AWS Whitepaper

Overview of Amazon EC2 Spot Instances

Overview of Amazon EC2 Spot Instances: AWS Whitepaper

Table of Contents

Overview of Amazon EC2 Spot Instances

Publication date: **March 5, 2021** (*Document History and Contributors*)

Abstract

This paper seeks to empower you to maximize value from your investments, improve forecasting accuracy and cost predictability, create a culture of ownership and cost transparency, and continuously measure your optimization status.

This paper provides an overview of Amazon EC2 Spot Instances, as well as best practices for using them effectively.

Introduction

In addition to On-Demand, Reserved Instances, and Savings Plans, the fourth Amazon Elastic Compute Cloud (Amazon EC2) pricing model is Spot Instances.

With Spot Instances, you can use spare Amazon EC2 computing capacity at discounts of up to 90% compared to On-Demand pricing. That means you can significantly reduce the cost of running your applications, or grow your application's compute capacity and throughput for the same budget. The only difference between On-Demand Instances and Spot Instances is that Spot Instances can be interrupted by EC2 with two minutes of notification when EC2 needs the capacity back.

Unlike Reserved Instances or Savings Plans, Spot Instances do not require a commitment in order to achieve cost savings over On-Demand pricing. However, because Spot Instances can be terminated by EC2 if there is no available capacity in the capacity pool (a combination of an instance type and an Availability Zone) in which they are running, they are best suited for flexible workloads.

When to Use Spot Instances

You can use Spot Instances for various fault-tolerant and flexible applications. Examples include stateless web servers, API endpoints, big data and analytics applications, containerized workloads, CI/CD high performance and high throughput computing (HPC/HTC), rendering workloads, and other flexible workloads.

Spot Instances are not suitable for workloads that are inflexible, stateful, fault-intolerant, or tightly coupled between instance nodes. Spot Instances are also not recommended for workloads that are intolerant of occasional periods when the target capacity is not completely available. We strongly warn against using Spot Instances for these workloads or for attempting to fail-over to On-Demand Instances to handle interruptions.

How to Launch Spot Instances

The most recommended service for launching Spot Instances is Amazon EC2 Auto Scaling because it enables you to launch and maintain a desired capacity, and to automatically request resources to replace any that are disrupted or manually terminated. When you configure an Auto Scaling group, you only need to specify the instance types and desired capacity based on your application needs. For more information, see Auto Scaling groups in the *Amazon EC2 Auto Scaling User Guide.*

If you require more flexibility, have built your own instance launch workflows, or want to control individual aspects of the instance launches or the scaling mechanisms, we recommend that you evaluate the use of EC2 Fleet in Instant mode as an alternative to EC2 Auto Scaling. This synchronous API allows you to specify a list of instance types and launch requirements, and provides more flexible capability than the EC2 RunInstances API call for launching Spot Instances or On-Demand Instances.

When you use AWS services for running your cloud workloads, you can also use them for launching Spot Instances. Examples include Amazon EMR, Amazon EKS, Amazon ECS, AWS Batch, and AWS Elastic Beanstalk. You can also launch Spot Instances by using third-party tools that integrate with the AWS cloud.

You can automate Spot Instance launches by using infrastructure as code tools (AWS CloudFormation, AWS CDK), or the AWS API, CLI, or SDKs. Spot Blueprints provides a guided wizard that enables you to generate infrastructure as code templates for AWS Cloudformation and Hashicorp terraform that adhere to Spot best practices.

How Spot Instances Work

Spot Instances perform exactly like other EC2 instances while running. However, they can be interrupted by Amazon EC2 when EC2 needs the capacity back.

When EC2 interrupts your Spot Instance, it either terminates, stops, or hibernates the instance, depending on the interruption behavior that you choose.

If EC2 interrupts your Spot Instance in the first hour, before a full hour of running time, you're not charged for the partial hour used. However, if you stop or terminate your Spot Instance, you pay for any partial hour used (as you do for On-Demand or Reserved Instances). For information about how you're billed for interrupted Spot Instances running on different operating systems, see Billing for interrupted Spot Instances in the *EC2 User Guide*.

