

Attempt 2

All questions

Question 1: **Correct**

Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. What command will do this?

☐

A. terraform apply

☒

B. terraform taint

(Correct)

☐

C. terraform refresh

☐

D. terraform graph

Question 2: **Correct**

After creating a new workspace "PROD" you need to run the command terraform select PROD to switch to it.

☐

TRUE

☒

FALSE

(Correct)

Explanation

By default, when you create a new workspace you are automatically switched to it

Question 3: **Correct**

Taint the resource "aws_instance" "baz" resource that lives in module bar which lives in module foo.

☐

A. terraform taint module.foo.bar.aws_instance.baz

☒

B. terraform taint module.foo.module.bar.aws_instance.baz

(Correct)

☐

C. terraform taint foo.bar.aws_instance.baz

☐

D. terraform taint module.foo.module.bar.baz

Explanation

Check resource addressing

<https://www.terraform.io/docs/internals/resource-addressing.html>

Question 4: **Correct**

The terraform import command can import resources directly into modules

☒

TRUE

(Correct)



FALSE

Question 5: **Correct**

State locking does not happen automatically and must be specified at run



TRUE



FALSE

(Correct)

Explanation

State locking happens automatically on all operations that could write state.

<https://www.terraform.io/docs/state/locking.html>

Question 6: **Correct**

Eric needs to make use of module within his terraform code. Should the module always be public and open-source to be able to be used?



TRUE



FALSE

(Correct)

Question 7: **Correct**

Which of the below options is a valid interpolation syntax for retrieving a data source ?



A. `${aws_instance.web.id.data}`



B. `${google_storage_bucket.backend}`



C. `${azurerm_resource_group.test.data}`



D. `${data.google_dns_keys.foo_dns_keys.key_signing_keys[0].ds_record}`

(Correct)

Explanation

Data source attributes are interpolated with the general syntax `data.TYPE.NAME.ATTRIBUTE`.

The interpolation for a resource is the same but without the `data.` prefix (TYPE.NAME.ATTRIBUTE).

<https://www.terraform.io/docs/configuration-0-11/interpolation.html#attributes-of-a-data-source>

Question 8: **Correct**

A colleague has informed you that a new version of a Terraform module that your team hosts on an Amazon S3 bucket is broken. The Amazon S3 bucket has versioning enabled. Your colleague tells you to make sure you are not using the latest version in your configuration. You have the following configuration block in your code that refers to the module:

```
module "infranet" { source = "s3::https://s3-us-west-2.amazonaws.com/infrabucket/infra_module.zip" }
```

Of the available choices, what is the best way to ensure that you are not using the latest version of the module?



A. Add a version key to the module configuration and specify a previous version



B. Add a version property to the module in Terraform's state file and specify a previous version



C. Delete the latest version of the module in S3 to rollback to the previous version

(Correct)



D. Add a module version constraint in your configuration's backend block and specify a previous version

Explanation

Because only Terraform Registries support module versioning by using the version key, you cannot configure a previous version of the module in the configuration. Deleting the latest version of the module in S3 is the only option of the available options that ensures you won't use the latest version. You could also modify the source URL to specify a versionId URL parameter for a previous version.

<https://www.terraform.io/docs/configuration/modules.html#source>

Question 9: Correct

What kind of resource dependency is stored in terraform.tfstate file?



A. Only explicit dependencies are stored in state file



B. Both implicit and explicit dependencies are stored in state file

(Correct)



C. Only implicit dependencies are stored in state file



D. No dependency information is stored in state file

Explanation

Terraform state captures all dependency information, both implicit and explicit. One purpose for state is to determine the proper order to destroy resources. When resources are created all of their dependency information is stored in the state. If you destroy a resource with dependencies, Terraform can still determine the correct destroy order for all other resources because the dependencies are stored in the state.

<https://www.terraform.io/docs/state/purpose.html#metadata>

Question 10: Correct

If you delete a remote backend from the configuration you will need to rebuild your state files locally



TRUE



FALSE

(Correct)

Explanation

As part of the re-initialization, Terraform will ask if you'd like to migrate your state back down to normal local state

Question 11: **Correct**

Which of the below command will upgrade the provider version to the latest acceptable one ?

☐

A. terraform plan upgrade

☐

B. terraform provider -upgrade

☒

C. terraform init -upgrade

(Correct)

☐

D. terraform init -update

Explanation

<https://www.terraform.io/docs/configuration/providers.html>

Question 12: **Correct**

Multiple configurations for the same provider can be used in a single configuration file.

☒

TRUE

(Correct)

☐

FALSE

Question 13: **Correct**

Jim has created several AWS resources from a single terraform configuration file. Someone from his team has manually modified one of the EC2 instance.

