Cloudformation Template Documentation

Overview

This document will provide an explanation of the deployed environment and a brief explanation of the CloudFormation templates.

Additional material

There is a high-level diagram which provides an overview of the solution in the same location as this document, the name of the file is *overview.pdf*

Deployment Strategy

I used CloudFormation templates to deploy the solution as it provides potential for further automation.

I tried to logically separate each part of the solution as it allowed for faster deployments and easier troubleshooting with the intention being to combine all the yml files into one larger all-encompassing CloudFormation template once everything was working.

The repo consists of the following templates:

vpc-m.yml

This is an aws template that I used to deploy a custom VPC, found here:
https://docs.aws.amazon.com/codebuild/latest/userguide/cloudformation-vpc-template.html

Only changes I made were to the CIDR blocks via the parameters, this can easily be stripped out or placed in another location to be retrieved during a build.

s3.yml

For the creation of an s3 bucket for use by the elastic beanstalk deployment.
I enabled versioning and encryption, not publicly viewable by default.

roles.yml

This was used for troubleshooting while running up the elastic beanstalk stack. I ran into quite a few issues and so I created a new role and allowed it to assume *arn:aws:iam::aws:policy/AWSElasticBeanstalkFullAccess* while it spun up resources for the beanstalk deployment. I eventually discovered that the issue was the method I was using to select instance types for the autoscaling group. I added some additional *OptionName* parameters in eb-m.yml which allowed it proceed further into the deployment.

rds-m.yml
For the deployment of the RDS Postgres solution into a private subnet of the custom VPC via providing a DBSubnetGroup, which requires subnets in at least two unique Availability Zones.
Deployed postgres 11.6 and allocated 20GB of storage.

eb-m.yml
For the elastic beanstalk deployment, unfortunately beanstalk likes to deploy into the default VPC, therefore I needed to add the following *OptionNames*:

- VPCID
- Subnets
- ELBSubnets -
- DBSubnets
- ELBScheme

The elastic beanstalk deployment also pulls the ruby app from s3.

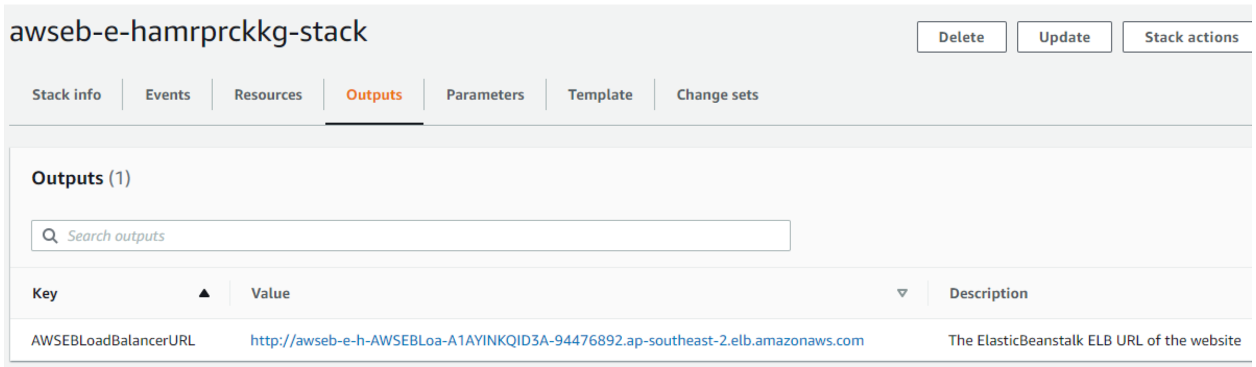
Testing

Was performed manually, will briefly outline some of the tests below:

