

FIAP GRADUAÇÃO

TECNOLOGIA EM DESENVOLVIMENTO DE SISTEMAS

DevOps Tools & Cloud Computing

Utilizando a CLI do Azure

PROF. Fábio Machado

proffabio.machado@fiap.com.br

- Conceito da CLI
- Instalando a CLI do Azure em seu ambiente de trabalho
- Utilizando em Nuvem o Azure Cloud Shell
- Por linha de comando vamos criar uma VM
- Criar scripts de automação

- Existem dois tipos de ferramentas de gerenciamento:

Ferramentas Visuais e Ferramentas Baseadas em Código

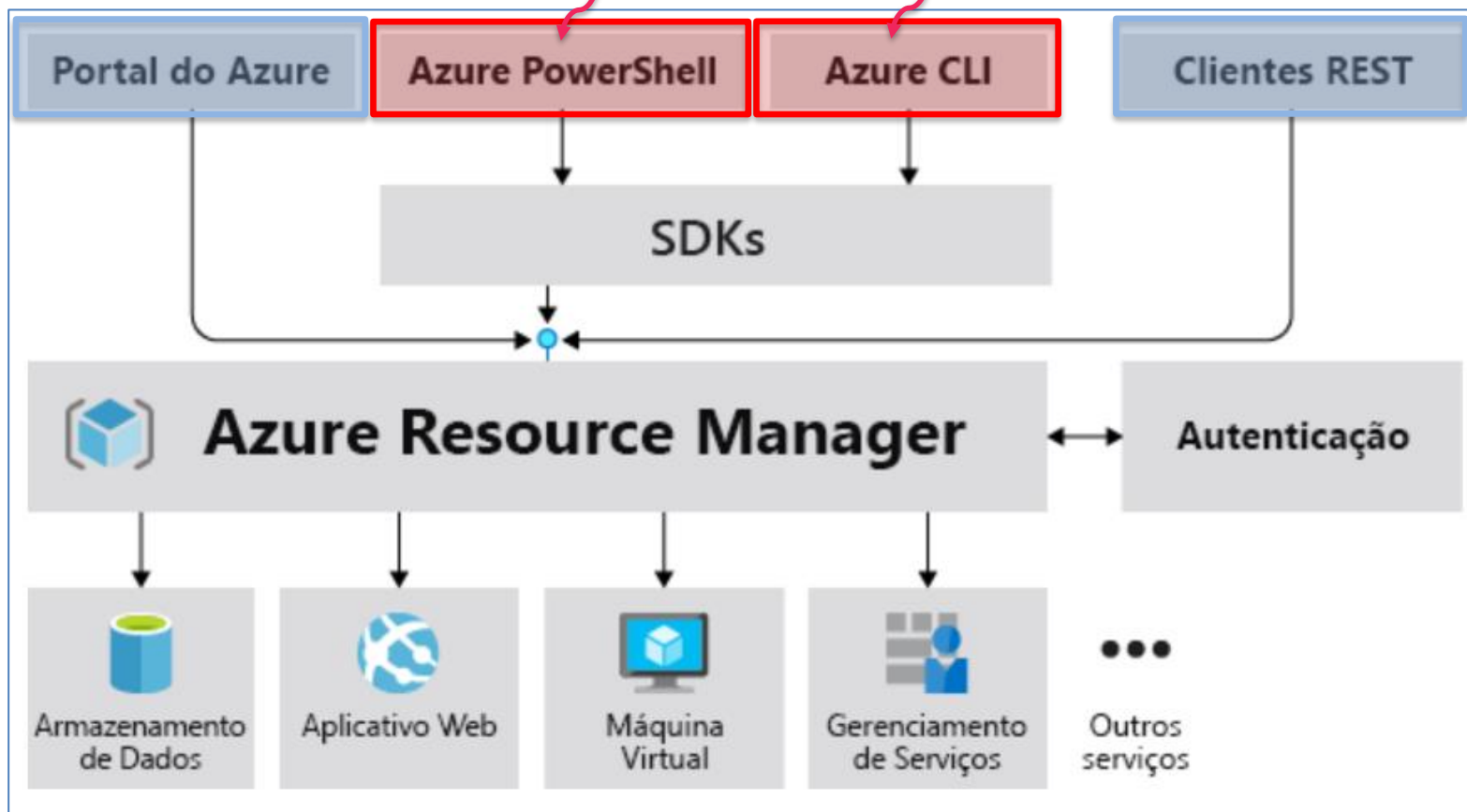
- As Ferramentas Visuais fornecem acesso total a todas as funcionalidades do Azure de uma maneira visual e amigável
- No entanto, as Ferramentas Visuais podem ser menos úteis ao tentar configurar uma grande implantação de recursos com interdependências e opções de configuração
- Para instalar e configurar rapidamente os recursos do Azure, uma Ferramenta Baseada em Código geralmente é a melhor escolha

- Embora possa levar algum tempo para entender os comandos e parâmetros corretos no início, uma vez digitados, eles podem ser salvos em arquivos e usados repetidamente conforme necessário
- Além disso, o código que executa a instalação e configuração pode ser armazenado em um controle de versão, e mantido junto com o código fonte do aplicativo em uma ferramenta como o Git por exemplo
- Essa abordagem para gerenciar recursos de hardware usando a mesma mentalidade que os desenvolvedores usam quando escrevem o código do aplicativo é conhecida como Infraestrutura como código (IaC)

- A CLI (Command Line Interface) do Azure corresponde à Interface de Linha de Comando do Azure, ou seja, é um conjunto de comandos usados para criar e gerenciar recursos do Azure
- A CLI do Azure está disponível nos serviços do Azure e foi criada para ajudar você a trabalhar de maneira rápida, com ênfase na automação
- Está disponível para instalação em ambientes Windows, macOS e Linux. Ela também pode ser executada em um Container do Docker e no Azure Cloud Shell
- Com a CLI do Azure você pode criar, gerenciar e excluir recursos que estão no Microsoft Azure, sendo possível também automatizar essas atividades usando-a como script de linha de comando

