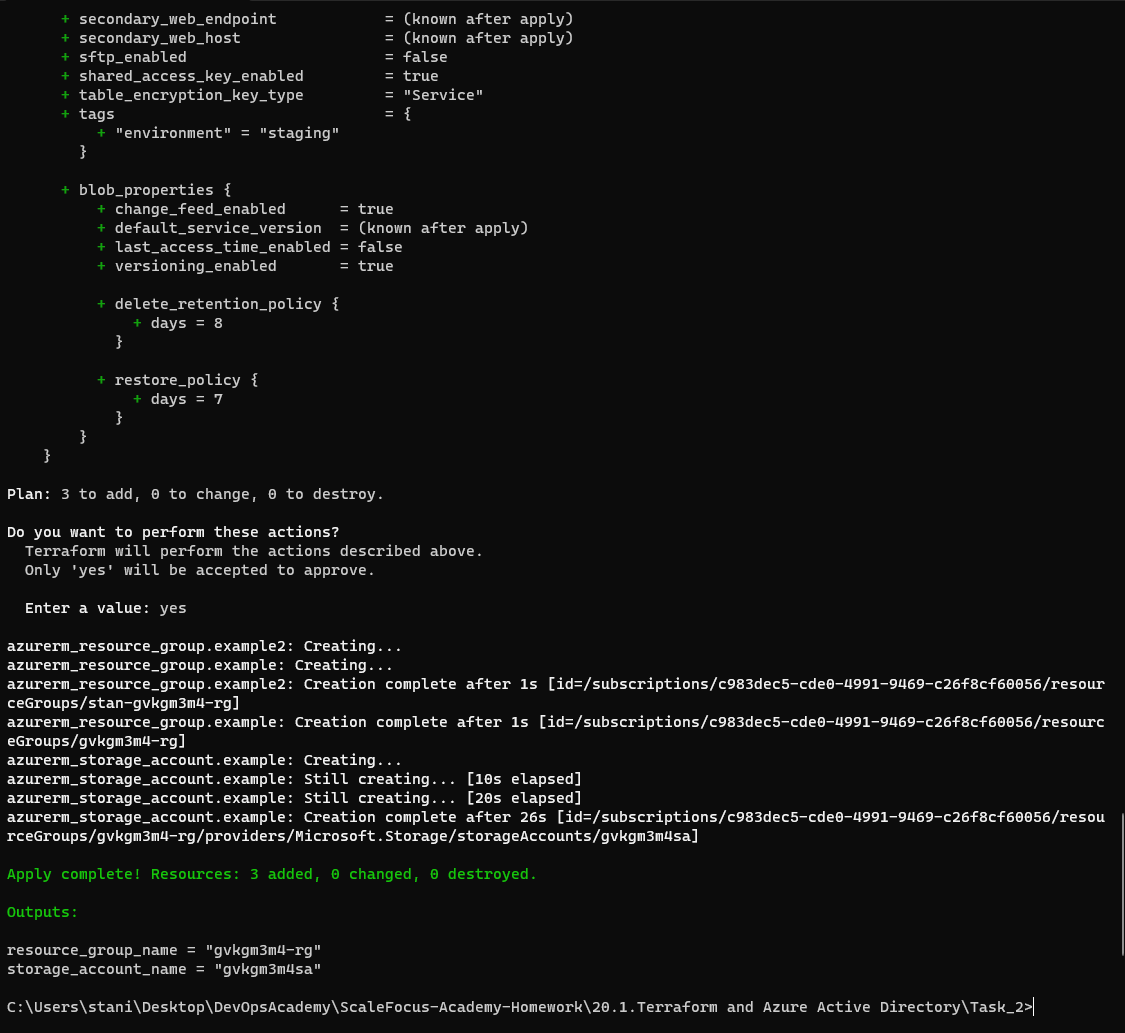
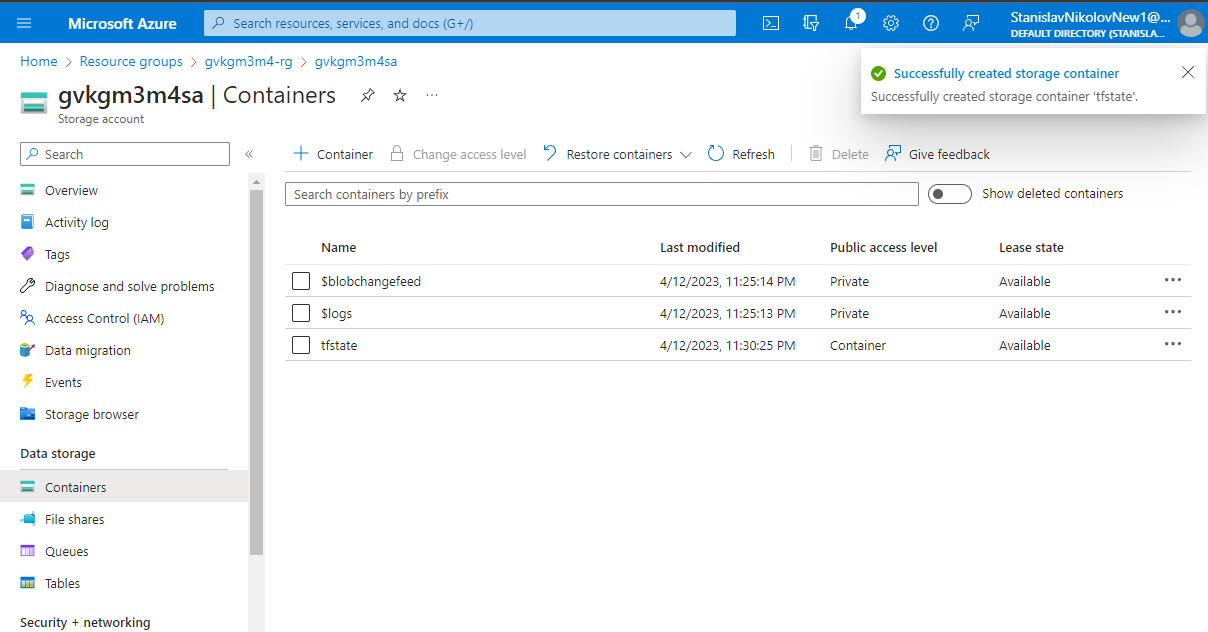
**Terraform modules**

