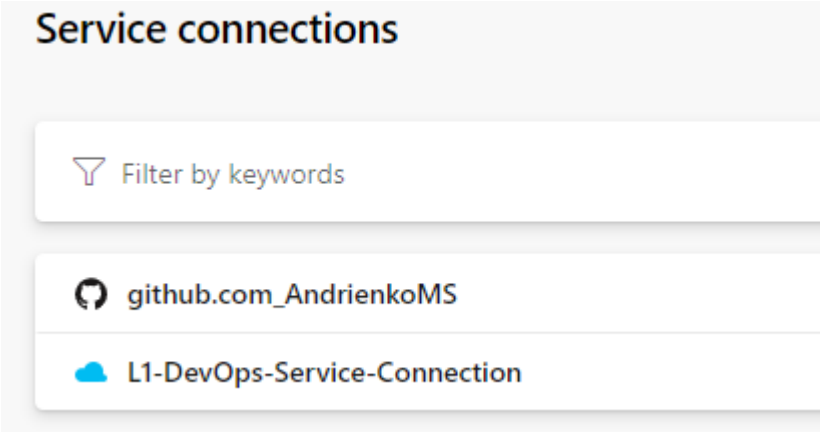
Prerequisites

1. Create azure subscription
2. Create azure devops organization
3. Read information about github flow branching strategy
4. terraform should be installed
5. Terraform knowledge is also required to do the stuff
6. Az cli should be installed

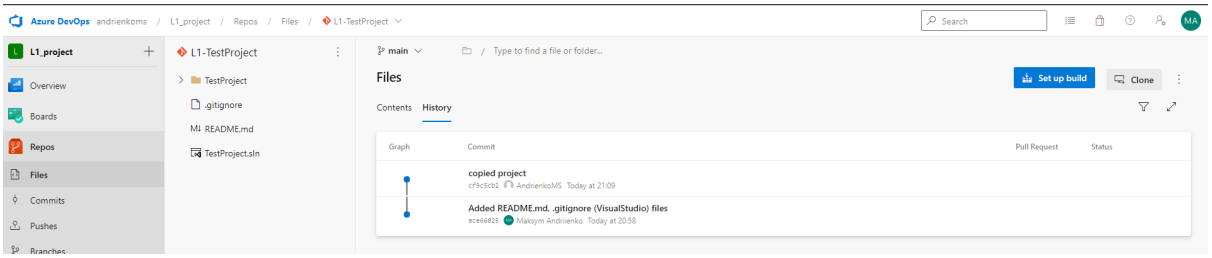
Homework

Part 1 – Configure application

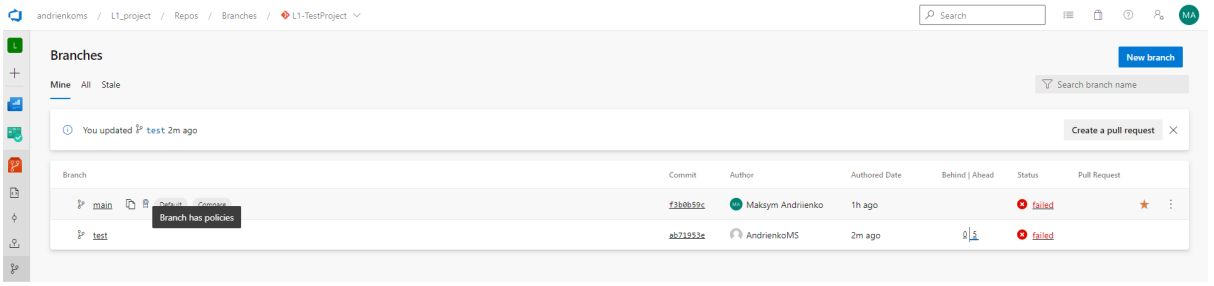
1. Create a service connection in a Azure DevOps project to your subscription - https://learn.microsoft.com/en-us/azure/devops/pipelines/library/service-endpoints?view=azure-devops&tabs=yaml

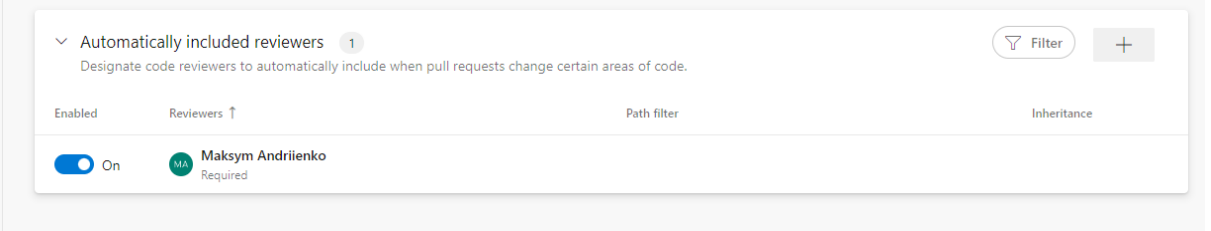


2. Find a .net pet project for the experiments
3. Build your app locally .net project via dotnet tool. dotnet restore/build/run
4. Create an Azure DevOps repo - https://learn.microsoft.com/en-us/azure/devops/repos/git/create-new-repo?view=azure-devops You can use import repository to import from existing source control version like github