Utilizamos até agora o Portal

Vamos acessar por linha de comando agora



Futuramente vamos utilizar plug-ins e extensões em SDKs

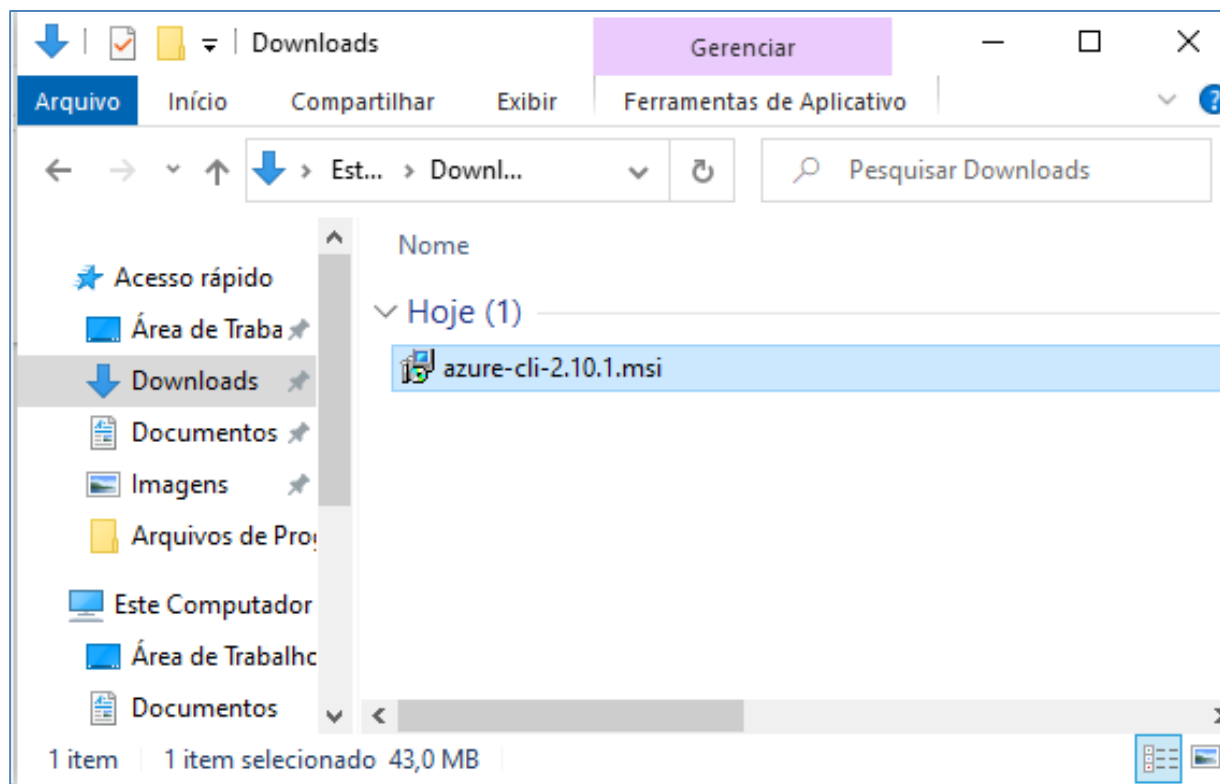
INSTALAR CLI DO AZURE



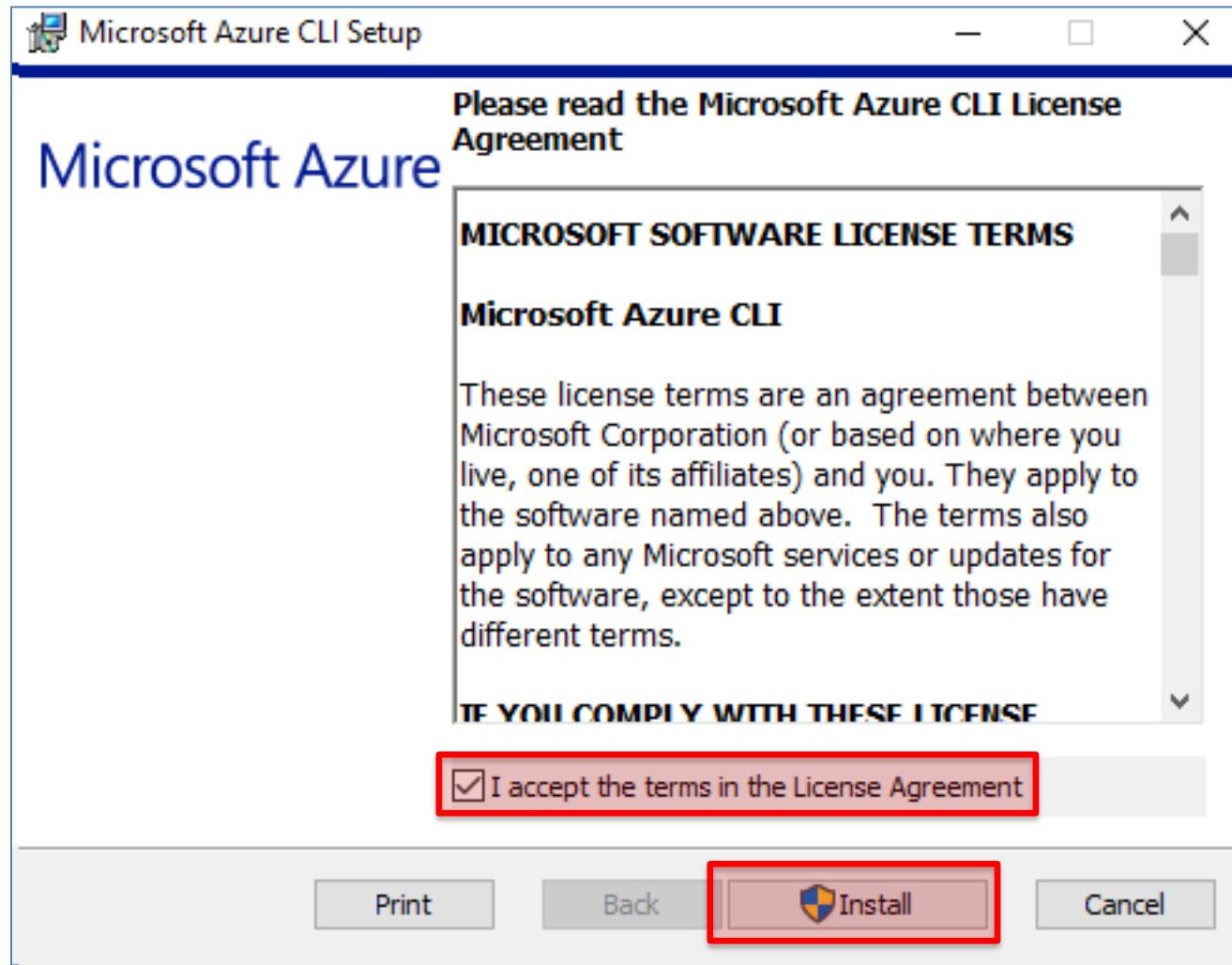
Windows

Baixar o instalador do link abaixo

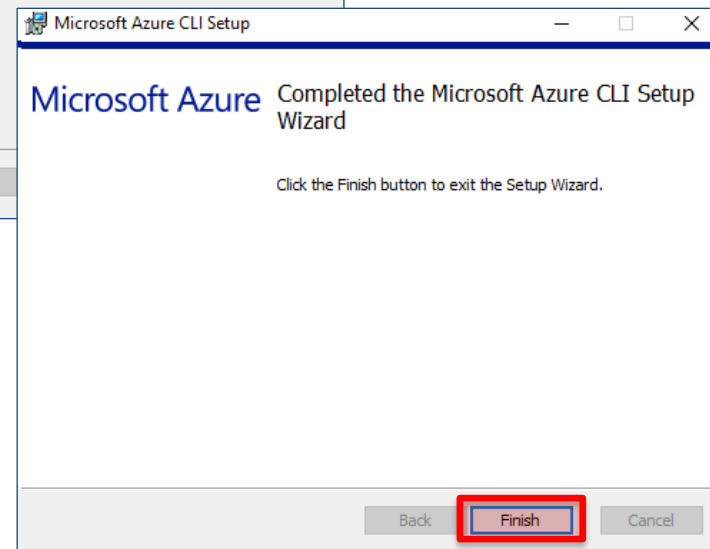
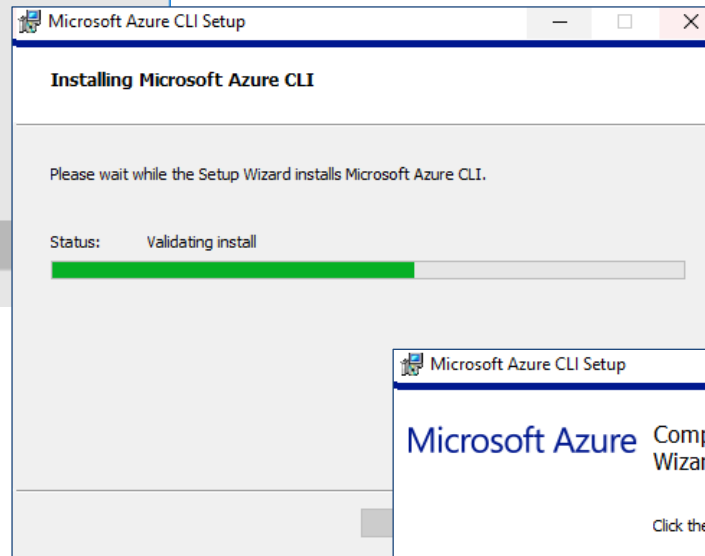
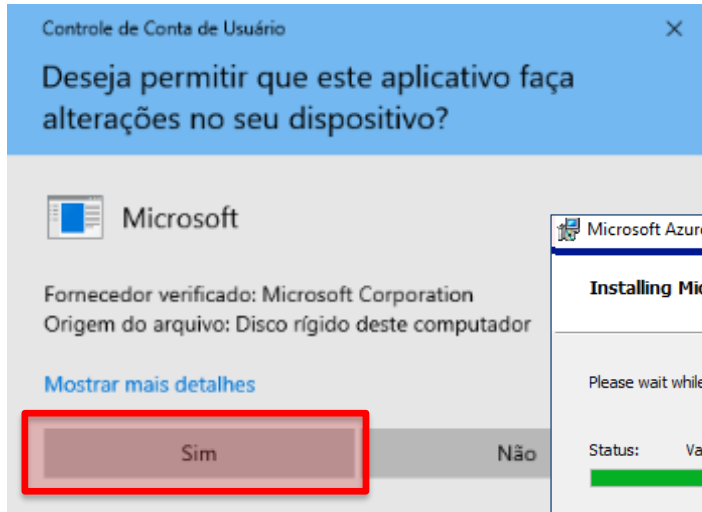
<https://aka.ms/installazurecliwindows>



Windows



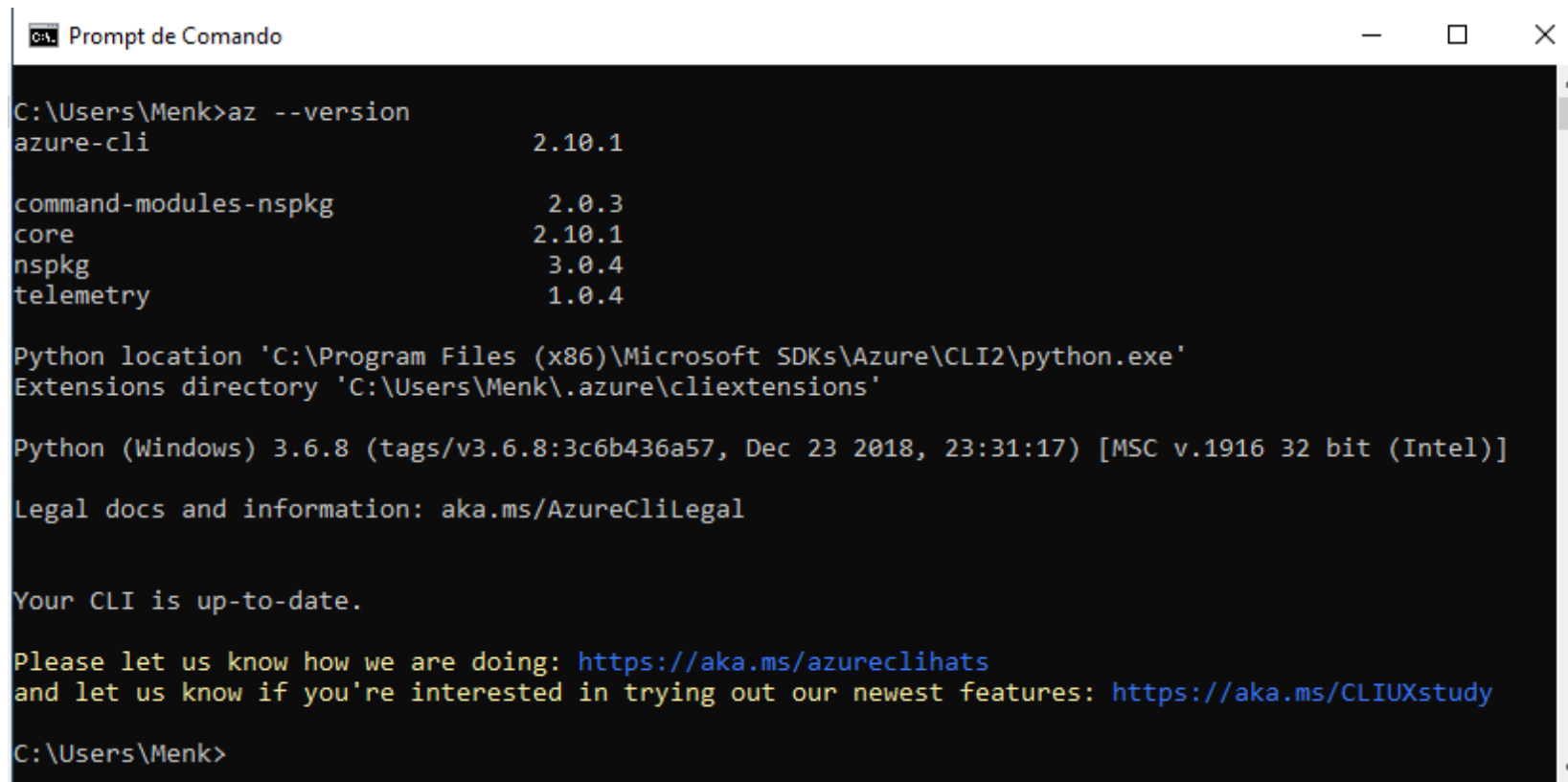
Windows



Windows

Verifique a instalação com o comando abaixo em uma janela do **Prompt de Comando** do Windows

az --version



```
C:\Users\Menk>az --version
azure-cli                        2.10.1

command-modules-nspkg           2.0.3
core                            2.10.1
nspkg                           3.0.4
telemetry                       1.0.4

Python location 'C:\Program Files (x86)\Microsoft SDKs\Azure\CLI2\python.exe'
Extensions directory 'C:\Users\Menk\.azure\cliextensions'

Python (Windows) 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 23 2018, 23:31:17) [MSC v.1916 32 bit (Intel)]

Legal docs and information: aka.ms/AzureCliLegal

Your CLI is up-to-date.

Please let us know how we are doing: https://aka.ms/azureclihats
and let us know if you're interested in trying out our newest features: https://aka.ms/CLIUXstudy

C:\Users\Menk>
```

Para ambientes MAC: Instalar a CLI do Azure

Mac



Instale o **Homebrew** executando o comando abaixo no **Terminal** do Mac

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

```
Menk — bash -c #!/bin/bash\012set -u\012\012# First check if the OS is Linux.\012if [[ "$(uname)" = "Linux" ]]; then\012...
iMac:~ Menk$
iMac:~ Menk$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
Password:
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew
==> The following existing directories will be made group writable:
/usr/local/bin
/usr/local/include
/usr/local/lib
/usr/local/sbin
/usr/local/lib/pkgconfig
==> The following existing directories will have their owner set to Menk:
/usr/local/bin
/usr/local/include
/usr/local/lib
/usr/local/sbin
/usr/local/lib/pkgconfig
==> The following existing directories will have their group set to admin:
/usr/local/bin
/usr/local/include
```

Para ambientes MAC: Instalar a CLI do Azure

Mac



Pressione **Enter** para continuar a instalação

```
Menk — bash -c #!/bin/bash\012set -u\012\012# First check if the OS is Linux.\012if [[ "$(uname)" = "Linux" ]]; then\012...
/usr/local/sbin
/usr/local/lib/pkgconfig
==> The following existing directories will have their group set to admin:
/usr/local/bin
/usr/local/include
/usr/local/lib
/usr/local/sbin
/usr/local/lib/pkgconfig
==> The following new directories will be created:
/usr/local/etc
/usr/local/share
/usr/local/var
/usr/local/opt
/usr/local/share/zsh
/usr/local/share/zsh/site-functions
/usr/local/var/homebrew
/usr/local/var/homebrew/linked
/usr/local/Cellar
/usr/local/Caskroom
/usr/local/Homebrew
/usr/local/Frameworks
==> The Xcode Command Line Tools will be installed.
Press RETURN to continue or any other key to abort
█
```

Para ambientes MAC: Instalar a CLI do Azure

Mac



Aguarde até o final da instalação

```
Menk — softwareupdate • bash -c #!/bin/bash\012set -u\012\012# First check if the OS is Linux.\012if [[ "$(uname)" = "...
==> /usr/bin/sudo /usr/bin/chgrp admin /usr/local/bin /usr/local/include /usr/local/lib /usr/local/sbin /usr/local
/lib/pkgconfig
==> /usr/bin/sudo /bin/mkdir -p /usr/local/etc /usr/local/share /usr/local/var /usr/local/opt /usr/local/share/zsh
/usr/local/share/zsh/site-functions /usr/local/var/homebrew /usr/local/var/homebrew/linked /usr/local/Cellar /usr
/local/Caskroom /usr/local/Homebrew /usr/local/Frameworks
==> /usr/bin/sudo /bin/chmod g+rx /usr/local/etc /usr/local/share /usr/local/var /usr/local/opt /usr/local/share/
zsh /usr/local/share/zsh/site-functions /usr/local/var/homebrew /usr/local/var/homebrew/linked /usr/local/Cellar /
usr/local/Caskroom /usr/local/Homebrew /usr/local/Frameworks
==> /usr/bin/sudo /usr/sbin/chown Menk /usr/local/etc /usr/local/share /usr/local/var /usr/local/opt /usr/local/sh
are/zsh /usr/local/share/zsh/site-functions /usr
Menk — -bash — 81x25
ar /usr/local/Caskroom /usr/local/Homebrew /usr/remote: Counting objects: 100% (65/65), done.
==> /usr/bin/sudo /usr/bin/chgrp admin /usr/localremote: Compressing objects: 100% (45/45), done.
are/zsh /usr/local/share/zsh/site-functions /usrremote: Total 777163 (delta 37), reused 40 (delta 20), pack-reused 777098
ar /usr/local/Caskroom /usr/local/Homebrew /usr/remote: Receiving objects: 100% (777163/777163), 310.29 MiB | 4.10 MiB/s, done.
==> /usr/bin/sudo /bin/mkdir -p /Users/Menk/LibraryReceiving objects: 100% (517870/517870), done.
==> /usr/bin/sudo /usr/sbin/chown Menk /Users/MenUpdating files: 100% (5424/5424), done.
==> Searching online for the Command Line ToolsTapped 2 commands and 5170 formulae (5,449 files, 340.3MB).
==> /usr/bin/sudo /usr/bin/touch /tmp/.com.appleAlready up-to-date.
==> Installing Command Line Tools for Xcode-11.0!==> Installation successful!
==> /usr/bin/sudo /usr/sbin/softwareupdate -i CSoftware Update Tool

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics
No analytics data has been sent yet (or will be during this `install` run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
https://github.com/Homebrew/brew#donations

==> Next steps:
- Run `brew help` to get started
- Further documentation:
https://docs.brew.sh
iMac~ Menk$
iMac~ Menk$
iMac~ Menk$
```

A instalação demora um pouco

Para ambientes MAC: Instalar a CLI do Azure



Agora que o Homebrew está instalado, vamos instalar a CLI do Azure através desse gerenciador de pacotes. Execute o comando abaixo em um **Terminal** do Mac

brew update && brew install azure-cli

```
Menk — curl - ruby -W0 --disable=gems,did_you_mean,rubyopt /usr/local/Homebrew/Library/...
iMac:~ Menk$ brew update && brew install azure-cli
Already up-to-date.
==> Downloading https://homebrew.bintray.com/bottles/openssl%401.1-1.1.1g.catalina.bottle.tar
==> Downloading from https://d29vzk4ow87wi7.cloudfront.net/1926679569c6af5337de812d86f4dad2b2
##### 100.0%
==> Downloading https://homebrew.bintray.com/bottles/gdbm-1.18.1_1.catalina.bottle.tar.gz
==> Downloading from https://d29vzk4ow87wi7.cloudfront.net/f7b5ab7363961fa6defcb66b4ffdf53652
##### 100.0%
==> Downloading https://homebrew.bintray.com/bottles/readline-8.8.4.catalina.bottle.tar.gz
==> Downloading from https://d29vzk4ow87wi7.cloudfront.net/6a61c8e7c783f32bd22c6885caa4d838fe
##### 100.0%
==> Downloading https://homebrew.bintray.com/bottles/sqlite-3.32.3.catalina.bottle.tar.gz
==> Downloading from https://d29vzk4ow87wi7.cloudfront.net/98f798c4a62c9db46cc6ac82c499a0a1b8
##### 100.0%
==> Downloading https://homebrew.bintray.com/bottles/xz-5.2.5.catalina.bottle.tar.gz
==> Downloading from https://d29vzk4ow87wi7.cloudfront.net/2dcc8e0121c934d1e34fdb37fcd70f0f7
##### 100.0%
==> Downloading https://homebrew.bintray.com/bottles/python%403.8-3.8.5.catalina.bottle.tar.g
==> Downloading from https://d29vzk4ow87wi7.cloudfront.net/3f0592dfd6bf38c25870b57ba3e71b5b4
##### 100.0%
==> Downloading https://homebrew.bintray.com/bottles/azure-cli-2.18.1.catalina.bottle.tar.gz
==> Downloading from https://d29vzk4ow87wi7.cloudfront.net/a1d1f3fd3301f8fee3b9f1e67fedb29a1b
##### 50.7%
```


