C05: Azure 101

Saturday, December 23, 2023 4:48 PM

Difficulty: Level 2
Help Sparkle Redberry with some Azure command line skills. Find the elf and the terminal on Christmas Island.

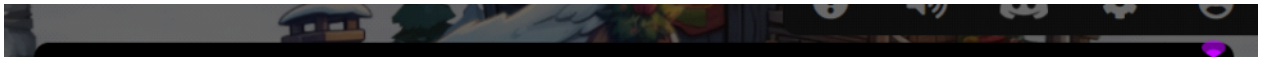
CONVERSATION w/ Elf Noel Boetie	**Sparkle Redberry (Rudolph's Rest Resort)** Hey, Sparkle Redberry here! So, I've been trying to learn about Azure and the Azure CLI and it's driving me nuts. Alabaster Snowball decided to use Azure to host some of his fancy new IT stuff on Geese Islands, and now us elves have to learn it too. Anyway, I know it's important and everyone says it's not as difficult as it seems, but honestly it still feels like quite a challenge for me. Alabaster sent us this Azure CLI reference as well. It's super handy, he said. Honestly, it just confuses me even more. If you can spare a moment, would you mind giving me a hand with this terminal? I'd be really grateful! Pretty please, with holly leaves on top! From <https://2023.holidayhackchallenge.com/badge?section=conversation&id=sparkleredberry> ** ------- Response after completing challenge ------- ** Wow, you did it! It makes quite a bit more sense to me now. Thank you so much! That Azure Function App URL you came across in the terminal looked interesting. It might be part of that new project Alabaster has been working on with the help of ChatNPT. Let me tell you, since he started using ChatNPT he's been introducing a lot of amazing innovation across the islands. Knowing Alabaster, he'll be delighted to tell you all about it! I think I last saw him on Pixel island. By the way, as part of the Azure documentation he sent the elves, Alabaster also noted that if Azure CLI tools aren't available in an Azure VM we should use the Azure REST API instead. I'm not really sure what that means, but I guess I know what I'll be studying up on next. From <https://2023.holidayhackchallenge.com/badge?section=conversation&id=sparkleredberry>
HINTS	**Azure CLI Reference** *From: Sparkle Redberry* *Terminal: Azure 101* The Azure CLI tools come with a builtin help system, but Microsoft also provides this handy cheatsheet. From <https://2023.holidayhackchallenge.com/badge?section=hint&id=hintAzure1011>

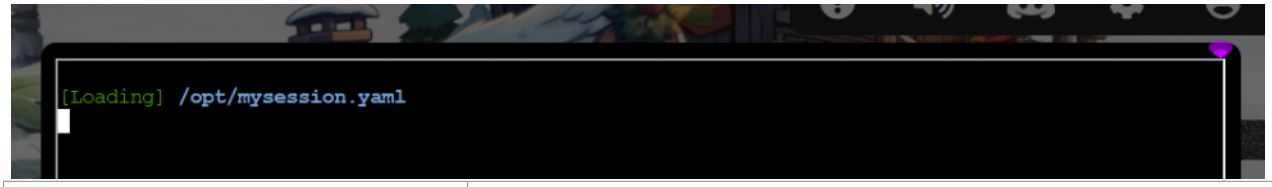
MY WORK AND ANSWER
I was not familiar with Azure groups so I relied on the hint provided as well as using Bing AI to explain the syntax. The terminal session timed out on me a few times as I did my sidebar exploration and made notes of items to remember for this challenge. Of note, the strong hint (in Discord from another player) of not messing with the yaml helped propel me along the path of a quicker solution.

I've flagged Azure as a future learning.