Database
Spun up an ec2 instance which I used to connect to the postgres db

```
[ec2-user@ip-10-10-10-52 ~]$ psql -U sysalykes -h mablerdsm.cbdlh26f7kda.ap-southeast-
2.rds.amazonaws.com -d mableDBm

Password for user sysalykes:

psql (9.2.24, server 11.6)

WARNING: psql version 9.2, server version 11.0.
         Some psql features might not work.
SSL connection (cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256)

Type "help" for help.

 mableDBm=> \l

                                   List of databases

   Name     |   Owner    | Encoding |   Collate    |    Ctype     |    Access privileges

-----------+-----------+----------+-------------+-------------+------------------------

mableDBm  | sysalykes | UTF8     | en_US.UTF-8 | en_US.UTF-8 |

postgres  | sysalykes | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
rdsadmin  | rdsadmin  | UTF8     | en_US.UTF-8 | en_US.UTF-8 | rdsadmin=CTc/rdsadmin
template0 | rdsadmin  | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/rdsadmin          +
          |           |          |             |             | rdsadmin=CTc/rdsadmin
template1 | sysalykes | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/sysalykes         +
          |           |          |             |             | sysalykes=CTc/sysalykes
(5 rows)

mableDBm=> \q
```

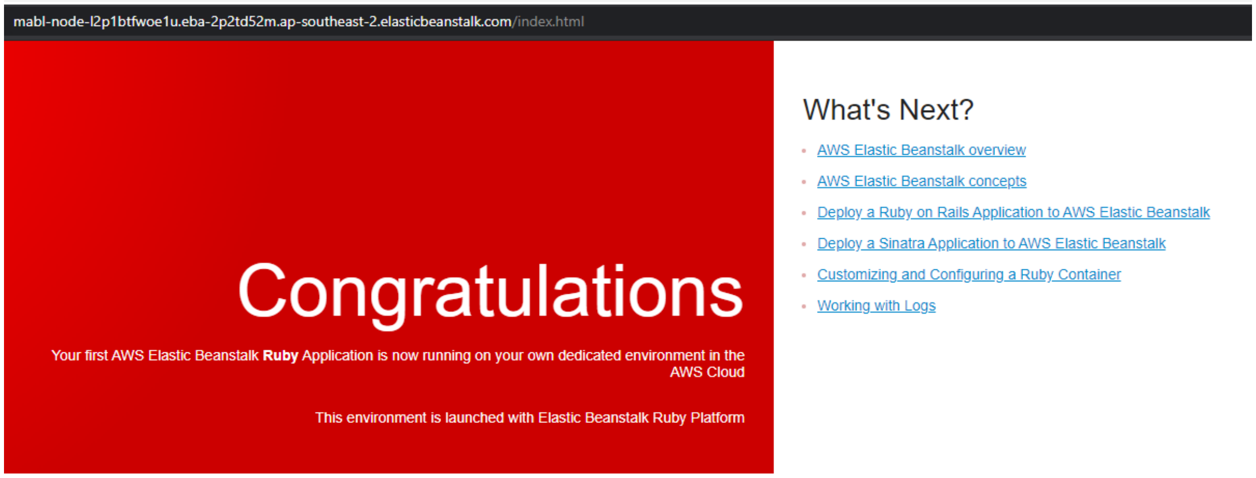
Elastic Beanstalk



awseb-e-hamrprckkg-stack

Outputs (1)

Key	Value	Description
AWSEBLoadBalancerURL	http://awseb-e-h-AWSEBLoa-A1AYINKQID3A-94476892.ap-southeast-2.elb.amazonaws.com	The ElasticBeanstalk ELB URL of the website



mabl-node-l2p1btfwoe1u.eba-2p2td52m.ap-southeast-2.elasticbeanstalk.com/index.html