Para ambientes MAC: Instalar a CLI do Azure

Mac



Aguarde até o final da instalação

```
Menk — -bash — 93x25

If you need to have sqlite first in your PATH run:
  echo 'export PATH="/usr/local/opt/sqlite/bin:$PATH"' >> ~/.profile

For compilers to find sqlite you may need to set:
  export LDFLAGS="-L/usr/local/opt/sqlite/lib"
  export CPPFLAGS="-I/usr/local/opt/sqlite/include"

==> python@3.8
Python has been installed as
  /usr/local/bin/python3

Unversioned symlinks `python`, `python-config`, `pip` etc. pointing to
`python3`, `python3-config`, `pip3` etc., respectively, have been installed into
  /usr/local/opt/python@3.8/libexec/bin

You can install Python packages with
  pip3 install <package>
They will install into the site-package directory
  /usr/local/lib/python3.8/site-packages

See: https://docs.brew.sh/Homebrew-and-Python
==> azure-cli
Bash completion has been installed to:
  /usr/local/etc/bash_completion.d
iMac:~ Menk$
```

Para ambientes MAC: Instalar a CLI do Azure

Mac



Verifique a instalação com o comando abaixo em uma janela do
Terminal do Mac

az --version

```
Menk — -bash — 97x22
iMac:~ Menk$ az --version
azure-cli                        2.10.1

command-modules-nspkg            2.0.3
core                             2.10.1
nspkg                            3.0.4
telemetry                        1.0.4

Python location '/usr/local/Cellar/azure-cli/2.10.1/libexec/bin/python'
Extensions directory '/Users/Menk/.azure/cliextensions'

Python (Darwin) 3.8.5 (default, Jul 21 2020, 10:48:26)
[Clang 11.0.3 (clang-1103.0.32.62)]

Legal docs and information: aka.ms/AzureCliLegal

Your CLI is up-to-date.

Please let us know how we are doing: https://aka.ms/azureclihats
and let us know if you're interested in trying out our newest features: https://aka.ms/CLIUXstudy
iMac:~ Menk$
```

Para ambientes LINUX: Instalar a CLI do Azure

Linux



Debian / Ubuntu (Ubuntu 16.04+ e Debian 8+)

A Azure mantém um script que executa todos os comandos de instalação em uma única etapa. Executá-lo usando **curl** e redirecione diretamente para **bash**, ou baixe o script para um arquivo e inspecione-o antes da execução

```
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

Verifique a instalação com o comando abaixo em uma janela do
Terminal

```
az --version
```

Para ambientes LINUX: Instalar a CLI do Azure

Linux



RHEL / Fedora / CentOS / Oracle Linux

1. Importe a **chave** do repositório da Microsoft

`sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc`

```
[devops@devops ~]$ sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
[sudo] senha para devops:
[devops@devops ~]$ █
```

Para ambientes LINUX: Instalar a CLI do Azure

Linux



RHEL / Fedora / CentOS / Oracle Linux

2. Crie o repositório local azure-cli

```
echo -e "[azure-cli]
```

```
name=Azure CLI
```

```
baseurl=https://packages.microsoft.com/yumrepos/azure-cli
```

```
enabled=1
```

```
gpgcheck=1
```

```
gpgkey=https://packages.microsoft.com/keys/microsoft.asc" | sudo tee /etc/yum.repos.d/azure-cli.repo
```

```
oracle@oralinux76:~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
[oracle@oralinux76 ~]$ sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc  
[sudo] senha para oracle:  
[oracle@oralinux76 ~]$  
[oracle@oralinux76 ~]$  
[oracle@oralinux76 ~]$ echo -e "[azure-cli]  
>  
> name=Azure CLI  
>  
> baseurl=https://packages.microsoft.com/yumrepos/azure-cli  
>  
> enabled=1  
>  
> gpgcheck=1  
>  
> gpgkey=https://packages.microsoft.com/keys/microsoft.asc" | sudo tee /etc/yum.repos.d/azure-cl  
i.repo  
[azure-cli]  
  
name=Azure CLI  
  
baseurl=https://packages.microsoft.com/yumrepos/azure-cli  
  
enabled=1  
  
gpgcheck=1  
  
gpgkey=https://packages.microsoft.com/keys/microsoft.asc  
[oracle@oralinux76 ~]$
```

Para ambientes LINUX: Instalar a CLI do Azure

Linux



RHEL / Fedora / CentOS / Oracle Linux

3. Instale com o comando **yum install**

sudo yum install azure-cli

```
---> O pacote python3-setuptools.noarch 0:39.2.0-10.el7 será instalado  
--> Resolução de dependências finalizada
```

Dependências resolvidas

```
=====
Package                Arq.      Versão      Repo      Tam.
=====
Instalando:
azure-cli               x86_64    2.10.1
Instalando para as dependências:
python3                 x86_64    3.6.8
python3-libs            x86_64    3.6.8
python3-pip             noarch    9.0.3
python3-setuptools      noarch    39.2.0
Resumo da transação
=====
Instalar 1 Package (+4 Dependent packages)

Tamanho total do download: 40 M
Tamanho depois de instalado: 386 M
Is this ok [y/d/N]: █

transaction test succeeded
Running transaction
  Instalando : python3-libs-3.6.8-13.0.1.el7.x86_64                1/5
  Instalando : python3-setuptools-39.2.0-10.el7.noarch            2/5
  Instalando : python3-pip-9.0.3-7.el7_8.noarch                  3/5
  Instalando : python3-3.6.8-13.0.1.el7.x86_64                  4/5
  Instalando : azure-cli-2.10.1-1.el7.x86_64                     5/5
  Verifying   : python3-3.6.8-13.0.1.el7.x86_64                 1/5
  Verifying   : python3-libs-3.6.8-13.0.1.el7.x86_64            2/5
  Verifying   : python3-setuptools-39.2.0-10.el7.noarch          3/5
  Verifying   : azure-cli-2.10.1-1.el7.x86_64                   4/5
  Verifying   : python3-pip-9.0.3-7.el7_8.noarch                 5/5

Instalados:
  azure-cli.x86_64 0:2.10.1-1.el7

Dependência(s) instalada(s):
  python3.x86_64 0:3.6.8-13.0.1.el7      python3-libs.x86_64 0:3.6.8-13.0.1.el7
  python3-pip.noarch 0:9.0.3-7.el7_8     python3-setuptools.noarch 0:39.2.0-10.el7

Concluído!
[devops@devops ~]$ █
```

Para ambientes LINUX: Instalar a CLI do Azure

Linux



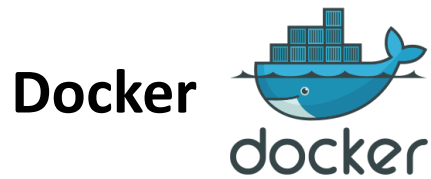
RHEL / Fedora / CentOS / Oracle Linux

Verifique a instalação com o comando abaixo em uma janela do Terminal

az --version

```
oracle@oralinux76:~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
[oracle@oralinux76 ~]$ az --version  
azure-cli                2.23.0  
  
core                     2.23.0  
telemetry                1.0.6  
  
Python location '/usr/bin/python3'  
Extensions directory '/home/oracle/.azure/cliextensions'  
  
Python (Linux) 3.6.8 (default, Mar  9 2021, 15:08:44)  
[GCC 4.8.5 20150623 (Red Hat 4.8.5-44.0.3)]  
  
Legal docs and information: aka.ms/AzureCliLegal  
  
Your CLI is up-to-date.  
  
Please let us know how we are doing: https://aka.ms/azureclihats  
and let us know if you're interested in trying out our newest features: https://aka.ms/CLIUXstudy  
[oracle@oralinux76 ~]$
```

Container



docker run --name azure-cli -h azure-cli -it -v <pathLocal>:<pathContainer> mcr.microsoft.com/azure-cli

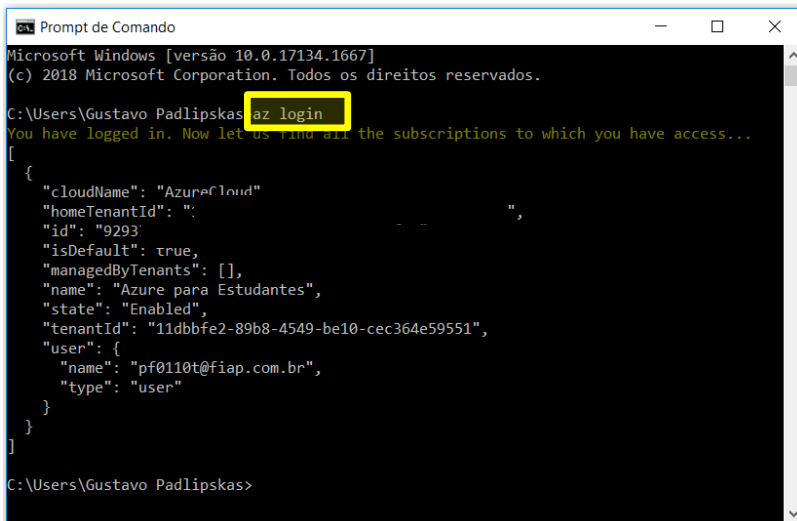
Exemplo:

docker run --name azure-cli -h azure-cli -it -v /Users/Menk/scripts:/scripts mcr.microsoft.com/azure-cli

```
Menk — com.docker.cli • docker run --name azure-cli -h azure-cli -it mcr.microsoft.com/azure-cli — 86x15
[iMac:~ Menk$ docker run --name azure-cli -h azure-cli -it mcr.microsoft.com/azure-cli ]
[azure-cli:/# ]
[azure-cli:/# cat /etc/os-release ]
NAME="Alpine Linux"
ID=alpine
VERSION_ID=3.19.1
PRETTY_NAME="Alpine Linux v3.19"
HOME_URL="https://alpinelinux.org/"
BUG_REPORT_URL="https://gitlab.alpinelinux.org/alpine/aports/-/issues"
azure-cli:/#
```


az login

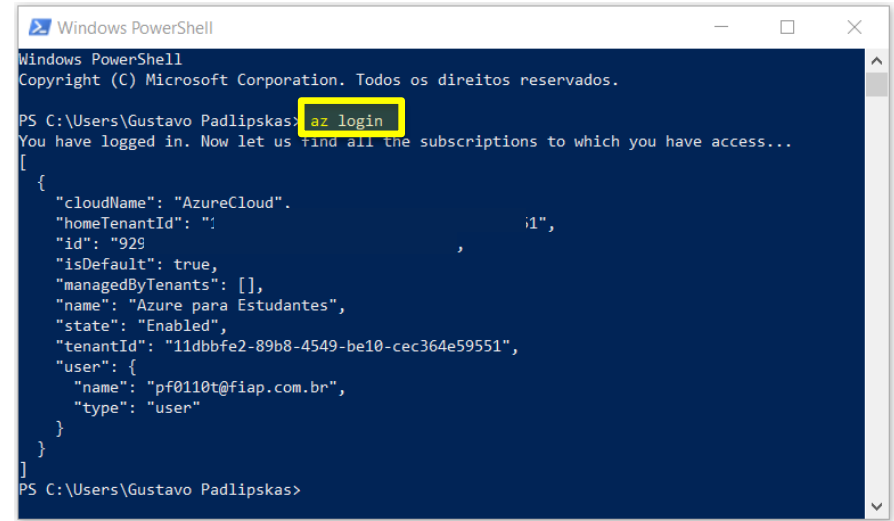
1 cmd do Windows