5. Create a branching policy for you application. Added yourself as a reviewer - https://learn.microsoft.com/en-us/azure/devops/repos/git/branch-policies?view=azure-devops&tabs=browser As branching strategy use a github flow (It will be applied by default when you strict commit to your main branch)



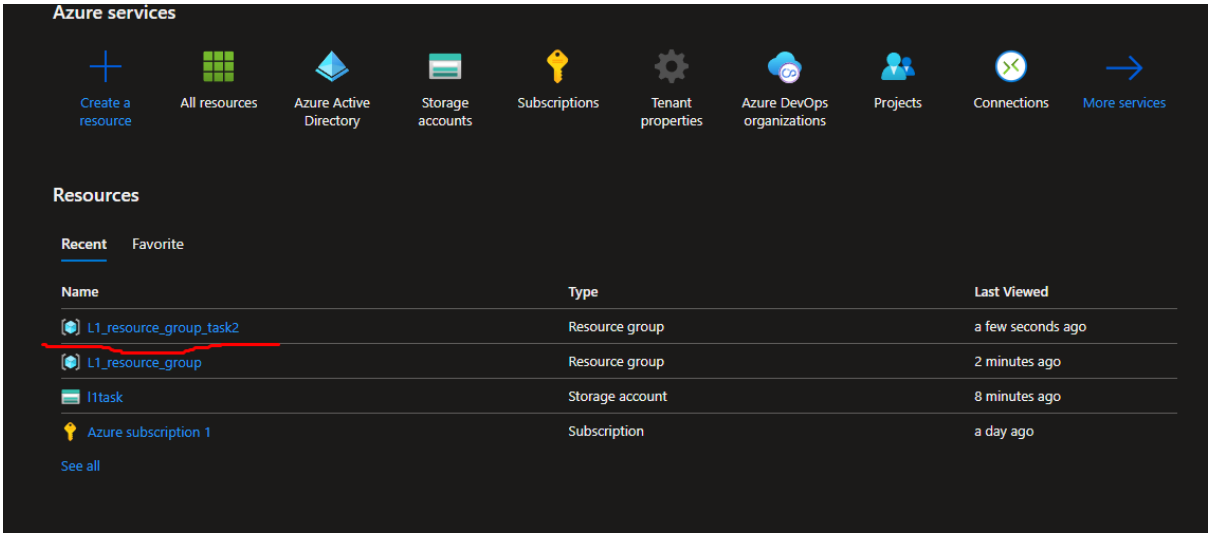


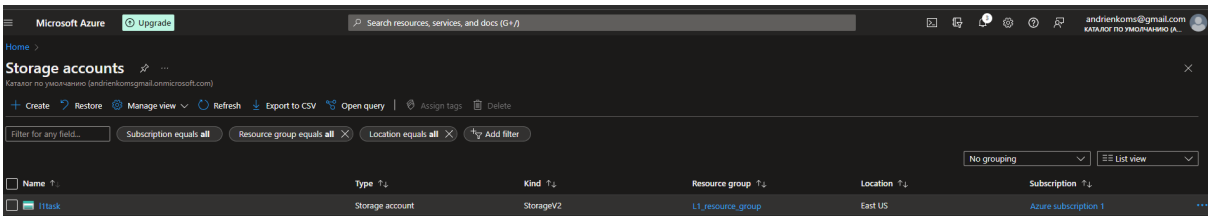
Part 2 – Configure a pipeline to deploy infrastructure

Below is describing on how to do it via terraform. If you want to use terraform you need to create service connection in manual way. Otherwise you won't be able to deploy your iac – Navigate to the last section

Terraform storage account

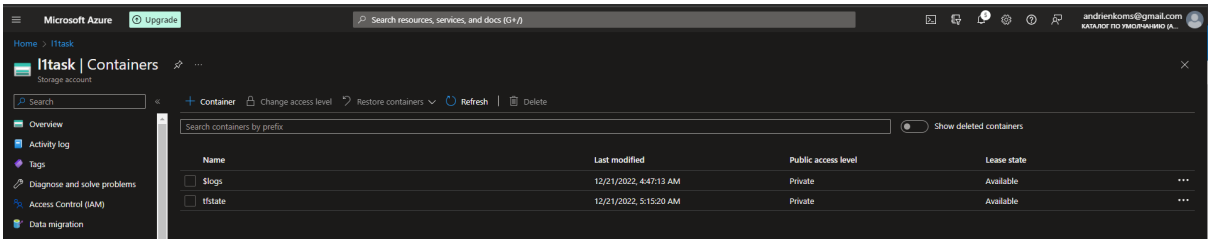
1. Create a separate resource group and deploy azure storage account - https://learn.microsoft.com/en-us/azure/storage/common/storage-account-create?tabs=azure-portal





