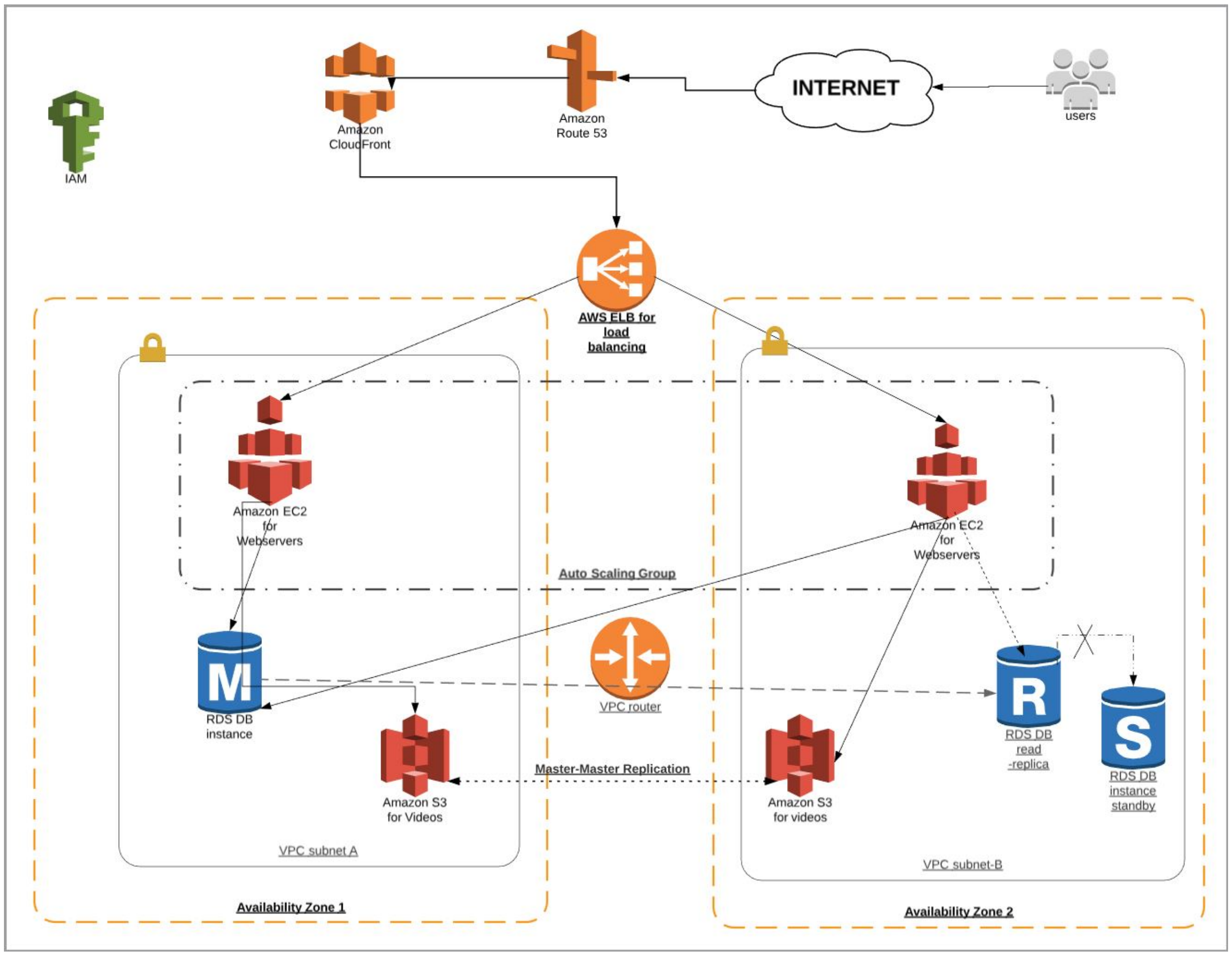
Migration of Trolls' Web-Application to AWS

A technical report on steps involved.

Compiled By: Keshaw Kumar Burnwal
UID:115819351

Migrating to AWS

Incorporating comments received for AWS migration plans, below is updated network architecture.



I recommend creating new Identity Access and Management (IAM) policies for various users. I am assuming that Troll is not hiring new employee any time soon.

Identity Access and Management (IAM)

GROUPS

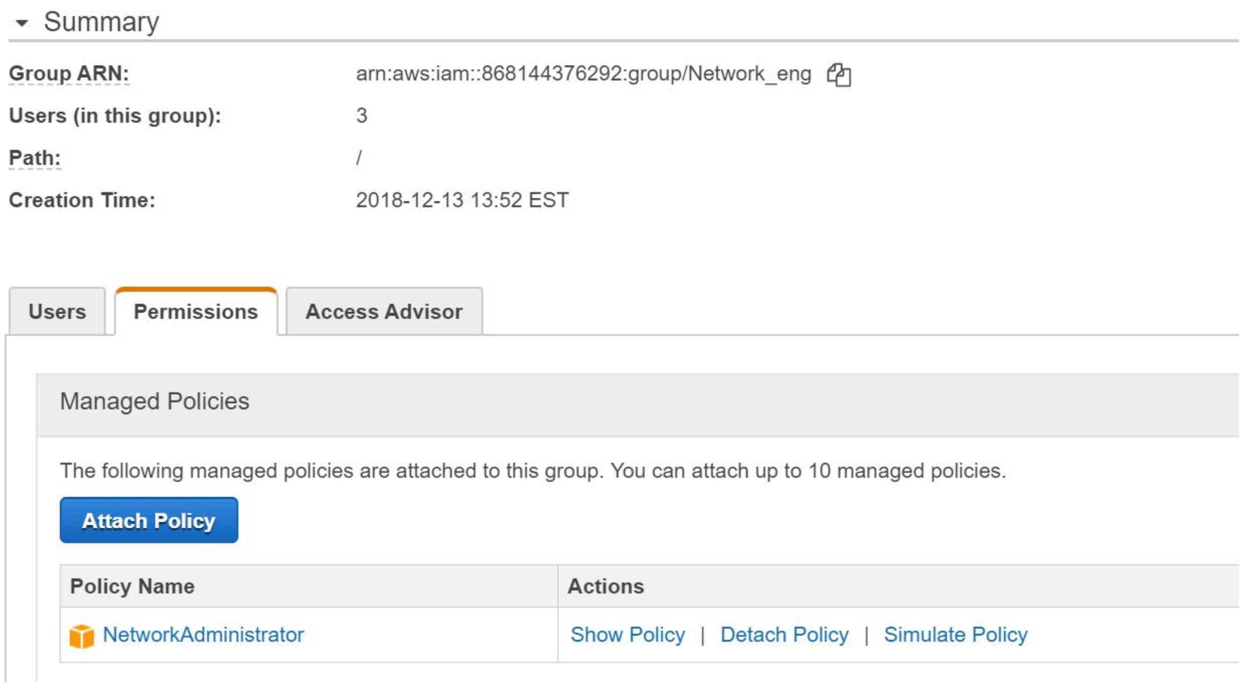
I recommend categorizing all possible users into below groups.

- **Network_eng** :- Users in this role are responsible for well-being of company's network and its components like routers, switches, routing policies, etc.

- **System_admin**:- Users in this role are employed as system administrators at Troll
- **Developers**:- Users in this role are responsible for building and testing company's web-application and related modules.
- **Management**:- Users in this role are responsible for taking all management related decisions.

Each of these groups should have policies as mentioned below.

Network_Eng. Users in this role are responsible for fixing any network related issues. Hence, they should administrator access.