```
Prompt de Comando
Microsoft Windows [versão 10.0.17134.1667]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Gustavo Padlipskas>az login
You have logged in. Now let us find all the subscriptions to which you have access...
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": ":",
    "id": "9293",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure para Estudantes",
    "state": "Enabled",
    "tenantId": "11dbbfe2-89b8-4549-be10-cec364e59551",
    "user": {
      "name": "pf0110t@fiap.com.br",
      "type": "user"
    }
  }
]
```

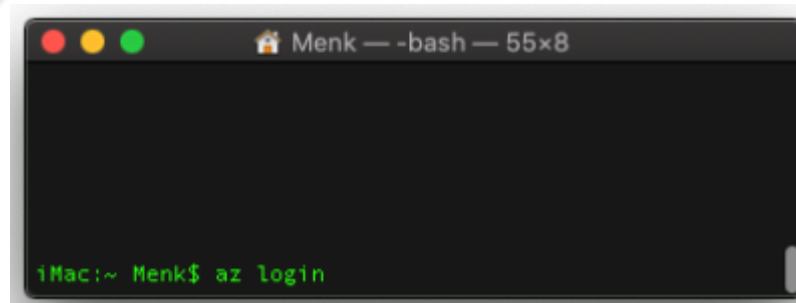
2 Windows PowerShell



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

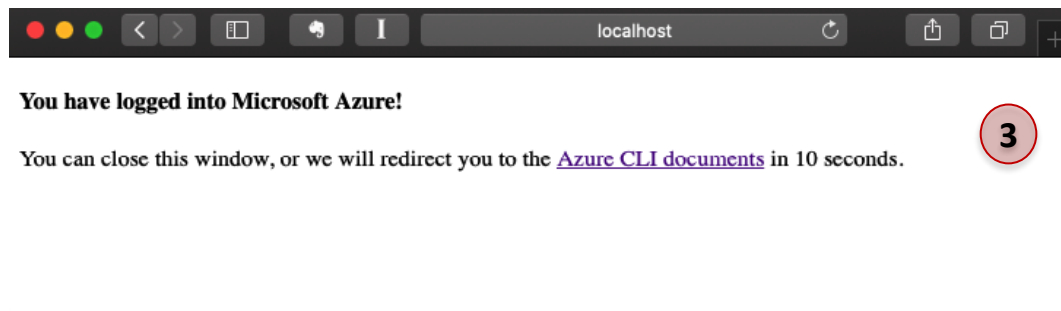
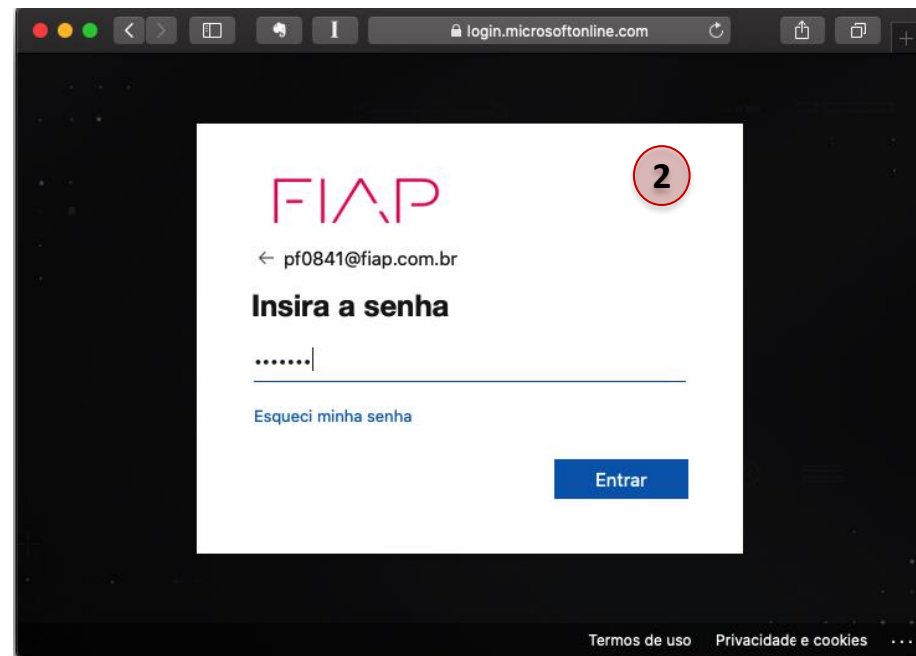
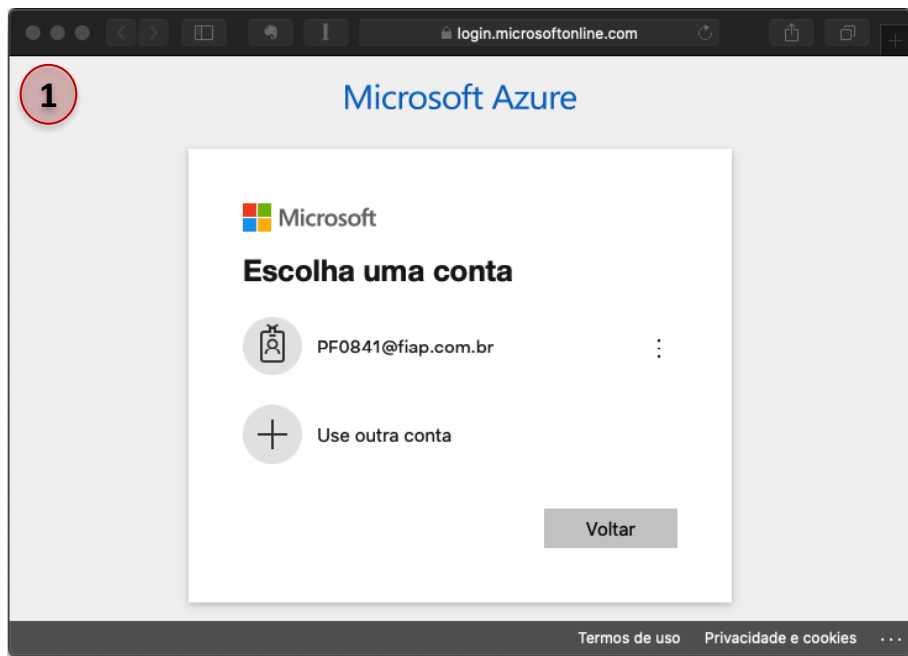
PS C:\Users\Gustavo Padlipskas>az login
You have logged in. Now let us find all the subscriptions to which you have access...
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": ":",
    "id": "9293",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure para Estudantes",
    "state": "Enabled",
    "tenantId": "11dbbfe2-89b8-4549-be10-cec364e59551",
    "user": {
      "name": "pf0110t@fiap.com.br",
      "type": "user"
    }
  }
]
```

3 Terminal Linux / Mac OS



```
Menk — -bash — 55x8

iMac:~ Menk$ az login
```

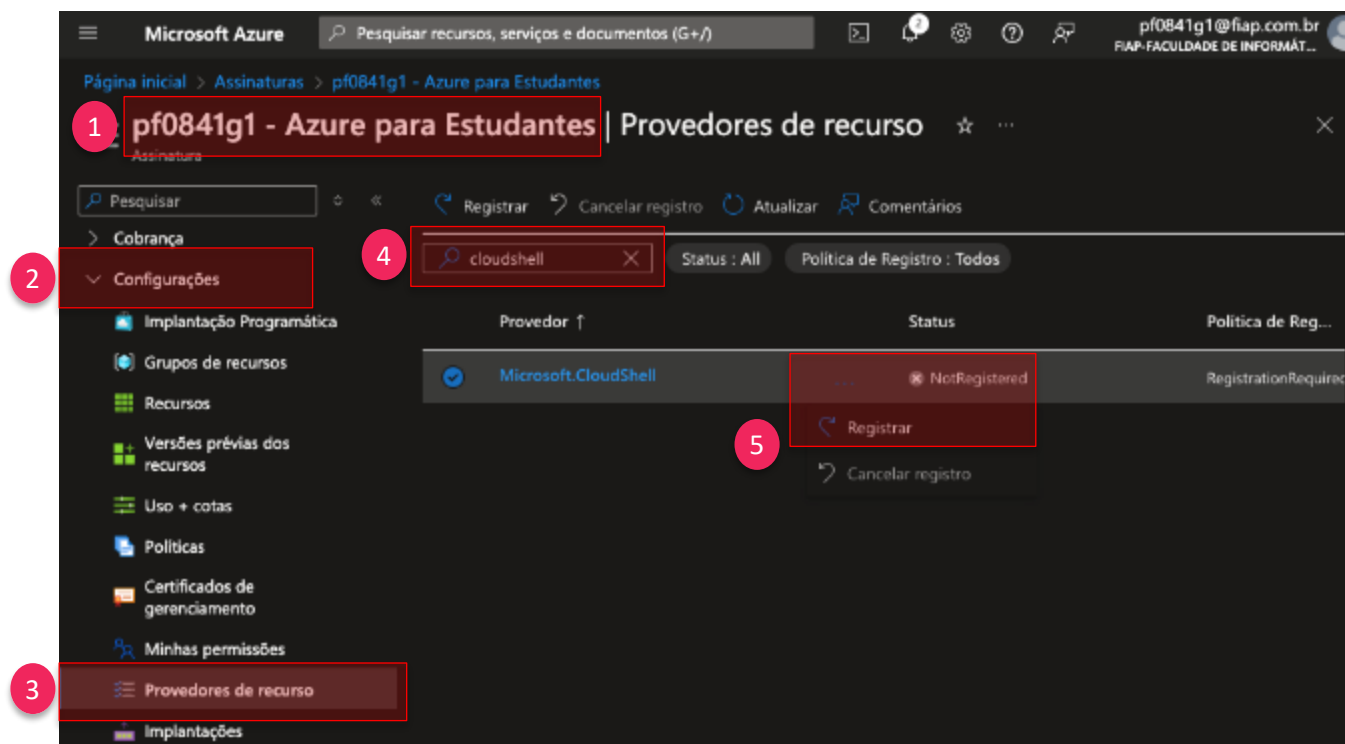




Para podermos unificar os comandos e scripts, vamos utilizar o **Azure Cloud Shell** que está disponível no Azure

Verifique se o registro da Cloud Shell está efetuado:

- 1 a 3) Entre em **Assinaturas -> Configurações -> Provedores de Recurso**
- 4) Procure por “cloudshell” no campos de pesquisa
- 5) Verifique o serviço



Para podermos unificar os comandos e scripts, vamos utilizar o **Azure Cloud Shell** que está disponível no Azure



Bem-vindo ao Azure Cloud Shell

Selecione Bash ou PowerShell. Você pode alterar shells a qualquer momento por meio do seletor de ambiente na barra de ferramentas do Cloud Shell. O ambiente usado mais recentemente será o padrão para a próxima sessão.

Bash

PowerShell

Guia de Introdução

Selecione uma assinatura para começar. Opcionalmente, você pode montar uma conta de armazenamento para persistir os arquivos entre sessões. [Saiba mais](#)

☐ Não é necessária uma conta de armazenamento ⓘ

☒ Montar conta de armazenamento ⓘ

ⓘ Todos os usuários na assinatura que têm permissões suficientes podem acessar esses recursos de armazenamento. [Saiba mais](#)

Assinatura da conta de armazenamento *

pf0841g1 - Azure para Estudantes

☐ Usar uma rede virtual privada existente [Saiba mais](#)

Aplicar

Anterior

Montar conta de armazenamento

O Azure Cloud Shell requer uma conta de armazenamento com compartilhamento de arquivos do Azure para persistir os arquivos. Selecione uma opção abaixo para montar uma conta de armazenamento. [Saiba mais](#)

☐ Selecionar a conta de armazenamento existente

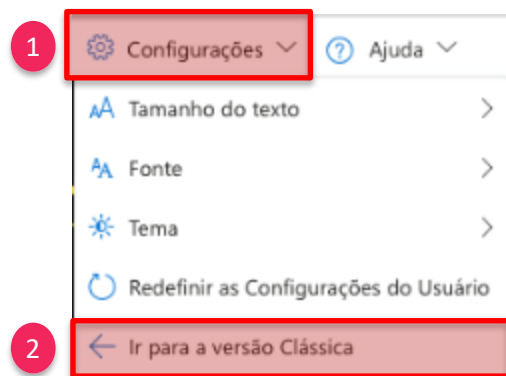
☒ Criaremos uma conta de armazenamento para você ⓘ

☐ Eu quero criar uma conta de armazenamento

Avançar

Anterior

Nesse momento vamos utilizar a versão clássica do Cloud Shell...



Tela do Azure Cloud Shell

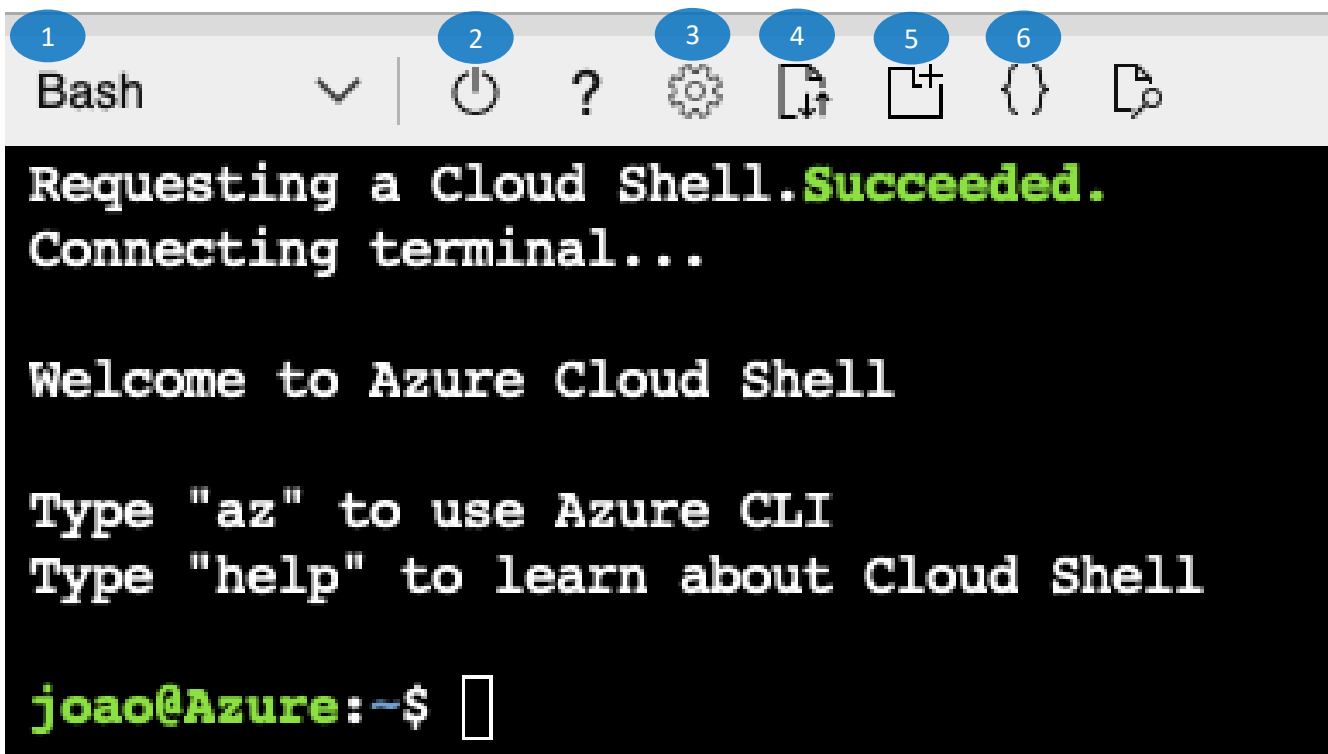
```
Bash  ▾ | 🔌 ? ⚙️ 📄 📁 {} 🗑️  
Requesting a Cloud Shell.Succeeded.  
Connecting terminal...  
  
Welcome to Azure Cloud Shell  
  
Type "az" to use Azure CLI  
Type "help" to learn about Cloud Shell  
  
joao@Azure:~$
```

Para uma melhor experiência maximize a tela do Azure Cloud Shell



Ou uma nova sessão em outra aba do Web Browser

Azure Cloud Shell



The screenshot shows the Azure Cloud Shell interface. At the top, there is a toolbar with icons for different shell types (1), a dropdown menu (2), a help icon (3), a settings icon (4), a file upload icon (5), a file download icon (6), and a text editor icon. Below the toolbar, the terminal displays the following text:

