AWS Permissions & Settings

AWS requirements for deployment

During deployment you will provide credentials to your AWS account. Given the large number of services involved, and the unpredictability of which specific API calls will be needed during provisioning, it is recommended you provide a user account with full access.

You do not need to keep this user around (or enabled) except during the initial provisioning, and for any subsequent runs to update the infrastructure.

You may need to reference these items during deployment:

- Creating a Secret Key Pair in AWS
- Logging in with AWS CLI
- Creating an SSL certificate
- Manually resource cleaning (if deployment fails or you need to remove items)

Resources Created During Deployment

If you do want to restrict the permissions, the following list of resources is created during the deployment process:

- An S3 bucket to keep Terraform state files;
- DynamoDB table to manage Terraform state files leases;
- An SSH keypair (or you can choose to use one which was already created), this is used with any EC2 hosts;
- A VPC containing all of the resources provisioned;
- A public subnet for the app servers, and a private subnet for the database (and Redis for now);
- An internet gateway to provide internet access for the VPC;
- An ALB which exposes the app server HTTPS endpoints to the world;
- A security group to lock down ingress to the app servers to 80/443 + SSH;
- A security group to allow the ALB to talk to the app servers;
- A security group to allow the app servers access to the database;
- An internal DNS zone;
- A DNS record for the database;
- An autoscaling group and launch configuration for each chain;
- A CodeDeploy application and deployment group targeting the corresponding autoscaling groups.

Each configured chain receives its own ASG (autoscaling group) and deployment group. When application updates are pushed to CodeDeploy, all autoscaling groups will deploy the new version using a blue/green strategy. Currently, there is only one EC2 host to run, and the ASG is configured to allow scaling up, but no triggers are set up to actually perform the scaling yet. This is something that may come in the future.

When deployment begins, Ansible creates the S3 bucket and DynamoDB table required for Terraform state management. This ensures that the Terraform state is stored in a centralized location, allowing multiple people to use Terraform on the same infra without interfering with one another. Terraform prevents interference by holding locks (via DynamoDB) against the state data (stored in S3).

This instruction was moved fromhttps://forum.poa.network/t/aws-settings-for-blockscout-terraform-deployment/1962

Last updated5 years ago