Summary

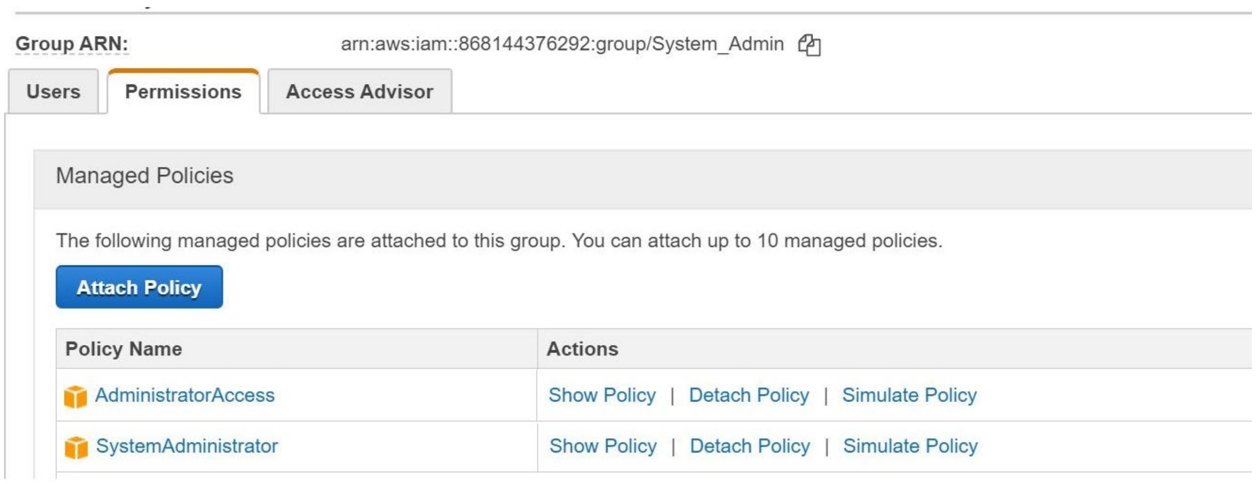
Group ARN:	arn:aws:iam::868144376292:group/Network_eng
Users (in this group):	3
Path:	/
Creation Time:	2018-12-13 13:52 EST

Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

Policy Name	Actions
NetworkAdministrator	Show Policy | Detach Policy | Simulate Policy

System_admin :- Employees having role of system administrators are responsible for fixing any issues that may arise in a machine. So, they should have administrative access to all machines.



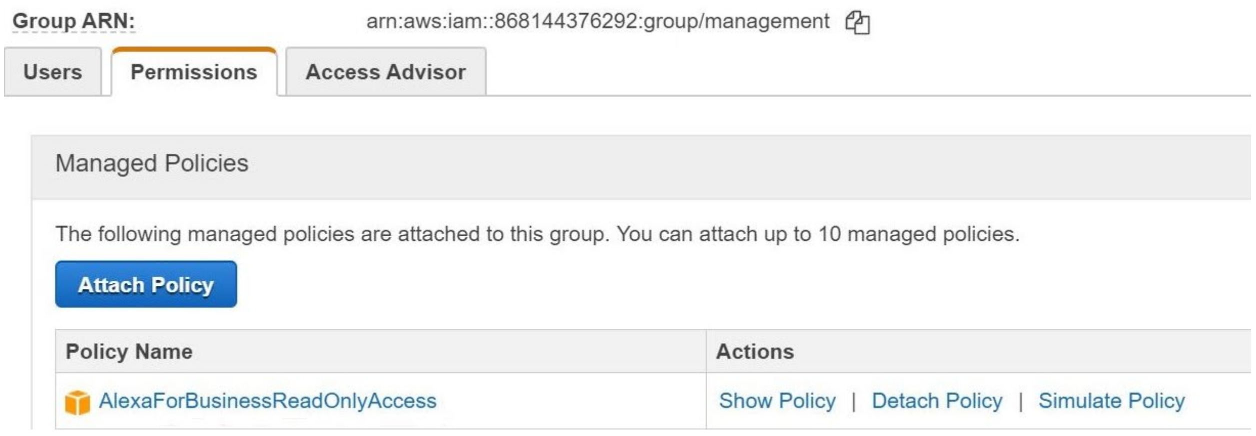
Group ARN: arn:aws:iam::868144376292:group/System_Admin

Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

Policy Name	Actions
AdministratorAccess	Show Policy | Detach Policy | Simulate Policy
SystemAdministrator	Show Policy | Detach Policy | Simulate Policy

Management: These are employees that make all business-related decisions. They are not expected to make any change in data. So, assigning them read-only permissions.


Group ARN: arn:aws:iam::868144376292:group/management

Users | Permissions | Access Advisor

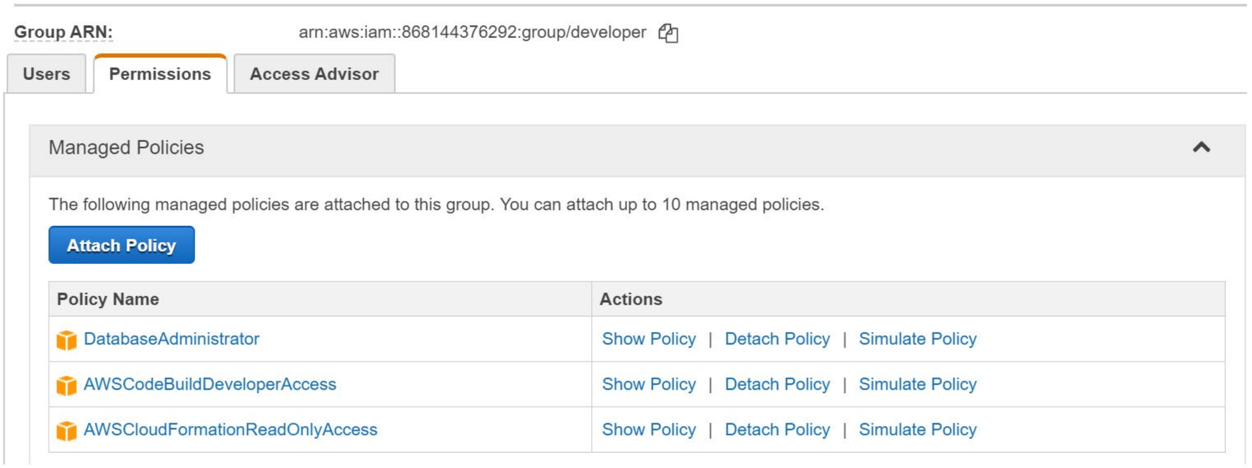
Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

Attach Policy

Policy Name	Actions
AlexaForBusinessReadOnlyAccess	Show Policy | Detach Policy | Simulate Policy

Developer: Employees in this role build and manage Troll's web-application. Furthermore, to ensure flawless auto-scalping of web-application, they need to test CloudFormation Templates. So giving them read-only permissions.