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

joao@Azure:~$
```

1 - Escolha entre um Terminal Bash ou PowerShell

2 – Reiniciar o Cloud Shell

3 – Configuração do tamanho do texto

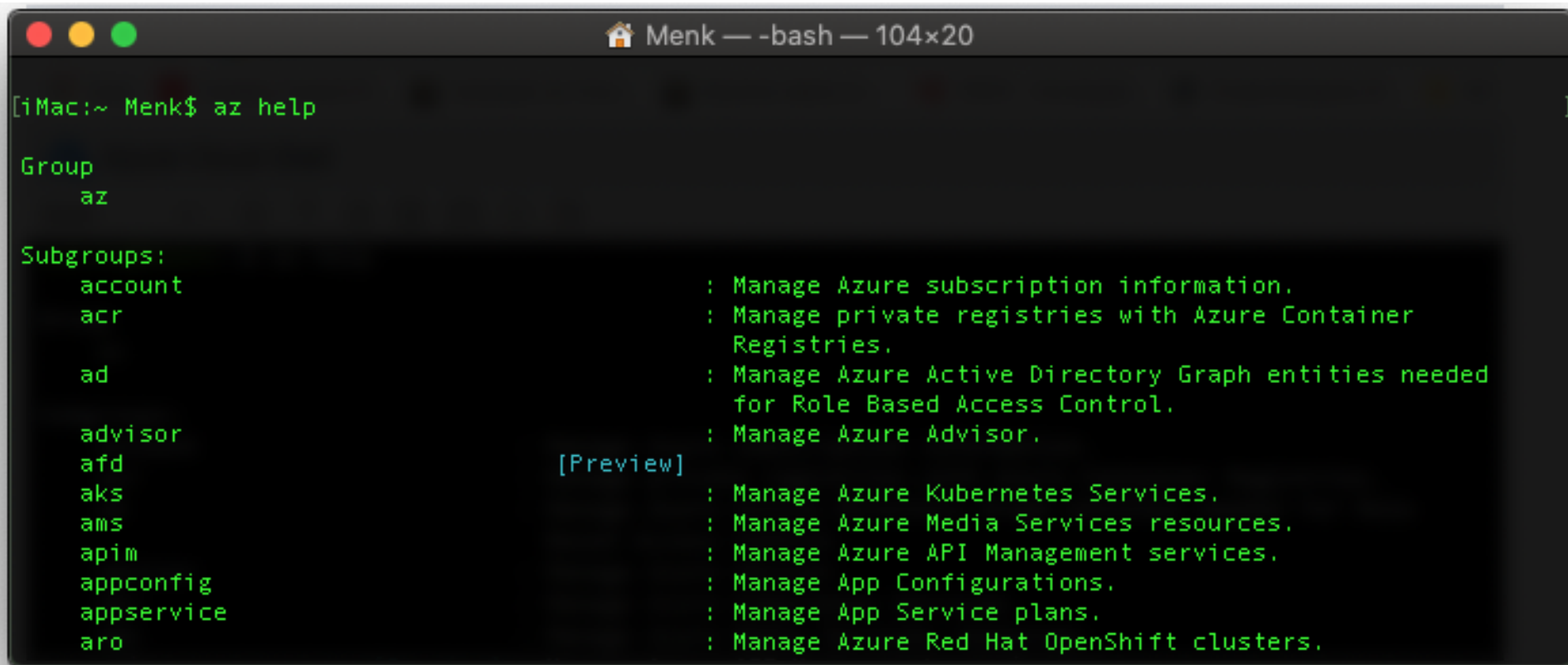
4 – Carregar Arquivos de sua máquina ou baixar Arquivos da Cloud Shell

5 – Abre uma nova Tab no navegador com uma nova sessão

6 – Editor de Texto interno



Exibindo o Help da CLI do Azure

az help

```
Menk — -bash — 104x20

[iMac:~ Menk$ az help]

Group
  az

Subgroups:
  account      : Manage Azure subscription information.
  acr          : Manage private registries with Azure Container Registries.
  ad           : Manage Azure Active Directory Graph entities needed for Role Based Access Control.
  advisor      : Manage Azure Advisor.
  afd          [Preview]
  aks          : Manage Azure Kubernetes Services.
  ams          : Manage Azure Media Services resources.
  apim         : Manage Azure API Management services.
  appconfig    : Manage App Configurations.
  appservice   : Manage App Service plans.
  aro          : Manage Azure Red Hat OpenShift clusters.
```


Para exibir alguns exemplos de utilização dos comandos da CLI do Azure digite:

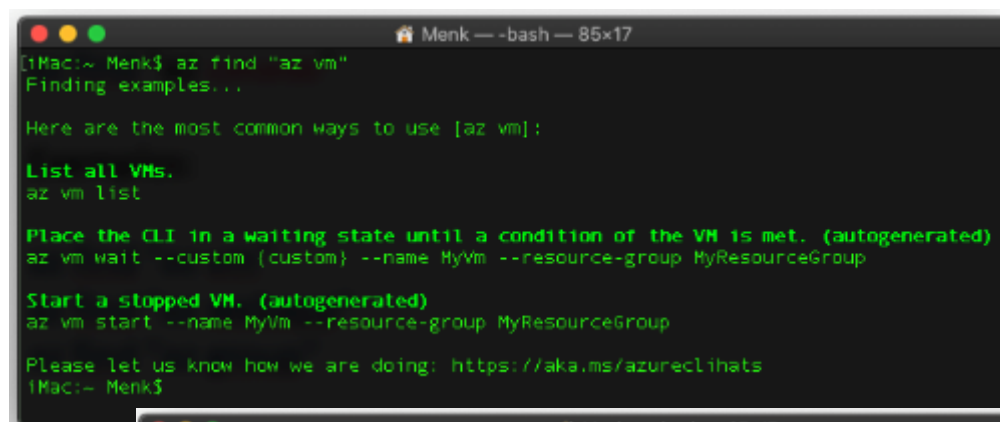
`az find "az comando"`

Exemplos:

`az find "az vm"`

`az find "az webapp"`

`az find "az group"`



```
Menk — -bash — 85x17
[1Mac:~ Menk$ az find "az vm"
Finding examples...

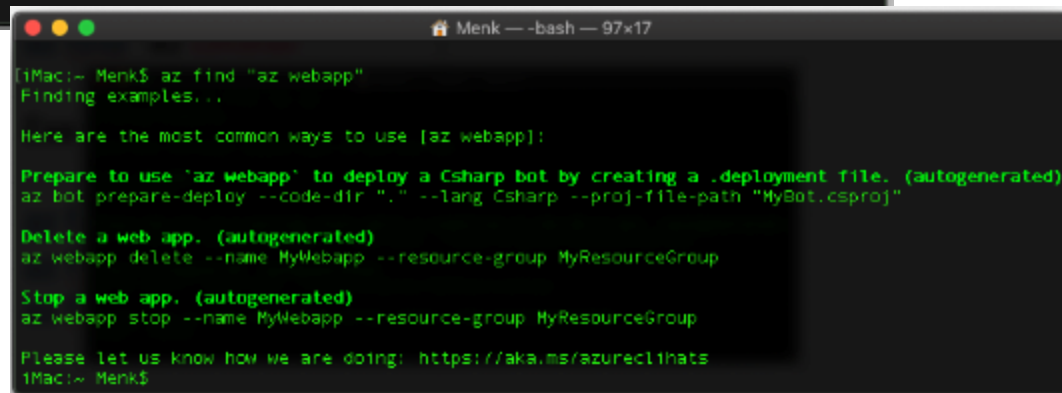
Here are the most common ways to use [az vm]:

List all VMs.
az vm list

Place the CLI in a waiting state until a condition of the VM is met. (autogenerated)
az vm wait --custom (custom) --name MyVm --resource-group MyResourceGroup

Start a stopped VM. (autogenerated)
az vm start --name MyVm --resource-group MyResourceGroup

Please let us know how we are doing: https://aka.ms/azureclihints
[1Mac:~ Menk$
```



```
Menk — -bash — 97x17
[1Mac:~ Menk$ az find "az webapp"
Finding examples...

Here are the most common ways to use [az webapp]:

Prepare to use 'az webapp' to deploy a Csharp bot by creating a .deployment file. (autogenerated)
az bot prepare-deploy --code-dir "." --lang Csharp --proj-file-path "MyBot.csproj"

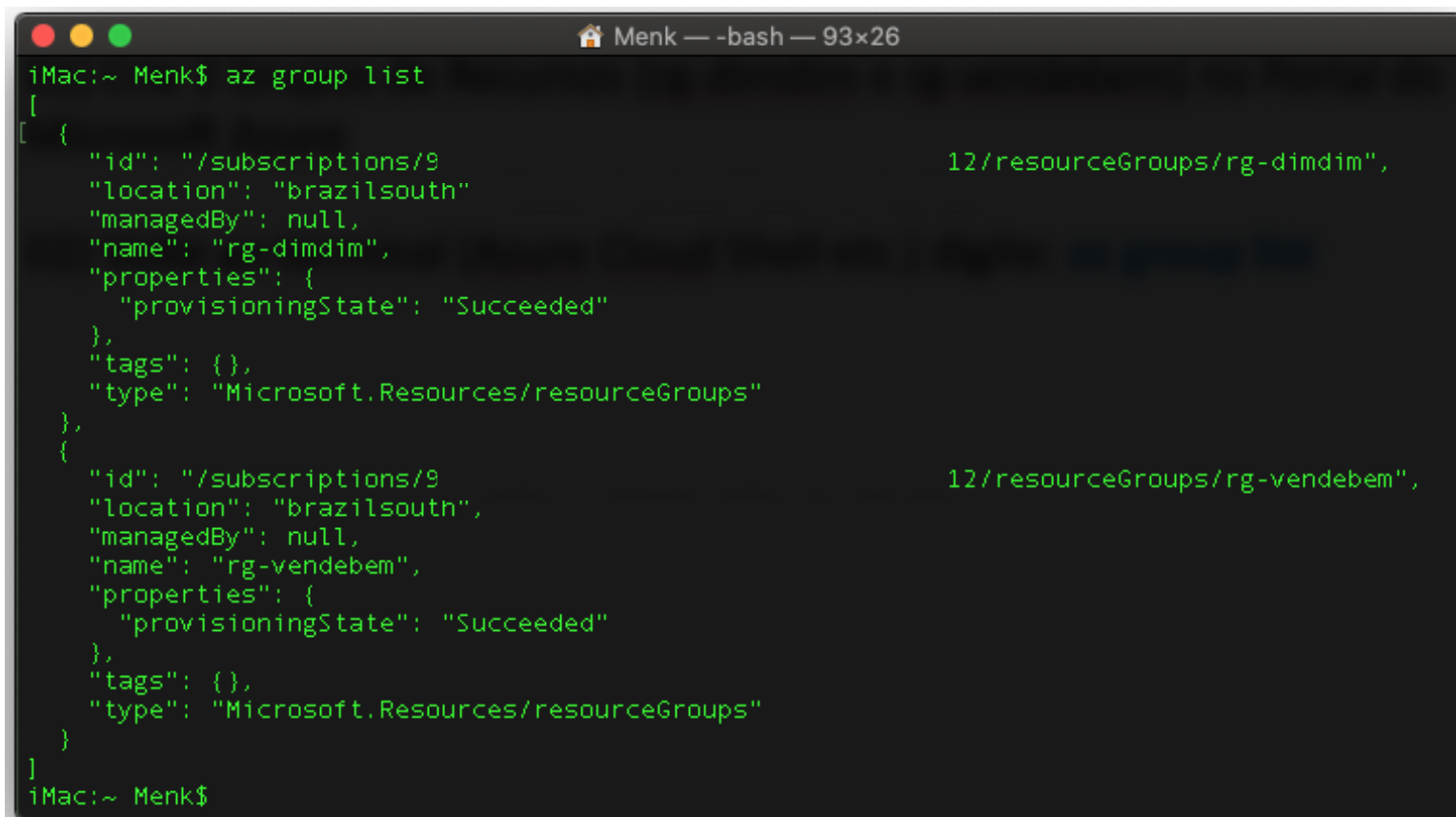
Delete a web app. (autogenerated)
az webapp delete --name MyWebapp --resource-group MyResourceGroup

Stop a web app. (autogenerated)
az webapp stop --name MyWebapp --resource-group MyResourceGroup

Please let us know how we are doing: https://aka.ms/azureclihints
[1Mac:~ Menk$
```

01) Crie 2 Grupos de Recursos: rg-dimdim e rg-vendebem no Portal do Microsoft Azure

02) Volte ao terminal e digite: **az group list --output table**

A terminal window titled 'Menk' with a width of 93x26. The prompt is 'iMac:~ Menk\$'. The command 'az group list' has been executed, resulting in a JSON array of two resource group objects. The first object is for 'rg-dimdim' and the second is for 'rg-vendebem'. Both are located in 'brazilsouth' and have a 'provisioningState' of 'Succeeded'.

```
iMac:~ Menk$ az group list
[
  {
    "id": "/subscriptions/91234567890123456789/resourceGroups/rg-dimdim",
    "location": "brazilsouth",
    "managedBy": null,
    "name": "rg-dimdim",
    "properties": {
      "provisioningState": "Succeeded"
    },
    "tags": {},
    "type": "Microsoft.Resources/resourceGroups"
  },
  {
    "id": "/subscriptions/91234567890123456789/resourceGroups/rg-vendebem",
    "location": "brazilsouth",
    "managedBy": null,
    "name": "rg-vendebem",
    "properties": {
      "provisioningState": "Succeeded"
    },
    "tags": {},
    "type": "Microsoft.Resources/resourceGroups"
  }
]
iMac:~ Menk$
```

Exibe detalhes de um Grupo de Recurso

```
echo "Entre com o nome do Grupo de Recursos:" &&  
read resourceGroupName &&  
az group show --name $resourceGroupName
```

```
joao@Azure:~$ echo "Entre com o nome do Grupo de Recursos:" &&  
> read resourceGroupName &&  
> az group show --name $resourceGroupName  
Entre com o nome do Grupo de Recursos:  
rg-dimdim  
{  
  "id": "/subscriptions/6a15/resourceGroups/rg-dimdim",  
  "location": "brazilsouth",  
  "managedBy": null,  
  "name": "rg-dimdim",  
  "properties": {  
    "provisioningState": "Succeeded"  
  },  
  "tags": {},  
  "type": "Microsoft.Resources/resourceGroups"  
}  
joao@Azure:~$
```

Criando um Grupo de Recurso

```
echo "Entre com o Nome do Grupo de Recursos:" &&  
read resourceGroupName &&  
echo "Entre com a localizacao (ex brazilsouth):" &&  
read location &&  
az group create --name $resourceGroupName --location $location
```

```
joao@Azure:~$ echo "Entre com o Nome do Grupo de Recursos:" &&  
> read resourceGroupName &&  
> echo "Entre com a localizacao (ex brazilsouth):" &&  
> read location &&  
> az group create --name $resourceGroupName --location $location  
Entre com o Nome do Grupo de Recursos:  
rg-corporacao  
Entre com a localizacao (ex brazilsouth):  
brazilsouth  
{  
  "id": "/subscriptions/6e[REDACTED]a15/resourceGroups/rg-corporacao",  
  "location": "brazilsouth",  
  "managedBy": null,  
  "name": "rg-corporacao",  
  "properties": {  
    "provisioningState": "Succeeded"  
  },  
  "tags": null,  
  "type": "Microsoft.Resources/resourceGroups"  
}  
joao@Azure:~$
```

Volte ao Portal e verifique se foi eliminado (é esperado um delay)

Excluindo um Grupo de Recurso