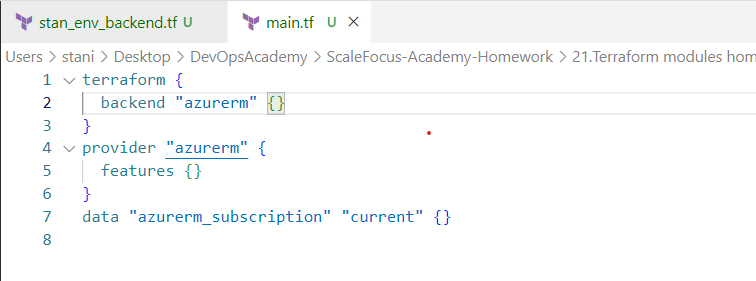
**Task 1: Setup and use remote terraform backend.**



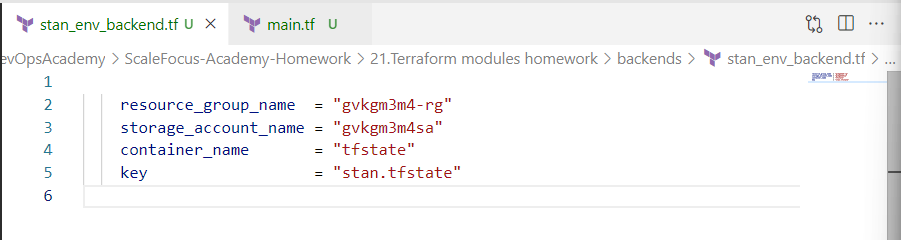
* Reapplying my terraform configurational file from the previous lab, because I destroyed the resources.



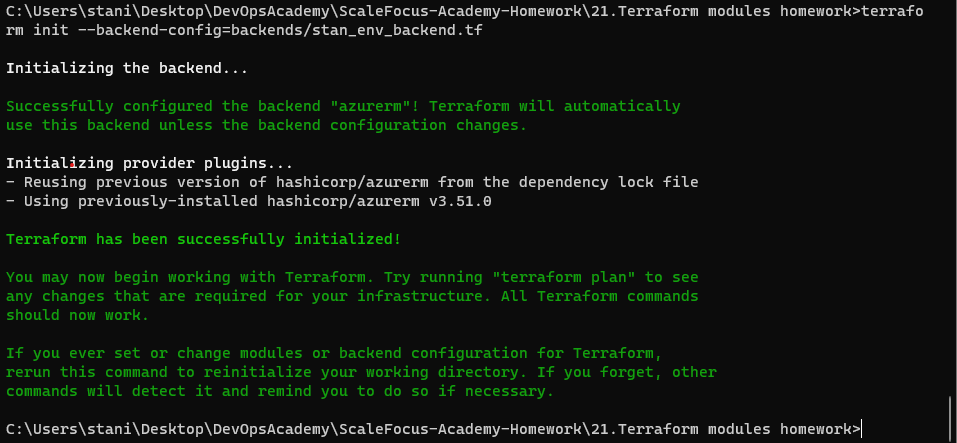
* Going to Azure portal and creating a container in the storage account named “tfstate”, where we are uploading an empty file, named “stan.tfstate”.



