

# Erfahrungen mit Stackit CLI und Vergleich mit Openstack CLI

## 1. Vergleich Stackit CLI help mit Openstack CLI help

### Stackit CLI help

```
stackit server create -h
Creates a server.
```

#### USAGE

```
stackit server create [flags]
```

#### EXAMPLES

Create a server from an image with id xxx

```
$ stackit server create --machine-type t1.1 --name server1 --image-id xxx
```

Create a server with labels from an image with id xxx

```
$ stackit server create --machine-type t1.1 --name server1 --image-id xxx --labels key=value,foo=bar
```

Create a server with a boot volume

```
$ stackit server create --machine-type t1.1 --name server1 --boot-volume-source-id xxx --boot-volume-source-type image --boot-volume-size 64
```

Create a server with a boot volume from an existing volume

```
$ stackit server create --machine-type t1.1 --name server1 --boot-volume-source-id xxx --boot-volume-source-type volume
```

Create a server with a keypair

```
$ stackit server create --machine-type t1.1 --name server1 --image-id xxx --keypair-name example
```

Create a server with a network

```
$ stackit server create --machine-type t1.1 --name server1 --image-id xxx --network-id yyy
```

Create a server with a network interface

```
$ stackit server create --machine-type t1.1 --name server1 --boot-volume-source-id xxx --boot-volume-source-type image --boot-volume-size 64 --network-interface-ids yyy
```

Create a server with an attached volume

```
$ stackit server create --machine-type t1.1 --name server1 --boot-volume-source-id xxx --boot-volume-source-type image --boot-volume-size 64 --volumes yyy
```

Create a server with user data (cloud-init)

```
$ stackit server create --machine-type t1.1 --name server1 --boot-volume-source-id xxx --boot-volume-source-type image --boot-volume-size 64 --user-data @path/to/file.yaml")
```

#### FLAGS

<code>--affinity-group string</code>	The affinity group the server is assigned to
<code>--availability-zone string</code>	The availability zone of the server
<code>--boot-volume-delete-on-termination</code>	Delete the volume during the termination of the server. Defaults to false
<code>--boot-volume-performance-class string</code>	Boot volume performance class
<code>--boot-volume-size int</code>	The size of the boot volume in GB. Must be provided when 'boot-volume-source-type' is 'image'
<code>--boot-volume-source-id string</code>	ID of the source object of boot volume. It can be either an image or volume ID
<code>--boot-volume-source-type string</code>	Type of the source object of boot volume. It can be either 'image' or 'volume'
<code>-h, --help</code>	Help for "stackit server create"
<code>--image-id string</code>	The image ID to be used for an ephemeral disk on the server. Either 'image-id' or 'boot-volume-...' flags are required
<code>--keypair-name string</code>	The name of the SSH keypair used during the server creation
<code>--labels stringToString</code>	Labels are key-value string pairs which can be attached to a server. E.g. '--labels key1=value1,key2=value2,...' (default [])
<code>--machine-type string</code>	Name of the type of the machine for the server. Possible values

are documented in <https://docs.stackit.cloud/stackit/en/virtual-machine-flavors-75137231.html>

-n, --name string	Server name
--network-id string	ID of the network for the initial networking setup for the server creation
--network-interface-ids strings	List of network interface IDs for the initial networking setup for the server creation
--security-groups strings	The initial security groups for the server creation
--service-account-emails strings	List of the service account mails
--user-data string	User data that is passed via cloud-init to the server
--volumes strings	The list of volumes attached to the server

#### GLOBAL FLAGS

-y, --assume-yes	If set, skips all confirmation prompts
--async	If set, runs the command asynchronously
-o, --output-format string	Output format, one of ["json" "pretty" "none" "yaml"]
-p, --project-id string	Project ID
--region string	Target region for region-specific requests
--verbosity string	Verbosity of the CLI, one of ["debug" "info" "warning" "error"] (default "info")

## Openstack CLI help

openstack server create -h

```
usage: openstack server create [-h] [-f {json,shell,table,value,yaml}]
                               [-c COLUMN] [--noindent] [--prefix PREFIX]
                               [--max-width <integer>] [--fit-width]
                               [--print-empty] --flavor <flavor>
                               [--image <image> | --image-property <key=value> | --volume <volume> | --snapshot
                               <snapshot>]
                               [--boot-from-volume <volume-size>]
                               [--block-device-mapping <dev-name=mapping>]
                               [--block-device <block-device>] [--swap <swap>]
                               [--ephemeral <size=size[,format=format]>]
                               [--network <network>] [--port <port>]
                               [--no-network] [--auto-network]
                               [--nic <net-id=net-uuid,port-id=port-uuid,v4-fixed-ip=ip-addr,v6-fixed-ip=ip-
                               addr,tag=tag,auto,none>]
                               [--password <password>]
                               [--security-group <security-group>]
                               [--key-name <key-name>]
                               [--property <key=value>]
                               [--file <dest-filename=source-filename>]
                               [--user-data <user-data>]
                               [--description <description>]
                               [--availability-zone <zone-name>]
                               [--host <host>]
                               [--hypervisor-hostname <hypervisor-hostname>]
                               [--server-group <server-group>]
                               [--hint <key=value>]
                               [--use-config-drive | --no-config-drive | --config-drive <config-drive-
                               volume>|True]
                               [--min <count>] [--max <count>] [--tag <tag>]
                               [--hostname <hostname>] [--wait]
                               [--trusted-image-cert <trusted-cert-id>]
                               <server-name>
```

