vault-password
```

To stop all Radiator instances in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud>
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_stop.yml --ask-vault-
password
```

To stop all Radiator instances in all the clouds, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_stop.yml --ask-vault-
password
```

Radiator instance name is for example *radiator@example*.

This command will stop the selected Radiator instances.

Restart Radiator instances

To restart some Radiator instances in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud> -l "
<instance1, instance2>" /var/lib/radiatorvnf/flex-
ansible/radiatorvnfm_radiator_restart.yml --ask-vault-password
```

To restart all Radiator instances in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud>
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_restart.yml --ask-vault-
password
```

To restart all Radiator instances in all the clouds, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_restart.yml --ask-vault-
password
```

Radiator instance name is for example *radiator@example*.

This command will restart the selected Radiator instances.

Update Radiator configuration

To update the Radiator configuration on some Radiator instances in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud> -l
<radiator_instance> /var/lib/radiatorvnf/flex-
ansible/radiatorvnfm_radiator_config_update.yml --ask-vault-password
```

To update the Radiator configurations on all Radiator instances in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud>
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_config_update.yml --ask-
vault-password
```

To update the Radiator configurations on all Radiator instances in all the clouds, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_config_update.yml --ask-
vault-password
```

Radiator instance name is for example *radiator@example*.

This command will update the Radiator configuration on the selected Radiator instances. Before the configuration can be updated, it must be uploaded to the Radiator VNF Manager with Upload configuration package command.

> **NOTE:** The command will restart all selected Radiator instances.

Label Radiator configuration

To update the Radiator configuration on some Radiator instances in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud> -l "
<instance1>, <instance2>" /var/lib/radiatorvnf/flex-
ansible/radiatorvnfm_radiator_config_label.yml -e "tagname=this_is_a_tag" --ask-
vault-password
```

To update the Radiator configurations on all Radiator instances in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud>
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_config_label.yml -e
"tagname=working" --ask-vault-password
```

Radiator instance name is for example *radiator@example*.

This command will label the currently active configuration as known to the Radiator VNF Manager with the given tag. The tag must be unique across the Radiator configuration repositories and well formed (rules available in https://git-scm.com/docs/git-check-ref-format).

Command parameters

- *tagname*

    Unique tag to label the currently active configuration with.

Switch Radiator configuration

To switch the Radiator configuration on some Radiator instances in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud> -l
<radiator_instance> /var/lib/radiatorvnf/flex-
ansible/radiatorvnfm_radiator_config_switch.yml -e "switchtag=working" --ask-
vault-password
```

To switch the Radiator configuration on all Radiator instances in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud>
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_config_switch.yml -e
"switchtag=working" --ask-vault-password
```

The command will fail if run without the mandatory parameter, but the output of the command shows the available tags:

```
TASK [Command line parameter check]
*********************************************************************************
*****************************************************************
fatal: [radiator@vv1]: FAILED! => changed=false
msg: |-
Command line is missing mandatory parameter! switchtag must be defined. Possible user
defined tags are:
- original
and possible automatically generated tags are:
- 2023-03-07T112912
- 2023-03-08T135201
```

> **NOTE:** The command will fail if the tag is not available for all the Radiator instances in the cloud. In that case run the command to specific Radiator instances.

This command will switch (or rollback) to previously labeled Radiator configuration available across the selected Radiator instances.

Command parameters

- *switchtag*

  Tag to switch/rollback to previously labeled configuration.

Update hosts

To update some Radiator VNF hosts in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud> -l
<host1> /var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_hosts_update.yml -
-ask-vault-password
```

To update all Radiator VNF hosts in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud>
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_hosts_update.yml --ask-
vault-password
```

To update all Radiator VNF hosts in all the clouds, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories -l <host1>
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_hosts_update.yml --ask-
vault-password
```

This command will update the selected Radiator VNF hosts. If the selected host has Radiator installed, this command does not update it nor restart the Radiator instances.

Destroy hosts

To destroy some Radiator VNF hosts in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud> -l
<host> /var/lib/radiatorvnf/flex-ansible/openstack_destroy_hosts.yml --ask-vault-
password
```

To destroy all Radiator VNF hosts in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud>
/var/lib/radiatorvnf/flex-ansible/openstack_destroy_hosts.yml --ask-vault-password
```

This command will destroy the selected Radiator VNF hosts and the Radiator instances deployed on the host. Destroyed host is not recreated unless the Radiator VNF Manager configuration file is uploaded that has configuration for the Radiator VNF host.