The Spot price for each instance type in each Availability Zone is determined by long-term trends in supply and demand for EC2 spare capacity. You pay the Spot price that's in effect, billed to the nearest second.

You can optionally specify a maximum price for your Spot Instance. If you don't specify a maximum price, the maximum price defaults to the On-Demand price. Note that you never pay more than the Spot price that is in effect when your Spot Instance is running. We recommend that you do not specify a maximum price, but rather let the maximum price default to the On-Demand price. A high maximum price does not increase your chances of launching a Spot Instance and does not decrease your chances of having your Spot Instance interrupted (because EC2 can still interrupt your Spot Instance when it needs the capacity back).

The Spot price for an instance type in an Availability Zone might change at any time, but in general, it does not change frequently. AWS publishes the current Spot price and the historical prices for Spot Instances through the DescribeSpotPriceHistory API, as well as in the AWS Management Console, which reflects the data from the API. This can help you assess the levels and timing of fluctuations in the Spot price over time.

Managing Spot Instance Interruptions

The best way for you to gracefully handle Spot Instance interruptions and minimize impact on your performance or availability is to architect your application to be fault-tolerant. To accomplish this, you can take advantage of EC2 instance rebalance recommendations and Spot Instance interruption notices.

An EC2 instance rebalance recommendation is a signal that notifies you when a Spot Instance is at elevated risk of interruption. The signal gives you the opportunity to proactively manage the Spot Instance in advance of the two-minute Spot Instance interruption notice. You can decide to rebalance your workload to new or existing Spot Instances that are not at an elevated risk of interruption. We've made it easy for you to use this signal by providing the Capacity Rebalancing feature in EC2 Auto Scaling groups. For more information, see Amazon EC2 Auto Scaling Capacity Rebalancing.

A Spot Instance interruption notice is a warning that is issued two minutes before Amazon EC2 interrupts a Spot Instance. If your workload is "time-flexible," you can configure your Spot Instances to be stopped or hibernated, instead of being terminated, when they are interrupted. Amazon EC2 automatically stops or hibernates your Spot Instances on interruption, and automatically resumes the instances when we have available capacity.

You can use the EC2 instance rebalance recommendation and/or the Spot Instance interruption notice to architect your workload with fault-tolerance in mind, so that you can capture notifications and save a job’s state to storage (for example, Amazon S3, Amazon EFS, or Amazon FSx), persist log files from the instance (or stream them continuously for a more fault-tolerant approach), drain connections from a Load Balancer, etc.

Some AWS and third-party services already handle Spot interruptions for you to decrease the impact on your application. For example, Amazon EKS running managed node groups with Spot Instances automatically launches replacement Kubernetes nodes when a rebalance recommendation or interruption notices are delivered for an existing node.

Spot Instance Limits

There is a limit on the number of running and requested Spot Instances per AWS account per Region. Spot Instance limits are managed in terms of the *number of virtual central processing units (vCPUs)* that your running Spot Instances are either using or will use pending the fulfillment of open Spot Instance requests. If you terminate your Spot Instances but do not cancel the Spot Instance requests, the requests count against your Spot Instance vCPU limit until Amazon EC2 detects the Spot Instance terminations and closes the requests.

There are seven Spot Instance limits:

- All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests
- All DL Spot Instance Requests
- All F Spot Instance Requests
- All G and VT Spot Instance Requests
- All Inf Spot Instance Requests
- All P Spot Instance Requests
- All X Spot Instance Requests

Each limit specifies the vCPU limit for one or more instance families. For information about the different instance families, generations, and sizes, see Amazon EC2 Instance Types.

With vCPU limits, you can use your limit in terms of the number of vCPUs that are required to launch any combination of instance types that meet your changing application needs. For example, say your All Standard Spot Instance Requests limit is 256 vCPUs, you could request 32 `m5.2xlarge` Spot Instances (32 x 8 vCPUs) or 16 `c5.4xlarge` Spot Instances (16 x 16 vCPUs), or a combination of any Standard Spot Instance types and sizes that total 256 vCPUs.