Create a new server

positional arguments:

<server-name>	New server name
---------------	-----------------

optional arguments:

-h, --help	show this help message and exit
--flavor <flavor>	Create server with this flavor (name or ID)
--image <image>	Create server boot disk from this image (name or ID)
--image-property <key=value>	Create server using the image that matches the

specified property. Property must match exactly one property.

`--volume <volume>` Create server using this volume as the boot disk (name or ID)  
This option automatically creates a block device mapping with a boot index of 0. On many hypervisors (libvirt/kvm for example) this will be device vda. Do not create a duplicate mapping using `--block-device-mapping` for this volume.

`--snapshot <snapshot>` Create server using this snapshot as the boot disk (name or ID)  
This option automatically creates a block device mapping with a boot index of 0. On many hypervisors (libvirt/kvm for example) this will be device vda. Do not create a duplicate mapping using `--block-device-mapping` for this volume.

`--boot-from-volume <volume-size>`  
When used in conjunction with the `--image` or `--image-property` option, this option automatically creates a block device mapping with a boot index of 0 and tells the compute service to create a volume of the given size (in GB) from the specified image and use it as the root disk of the server. The root volume will not be deleted when the server is deleted. This option is mutually exclusive with the `--volume` and `--snapshot` options.

`--block-device-mapping <dev-name=mapping>`  
**Deprecated** Create a block device on the server. Block device mapping in the format  
`<dev-name>=<id>:<type>:<size(GB)>:<delete-on-terminate>`  
`<dev-name>`: block device name, like: vdb, xvdc (required)  
`<id>`: Name or ID of the volume, volume snapshot or image (required)  
`<type>`: volume, snapshot or image; default: volume (optional)  
`<size(GB)>`: volume size if create from image or snapshot (optional)  
`<delete-on-terminate>`: true or false; default: false (optional)  
Replaced by `--block-device`