* Here we are adding the backend configuration file called “main.tf”.
* We also have another file, called “stan\_env\_backend.tf”, where we have the variables for the resource group, storage account, container name and key, which is the empty file (stan.tfstate), that I uploaded.

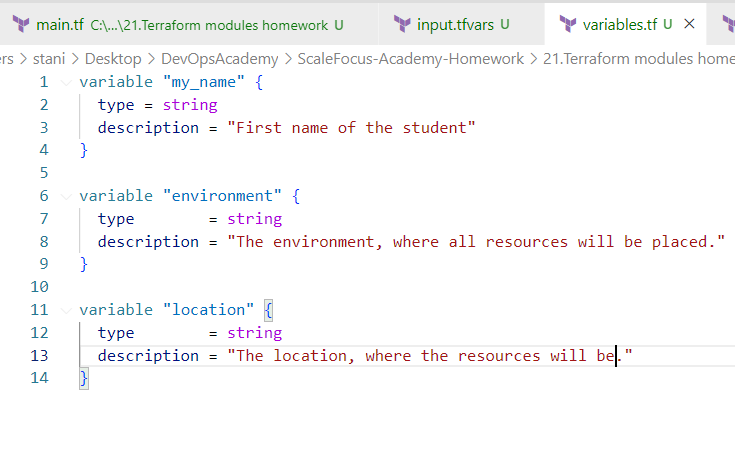


* As I said, here are the values for the backend configuration.
* Also worth mentioning that this file is in subdirectory “backends”, because we can have multiple modules for backend.

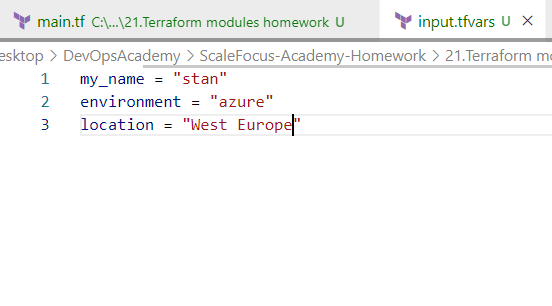


* terraform init --backend-config=backends/\_env\_backend.tf
* With this we have finalized our remote backend setup and we can define different backends and switch between them using the command option --backend-config.

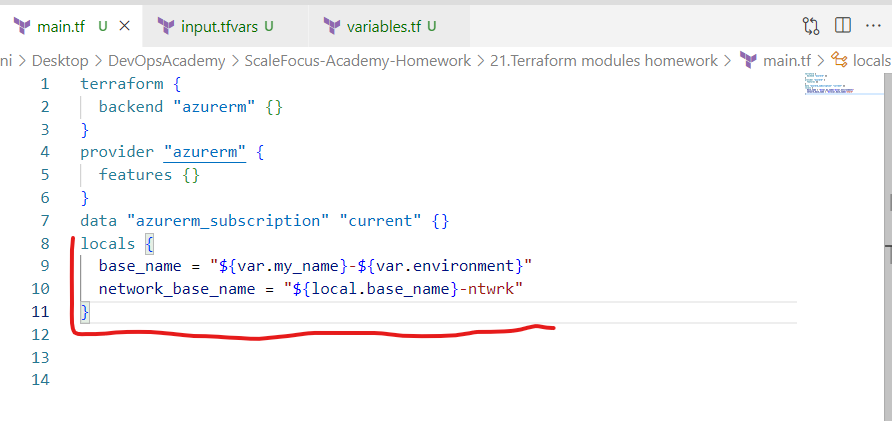
**Task 2: Define the network resources from your second midterm assignment.**



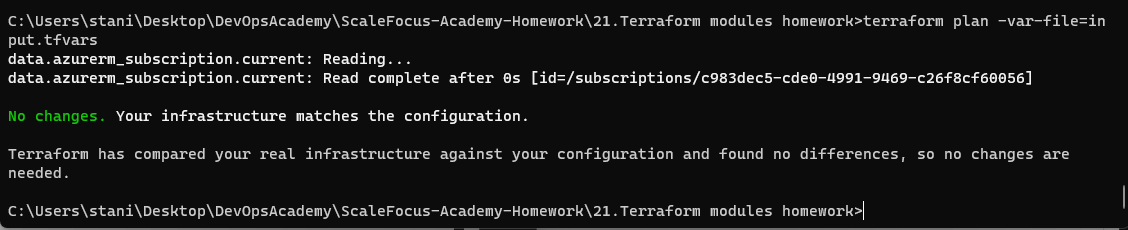
* Creating a variable file (variable.tf), in which I am declaring 3 variables for “location”, “environment” and “my\_name”.