Show configuration

To show the configuration known to the Radiator VNF Manager, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud>
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_configuration_show.yml --ask-vault-
password
```

This command will show the uploaded combined configuration for the selected cloud. The configuration is shown according to what Radiator VNF Manager knows, that is the shown configuration is not necessarily yet deployed to the hosts, but it has been uploaded to the Radiator VNF Manager.

Commands to manage Radiator VNF Manager

Import certificates to Radiator VNF Manager

Create server certificate file and certificate key file to the Radiator VNF Manager disk in PEM format. The server certificate file may include intermediate certificates in addition to a server certificate, in such case the order must be server certificate first and then intermediate certificates.

To import the certificate, run the command:

```
ansible-playbook /var/lib/radiatorvnf/flex-
ansible/radiatorvnfm_certificate_import.yml -e "servercert=/tmp/server-crt.pem
serverkey=/tmp/server-key.pem" --ask-vault-password
```

The server key password is asked interactively. If the key is not password protected, run the command:

```
ansible-playbook /var/lib/radiatorvnf/flex-
ansible/radiatorvnfm_certificate_import.yml -e "servercert=/tmp/server-crt.pem
serverkey=/tmp/server-key.pem serverkeypw=''" --ask-vault-password
```

This command processes the certificate files to proper locations under /etc directory.

Command parameters

- *servercert*

  Full path to server certificate file on the Radiator VNF Manager disk.

- *serverkey*

  Full path to server certificate key file on the Radiator VNF Manager disk.

Export Radiator VNF Manager data

To export the Radiator VNF Manager data, run the command:

```
ansible-playbook /var/lib/radiatorvnf/flex-ansible/radiatorvnfm_export.yml --ask-
vault-password
```

This command exports the Radiator VNF Manager configuration regarding the Radiator VNF hosts and Radiator instances and some Radiator VNF Manager specific configuration. Copy the export package from the

Radiator VNF Manager to a safe location.

> **NOTE:** When exporting the data due to replacing the Radiator VNF Manager, run
> Commission Radiator VNF Manager before the current Radiator VNF Manager is removed. This will
> ensure the Radiator VNF hosts internal firewall accept the new Radiator VNF Manager after importing
> the package.

Import Radiator VNF Manager exported data

> **NOTE:** Exported data can only be imported to an empty Radiator VNF Manager. Empty Radiator VNF
> Manager means deployed and ready-to-use Radiator VNF Manager where none of the playbooks have
> been run. See chapters *Radiator VNF Manager deployment* and *Radiator VNF Manager set up* in
> Deployment and configuration guide for Radiator VNF Manager on how to deploy and set up Radiator
> VNF Manager.

Copy the export package to the Radiator VNF Manager disk.

To import the Radiator VNF Manager data, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_import.yml -e
"exportfile=/tmp/radiator_vnfm_export.gz" --ask-vault-password
```

This command unpacks the export package and processes all the data available. The exported package and
any temporary files created while unpacking are removed automatically. Once the import succeeds, the
Radiator VNF hosts can be managed from the newly imported Radiator VNF Manager, provided that the
Commission Radiator VNF Manager was run to allow it.

> **NOTE:** The import operation will most likely fail if the Commission Radiator VNF Manager job has not
> been run to allow the imported Radiator VNF Manager to manage the existing Radiator VNF hosts. If
> this happens, simply remove the newly imported Radiator VNF Manager and follow the instructions on
> the Deployment and configuration guide for Radiator VNF Manager chapter *Radiator VNF Manager
> upgrade process* to perform successful import and subsequent tasks.

After verifying successful operation with Radiator VNF hosts from the imported Radiator VNF Manager, run
Decommission Radiator VNF Manager to ensure the Radiator VNF hosts firewall does not have anymore the
obsolete Radiator VNF Manager IP details

Command parameters

- *exportfile*

    Full path to Radiator VNF Manager export package on the Radiator VNF Manager disk.

Export Radiator VNF Manager rootCA

Copy rootCA file from Radiator VNF Manager directory /var/lib/radiatorvnf/certificates/local/root-CA-crt.pem

Commission Radiator VNF Manager

To commission Radiator VNF Manager to some Radiator VNF hosts in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud> -l
<host> /var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_hosts_manager.yml -
e "action=enabled sourceip='<10.10.10.10/32, 20.20.20.20/32'>" --ask-vault-
password
```

To commission Radiator VNF Manager to all Radiator VNF hosts in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud>
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_hosts_manager.yml -e
"action=enabled sourceip=<10.10.10.10/32>" --ask-vault-password
```