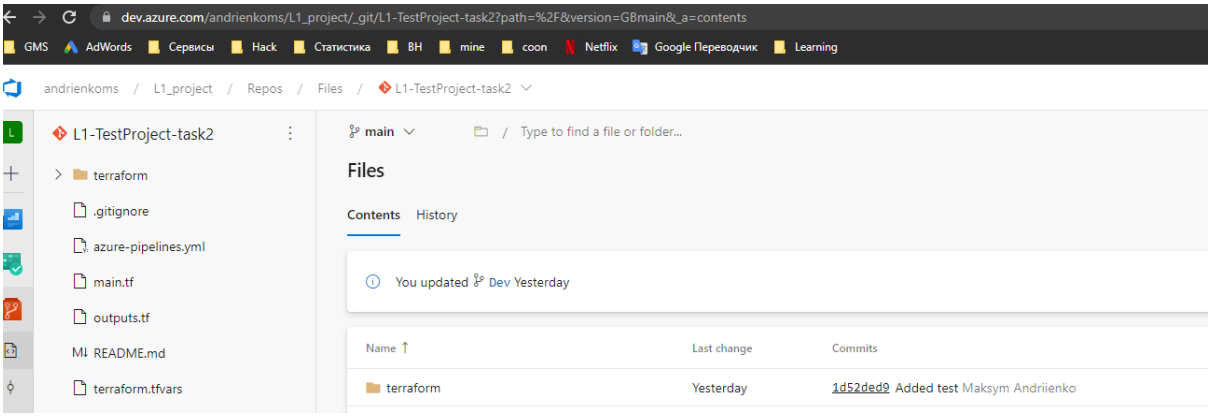
2. Create a container with the name "tfstate" and remember the name. use portal settings

In this storage account you will be store your tf state file



Terraform preparation

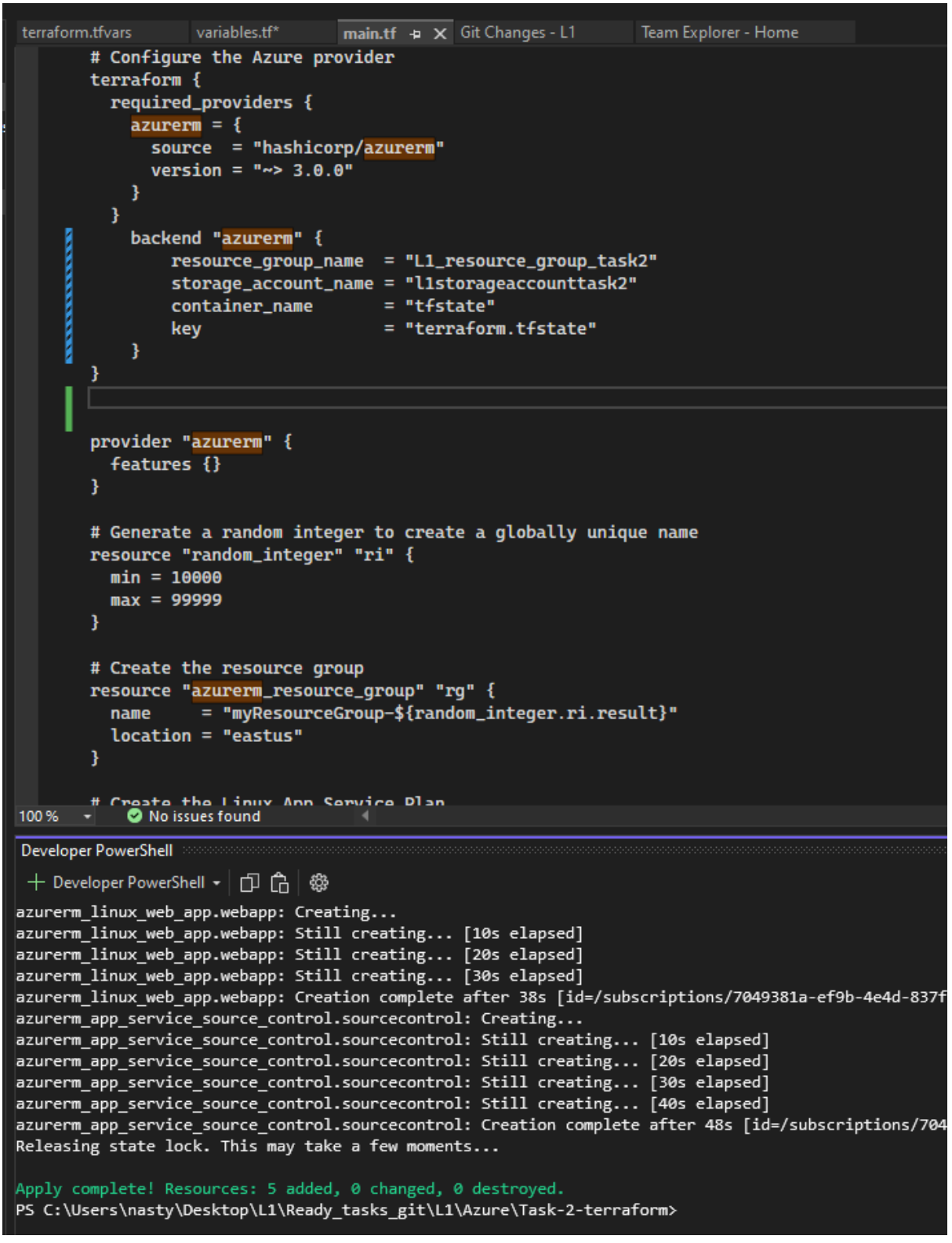
1. Create another repo to store devops code
2. Create a folder terraform



