



Amazon Elastic Compute Cloud

An Introduction to Spot Instances

API version 2011-05-01

May 26, 2011



Table of Contents

Overview	1
Tutorial #1: Choosing Your Maximum Price	2
Core Concepts	2
Step #1: Select the Region	3
Step #2: Browse the Spot Instance Pricing History	3
Step #3: Determine the Maximum Price Bid for Your Request	4
Tutorial #2: Launching a Spot Instance Request	6
Step #1: Choosing an AMI	6
Step #2: Configuring the Instance Details	7
Step #3: Configuring a Kernel ID and RAM Disk ID	8
Step #4: Setting up a Key Pair	9
Step #5: Setting up a Security Group	9
Step #6: Completing the Launch	10
Step #7: Viewing your Instance	11
Step #8: Cleaning up your Instance	12
Tutorial #3: How to View and Cancel Spot Instance Requests	13
Step #1: Viewing Your Spot Requests	13
Step #2: Canceling Your Spot Request	14
Spot Instances: Example Applications	15
Best Practices for Using Spot Instances	17

Overview

Spot Instances are a new way to purchase and consume Amazon EC2 Instances. They allow customers to bid on unused Amazon EC2 capacity and run those instances for as long as their bid exceeds the current Spot Price. Amazon EC2 changes the Spot Price periodically based on supply and demand, and customers whose bids meet or exceed it gain access to the available Spot Instances. Spot Instances are complementary to On-Demand Instances and Reserved Instances, providing another option for obtaining compute capacity.

For customers with flexibility in when their applications can run, Spot Instances can significantly lower their Amazon EC2 costs for use cases like batch processing, scientific research, image processing, video encoding, data and web crawling, financial analysis, and testing. Additionally, Spot Instances can provide access to large amounts of additional capacity with no commitment for urgent needs.

The tutorial that follows assumes you currently have already signed up for Amazon EC2 and are using the AWS Management Console for Amazon EC2. If you have not signed up yet, please follow the instructions in our [Getting Started Guide for Amazon EC2](#).

Tutorial #1: Choosing Your Maximum Price

Core Concepts

To use Spot Instances, you place a Spot Instance request specifying the maximum price you are willing to pay per instance hour. If your maximum price bid exceeds the current Spot Price, your request is fulfilled and your instances will run until either you choose to terminate them or the Spot Price increases above your maximum price (whichever is sooner).

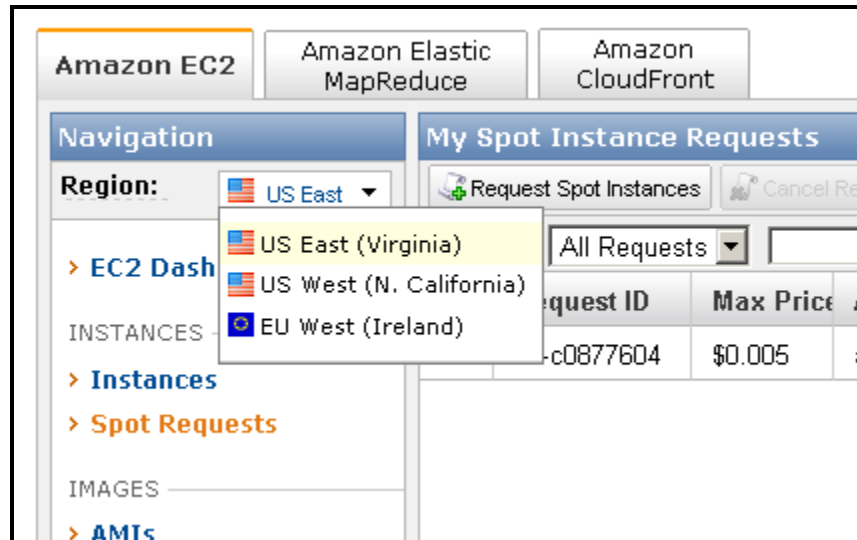
It's important to note three points:

- (1) You will often pay less per hour than your maximum bid price. Amazon EC2 adjusts the Spot Price periodically as requests come in and available supply changes. Everyone pays that same Spot Price for that period regardless of whether their maximum bid price was higher. You will never pay more than your maximum bid price per hour.
- (2) If you're running Spot Instances and your maximum price no longer meets or exceeds the current Spot Price, your instances will be terminated. This means that you will want to make sure that your workloads and applications are flexible enough to take advantage of this opportunistic capacity.
- (3) Spot Instance Requests can be one-time or persistent. A one-time request will only be satisfied once; a persistent request is eligible for consideration again after the associated Spot Instance is terminated.

Spot Instances perform exactly like other Amazon EC2 instances while running, and like other Amazon EC2 instances, Spot Instances can be terminated when you no longer need them. If you terminate your instance, you pay for any partial hour used (as you do for On-Demand or Reserved Instances). However, if the Spot Price goes above your maximum price and your instance is terminated by Amazon EC2, you will **not** be charged for any partial hour of usage.

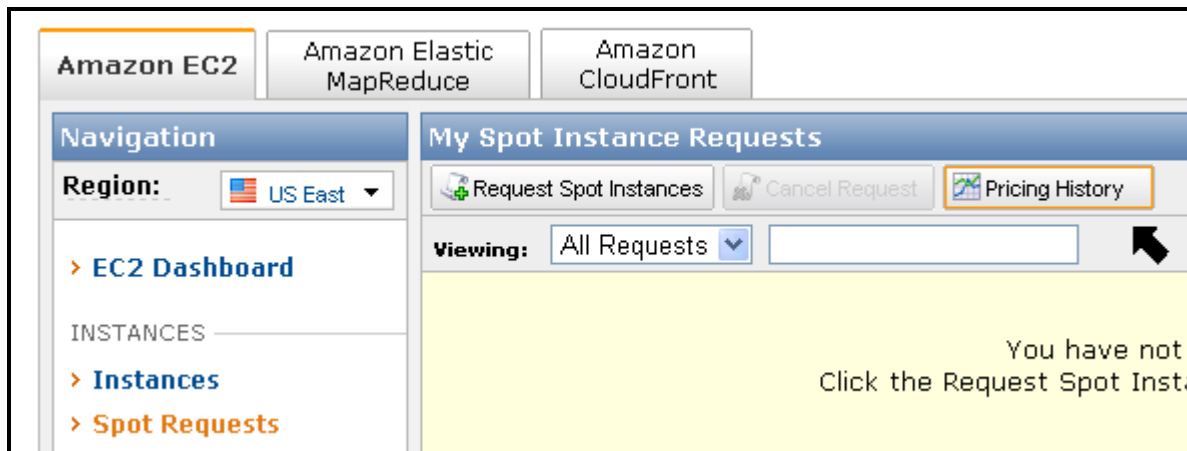
Step #1: Select the Region

From the [AWS Management Console for EC2](#), select the Region in which you want to request Spot Instances from the drop-down list as shown in the following example.