`--block-device <block-device>`  
Create a block device on the server.  
Either a path to a JSON file or a CSV-serialized string describing the block device mapping.  
The following keys are accepted for both:  
`uuid=<uuid>`: UUID of the volume, snapshot or ID (required if using source image, snapshot or volume),  
`source_type=<source_type>`: source type (one of: image, snapshot, volume, blank),  
`destination_type=<destination_type>`: destination type (one of: volume, local) (optional),  
`disk_bus=<disk_bus>`: device bus (one of: uml, lxc, virtio, ...) (optional),  
`device_type=<device_type>`: device type (one of: disk, cdrom, etc. (optional),  
`device_name=<device_name>`: name of the device (optional),  
`volume_size=<volume_size>`: size of the block device in MiB (for swap) or GiB (for everything else) (optional),  
`guest_format=<guest_format>`: format of device (optional),  
`boot_index=<boot_index>`: index of disk used to order boot disk (required for volume-backed instances),  
`delete_on_termination=<true|false>`: whether to delete the volume upon deletion of server (optional),  
`tag=<tag>`: device metadata tag (optional),  
`volume_type=<volume_type>`: type of volume to create

```

                (name or ID) when source is blank, image or snapshot
                and dest is volume (optional)
--swap <swap>      Create and attach a local swap block device of
                   <swap_size> MiB.
--ephemeral <size=size[,format=format]>
                   Create and attach a local ephemeral block device of
                   <size> GiB and format it to <format>.
--network <network> Create a NIC on the server and connect it to network.
                   Specify option multiple times to create multiple NICs.
                   This is a wrapper for the '--nic net-id=<network>'
                   parameter that provides simple syntax for the standard
                   use case of connecting a new server to a given
                   network. For more advanced use cases, refer to the '--
                   nic' parameter.
--port <port>      Create a NIC on the server and connect it to port.
                   Specify option multiple times to create multiple NICs.
                   This is a wrapper for the '--nic port-id=<port>'
                   parameter that provides simple syntax for the standard
                   use case of connecting a new server to a given port.
                   For more advanced use cases, refer to the '--nic'
                   parameter.
--no-network       Do not attach a network to the server. This is a
                   wrapper for the '--nic none' option that provides a
                   simple syntax for disabling network connectivity for a
                   new server. For more advanced use cases, refer to the
                   '--nic' parameter. (supported by --os-compute-api-
                   version 2.37 or above)
--auto-network     Automatically allocate a network to the server. This
                   is the default network allocation policy. This is a
                   wrapper for the '--nic auto' option that provides a
                   simple syntax for enabling automatic configuration of
                   network connectivity for a new server. For more
                   advanced use cases, refer to the '--nic' parameter.
                   (supported by --os-compute-api-version 2.37 or above)
--nic <net-id=net-uuid,port-id=port-uuid,v4-fixed-ip=ip-addr,v6-fixed-ip=ip-addr,tag=tag,auto,none>
                   Create a NIC on the server.
                   NIC in the format:
                   net-id=<net-uuid>: attach NIC to network with this
                   UUID,
                   port-id=<port-uuid>: attach NIC to port with this
                   UUID,
                   v4-fixed-ip=<ip-addr>: IPv4 fixed address for NIC
                   (optional),
                   v6-fixed-ip=<ip-addr>: IPv6 fixed address for NIC
                   (optional),
                   tag: interface metadata tag (optional) (supported by
                   --os-compute-api-version 2.43 or above),
                   none: (v2.37+) no network is attached,
                   auto: (v2.37+) the compute service will automatically
                   allocate a network.
                   Specify option multiple times to create multiple NICs.
                   Specifying a --nic of auto or none cannot be used with
                   any other --nic value.
                   Either net-id or port-id must be provided, but not
                   both.
--password <password>
                   Set the password to this server. This option requires
                   cloud support.
--security-group <security-group>
                   Security group to assign to this server (name or ID)
                   (repeat option to set multiple groups)
--key-name <key-name>
                   Keypair to inject into this server
--property <key=value>
                   Set a property on this server (repeat option to set
                   multiple values)
--file <dest-filename=source-filename>
                   File(s) to inject into image before boot (repeat
                   option to set multiple files)(supported by --os-
                   compute-api-version 2.57 or below)
--user-data <user-data>

```

```

        User data file to serve from the metadata server
--description <description>
    Set description for the server (supported by --os-
compute-api-version 2.19 or above)
--availability-zone <zone-name>
    Select an availability zone for the server. Host and
node are optional parameters. Availability zone in the
format <zone-name>:<host-name>:<node-name>, <zone-
name>::<node-name>, <zone-name>:<host-name> or <zone-
name>
--host <host>
    Requested host to create servers. (admin only)
(supported by --os-compute-api-version 2.74 or above)
--hypervisor-hostname <hypervisor-hostname>
    Requested hypervisor hostname to create servers.
(admin only) (supported by --os-compute-api-version
2.74 or above)
--server-group <server-group>
    Server group to create the server within (this is an
alias for '--hint group=<server-group-id>')
--hint <key=value>
    Hints for the scheduler
--use-config-drive
    Enable config drive.
--no-config-drive
    Disable config drive.
--config-drive <config-drive-volume>|True
    **Deprecated** Use specified volume as the config
drive, or 'True' to use an ephemeral drive. Replaced
by '--use-config-drive'.
--min <count>
    Minimum number of servers to launch (default=1)
--max <count>
    Maximum number of servers to launch (default=1)
--tag <tag>
    Tags for the server. Specify multiple times to add
multiple tags. (supported by --os-compute-api-version
2.52 or above)
--hostname <hostname>
    Hostname configured for the server in the metadata
service. If unset, a hostname will be automatically
generated from the server name. A utility such as
cloud-init is required to propagate the hostname in
the metadata service to the guest OS itself.
(supported by --os-compute-api-version 2.90 or above)
--wait
    Wait for build to complete
--trusted-image-cert <trusted-cert-id>
    Trusted image certificate IDs used to validate
certificates during the image signature verification
process. May be specified multiple times to pass
multiple trusted image certificate IDs. (supported by
--os-compute-api-version 2.63 or above)

output formatters:
  output formatter options

  -f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}
    the output format, defaults to table
  -c COLUMN, --column COLUMN
    specify the column(s) to include, can be repeated to
show multiple columns

json formatter:
  --noindent
    whether to disable indenting the JSON

shell formatter:
  a format a UNIX shell can parse (variable="value")

  --prefix PREFIX
    add a prefix to all variable names

table formatter:
  --max-width <integer>
    Maximum display width, <1 to disable. You can also use
the CLIFF_MAX_TERM_WIDTH environment variable, but the
parameter takes precedence.

  --fit-width
    Fit the table to the display width. Implied if --max-
width greater than 0. Set the environment variable
CLIFF_FIT_WIDTH=1 to always enable

```

```
--print-empty      Print empty table if there is no data to show.
```

### Fazit:

Hier sieht man z.B. dass die **Openstack CLI** Hilfe ausführlicher ist.