3. Add app service implementation - https://learn.microsoft.com/en-us/azure/app-service/provision-resource-terraform
4. Integrate application insights with app service
5. Updated backend “azurerm” according to the guide - https://learn.microsoft.com/en-us/azure/developer/terraform/store-state-in-azure-storage?tabs=azure-cli

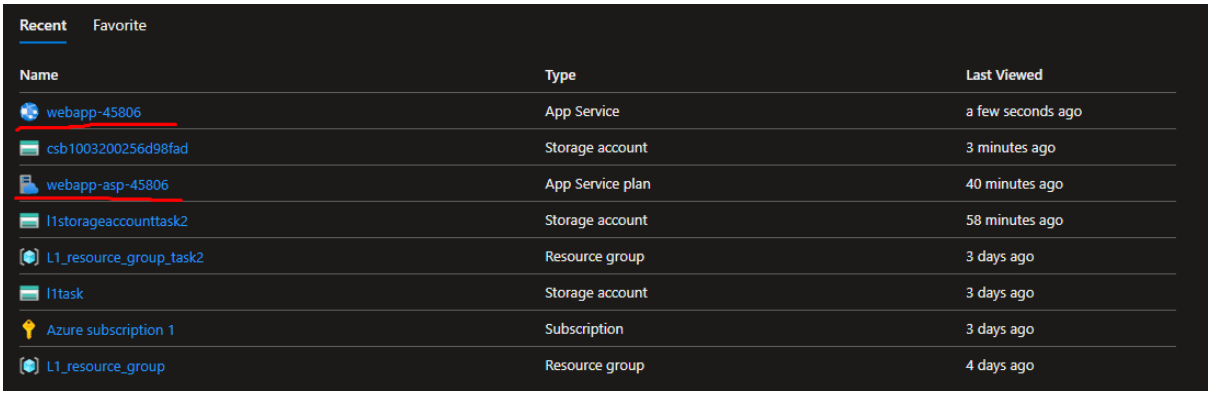
```
backend "azurerm" {
    resource_group_name  = "L1_resource_group_task2"
    storage_account_name = "l1storageaccounttask2"
    container_name       = "tfstate"
    key                  = "terraform.tfstate"
}
```

6. Run az login or Connect-AzAccount to connect the azure subscription from your loca


terraform.tfvars
variables.tf*
main.tf
Git Changes - L1
Team Explorer - Home
# Configure the Azure provider
terraform {
required_providers {
azurerm = {
source = "hashicorp/azurerm"
version = "~> 3.0.0"
}
}
backend "azurerm" {
resource_group_name = "L1_resource_group_task2"
storage_account_name = "l1storageaccounttask2"
container_name = "tfstate"
key = "terraform.tfstate"
}
}
provider "azurerm" {
features {}
}
# Generate a random integer to create a globally unique name
resource "random_integer" "ri" {
min = 10000
max = 99999
}
# Create the resource group
resource "azurerm_resource_group" "rg" {
name = "myResourceGroup-${random_integer.ri.result}"
location = "eastus"
}
100 %
No issues found
Developer PowerShell
Developer PowerShell
azurerm_linux_web_app.webapp: Creating...
azurerm_linux_web_app.webapp: Still creating... [10s elapsed]
azurerm_linux_web_app.webapp: Still creating... [20s elapsed]
azurerm_linux_web_app.webapp: Still creating... [30s elapsed]
azurerm_linux_web_app.webapp: Creation complete after 38s [id=/subscriptions/7049381a-ef9b-4e4d-837f
azurerm_app_service_source_control.sourcecontrol: Creating...
azurerm_app_service_source_control.sourcecontrol: Still creating... [10s elapsed]
azurerm_app_service_source_control.sourcecontrol: Still creating... [20s elapsed]
azurerm_app_service_source_control.sourcecontrol: Still creating... [30s elapsed]
azurerm_app_service_source_control.sourcecontrol: Still creating... [40s elapsed]
azurerm_app_service_source_control.sourcecontrol: Creation complete after 48s [id=/subscriptions/704
Releasing state lock. This may take a few moments...
Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
PS C:\Users\nasty\Desktop\L1\Ready_tasks_git\L1\Azure\Task-2-terraform>