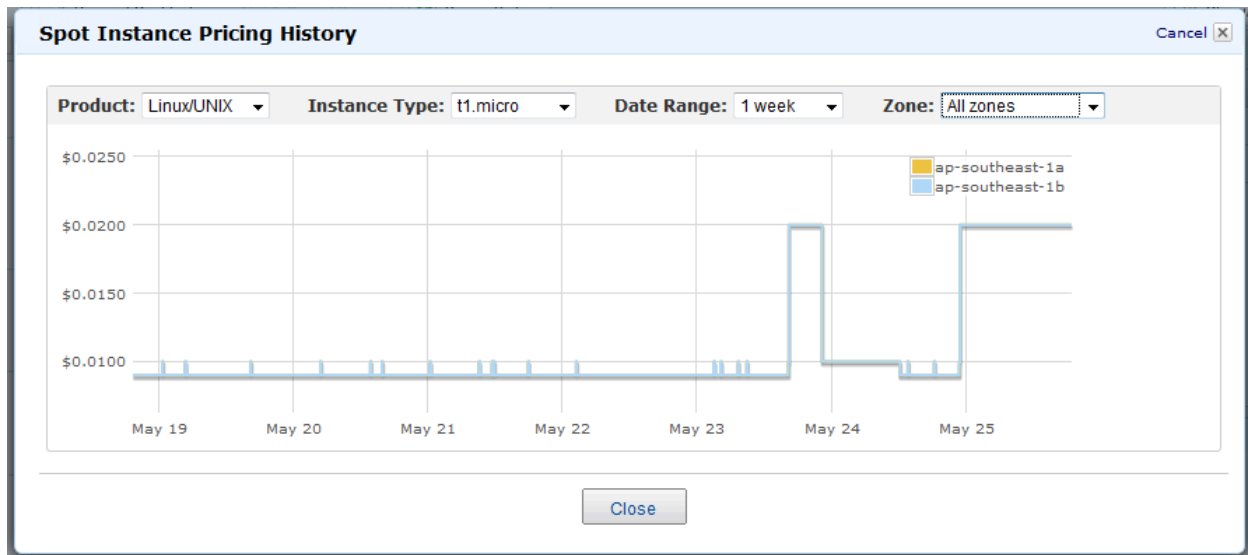


Step #2: Browse the Spot Instance Pricing History

Click the Spot Requests tab in the navigation panel, and then click the Pricing History button to bring up the Spot Instance pricing history.



Once the Spot Pricing History dialog is available, you can filter the pricing history that you are examining.



Step #3: Determine the Maximum Price Bid for Your Request

There are many ways to approach determining the maximum price for your request. We suggest you consider the following three examples of common approaches as a starting point:

Reducing Cost: You have a batch processing job to run. It will take a number of hours or days to run but is flexible in when it can be started and completed, so you would like to see if you can complete it for less cost than with On-Demand Instances. Observing the price history for your desired instance type in a Region, you see two options:

First, you could bid at the upper end of the range of Spot Prices (which still below the On-Demand price) expecting your one-time Spot request would be fulfilled and it would likely run for enough consecutive compute time to complete the job.

Second, you could bid at the lower end of the price range, planning on combining many instances launched over time through a persistent request that would run in aggregate for enough time to complete the job at even lower total cost. (We will explain how to automate this task later in this tutorial.)

Value-based Computing: You have a data processing job to run. You understand the value of the job's results well enough to know how much it is worth spending on compute resources to run it. Observing the Spot Price history, you choose a bid price at which the cost of the computing time is worth the return in value from the job's results. You create a persistent bid and allow it to run intermittently as the Spot Price fluctuates at or below your bid.

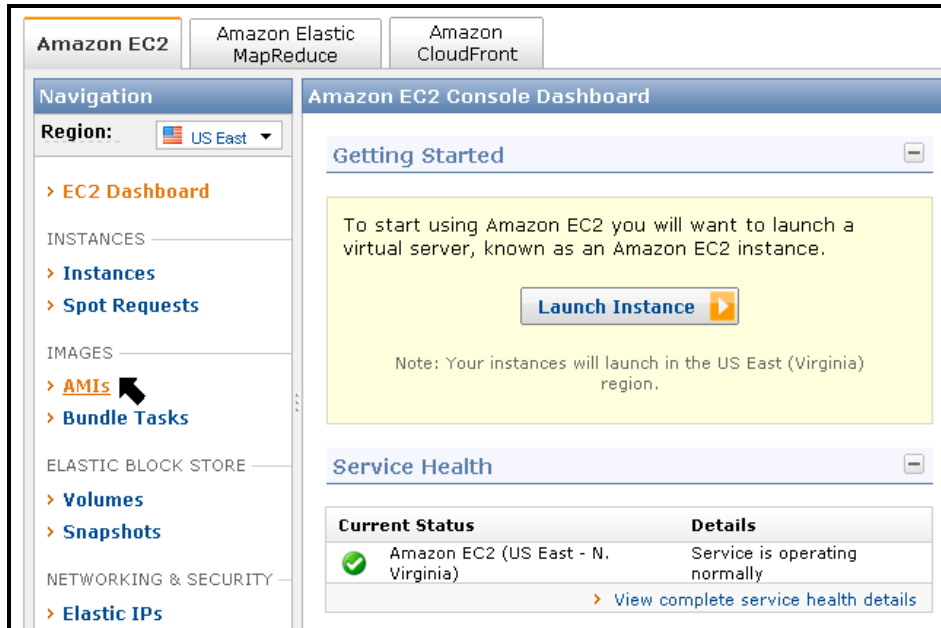
Additional Capacity Urgently Needed: You have an unanticipated, short-term need for additional capacity that is not available as On-Demand Instances. Observing the Spot Price history, you bid above the highest historical price to provide a high likelihood that your request will be fulfilled and will continue computing until it completes.

Once you have chosen your bid price, you are ready to request a Spot Instance.