Group ARN: arn:aws:iam::868144376292:group/developer

Users | Permissions | Access Advisor

Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

Attach Policy

Policy Name	Actions
DatabaseAdministrator	Show Policy | Detach Policy | Simulate Policy
AWSCodeBuildDeveloperAccess	Show Policy | Detach Policy | Simulate Policy
AWSCloudFormationReadOnlyAccess	Show Policy | Detach Policy | Simulate Policy

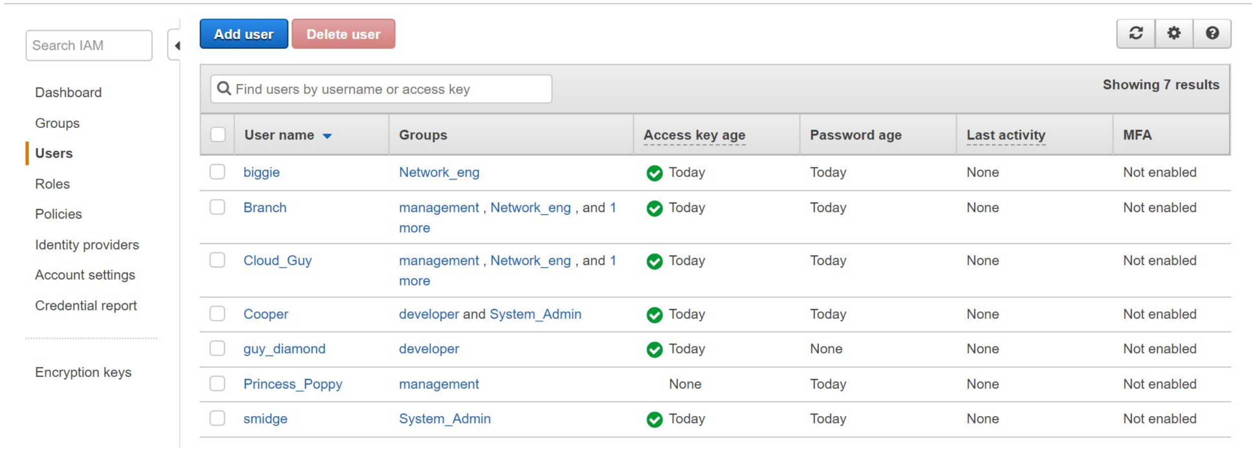
USERS

Assuming no employees are hired by Troll, I recommend assigning groups to entire leadership and development team as shown below

Employee Name	**Assigned Group**
Princess Poppy	Management
Branch	Management, Network_Eng, System_Admin
Cloud Guy	Management, Network_Eng, System_Admin
Smidge	System_Admin

Guy Diamond	Developer
Biggie	Network_Eng
Cooper	Developer, System_Admin.

Below is corresponding users in Identity and Access Management (IAM)

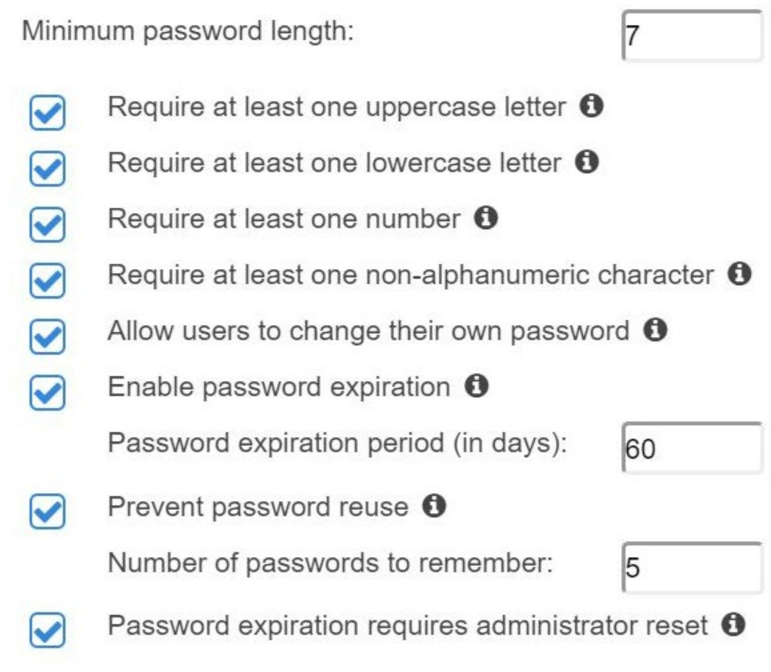


Showing 7 results

User name	Groups	Access key age	Password age	Last activity	MFA
biggie	Network_eng	✓ Today	Today	None	Not enabled
Branch	management , Network_eng , and 1 more	✓ Today	Today	None	Not enabled
Cloud_Guy	management , Network_eng , and 1 more	✓ Today	Today	None	Not enabled
Cooper	developer and System_Admin	✓ Today	Today	None	Not enabled
guy_diamond	developer	✓ Today	None	None	Not enabled
Princess_Poppy	management	None	Today	None	Not enabled
smidge	System_Admin	✓ Today	Today	None	Not enabled

Password Policy

Weak password is one of the most common attack vectors. Hence, I recommend a strong password policy. Below policy is in-line with industry standards.



Minimum password length: 7

- [x] Require at least one uppercase letter
- [x] Require at least one lowercase letter
- [x] Require at least one number
- [x] Require at least one non-alphanumeric character
- [x] Allow users to change their own password
- [x] Enable password expiration

Password expiration period (in days): 60

- [x] Prevent password reuse

Number of passwords to remember: 5

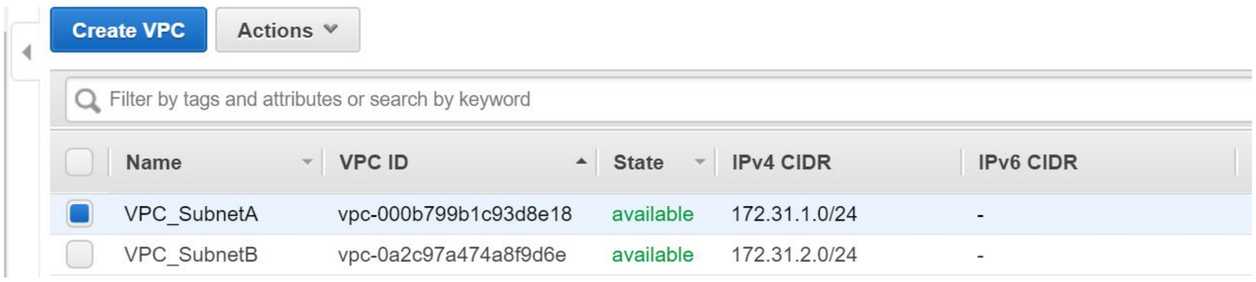
- [x] Password expiration requires administrator reset

AWS Instances

Once all user access policies are in place, I suggest spinning up various AWS instances.

VPC and Subnets

As shown in architecture diagram above, I suggest maintaining two virtual private cloud (VPCs) for systems running on different availability zone. This redundancy ensures better website response time and minimum business impact in case of machine outage.

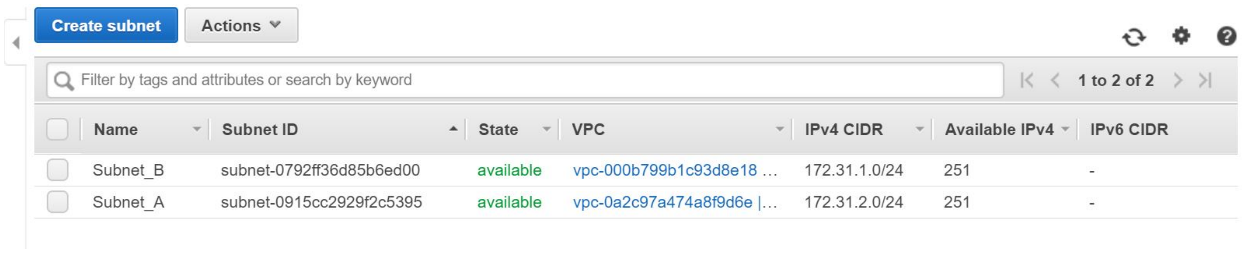


Create VPC | Actions

Filter by tags and attributes or search by keyword

	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
☑	VPC_SubnetA	vpc-000b799b1c93d8e18	available	172.31.1.0/24	-
☐	VPC_SubnetB	vpc-0a2c97a474a8f9d6e	available	172.31.2.0/24	-

Each of these VPCs should have one subnet each.