Hervorzuheben ist hier etwas elementares. Die **Openstack CLI** beschreibt die einzelnen Parameter genauer und erwähnt z.B., ob jeweils ein Name oder eine ID erwartet wird (name or ID) sowie ob Beziehungen zwischen Parametern bestehen (This option is mutually exclusive with the ``--volume`` and ``--snapshot`` options).

So gibt es beispielsweise bei der **Stackit CLI** eine Beziehung zwischen `--boot-volume-source-id` und `--boot-volume-source-type`, was ich aus der Hilfe nicht rauslesen konnte. Es gibt bei der **Stackit CLI** z.B. auch den Parameter `--volumes` und `--security-groups`. Bei beiden muss man die ID und nicht den Namen übergeben. Das sehe ich in der Hilfe nicht. Da helfen mir auch die Examples nichts, da hier als Wert nur xxx oder yyy angegeben wird, was ja echt alles sein kann. Weiter wäre es natürlich auch sehr hilfreich gewesen, wenn mir die **Stackit CLI** Hilfe Informationen zur Namenskonvention bei Volumenamen und Servernamen gegeben hätte (bezüglich Namenlänge und verbotene Zeichen). So musste ich das erst mühsam beim Testen und mit Nachfragen rausfinden. Den Zeit- und Nerven-Aufwand hätte ich mir sparen können, wenn ich das aus der Hilfe hätte rauslesen können.

## 2. Vergleich Stackit CLI Doku mit der Openstack CLI Doku

### 2.1. Stackit CLI Doku

**Stackit CLI** Doku: <https://docs.stackit.cloud/stackit/de/erstellung-einer-vm-mit-der-iaas-api-305529544.html#HowtocreateaVMviaIaaSAPI-ErstellungeinesNetzwerks>

Hier fehlt für mich eine intuitive Struktur.

Da gibt's z.B. die Seite "Erstellung einer VM mit der IaaS-API". Die ganzen anderen Seiten haben so Bezeichnungen, wie man dies oder das macht. Auf diese Art in dieser Doku etwas zu suchen, wäre nicht mein Stil. Ich würde mich sogar etwas aus dem Fenster lehnen und behaupten, dass irgendwelche Leute, die Erfahrung z.B. mit der **Openstack CLI** haben, auf diese Art auch nur sehr umständlich (wenn überhaupt) etwas finden. Da wünscht man sich stattdessen eher sowas wie eine Command List, um dort näheres zu den Kommandos zu erfahren (siehe z.B. Command List bei der **Openstack CLI** Doku: <https://docs.openstack.org/python-openstackclient/latest/cli/command-list.html>). Benutzer der **Openstack CLI** wissen ja bereits, was man alles braucht bzw. welche **Openstack CLI** Befehle zu verwenden sind, um zum Beispiel einen Server mit Volumes, Security Groups, Security Group Rules usw. aufzusetzen. Solche Leute suchen höchstwahrscheinlich eher nach Stackit Befehlen, die gleich oder ähnlich lauten und würden hier höchstwahrscheinlich mit einer Command List viel schneller finden, was sie brauchen.

Die Art und Weise wie die Doku bei **Stackit CLI** strukturiert ist halte ich für gar nicht intuitiv. Schaut man sich im Vergleich dazu mal die Startseite zur **Openstack CLI** Doku an, so sehe ich für mich dort auf einen Blick eine klare Struktur und kann mich direkt da reinklicken, was mich interessiert. Für mich war da bei weitem die Command List am hilfreichsten.

### 2.2. Openstack CLI Doku

Openstack CLI Doku: <https://docs.openstack.org/python-openstackclient/latest/index.html>

Hier ist das sehr gut strukturiert. Ich kann z.B. auf "Command List" klicken und sehe dann eine detaillierte Beschreibung jedes einzelnen **Openstack CLI** Befehls.

Auf <https://docs.openstack.org/python-openstackclient/latest/cli/decoder.html> findet man z.B. eine Mapping List, bei der man sich Openstack-seitig die Mühe gemacht hat, um die Befehle anderer CLIs in die **Openstack CLI** zu übersetzen bzw. bzw. die Ausführung dieser Befehle mit der **Openstack CLI** zu zeigen. Hierbei werden die Befehle der Cinder CLI, Glance CLI, Keystone CLI, Neutron CLI, Nova CLI und Swift CLI übersetzt. Man hat sich hier richtig viel Mühe gegeben, um Usern bei der Umstellung zu helfen, was mir bei der **Stackit CLI** absolut fehlt.