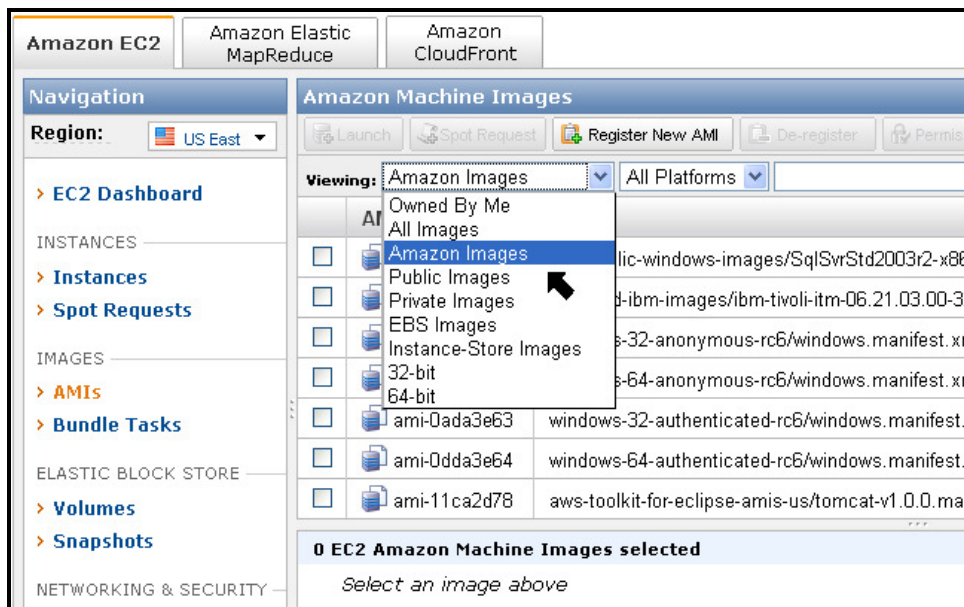
Tutorial #2: Launching a Spot Instance Request

Step #1: Choosing an AMI

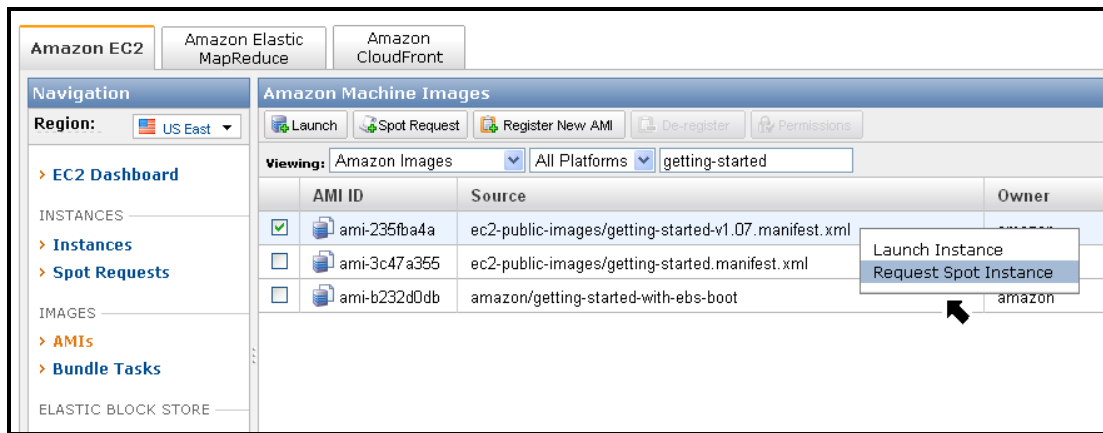
From the AWS Management Console for EC2 (<https://console.aws.amazon.com/ec2/home>), select the Amazon Machine Image (AMI) that you would like to run. To get a list of the current AMIs, navigate to the **AMIs** view in the AWS Management Console as shown below.



Once you have opened the AMIs view, filter the AMIs list so that you only see those AMIs created by Amazon. In the **Viewing** list, click **Amazon Images** as shown below.



Scroll down the list and right click the image “ec2-public-images/getting-started-v1.07.manifest.xml” and select **Request Spot Instance**. This will open the launch wizard for that AMI.



Step #2: Configuring the Instance Details

The current Spot Price is shown on the first screen for reference. You will need to specify a number of parameters for your request:

- Max Price: The maximum bid price you are willing to pay per instance-hour.
- Persistent: Whether your request is one-time or persistent. By default, it is one-time.
 - A one-time request can only be fulfilled once.
 - A persistent request is considered for fulfillment whenever there is no Spot Instance running for the same request.
- Request Validity Period: The length of time that your request will remain valid. You can specify both a starting and ending time for this period. By default, a Spot Request will be considered for fulfillment from the moment it is created until it is either fulfilled or canceled by you. However you can constrain the validity period if you need to.
- Launch Group: A Launch Group is a label that groups a set of requests together. All requests in a launch group have their instances started **and terminated** together.
- Availability Zone Group: An Availability Zone Group is a label that groups a set of requests together in the same Availability Zone. All requests that share an Availability Zone group and that are fulfilled at the same time will start Spot Instances in the same Availability Zone.
- An Availability Zone. You may also specify an explicit Availability Zone that you want for your Spot Instances.

Request Instances Wizard Cancel X

CHOOSE AN AMI | **INSTANCE DETAILS** | CREATE KEY PAIR | CONFIGURE FIREWALL | REVIEW

Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: 1 **Availability Zone:** No Preference ▾

Instance Type: Small (m1.small, 1.7 GB) ▾

Launch Instances

Request Spot Instances

Spot Instances let you pay for compute capacity by the hour at a Spot Price that fluctuates based on supply and demand. You specify a maximum price you are willing to pay per hour, and your instance only runs when the Spot Price is at or below that price. This allows for cost reduction on compute tasks with flexible start and end times.

Current Price: \$0.037 **Persistent Request?**

Max Price: \$ (Ex: 0.045 = 4.5 cents/hour) **Launch Group:**

Request Valid From: any time edit **Availability Zone Group:**

Request Valid Until: any time edit

< Back **Continue** ▶

Click **Continue**.

Step #3: Configuring a Kernel ID and RAM Disk ID

Although the next screen allows you to configure additional details like the Kernel ID and RAM Disk ID, we will not discuss these options in this tutorial. Click **Continue** to proceed.

Request Instances Wizard

CHOOSE AN AMI | **INSTANCE DETAILS** | CREATE KEY PAIR | CONFIGURE FIREWALL | REVIEW

Number of Instances: 1

Availability Zone: us-east-1a

Advanced Instance Options

Here you can choose a specific **kernel** or **RAM disk** to use with your instances. You can also choose to enable CloudWatch Monitoring or enter data that will be available from your instances once they launch.