Congratulations

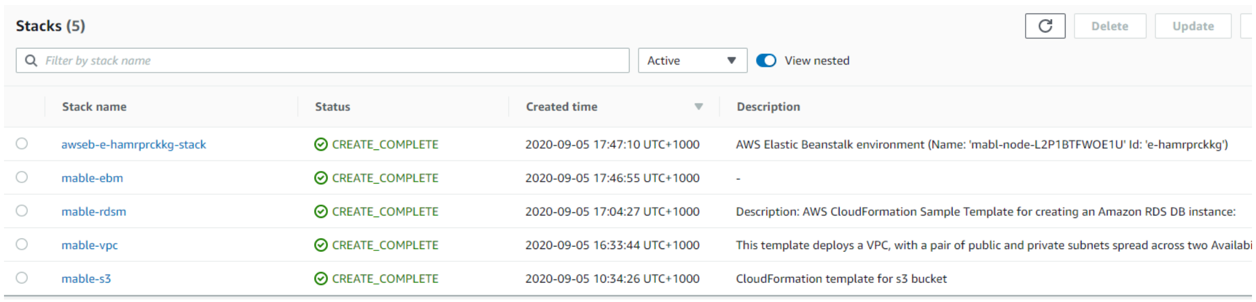
Your first AWS Elastic Beanstalk **Ruby** Application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Ruby Platform

What's Next?

- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploy a Ruby on Rails Application to AWS Elastic Beanstalk
- Deploy a Sinatra Application to AWS Elastic Beanstalk
- Customizing and Configuring a Ruby Container
- Working with Logs

Cloudformation Stacks



Stacks (5)

Stack name	Status	Created time	Description
awseb-e-hamrprckkg-stack	CREATE_COMPLETE	2020-09-05 17:47:10 UTC+1000	AWS Elastic Beanstalk environment (Name: 'mabl-node-L2P1BTFWOE1U' Id: 'e-hamrprckkg')
mable-ebm	CREATE_COMPLETE	2020-09-05 17:46:55 UTC+1000	-
mable-rdsm	CREATE_COMPLETE	2020-09-05 17:04:27 UTC+1000	Description: AWS CloudFormation Sample Template for creating an Amazon RDS DB instance:
mable-vpc	CREATE_COMPLETE	2020-09-05 16:33:44 UTC+1000	This template deploys a VPC, with a pair of public and private subnets spread across two Availabi
mable-s3	CREATE_COMPLETE	2020-09-05 10:34:26 UTC+1000	CloudFormation template for s3 bucket

Local Ruby on Rails environment – Connectivity to aws
Was able to successfully connect to the postgres RDS database via activerecord (when I had it located in a public subnet) via the rails console once database.yml was pointing to the correct location.
Added the following to the Gemfile https://rubygems.org/gems/aws-sdk-s3/ for connectivity to s3.

Solution Deployment Brief Steps

Run the templates in the following order:

- vpc-m.yml - take note of the VPCId, Private and Public Subnetids
- s3.yml - then upload ruby-sample.zip to the bucket once created
- rds-m.yml - need to specify the username/password also amend the template with vpcid and private subnets
- eb-m.yml - replace
    - VPCId: VPCId
    - Subnets: PrivateSubnet1, PrivateSubnet2
    - ELBSubnets : PublicSubnet1, PublicSubnet2
    - DBSubnets : PrivateSubnet1, PrivateSubnet2

Further CloudFormation Optimisations with more time permitting

- Once the VPC is created, it is then required to manually replace any hardcoded values for VPCIDs and Subnets in the CloudFormation templates, I would seek to auto populate these values.
- Potentially create a single CloudFormation template for the entire stack, depending on use case.
- Implement S3 encryption using AWS-KMS, again dependent on use case.
- Harden the environment with better management of security groups and usage of policies
- Use secrets manager for deployment of the rds database (username/password)
- Create a code repo specifically for a RoR webapp on elastic beanstalk and use CodePipeline for automating deployments
- Tweak max/min size of autoscaling group for the elastic beanstalk deployment (via MinSize and MaxSize) dependent on environment requirements
- Configure Elastic Beanstalk app to use SSL
- Encrypt the RDS database
- Create a runbook for deployment of the stack(s) including examples on how to run the cloudformation templates via aws cli

    -
      ```
      aws cloudformation create-stack --stack-name some-test --
      template-body file://./s3.yml
      ```