7

VM_MAKER Command Reference

The vm_maker utility provides commands to create, destroy, and otherwise manipulate Oracle Linux KVM guests.

- About the vm_maker Utility
  You can use the vm_maker command-line utility to administer an Oracle Linux KVM guest.
- add-bonded-bridge Command
  Use the --add-bonded-bridge command to add a bridge over a bonded Ethernet interface.
- add-single-bridge Command
  Use the --add-single-bridge command to add a bridge over a single Ethernet interface.
- allocate-bridge Command
  Use the --allocate-bridge command to associate a network bridge with a guest.
- attach Command
  Use the --attach command to attach a disk image to a guest.
- autostart Command
  Use the --autostart command to manage automatic starting of guests.
- boot-from-hd Command
  Use the --boot-from-hd command to configure a guest to boot using its primary boot device after a previous --boot-from-iso command.
- boot-from-iso Command
  Use the --boot-from-iso command to configure a guest to boot from a specified diagnostic ISO file.
- change-sf-vlan Command
  Use the --change-sf-vlan command to specify the cluster and storage VLANs for Exadata Secure RDMA Fabric Isolation.
- check Command
  Use the --check command to check and display the system configuration status
- cleanup-ip Command
  Use the --cleanup-ip command to remove and clean up a network interface plumbed by the --setup-ip command.
- configure-system Command
  Use the --configure-system command to configure the system.
- console-log Command
  Use the --console-log command to manage logging of console output for an Oracle Linux KVM guest.
- create Command
  Use the --create command to create a disk image.
- detach Command
  Use the --detach command to detach a disk image from a guest.

- dumpxml Command
  Use the --dumpxml command to send the XML definition of the specified guest to standard output.
- list Command
  Use the --list command to display information about guests.
- list-domains Command
  Use the --list-domains command to list all guests.
- make-base-image Command
  Use the --make-base-image command to enable UEFI booting in the base system image for KVM guests.
- network-discovery Command
  Use the --network-discovery command to identify the network for the Oracle RAC single client access name (SCAN) and other interfaces.
- reboot Command
  Use the --reboot command to restart a specific guest.
- remove-bridge Command
  Use the --remove-bridge command to remove a network bridge.
- remove-domain Command
  Use the --remove-domain command to stop and remove a specific guest or all guests.
- secure-boot Command
  Use the --secure-boot command to control the KVM Guest Secure Boot feature.
- set Command
  Use the --set command to configure vCPUs or memory for a guest.
- setup-ip Command
  Use the --setup-ip command to plumb a network interface on a KVM host.
- start-domain Command
  Use the --start-domain command to create and start an Oracle Linux KVM guest.
- stop-domain Command
  Use the --stop-domain command to stop the specified Oracle Linux KVM guest.
- update-mac Command
  Use the --update-mac command to update the guest network configuration.
- validate-ip Command
  Use the --validate-ip command to validate specified network details for an Oracle Linux KVM guest.
- vhostmd Command
  Use the --vhostmd command to enable, disable, or show the status of the vhostmd service.

7.1 About the vm_maker Utility

You can use the vm_maker command-line utility to administer an Oracle Linux KVM guest.

> **Note**
>
> On Exadata, direct manipulation of KVM guests using the virsh command is not permitted, except as explicitly instructed in Oracle documentation or under the direction of Oracle Support.

The vm_maker command-line utility is located on each KVM host at /opt/exadata_ovm/vm_maker.

Syntax

The general syntax for the vm_maker command is shown here. The following topics describe the individual commands.

```
vm_maker --command --option [--option ...]
```

Some commands or options take arguments, such as the guest name. Only enough characters to uniquely identify the guest name are required.

> **Note**
>
> Oracle Exadata Deployment Assistant (OEDA) contains the OEDA Command Line Interface (OEDACLI), which provides an alternative command-line utility to administer Oracle Linux KVM guests. A primary difference is that OEDACLI can orchestrate operations across multiple guests, while vm_maker is limited to operating on the Oracle Linux KVM host where it resides. Consider your requirements before deciding which utility to use.

Related Topics

- OEDA Command Line Interface

7.2 add-bonded-bridge Command

Use the --add-bonded-bridge command to add a bridge over a bonded Ethernet interface.