Kernel ID: Use Default ▾

RAM Disk ID: Use Default ▾

Monitoring: Enable CloudWatch Monitoring for this instance
(additional charges will apply)

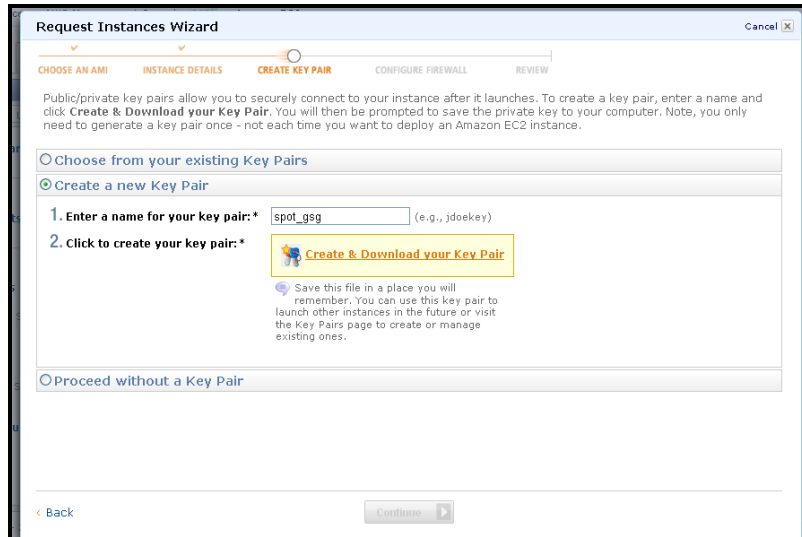
User Data:

base64 encoded

< Back **Continue** ▶

Step #4: Setting up a Key Pair

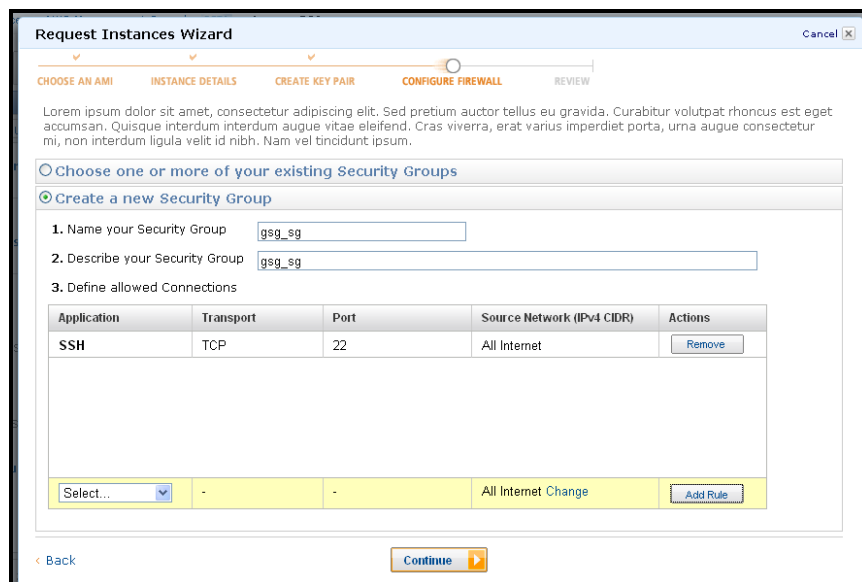
If you have already used the AWS Management Console (and your cookies are still configured), you will be prompted to select your existing key pair. If you haven't used the console before, please follow the instructions in our Getting Started Guide for Amazon EC2 at <http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/>.



Step #5: Setting up a Security Group

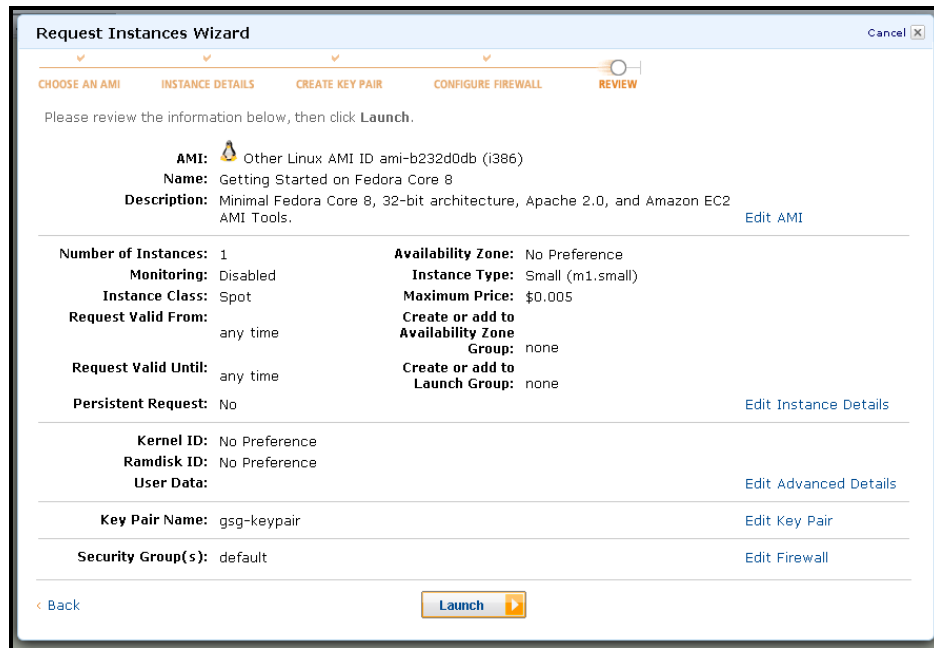
Click **Choose one or more of your existing Security Groups** and then select the security group you would like to use from the list. You can also set up a new group by filling in your security group name and description, selecting the application type, and adding the rule.

Click **Continue**.

