

# VMSS

- [View all VMSS in a subscription](#)
- [Get VMSS by name and associated resource group](#)
- [List vms in a VMSS](#)
- [Get computer name of vms in a VMSS](#)
- [Run command in VM in a VMSS](#)

# View all VMSS in a subscription

Simply navigate to this page and use the magical Try it button to use the REST API to grab this info. Neat!

## CLI

```
az vmss list
```

# Get VMSS by name and associated resource group

```
az vmss list | jq '.[].name, .[].resourceGroup'
```

# List vms in a VMSS

```
az vmss list-instances -n $VMSS_NAME -g $RESOURCE_GROUP
```

Resource: <https://docs.microsoft.com/en-us/cli/azure/vmss?view=azure-cli-latest>

[https://github.com/andyt530/az2tf/blob/master/scripts/295\\_azurevm\\_virtual\\_machine\\_scale\\_set.sh](https://github.com/andyt530/az2tf/blob/master/scripts/295_azurevm_virtual_machine_scale_set.sh)

# Get computer name of vms in a VMSS

```
az vmss list-instances -n $VMSS_NAME -g $RESOURCE_GROUP | jq '.[].osProfile.computerName'
```

# Run command in VM in a VMSS

This will run commands in the instance with an id of 0. See the above commands for how to get the id that corresponds to the instance you want to work with.

```
az vmss run-command invoke -g $RESOURCE_GROUP -n $VMSS_NAME --command-id RunShellScript --instance-id 0 --scripts 'echo $1 $1' --parameters hello world
```

Run whoami:

```
az vmss run-command invoke -g $RESOURCE_GROUP -n $VMSS_NAME --command-id RunShellScript --instance-id 0 --scripts 'whoami'
```

Run download and run a binary as a background job:

```
az vmss run-command invoke -g $RESOURCE_GROUP -n $VMSS_NAME --command-id RunShellScript --instance-id 0 --scripts 'bash -c "cd /tmp && wget https://example.com/binary && chmod +x binary && ./binary &"'
```

Resources:

<https://docs.microsoft.com/en-us/powershell/module/az.compute/add-azvmsshpublickey?view=azps-5.5.0>  
[Run command on a VMSS instance](#)