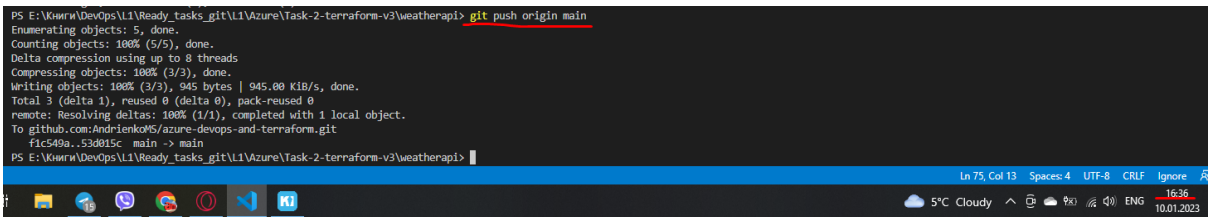
7. Run terraform apply to deploy infrastructure

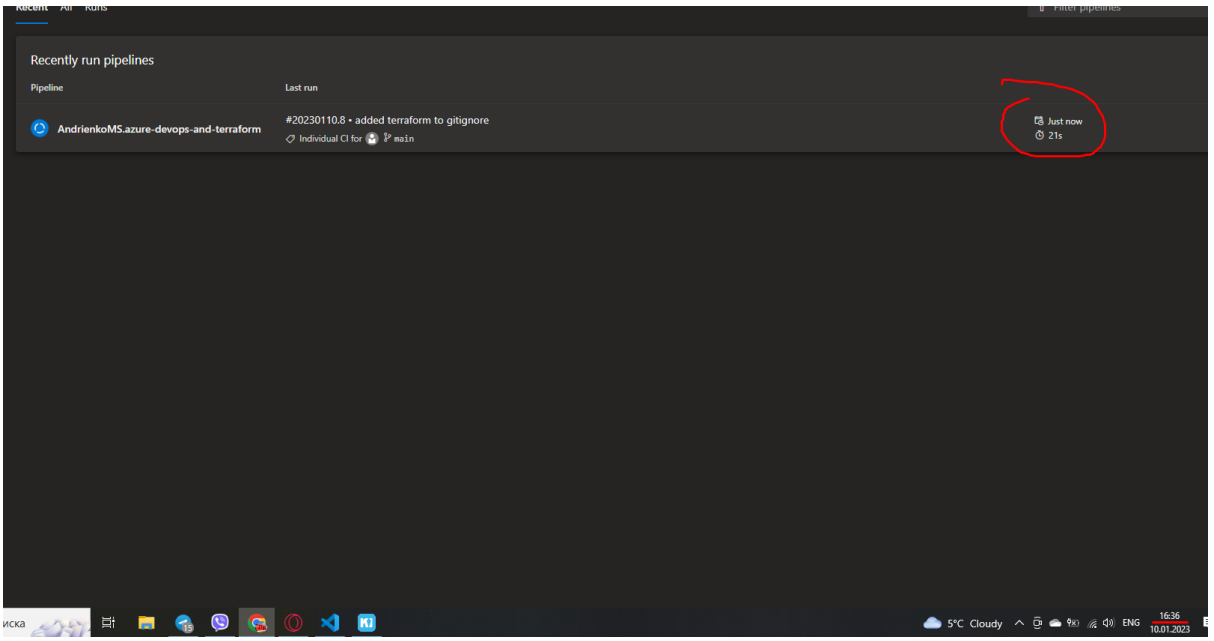
```
azurerm_resource_group.rg: Destruction complete after 1m20s
random_integer.ri: Destroying... [id=43340]
random_integer.ri: Destruction complete after 0s
Releasing state lock. This may take a few moments...

Destroy complete! Resources: 5 destroyed.
PS C:\Users\nasty\Desktop\L1\Ready_tasks_git\L1\Azure\Task-2-terraform>
```