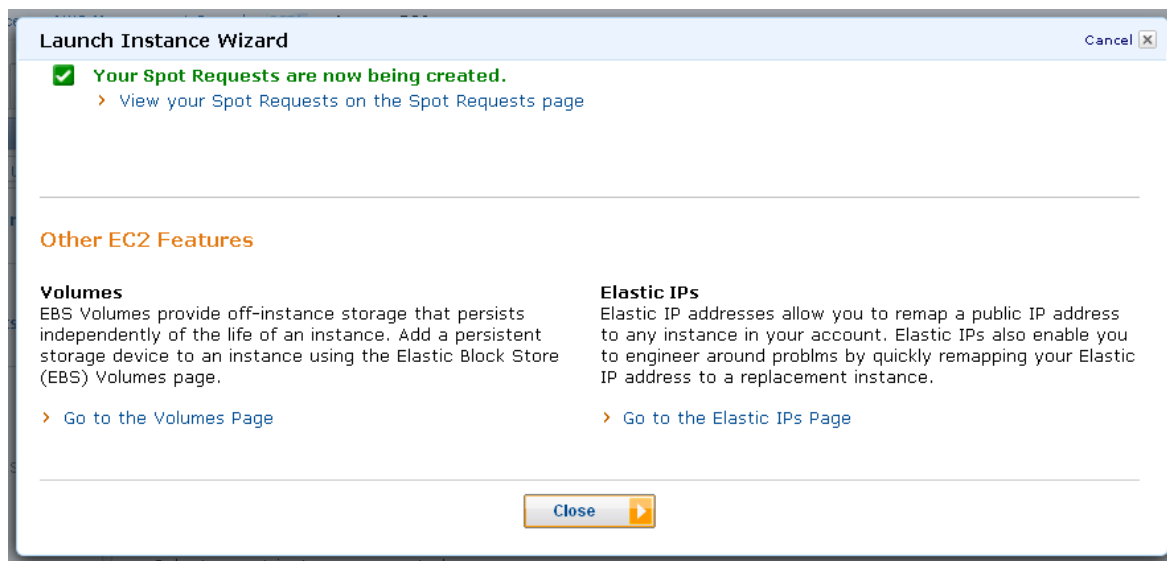


Step #6: Completing the Launch

After you have confirmed the details of your launch, click the **Launch** button to place your request for a Spot Instance. You'll see your new Spot Instance Request displayed in the right pane in the Console.

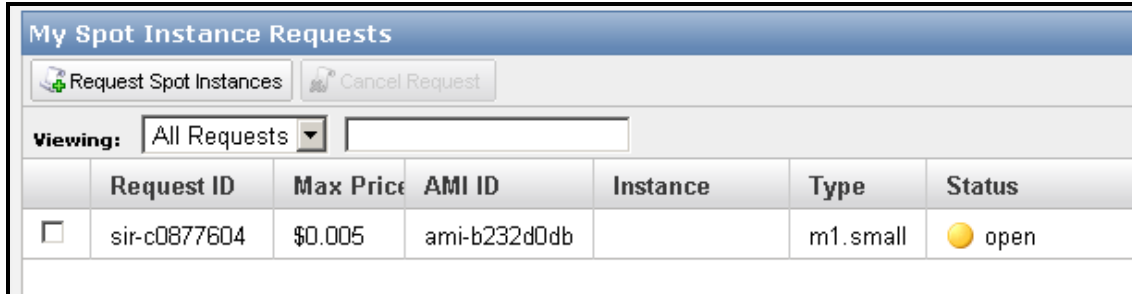


Once you press the **Launch** button, you will be presented with a final confirmation window similar to the one shown below.



Step #7: Viewing your Instance

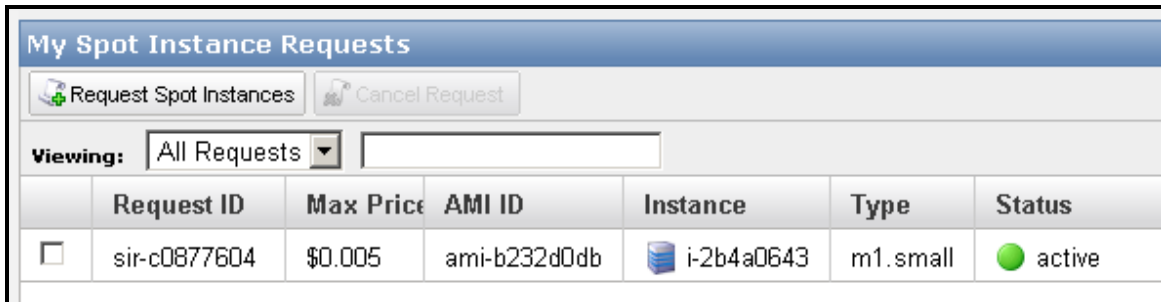
To view your Spot Instance Request, click the Spot Requests item in the navigation panel. All of your Requests should appear. The current request appears in the “open” status until it is fulfilled.




The screenshot shows the 'My Spot Instance Requests' interface. At the top, there are buttons for 'Request Spot Instances' and 'Cancel Request'. Below that is a 'Viewing:' section with a dropdown menu set to 'All Requests' and an empty search box. The main content is a table with the following data:

	Request ID	Max Price	AMI ID	Instance	Type	Status
<input type="checkbox"/>	sir-c0877604	\$0.005	ami-b232d0db		m1.small	● open

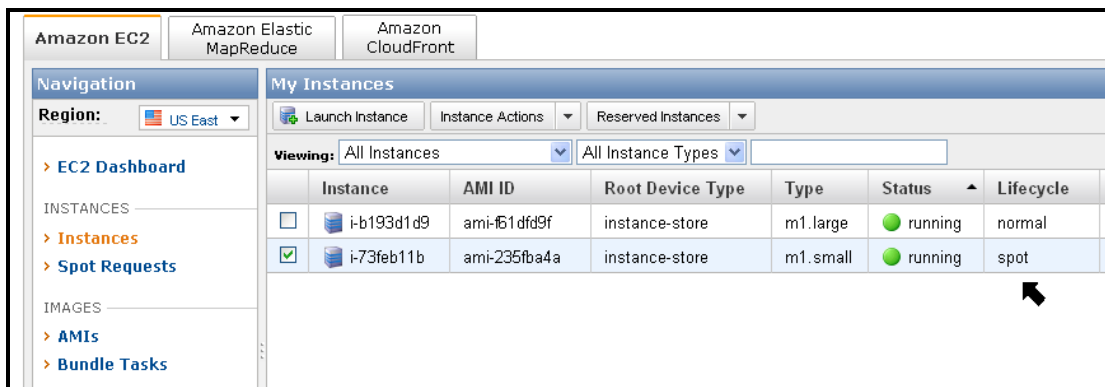
Your Spot Instance Request will be fulfilled based on the maximum price you specified, available Spot Instance capacity, requests submitted by other Amazon EC2 users, and the additional constraints you attached to the request (e.g. launch groups, or Availability Zones). When it is finally fulfilled, the Status column changes from “open” to “active”, and the instance ID is reflected as shown below.





The screenshot shows the 'My Spot Instance Requests' interface. The 'Viewing:' section remains the same. The table now shows the request as 'active' with an instance ID:

	Request ID	Max Price	AMI ID	Instance	Type	Status
<input type="checkbox"/>	sir-c0877604	\$0.005	ami-b232d0db	 i-2b4a0643	m1.small	● active

To view your actual running instance, simply click the **Instances** item on the navigation panel. The **Lifecycle** column shows “spot” to denote that your new instance is a Spot instance.



