Jack Roper

Follow 13 Followers About

Converting information from a spreadsheet into Terraform formatted variable files...part 2!

Jack Roper Aug 13, 2020 · 2 min read

In part one of this post, I discussed how a simple CSV file containing a list of subnets could be read into a powershell script and and converted into a Terraform configuration file containing those subnets as variables, ready for use in Terraform. This streamlines the workflow and delivery of projects from customer to engineer.

Working with my excellent colleague, **Joseph Gadd**, he developed a solution in python to take this concept a step further. I do not take any credit for his script as it was all his own work!

The idea of the script is that it can take multiple variable types as input from the spreadsheet, and format them into the correct layout in the terraform configuration file. It can handle simple variable blocks with a description, maps, and lists.

The script is publicly available on my GitHub with Joes permission, it also includes example input and output files. See the readme file for instructions on how to use.

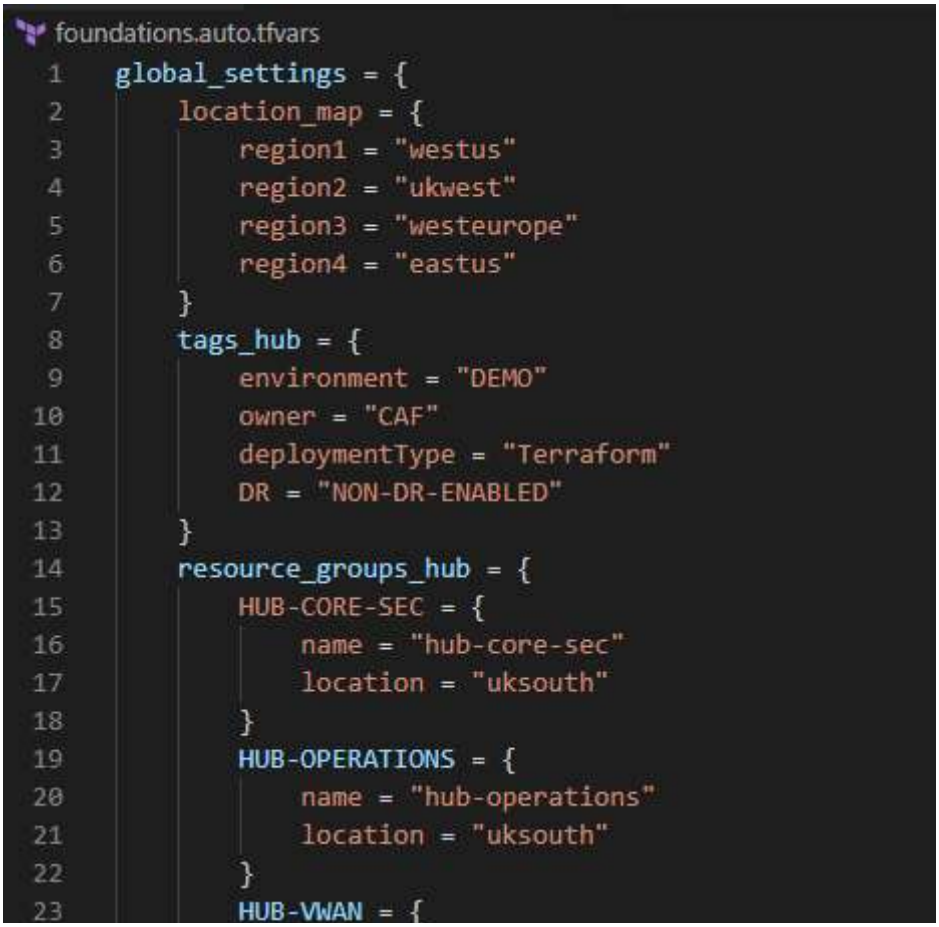
https://github.com/jwroper/excel-to-terraform-convertor

The input spreadsheet looks like this:

	A	B	C
1	foundations.auto.tfvars	global_settings	autovars
2	location_map		
3	region1	westus	
4	region2	ukwest	
5	region3	westeurope	

9	owner	CAF	
10	deploymentType	Terraform	
11	DR	NON-DR-ENABLED	
12	resource_groups_hub		HUB-CORE-SEC
13	name	hub-core-sec	^
14	location	uksouth	^
15		^	HUB-OPERATIONS
16	name	hub-operations	^
17	location	uksouth	^
18		^	HUB-VWAN
19	name	hub-vwan	^
20	location	uksouth	^
21			^
22	foundations.auto.tfvars	accounting_settings	autovars
23	general_settings		
24	azure_activity_logs_name	act_logs	
25	analytics_workspace_name	cafalogs	
26	solution_plan_map		ADAssesment
27	publisher	Microsoft	^
28	product	OMSGallery/ADAssesment	^
29		^	ADReplication
30	publisher	Microsoft	^
31	product	OMSGallery/ADReplication	^
32			^
33			

... and after conversion, a foundation.auto.tfvars file is generated, that looks like this:



```
    }
}
accounting_settings = {
    general_settings = {
        azure_activity_logs_name = "act_logs"
        analytics_workspace_name = "cafalogs"
    }
    solution_plan_map = {
        ADAssesment = {
            publisher = "Microsoft"
            product = "OMSGallery/ADAssesment"
        }
        ADReplication = {
            publisher = "Microsoft"
            product = "OMSGallery/ADReplication"
        }
    }
}
```

Information can be gathered easily from the customer, collected in the spreadsheet and be deployed into production in no time!

Hope you enjoyed this article and it helps you to streamline the delivery of your projects!

Originally published at https://azurelynot.blogspot.com on August 13, 2020.

Python Terraform Cloud Adoption