**What is scalability:**

Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands. I will describe how to achieve a scalable and high-availability WordPress Architecture using AWS resources. WordPress is a feature-rich CMS, which is extremely popular. We will use the key AWS services such as Elastic ache, EFS, CloudFront, and Load Balancer to make it scalable and fault tolerant.

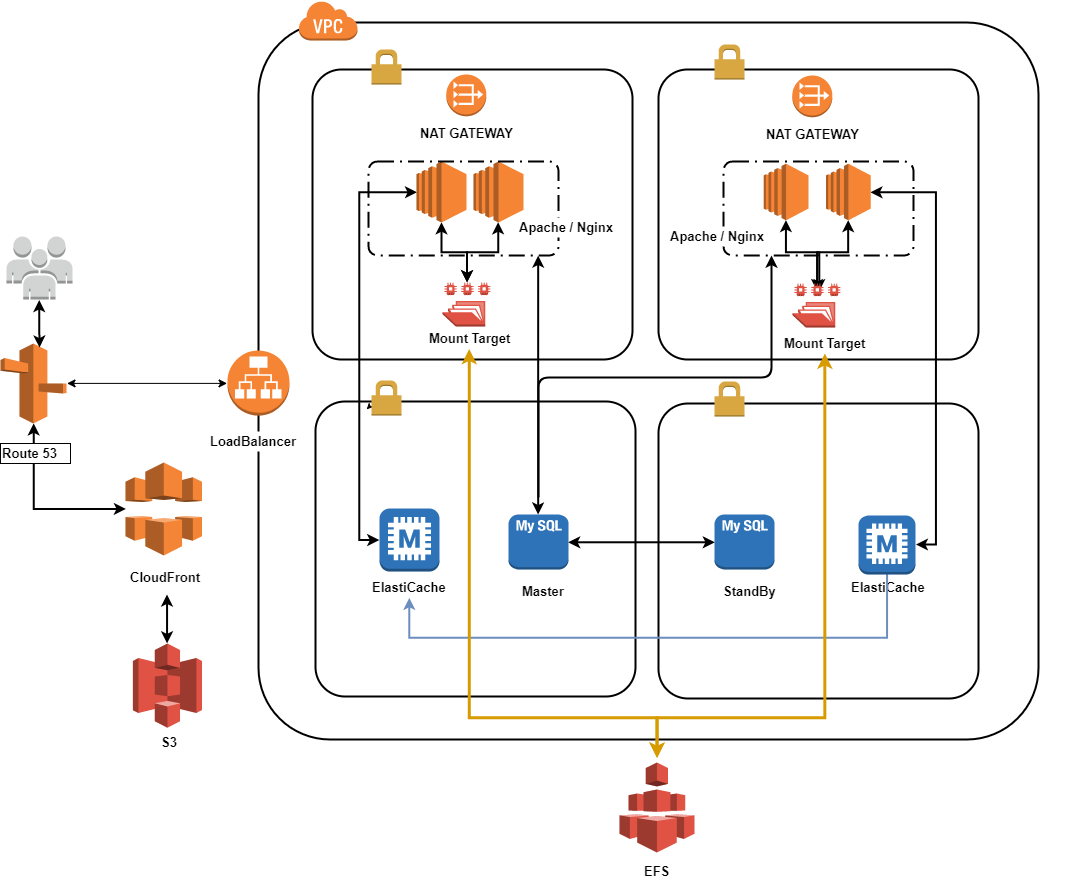
**Website Architecture:**

A good website architecture is to provide the below principles:

1. **Elasticity** – We want our website to scale up and down.
2. **Loose Coupling** – We will be distributing our services across multiple AWS products –
   * The database will be on **AWS Aurora.**
   * Servers on **EC2.**
   * Storage on **EFS.**

**Architecture Diagram:**

 The Architecture Diagram of WordPress is given below in detail:



The detail is given below:

We will register our **Application Load Balancer** with Route **53.**  **CloudFront** will be fetching our uploads from the **S3.**To push your media files to S3, you would need to use the [AWS Plugin](https://wordpress.org/plugins/amazon-web-services/). Since the application is stored in numerous instances, we would need to save the WordPress files on **EFS**. EFS is a network file storage that can be mounted on multiple devices. This makes updating WordPress files extremely easy. As there is always a chance of an instance failing, we would use **ElastiCache** Memcached to store the sessions. The database can be either Aurora or RDS. We will be using a Master and Read replica in case one of the database insistencies, won’t have an impact on the website.

