

ITS Private Cloud Command Line Interface Cheat Sheet

Global options account: vss-cli

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

Global options format: vss-cli

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

Virtual Machine deployment: vss-cli compute vm mk

Shell VM	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
Clone VM or Template	{from-clone from-template} \ --description={description} \ --source {sourceVM} \ --custom-spec '{"hostname": "ubuntu-clone", "domain": "local", "interfaces": [{"dhcp":true}]}' \ --power-on
Clone VM (from snapshot)	from-clone \ --snapshot {snapId} \ --description={description} \ --source {sourceVM} \ --custom-spec vm-clone-spec.yaml \ --power-on
CLI Spec (existing)	from-file {cli-spec-file}.{json yaml}
CLI Spec (new)	from-file --edit --save --spec-template {shell clib template clone}
From image	Note: Upload OVF and VMDK or OVA to VSKEY-STOR and run vss-cli compute image personal sync ahead of the following command: 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
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 update: <code>vss-cli compute vm set {id}</code>	
CPU	<code>cpu count {n}</code>
Memory	<code>memory size {nGiB}</code>
Disk New (basic)	<code>disk mk -i {nGiB} -i {nGiB}</code>
Power State	<code>state {on off reboot reset shutdown suspend}</code>
Disk New (adv.)	<code>disk mk -i '{"capacity_gb": 100, "scsi": 1}'</code>
Disk Copy (adv.)	<code>disk cp -i '{"capacity_gb": 100, "backing_vmdk": "[XXNN] VM/vmdk"}'</code>
Disk Update	<code>disk up {diskN} --capacity {nGiB}</code>
Disk Update (adv.)	<code>disk up {diskN} --capacity {nGiB} --scsi {busN} --notes "data disk e:/"</code>
Disk Delete	<code>disk rm {diskN} {diskN}</code>
NIC New	<code>nic mk -n {netName}</code>
NIC new (adv.)	<code>nic mk --net {NetName}={NICType}</code>
NIC Update Net	<code>nic up {nicN} --network {netName}</code>
NIC Update State	<code>nic up {nicN} --state {connect disconnect}</code>
NIC Update Type	<code>nic up {nicN} --adapter {type}</code>
NIC Delete	<code>nic rm {nicN} {nicN}</code>
Storage Type	<code>storage-type {ssd hdd}</code>
Domain	<code>domain {name-or-id}</code>
Add vTPM	<code>tpm mk</code>
Enable/Disable VBS	<code>vbs on off</code>
Enable/disable secure boot	<code>secure-boot --on/--off</code>
Firmware	<code>firmware bios efi</code>
Convert to/from template	<code>template --{on off}</code>
Administrator	<code>admin "Admin Name" admin.email@utoronto.ca 416-666-6666</code>
Informational contacts	<code>inform {email-1} {email-n}</code>
NOTE: Add <code>-wait</code> to global <code>vss-cli</code> command and get immediate change request results.	

Virtual Machine query (basic): <code>vss-cli {ls get} {id}</code>	
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): <code>vss-cli --columns</code>	
Search for a VM by IP address	<code>--columns moref,name,folder.path,'disks[*].capacity_gib',cpu_count,\nmemory_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 <code>--columns</code> option, use: <code>vss-cli --output yaml compute vm ls -c 1</code>	

Virtual Machine Snapshot	
Create now	<code>compute vm set {id-or-name} snapshot mk \n--description {description} --lifetime {n<72} --memory</code>
Create later	<code>compute vm set {id-or-name} snapshot mk \n--description {description} --lifetime {n<72} --memory \n--timestamp={YYYY-MM-DD HH:MM}</code>
List	<code>compute vm get {id-or-name} snapshot</code>
Revert	<code>compute vm set {id-or-name} snapshot re {snapId}</code>
Remove	<code>compute vm set {id-or-name} snapshot rm {snapId}</code>
Extend snapshot life	<code>request snapshot set {requestId} duration --lifetime 72</code>
NOTE: Remember that every change, new vm submission and snapshot request create a unique ID.	

Virtual Machine (restore) <code>vss-cli compute vm res {id}</code>	
Fetch VM restore Points	<code>vss-cli compute vm get {id} restore-point ls -f timestamp=2023-11</code>
Submit Restore	<code>vss-cli compute vm res {id} --timestamp=2023-11-11 --reason 'OS corrupted'</code>

Virtual Machine Inventory

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

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

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

Configure: vss-cli

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

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 decommission

vss-cli compute vm rm --show-info {id} ... {idN}
--