Dear Lily Sawyer,

Here are some points we consider:

- The company is a startup. We are lucky because we can provide more of a greenfield approach, and you are more likely to be able to make changes to your application stack. Compared to an enterprise, where change can take a long time, and they may end up only being able to migrate part of their system at a time.
- Again, being a startup, it is probably easy to migrate your data into *AWS*.
- you have a single server and already experience downtime, so you are probably OK with some downtime during migration and setup.
- Being a mixed API and web app they you need to host some static content as well.

I Have Two Solution's For You

In this case,

Our First solution based on Elastic Beanstalk based on these requirements:

- You has a Python application that Elastic Beanstalk has support for.
- You will need to scale, and this is one of the fundamental features of Elastic Beanstalk.
- You want to remove downtime when deploying and Elastic Beanstalk's Blue/Green deployment can assist with this.

Other options like Lambda or Fargate can be more complex to set up. Lightsail and EC2 can be easy but they don't provide that automatic scaling;

Background Information

- Route 53 - AWS' domain and DNS management.

- Elastic Load Balancing - Distributes traffic across application servers, such as EC2, Lambda or Fargate. Can use health checks to know which servers should service requests. Charges are based on the number of hours the load balancer runs and (at a high level) the amount of traffic it services. The actual charging rules can be quite complex.

- Elastic Beanstalk with Autoscaling EC2 Group - Ties together EC2, RDS and Elastic Load Balancing with simple configuration and deployment. Its primary advantage is

how it can facilitate the autoscaling of EC2 instances. Billing is based on a combination of the EC2, RDS and ELB that you use. Deployment is made easy with CLI tools, and we can use rolling deployments so there is no downtime. It has support for several languages including Python, NodeJS, Java and Go.

- RDS - Relational database hosting platform. Charged in the same way as EC2 (i.e. a virtual machine with a set amount of resources). Servers can be resized but must be restarted to do so.

- S3 - Store objects in the cloud. Charged based on the amount of data being stored, how it's stored, and for retrieval.

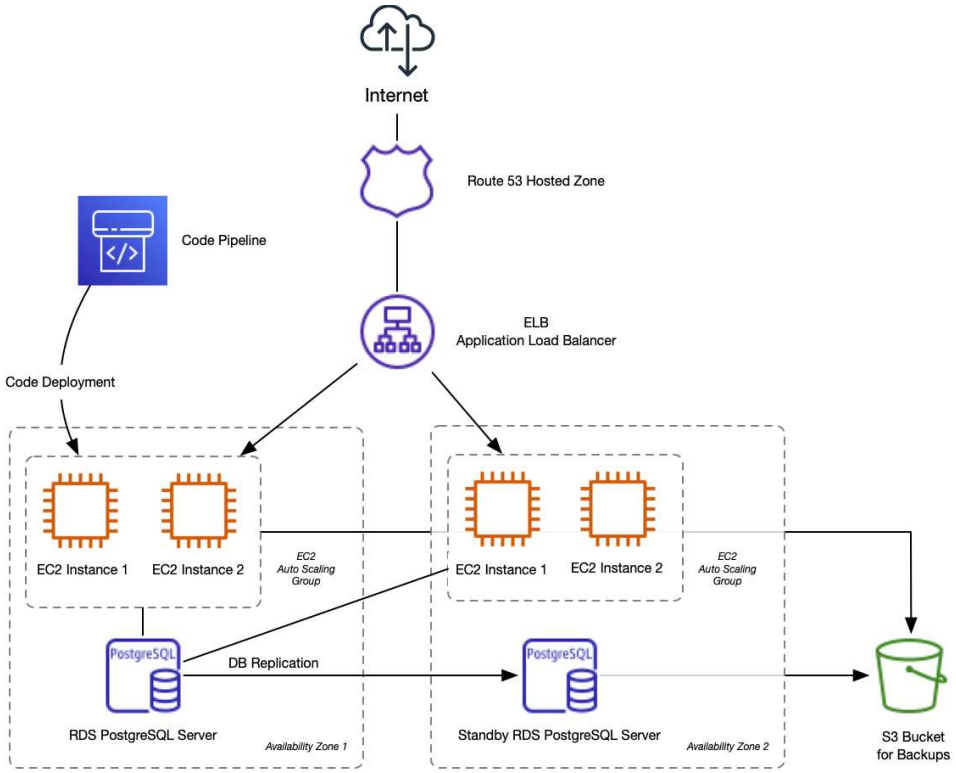
- CodePipeline - This will build code and can deploy it to various AWS services. It is charged per pipeline. Can be used with Elastic Beanstalk for blue/green no-downtime deployments.

The services we will use are:

- Route 53
- Elastic Load Balancing
- Elastic Beanstalk with Autoscaling EC2 Group
- RDS
- S3
- CodePipeline

The concrete pricing differs in each region, however, the method of price calculation is the same. For example, an EC2 instance in Region A may cost more than the same instance in Region B, however, both are billed at an hourly rate. The AWS pricing calculator at https://calculator.aws/ can also give some guidance.

The following diagram illustrates our solution architecture:



Explanation

Route 53 Hosted between Internet And ELB
Then Code Pipeline Deploy Model to AWS Compute Services EC2
Ec2 Instances are putted in Auto Scaling EC2
Ec2 conntected to RDS for AWS Database Service
Ec2 and RDS sending data to s3 Bucket For Backup

Our Second solution based on Elastic Beanstalk based on these requirements:

- You has a Python application that Elastic Beanstalk has support for.
- You will need to scale, and this is one of the fundamental features of Elastic Beanstalk.
- You want to remove downtime when deploying and Elastic Beanstalk's Blue/Green deployment can assist with this.

