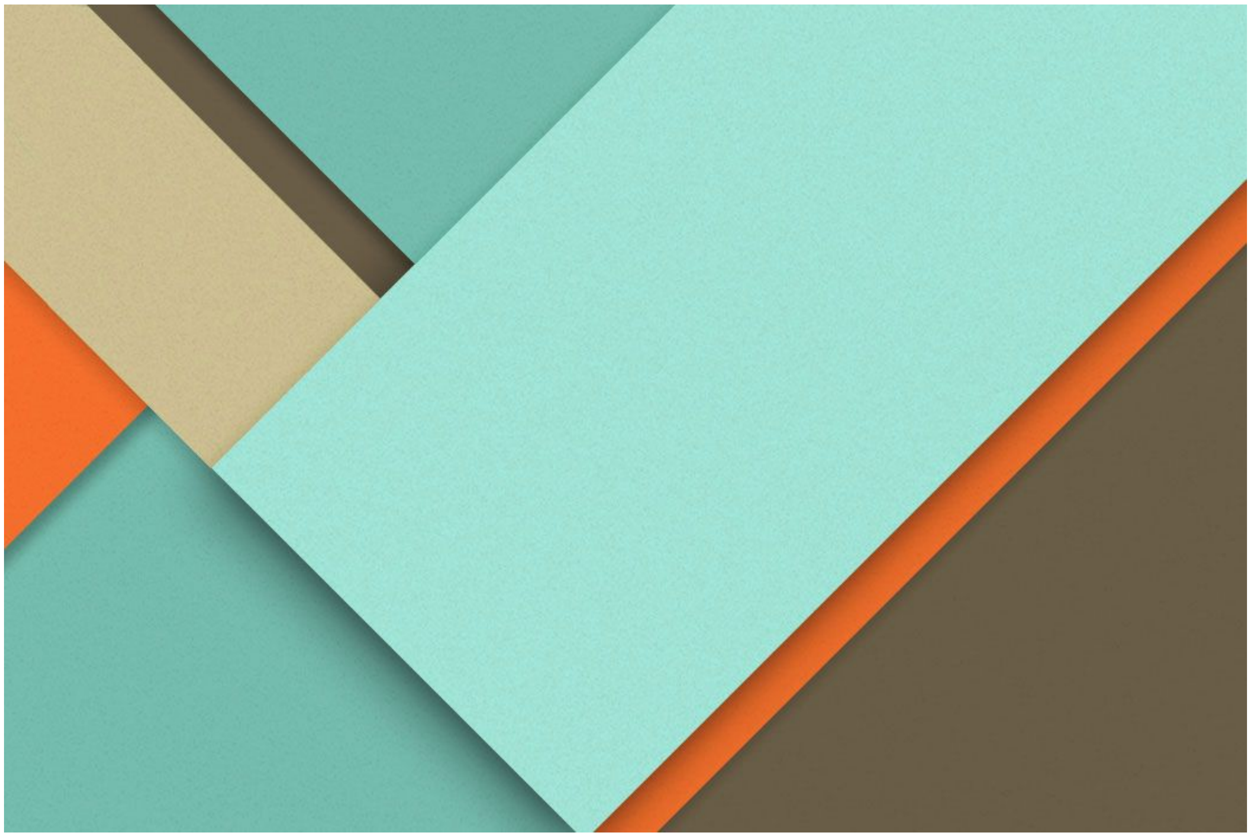


Running services on AWS spot instances without interruption

12.07.2019

Hari Boddu

Overview

Cloud computing is the on-demand delivery of compute power, database, storage, applications, and other IT resources via the internet with pay-as-you-go pricing.Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud.

Amazon EC2:

Amazon Elastic Compute Cloud (EC2) provides virtual servers -- called instances -- for compute capacity. The EC2 service offers dozens of instance types with varying capacities and sizes, tailored to specific workload types and applications, such as memory-intensive and accelerated-computing jobs. AWS also provides an Auto Scaling tool to dynamically scale capacity to maintain instance health and performance.

An *Amazon Machine Image (AMI)* is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch an *instance*, which is a copy of the AMI running as a virtual server in the cloud.Just like renting or leasing a house, when you pay for servers(instances) from AWS, there are different options.

1.On-Demand :

With On-Demand instances, you pay for compute capacity by per hour or per second depending on which instances you run. No long-term commitments or upfront payments are needed. You can increase or decrease your compute capacity depending on the demands of your application and only pay the specified per hourly rates for the instance you use.

2.Reserved:

A Reserved Instance is a reservation of resources and capacity, for either one or three years, for a particular Availability Zone within a region. Unlike on-demand, when you purchase a reservation, you commit to paying for *all* of the hours of the 1- or 3-year term; in exchange, the hourly rate is lowered significantly.

3.Spot:

Amazon EC2 Spot Instances let you take advantage of unused EC2 capacity in the AWS cloud. Spot Instances are available at up to a 90% discount compared to On-Demand prices.

On first look,it might occur that spot is the best option because of the 90% discount offered.But the problem with choosing spot instances is,With spot instances, the commitment concept gets reversed. Instead of the lack of a commitment benefiting you, it applies the other way to AWS. With on-demand, you do not commit to AWS. With spot, AWS does not commit to you.Spot Instances can be interrupted by Amazon EC2 with two minutes of notification when EC2 needs the capacity back.

How do we run our services?

ECS cluster:

An Amazon ECS cluster is a logical grouping of tasks or services. If you are running tasks or services that use the EC2 launch type, a cluster is also a grouping of container instances.

