

ITS Private Cloud Command Line Interface Cheat Sheet

Global options account: `vss-cli`

Username	<code>--username</code>	VSS_API_USER env variable
Password	<code>--password</code>	VSS_API_USER_PASS env variable.
MFA TOTP Token	<code>--totp</code>	If MFA enabled, will be prompted.
Configuration Path	<code>--config</code>	Alternative configuration path.
NOTE: Default configuration file is available at <code>~/vss-cli/config.yaml</code>		

Global options format: `vss-cli`

Table format to use	<code>--table-format</code>	csv, fancy_grid, github, grid, html, jira, ect.
Custom columns	<code>--columns</code>	Any VM attribute: moref, name, cpu_count, ip_address
Truncates column values	<code>--columns-width</code>	0: auto, -1: disable
Don't use headers	<code>--no-headers</code>	When printing tables
Sort output	<code>--sort-by</code>	Sort table by the jsonpath expression.
Output	<code>--output</code>	json yaml table auto ndjson

Virtual Machine deployment: `vss-cli compute vm mk`

Shell VM	<pre>shell \ --description="Ubuntu OS VM deployment" \ --client={client} --usage=Prod \ --os="ubuntu64Guest" --folder=VSS-CLI-VMs \ --memory={nGiB} --cpu={n} \ --net={netName} \ --disk={nGiB} --disk='{"capacity_gb": 100}' \ --iso {isoName} \ --extra-config disk.EnableUUID=TRUE \ --power-on \ --firmware efi \ ubuntu-2004-vm</pre>
Clone VM or Template	<pre>{from-clone from-template} \ --description={description} \ --source {sourceVM} \ --custom-spec '{"hostname": "ubuntu-clone", "domain": "local", "interfaces": [{"dhcp":true}]}' \ --power-on</pre>
Clone VM (from snapshot)	<pre>from-clone \ --snapshot {snapId} \ --description={description} \ --source {sourceVM} \ --custom-spec vm-clone-spec.yaml \ --power-on</pre>
CLI Spec (existing)	<code>from-file {cli-spec-file}.{json yaml}</code>
CLI Spec (new)	<code>from-file --edit --save --spec-template {shell clib template clone}</code>
From image	<p>Note: Upload OVF and VMDK or OVA to VSKEY-STOR and run <code>vss-cli compute image personal sync</code> ahead of the following command:</p> <pre>from-image --source {image-name} --client={client} --usage=Prod \ --os="ubuntu64Guest" --folder=VSS-CLI-VMs \ --memory={nGiB} --cpu={n} \ --net={netName} \ --disk={nGiB} \ ubuntu-2004-vm</pre>

NOTE: Retirement requests can be added to shell, from-clone, from-template, from-clib as follows:

```
--retire-type {timedelta|datetime}: Retirement request type.
--retire-warning {n}: Days before retirement date to notify.
--retire-value {hours},{days},{months}: Value for given retirement type.
```

Virtual Machine query: `vss-cli {ls|get {id}}`

Search for vm	<code>vss-cli compute vm ls</code>
Search for vm by name	<code>vss-cli compute vm ls -f name={vm-name}</code>
Get a vm attribute	<code>vss-cli compute vm get {id} {cpu memory disk state nic storage-type}</code>

Virtual Machine query (advanced): `vss-cli --columns`

Search for a VM by IP address	<code>--columns moref,name,folder.path,'disks[*].capacity_gib',cpu_count,\ memory_gb,ip_address,power_state compute vm ls -f ip_address={ip}</code>
List including storage type and preferences	<code>--columns moref,name,folder.path,storage_type,preferences compute vm ls</code>

NOTE: To show all vm attributes that can be used with the `--columns` option, use:

```
vss-cli --output yaml compute vm ls -c 1
```

Virtual Machine update: vss-cli compute vm set {id}

CPU	cpu count {n}
Memory	memory size {nGiB}
Disk New (basic)	disk mk -i {nGiB} -i {nGiB}
Power State	state {on off reboot reset shutdown suspend}
Disk New (adv.)	disk mk -i '{"capacity_gb": 100, "scsi": 1}'
Disk Update	disk up {diskN} --capacity {nGiB}
Disk Update (adv.)	disk up {diskN} --capacity {nGiB} --scsi {busN} --notes "data disk e:/"
Disk Delete	disk rm {diskN} {diskN}
NIC New	nic mk -n {netName}
NIC Update Net	nic up {nicN} --network {netName}
NIC Update State	nic up {nicN} --state {connect disconnect}
NIC Update Type	nic up {nicN} --adapter {type}
NIC Delete	nic rm {nicN} {nicN}
Storage Type	storage-type {ssd hdd}
Add vTPM	tpm mk
Convert to/from template	template --{on off}

NOTE: Add `-wait` to global `vss-cli` command and get immediate change request results.**Virtual Machine Snapshot**

Create now	compute vm set {id-or-name} snapshot mk \ --description {description} --lifetime {n<72} --memory
Create later	compute vm set {id-or-name} snapshot mk \ --description {description} --lifetime {n<72} --memory \ --timestamp={YYYY-MM-DD HH:MM}
List	compute vm get {id-or-name} snapshot
Revert	compute vm set {id-or-name} snapshot re {snapId}
Remove	compute vm set {id-or-name} snapshot rm {snapId}
Extend snapshot life	request snapshot set {requestId} duration --lifetime 72

NOTE: Remember that every change, new vm submission and snapshot request create a unique ID.**Virtual Machine Inventory**

Create	compute inventory mk --fmt {json csv} --all
Download	compute inventory dl {id}
Download using the "Create" command by adding <code>--wait</code> to the global <code>vss-cli</code> .i.e. <code>vss-cli --wait compute inventory mk --fmt csv --all</code>	

Virtual Machine Folders

Create	compute folder mk --parent={folder} {folder}
Rename	compute folder set {folder} name {new-name}
Move	compute folder set {folder} parent {nparent}

OVF/OVA parsing

Get Deployment Option parameters	ovf get {filePath} deployment-params
Generate additional-params spec	ovf get {filePath} params-spec
Get property parameters	ovf get {filePath} property-params

Account MFA

Enable MFA	account set mfa mk {EMAIL AUTHENTICATOR SMS}
Disable MFA	account set mfa rm
Verify	account set mfa verify

Raw requests

<code>vss-cli raw {GET POST PUT DELETE PATCH} {resource}</code>
For example: <code>Vss-cli raw GET /status</code>

Configure: vss-cli

Initialize	configure mk
List	configure ls
Set	configure set {attr} {name}
NOTE: Default configuration file is available at <code>~/.vss-cli/config.yaml</code>	

Account Notifications

Format	account set notification format \ {html text}
Filter	account set notification request \ {all none error completion submission}
Method	account set notification method \ {mail message}

Virtual Machine decomission

<code>vss-cli compute vm rm --show-info {id} ... {idN}</code>