Syntax

```
vm_maker --add-bonded-bridge bridge_name --first-slave interface1 --second-slave interface2
[ --vlan vlanid ] [ --bond-mode { active-backup | lacp }]
```

Options

- *bridge_name*: The name to use for the new bridge. The bridge name must begin with vm and be followed by an interface name. For example, vmeth1.
- --first-slave: The first part of the interface pair that comprises the bonded bridge.
- --second-slave: The second part of the interface pair that comprises the bonded bridge.
- --vlan: Optionally, specifies the VLAN ID to use for the bonded bridge.
- --bond-mode: Optionally, specifies the bonding mode to use for the bonded bridge:

  - active-backup configures a bonded bridge where one network interface handles all of the network traffic, and the other network interface only becomes active if the first one fails. This mode provides fault tolerance and is the default setting.
  - lacp configures a bonded bridge where both network interfaces share the workload using Link Aggregation Control Protocol (LACP). This mode provides fault tolerance and increases the network bandwidth. Also known as Dynamic Link Aggregation, this bonding mode requires a switch that supports IEEE 802.3ad dynamic links.

7.3 add-single-bridge Command

Use the --add-single-bridge command to add a bridge over a single Ethernet interface.

Syntax

```
vm_maker --add-single-bridge bridge_name [ --vlan vlanid ]
```

Options

- *bridge_name*: The name to use for the new bridge. The bridge name must begin with 'vm' followed by an interface name, for example, vmeth1.
- --vlan: Optionally, specifies the VLAN ID to use for the bridge.

7.4 allocate-bridge Command

Use the --allocate-bridge command to associate a network bridge with a guest.

Syntax

```
vm_maker --allocate-bridge bridge-name --domain guest-name [ --vlan vlanid ]
```

Options

- *bridge-name*: The name of the previously created network bridge that you want to associate with the specified guest.
- *guest-name*: The name of the guest to which you want to add the network bridge.
- --vlan: Optionally, specifies the VLAN ID to use with the network bridge.

Usage Notes

The output from the --allocate-bridge command includes a series of additional manual steps that must be performed to complete the procedure.

In the first manual step, you must select a unique network interface name to use inside the KVM guest. This guest interface maps to the bridged network interface on the KVM host. You can use your naming convention for this interface name, or you can choose to carry through the interface name from the KVM host. Whatever your choice, ensure that the NAME="*guest-interface*" entry is adjusted accordingly in step 2.

The following is an example of the manual steps.

> **Note**
>
> Your procedure will be specific to your environment. So, ensure that you perform the additional manual steps outlined in your output from the --allocate-bridge command. Do not copy the commands from the following example.

```
[INFO] Please perform the following manual steps:
[INFO] 1. Determine a unique network interface name within the domain to which
[INFO]    you are attaching this interface. Typically bonded
[INFO]    interfaces are named bondeth<number>, for example 'bondeth1', and
[INFO]    non-bonded interfaces are named eth<number>, for example 'eth2'.
[INFO]    The name must be unique within the domain. In the example below
[INFO]    the name 'bondeth0' has been chosen.
[INFO] 2. Add the following line to the file
[INFO]    '/etc/udev/rules.d/70-persistent-net.rules' within the domain:
[INFO]    SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", KERNELS=="0000:00:PCI-slot-number.0", 
ATTR{type}=="1", NAME="guest-interface"
[INFO] You MUST execute steps 4 and 5! You CANNOT simply reboot from within
[INFO] the domain.
[INFO] 4. vm_maker --stop-domain guest-name
[INFO] 5. vm_maker --start-domain guest-name
[INFO] 6. Login to the domain and run the following command:
[INFO]    /opt/oracle.cellos/ipconf.pl -nocodes
[INFO]    This command will ask you for configuration information for this
[INFO]    new interface.
[INFO] NOTE: if you have more than one interface to add to this domain,
[INFO] please execute the instructions above, and then call this command again.
[INFO] The domain must be stopped and started between invocations of this
[INFO] command.
```

7.5 attach Command

Use the --attach command to attach a disk image to a guest.

Syntax

```
vm_maker --attach --disk-image image_name --domain guest_name [ --no-reflink ]
```

Options

- --disk-image: The name of the disk image you want to attach.
  - If you specify a disk image name that is not a fully-qualified file name (for example, shared.img), then the shared disk image must already exist in /EXAVMIMAGES on the KVM host. In this case, the shared disk image file is copied into the guest-specific directory at /EXAVMIMAGES/GuestImages/*guest_name*. By default, a reflink copy is created in the guest-specific directory, which references the shared image file in /EXAVMIMAGES. To attach a shared disk image file without using a reflink, add the --no-reflink option. In this case, the image file in the guest-specific directory is a discrete copy of the image file in /EXAVMIMAGES.