For more information, see Monitor Spot Instance limits and usage and Request a Spot Instance limit increase in the *Amazon EC2 User Guide for Linux Instances*.

Spot Instance Best Practices

Your instance type requirements, budget requirements, and application design will determine how to apply the following best practices for your application:

- **Be flexible about instance types.** A Spot Instance pool is a set of unused EC2 instances with the same instance type (for example, m5.large) and Availability Zone (for example, us-east-1a). You should be flexible about which instance types you request and in which Availability Zones you can deploy your workload. This gives Spot a better chance to find and allocate your required amount of compute capacity. For example, don't just ask for c5.large if you'd be willing to use larges from the c4, m5, and m4 families.
- **Use the price and capacity optimized allocation strategy.** Allocation strategies in EC2 Auto Scaling groups help you to provision your target capacity without the need to manually look for the Spot Instance pools with spare capacity. We recommend using the `price-capacity-optimized` strategy because this strategy automatically provisions instances from the most-available Spot Instance pools that also have the lowest possible price. Because your Spot Instance capacity is sourced from pools with optimal capacity, this decreases the possibility that your Spot Instances are interrupted. For more information about allocation strategies, see Spot Instances in the *Amazon EC2 Auto Scaling User Guide*.
- **Use proactive capacity rebalancing.** Capacity Rebalancing helps you maintain workload availability by proactively augmenting your Auto Scaling group with a new Spot Instance before a running Spot Instance receives the two-minute interruption notice. When Capacity Rebalancing is enabled, Auto Scaling attempts to proactively replace Spot Instances that have received a rebalance recommendation, providing the opportunity to rebalance your workload to new Spot Instances that are not at elevated risk of interruption.
- **Use integrated AWS services to manage your Spot Instances.** Other AWS services integrate with Spot to reduce overall compute costs without the need to manage the individual instances or fleets. We recommend that you consider the following solutions for your applicable workloads: Amazon EMR, Amazon ECS, AWS Batch, Amazon EKS, SageMaker, AWS Elastic Beanstalk, and Amazon GameLift. To learn more about Spot best practices with these services, see the Amazon EC2 Spot Instances Workshops Website.
- **Choose the modern and correct launch tool for Spot Instances.** If one of the AWS integrated services is not a good fit for your workload, and you still need to build your application with control over the launch of Spot Instances, use the right tool. For most workloads, you should use EC2 Auto Scaling because it supplies a more comprehensive feature set for a wide variety of

workloads, like ELB-backed applications, containerized workloads, and queue processing jobs. If you need more control over individual requests and are looking for a “launch only” tool, use EC2 Fleet in `instant` mode as a drop-in replacement to RunInstances, but with a wider set of capabilities, such as instance type diversification and allocation strategies. For more information, see Which is the best Spot request method to use? in the *Amazon EC2 User Guide*.

Spot Integration with Other AWS Services

Amazon EC2 Spot Instances integrate with several AWS services.

Amazon EMR Integration

You can run Amazon EMR clusters on Spot Instances and significantly reduce the cost of processing vast amounts of data for your analytics workloads. You can run your EMR clusters by easily mixing Spot Instances with On-Demand and Reserved Instances using the EMR Instance Fleets feature. You can use EMR allocation strategies to launch Spot Instances from the most-available capacity pools.

EC2 Auto Scaling Integration

You can use Amazon EC2 Auto Scaling groups to launch and manage Spot Instances, maintain application availability, diversify instance type and purchase option (On-Demand/Spot) selection, and scale your Amazon EC2 capacity using dynamic, scheduled, and predictive scaling policies. For more information, see Requesting Spot Instances for fault-tolerant and flexible applications in the *Amazon EC2 Auto Scaling User Guide*.