Create subnet | Actions

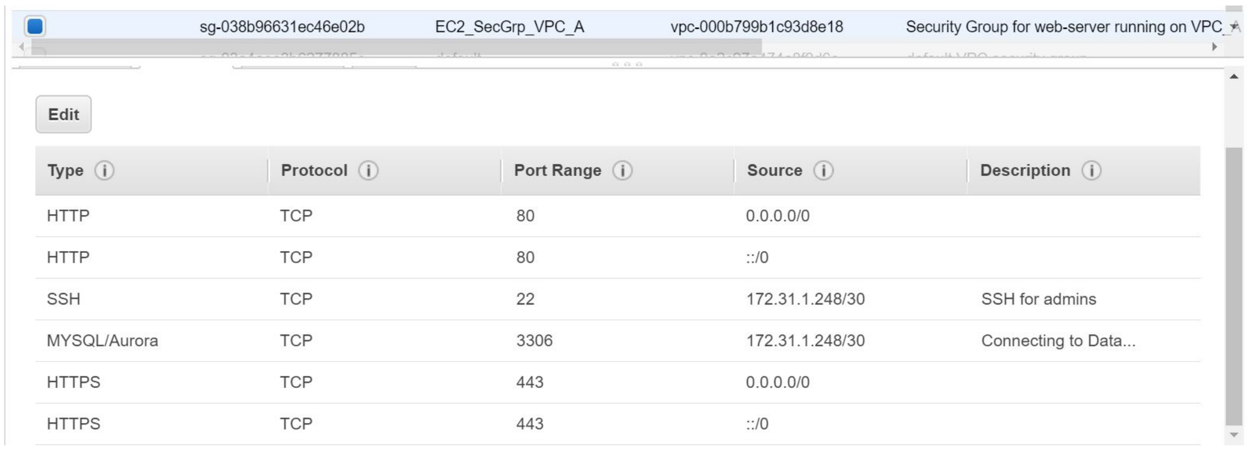
Filter by tags and attributes or search by keyword

1 to 2 of 2

	Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR
☐	Subnet_B	subnet-0792ff36d85b6ed00	available	vpc-000b799b1c93d8e18 ...	172.31.1.0/24	251	-
☐	Subnet_A	subnet-0915cc2929f2c5395	available	vpc-0a2c97a474a8f9d6e |...	172.31.2.0/24	251	-

EC2 Instances

Inside each of these subnets, an AWS EC2 machine running. I recommend allowing restricted input and output communications. Below is a sample incoming policy.



☑	sg-038b96631ec46e02b	EC2_SecGrp_VPC_A	vpc-000b799b1c93d8e18	Security Group for web-server running on VPC_

Edit

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
SSH	TCP	22	172.31.1.248/30	SSH for admins
MYSQL/Aurora	TCP	3306	172.31.1.248/30	Connecting to Data...
HTTPS	TCP	443	0.0.0.0/0	
HTTPS	TCP	443	::/0	

In addition to these incoming rules, one should add another incoming policy for connection between EC2 and S3 bucket. This policy depends upon port at which web-application will interact with S3. Administrators can SSH to this EC2 machine. Also, web-application interacts with Amazon RDS. Hence, I opened port 3306. I am assuming MySQL will be used for database storage.

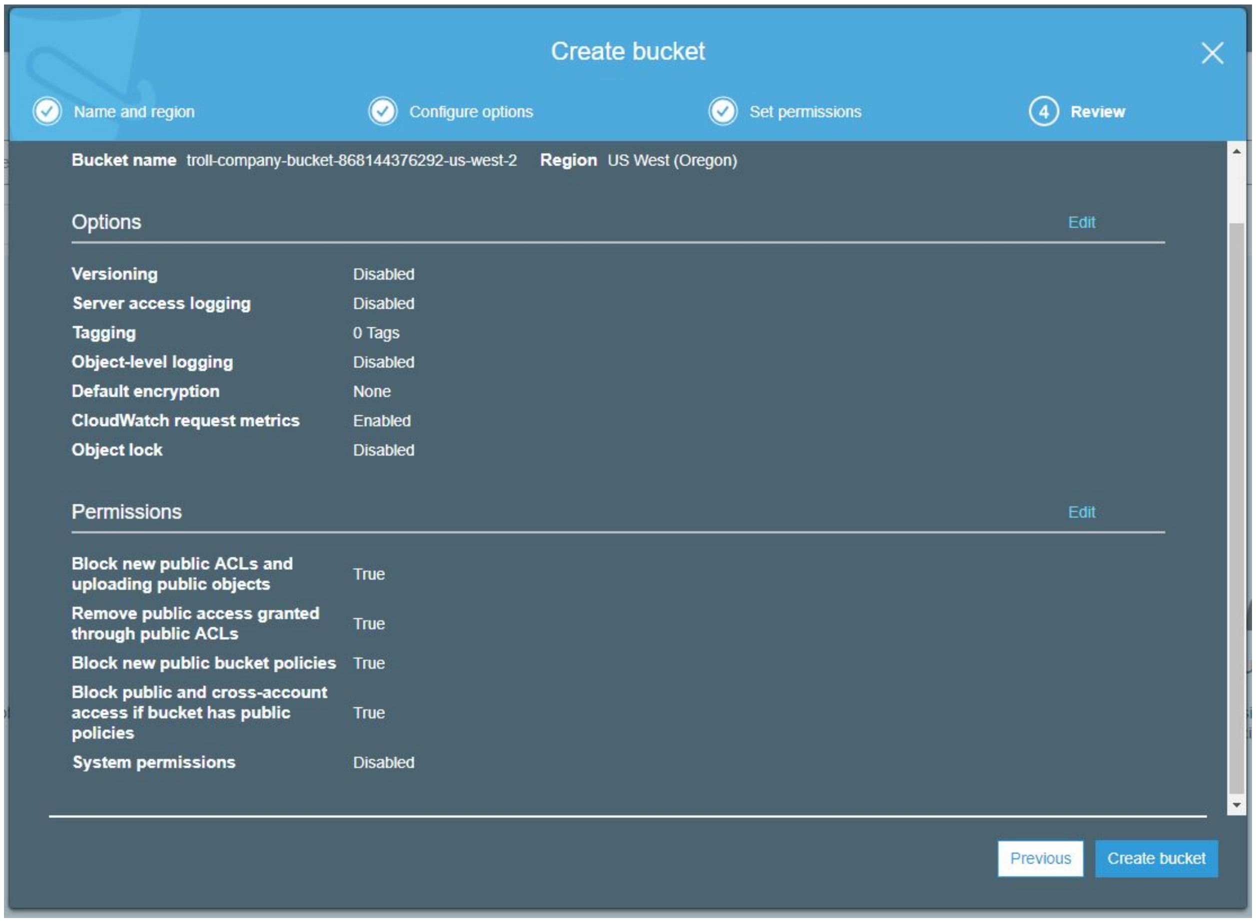
Furthermore, having restricted outgoing policy is recommended. Ports used for outgoing connection depends upon application running. Once that is known, I recommend updating outgoing policy with the same.

Each of these EC2 instances should use elastic IP. Also, there will be one EC2 instance in each availability zone.

S3 Bucket

S3 will store videos. This is non-critical information. Therefore, I don't recommend encrypting objects stored here. Furthermore, I don't recommend logging all access to S3 buckets. However, I recommend enabling AWS CloudTrail. Logs, thus produced, can be used for auditing purpose. AWS CloudWatch should also be enabled. This helps engineers to monitor for any downtime or any similar issues

Below is snapshot of recommended S3 bucket configuration.