  - If you specify a fully-qualified disk image name in a guest-specific directory (for example, /EXAVMIMAGES/GuestImages/guest01.example.com/pv2_vgexadb.img), then the disk image file must already exist in the specified location. In this case, the disk image file can only be attached to the guest identified in the file path (for example, guest01.example.com), and no additional copy occurs.

- --domain: The name of the guest.
- --no-reflink: Use this option to create a discrete copy in the guest-specific image directory of the shared image file (not a reflink). This option is ignored for non-shared image files.

Usage Notes

- A KVM guest can have a maximum of 112 disks attached.

7.6 autostart Command

Use the --autostart command to manage automatic starting of guests.

Syntax

```
vm_maker --autostart guest_name { --enable | --disable }
```

Options

- *guest_name*: The name of the guest to configure.
- --enable: Configure the guest to start automatically when the KVM host is started.
- --disable: Configure the guest so that it does not start automatically when the KVM host is started.

7.7 boot-from-hd Command

Use the --boot-from-hd command to configure a guest to boot using its primary boot device after a previous --boot-from-iso command.

Syntax

```
vm_maker --boot-from-hd --domain guest-name
```

Options

- *guest-name*: The name of the guest that you want to configure.

Usage Notes

The --boot-from-hd command configures the guest to boot using its primary boot device. You must start the domain separately by using the --start-domain command.

7.8 boot-from-iso Command

Use the --boot-from-iso command to configure a guest to boot from a specified diagnostic ISO file.

Syntax

```
vm_maker --boot-from-iso ISO-file --domain guest-name
```

Options

- *ISO-file*: The name of the diagnostic ISO that you want to use to boot the specified guest.
- *guest-name*: The name of the guest that you want to boot using the specified ISO file.

Usage Notes

The --boot-from-iso command configures the guest to boot using the specified ISO file.

The output from the --boot-from-iso command includes additional commands to boot the guest and configure the network while in diagnostic mode.

The output also contains commands to stop the domain and reconfigure it to boot using its primary boot device. Use these commands when you are finished using the guest in diagnostic mode.

For example:

```
# vm_maker --boot-from-iso /root/home/diagnostics.iso --domain dm01vm01
[INFO] Running 'virsh undefine dm01vm01.example.com'...
[INFO] Running 'virsh define /var/log/exadatatmp/dm01vm01.example.com.xml.new.357b'...
[INFO] The domain 'dm01vm01.example.com' is ready for booting.
[INFO] Run the following command to boot from the diagnostic iso:
[INFO]
[INFO] virsh start dm01vm01.example.com --console
[INFO]
[INFO] If network is needed to be setup on the VM, run
[INFO] setup_management.sh from the console after the guest has booted.
[INFO]
[INFO] When finished, run the following commands to restore
[INFO] the domain to boot from its hard disk:
[INFO]
[INFO] vm_maker --stop-domain dm01vm01.example.com --force
[INFO] vm_maker --boot-from-hd --domain dm01vm01.example.com
```

7.9 change-sf-vlan Command

Use the --change-sf-vlan command to specify the cluster and storage VLANs for Exadata Secure RDMA Fabric Isolation.

Syntax

```
vm_maker --change-sf-vlan [ --new-cluster-vlan vlanid ] [ --new-storage-vlan vlanid ] --domain guest_name --restart-domain
```

Options

- --new-cluster-vlan: Specifies the VLAN ID to use for the cluster network partition.
- --new-cluster-vlan: Specifies the VLAN ID to use for the storage network partition.

- --domain: The name of the guest that is the subject of the command.
- --restart-domain: Restarts the guest to finalize the VLAN settings (required).

7.10 check Command

Use the --check command to check and display the system configuration status

Syntax

```
vm_maker --check
```

7.11 cleanup-ip Command

Use the --cleanup-ip command to remove and clean up a network interface plumbed by the --setup-ip command.

Syntax

```
vm_maker --cleanup-ip network_interface [ --vlan vlanid ]
```

Options

- *network_interface*: Specifies the name of the network interface to clean up.
- --vlan: Optionally, specifies the VLAN ID used with the specified network interface.