My question to Bing AI:	Bing AI response:

{
"environmentName": "AzureCloud",
"id": "2b0942f3-9bca-484b-a508-abdae2db5e64",
"isDefault": true,
"name": "northpole-sub",
"state": "Enabled",
"tenantId": "90a38eda-4006-4dd5-924c-6ca55cacc14d",
"user": {
"name": "northpole@northpole.invalid",
"type": "user"
}
}
(END)
	"environmentName": "AzureCloud": the Azure environment where the subscription is located.
	"id": "2b0942f3-9bca-484b-a508-abdae2db5e64": the unique identifier (GUID) for subscription.
	"isDefault": true: This subscription is set as the default.
	"name": "northpole-sub": the name of the subscription.
	"state": "Enabled": This shows the current state of the subscription, which is enabled in this case.
	"tenantId": "90a38eda-4006-4dd5-924c-6ca55cacc14d": The GUID for Azure AD tenant that the subscription belongs to.
	"user": an object containing info about the user associated with the subscription. It includes:
	• "name": "northpole@northpole.invalid": The username of the associated user.
	• "type": "user": The type of the associated user.
	
	Please mythat the actual values (like the id, tenantId, and user.name) are likely placeholders and not real.





You may not know this but the Azure cli help messages are very easy to access. First, try typing: *$ az help | help*	$ az help | less

Next, you've already been configured with credentials. Use 'az' and your 'account' to 'show' your current details and make sure to pipe to less (| less)	elf@3d47610a4d66:~$ az account show --output table
	EnvironmentName IsDefault Name State TenantId
	---------- ---------- ---------- ------- ----------
	AzureCloud True northpole-sub Enabled 90a38eda-4006-4dd5-924c-6ca55cacc14d
	
	elf@3d47610a4d66:~$
	
	elf@716c1075b85d:~$ az account show
	{
	"environmentName": "AzureCloud",
	"id": "2b0942f3-9bca-484b-a508-abdae2db5e64",
	"isDefault": true,
	"name": "northpole-sub",
	"state": "Enabled",
	"tenantId": "90a38eda-4006-4dd5-924c-6ca55cacc14d",
	"user": {
	"name": "northpole@northpole.invalid",
	"type": "user"
	}
	}

Excellent! Now get a list of resource groups in Azure.
 	
There are 2 resource groups in Azure
	elf@716c1075b85d:~$ az group list
	[
	{
	"id": "/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg1",
	"location": "eastus",
	"managedBy": null,
	"name": "northpole-rg1",
	"properties": {
	"provisioningState": "Succeeded"
	},
	"tags": {}
	},
	{
	"id": "/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg2",
	"location": "westus",
	"managedBy": null,
	"name": "northpole-rg2",
	"properties": {
	"provisioningState": "Succeeded"
	},
	"tags": {}
	}
	]