```
echo "Entre com o nome do Grupo de Recursos:" &&  
read resourceGroupName &&  
az group delete --name $resourceGroupName
```

```
joao@Azure:~$ echo "Entre com o nome do Grupo de Recursos:" &&  
> read resourceGroupName &&  
> az group delete --name $resourceGroupName  
Entre com o nome do Grupo de Recursos:  
rg-corporacao  
Are you sure you want to perform this operation? (y/n): y  
joao@Azure:~$
```

Volte ao Portal e verifique se foi eliminado (é esperado um delay)

Para excluir todos os Grupos de Recursos que estão vazios

```
for i in `az group list -o tsv --query [].name`;  
do if [ "$(az resource list -g $i -o tsv)" ]; then  
    echo "$i não esta vazio";  
else az group delete -n $i -y --no-wait;  
fi;  
done
```

```
joao@Azure:~$ for i in `az group list -o tsv --query [].name`; do if [ "$(az resource list -g $i -o tsv)" ]; then echo "$i nao esta vazio"; else az group delete -n $i -y --no-wait; fi; done  
cloud-shell-storage-eastus nao esta vazio  
joao@Azure:~$
```

CRIANDO UMA VM PELA CLI DO AZURE

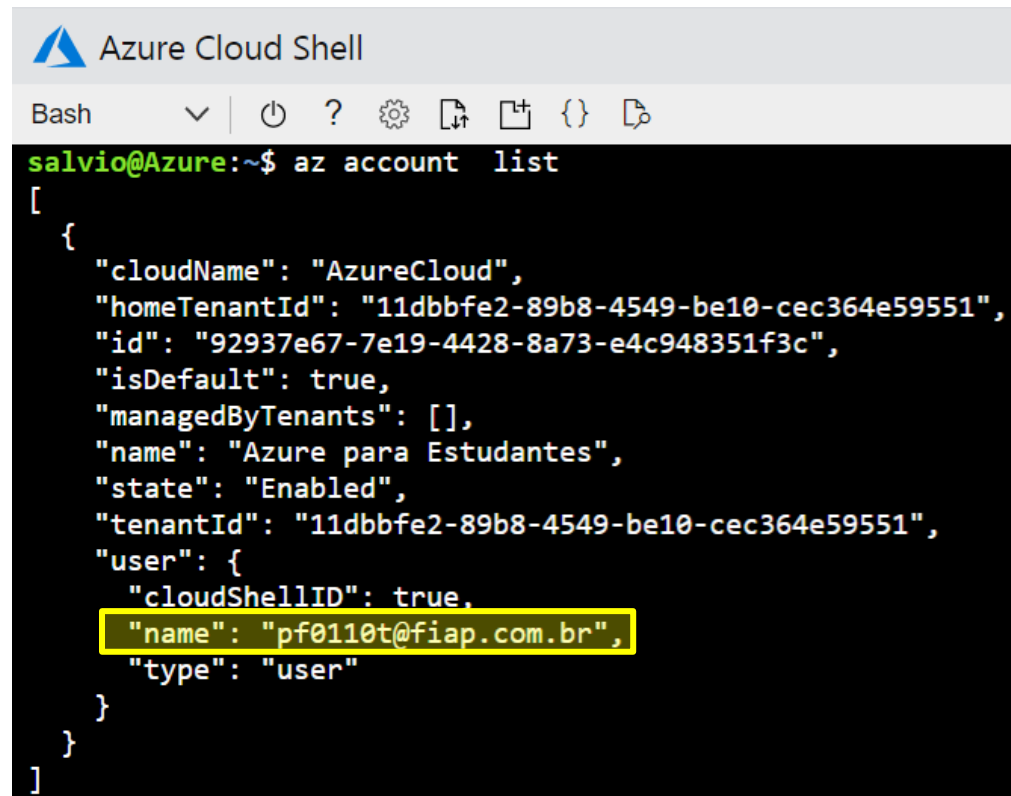
FIAP



Criando uma VM pelo CLI do Azure

Para recuperar as informações da nossa conta no Azure execute o comando abaixo:

az account list



```
Azure Cloud Shell
Bash
salvio@Azure:~$ az account list
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "11dbbfe2-89b8-4549-be10-cec364e59551",
    "id": "92937e67-7e19-4428-8a73-e4c948351f3c",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure para Estudantes",
    "state": "Enabled",
    "tenantId": "11dbbfe2-89b8-4549-be10-cec364e59551",
    "user": {
      "cloudShellID": true,
      "name": "pf0110t@fiap.com.br",
      "type": "user"
    }
  }
]
```


Criando uma VM pelo CLI do Azure

No Portal da Azure, por ser uma interface gráfica, a escolha do Data Center, imagem da VM, tamanho de disco, etc são facilmente selecionados

Região *	(South America) Sul do Brasil
Imagem *	Oracle Linux 7.8 – Gen1
	Procurar todas as imagens públicas e privadas
Tamanho *	Standard_B1ms - 1 vcpu, 2 GiB memória (R\$ 99,32/mês)
	Selecionar tamanho
Pilha de runtime *	Python 3.6
Computação + armazenamento	<div><div>Básico</div><div>2 vCores, 50 GB de armazenamento</div><div>Configurar servidor</div></div>

Para criarmos os recursos via linha de comando, precisamos recuperar algumas informações da Azure, como por exemplo, a lista de nomes das localizações dos Datacenters, Imagens das VMs, Tamanhos etc

Esse comando lista as localizações dos Datacenters Disponíveis no mundo para sua conta

```
az account list-locations
```

1) Utilize o comando abaixo para obter a lista dessas localizações

```
az account list-locations --output table > datacenters.txt
```

```
joao@Azure:~$ az account list-locations --output table > datacenters.txt
joao@Azure:~$
```

Criando uma VM pelo CLI do Azure

2) Com o arquivo aberto, procure no campo **RegionalDisplayName** o **nome da localização desejada** e recupere o valor da coluna **Name**

1

2

3

4

5

Clique no botão do Editor e do lado esquerdo da tela procure por `datacenters.txt`

1	DisplayName	Name	RegionalDisplayName
2			
3	East US	eastus	(US) East US
4	East US 2	eastus2	(US) East US 2
5	South Central US	southcentralus	(US) South Central US
6	West US 2	westus2	(US) West US 2
7	West US 3	westus3	(US) West US 3
8	Australia East	australiaeast	(Asia Pacific) Austr
9	Southeast Asia	southeastasia	(Asia Pacific) Southe
10	North Europe	northeurope	(Europe) North Europe
11	UK South	uksouth	(Europe) UK South
12	West Europe	westeurope	(Europe) West Europe
13	Central US	centralus	(US) Central US
14	North Central US	northcentralus	(US) North Central US
15	West US	westus	(US) West US
16	South Africa North	southafricanorth	(Africa) South Africa
17	Central India	centralindia	(Asia Pacific) Centra
18	East Asia	eastasia	(Asia Pacific) East Asia
19	Japan East	japaneast	(Asia Pacific) Japan East
20	Jio India West	jioindiawest	(Asia Pacific) Jio India Wes
21	Korea Central	koreacentral	(Asia Pacific) Korea Central
22	Canada Central	canadacentral	(Canada) Canada Central
23	France Central	francecentral	(Europe) France Central
24	Germany West Central	germanywestcentral	(Europe) Germany West Centra
25	Norway East	norwayeast	(Europe) Norway East
26	Switzerland North	switzerlandnorth	(Europe) Switzerland North
27	UAE North	uaenorth	(Middle East) UAE North
28	Brazil South	brazilsouth	(South America) Brazil South
29	Central US (Stage)	centralusstage	(US) Central US (Stage)
30	East US (Stage)	eastusstage	(US) East US (Stage)
31	East US 2 (Stage)	eastus2stage	(US) East US 2 (Stage)
32	North Central US (Stage)	northcentralusstage	(US) North Central US (Stage)
33	South Central US (Stage)	southcentralusstage	(US) South Central US (Stage)
34	West US (Stage)	westusstage	(US) West US (Stage)
35	West US 2 (Stage)	westus2stage	(US) West US 2 (Stage)

Já temos o nome do Datacenter e agora vamos recuperar a lista de Imagens de VMs disponíveis no Marketplace

Sintaxe:

```
az vm image list [--all]
                  [--location]
                  [--offer]
                  [--publisher]
                  [--sku]
                  [--subscription]
```

--all

Recupere a lista de imagens do serviço Azure on line ao invés de usar uma lista de imagens offline

--offer -f

Nome da Imagem Oferecida (realiza busca parcial)

--publisher -p

Nome do Publicador da Imagem (realiza busca parcial)



1) Utilize o comando abaixo para obter a lista das VMs da *Canonical*

```
az vm image list --all -p canonical -f UbuntuServer --output table > vmsubuntu.txt
```

Criando uma VM pelo CLI do Azure

2) Com o arquivo aberto procure sua Imagem em **Offer** e recupere o valor do campo **Urn** (Nome da VM) conforme selecionado abaixo

```
Bash
```

ARQUIVOS

azures

cloudrive

bash_history

bash_logout

bashrc

profile

tmux.conf

vimrc

vmsubuntu.txt

vmsubuntu.txt

2799	x64	UbuntuServer	Canonical	19_04-daily-gen2	Canonical:UbuntuServer:19_04-daily-gen2:19.04.20191
2800	x64	UbuntuServer	Canonical	19_04-daily-gen2	Canonical:UbuntuServer:19_04-daily-gen2:19.04.20200
2801	x64	UbuntuServer	Canonical	19_04-daily-gen2	Canonical:UbuntuServer:19_04-daily-gen2:19.04.20200
2802	x64	UbuntuServer	Canonical	19_04-daily-gen2	Canonical:UbuntuServer:19_04-daily-gen2:19.04.20200
2803	x64	UbuntuServer	Canonical	19_04-daily-gen2	Canonical:UbuntuServer:19_04-daily-gen2:19.04.20200
2804	x64	UbuntuServer	Canonical	19_04-daily-gen2	Canonical:UbuntuServer:19_04-daily-gen2:19.04.20200
2805	x64	UbuntuServer	Canonical	19_04-daily-gen2	Canonical:UbuntuServer:19_04-daily-gen2:19.04.20200
2806	x64	UbuntuServer	Canonical	19_04-daily-gen2	Canonical:UbuntuServer:19_04-daily-gen2:19.04.20200
2807	x64	UbuntuServer	Canonical	19_04-daily-gen2	Canonical:UbuntuServer:19_04-daily-gen2:19.04.20200
2808	x64	UbuntuServer	Canonical	19_04-gen2	Canonical:UbuntuServer:19_04-gen2:19.04.201908230
2809	x64	UbuntuServer	Canonical	19_04-gen2	Canonical:UbuntuServer:19_04-gen2:19.04.201910050
2810	x64	UbuntuServer	Canonical	19_04-gen2	Canonical:UbuntuServer:19_04-gen2:19.04.201911080
2811	x64	UbuntuServer	Canonical	19_04-gen2	Canonical:UbuntuServer:19_04-gen2:19.04.201911131
2812	x64	UbuntuServer	Canonical	19_04-gen2	Canonical:UbuntuServer:19_04-gen2:19.04.202001220
2813	x64	UbuntuServer	Canonical	19_10-daily-gen2	Canonical:UbuntuServer:19_10-daily-gen2:19.10.20200
2814	x64	UbuntuServer	Canonical	19_10-daily-gen2	Canonical:UbuntuServer:19_10-daily-gen2:19.10.20200
2815	x64	UbuntuServer	Canonical	19_10-daily-gen2	Canonical:UbuntuServer:19_10-daily-gen2:19.10.20200
2816	x64	UbuntuServer	Canonical	19_10-daily-gen2	Canonical:UbuntuServer:19_10-daily-gen2:19.10.20200
2817	x64	UbuntuServer	Canonical	19_10-daily-gen2	Canonical:UbuntuServer:19_10-daily-gen2:19.10.20200
2818	x64	UbuntuServer	Canonical	19_10-daily-gen2	Canonical:UbuntuServer:19_10-daily-gen2:19.10.20200
2819	x64	UbuntuServer	Canonical	19_10-daily-gen2	Canonical:UbuntuServer:19_10-daily-gen2:19.10.20200
2820					

Criando uma VM pelo CLI do Azure

O comando abaixo recupera a lista dos tamanhos disponíveis para VMs

Sintaxe:

```
az vm list-sizes --location  
[--subscription]
```



1) Utilize o comando abaixo para obter a lista dos Tamanhos das VMs

```
az vm list-sizes -l brazilsouth --output table > vmsizes.txt
```

2) Com o arquivo aberto procure o tamanho desejado da VM e recupere o valor do campo **Name** (Tamanho da VM)

MaxDataDiskCount	MemoryInMb	Name	NumberOfCores	OsDiskSizeInMb	ResourceDiskSizeInMb
2	2048	Standard_A1_v2	1	1047552	10240
4	16384	Standard_A2m_v2	2	1047552	20480
4	4096	Standard_A2_v2	2	1047552	20480
8	32768	Standard_A4m_v2	4	1047552	40960
8	8192	Standard_A4_v2	4	1047552	40960
16	65536	Standard_A8m_v2	8	1047552	81920
16	16384	Standard_A8_v2	8	1047552	81920
2	512	Standard_B1ls	1	1047552	4096
2	2048	Standard_B1ms	1	1047552	4096
2	1024	Standard_B1s	1	1047552	4096
4	8192	Standard_B2ms	2	1047552	16384
4	4096	Standard_B2s	2	1047552	8192
8	16384	Standard_B4ms	4	1047552	32768
16	32768	Standard_B8ms	8	1047552	65536
16	49152	Standard_B12ms	12	1047552	98304
32	65536	Standard_B16ms	16	1047552	131072
32	81920	Standard_B20ms	20	1047552	163840
4	8192	Standard_D2ds_v4	2	1047552	76800
8	16384	Standard_D4ds_v4	4	1047552	153600
16	32768	Standard_D8ds_v4	8	1047552	307200



```
az group create --name rg-vmubuntu --location brazilsouth
```

Botão do contêiner de terminal

| Criando uma VM pelo CLI do Azure

O comando **az vm** gerencia as Máquinas Virtuais em seu ambiente

az vm create --name --resource-group

```
...
[--admin-password]
[--admin-username]
...
[--attach-data-disks]
...
[--generate-ssh-keys]
...
[--ephemeral-os-disk {false, true}]
...
[--image]
...
[--location]
...
[--os-type {linux, windows}]
...
[--public-ip-address]
...
[--size]
...
[--tags]
...
[--vnet-name]
...
```



| Criando uma VM pelo CLI do Azure

Agora vamos criar a **Máquina Virtual** com a **Imagem** do Servidor Ubuntu através do comando abaixo