Example 7-1 Cleaning Up a Network Interface

The following shows an example of using the --cleanup-ip command to remove and clean up a network interface plumbed by the --setup-ip command.

```
# ping 192.0.2.1
PING 192.0.2.1 (192.0.2.1) 56(84) bytes of data.
From 10.32.106.12 icmp_seq=1 Destination Host Unreachable
^C
--- 192.0.2.1 ping statistics ---
2 packets transmitted, 0 received, +1 errors, 100% packet loss, time 999ms
# /opt/exadata_ovm/vm_maker --setup-ip 192.0.2.6 --interface eth3 --netmask 255.255.255.128 --gateway
192.0.2.1 --vlan 144
[INFO] Interface eth3 currently down
[INFO] Bringing up interface eth3
[INFO] Interface eth3.144 is plumbed.
# ping 192.0.2.1
PING 192.0.2.1 (192.0.2.1) 56(84) bytes of data.
64 bytes from 192.0.2.1: icmp_seq=2 ttl=255 time=0.317 ms
64 bytes from 192.0.2.1: icmp_seq=3 ttl=255 time=0.302 ms
64 bytes from 192.0.2.1: icmp_seq=4 ttl=255 time=0.307 ms
^C
--- 192.0.2.1 ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3077ms
rtt min/avg/max/mdev = 0.302/0.308/0.317/0.021 ms
# ip rule
0: from all lookup local
32757: from 192.0.2.6 to 192.0.2.1 iif eth3.144 lookup main
```

```
32758: from all to 10.32.96.5 lookup 220
32759: from 10.32.96.5 lookup 220
32760: from all to 10.32.106.0/23 lookup 210
32761: from 10.32.106.0/23 lookup 210
32762: from all to 192.168.0.4 lookup 181
32763: from 192.168.0.4 lookup 181
32764: from all to 192.168.0.3 lookup 180
32765: from 192.168.0.3 lookup 180
32766: from all lookup main
32767: from all lookup default
# ip a s | grep eth3
5: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
67: eth3.144@eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
 inet 192.0.2.6/25 scope global eth3.144
# /opt/exadata_ovm/vm_maker --cleanup-ip eth3 --vlan 144
[INFO] Deleting routing rules.
[INFO] Flushing ip-address from interface.
# ip a s | grep eth3
5: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
```

7.12 configure-system Command

Use the --configure-system command to configure the system.

Syntax

```
vm_maker --configure-system [ --no-build-base-image ] [ --qiniq ]
```

Usage Notes

This command creates /EXAVMIMAGES and creates the virtual functions. If you include the --no-build-base-image option, then vm_maker does not check for the existence of the base image. The --qinq option performs additional configuration to support Exadata Secure RDMA Fabric Isolation.

7.13 console-log Command

Use the --console-log command to manage logging of console output for an Oracle Linux KVM guest.

Syntax

```
vm_maker --console-log { enable | disable | status } --domain guest_name [ --restart-domain ]
```

Options

- --console-log enable: Configures the specified guest to write console output to a log file.

  Enabling logging on a guest where logging was previously disabled requires a guest restart to implement the change. You can use the --restart-domain option or restart the guest separately.
- --console-log disable: Configures the specified guest to not write console output to a log file.

Disabling logging on a guest where logging was previously enabled requires a guest restart to implement the change. You can use the --restart-domain option or restart the guest separately.

- --console-log status: Reports the console logging status for the specified guest.
- *guest_name*: The name of an the existing guest that is the subject of the command.
- --restart-domain: Optionally restarts the specified guest to implement a console logging change. This option is only valid in conjunction with --console-log enable or --console-log disable.

7.14 create Command

Use the --create command to create a disk image.

Syntax

```
vm_maker --create --disk-image image_name [ --size size ] [ --filesystem filesystem_type ]
 [ --from-zip zipfile ] [ --attach --domain guest_name [ --no-reflink ]]
```