Bucket name troll-company-bucket-868144376292-us-west-2 **Region** US West (Oregon)

Options

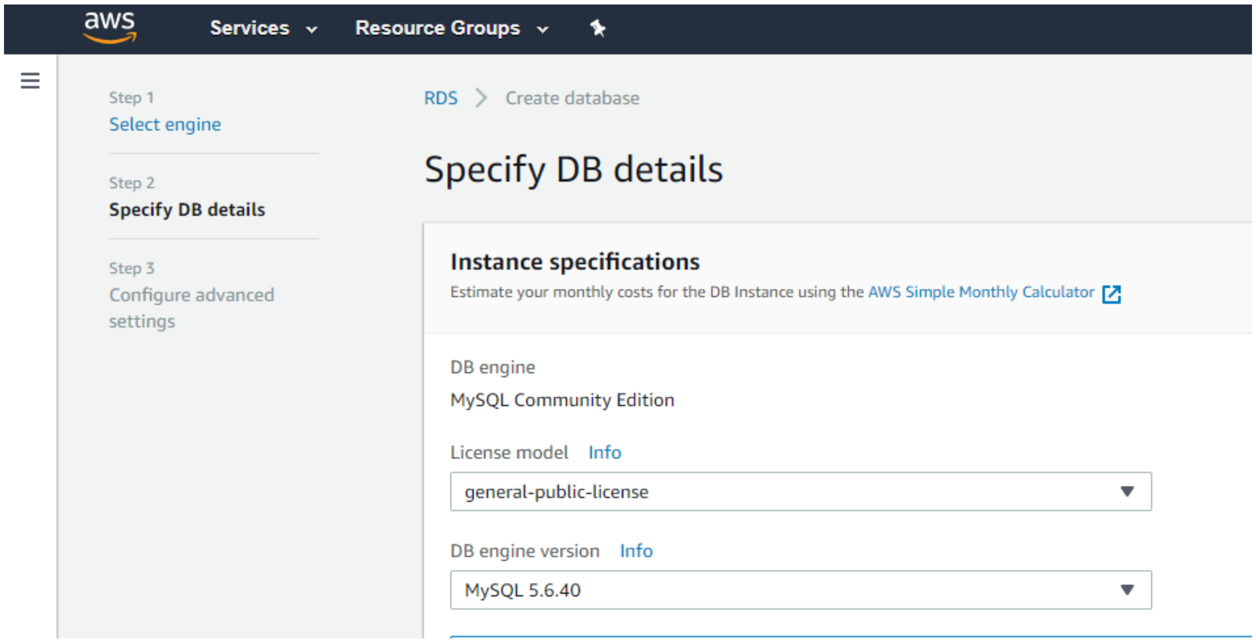
Versioning	Disabled
Server access logging	Disabled
Tagging	0 Tags
Object-level logging	Disabled
Default encryption	None
CloudWatch request metrics	Enabled
Object lock	Disabled

Permissions

Block new public ACLs and uploading public objects	True
Remove public access granted through public ACLs	True
Block new public bucket policies	True
Block public and cross-account access if bucket has public policies	True
System permissions	Disabled

AWS Relational Database Service(RDS)

Assuming Troll's web application was earlier connected to MySQL DB, I recommend using the same at AWS RDS. All communication between RDS and EC2 instances should happen using TLS protocol. Also, AES-256 should be used to encrypt data stored at RDS. This box should have only two incoming ports open (22 for SSH and 3306 for DB connection). Similarly, outgoing connection should also be restricted to minimum required.



Specify DB details

Instance specifications

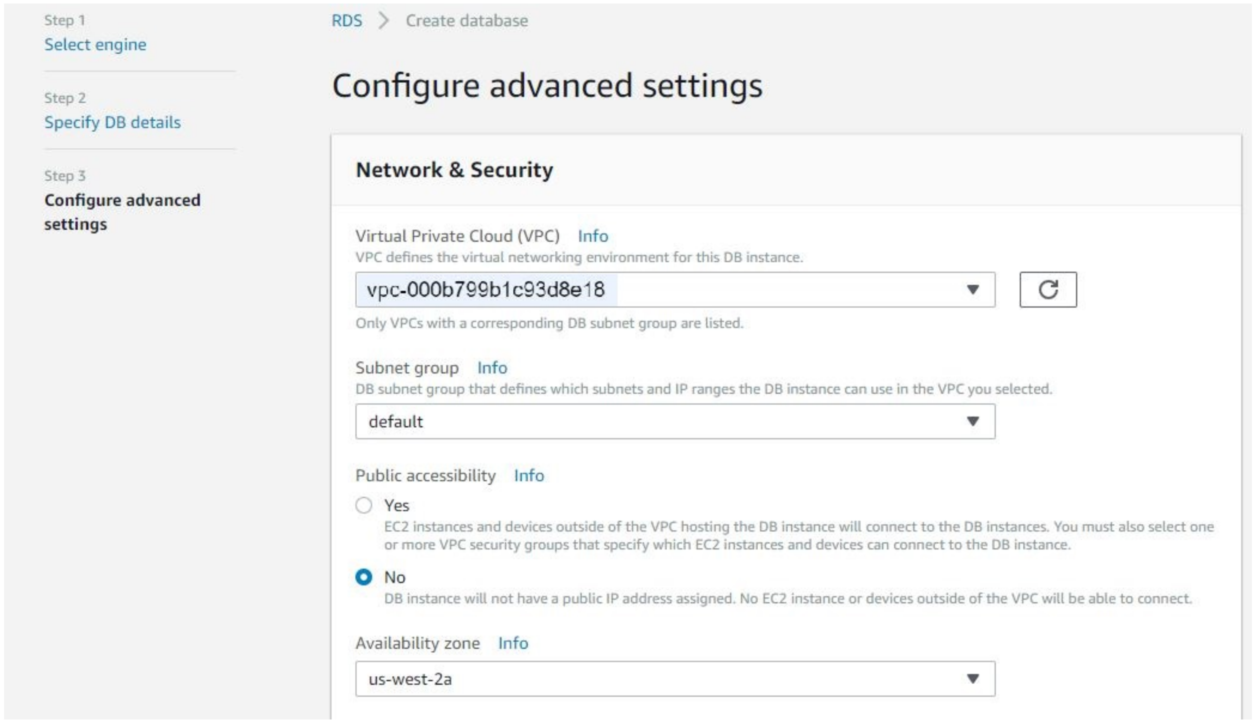
Estimate your monthly costs for the DB Instance using the AWS Simple Monthly Calculator

DB engine
MySQL Community Edition

License model Info
general-public-license

DB engine version Info
MySQL 5.6.40

There should be one RDS in each availability zone. Below is sample configuration.



Configure advanced settings

Network & Security

Virtual Private Cloud (VPC) Info
VPC defines the virtual networking environment for this DB instance.
vpc-000b799b1c93d8e18
Only VPCs with a corresponding DB subnet group are listed.

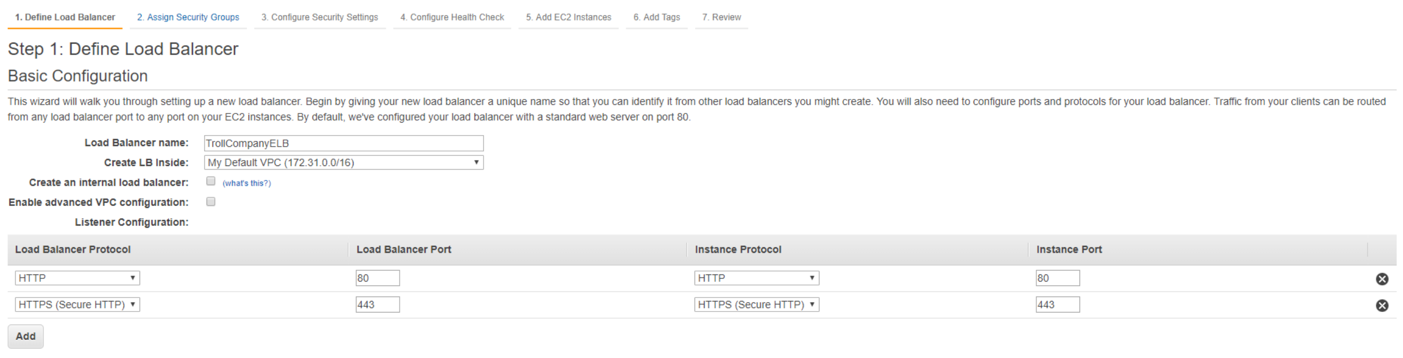
Subnet group Info
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.
default