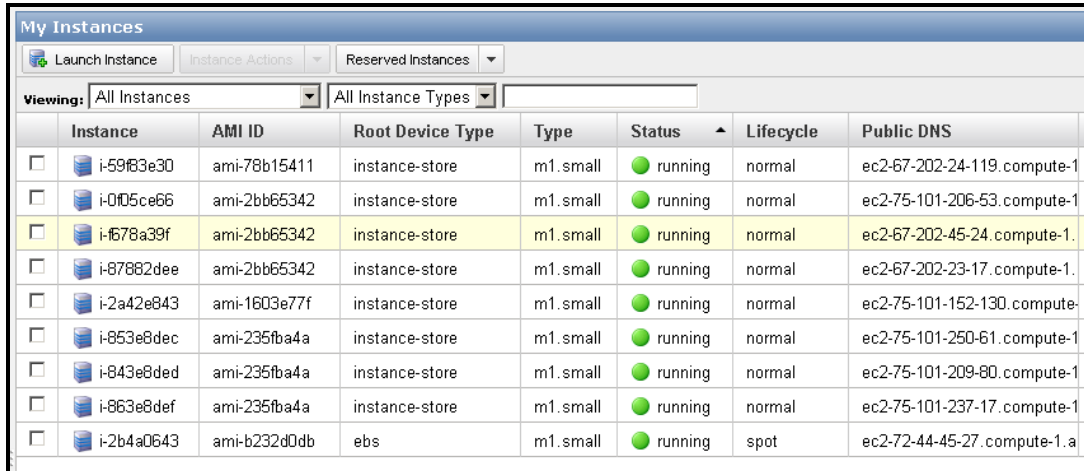
The screenshot shows the 'My Instances' console. The navigation panel on the left has 'Instances' selected. The main content area shows a table of instances:

	Instance	AMI ID	Root Device Type	Type	Status	Lifecycle
<input type="checkbox"/>	 i-b193d1d9	ami-b61dfd9f	instance-store	m1.large	● running	normal
<input checked="" type="checkbox"/>	 i-73feb11b	ami-235fba4a	instance-store	m1.small	● running	spot

An arrow points to the 'spot' value in the Lifecycle column of the second instance row.

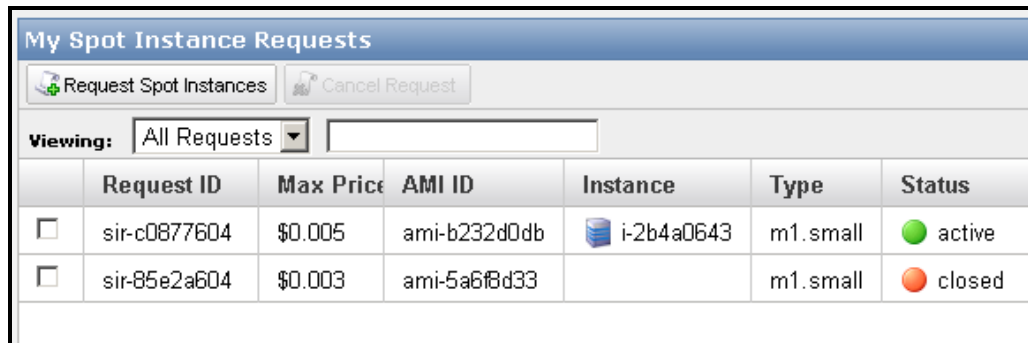
Step #8: Cleaning up your Instance

To clean up your Spot Instance, simply click the **Instances** item in the navigation panel. Then right click the instance you created and click **Terminate**.



Instance	AMI ID	Root Device Type	Type	Status	Lifecycle	Public DNS
<input type="checkbox"/> i-59f83e30	ami-78b15411	instance-store	m1.small	running	normal	ec2-67-202-24-119.compute-1
<input type="checkbox"/> i-0f05ce66	ami-2bb65342	instance-store	m1.small	running	normal	ec2-75-101-206-53.compute-1
<input type="checkbox"/> i-1678a39f	ami-2bb65342	instance-store	m1.small	running	normal	ec2-67-202-45-24.compute-1
<input type="checkbox"/> i-87882dee	ami-2bb65342	instance-store	m1.small	running	normal	ec2-67-202-23-17.compute-1
<input type="checkbox"/> i-2a42e843	ami-1603e77f	instance-store	m1.small	running	normal	ec2-75-101-152-130.compute-1
<input type="checkbox"/> i-853e8dec	ami-235fba4a	instance-store	m1.small	running	normal	ec2-75-101-250-61.compute-1
<input type="checkbox"/> i-843e8ded	ami-235fba4a	instance-store	m1.small	running	normal	ec2-75-101-209-80.compute-1
<input type="checkbox"/> i-863e8def	ami-235fba4a	instance-store	m1.small	running	normal	ec2-75-101-237-17.compute-1
<input type="checkbox"/> i-2b4a0643	ami-b232d0db	ebs	m1.small	running	spot	ec2-72-44-45-27.compute-1.a

When your instance is terminated (either by you or because the Spot Price moves above your maximum price) the status of your Spot Instance Request will change to “closed.”