Options

- --disk-image: Create a disk image using the specified image name:
  - If you specify a disk image name that is not a fully-qualified file name (for example, shared.img), a shared disk image file is created in /EXAVMIMAGES on the KVM host. In this case, the disk image file can be attached to any guest, either by adding the --attach option or by using a separate vm_maker --attach command.
  - If you specify a fully-qualified disk image name in a guest-specific directory (for example, /EXAVMIMAGES/GuestImages/guest01.example.com/pv2_vgexadb.img), a guest-specific disk image file is created in the specified location. In this case, the disk image file can only be attached to the guest identified in the file path (for example, guest01.example.com).
- --size: Optionally specifies the size of the image. If not specified, the default size is 50 GB.
- --filesystem: Specifies the filesystem type to use for the disk image. Allowed filesystem types are: ext3, ext4, and xfs. The default is xfs.
- --from-zip: Indicates the disk image should be created from the specified ZIP file.
- --attach: Use this option to directly attach the disk image to a guest after creation. If you specify this option, then you must also specify the --domain option.
  - If you are attaching a shared disk image (created in /EXAVMIMAGES), then a copy of the shared disk image file is created in the guest-specific directory at /EXAVMIMAGES/GuestImages/*guest_name*. By default, a reflink copy is created in the guest-specific directory, which references the shared image file in /EXAVMIMAGES. To attach a shared disk image file without using a reflink, add the --no-reflink option. In this case, the new disk image in the guest-specific directory is a discrete copy of the image file in /EXAVMIMAGES.
  - No additional copy occurs when you attach a non-shared image file that resides in a guest-specific directory.
- --domain: Use this option to specify which guest to attach the disk image to after creation.
- --no-reflink: Use this option to create a discrete copy in the guest-specific image directory of a shared image file (not a reflink). This option is ignored for non-shared image files.

Usage Notes

- A KVM guest can have a maximum of 112 disks attached.

7.15 detach Command

Use the --detach command to detach a disk image from a guest.

Syntax

```
vm_maker --detach --disk-image image_name --domain guest_name [ --delete ]
```

Options

- --disk image: The name of the disk image that you want to detach.
- --domain: The name of the guest from which you want to detach the disk image.
- --delete: Optionally specifies to delete the disk image from /EXAVMIMAGES. This option succeeds only if the disk image is not in use by any other guest. Otherwise, the command fails and reports an error.

7.16 dumpxml Command

Use the --dumpxml command to send the XML definition of the specified guest to standard output.

Syntax

```
vm_maker --dumpxml guest_name
```

Usage Notes

You can save the data by piping the output to a file, for example:

```
# vm_maker --dumpxml guest1.example.com > guest.xml
```

7.17 list Command

Use the --list command to display information about guests.

Syntax

To view information about disk images:

```
vm_maker --list --disk-image [ --domain guest_name ]
```

To view information about guests:

```
vm_maker --list --domain [ guest_name [ --detail ]]
```

To view information about virtual CPU configuration for guests:

```
vm_maker --list --vcpu [ --domain guest_name ]
```

To view information about memory configuration for guests:

```
vm_maker --list --memory [ --domain guest_name ]
```

Options

- --disk-image: List all shared disk images in /EXAVMIMAGES.
- –-domain: List all guests and their status. If you include the guest name, then the information displayed is limited to that guest.
- --vcpu: List the virtual CPU usage for the entire system.
- --memory: List the memory usage for the entire system.

Optional Arguments

- --domain *guest_name*: Limits the information displayed to a single guest.
- --detail: Enables detailed output.

7.18 list-domains Command

Use the --list-domains command to list all guests.

Syntax

```
vm_maker --list-domains
```

7.19 make-base-image Command

Use the --make-base-image command to enable UEFI booting in the base system image for KVM guests.

Syntax

```
vm_maker --make-base-image --uefi-boot
```

Usage Notes

Exadata support for UEFI in KVM is introduced in Oracle Exadata System Software release 24.1.0. The UEFI boot framework underpins the KVM Guest Secure Boot feature, which controls the binaries that are allowed to boot a KVM guest.

This command enables UEFI booting in the base system image. The command only applies if the KVM host is using a fresh system image (which has not been updated using patchmgr). If the KVM host has been updated using patchmgr, then you must download and install a UEFI-enabled system image.

After command completion, all new KVM guests will automatically use the UEFI boot framework in Oracle Linux KVM and have KVM Guest Secure Boot enabled. It is no longer possible to create a new guest without UEFI.

7.20 network-discovery Command

Use the `--network-discovery` command to identify the network for the Oracle RAC single client access name (SCAN) and other interfaces.

Syntax

```
vm_maker --network-discovery XML-config-file
```

Options

XML-config-file: The full file name of the guest XML configuration file. For example, /EXAVMIMAGES/conf/*domain*.xml.

7.21 reboot Command

Use the `--reboot` command to restart a specific guest.

Syntax

```
vm_maker --reboot guest_name
```

7.22 remove-bridge Command

Use the `--remove-bridge` command to remove a network bridge.

Syntax

```
vm_maker --remove-bridge bridge_name [ --vlan vlanid ] [ --force ]
```

Options

- *bridge_name*: The name of the bridge to remove. The bridge name starts with 'vm' and is followed by an interface name, for example, vmeth1.
- --vlan: Optionally, specifies the VLAN ID for the bridge.
- --force: Use this option to forcefully remove the bridge.