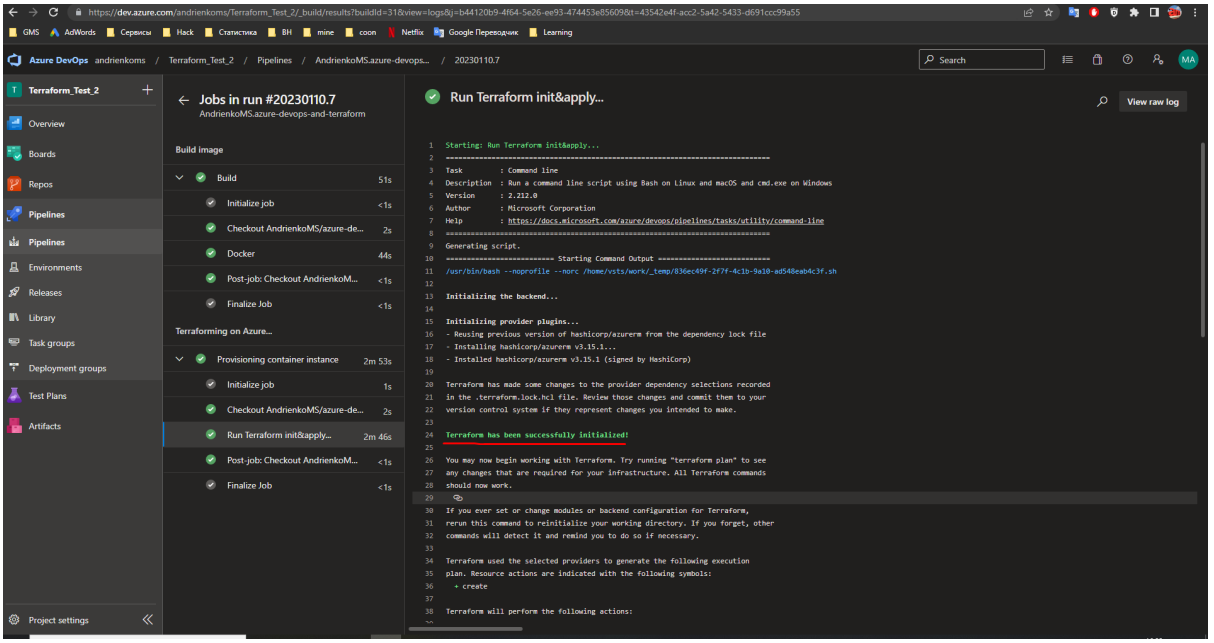
Create a terraform pipeline

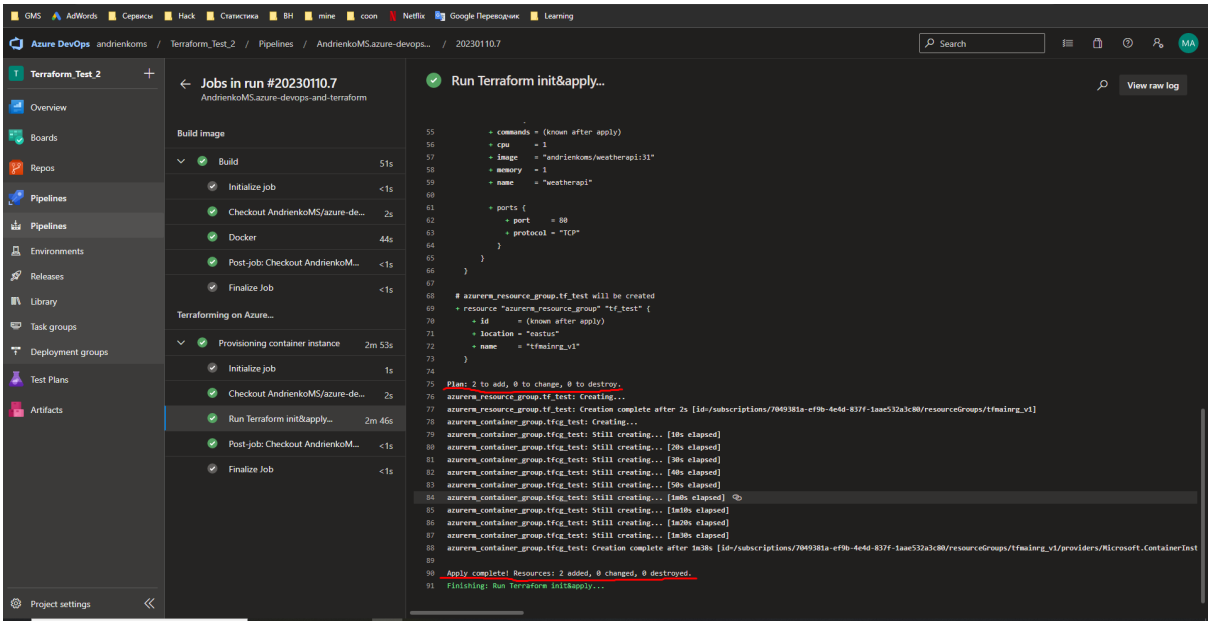
1. Create a yaml pipeline with the following steps: terraform install, terraform init, terraform plan/apply. Plan is an optional one
2. Inside yaml pipeline add trigger to main branch. The scenario – when main is updated, pipeline should run automatically - https://learn.microsoft.com/en-us/azure/devops/pipelines/yaml-schema/trigger?view=azure-pipelines