Ok, now use one of the resource groups to get a list of function apps. *Note: Some of the information returned from this command relates to other cloud assets used by Santa and his elves.* *(not* az functionapp show --name northpole-rg1) NOTE: The information here will be relevant to the Certificate Sshenanigans challenge	elf@716c1075b85d:~$ az functionapp list --resource-group northpole-rg1 [{ "appServicePlanId": "/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg1/providers/Microsoft.Web/serverfarms/EastUSLinuxDynamicPlan", "availabilityState": "Normal", "clientAffinityEnabled": false, "clientCertEnabled": false, "clientCertExclusionPaths": null, "clientCertMode": "Required", "cloningInfo": null, "containerSize": 0, "customDomainVerificationId": "201F74B099FA881DB9368A26C8E8B8BB8B9AF75BF450AF717502AC151F59DBEA", "dailyMemoryTimeQuota": 0,

```
  "defaultHostName": "northpole-ssh-certs-fa.azurewebsites.net",
  "enabled": true,
  "enabledHostNames": [
   "northpole-ssh-certs-fa.azurewebsites.net"
  ],
  "extendedLocation": null,
  "hostNameSslStates": [
   {
    "certificateResourceId": null,
    "hostType": "Standard",
    "ipBasedSslResult": null,
    "ipBasedSslState": "NotConfigured",
    "name": "northpole-ssh-certs-fa.azurewebsites.net",
     "sslState": "Disabled",
    "thumbprint": null,
    "toUpdate": null,
    "toUpdateIpBasedSsl": null,
    "virtualIPv6": null,
    "virtualIp": null
   },
   {
    "certificateResourceId": null,
    "hostType": "Repository",
    "ipBasedSslResult": null,
    "ipBasedSslState": "NotConfigured",
    "name": "northpole-ssh-certs-fa.scm.azurewebsites.net",
    "sslState": "Disabled",
    "thumbprint": null,
    "toUpdate": null,
    "toUpdateIpBasedSsl": null,
    "virtualIPv6": null,
    "virtualIp": null
   }
  ],
  "hostNames": [
   "northpole-ssh-certs-fa.azurewebsites.net"
  ],
  "hostNamesDisabled": false,
  "hostingEnvironmentProfile": null,
  "httpsOnly": false,
  "hyperV": false,
  "id": "/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-rg1/pro
viders/Microsoft.Web/sites/northpole-ssh-certs-fa",
  "identity": {
   "principalId": "d3be48a8-0702-407c-89af-0319780a2aea",
   "tenantId": "90a38eda-4006-4dd5-924c-6ca55cacc14d",
   "type": "SystemAssigned",
   "userAssignedIdentities": null
  },
  "inProgressOperationId": null,
  "isDefaultContainer": null,
  "isXenon": false,
  "keyVaultReferenceIdentity": "SystemAssigned",
  "kind": "functionapp,linux",
  "lastModifiedTimeUtc": "2023-11-09T14:43:01.183333",
  "location": "East US",
  "maxNumberOfWorkers": null,
  "name": "northpole-ssh-certs-fa",
  "outboundIpAddresses": "",
  "possibleOutboundIpAddresses": "",
  "publicNetworkAccess": null,
  "redundancyMode": "None",
  "repositorySiteName": "northpole-ssh-certs-fa",
  "reserved": true,
  "resourceGroup": "northpole-rg1",
  "scmSiteAlsoStopped": false,
  "siteConfig": {
   "acrUseManagedIdentityCreds": false,
   "acrUserManagedIdentityId": null,
   "alwaysOn": false,
   "antivirusScanEnabled": null,
```

```
"apiDefinition": null,
"apiManagementConfig": null,
"appCommandLine": null,
"appSettings": null,
"autoHealEnabled": null,
"autoHealRules": null,
"autoSwapSlotName": null,
"azureMonitorLogCategories": null,
"azureStorageAccounts": null,
"connectionStrings": null,
"cors": null,
"customAppPoolIdentityAdminState": null,
"customAppPoolIdentityTenantState": null,
"defaultDocuments": null,
"detailedErrorLoggingEnabled": null,
"documentRoot": null,
"elasticWebAppScaleLimit": null,
"experiments": null,
"fileChangeAuditEnabled": null,
"ftpsState": null,
"functionAppScaleLimit": 200,
"functionsRuntimeScaleMonitoringEnabled": null,
"handlerMappings": null,
"healthCheckPath": null,
"http20Enabled": true,
"http20ProxyFlag": null,
"httpLoggingEnabled": null,
"ipSecurityRestrictions": null,
"ipSecurityRestrictionsDefaultAction": null,
"javaContainer": null,
"javaContainerVersion": null,
"javaVersion": null,
"keyVaultReferenceIdentity": null,
"limits": null,
"linuxFxVersion": "Python|3.11",
"loadBalancing": null,
"localMySqlEnabled": null,
"logsDirectorySizeLimit": null,
"machineKey": null,
"managedPipelineMode": null,
"managedServiceIdentityId": null,
"metadata": null,
"minTlsCipherSuite": null,
"minTlsVersion": null,
"minimumElasticInstanceCount": 0,
"netFrameworkVersion": null,
"nodeVersion": null,
"numberOfWorkers": 1,
"phpVersion": null,
"powerShellVersion": null,
"preWarmedInstanceCount": null,
"publicNetworkAccess": null,
"publishingPassword": null,
"publishingUsername": null,
"push": null,
"pythonVersion": null,
"remoteDebuggingEnabled": null,
"remoteDebuggingVersion": null,
"requestTracingEnabled": null,
"requestTracingExpirationTime": null,
"routingRules": null,
"runtimeADUser": null,
"runtimeADUserPassword": null,
"scmIpSecurityRestrictions": null,
"scmIpSecurityRestrictionsDefaultAction": null,
"scmIpSecurityRestrictionsUseMain": null,
"scmMinTlsVersion": null,
"scmType": null,
"sitePort": null,
"sitePrivateLinkHostEnabled": null,
"storageType": null,
```

```
    "supportedTlsCipherSuites": null,
    "tracingOptions": null,
    "use32BitWorkerProcess": null,
    "virtualApplications": null,
    "vnetName": null,
    "vnetPrivatePortsCount": null,
    "vnetRouteAllEnabled": null,
    "webSocketsEnabled": null,
    "websiteTimeZone": null,
    "winAuthAdminState": null,
    "winAuthTenantState": null,
    "windowsConfiguredStacks": null,
    "windowsFxVersion": null,
    "xManagedServiceIdentityId": null
   },
   "slotSwapStatus": null,
   "state": "Running",
   "storageAccountRequired": false,
   "suspendedTill": null,
   "tags": {
    "create-cert-func-url-path": "/api/create-cert?code=candy-cane-twirl",
    "project": "northpole-ssh-certs"
   },
   "targetSwapSlot": null,
   "trafficManagerHostNames": null,
   "type": "Microsoft.Web/sites",
   "usageState": "Normal",
   "virtualNetworkSubnetId": null,
   "vnetContentShareEnabled": false,
   "vnetImagePullEnabled": false,
  }
]

elf@4b2ab6edff91:~$ az functionapp list --resource-group northpole-rg2
[]
```