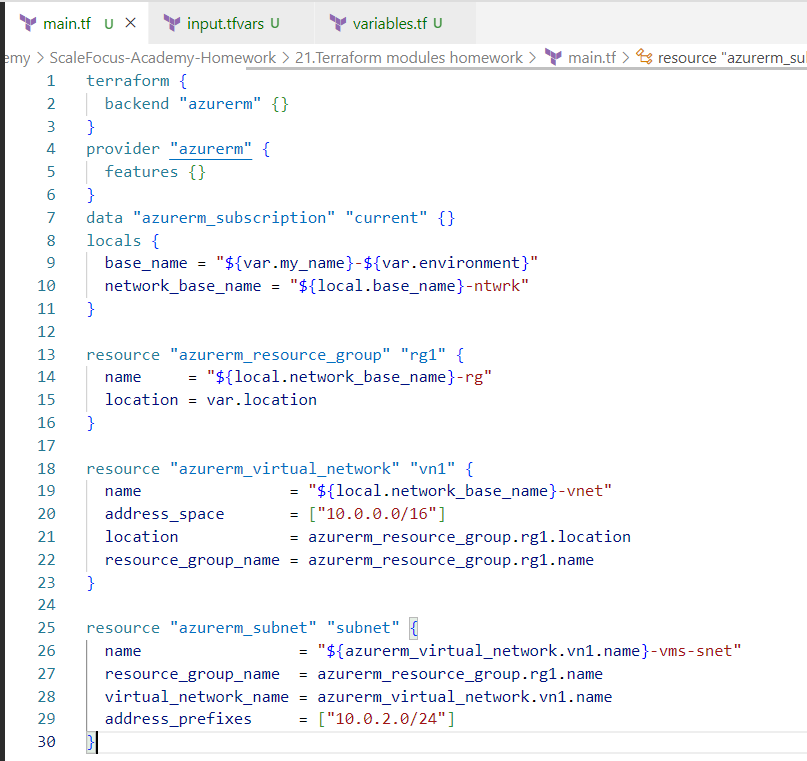
* We also need a file (input.tfvars), in which we are defining the values of the variables, so that we will build a skeleton, which we can use multiple times.



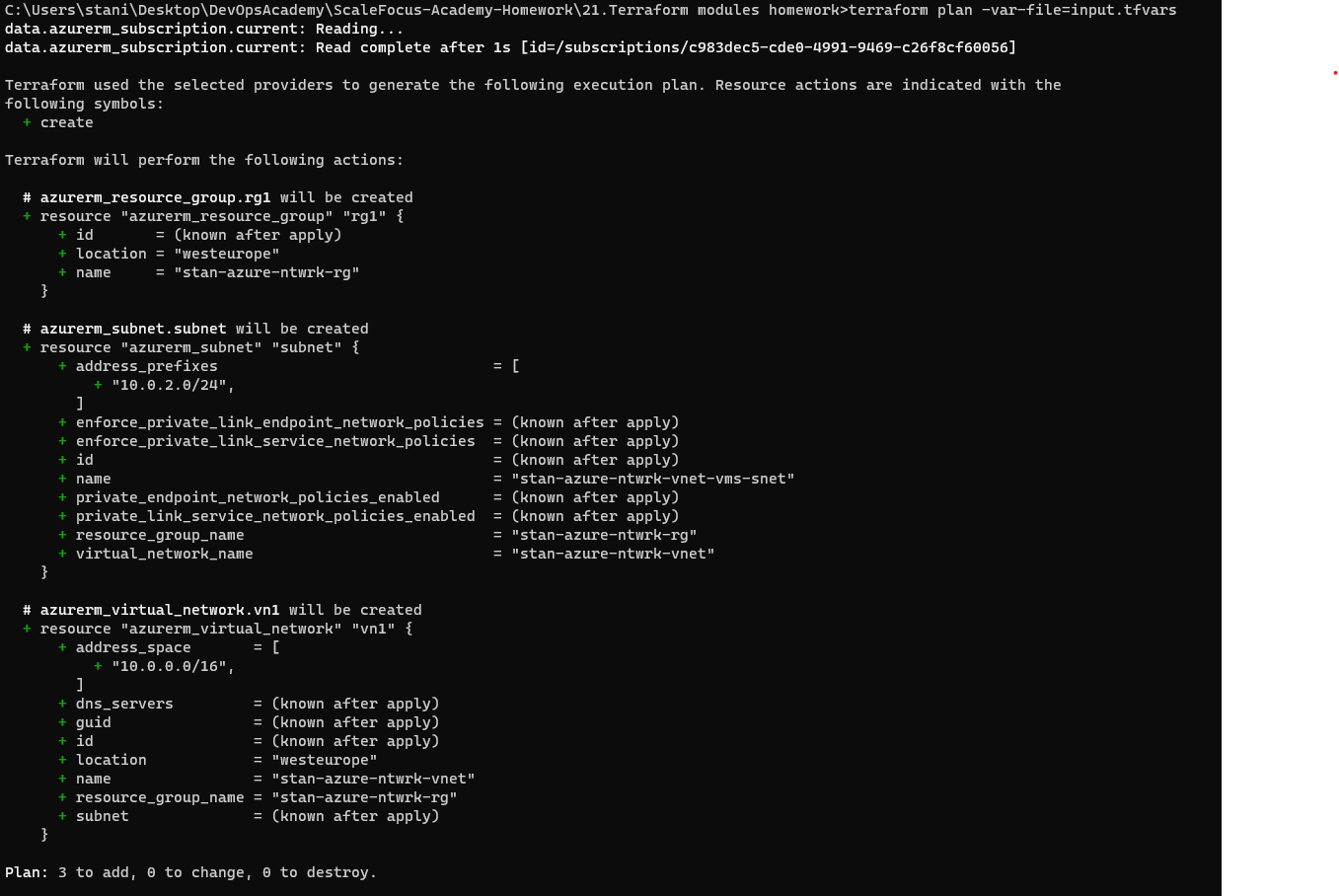
* Defining a locals block, in which we have some variables for “base\_name” and “network\_base\_name”.



* terraform plan -var-file=input.tfvars
* As expected, we shouldn’t have any errors, and the output should be like this.
* We are executing here a “terraform plan” with a specific “var-file” flag for the variables definitions.



* In the end of the second task: How the configurational file looks like:
* I created a resource group “rg1”, virtual network “vn1” and subnet “subnet”.



* Execute “terraform plan” with the input from my “input.tfvars” file, here we can see everything that will be created.
* Our plan should not throw any errors.

**Task 3: Define and group the virtual machine and its resources into a module.**

****

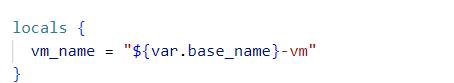
* As we can see here, we are creating in a subdirectory called “vm\_module”:
* Resource group – the group where the virtual machine will be placed. We will use this resource group for the resto of the VM components also, so we can have clear understanding what belongs and where.
* Public IP – this is not related directly to our virtual machine but to the network interface that is used by the virtual machine.
* Network interface – we need to define a network interface before we create a virtual machine.

****

* Network security group (NSG) – which will be configured for management and service public access by the virtual machine
* Assignment of the NSG to the network interface – this is a separate resource in terraform because of the API functionality of the cloud provider

****

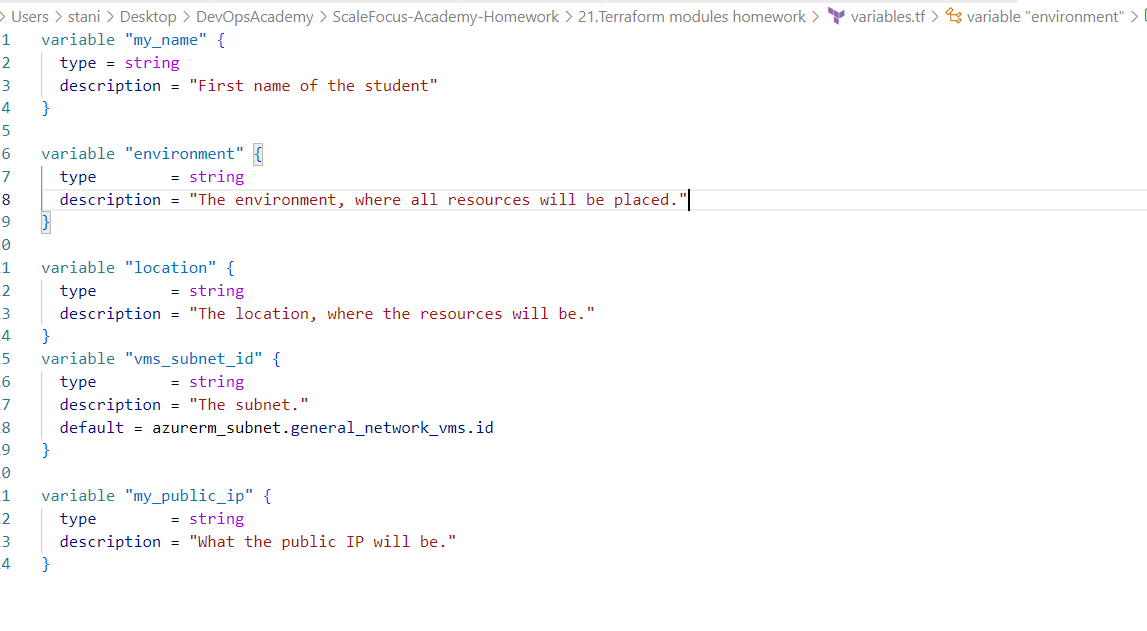
* And finally the Linux virtual machine.

****

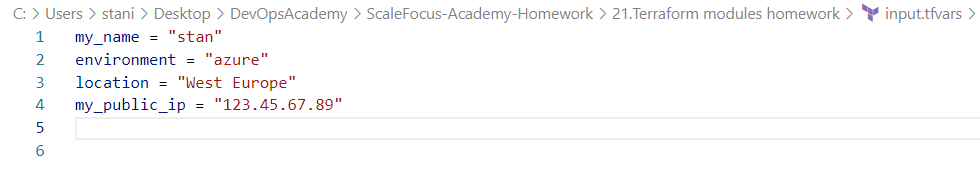
* Also in a local block we are adding local variable called vm\_name.

****

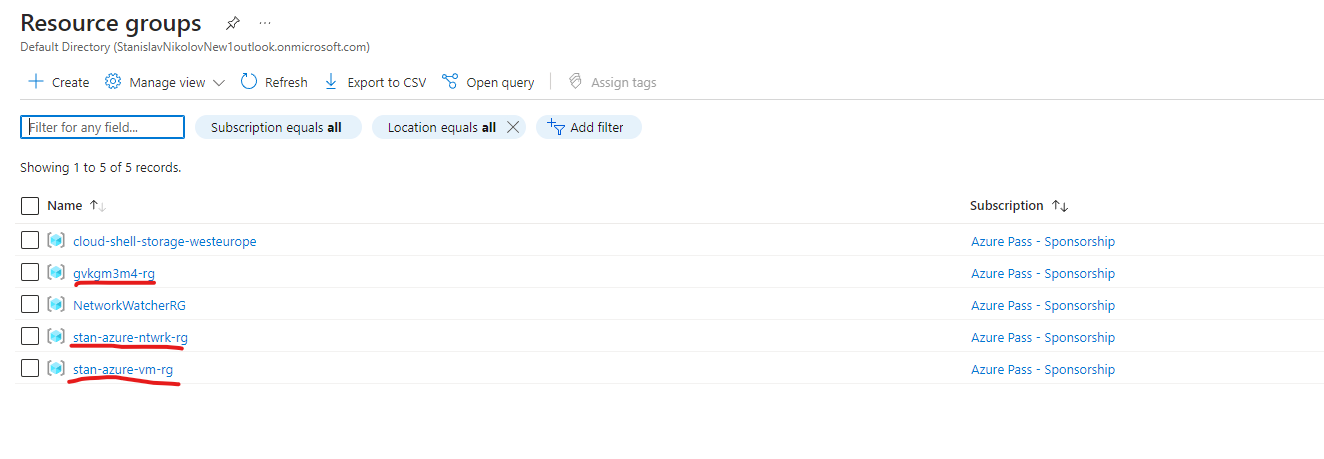
* Now we are back to the network configuration file, where we are using the module “vm”, that we created.