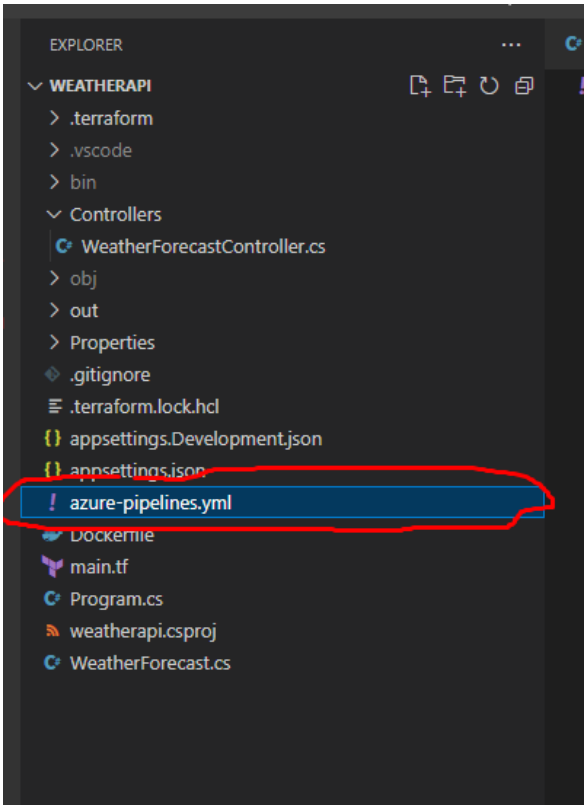
3. Added 3 steps – terraform install, terraform init, terraform plan/apply. Plan is an optional one. You may add it as 4th step





Part 3 – Create a deploy pipeline to app service

1. Add yml pipeline to the application folder

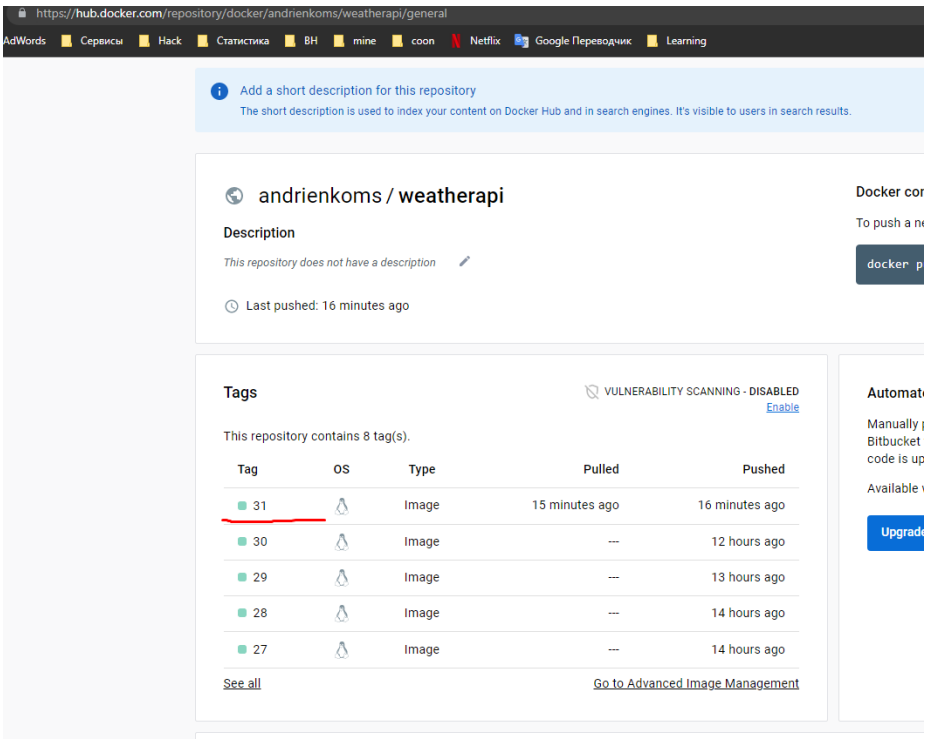


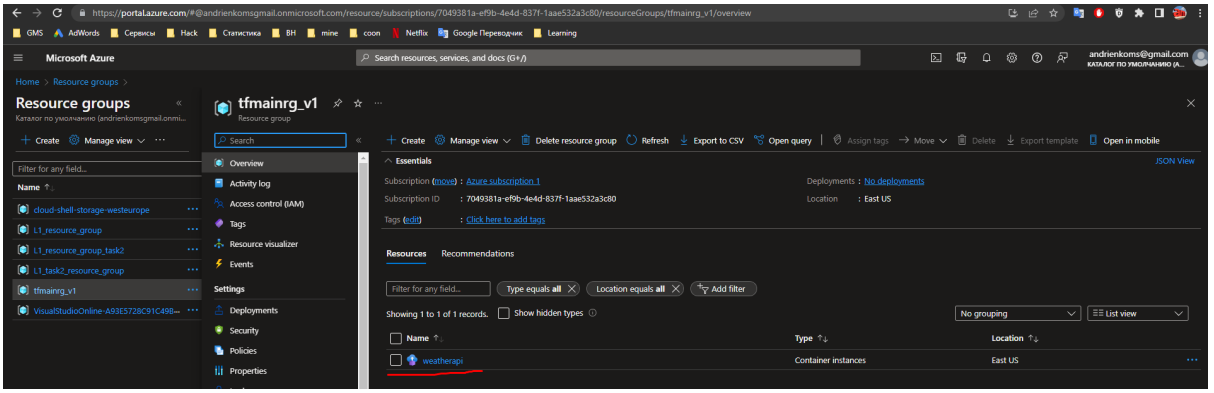
2. Your pipeline structure should contain 2 stages. 1st – build, create zip archieve, and publish an artifact. 2nd – download an artifact and deploy it to azure app service

```yaml
stages:
- stage: Build
  displayName: Build image
  jobs:
  - job: Build
    displayName: Build
    pool:
      vmImage: ubuntu-latest
    steps:
    - task: Docker@2
      inputs:
        containerRegistry: 'Andrienkoms Docker Hub'
        repository: 'andrienkoms/weatherapi'
        command: 'buildAndPush'
        Dockerfile: '**/Dockerfile'
        tags: |
          $(tag)
- stage: Provision
  displayName: 'Terraforming on Azure...'
  dependsOn: Build
  jobs:
  - job: Provision
    displayName: 'Provisioning container instance'
    pool:
      vmImage: ubuntu-latest
    variables:
    - group: Terrafrom_service_principle_vars
    steps:
    - script: |
        set -e
```

