Azure CLI Microsoft cloud CLI

START HERE

`az login`
Sign in via browser

`az find "<text>"`
Find a command

`az account show`
Which subscription?

`--query "<jmespath>" -o tsv`
Filter the output

`az group list -o table`
List resource groups

`az vm list -g <rg> -o table`
List VMs

`az aks get-credentials -g <rg> -n <name>`
Wire up kubectl

`az <group> --help`
Explore commands

Login user, service principal, identity → `token cache` → **Subscription** billing and access scope → `--resource-group` → **Resource group** container for resources → `az <group> <cmd>` → **Resource** vm, aks, storage, ... → `--query, -o` → **Your terminal** json, table, tsv

CONCEPTS

Shape: az <group> [<subgroup>] <command> --options. Add -h at any level. Check az account show before changing anything.

Resource group, location and name are the three arguments most commands need. Set defaults with az config set defaults.group=<rg> defaults.location=<region>.

LOGIN AND ACCOUNTS

`az login`
Sign in through the browser

`az login --use-device-code`
Sign in on a headless machine

`az account show`
Show the active subscription

`az account list -o table`
List subscriptions you can use

`az account set -s <name-or-id>`
Switch subscription

RESOURCE GROUPS

`az group list -o table`
List resource groups

RESOURCE GROUPS (cont.)

`az group create -n <rg> -l <region>`
Create a resource group

`az group delete -n <rg> --yes --no-wait`
Delete a group and all its contents
Removes every resource inside it. Double-check the name.

`az resource list -g <rg> -o table`
List everything in a group

VIRTUAL MACHINES

`az vm list -g <rg> -o table`
List VMs

`az vm create -g <rg> -n <name> --image Ubuntu2204 --generate-ssh-keys`
Create a Linux VM

`az vm show -g <rg> -n <name> -d`
Show a VM with IPs and power state

`az vm start -g <rg> -n <name>`
Start a VM

`az vm deallocate -g <rg> -n <name>`
Stop and release compute billing

STORAGE

Storage commands default to account keys. Add --account-name <acct> --auth-mode login to use your Entra ID login instead.

`az storage account list -o table`
List storage accounts

`az storage account create -n <acct> -g <rg> -l <region> --sku Standard_LRS`
Create a storage account

`az storage container create -n <container>`
Create a blob container

`az storage blob upload -f <file> -c <container> -n <blob>`
Upload a blob

`az storage blob list -c <container> -o table`
List blobs

AKS AND CONTAINERS

`az aks list -o table`
List AKS clusters

`az aks create -g <rg> -n <name> --node-count 2 --generate-ssh-keys`
Create a cluster

AKS AND CONTAINERS (cont.)

`az aks get-credentials -g <rg> -n <name>`
Merge cluster access into kubeconfig
Add --overwrite-existing to replace a stale entry; add --admin only if you need cluster-admin.

`az acr login -n <registry>`
Docker login to a registry

ROLES AND IDENTITY

`az role assignment list --assignee <user-or-id> --all -o table`
List a principal's roles

`az role assignment create --assignee <id> --role <role> --scope <scope>`
Grant a role
Scope is a subscription, group or resource ID, e.g. /subscriptions/<id>/resourceGroups/<rg>.

`az role assignment delete --assignee <id> --role <role> --scope <scope>`
Remove a role

`az ad sp create-for-rbac --name <name> --role Contributor --scopes <scope>`
Create a service principal
The secret is printed once; store it safely.

OUTPUT AND --QUERY

`-o table`
Human-readable table

`-o json / -o jsonc`
JSON (default) or coloured JSON

`-o tsv`
Plain values for scripts

`--query "[].name"`
Pick one field from every item

`--query "[].{Name:name, Loc:location}"`
Build a custom table

`--query "[?location=='westeurope'].name"`
Filter items

JMESPath quoting

Literal	Write it as
String	`'value'` (single quotes)
Number or false	`` `50` ``, `` `false` `` (backticks)
Whole query (bash)	`"..."` or `'...'`
Backticks in bash "..."	`escape them: \`50\``

In bash wrap the query in single quotes so backticks stay literal; in PowerShell double the backticks or escape them. Double quotes inside a filter give empty output.

OUTPUT AND --QUERY (cont.)

-o tsv strips JSON quotes, so VAR=$(az ... -o tsv) gives a clean value.

HELP AND EXTENSIONS

`az <group> -h`
List subgroups and commands

`az find "<text>"`
Search commands by keyword

`az upgrade`
Update the CLI

RECIPES

Switch subscription by name

```
az account list --query "[].name" -o tsv
az account set -s "<name>"
az account show -o table
```

Create a group and VM, then connect

```
az group create -n demo -l westeurope
az vm create -g demo -n vm1 --image Ubuntu2204 --generate-ssh-keys
IP=$(az vm show -g demo -n vm1 -d --query publicIps -o tsv)
ssh azureuser@$IP
```

Use an AKS cluster

```
az aks get-credentials -g <rg> -n <name>
kubectl config current-context
kubectl get nodes
```