Common Additional Tasks

Cleaning Deployment cache

Despite the fact that the Terraform cache is automatically cleared before each deployment, you may also want to manually force the cleaning process. To clear the Terraform cache, Run theansible-playbook clean.yml command.

Migrating deployer to another machine

You can easily manipulate your deployment from any machine with sufficient prerequisites. If theupload_debug_info_to_s3 variable is set to true, the deployer will automatically upload yourall.yml file to the s3 bucket, so you can download it to any other machine. Simply download this file to yourgroup_vars folder and your new deployer will pick up the current deployment instead of creating a new one.

Attaching the existing RDS instance to the current deployment

Rather than create a new database, you may want to add an existing instance to use with the deployment. To do this, configure all proper values atgroup_vars/all.yml , including your DB ID and name, and execute theansible-playbook attach_existing_rds.yml command. This will add the current DB instance into the Terraform-managed resource group. After that runansible-playbook deploy_infra.yml as you normally would.

Notes

1. While executingansible-playbook attach_existing_rds.yml
2. the S3 and DynamoDB instances will be automatically created (ifbackend
3. variable is set totrue
4.) to store Terraform state files.
5. The actual name of your resource must include the prefix you are using with this deployment.
6. Example:
7. Real resource: tf-poaprefix
8. variable: tfchain_db_id
9. variable: poa
10. Make sure MultiAZ is disabled on your database.
11. Make sure that all the variables atgroup_vars/all.yml
12. are exactly the same as your existing DB.
13.

Using AWS CodeDeploy to Monitor and manage a BlockScout deployment

BlockScout deployment can be managed through the AWS consoleA brief tutorial is available on our forum .

Last updated4 years ago