Workspace Assignment

You are managing infrastructure for multiple environments (dev, staging, and prod) using Terraform workspaces.

Your goal is to create an S3 bucket and an EC2 instance, where:

The S3 bucket name should be prefixed with the workspace name.

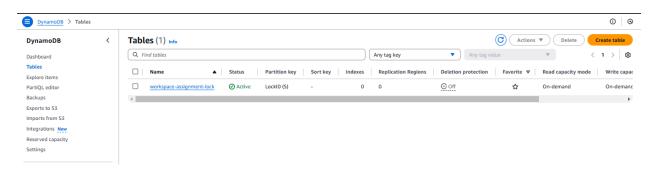
The EC2 instance type should vary based on the workspace (t2.micro for dev, t3.small for staging, and t3.medium for prod).

Store Terraform state in an S3 backend with state locking enabled using a DynamoDB table. Use variables for flexibility, locals for calculated values, and functions to dynamically construct names.

S3 Bucket creation



Create a table in dynamoDB



PS C:\Users\HP\Downloads\workspace-assignment> terraform workspace new staging Created and switched to workspace "staging"!

You're now on a new, empty workspace. Workspaces isolate their state, so if you run "terraform plan" Terraform will not see any existing state for this configuration.

PS C:\Users\HP\Downloads\workspace-assignment> terraform workspace new dev Created and switched to workspace "dev"!

You're now on a new, empty workspace. Workspaces isolate their state, so if you run "terraform plan" Terraform will not see any existing state for this configuration.

PS C:\Users\HP\Downloads\workspace-assignment> terraform workspace list default

* dev prod

PS C:\Users\HP\Downloads\workspace-assignment> terraform workspace select staging Switched to workspace "staging".

PS C:\Users\HP\Downloads\workspace-assignment> terraform init

Initializing the backend...

staging

Initializing provider plugins...

- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/random from the dependency lock file
- Using previously-installed hashicorp/random v3.7.1
- Using previously-installed hashicorp/aws v5.89.0

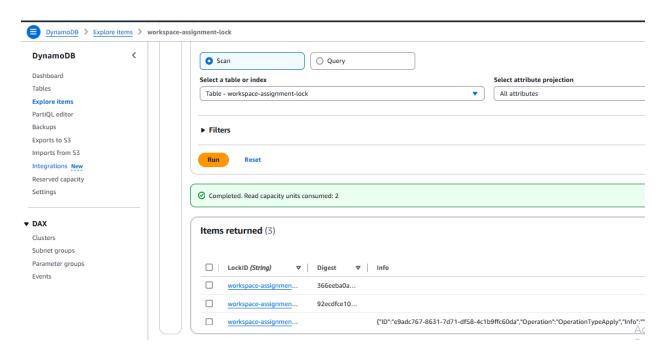
Terraform has been successfully initialized!

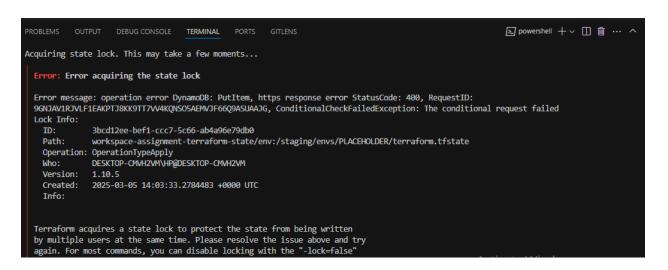
You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

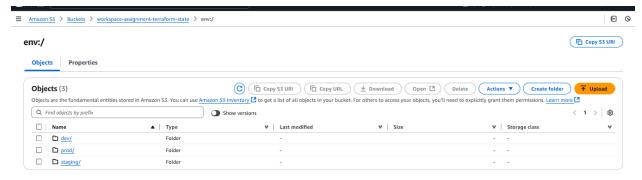
PS C:\Users\HP\Downloads\workspace-assignment>

Acquiring State lock





State file



```
Pretty print

{
    "version": 4,
    "terraform_version": "1.10.5",
    "serial": 3,
    "lineage": "edv673ee-4dd4-ad08-3ea1-be09c0574ad5",
    "uitpues": {},
    "esources": {},
    "mode: "managed",
    "myoe": "msw_instance",
    "name": "workspace_instance",
    "provider": "provider(\"registry.terraform.io/hashicorp/aws\"]",
    "instances": {
        "ami": "ami-04b4f1a0f54c11d0",
        "arin: "ami-04b4f1a0f54c11d0",
        "arin: "arin: assicizius-east-1:741448948407:instance/1-0ed5dd8734752074f",
        "associate_public_ip_address": true,
        "capacity_reservation_specification": [
        {
            ("capacity_reservation_specification": [
            ("capacity_reservation_specification": [
            ("capacity_reservation_target": []
            )
            ("capu_core_count": 1,
            "core_count": 1,
```

EC2 creation