Usage Notes

A network bridge can be removed only if the bridge is not being used by the KVM host or other guests.

7.23 remove-domain Command

Use the `--remove-domain` command to stop and remove a specific guest or all guests.

Syntax

```
vm_maker --remove-domain { guest_name | --all [ --reset-kvmhost ]}
```

Options

- *guest_name*: The name of a specific guest to shutdown and then delete. You can use --all instead of a single guest name if you want to remove all guests.
- --reset-kvmhost: Resets the system to the state when the KVM host had no guests. This option can only be used when you specify --all for the guest name.

Usage Notes

When you use the --remove-domain command vm_maker forces a shutdown of the guest and deletes it. If -- all is specified for the guest name, then the command shutdowns and deletes all guests and intermediate image files. Underlying network bridges are removed if they are not being used anymore.

If you include the option --reset-kvmhost, then the system is reset to the state it was in before the guests were created.

7.24 secure-boot Command

Use the --secure-boot command to control the KVM Guest Secure Boot feature.

Syntax

```
vm_maker --secure-boot {enable|disable|status} { --domain domain-name | --system }
```

Options

- enable: Enables all new UEFI boot-enabled KVM guests..

  You can only enable the KVM Guest Secure Boot feature on a UEFI boot-enabled KVM guest. The KVM Guest Secure Boot feature is not available on guests configured with the non-UEFI virtual bios.

  If you enable the KVM Guest Secure Boot feature on the KVM host (using the --system option), then all new UEFI boot-enabled KVM guests are created with KVM Guest Secure Boot enabled. Enabling or disabling KVM Guest Secure Boot at the system level has no effect on existing KVM guests.

  By default, the KVM Guest Secure Boot feature is enabled on new UEFI boot-enabled KVM guests. Consequently, this option is only required to enable KVM Guest Secure Boot on a guest or system where it was previously disabled.
- disable: Disables the KVM Guest Secure Boot feature.

  You can only disable the KVM Guest Secure Boot feature on a UEFI boot-enabled KVM guest. The KVM Guest Secure Boot feature is not available on guests configured with the non-UEFI virtual bios.

  If you disable the KVM Guest Secure Boot feature on the KVM host (using the --system option), then all new UEFI boot-enabled KVM guests are created with KVM Guest Secure Boot disabled. Enabling or disabling KVM Guest Secure Boot at the system level has no effect on existing KVM guests.

- status: Outputs the status of the KVM Guest Secure Boot feature for the specified KVM guest or for the KVM host.
- --domain *domain-name*: Applies the command to the specified KVM guest.
- --system: Applies the command to the KVM host.

Usage Notes

KVM Guest Secure Boot leverages the UEFI boot framework in Oracle Linux KVM to restrict which binaries can boot the KVM guest. Exadata support for UEFI in KVM is introduced in Oracle Exadata System Software release 24.1.0. Consequently, KVM Guest Secure Boot is available only on new KVM guests starting with Oracle Exadata System Software release 24.1.0.

7.25 set Command

Use the --set command to configure vCPUs or memory for a guest.

Syntax

To configure the number of virtual CPUs for a guest:

```
vm_maker --set --vcpu vcpu [ --domain guest_name [ --config ] [ --force ]]
```

To configure the memory for a guest:

```
vm_maker --set --memory size --domain guest_name [ --restart-domain ] [ --force ]
```

To configure the MAC for specified Ethernet interface:

```
vm_maker --set --mac mac_address --interface interface_name --domain guest_name [ --restart-domain ]
```

Options

- --vcpu: Specifies the number of virtual CPUs for a guest.
- --memory: Sets the memory size for the specified guest. Memory can be specified using K, M, or G, for example, 40G or 25000M.
- --mac: Sets the media access control (MAC) address for the specified network interface.
- –-domain: When used with --vcpu or --memory, this argument indicates that the setting applies to a single guest. Otherwise the setting applies to all guests. You must use --domain when changing the memory size.
- --config: Indicates that the number of virtual CPUs should not be changed until you restart the guest. This option can only be used when modifying a single guest.
- --interface: The Ethernet network interface to configure.
- --restart-domain: Use this option to automatically restart the guest after vm_maker changes the configuration.
- --force: Forces the new setting to be implemented. This option can only be used when modifying a single guest.

Usage Notes

- Changing the number of virtual CPUs for a guest happens immediately by default, without restarting the guest.
- When changing the memory, you must restart the guest before the settings take effect.