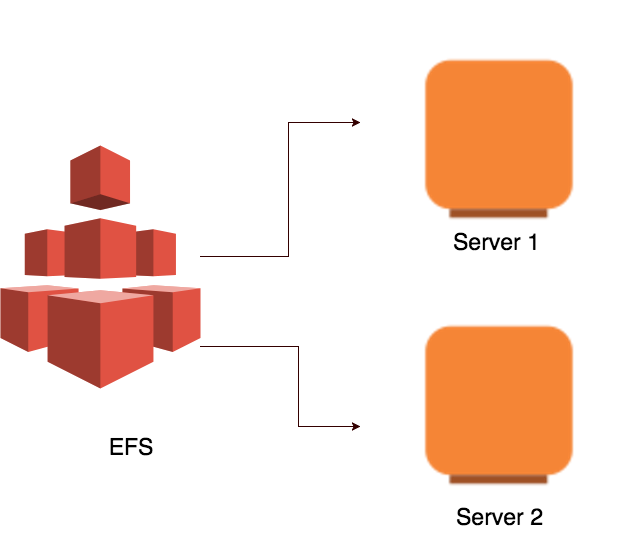
**Why store files on S3?**

There are many benefits to storing your media files in S3.

* Easy website migration. By having files stored on S3 website migration can be done effortlessly.
* Saves storage space. WordPress converts an image into multiple resolutions. This could become troublesome if you have a heavily used website. S3 can store an unlimited number of files. By saving the media files on S3 you will save space.

**Why** [**EFS**](https://www.infinitypp.com/amazon-aws/ebs-volume-vs-efs-volume-for-aws-storage/)**?**

The EFS distributes your application code across multiple EC2 instances. This makes the deployment of code very efficient. Instead of deploying the application code to multiple instances, it would need to push it to a single instance which has the EFS mounted on it. Moreover, installing themes and plugins can be done on a single instance and the changes get reflected across all other mount targets!



**Increasing the performance of EFS:**

To increase the performance, Bytecode caching must be used. By default, PHP 5.5 comes with OpCache. OpCache caches the compiled PHP code in memory, so the script doesn’t have to be compiled again. When a PHP file is updated, the cached compiled code is updated as well. OpCache can reduce the execution time by 70%.

**Scalability:**

It uses an Application Load Balancer (Layer 7), and the number of instances can scale depending on the CPU, RAM usage, or scheduled plan. This means we don’t need to worry about the amount of traffic our servers receive. To register for a health check, there are many techniques that you could use. You could have a health check to the wp-login page or a custom PHP file that invokes a WordPress method such as retrieving the WordPress site URL.

**Instance types:**

AWS has many different types of instances. But while using more than one instance this becomes very flexible. If you’re happy with the instance type try switching it to a reserved instance plan. With the Reserved instance plan, you could be saving up to 75% of your EC2 billing!

**Database types:**

WordPress like every other CMS relies on a database. This is why selecting the right database is extremely important in architectural design. AWS provides two different types of databases AWS RDS and AWS Aurora.

1. **AWS Aurora:** It is designed and developed by AWS to provide a very cost-effective database while offering the five-time throughput of MySQL. In addition, it copies the data across three availability zone and on SSD. By using AWS Aurora, you will **save cost** and provide a better-tuned and **scalable** database.
2. **AWS RDS:** It is a managed MySQL that has been out there for a long time. A key advantage of AWS RDS is the option to select the type of SSD storage device.

# **Scaling the Web Tier:**

To evolve your single-server architecture into a multi-server, scalable architecture, you must use five key components:

* **Amazon EC2 instances**
* **Amazon Machine Images (AMIs)**
* **Load balancers**
* **Automatic scaling**
* **Health checks**

AWS provides a wide variety of EC2 instance types so you can choose the best server configuration for both performance and cost. Generally speaking, the compute-optimized (for example, C4) instance type may be a good choice for a WordPress web server. It can deploy your instances across multiple Availability Zones within an AWS Region to increase the reliability of the overall architecture.

Because it has complete control of your EC2 instance, it can log in with root access to install and configure all of the software components required to run a WordPress website. After that, it can save that configuration as an AMI, which use to launch new instances with all the customizations that are made.

AWS provides this capability through [Elastic Load Balancing](http://aws.amazon.com/elasticloadbalancing/) a highly available service that distributes traffic to multiple EC2 instances. Because the website is serving content to your users via HTTP or HTTPS, it was recommended that we make use of the Application Load Balancer, an application-layer load balancer with content routing and the ability to run multiple WordPress websites on different domains, if required.

Elastic Load Balancing supports the distribution of requests across multiple Availability Zones within an AWS Region. You can also configure a health check so that the Application Load Balancer automatically stops sending traffic to individual instances that have failed. AWS recommends using the WordPress admin login page for the health check because this page confirms both that the web server is running and that the web server is configured to serve PHP files correctly.

[**AWS Auto Scaling**](http://aws.amazon.com/autoscaling/)**:**

Elasticity is a key characteristic of the AWS Cloud. It is an AWS service that helps to automate this provisioning to scale your Amazon EC2 capacity up or down according to conditions you define with no need for manual intervention. It can configure AWS Auto Scaling so that the number of EC2 instances that are used increases seamlessly during demand spikes to maintain performance and decreases automatically when traffic diminishes, to minimize costs.

**Elastic Load Balancing** also supports the dynamic addition and removal of Amazon EC2 hosts from the load-balancing rotation. Elastic Load Balancing itself also dynamically increases and decreases the load-balancing capacity to adjust to traffic demands with no manual intervention.