3. To deploy .zip to app service use azure app service deployment task

```
247  Removing intermediate container f9c575ee9f9d
248   ---> ad4185e8181f
249  Step 22/22 : LABEL image.base.ref.name=mcr.microsoft.com/dotnet/sdk:7.0
250   ---> Running in c3eeec1acd68
251  Removing intermediate container c3eeec1acd68
252   ---> 7fe14f2d64cf
253  Successfully built 7fe14f2d64cf
254  Successfully tagged ***/weatherapi:31
255  /usr/bin/docker images
256  /usr/bin/docker push ***/weatherapi:31
257  REPOSITORY                    TAG         IMAGE ID       CREATED          SIZE
258  ***/weatherapi         31          7fe14f2d64cf   1 second ago     772MB
259  <none>                        <none>      7d1822a98dc0   15 seconds ago   827MB
260  mcr.microsoft.com/dotnet/sdk  7.0         870663663d98   2 weeks ago      768MB
261  node                          14-alpine   b4fb2cece133   4 weeks ago      123MB
262  node                          16-alpine   bb97fd22e6f8   4 weeks ago      118MB
263  node                          18-alpine   6d7b7852bcd3   4 weeks ago      169MB
264  ubuntu                        22.04       6b7dfa7e8fdb   4 weeks ago      77.8MB
265  ubuntu                        20.04       d5447fc01ae6   4 weeks ago      72.8MB
266  ubuntu                        18.04       251b86c83674   4 weeks ago      63.1MB
267  node                          14          c08c80352dd3   5 weeks ago      915MB
268  node                          16          993a4cf9c1e8   5 weeks ago      910MB
269  node                          18          209311a7c0e2   5 weeks ago      991MB
270  buildpack-deps                buster      623b2dda3870   5 weeks ago      803MB
271  buildpack-deps                bullseye    8cbf14941d59   5 weeks ago      835MB
272  debian                        10          528ac3ebe420   5 weeks ago      114MB
273  debian                        11          291bf168077c   5 weeks ago      124MB
274  alpine                        3.16        bfe296a52501   8 weeks ago      5.54MB
275  moby/buildkit                 latest      383075513bdc   2 months ago     142MB
276  alpine                        3.14        dd53f409bf0b   5 months ago     5.6MB
277  alpine                        3.15        c4fc93816858   5 months ago     5.58MB
278  The push refers to repository [docker.io/***/weatherapi]
```


https://hub.docker.com/repository/docker/andrienkoms/weatherapi/general
AdWords Сервисы Hack Статистика BH mine coon Netflix Google Переводчик Learning
Add a short description for this repository
The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.
andrienkoms / weatherapi
Description
This repository does not have a description
Last pushed: 16 minutes ago
Docker cor
To push a ne
docker p
Tags
VULNERABILITY SCANNING - DISABLED
Enable
This repository contains 8 tag(s).
Tag OS Type Pulled Pushed
31 Image 15 minutes ago 16 minutes ago
30 Image --- 12 hours ago
29 Image --- 13 hours ago
28 Image --- 14 hours ago
27 Image --- 14 hours ago
See all
Go to Advanced Image Management
Automat
Manually
Bitbucket
code is up
Available
Upgrade


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Resource groups
Resource groups
Create Manage view
Filter for any field...
Name
cloud-shell-storage-westeurope
L1_resource_group
L1_resource_group_task2
L1_task2_resource_group
tfmainrg_v1
VisualStudioOnline-A93E5728C91C498...
tfmainrg_v1
Resource group
Search
Overview
Activity log
Access control (IAM)
Tags
Resource visualizer
Events
Settings
Deployments
Security
Policies
Properties
Create Manage view Delete resource group Refresh Export to CSV Open query Assign tags Move Delete Export template Open in mobile
Essentials
JSON View
Subscription (move) : Azure subscription 1
Subscription ID : 7049381a-ef9b-4e4d-837f-1aae532a3c80
Tags (edit) : Click here to add tags
Deployments : No deployments
Location : East US
Resources Recommendations
Filter for any field...
Type equals all
Location equals all
Add filter
Showing 1 to 1 of 1 records.
Show hidden types
No grouping
List view
Name
Type
Location
weatherapi
Container instances
East US