Amazon Elastic Container Service (Amazon ECS) is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster. you can host your tasks on a cluster of Amazon Elastic Compute Cloud (Amazon EC2) instances that you manage by using the EC2 launch type. Amazon ECS lets you launch and stop container-based applications with simple API calls, allows you to get the state of your cluster from a centralized service, and gives you access to many familiar Amazon EC2 features.

Docker is a technology that allows you to build, run, test, and deploy distributed applications that are based on Linux containers. Amazon ECS uses Docker images in task definitions to launch containers on Amazon EC2 instances in your clusters . Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime.

Till know we have explored about ECS cluster and different instance purchase options.

The immediate idea would be to use ec2 spot instances for the cluster because they are cheap.So we need request a lot of instances depending on the requirements for the ecs cluster.And there is an easy and better way to do this.

Spot Fleet:

A *Spot Fleet* is a collection, or fleet, of Spot Instances, and optionally On-Demand Instances.

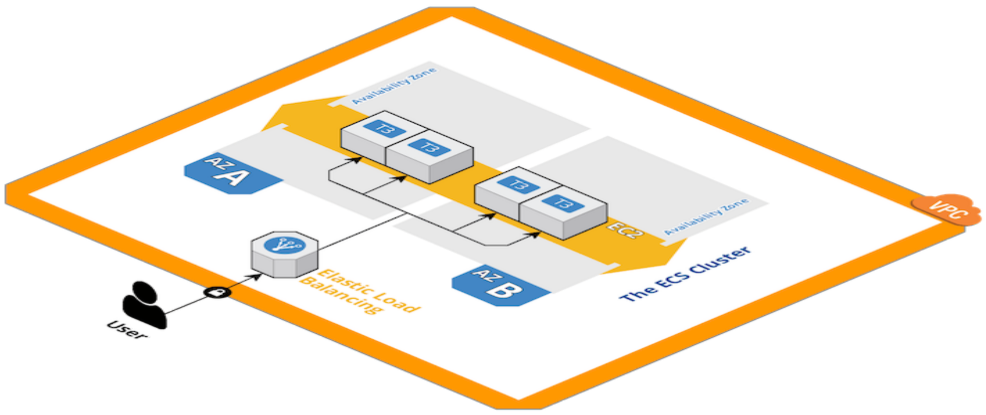
The Spot Fleet attempts to launch the number of Spot Instances and On-Demand Instances to meet the target capacity that you specified in the Spot Fleet request. The request for Spot Instances is fulfilled if there is available capacity and the maximum price you specified in the request exceeds the current Spot price. The Spot Fleet also attempts to maintain its target capacity fleet if your Spot Instances are interrupted.Using Spot Fleets dramatically makes the system more robust

ECS and Spot Fleets: A Great Fit Together

Docker provides a consistent, standard binary format to deploy. If it works in one Docker environment, then it works in another. Containers can be pulled down in seconds, making them an excellent fit for Spot Instances, where containers might move around during an interruption.

ECS provides a great ecosystem to run Docker containers. ECS supports a feature called container instance draining that allows me to tell ECS to relocate the Docker containers to other EC2 instances.

Spot Instances fire off a two-minute warning signal letting me know when it's about to terminate the instance.



Connection instance Draining:

There are times when you might need to remove a container instance from a cluster;for example, to perform system updates, scale down the cluster size or if a spot instance is going to be terminated.Container instance draining enables you to remove a container instance from a cluster without impacting tasks in your cluster.

When you set a container instance to DRAINING, Amazon ECS prevents new tasks from being scheduled for placement on the container instance. Service tasks on the draining container instance that are in the PENDING state are stopped immediately. If there are container instances in the cluster that are available, replacement service tasks are started on them.

Service tasks on the container instance that are in the RUNNING state are stopped and replaced according to the service's deployment configuration parameters.

So everything looks fine,we receive a spot instance termination notice and drain that particular instance.

But!!!

2 minutes might not be the ideal interval to smoothly transition our services without any disruption.

Solution:AWS Lambda Function and an EC2 Auto Scaling Group

AWS Lambda Function:

AWS Lambda is a compute service that lets you run code without provisioning or managing servers. AWS Lambda executes your code only when needed and scales automatically, from a few requests per day to thousands per second. You pay only for the compute time you consume - there is no charge when your code is not running. With AWS Lambda, you can run code for virtually any type of application or backend service - all with zero administration.

Auto Scaling Group(ASG):

An *Auto Scaling group* contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. An Auto Scaling group also enables you to use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies. Both maintaining the number of instances in an Auto Scaling group and automatic scaling are the core functionality of the Amazon EC2 Auto Scaling service.

The size of an Auto Scaling group depends on the number of instances you set as the desired capacity. You can adjust its size to meet demand, either manually or by using automatic scaling.

We wrote a lambda function and set a cron expression so that it runs periodically every 5 minutes.

What does our AWS lambda code do?

We check the spot price of the instance types and compare it with our predefined maximum price set.

3 cases arise:

1.Spot price <70% of Max price

2.Spot price between 70-90% of Max price:

3.Spot >90% of Max price

Case 1 :

Set the spot fleet capacity to maximum and drain any on demand instance present in the cluster and set on demand asg capacity to zero.

Case 2:

Do Nothing

Case 3 :

Drain any spot instance present in the cluster.Set the spot fleet capacity to zero and on demand asg capacity to maximum.

Basically we are checking the possibility of a spot instance getting terminated and terminating those instances before hand and proving backup(ASG) so that we can have an extra amount of time to transition everything rather than the 2 minutes provided by AWS.

So in this way we will get a proper amount of time interval to seamlessly transition our services without any disruption.