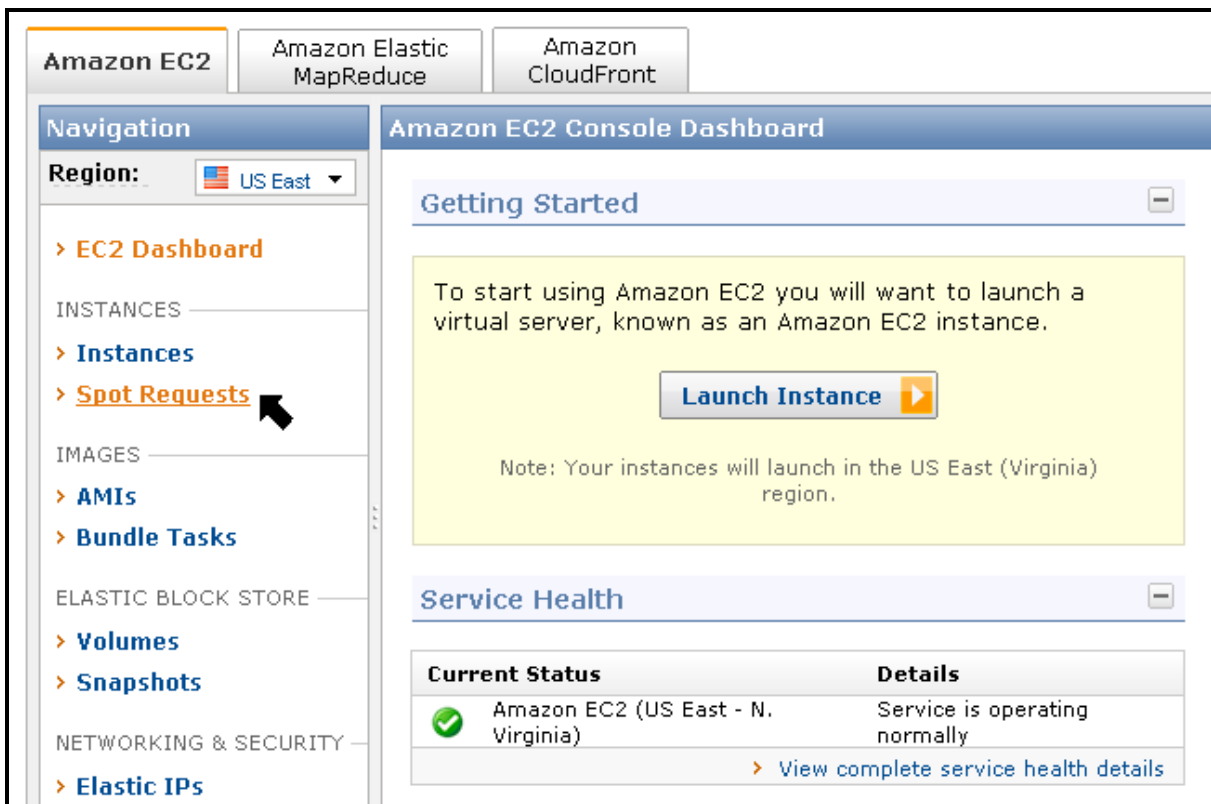
Request ID	Max Price	AMI ID	Instance	Type	Status
<input type="checkbox"/> sir-c0877604	\$0.005	ami-b232d0db	i-2b4a0643	m1.small	active
<input type="checkbox"/> sir-85e2a604	\$0.003	ami-5a6f8d33		m1.small	closed

Tutorial #3: How to View and Cancel Spot Instance Requests

This tutorial assumes that you have an existing Spot Instance request that you can practice with and that you are already logged into the AWS Management Console. If you need to create a new request, please follow the steps in Tutorial #2.

Step #1: Viewing Your Spot Requests

Once you have your request set up, you can view that request in the AWS Management Console by clicking on the **Spot Requests** item in the navigation panel.

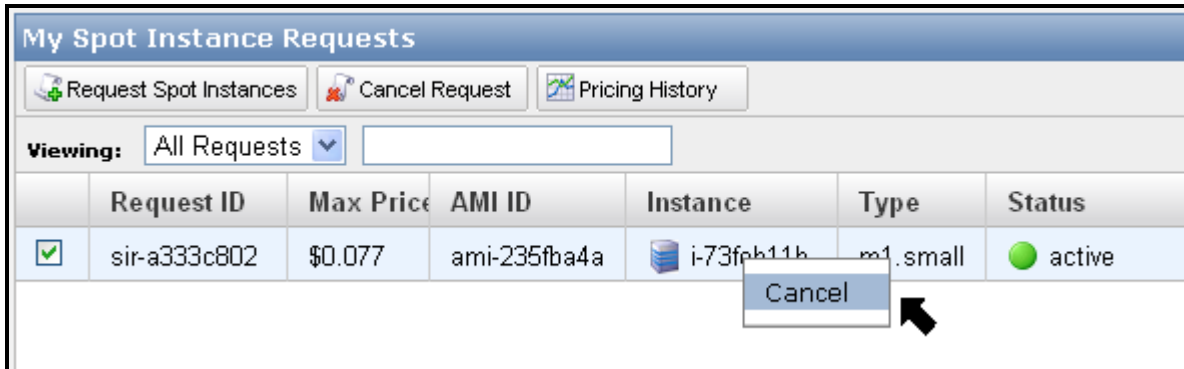


Once the view opens, you can see all of the Spot requests as shown below.

My Spot Instance Requests						
Request Spot Instances						
Cancel Request						
Pricing History						
Viewing: All Requests						
	Request ID	Max Price	AMI ID	Instance	Type	Status
<input checked="" type="checkbox"/>	sir-a333c802	\$0.077	ami-235fba4a	i-73feb11b	m1.small	● active

Step #2: Canceling Your Spot Request

Once you are able to view the Spot Instance requests, you can cancel a request by right clicking, and selecting **Cancel**.



My Spot Instance Requests

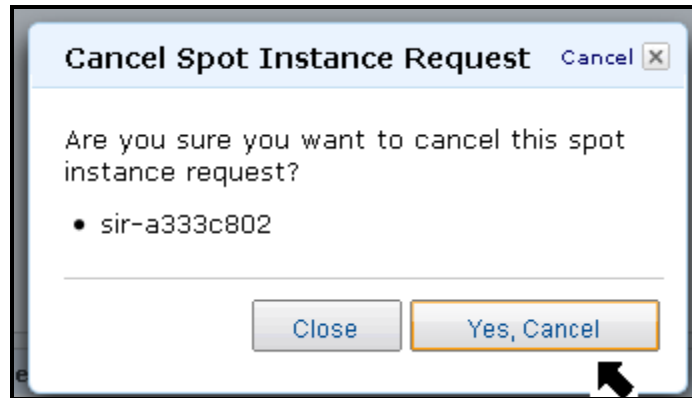
Request Spot Instances Cancel Request Pricing History

Viewing: All Requests

	Request ID	Max Price	AMI ID	Instance	Type	Status
<input checked="" type="checkbox"/>	sir-a333c802	\$0.077	ami-235fba4a	i-73feb11b	m1.small	active

Cancel

When the pop-up box opens, simply click **Yes, Cancel** to completely cancel the request.