```
az vm create \  
  --resource-group rg-vmubuntu \  
  --name vm-ubuntu \  
  --image Canonical:UbuntuServer:19_04-gen2:19.04.201908230 \  
  --size Standard_B2s \  
  --vnet-name nnet-Linux \  
  --nsg nsgr-linux \  
  --public-ip-address pip-ubuntu \  
  --authentication-type password \  
  --admin-username adminx \  
  --admin-password Fiap@2tdsvms
```



Criando uma VM pelo CLI do Azure

Algumas VMs requerem um aceite nos termos legais antes de criá-las

```
Menk — -bash — 96x22
iMac:~ Menk$ az vm create\
> --resource-group gr-corporacao\
> --name vm-teste-penetracao\
> --image kali-linux:kali-linux:kali:2019.2.0\
> --size Standard_B1ms\
> --storage-sku Standard_LRS\
> --data-disk-sizes-gb 30\
> --admin-username intruder\
> --admin-password cloud-2020@site
Azure Error: MarketplacePurchaseEligibilityFailed
Message: Marketplace purchase eligibility check returned errors. See inner errors for details.
Exception Details:
  Error Code: BadRequest
  Message: Offer with PublisherId: 'kali-linux', OfferId: 'kali-linux' cannot be purchased
  due to validation errors. For more information see details. Correlation Id: '0055236a-c3e6-44b7-
  b59b-a5f18c8c5219' You have not accepted the legal terms on this subscription: '9cc674fb-9385-4
  9d0-a6bb-449743b9fd12' for this plan. Before the subscription can be used, you need to accept th
  e legal terms of the image. To read and accept legal terms, use the Azure CLI commands described
  at https://go.microsoft.com/fwlink/?linkid=2110637 or the PowerShell commands available at http
  s://go.microsoft.com/fwlink/?linkid=862451. Alternatively, deploying via the Azure portal provid
  es a UI experience for reading and accepting the legal terms. Offer details: publisher='kali-lin
  ux' offer = 'kali-linux', sku = 'kali', Correlation Id: '0055236a-c3e6-44b7-b59b-a5f18c8c5219'.
```



Execute o comando **az vm image terms accept** para aceitar os termos legais da VM escolhida, informando a imagem escolhida. Exs.:

az vm image terms accept --urn kali-linux:kali:kali-2023-3:2023.3.0

az vm image terms accept --urn Oracle:Oracle-Database-Ee:12.1.0.2:12.1.20170220

Criando uma VM pelo CLI do Azure

Depois de executar o comando, a CLI do Azure exibe informações importantes, uma delas é o IP Público

Copie o valor de **publicIpAddress**

```
Bash
joao@Azure:~$ az vm create \
> --resource-group rg-vmububu \
> --name vm-ububu \
> --image Canonical:UbuntuServer:19_04-gen2:19.04.201908230 \
> --size Standard_B2s \
> --vnet-name nnet-linux \
> --nsg nsgr-linux \
> --public-ip-address pip-ubuntu \
> --authentication-type password \
> --admin-username admln \
> --admin-password Fiap@2tds2022
It is recommended to use parameter "--public-ip-sku Standard" to create new VM with Standard public IP. Please note that the default public IP u
e.
{
  "fqdns": "",
  "id": "/subscriptions/9cc674fb-9385-49d0-a6bb-449743b9fd12/resourceGroups/rg-vmububu/providers/Microsoft.Compute/virtualMachines/vm-ububu",
  "location": "brazilsouth",
  "macAddress": "00-22-48-36-33-E5",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "20.226.122.64",
  "resourceGroup": "rg-vmububu",
  "zones": ""
}
joao@Azure:~$
```

Criando uma VM pelo CLI do Azure

A Máquina Virtual que será disponibilizada terá as seguintes configurações



Máquina virtual

Nome do computador	vm-ubuntu
Estado de integridade	-
Sistema operacional	Linux (ubuntu 19.04)
Editor	Canonical
Oferta	UbuntuServer
Plano	19_04-gen2
Geração de VM	V2
Status do agente	Ready
Versão do agente	2.7.3.0
Grupo de hosts	Nenhum
Host	-
Grupo de posicionamento por proximidade	-
Status de Colocalização	N/D
Grupo de reserva de capacidade	-



Rede

Endereço IP público	20.226.122.64
Endereço IP público (IPv6)	-
Endereço IP privado	10.0.0.4
Endereço IP privado (IPv6)	-
Rede virtual/sub-rede	nnet-linux/vm-ubuntuSubnet
Nome DNS	Configurar



Tamanho

Tamanho	Standard B2s
vCPUs	2
RAM	4 GiB

Disco do SO



Trocar o disco do SO

Nome do disco	Tipo de armazena...	Tamanho (GiB)	IOPS Máxima	Taxa de transferê...	Criptografia ⓘ
vm-ubuntu_OsDisk_1_1dcc7f77444e4bc4a6ff358a52510d7e	LRS do SSD Premium	30	120	25	SSE com PMK

Entrando na parte de Segurança, vamos abrir as portas da VM para a conectividade

Vamos criar uma regra de **Grupo de Segurança** (NSG) para liberar a porta 80 em nosso Servidor. Execute o comando abaixo:

```
az network nsg rule create \  
  --resource-group rg-vmubuntu \  
  --nsg-name nsgsr-linux \  
  --name port_80 \  
  --protocol tcp \  
  --priority 1010 \  
  --destination-port-range 80
```

```
joao@Azure:~$ az network nsg rule create \  
> --resource-group rg-vmubuntu \  
> --nsg-name nsgsr-linux \  
> --name port_80 \  
> --protocol tcp \  
> --priority 1010 \  
> --destination-port-range 80  
{  
  "access": "Allow",  
  "description": null,  
  "destinationAddressPrefix": "*",  
  "destinationAddressPrefixes": [],  
  "destinationApplicationSecurityGroups": null,  
  "destinationPortRange": "80",  
  "destinationPortRanges": [],  
  "direction": "Inbound",  
  "etag": "W/\"c49bd5fe-e089-4587-9ae8-1eb525dcf490\"",  
  "id": "/subscriptions/9cc674fb-9385-49d0-a6bb-449743b9fd12/resou  
  "name": "port_80",  
  "priority": 1010,  
  "protocol": "Tcp",  
  "provisioningState": "Succeeded",  
  "resourceGroup": "rg-vmubuntu",  
  "sourceAddressPrefix": "*",  
  "sourceAddressPrefixes": [],  
  "sourceApplicationSecurityGroups": null,  
  "sourcePortRange": "*",  
  "sourcePortRanges": [],  
  "type": "Microsoft.Network/networkSecurityGroups/securityRules"  
}
```

NSG = Network Security Groups

Criando uma VM pelo CLI do Azure

Tudo certo, iremos nos **conectar na VM** através do comando abaixo:

Sintaxe:

`ssh user@publicIpAddress`

Onde **user** é o nome do Adm da VM e **publicIpAddress** é o IP copiado anteriormente

Exemplo:

`ssh admlnx@seu-IP`

```
Menk — admlnx@vm-ubuntu: ~ — ssh admlnx@20.226.122.64 — 108x35
[Mac:~ Menk] ssh admlnx@20.226.122.64
The authenticity of host '20.226.122.64 (20.226.122.64)' can't be established.
ECDSA key fingerprint is SHA256:sn9ftW4vnp/6QUn6j6EAsM5n8fT44GaqDN+e9nZn4/o.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '20.226.122.64' (ECDSA) to the list of known hosts.
admlnx@20.226.122.64's password:
Welcome to Ubuntu 19.04 (GNU/Linux 5.0.8-1016-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Jul 26 17:45:03 UTC 2022

System load:  0.08               Processes:    131
Usage of /:   4.3% of 28.90GB     Users logged in:  0
Memory usage: 7%                 IP address for eth0: 10.0.0.4
Swap usage:   0%

0 updates can be installed immediately.
0 of these updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

admlnx@vm-ubuntu:~$
```




Automação do Gerenciamento e da Criação de Recursos

Podemos realizar vários procedimentos automatizando comandos

Vamos criar a automação de Redimensionamento de um Servidor Virtual, deixando a elasticidade muito mais fácil de ser utilizada pelos usuários

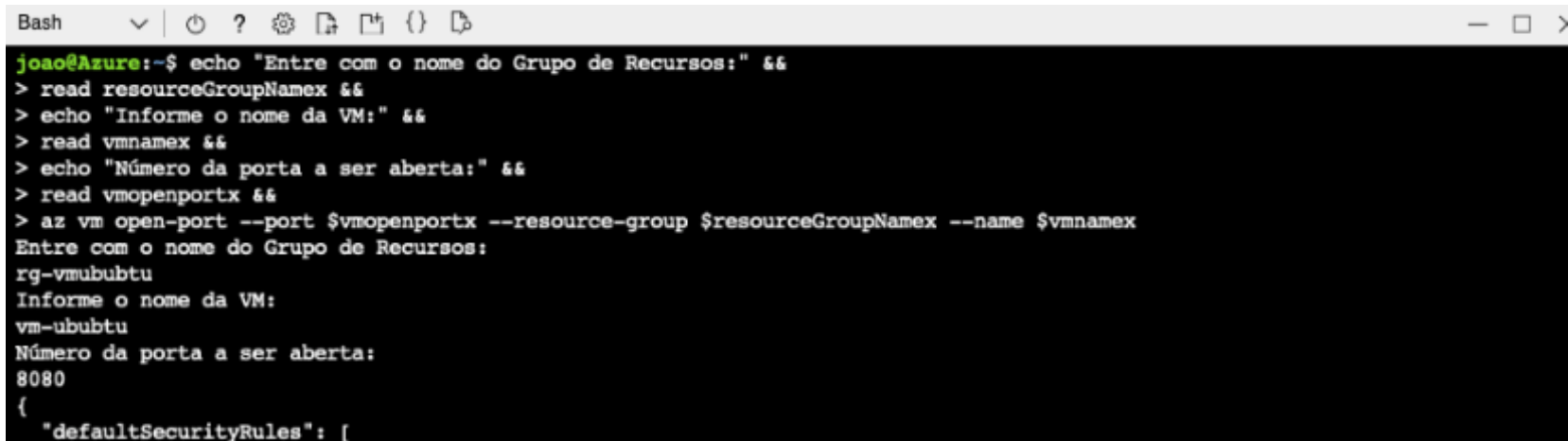
```
echo "Entre com o nome do Grupo de Recursos:" &&  
read resourceGroupName &&  
echo "Informe o nome da VM:" &&  
read vmname &&  
echo "Escolha o novo tamanho:" &&  
select vmsize in "01 CPU / 2GB RAM" "02 CPUs / 8GB RAM" "04 CPUs / 16GB RAM"; do case  
$vmsize in "01 CPU / 2GB RAM") export vmsize="Standard_B1ms";; "02 CPUs / 8GB RAM")  
export vmsize="Standard_D2s_v3";; "04 CPUs / 16GB RAM") export vmsize="Standard_B4ms";;  
esac; break; done &&  
az vm resize -g $resourceGroupName -n $vmname --size $vmsize
```



```
Bash
joao@Azure:~$ echo "Entre com o nome do Grupo de Recursos:" &&
> read resourceGroupName &&
> echo "Informe o nome da VM:" &&
> read vmName &&
> echo "Escolha o novo tamanho:" &&
> select disksize in "01 CPU / 2GB RAM" "02 CPUs / 8GB RAM" "04 CPUs / 16GB RAM"; do case $disksize in "01 CPU / 2GB RAM") export disksize="Standard_B1ms"; "02 CPUs / 8GB RAM") export disksize="Standard_D2s_v3"; "04 CPUs / 16GB RAM") export disksize="Standard_B4ms"; esac; break; done &&
> az vm resize -g $resourceGroupName -n $vmName --size $disksize
Entre com o nome do Grupo de Recursos:
rg-vmubutu
Informe o nome da VM:
vm-ubutu
Escolha o novo tamanho:
1) 01 CPU / 2GB RAM
2) 02 CPUs / 8GB RAM
3) 04 CPUs / 16GB RAM
#? 1
{
  "additionalCapabilities": null,
  "applicationProfile": null,
  "availabilitySet": null,
  "billingProfile": null
}
```

Dependendo do serviço que estamos criando precisamos abrir portas de um Servidor Virtual, com o script abaixo realizaremos essa tarefa

```
echo "Entre com o nome do Grupo de Recursos:" &&
read resourceGroupNamex &&
echo "Informe o nome da VM:" &&
read vmnamex &&
echo "Número da porta a ser aberta:" &&
read vmopenportx &&
az vm open-port --resource-group $resourceGroupNamex --name $vmnamex --port $vmopenportx --priority $((RANDOM%4096+100))
```

A terminal window titled 'Bash' with standard window controls. It shows the execution of the script. The user is 'joao@Azure'. The script prompts for the resource group name, VM name, and port number. The user enters 'rg-vmububu', 'vm-ubuntu', and '8080'. The script then runs the 'az vm open-port' command. The output shows the command being executed and the beginning of a JSON response for 'defaultSecurityRules'.

