Terraform Associate Study Guide

Also see how I recommend studying: https://www.selikoff.net/2022/08/21/how-i-recommend-studying-for-the-terraform-associate-exam/

And my take on the online exam experience: https://www.selikoff.net/2022/08/20/my-first-online-cert-psi-terraform-associate-exam/

Note: I assume you know what IaC/DevOps is and have left that out of this document

Workflow

- Write (code)
- Plan (terraform plan)
- Create (terraform apply)

Commands

Command	Explanation
terraform state list terraform state rm terraform state show	List resources Delete resources Show details for single resource
terraform init	• Download plugins from providers (from public restigry or private URL) • Initializes working directory and create backend for storing state files • Working directory contains config, hidden .terraform directory as cache, state data (unless storing locally) • State file is terraform.tfstate if single workspace or terraform.tfstate.d is multiple workspaces • If reinitialize, will redownload provider if needed. Pass -reconfigure or -migrate-state • Installs child modules if present
terraform validate	Local validation (doesn't access any remote services)
terraform plan	• Displays plan for what will happen when run apply • Tests credentials since reads real state • Does not change anything except state file
terraform apply	• Deploys to real infrastructure • Updates state file • -auto-approve so can run unprompted

	• 10 concurrent operations by default
terraform destroy	Destroys all resources in state file
terraform refresh	Update state file
terraform graph	Creates dot file
terraform fmt	Formats all .tf files and outputs list of files changed
terraform version	Installed version
terraform workspace select terraform workspace new *name*	Choose workspace Create new workspace
terraform console	For testing expressions
terraform output *resource*	Get info from state file
terraform import	• Maps an existing reference so can reference by an id • Not guaranteed resource exists or importing with two names

Concepts

Command	**Explanation**
Provider	• Plugins, provide by vendors (ex: AWS) • .terraform.lock.hcl to pin versions • Can come from registry or folder
Constraints	Primitive (string, number, boolean) or any (primitive type to be decided)
Complex types	Collections (list, set, map) or structural (like object)
Resources vs data	Resources managed by terraform. Data sources already exist and are just referenced
State	• JSON file with metadata representing what is currently deployed • By default, stored in terraform.tfstate file in directory with code • Better to store remotely • Can contain sensitive data
HCL (HashiCorp Config Language) functions	Can’t add own functions. Only built ins. Examples: • contains([1, 2], 2) • file(fileName) • flatten([], [1], [[1,2]]) • join(“,” [“a,”, b”]) • max(1,2,3) • timestamp()
Variables	• Input referenced as var.name • Output referenced as module.modName.outputName • Root or child modules can use output variables • Best practice - store in terraform.tfvars

	- Sensitive parameter defaults to false. Set to true to avoid showing value in logs
Logging	- TF_LOG – 5 levels of logging - TF_LOG_PATH – location
Modules	- Container for related reousrces. - Root module by default - Can call other modules - Folder or code - Goal is code reuse - Meta args – count, dynamic block (for_each), providers (pass to child), depends_on (create explicit dependencies)
Provisioner	- Execute custom scripts/commands - local-exec, remote-exec, null_resources - Create and destroy time - Try to avoid using provisioners - Independent of state file

Cloud/Integrations

Command	**Explanation**
Features	- Shared state/secrets, access control, private registry, policy controls - Remote terraform execution/state mgmt - Hosted service – app.terraform.io
Not free for	>5 users, different permission levels, enforce policies
Business tier	More concurrent runs, private environments, SSO
Enterprise	On-prem version
Vault	Secret management for longer lived creds and inject short lived keys. Has ACLs
Sentinel	Codifies policy as code, own language. In Terraform Enterprise between plan and apply. Includes sandboxing/guardrails for automation. Codification and testing/automation