Find a way to list the only VM in one of the resource groups you have access to.


```
elf@4b2ab6edff91:~$ az vm list --resource-group northpole-rg2
[
 {
  "id": "/subscriptions/2b0942f3-9bca-484b-a508-abdae2db5e64/resourceGroups/northpole-
rg2/providers/Microsoft.Compute/virtualMachines/NP-VM1",
  "location": "eastus",
  "name": "NP-VM1",
  "properties": {
   "hardwareProfile": {
    "vmSize": "Standard_D2s_v3"
   },
   "provisioningState": "Succeeded",
   "storageProfile": {
    "imageReference": {
     "offer": "UbuntuServer",
     "publisher": "Canonical",
     "sku": "16.04-LTS",
     "version": "latest"
    },
    "osDisk": {
     "caching": "ReadWrite",
     "createOption": "FromImage",
     "managedDisk": {
      "storageAccountType": "Standard_LRS"
     },
     "name": "VM1_OsDisk_1"
    }
   },
   "vmId": "e5f16214-18be-4a31-9ebb-2be3a55cfcf7"
  },
  "resourceGroup": "northpole-rg2",
  "tags": {}
 }
]
```

Find a way to invoke a run-command against the only Virtual Machine (VM) so you can RunShellScript and get a directory listing to reveal a file on the Azure VM."

For more info:
https://learn.microsoft.com/en-us/cli/azure/vm/run-command?view=azure-cli-latest#az-vm-run-command-invoke

Explanation of command and output:
This Azure CLI command is using the az vm run-command invoke command to execute a shell script on a virtual machine (VM) named NP-VM1 in the resource group northpole-rg2. The shell script being run is ls -l /home/, which lists the contents of the /home/ directory in a long format on the VM.

The output of the command is in JSON format and contains two main parts:

1. **StdOut**: This is the standard output from the command that was run on the VM. The message "total 8\ndrwxr-x--- 1 0 0 4096 Dec 2 22:16 adminuser \n" indicates that there is a directory named adminuser in the /home/ directory of the VM. The directory was last modified on Dec 2 at 22:16. The drwxr-x--- part represents the permissions of the directory, indicating that the owner has read, write, and execute permissions, while the group has only read and execute permissions.
2. **StdErr**: This is the standard error output from the command that was run on the VM. In this case, the message is empty, which means no error occurred during the execution of the command.

