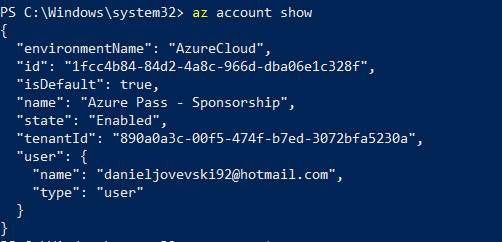
Task 1: Install terraform and Azure CLI

1. Use official guidelines to install the latest version of terraform and Azure CLI

2. Authenticate with Azure CLI

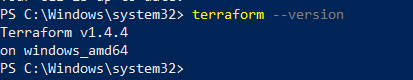


3. Set the exercise subscription as default for Azure CLI

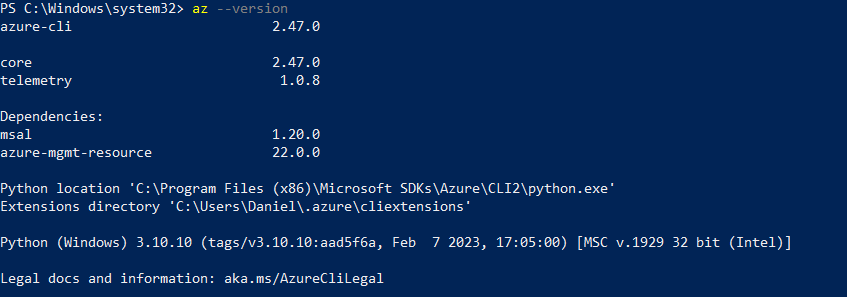
4. Provide console print screen:

4.1 Time and date when the exercise was worked

4.2 Output of the terraform command that will print out the Terraform version installed

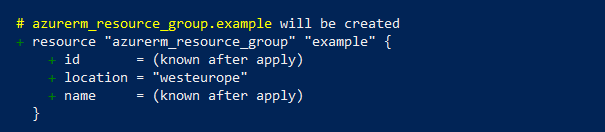


4.3 Azure CLI output of the current subscription

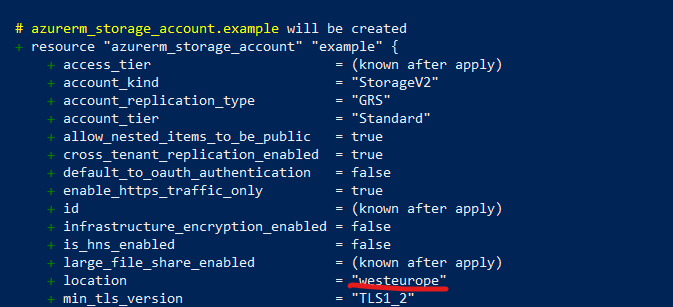


Task 2: Define your first terraform infrastructure code

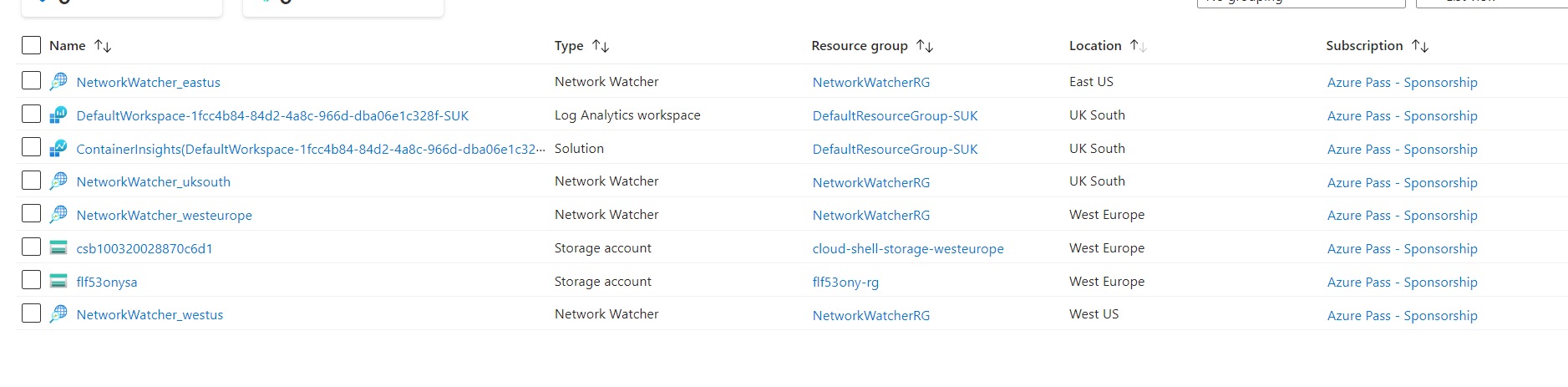
* 2.7 Check the results from your terraform plan now. There should be no issues and it will show what it will manage. Take the time to answer the following questions:
* 2.7.1 How many resources have you defined in your code and how many resources does the plan output show? Are they the same and why?
* 3 resources. They are the same because we specified them in main.tf
* 2.7.2 What is the location of your resource group and what is the location of the storage account?
* Resource group location: westeurope