Public accessibility Info
Yes
EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.
[✓ No]
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

Availability zone Info
us-west-2a

Elastic Load Balancer (ELB)

I recommend spinning up a classic ELB that balances load between the two VPCs. This load balancer should balance load for both HTTP and HTTPS protocol. Below is sample configuration


1. Define Load Balancer
2. Assign Security Groups
3. Configure Security Settings
4. Configure Health Check
5. Add EC2 Instances
6. Add Tags
7. Review

Step 1: Define Load Balancer

Basic Configuration

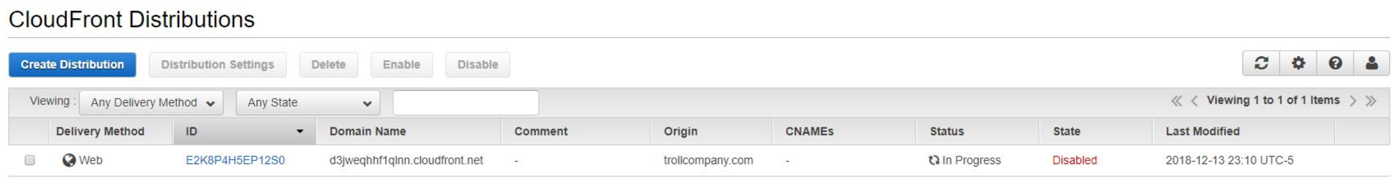
This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

Load Balancer name:	TrollCompanyELB
Create LB Inside:	My Default VPC (172.31.0.0/16)
Create an internal load balancer:	☐ (what's this?)
Enable advanced VPC configuration:	☐
Listener Configuration:	

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	80
HTTPS (Secure HTTP)	443	HTTPS (Secure HTTP)	443

AWS CloudFront

To ensure quick website response time, I recommend spinning up AWS CloudFront. It should deliver content for both HTTP and HTTPS. Below is a sample example.


CloudFront Distributions

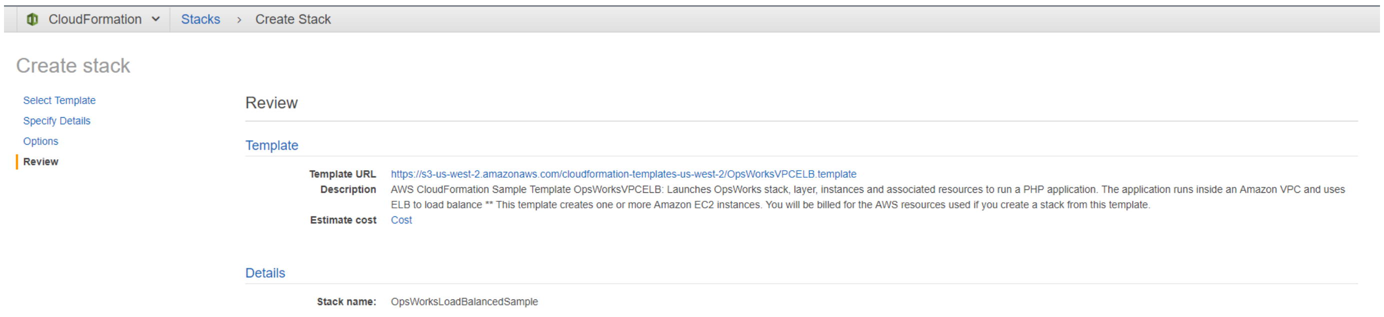
Delivery Method	ID	Domain Name	Comment	Origin	CNAMEs	Status	State	Last Modified
Web	E2K8P4H5EP12S0	d3jweqhhf1qlnn.cloudfront.net	-	trollcompany.com	-	In Progress	Disabled	2018-12-13 23:10 UTC-5

Various Security Features

Resiliency

To ensure resiliency, I had already recommended replicating VPC at two different availability zone. Furthermore, I recommend using AWS Auto Scaling on EC2 instances. This auto scaling can be done using CloudFormation templates like Load-balanced AWS OpsWorks stack in a Amazon VPC or Elastic Beanstalk application in a Amazon Virtual Private Cloud, depending upon technology used to build Troll's website.

Below is sample configuration for Load-Balanced AWS OpsWork stack.



CloudFormation › Stacks › Create Stack

Create stack

- Select Template
- Specify Details
- Options
- **Review**

Review

Template

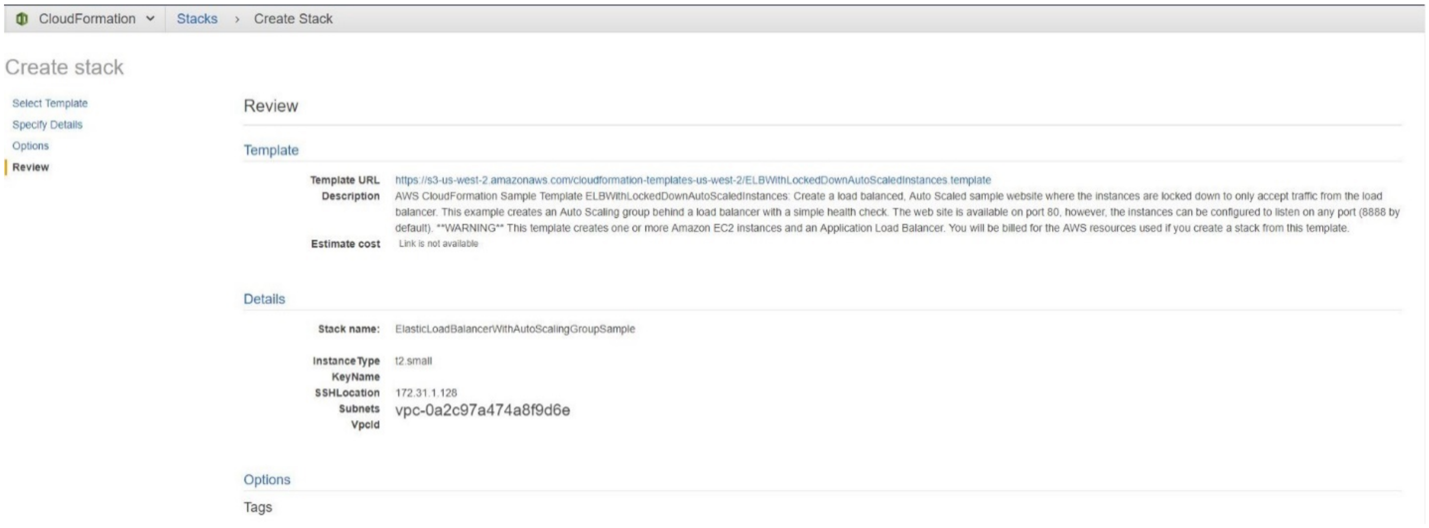
Template URL https://s3-us-west-2.amazonaws.com/cloudformation-templates-us-west-2/OpsWorksVPCELB.template
Description AWS CloudFormation Sample Template OpsWorksVPCELB: Launches OpsWorks stack, layer, instances and associated resources to run a PHP application. The application runs inside an Amazon VPC and uses ELB to load balance ** This template creates one or more Amazon EC2 instances. You will be billed for the AWS resources used if you create a stack from this template.
Estimate cost Cost

Details

Stack name: OpsWorksLoadBalancedSample

AS mentioned earlier, EC2 will have elastic IP. Thus, outage of any one EC2 instance won't hamper overall functionality of the website.