The "Provisioning succeeded" status for both StdOut and StdErr indicates that the command was successfully executed on the VM.

```
elf@2ed8ff539bbd:~$ az vm run-command invoke --resource-group northpole-rg2 --name NP-VM1 --command-id RunShellScript --scripts "ls -l /home/"
{
 "value": [
  {
   "code": "ComponentStatus/StdOut/succeeded",
   "displayStatus": "Provisioning succeeded",
   "level": "Info",
   "message": "total 8\ndrwxr-x--- 1 0 0 4096 Dec  2 22:16 adminuser\n",
   "time": 1703496318
  },
  {
   "code": "ComponentStatus/StdErr/succeeded",
   "displayStatus": "Provisioning succeeded",
   "level": "Info",
   "message": "",
   "time": 1703496318
  }
 ]
}


elf@2e46b821181b:~$ az vm run-command invoke --resource-group northpole-rg2 --name NP-VM1 --command-id RunShellScript --scripts "ls -l /home/ > /home/lady-cee-results.json"
{
 "value": [
  {
   "code": "ComponentStatus/StdOut/succeeded",
   "displayStatus": "Provisioning succeeded",
   "level": "Info",
   "message": "",
   "time": 1703499803
  },
  {
   "code": "ComponentStatus/StdErr/succeeded",
   "displayStatus": "Provisioning succeeded",
   "level": "Info",
   "message": "",
   "time": 1703499803
  }
 ]
}

elf@2e46b821181b:~$ cat /home/lady0cee-results.json
cat: /home/lady0cee-results.json: No such file or directory

elf@2e46b821181b:~$ az vm run-command invoke --resource-group northpole-rg2 --name NP-VM1 --command-id RunShellScript --scripts "ls -la"
{
 "value": [
  {
   "code": "ComponentStatus/StdOut/succeeded",
   "displayStatus": "Provisioning succeeded",
   "level": "Info",
   "message": "total 48\ndrwxr-x--- 1 0 0 4096 Dec  4 20:38 .\ndrwxr-x--- 1 0 0 4096 Dec  4 20:38 ..\ndrwxr-x--- 1 0 0 4096 Dec  4 20:37 bin\ndrwxr-xr-x 1 0 0 4096 Dec  4 20:38 etc\ndrwxr-x--- 1 0 0 4096 Dec 25 10:23 home\n-rwxr-x--- 1 0 0    0 Dec  4 20:37 jinglebells\ndrwxr-xr-x 1 0 0 4096 Dec  4 20:37 lib\ndrwxr-xr-x 1 0 0 4096 Dec  4 20:37 lib64\ndrwxr-xr-x 1 0 0 4096 Dec  4 20:37 usr\n",
   "time": 1703500515
  },
  {
   "code": "ComponentStatus/StdErr/succeeded",
   "displayStatus": "Provisioning succeeded",
   "level": "Info",
   "message": "",
   "time": 1703500515
  }
 ]
}

Great, you did it all!
```

Comments of Interest from other Players

Azure uses resource groups. Permissions can apply globally but also at the resource group level. So what you may not be able to do at the global level, you might be able to do at a resource group level by specifying a --resource-group or -g
permissions can be at various levels in azure.. subscription, resource group, workspace, etc.
permanent or eligible (PIM)

Something along this lines should do it (just replace values appropriately) ||az vm run-command invoke -g someresourcegroup --name somevmname --command-id RunShellScript --scripts "echo helloworld"|

NOTE: Every time you exit, it restarts from the beginning

Why the CLI interface not allowed to scroll up, then how should I examine what is the VM name of that particular resource group? ||az vm list --resource-group <YourResourceGroupName> --query "[].name" -o tsv||

vm run-command. I try az vm run-command invoke -g northpole-rg2 --name VM1_OsDisk_1 --command-id RunShellScript --scripts "echo helloworld"