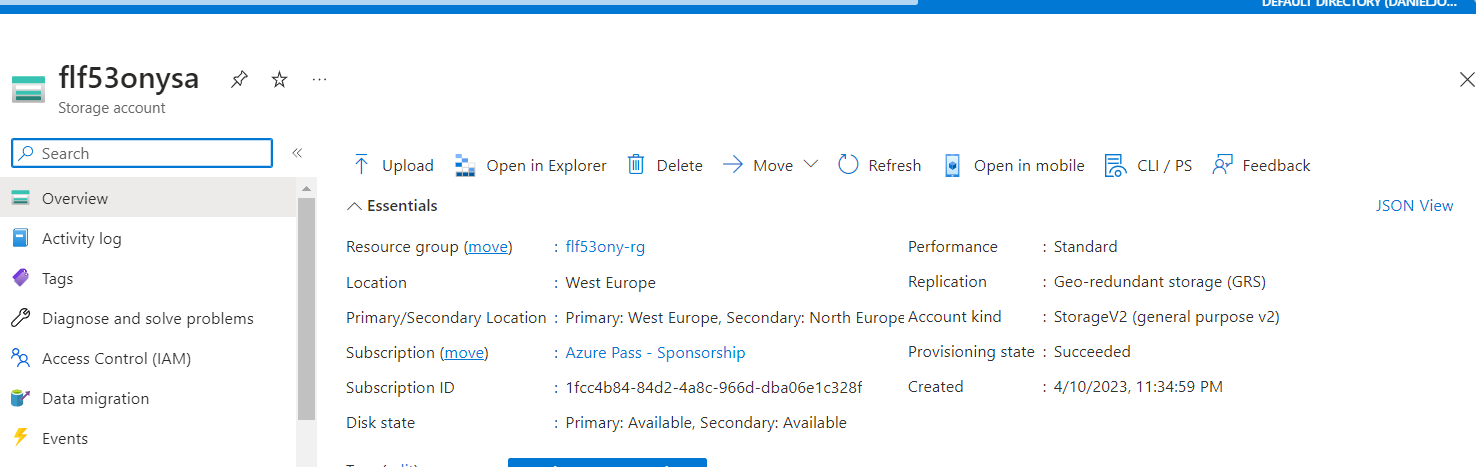


* Storage account location : westerope



* 2.8 Deploy your code to on the subscription and answer the questions bellow:
* 2.8.1 How many resources do you have on your subscription? (To list all resources, type “All resources” in the search bar on the top in Azure Portal)
* In All resources it only shows newly create storage account
* 2.8.2 Are the number of resources shown in the All resources portal window the same with the ones from your plan?
* No, it only shows the storage account
* 2.8.3 Give short explanation about the resources that are not shown?
* The resource group is created and it is shown in the resourcegroup menu
* The **random\_string.random** resource block defines a resource that generates a random string using a cryptographic random generator. The generated string can be used for various purposes in your Terraform configuration, such as creating unique names for resources that need to be created with unique identifiers. The **random\_string** resource created by Terraform cannot be found in the Azure portal.
* The **random\_string** resource is a Terraform-specific construct and is used to generate a random string within your Terraform configuration. This resource is not created within Azure itself and does not appear in the Azure portal.
* 2.8.4 Provide print screen of your portal with all resources.





4.2. In this task we will need to go over our terraform plan and identify the reasons why our resources are being replaced.

4.2.1. Search for the term “forces replacement” and node the resource name and the parameter that forces replacement. Describe the reason behind it

In Terraform, "forces replacement" refers to a situation where a resource must be destroyed and recreated in order to apply changes to its configuration. This occurs when a configuration change is made that cannot be applied to the existing resource without destroying it first. In such cases, Terraform will display a warning message indicating that the resource will be replaced.

The reason behind forcing a replacement is to ensure that the resource is in the desired state after the configuration change is applied. In some cases, making certain changes to a resource's configuration may not be possible without destroying and recreating it. By forcing a replacement, Terraform ensures that the configuration change is applied correctly and that the resource is in the desired state.

Task 3: Using variables and outputs

1.6.1. You are seeing that the code is asking you to insert an input value. Type your first name in lowercase and press enter.

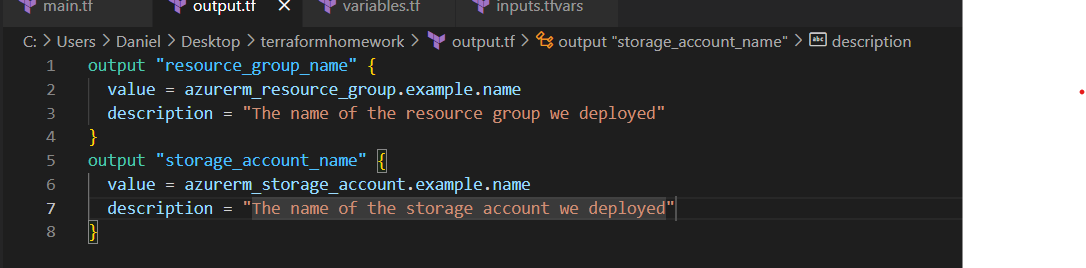
1.6.2. Please answer the following questions:

• How many variables do we have defined, and which are they

* We defined two variables one for resourcegroup name that we need to imput, the other one for location that we defined to be “West Europe”

• Why did terraform asked us to input a value only for the my\_name variable?

* Because we didn’t specify default value and we need to enter

3.2. Inside the outputs.tf file define an output value named resource\_group\_name with the value of the name of the resource group that we create, like shown below:   
3.3. Do the same for the output value named storage\_account\_name where the value will be the name of the storage account by using the example from step 3.2   
3.4. Execute the terraform plan with the input variable file switch. It should show you again 2 resources for destroy and 2 resources to create. You will also see at the bottom that there will be outputs.   