Now to discard the manual change, Jim wants to destroy and recreate the EC2 instance. What is the best way to do it ?

☐

A. terraform refresh

☒

B. terraform taint

(Correct)

☐

C. terraform recreate

☐

D. terraform destroy

Question 14: **Correct**

Which of the below options is the equivalent Terraform 0.12 version of the snippet which is written in Terraform 0.11 ?

"\${var.instance_id}"

☒

A. var.instance_id

(Correct)



B. variable.instance_id



C. var.instance_ids



D. None of the above

Question 15: Correct

You cannot publish your own modules on the Terraform Registry.



FALSE

(Correct)



TRUE

Explanation

You CAN publish your own modules on the Terraform Registry

Question 16: Correct

Which of the following sub command is used to retrieve a list of resources that are part of the state file?



A. terraform state view



B. terraform state list

(Correct)



C. terraform list



D. terraform view

Question 17: Correct

Dawn has created the below child module. Without changing the module, can she override the instance_type from t2.micro to t2.large form her code while calling this module ?

```
resource "aws_instance" "myec2" {  
  ami = "ami-082b5a644766e0e6f"  
  instance_type = "t2.micro"  
}
```



YES



NO

(Correct)

Explanation

As the instance_type is hard-coded in source module, you will not be able to change its value from destination module. Instead of hard-coding you should use variable with default values.

Question 18: Correct

Mary has created a database instance in AWS and for ease of use is outputting the value of the database password with the following code:

```
output "db_password" {  
    value = local.db_password  
}
```

Mary wants to hide the output value in the CLI after terraform apply? What is the best way?

☐

A. Encrypt the value using encrypt() function

☒

B. Use sensitive parameter

(Correct)

☐

C. Use cryptographic hash

☐

D. Use secure parameter

Question 19: Correct

Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes.

☒

TRUE

(Correct)

☐

FALSE

Explanation

<https://www.terraform.io/docs/state/sensitive-data.html>

Question 20: Correct

Your manager has instructed you to start using terraform for the entire infra provisioning of the application stack. There are 4 environments – DEV , QA , UAT , and PROD. The application team has asked for complete segregation between these environments including the backend , state , and also configurations ,since there will be unique resources in different environments . What is the possible way to structure the terraform code to facilitate that.

☐

A. Implement terraform workspaces , and map each environment with one workspace.

☐

B. Enable remote backend storage . Configure 4 different backend storages , one for each environment.

☒

C. Completely separate the working directories , keep one for each environment . For each working directory , maintain a separate configuration file , variables file , and map to a different backend.

(Correct)

☐

D. Completely separate the working directories , keep one for each environment . For each working directory , maintain a separate configuration file , variables file , and map to the same backend.

Explanation

In particular, organizations commonly want to create a strong separation between multiple deployments of the same infrastructure serving different development stages (e.g. staging vs. production) or different internal teams. In this case, the backend used for each deployment often belongs to that deployment, with different credentials and access controls. Named workspaces are *not* a suitable isolation mechanism for this scenario.

<https://www.terraform.io/docs/state/workspaces.html>

Question 21: **Correct**

Every region in AWS has a different AMI ID for Linux and these are keep on changing. What is the best approach to create the EC2 instances that can deal with different AMI IDs based on regions? ?



A. Use data source aws_ami

(Correct)



B. Create a map of region to ami id



C. create different configuration file for different region



D. None of the above

Explanation

<https://www.terraform.io/docs/configuration/data-sources.html>

Question 22: **Correct**

You have already set `TF_LOG = DEBUG` to enable debug log. Now you want to always write the log to the directory you're currently running terraform from. what should you do to achieve this.



A. run the command `export TF_DEBUG_PATH=./terraform.log`



B. run the command `export TF_LOG_PATH=./terraform.log`

(Correct)



C. No explicit action required. Terraform will take care of this as you have enable TF_LOG



D. run the command `export TF_LOG_FILE=./terraform.log`

Explanation

<https://www.terraform.io/docs/commands/environment-variables.html>

C is not a correct option as Terraform does not persist log.

Question 23: **Correct**

Which feature of Terraform allows multiple state files for a single configuration file depending upon the environment?



A. Terraform Modules



B. Terraform Remote Backends



C. Terraform Workspaces

(Correct)



D. Terraform Enterprise

Question 24: Correct

A variable az has the following default value. What will be the datatype of the variable ?

```
az=["us-west-1a","us-east-1a"]
```



A. Map



B. List

(Correct)



C. Object



D. String

Question 25: Correct

Your company has a lot of workloads in AWS , and Azure that were respectively created using CloudFormation , and AzureRM Templates. However , now your CIO has decided to use Terraform for all new projects , and has asked you to check how to integrate the existing environment with terraform code. What should be your next plan of action?