7.26 setup-ip Command

Use the `--setup-ip` command to plumb a network interface on a KVM host.

Syntax

```
vm_maker --setup-ip ip_address --interface network_interface
  --netmask netmask --gateway gateway_address [ --vlan vlanid ]
```

```
vm_maker --setup-ip ip_address --discover
  --netmask netmask --gateway gateway_address [ --vlan vlanid ]
```

Options

- *ip_address*: Specifies the IP address to plumb on the network interface.
- `--interface`: Specifies the name of the network interface to plumb.
- `--discover`: Instructs `vm_maker` to identify the network interface to plumb instead of specifying it by name.
- `--netmask`: Specifies the IP netmask to plumb on the network interface.
- `--gateway`: Specifies the IP network gateway address to plumb on the network interface.
- `--vlan`: Optionally, specifies the VLAN ID to to plumb on the network interface.

Usage Notes

Use the `--setup-ip` command to plumb a network interface on a KVM host, which you can use for testing and troubleshooting purposes. Use the `--cleanup-ip` command to remove and clean up the network interface.

Example 7-2 Setting Up a Network Interface

The following shows an example of using the `--setup-ip` command plumb a network interface and use it to access the network gateway.

```
# ping 192.0.2.1
PING 192.0.2.1 (192.0.2.1) 56(84) bytes of data.
From 10.32.106.12 icmp_seq=1 Destination Host Unreachable
^C
--- 192.0.2.1 ping statistics ---
2 packets transmitted, 0 received, +1 errors, 100% packet loss, time 999ms
# /opt/exadata_ovm/vm_maker --setup-ip 192.0.2.6 --interface eth3 --netmask 255.255.255.128 --gateway 
192.0.2.1 --vlan 144
[INFO] Interface eth3 currently down
[INFO] Bringing up interface eth3
[INFO] Interface eth3.144 is plumbed.
# ping 192.0.2.1
```

```
PING 192.0.2.1 (192.0.2.1) 56(84) bytes of data.
64 bytes from 192.0.2.1: icmp_seq=2 ttl=255 time=0.317 ms
64 bytes from 192.0.2.1: icmp_seq=3 ttl=255 time=0.302 ms
64 bytes from 192.0.2.1: icmp_seq=4 ttl=255 time=0.307 ms
^C
--- 192.0.2.1 ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3077ms
rtt min/avg/max/mdev = 0.302/0.308/0.317/0.021 ms
# ip rule
0: from all lookup local
32757: from 192.0.2.6 to 192.0.2.1 iif eth3.144 lookup main
32758: from all to 10.32.96.5 lookup 220
32759: from 10.32.96.5 lookup 220
32760: from all to 10.32.106.0/23 lookup 210
32761: from 10.32.106.0/23 lookup 210
32762: from all to 192.168.0.4 lookup 181
32763: from 192.168.0.4 lookup 181
32764: from all to 192.168.0.3 lookup 180
32765: from 192.168.0.3 lookup 180
32766: from all lookup main
32767: from all lookup default
# ip a s | grep eth3
5: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
67: eth3.144@eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
 inet 192.0.2.6/25 scope global eth3.144
# /opt/exadata_ovm/vm_maker --cleanup-ip eth3 --vlan 144
[INFO] Deleting routing rules.
[INFO] Flushing ip-address from interface.
# ip a s | grep eth3
5: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
```

7.27 start-domain Command

Use the `--start-domain` command to create and start an Oracle Linux KVM guest.

Syntax

```
vm_maker --start-domain { guest_name | XML-config-file | --all }
  [ --no-discovery ] [ --keep ] [ --locked ] [ --ssh-key key_file ]
  [ --secure-boot { enable | disable }] [ --console ] [ --console-log { enable | disable }]
```

Options

- *guest_name*: The name of an existing guest to start.
- *XML-config-file*: The full file name of the XML configuration file for creating a new guest.
- `--all`: Starts all existing domains on the host.
- `--no-discovery`: Optionally specifies to skip network discovery and bridge creation when creating a new guest.
- `--keep`: Indicates that the guest should not be removed even if the first boot fails.
- `--locked`: If specified in conjunction with the `--ssh-key` option, password-based authentication is disabled for the `root` user, and access to the guest requires key-based authentication.

Otherwise, the option forces users to choose a new password when they log in for the first time.

- --ssh-key: Optionally specifies the location of a file containing an SSH public key, which is used for authenticating the root user.
- --secure-boot: Optionally controls the KVM Guest Secure Boot feature.