```
Bash
joao@Azure:~$ echo "Entre com o nome do Grupo de Recursos:" &&
> read resourceGroupNamex &&
> echo "Informe o nome da VM:" &&
> read vmnamex &&
> echo "Número da porta a ser aberta:" &&
> read vmopenportx &&
> az vm open-port --port $vmopenportx --resource-group $resourceGroupNamex --name $vmnamex
Entre com o nome do Grupo de Recursos:
rg-vmububu
Informe o nome da VM:
vm-ubuntu
Número da porta a ser aberta:
8080
{
  "defaultSecurityRules": [
```

O modo correto é criar um Shell Script com o conteúdo desejado e executar sempre que necessário

```
touch abrePorta.sh
```

```
nano abrePorta.sh
```

```
echo "Entre com o nome do Grupo de Recursos:" &&
```

```
read resourceGroupName &&
```

```
echo "Informe o nome da VM:" &&
```

```
read vmname &&
```

```
echo "Número da porta a ser aberta:" &&
```

```
read vmopenport &&
```

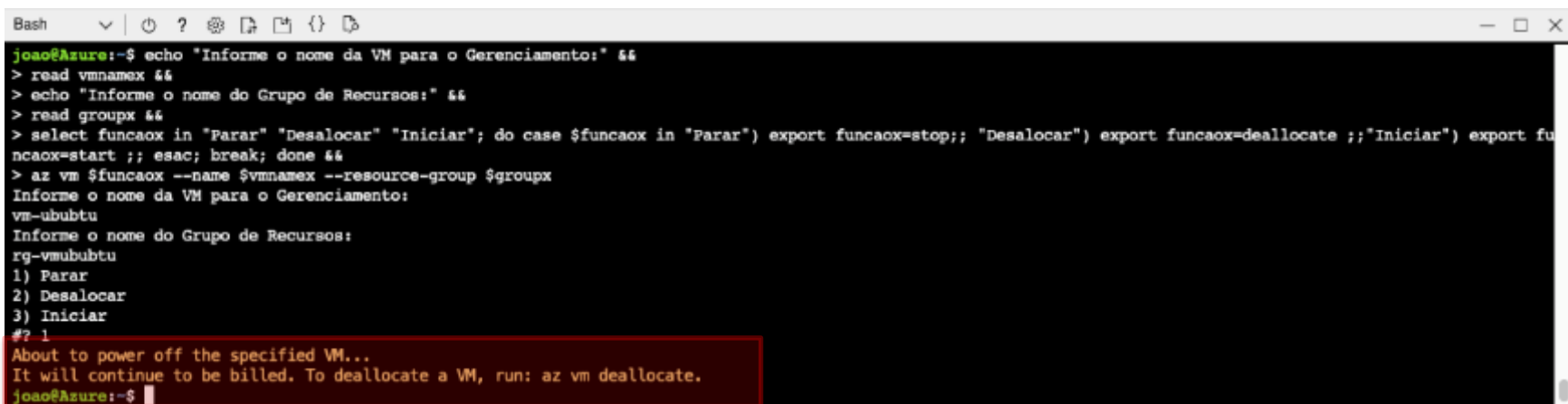
```
az vm open-port --resource-group $resourceGroupName --name $vmname --port $vmopenport --priority $((RANDOM%4096+100))
```

```
chmod 744 abrePorta.sh
```

```
./abrePorta.sh
```

Nada melhor que o próprio usuário gerenciar seus recursos em nuvem. Esse script permite que o gerenciamento de uma VM seja feita de forma bem simples

```
echo "Informe o nome da VM para o Gerenciamento:" &&
read vmnamex &&
echo "Informe o nome do Grupo de Recursos:" &&
read groupx &&
select funcao in "Parar" "Desalocar" "Iniciar"; do case $funcao in "Parar") export funcao=stop;;
"Desalocar") export funcao=deallocate ;; "Iniciar") export funcao=start ;; esac; break; done &&
az vm $funcao --name $vmnamex --resource-group $groupx
```



```
Bash
joao@Azure:~$ echo "Informe o nome da VM para o Gerenciamento:" &&
> read vmnamex &&
> echo "Informe o nome do Grupo de Recursos:" &&
> read groupx &&
> select funcao in "Parar" "Desalocar" "Iniciar"; do case $funcao in "Parar") export funcao=stop;; "Desalocar") export funcao=deallocate ;; "Iniciar") export funcao=start ;; esac; break; done &&
> az vm $funcao --name $vmnamex --resource-group $groupx
Informe o nome da VM para o Gerenciamento:
vm-ububu
Informe o nome do Grupo de Recursos:
rg-vmubu
1) Parar
2) Desalocar
3) Iniciar
#? 1
About to power off the specified VM...
It will continue to be billed. To deallocate a VM, run: az vm deallocate.
joao@Azure:~$
```

Prover uma VM Linux de forma rápida para a realização de testes em novas tecnologias

```
## Variáveis
grupox="rg-lnx-new-tech" &&
locationx="brazilsouth" &&
## Criação do Grupo de Serviços
az group create --name $grupox --location $locationx &&
## Dados da VM
echo "Informe o nome da VM:" &&
read vmnamex &&
echo "Escolha o tipo da imagem:" &&
select vmtypex in "Oracle 8.9" "Ubuntu 19"; do case $vmtypex in "Oracle 8.9") export vmtypex="Oracle:Oracle-Linux:ol89-lvm-gen2:8.9.5";; "Ubuntu 19") export vmtypex="Canonical:UbuntuServer:19_04-gen2:19.04.2019";;
echo "Informe o tamanho do disco (em GB - somente numeros):" &&
read disksizex &&
echo "Informe o usuário:" &&
read usuarioux &&
echo "Informe a senha:" &&
read senhax &&
az vm create --resource-group $grupox --name $vmnamex --image $vmtypex --size Standard_B1ms --data-disk-sizes-gb $disksizex --storage-sku Standard_LRS --admin-username $usuarioux --admin-password $senhax
```

Para executar a criação completa de um Web App de forma automatizada

Variaveis do Script

```
rg=rg-app-quiz &&
```

```
location=brazilsouth &&
```

```
appServicePlanName=plan-app-quiz &&
```

```
webAppName=appquizrm9999 &&
```



Altere para seu RM

```
codigoFonte=https://github.com/profjoaomenk/jsQuiz.git &&
```

```
sku=F1 &&
```

Cria um novo Grupo de Recursos

```
az group create --name $rg --location $location &&
```

Cria um novo Plano de Serviço para o App

```
az appservice plan create --name $appServicePlanName \
```

```
--location $location \
```

```
--resource-group $rg \
```

```
--sku $sku &&
```

Cria um novo Servico de Aplicativo

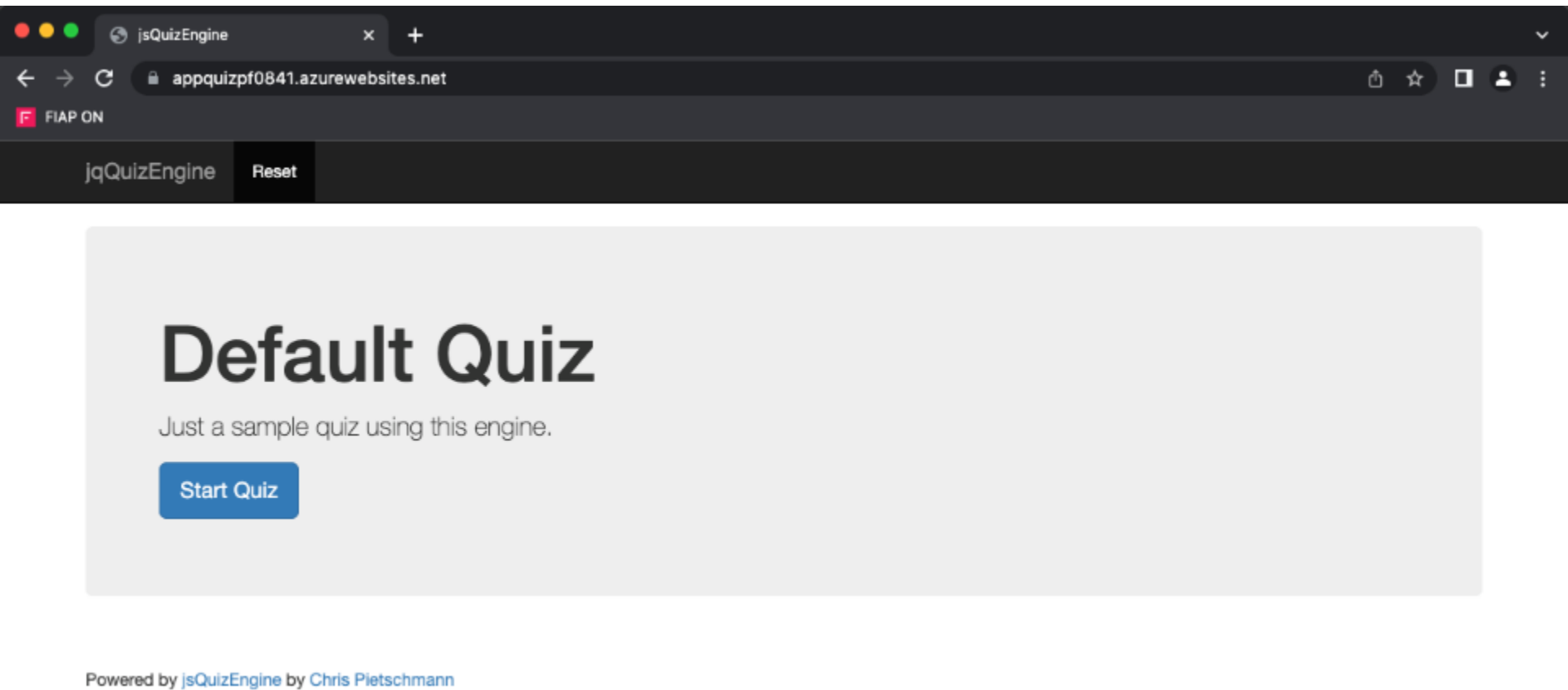
```
az webapp create -g $rg \
```

```
-p $appServicePlanName \
```

```
-n $webAppName \
```

```
--deployment-source-url $codigoFonte
```

Para executar a criação completa de um Web App de forma automatizada – **Testando a Aplicação**



Para deletar nossos recursos criados e limpar o Lab:

```
az group delete --name rg-vmubuntu -y
```

```
az group delete --name NetworkWatcherRG -y
```

```
az group delete --name rg-lnx-new-tech -y
```

```
az group delete --name rg-app-quiz -y
```

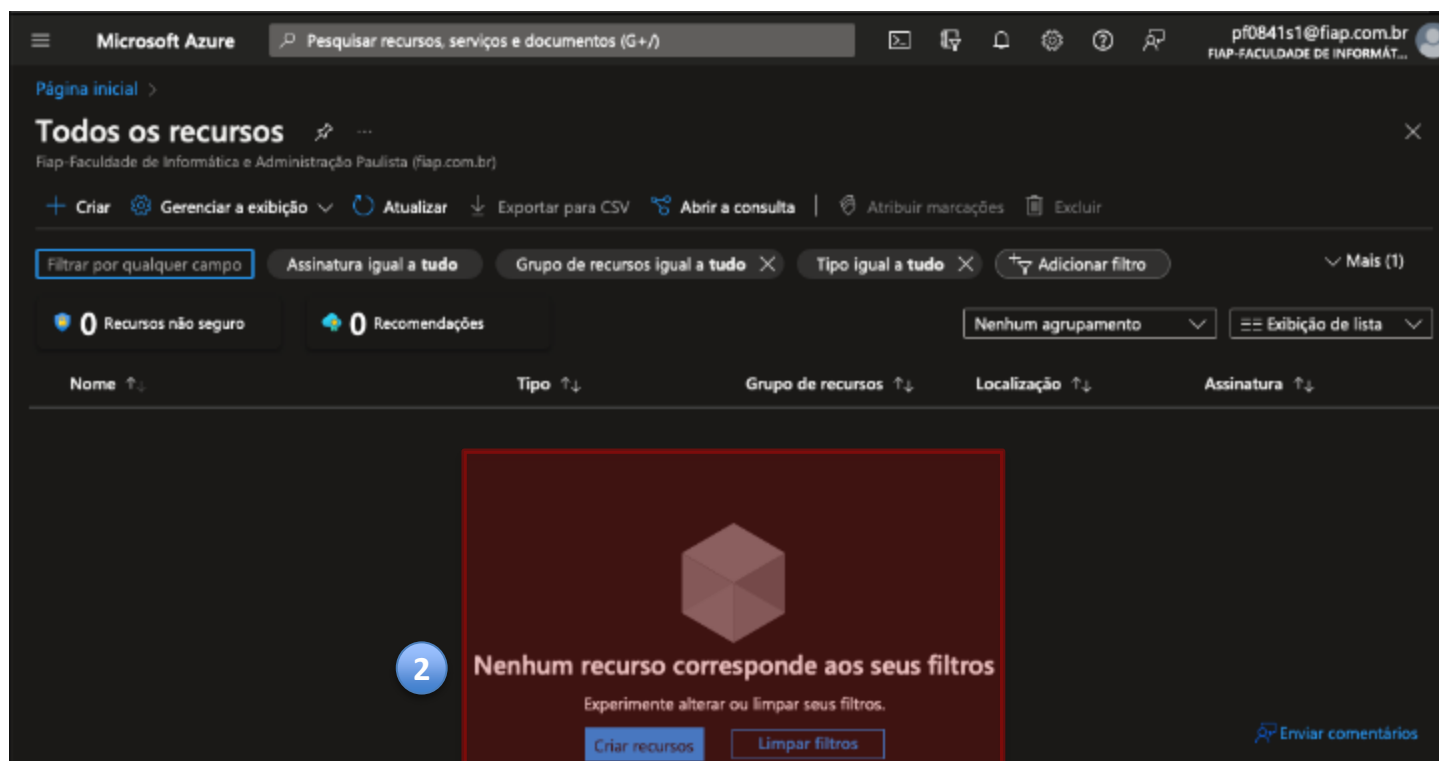
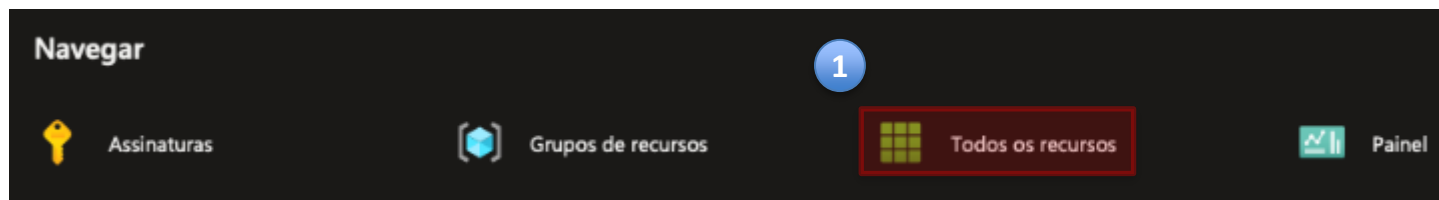
```
az group delete --name cloud-shell-storage-eastus -y
```

Deletar todos os Grupos de Recursos criados em uma linha somente:

```
az group delete --name rg-vmubuntu -y ; az group delete --name NetworkWatcherRG -y ; az group delete --name rg-lnx-new-tech -y ; az group delete --name rg-app-quiz -y ; az group delete --name cloud-shell-storage-eastus -y
```


ATENÇÃO

VERIFIQUE SE NÃO EXISTE ALGUM RECURSO CRIADO



Copyright © 2024 Prof. João Menk

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor)