

Azure & Terraform syscheck

This is intended to check for a working local environment. Since login and connect to Azure are somewhat complicated. Please do:

1. Login to Azure with your browser. CLI login will not work without this!
2. Do a CLI login into Azure by calling *login.tf*
3. If you do not have *login.tf*, please install it according to *004_Configuration.pdf* in Documentation
4. Check Azure connection by issuing *az group list | grep name*
5. Login to GitHub by calling *gitlogin.sh*
6. If you do not have *gitlogin.sh*, please install it according to *004_Configuration.pdf* in Documentation. It should succeed with something like:

```
$ gitlogin.sh  
Hi <user>! You've successfully authenticated, but GitHub does not provide shell access.
```

7. Browse to <https://docs.microsoft.com/en-us/azure/developer/terraform/create-resource-group?tabs=azure-cli>

8. Download *main.tf* and *variables.tf* into directory *001_SystemCheck*

```
# This is main.tf

terraform {
  required_version = ">=0.12"

  required_providers {
    azurerm = {
      source = "hashicorp/azurerm"
      version = "~>2.0"
    }
  }
}

provider "azurerm" {
  features {}
}

resource "random_pet" "rg-name" {
  prefix = var.resource_group_name_prefix
}

resource "azurerm_resource_group" "rg" {
  name       = random_pet.rg-name.id
  location   = var.resource_group_location
}
```

```
#This is variable.tf

variable "resource_group_name_prefix" {
  default     = "rg"

  description = "Prefix of the resource group name that's combined with a random ID
so name is unique in your Azure subscription."
}

variable "resource_group_location" {
  default = "eastus"

  description = "Location of the resource group."
}
```

9. Change location from “eastus” to “westeurope”

Next you execute terraform:

1. terraform init
2. terraform plan -out main.tfplan
3. terraform apply main.tfplan
4. az group list | fgrep name
5. terraform plan -destroy -out main.destroy.tfplan
6. terraform apply main.destroy.tfplan
7. az group list | fgrep name