Further, AWS CloudFormation template for Auto-Scaling of ELB can be used. Below is a sample configuration



CloudFormation › Stacks › Create Stack

Create stack

- Select Template
- Specify Details
- Options
- **Review**

Review

Template

Template URL https://s3-us-west-2.amazonaws.com/cloudformation-templates-us-west-2/ELBWithLockedDownAutoScaledInstances.template
Description AWS CloudFormation Sample Template ELBWithLockedDownAutoScaledInstances: Create a load balanced, Auto Scaled sample website where the instances are locked down to only accept traffic from the load balancer. This example creates an Auto Scaling group behind a load balancer with a simple health check. The web site is available on port 80, however, the instances can be configured to listen on any port (8888 by default). **WARNING** This template creates one or more Amazon EC2 instances and an Application Load Balancer. You will be billed for the AWS resources used if you create a stack from this template.
Estimate cost Link is not available

Details

Stack name: ElasticLoadBalancerWithAutoScalingGroupSample

InstanceType t2.small
KeyName
SSHLocation 172.31.1.128
Subnets vpc-0a2c97a474a8f9d6e
VpcId

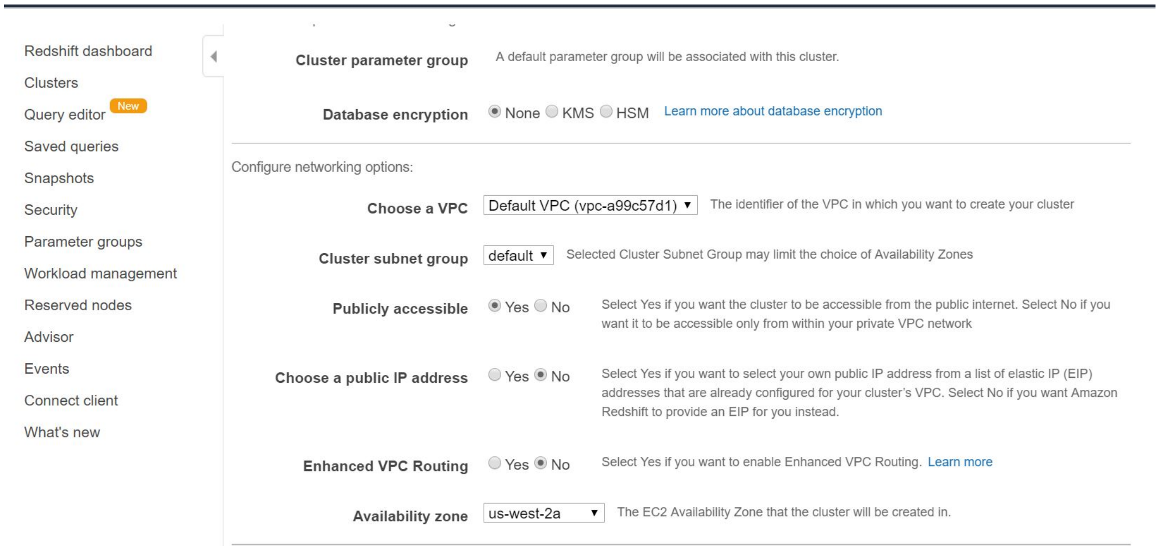
Options

Tags

Data Protection:

To ensure data protection, I recommend using AWS Redshift. This tool monitors VPC instances for data leakage.

A sample configuration can be



Cluster parameter group: A default parameter group will be associated with this cluster.

Database encryption: [✓ None] KMS HSM Learn more about database encryption

Configure networking options:

Choose a VPC: Default VPC (vpc-a99c57d1) The identifier of the VPC in which you want to create your cluster

Cluster subnet group: default Selected Cluster Subnet Group may limit the choice of Availability Zones

Publicly accessible: [✓ Yes] No Select Yes if you want the cluster to be accessible from the public internet. Select No if you want it to be accessible only from within your private VPC network

Choose a public IP address: Yes [✓ No] Select Yes if you want to select your own public IP address from a list of elastic IP (EIP) addresses that are already configured for your cluster's VPC. Select No if you want Amazon Redshift to provide an EIP for you instead.

Enhanced VPC Routing: Yes [✓ No] Select Yes if you want to enable Enhanced VPC Routing. Learn more

Availability zone: us-west-2a The EC2 Availability Zone that the cluster will be created in.

Using steps mentioned at here, I recommend enabling audit logging for RedShift.

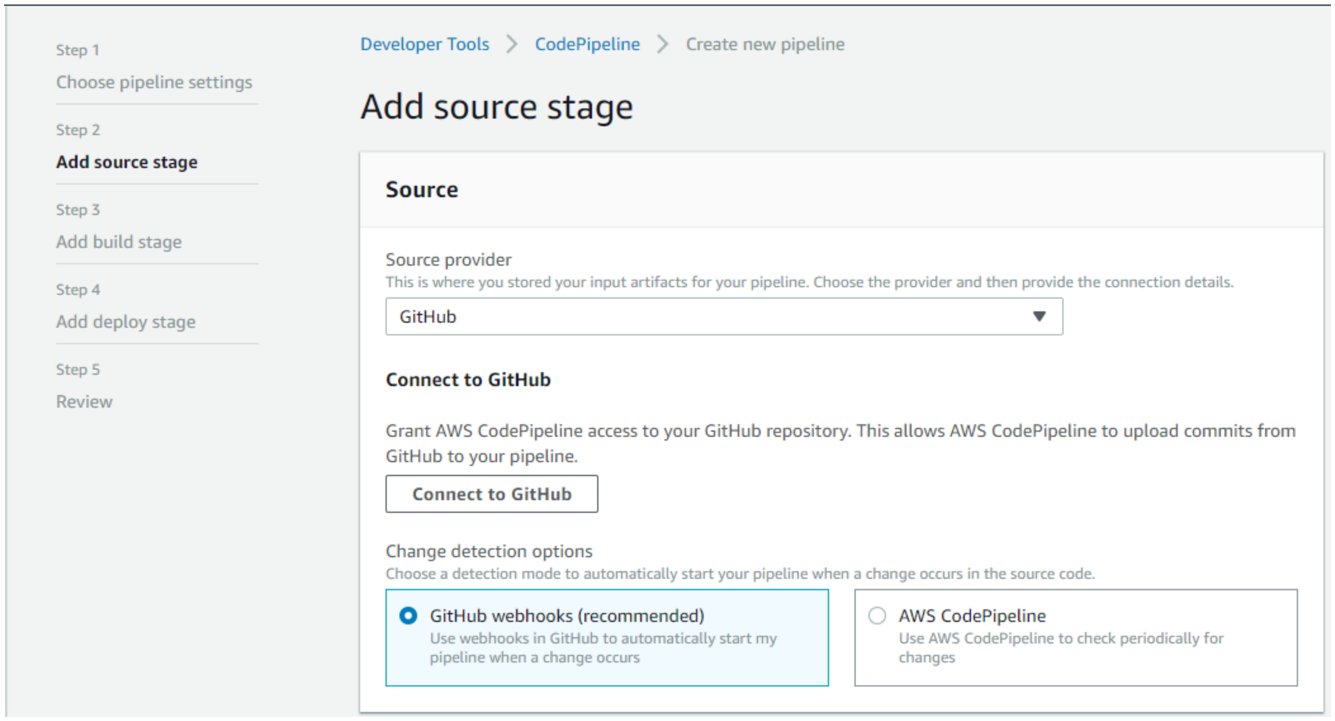
S3 bucket: For S3 buckets, I recommend providing write permissions to a limited number of APIs. Most of APIs should have read-only access. S3 does not entail object -level encryption as it is storing non-critical data, videos. As mentioned earlier, S3 must have AWS CloudTrail enabled. S3 bucket should not be publicly readable/writeable.

For EC2 instances: Using earlier mentioned security-group, will restrict access to EC2 instances. Also, IAM credentials should expire as soon as it compromised. Each EC2 instance must have auto-patching policy enabled.

Secure Coding Practices

All applications installed in this segment must be done from AWS account of user belonging to Developer group. I am assuming this firm uses GitHub as their source code repository.