A. Tell the CIO that this is not possible . Resources created in CloudFormation , and AzureRM templates cannot be tracked using terraform.



B. This is only possible in Terraform Enterprise , which has the TerraformConverter exe that can take any other template language like AzureRM and convert to Terraform code.



C. Just write the terraform config file for the new resources , and run terraform apply , the state file will automatically be updated with the details of the new resources to be imported.



D. Use terraform import command to import each resource one by one .

(Correct)

Question 26: Correct

Your team uses terraform OSS . You have created a number of reusable modules for important , independent network components that you want to share with your team to enhance consistency . What is the correct option/way to do that?



A. Terraform modules cannot be shared in OSS version . Each developer needs to maintain their own modules , and leverage them in the main tf file.



B. Terraform module sharing is only available in Enterprise version via terraform private module registry , so no way to enable it in OSS version.



C. Store your modules in a NAS/ shared file server , and ask your team members to directly reference the code from there . This is the only viable option in terraform OSS ,which is better than individually maintaining module versions for every developer.



D. Upload your modules with proper versioning in the terraform public module registry . Terraform OSS is directly integrated with the public module registry , and can reference the modules from the code in the main tf file.

(Correct)

Explanation

<https://www.terraform.io/docs/registry/index.html>

Question 27: Correct

You wanted to destroy some of the dependent resources from real infrastructure. You choose to delete those resources from your configuration file and run terraform plan and then apply. Which of the following way your resources would be destroyed ?



A. Those would be destroyed in the order in which they were written in the configuration file previously before you have deleted them from configuration file.



B. Terraform can still determine the correct order for destruction from the state even when you delete one or more items from the configuration.

(Correct)



C. The resource will be destructed in random order as you have already deleted them from configuration



D. You can not destroy resources by deleting them from configuration file and running plan and apply

Explanation

B. Is the correct ans. D is not true because you can destroy resources by deleting them from configuration file and running plan and apply

Terraform typically uses the configuration to determine dependency order. However, when you delete a resource from a Terraform configuration, Terraform must know how to delete that resource. Terraform can see that a mapping exists for a resource not in your configuration and plan to destroy. However, since the configuration no longer exists, the order cannot be determined from the configuration alone.

To ensure correct operation, Terraform retains a copy of the most recent set of dependencies within the state. Now Terraform can still determine the correct order for destruction from the state when you delete one or more items from the configuration.

Question 28: Correct

Talal is a DevOps engineer and he has deployed the production infrastructure using Terraform. He is using a very large configuration file to maintain and update the actual infrastructure. As the infrastructure have grown to a very complex and large, he has started experiencing slowness when he run runs terraform plan. What are the options for him to resolve this slowness?

☐

A. Run terraform refresh every time before running terraform plan.

☒

B. Use `-refresh=false` flag as well as the `-target` flag with terraform plan in order to work around this.

(Correct)

☒

C. break large configurations into several smaller configurations that can each be independently applied.

(Correct)

☐

D. Use `-refresh=true` flag as well as the `-target` flag with terraform plan in order to work around this.

Explanation

For larger infrastructures, querying every resource is too slow. Many cloud providers do not provide APIs to query multiple resources at once, and the round trip time for each resource is hundreds of milliseconds. On top of this, cloud providers almost always have API rate limiting so Terraform can only request a certain number of resources in a period of time. Larger users of Terraform make heavy use of the `-refresh=false` flag as well as the `-target` flag in order to work around this. In these scenarios, the cached state is treated as the record of truth.

Although **option B** is a solution, but its not always recommended. Instead of using `-target` as a means to operate on isolated portions of very large configurations, prefer instead to break large configurations into several smaller configurations that can each be independently applied. [Data sources](#) can be used to access information about resources created in other configurations, allowing a complex system architecture to be broken down into more manageable parts that can be updated independently.

Option A and D is not correct because in both the cases terraform will query every resources of the infrastructure.

Question 29: Correct

Which of the below backends support state locking ?

☒

A. azurerm

(Correct)

☐

B. artifactory

☒

C. Consul

(Correct)



D. S3

(Correct)

Question 30: Correct

Which of the following connection types are supported by the remote-exec provisioner?
(select two)



A. SSh

(Correct)



B. UDP



C. winrm

(Correct)



D. RDP

Explanation

<https://www.terraform.io/docs/provisioners/remote-exec.html>

Retake testContinue