  KVM Guest Secure Boot leverages the UEFI boot framework in Oracle Linux KVM to restrict which binaries can boot the KVM guest. Exadata support for UEFI in KVM is introduced in Oracle Exadata System Software release 24.1.0. Consequently, KVM Guest Secure Boot is available only on new UEFI boot-enabled KVM guests starting with Oracle Exadata System Software release 24.1.0.

  - --secure-boot enable: Enables KVM Guest Secure Boot on the KVM guest. This is the default for all new UEFI boot-enabled KVM guests.
  - --secure-boot disable: Creates the KVM guest with KVM Guest Secure Boot disabled.
- --console: Optionally causes vm_maker to display console messages when starting existing guests.
- --console-log: Optionally specifies how logging of console output is handled:
  - --console-log enable: Console output is written to a log file.
  - --console-log disable: Console output is not written to a log file.

7.28 stop-domain Command

Use the --stop-domain command to stop the specified Oracle Linux KVM guest.

Syntax

```
vm_maker --stop-domain { guest_name | --all } [ --force | --destroy ]
```

Options

- *guest_name*: The name of the guest to stop.
- --all: Stops all guests instead of a specific one.
- --force: Attempts to flush the cache for the disk image files and immediately shuts down the guest.
- --destroy: Immediately shuts down the guest, without flushing the cache for the disk image files. Use --destroy only as a last resort if the guest is unresponsive.

Usage Notes

Without --force or --destroy, a graceful shutdown occurs.

7.29 update-mac Command

Use the --update-mac command to update the guest network configuration.

Syntax

```
vm_maker --update-mac --domain guest_name
```

Options

- --domain: The name of the guest that is the subject of the command.

Usage Notes

This command is primarily used to update the guest network configuration after a guest is moved to a different KVM host.

Related Topics

- Manually Moving a Guest to a Different KVM Host
  You can manually move an Oracle Linux KVM guest to a different KVM host.

7.30 validate-ip Command

Use the --validate-ip command to validate specified network details for an Oracle Linux KVM guest.

Syntax

```
vm_maker --validate-ip ip_address --interface network_interface
  --netmask netmask --gateway gateway_address [ --vlan vlanid ]
```

```
vm_maker --validate-ip ip_address --discover
  --netmask netmask --gateway gateway_address [ --vlan vlanid ]
```

Options

- *ip_address*: Specifies the IP address to use in the validation.
- --interface: Instructs vm_maker to perform interface validation and specifies the network interface to use in the validation.
- --discover: Instructs vm_maker to perform discovery validation.
- --netmask: Specifies the IP netmask to use in the validation.
- --gateway: Specifies the IP network gateway address to use in the validation.
- --vlan: Optionally, specifies the VLAN ID to use in the validation.

Usage Notes

Use the --validate-ip command to verify if the provided network details are valid to use for a KVM guest. It can be used for troubleshooting purposes during deployment of KVM guests.

Two types of validation are available:

- When you use the --interface option, vm_maker validates the specified interface by temporarily plumbing the specified network with the provided details and checking that it can reach the specified gateway.
- When you use the --discover option, vm_maker discovers interfaces that valid for the specified network details.

Example 7-3 Interface Validation

The following shows an example of interface validation using the --interface option.

```
# /opt/exadata_ovm/vm_maker --validate-ip 192.0.2.34 --netmask 255.255.240.0 --interface eth1 --gateway 192.0.2.1
[INFO] Interface eth1 is part of bondeth0. Validating bondeth0.
[INFO] Validating network settings, this can take up to a minute.
[INFO] Interface bondeth0 with ip address 192.0.2.34 and netmask 255.255.240.0 can ping gateway 192.0.2.1
```

Example 7-4 Discovery Validation

The following shows an example of discovery validation using the --discover option.

```
# /opt/exadata_ovm/vm_maker --validate-ip 192.0.2.34 --netmask 255.255.240.0 --discover --gateway 192.0.2.1
[INFO] Potential interfaces for discovery: eth0 eth1 eth2 eth3 eth4
[INFO] Discovering interface eth0
[INFO] Discovering interface eth1
[INFO] Discovering interface eth2
[INFO] Discovering interface eth3
[INFO] Discovering interface eth4
[INFO] Interfaces found: eth1 eth2
```

7.31 vhostmd Command

Use the --vhostmd command to enable, disable, or show the status of the vhostmd service.

Syntax

```
vm_maker --vhostmd { enable | disable | status }
```