AWS CodePipeline: I recommend using AWS CodePipeline to automate build, test, and deploy phases of web-application. This has capability to fetch code from GitHub, one of the most common source repositories. Below is a sample configuration form.



Add source stage

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
GitHub

Connect to GitHub

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

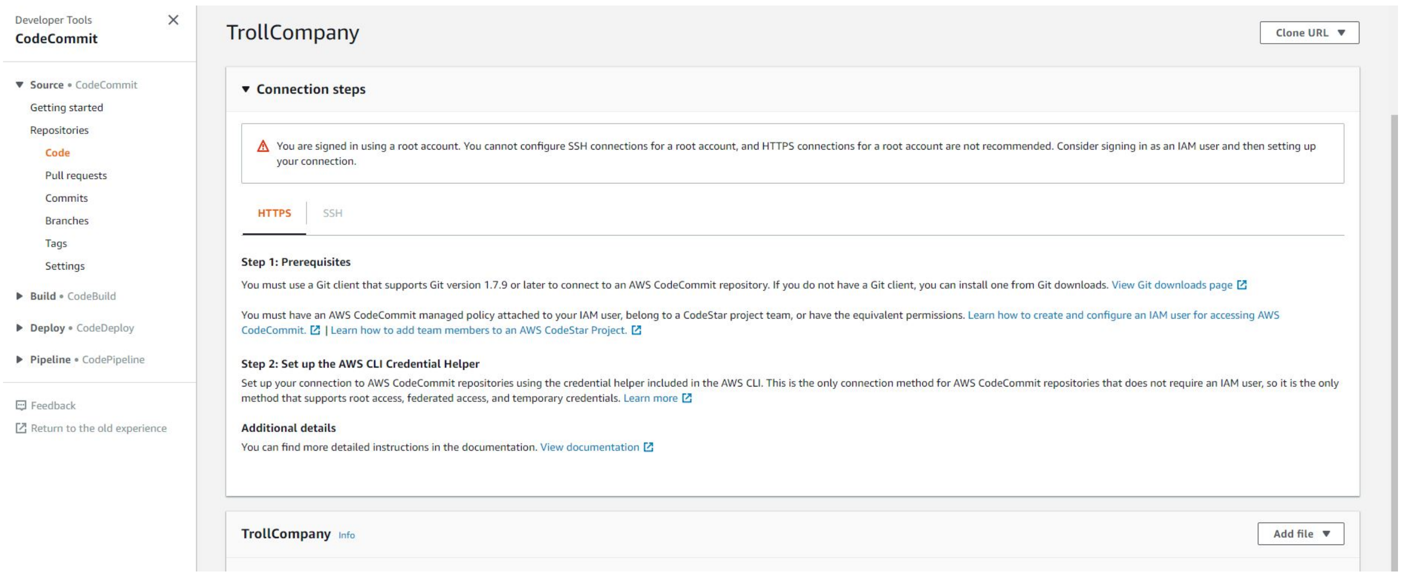
Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

[✓ GitHub webhooks (recommended)] Use webhooks in GitHub to automatically start my pipeline when a change occurs

AWS CodePipeline: Use AWS CodePipeline to check periodically for changes

Furthermore, web developer can configure all build and deploy parameters.

AWS CodeCommit: To ensure seamless collaboration amongst various web-developers, I recommend using AWS CodeCommit. Below is a sample snapshot. Here developers can add GitHub repository details.



TrollCompany

Connection steps

You are signed in using a root account. You cannot configure SSH connections for a root account, and HTTPS connections for a root account are not recommended. Consider signing in as an IAM user and then setting up your connection.

Step 1: Prerequisites
You must use a Git client that supports Git version 1.7.9 or later to connect to an AWS CodeCommit repository. If you do not have a Git client, you can install one from Git downloads. View Git downloads page

You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. Learn how to create and configure an IAM user for accessing AWS CodeCommit. | Learn how to add team members to an AWS CodeStar Project.

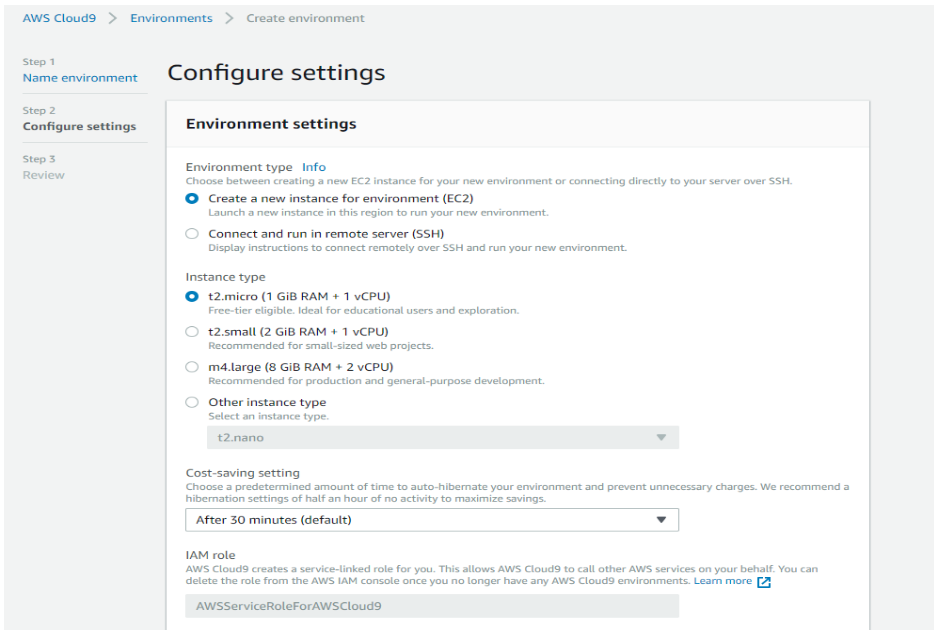
Step 2: Set up the AWS CLI Credential Helper
Set up your connection to AWS CodeCommit repositories using the credential helper included in the AWS CLI. This is the only connection method for AWS CodeCommit repositories that does not require an IAM user, so it is the only method that supports root access, federated access, and temporary credentials. Learn more

Additional details
You can find more detailed instructions in the documentation. View documentation

TrollCompany

AWS Cloud9: This is cloud-based integrated development environment (IDE) that allows developer to write, run, and debug code with just a browser. This tool includes a code editor, debugger, and terminal.

Cloud9 comes pre-packaged with essential tools for popular programming languages including JavaScript, Python, PHP, etc. I recommend using configuration like those mentioned below.



Configure settings

Environment settings

Environment type Info
Choose between creating a new EC2 instance for your new environment or connecting directly to your server over SSH.
● Create a new instance for environment (EC2)
Launch a new instance in this region to run your new environment.
○ Connect and run in remote server (SSH)
Display instructions to connect remotely over SSH and run your new environment.

Instance type
● t2.micro (1 GiB RAM + 1 vCPU)
Free-tier eligible. Ideal for educational users and exploration.
○ t2.small (2 GiB RAM + 1 vCPU)
Recommended for small-sized web projects.
○ m4.large (8 GiB RAM + 2 vCPU)
Recommended for production and general-purpose development.
○ Other instance type
Select an instance type.
t2.nano

Cost-saving setting
Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.
After 30 minutes (default)

IAM role
AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. Learn more
AWSServiceRoleForAWSCloud9

<u>Compliance</u>

PCI Compliance: As the company's web-apps. use credit cards for payment, it must be PCI compliant. All services running at AWS is PCI-DSS compliant. Hence, developer at Trolls need not worry about PCI compliance.

GDPR compliance: GDPR compliance is mandated if the firm has customers from European Union (EU). AWS have many GDPR complaint services. The Trolls can use Amazon Guard Duty to ensure GDPR compliance, in case required.