az login - can help u to connect with azure

az account set --subscription <subscriptionid>

az deployment sub create -l eastus -f <file1>.bicep {create resource group in context of subscription}

az deployment group create --resource-group biceps --template-file .\armtemplates.json

az deployment group create --resource-group biceps --template-file .\biceptemplates.bicep {it will create template file automatically}

**how to preview bicep changes before running**

az deployment group create --resource-group biceps --template-file .\biceptemplates.bicep --confirm-with-what-if or -c

adding ip by biceps in whitelist of firewall for sql will not override existing one

az bicep build -f <file>.bicep {will create arm template}

after this using arm tool extension create separate parameters.json file

now run command like

az deployment sub create -l eastus -f <file>.bicep -p <param>.json

e.g.

az deployment sub create -l eastus -f .\rg.bicep -p .\rg.parameters.json

**Octopus Tool**

Deployment tool – it can do parallel deployment and can spin n number of vms

Larger number of servers, with different complexity in parallel

Server (octopus deploy)- client (tenant – a target to deploy a server)

Server- windows

Tenants – linux, windows

Graphical user interface

Description automatically generated

Window – t2.medium (AWS) or equivalent in azure

Download octopus trial version (free for 30 days)

Download sql server express (free edition)

**Workflow in octopus**

**Tell octopus where to deploy your software**

* Create your first environment
* Create your first deployment target

Go to infra tab > add environment (say dev, int etc)

Deployment targets > Azure vm, Azure webapp etc (Listening tentacles, polling tentatcles)

**Package and upload your software**

Upload your package or add an external feed

**Define your deployment process**

Create your first project

Define its deployment process

**Deploy your release**

Create a release

Deploy a release

pip install -r requirements.txt -t .

**Github actions**

Workflow – repeatable process that is defined in a file – set of steps

Workflow yaml file- code file

.github/workflows

Runners/git-hub hosted runners/hosted runners – computers that configured to run deployment steps for a workflow – have biceps/azure tools installed

When workflow runs, hosted runner is automatically created. You can create self hosted runners

Jobs – set of steps

Steps type

Run – run a single command or sequence of commands in bash, powershell or window command shell

Action -convenient way to access many capabilities without writing script

Debugging

Graphical user interface, text, application, chat or text message

Description automatically generated

git push origin v1.11 (push code along with tags)

git push --tags (push tags only)

1

I think many people believe that tags mean that multiple commits can have the same tag (how tags usually work). Like adding a tag to make a certain type of commit (ex: prod?) Well, in git, a tag is unique to a commit.

However, if you really want to use a tag you already used before you will need to delete it from remote and local and recreate it. But you should not want to do this...

//remove remote tag

git push origin :tag\_name

//remove tag from local

git tag -d tag\_name

//add tag to latest commit

git tag tag\_name

git log

git push origin tag\_name

**Repository Disptach**

In order to do this you need to set following variables in postman

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Pwd

/home/runner/work/azure-biceps/azure-biceps

$HOME/work/repo-name/repo-name/

Home

/home/runner

Pwd

eew

Graphical user interface, text, application, email

Description automatically generated