****

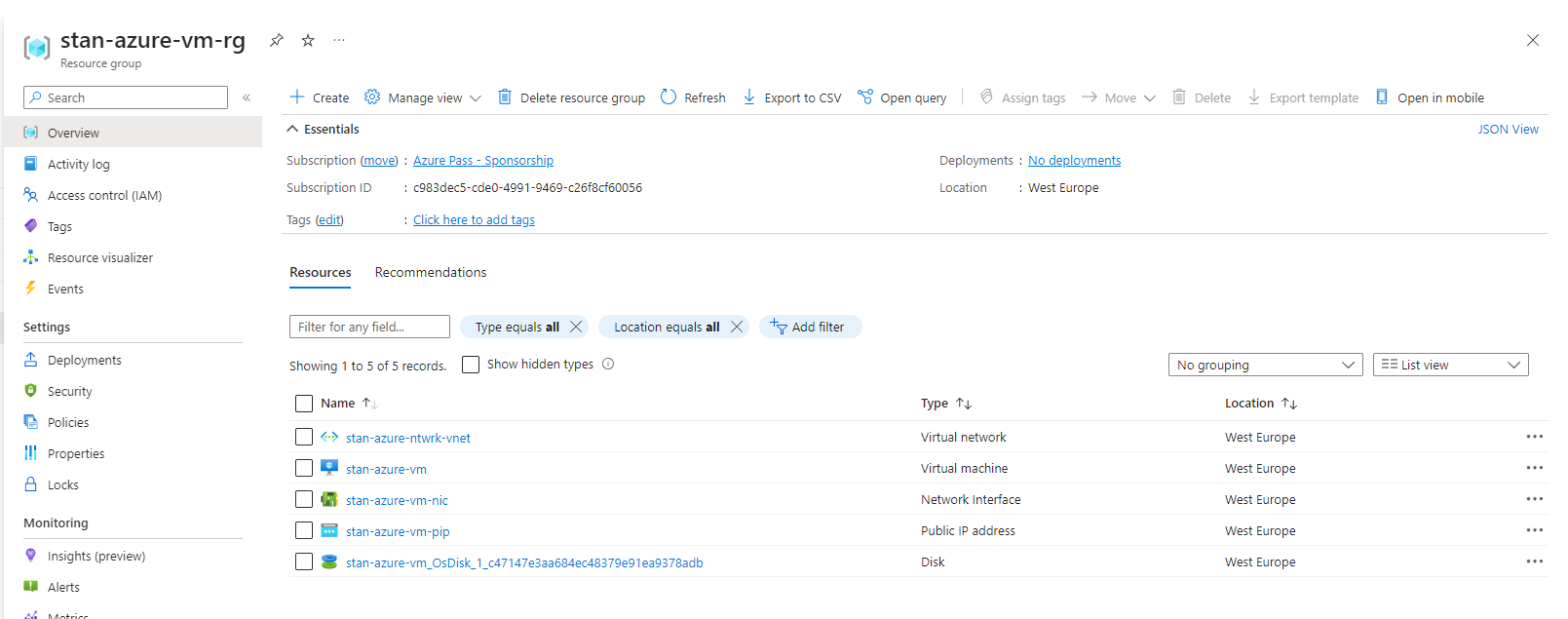
* The variable file should look like this, I didn’t use a variable for the password, and just hardcoded the password.

****

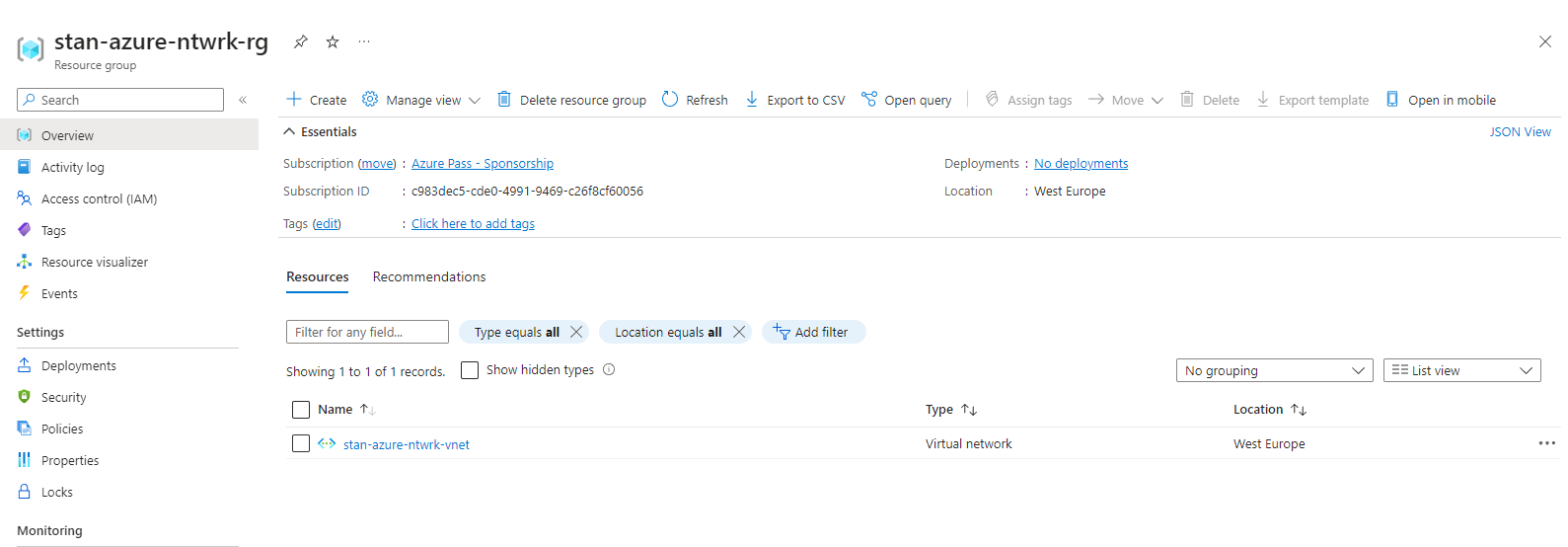
* And that’s how the input.tfvars should look like.