Other options like Lambda or Fargate can be more complex to set up. Lightsail and EC2 can be easy but they don't provide that automatic scaling;

Background information

- Elastic Load Balancing - Distributes traffic across application servers, such as EC2, Lambda or Fargate. Can use health checks to know which servers should service requests. Charges are based on the number of hours the load balancer runs and (at a high level) the amount of traffic it services. The actual charging rules can be quite complex.

- Elastic Beanstalk with Autoscaling EC2 Group - Ties together EC2, RDS and Elastic Load Balancing with simple configuration and deployment. Its primary advantage is how it can facilitate the autoscaling of EC2 instances. Billing is based on a combination of the EC2, RDS and ELB that you use. Deployment is made easy with CLI tools, and we can use rolling deployments so there is no downtime. It has support for several languages including Python, NodeJS, Java and Go.

- RDS - Relational database hosting platform. Charged in the same way as EC2 (i.e. a virtual machine with a set amount of resources). Servers can be resized but must be restarted to do so.

- S3 - Store objects in the cloud. Charged based on the amount of data being stored, how it's stored, and for retrieval.

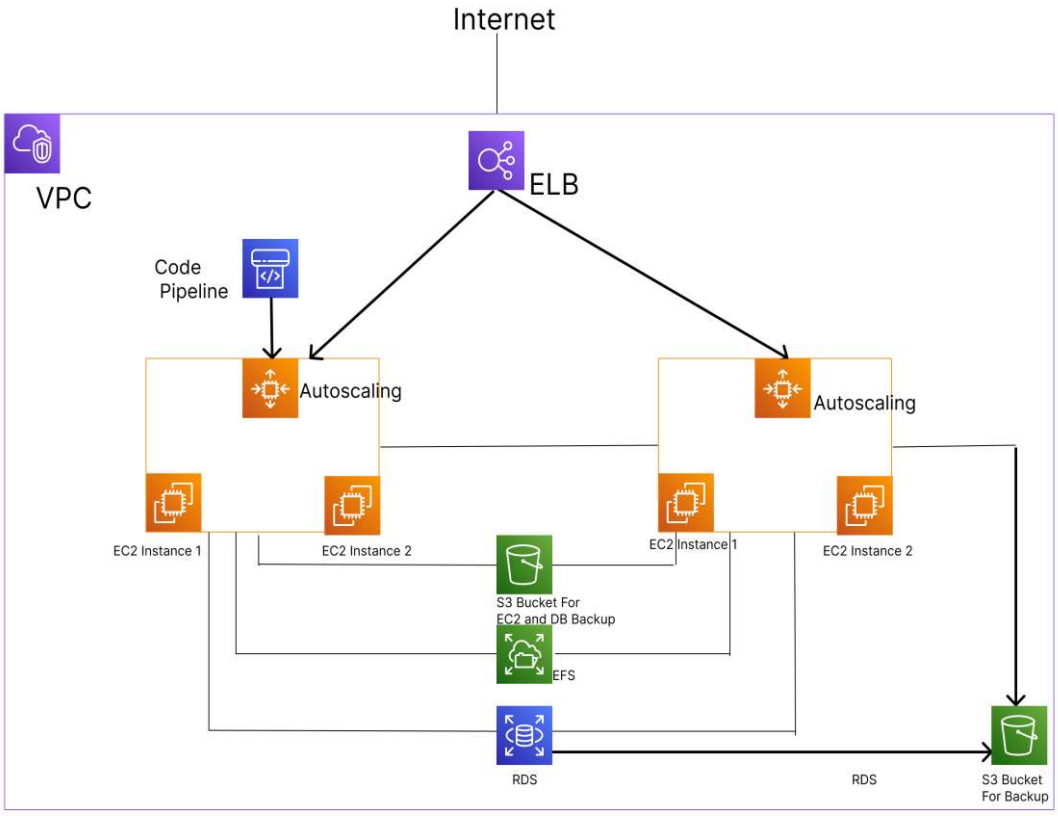
- CodePipeline - This will build code and can deploy it to various AWS services. It is charged per pipeline. Can be used with Elastic Beanstalk for blue/green no-downtime deployments.

The services we will use are:

- VPC
- Elastic Load Balancing
- Elastic Beanstalk with Autoscaling EC2 Group
- EFS
- RDS
- S3
- CodePipeline

The concrete pricing differs in each region, however, the method of price calculation is the same. For example, an EC2 instance in Region A may cost more than the same instance in Region B, however, both are billed at an hourly rate. The AWS pricing calculator at https://calculator.aws/ can also give some guidance.

The following diagram illustrates our solution architecture:



Explanation

Virtual Private Cloud Hosted between Internet And ELB
Then Code Pipeline Deploy Model to AWS Compute Services EC2
Ec2 Instances are putted in Auto Scaling EC2
Ec2 Conntected to RDS for AWS Database Service
Ec2 Conntected to EFS for enables users to control who can read the files on their system.
Ec2 Conntected to S3 for EC2 & DB Backup
Ec2 and RDS sending data to s3 Bucket For Backup & Secure in VPC

Best regards,

Sahil Negi