Amazon EKS Integration

You can cost-optimize your Kubernetes-based workloads using Amazon EKS, by launching Spot Instances in EKS managed node groups. EKS managed node groups manage the entire Spot Instance lifecycle, by replacing soon-to-be-interrupted Spot Instances with newly launched instances, to reduce the chances of impact on your application performance or availability when Spot Instances are interrupted (when EC2 needs the capacity back). To learn more, see Managed node groups in the *Amazon EKS User Guide*.

Amazon ECS Integration

You can run Amazon ECS clusters on Spot Instances to reduce the operational cost of running containerized applications. Amazon ECS supports automatic draining of Spot Instances that are soon-to-be interrupted. For more information, see Using Spot Instances in the *Amazon Elastic Container Service Developer Guide*.

Amazon ECS with AWS Fargate Spot Integration

If your containerized tasks are interruptible and flexible, you can choose to run your ECS tasks with the AWS Fargate Spot capacity provider, meaning that your tasks will run on AWS Fargate, a serverless containers platform, and you will benefit from cost savings driven by Fargate Spot. For more information, see AWS Fargate capacity providers in the *Amazon Elastic Container Service Developer Guide*.

Amazon Batch Integration

AWS Batch plans, schedules, and executes your batch computing workloads on AWS. AWS Batch dynamically requests Spot Instances on your behalf, reducing the cost of running your batch jobs.

Amazon SageMaker Integration

Amazon SageMaker makes it easy to train machine learning models using managed Spot Instances. Managed Spot training can optimize the cost of training models by up to 90% over On-Demand Instances. SageMaker manages the Spot interruptions on your behalf. For more information, see Managed Spot Training in Amazon SageMaker in the *Amazon SageMaker Developer Guide*.

Amazon Gamelift Integration

Amazon GameLift is a game server hosting solution that deploys, operates, and scales cloud servers for multiplayer games. Support for Spot Instances in Amazon Gamelift gives you the opportunity to significantly lower your hosting costs. When creating fleets of hosting resources, you can choose between On-Demand Instances or Spot Instances. While Spot Instances might be interrupted with two minutes of notification, Amazon GameLift's FleetIQ minimizes the chance of interruptions. For more information, see Using Spot Instances with GameLift in the *Amazon GameLift Developer Guide*.

AWS Elastic Beanstalk Integration

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS. You can simply upload your code, and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and auto scaling, to application health monitoring. You can use Spot Instances in your Elastic Beanstalk environments for cost optimizing the underlying infrastructure of your web applications. For information about using Spot Instances with Elastic Beanstalk, see Spot Instance support in the *AWS Elastic Beanstalk Developer Guide*.

Conclusion

Whether you have flexible compute needs or want to augment capacity without growing your budget, Spot Instances can be a great way to optimize your AWS costs and/or build with scale in mind. By properly architecting your workloads, you can take advantage of Spot Instances for a wide range of needs. For more information, see Amazon EC2 Spot Instances.

Resources

- AWS Architecture Center
- AWS Whitepapers
- AWS Architecture Monthly
- AWS Architecture Blog
- This Is My Architecture videos
- AWS Documentation

Document History and Contributors

Document History

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Minor update	Adjusted page layout.	April 30, 2021
Minor update	Content updated to reflect current best practices. Name of whitepaper changed from "Leveraging Amazon EC2 Spot Instances at Scale" to "Overview of Amazon EC2 Spot Instances" to better reflect content.	March 5, 2021
Minor update	Spot Instance Limits was updated.	February 3, 2021
Initial publication	Leveraging Amazon EC2 Spot Instances at Scale published.	March 1, 2018

Note

To subscribe to RSS updates, you must have an RSS plug-in enabled for the browser you are using.

Contributors

The following individuals and organizations contributed to this document:

- Amilcar Alfaro, Sr. Product Marketing Manager, AWS

- Erin Carlson, Marketing Manager, AWS
- Keith Jarrett, WW BD Lead - Cost Optimization, AWS Business Development
- Ran Sheinberg, Principal Solutions Architect, AWS