To commission Radiator VNF Manager to all Radiator VNF hosts in all the clouds, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_hosts_manager.yml -e
"action=enabled sourceip=<10.10.10.10/32>" --ask-vault-password
```

> **NOTE:** Defining 0.0.0.0/0 as *sourceip* with *action=enabled* will lift the restriction and allow any Radiator VNF Manager regardless of the IP address to manage the hosts.

Command parameters

- *action*

  Possible values are *enabled* and *disabled*.

- *sourceip*

  IP address to be added to the Radiator VNF hosts. IP address can be with or without CIDR. Multiple IP addresses can be specified with comma.

Decommission Radiator VNF Manager

To decommission Radiator VNF Manager from some Radiator VNF hosts in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud> -l
<host> /var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_hosts_manager.yml -
e "action=disabled sourceip='<10.10.10.10/32, 20.20.20.20/32>'" --ask-vault-
password
```

To decommission Radiator VNF Manager from all Radiator VNF hosts in the cloud, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories/<cloud>
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_hosts_manager.yml -e
"action=disabled sourceip=<10.10.10.10/32>" --ask-vault-password
```

To decommission Radiator VNF Manager from all Radiator VNF hosts in all the clouds, run the command:

```
ansible-playbook -i /var/lib/radiatorvnf/flex-ansible/inventories
/var/lib/radiatorvnf/flex-ansible/radiatorvnfm_radiator_hosts_manager.yml -e
"action=disabled sourceip=<10.10.10.10/32>" --ask-vault-password
```

> **NOTE:** Defining 0.0.0.0/0 as *sourceip* with *action=disabled* will limit the allowed Radiator VNF Managers that are able to manage the hosts. Before setting the limitation, run
> [Commission Radiator VNF Manager](#) to guarantee that some Radiator VNF Manager is able to manage the hosts.

Command parameters

- *action*

  Possible values are *enabled* and *disabled*.

- *sourceip*

  IP address to be removed from the Radiator VNF hosts. IP address can be with or without CIDR. Multiple IP addresses can be specified with comma.

Define proxy for Radiator VNF Manager

To define Radiator VNF Manager to use a proxy, run the command:

```
ansible-playbook /var/lib/radiatorvnf/flex-ansible/radiatorvnfm_proxyuse.yml -e
"proxyhost=<proxy.example.com> proxyport=<8080>" --ask-vault-password
```

Command parameters

- *proxyhost*

  Proxy address.

- *proxyport*

  Proxy port.

Configure LDAP authentication to Radiator VNF Manager

Copy the LDAP CA certificate file to the Radiator VNF Manager disk if CA certificate is needed.

To configure Radiator VNF Manager to use LDAP authentication, run the command:

```
ansible-playbook /var/lib/radiatorvnf/flex-ansible/radiatorvnfm_ldapuse.yml -e
"ldapuri=<LDAP URI> ldapbinddn=<LDAP search DN> ldapsearchbase=<LDAP search base>
ldapschema=<LDAP schema> ldapfilter=<Search filter criteria> ldapsshpubkey=
<Variable where user's SSH public key is stored> ldapcertfile=<LDAP CA
certificate> ldapadmingroups=<LDAP user group> ldapusergroups=<LDAP user group>" -
-ask-vault-password
```

The bind DN password is asked interactively.

Command parameters

- *ldapuri*

  LDAP URI, only LDAPS is supported, for example ldaps://ldaps.example.com:636

- *ldapbinddn*

  The DN used to search the LDAP, for example cn=adminuser,ou=people,dc=example,dc=com

- *ldapbindpw*

  The password of the bind DN

- *ldapsearchbase*

  The LDAP search base when looking for entries, for example dc=example,dc=com

- *ldapschema*

  The Schema Type in use on the LDAP server. Schema affects for example what default attribute names are looked for, like group membership can be listed by name or DN. Possible values are *rfc2307*, *rfc2307bis*, *IPA* and *AD*. Default is **rfc2307**.

- *ldapfilter*

  Filter criteria for who are allowed to access. For example limiting access to the "allowedusers" group: memberOf=cn=allowedusers,ou=Groups,dc=example,dc=com

- *ldapsshpubkey*

  LDAP authentication can be used with SSH access also, if the user's SSH public key is available in LDAP. Specify the LDAP attribute that contains the user's SSH public key, for example: nsSshPublicKey

- *ldapcertfile*

  The file containing CA certificates for the LDAP when LDAPS is used

- *ldapadmingroups*

  List of LDAP groups, separated by pipe, where authenticated users belong that should have administrative access to Radiator VNF Manager GUI. For example group1|group2|group3

- *ldapusergroups*

  List of LDAP groups, separated by pipe, where authenticated users belong that should have user access to Radiator VNF Manager. For example group1|group2|group3

This command will configure LDAP authentication to Radiator VNF Manager GUI and optionally also for SSH access. For the Radiator VNF Manager GUI, the LDAP authentication overwrites the plain username/password authentication which means it is no longer available. For the SSH, the LDAP authentication is additional authentication, the previously available SSH authentication is not affected.