****

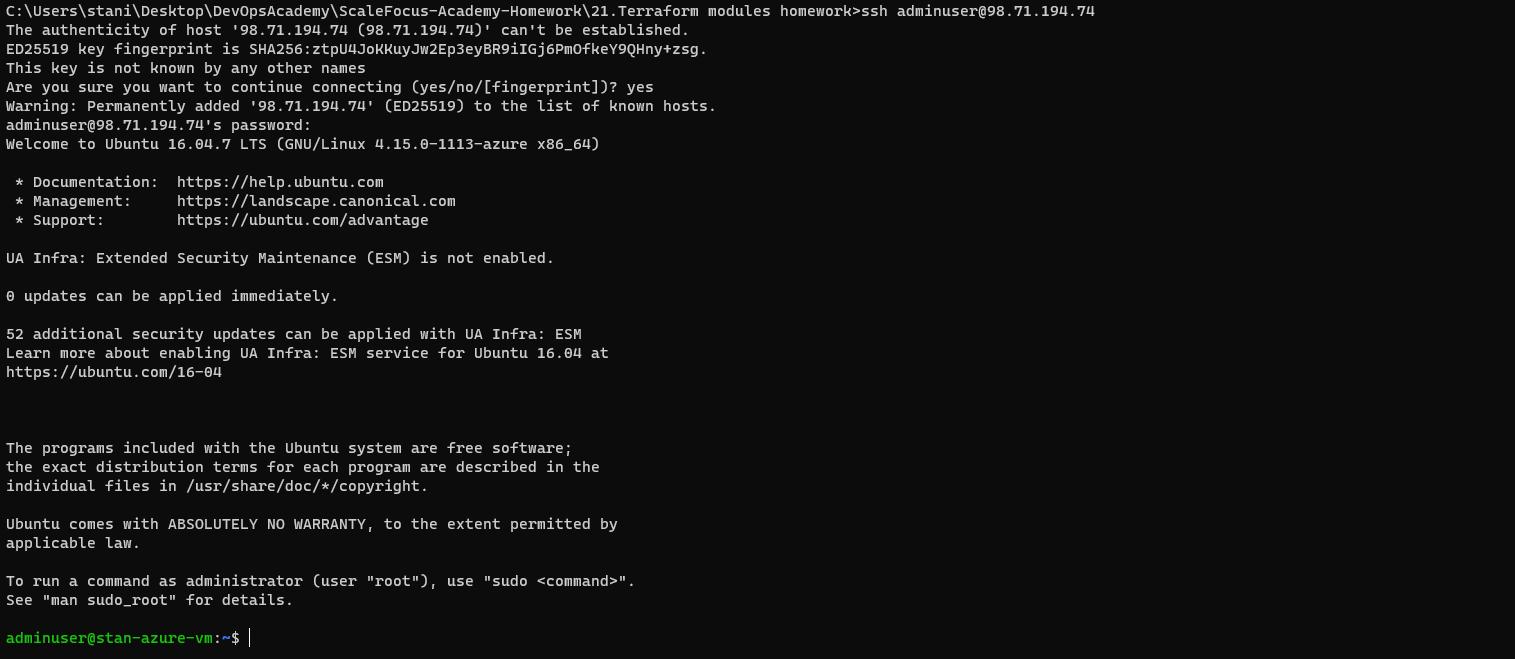
* After terraform apply, we should have 2 resources, and the first resource was for the first task.



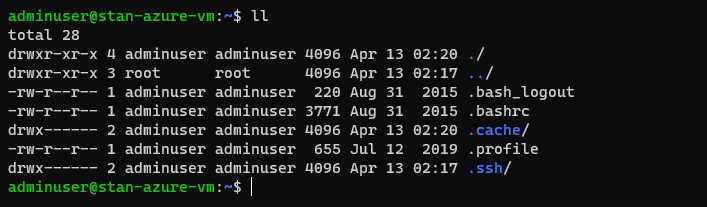
* The resource group of the virtual machine.



* The resource group of the network.



* Connecting to the virtual machine, that we just created.



* Listing the files of the virtual machine.