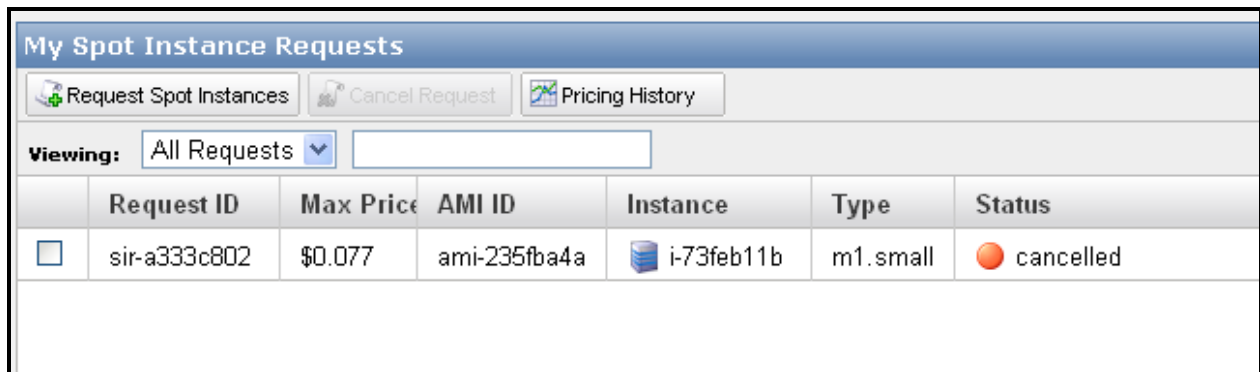
Cancel Spot Instance Request

Are you sure you want to cancel this spot instance request?

- sir-a333c802

Close Yes, Cancel

Once a request is canceled, it will show up in the “canceled” state through the console as shown below.



My Spot Instance Requests

Request Spot Instances Cancel Request Pricing History

Viewing: All Requests

	Request ID	Max Price	AMI ID	Instance	Type	Status
<input type="checkbox"/>	sir-a333c802	\$0.077	ami-235fba4a	i-73feb11b	m1.small	cancelled

Spot Instances: Example Applications

Now that you have walked through the mechanics of creating Spot Instance requests, we will walk through how a couple of sample applications can take advantage of Spot Instances.

Spot Instances are well suited to a number of different kinds of applications. In general, they can save you money if your application doesn't depend on instances being started immediately, can start and perform useful work without manual intervention, and can resume work after being interrupted. Example workloads include web and data crawling, financial analysis, grid computing, media transcoding, scientific research, and batch processing.

Consider an application that resizes a backlog of images stored in Amazon S3. If the backlog is represented as Amazon SQS messages, then at start-up each instance can simply read the next message off the queue, fetch the image, resize it, and store the resized image in Amazon S3 again in a new location.

Amazon SQS ensures that each message is seen by a single instance, ensuring that work is not duplicated. Because Amazon SQS hides the message for a period without deleting it, other instances will not see the message while it's being processed. When the instance processing has stored the new image in Amazon S3, the message can be deleted. If the instance is interrupted before it deletes the message, it will eventually become visible in the queue again and another instance can pick it up and do the work.

In this example we'll assume resizing the images may not be urgent and that keeping costs down is more important than a quick turnaround. One way to achieve this would be to create a number of persistent requests with a maximum price that's reasonably low. It could even be below the current Spot Price, but you should review the historical Spot Prices and probably start with a maximum price at or above the lowest historical price to try to ensure you get some compute time. Over time as the Spot Price changes, Spot Instances will be launched and terminated for you whenever the Spot Price drops below your maximum price. When this happens, your instances will resume working on the images in the SQS queue.

If your work is more urgent, then choosing a higher maximum price is an ideal way to get additional computing time quickly. In the previous example, if we assume a sudden need arises to get the remaining images resized quickly, you could create additional requests with a maximum price that is above the current Spot Price to increase the likelihood that your Spot Instances are started. Your maximum price should be set according to how urgent the work is (and therefore how important it is that your Spot Instances are interrupted less frequently). As a general rule, you should choose a price that reflects the maximum you're prepared to pay for the compute you need. You will never pay more than your maximum price, and often you may pay significantly less.

Knowing how much compute time you need is useful when selecting a maximum price. Assume in this case each image takes 1 minute to resize and you have 120 images remaining in the queue (requiring

two hours of compute time to complete.) To simplify the example, let's assume you only wanted to start one Spot Instance. You should review several hours of historical prices and set your maximum price at a level that will give you a high probability of running for a minimum of two hours given past prices.

Consider as a second example running a Hadoop cluster. In general, the master node in a Hadoop cluster is less tolerant to being terminated than a worker node. Mixing Spot Instances with On-Demand Instances helps to solve this problem. The master node can be run as an On-Demand Instance, thus ensuring that it won't be interrupted, and the worker nodes, which are much more tolerant to being interrupted, can be run as Spot Instances at a significantly reduced cost.

Alternatively, you could create a request with a high maximum price that starts your master node, and then place requests with a lower maximum price to start your worker nodes. In this case, you may choose to use a launch group to ensure the master and worker nodes are all started together, and an Availability Zone group to ensure they're all started in the same Availability Zone to minimize transfer costs and latency.

Best Practices for Using Spot Instances

Finally, there are a number of best practices that are good to keep in mind when running applications on Spot Instances in order to minimize the potential impact from your instance being interrupted.

Save Your Work Frequently: Because Spot Instances can be terminated with no warning, it is important to build your applications in a way that allows you to make progress even if your application is interrupted. There are many ways to accomplish this, two of which are adding checkpoints to your application or splitting your work into small increments.

Add Checkpoints: Depending on fluctuations in the Spot Price caused by changes in the supply or demand for Spot capacity, Spot Instance requests may not be fulfilled immediately and may be terminated without warning. In order to protect your work from potential interruptions, we recommend inserting regular checkpoints to save your work periodically. One way to do this is by saving all of your data to an Amazon EBS volume.

Another approach is to run your instances using Amazon EBS-backed AMIs. By setting the `DeleteOnTermination` flag to false as part of your launch request, the Amazon EBS volume used as the instance's root partition will persist after instance termination, and you can recover all of the data saved to that volume. You can read more details on the use of Amazon EBS-backed AMIs [here](#).

Note: When using this technique with a persistent request, bear in mind that a new EBS volume will be created for each new Spot Instance.

Split up Your Work: Another best practice is to split your workload into small increments if possible. Using Amazon SQS, you can queue up work increments and keep track of what work has already been done (as in the example from the previous section). When using this approach, ensure that processing a unit of work is idempotent (can be safely processed multiple times) to ensure that resuming an interrupted task doesn't cause problems.

You can do this by enqueueing a message to your Amazon SQS queue for each increment of work. You can then build an AMI that, when run, discovers the queue from which to pull its work. Discovery can be done by building it into the AMI, passing in user data or by storing the configuration remotely (for example in Amazon SimpleDB or Amazon S3), which will tell the AMI in which queue to look.

More details on using Amazon SQS with Amazon EC2 and a detailed walkthrough on how to set up this type of architecture can be found [here](#).

Test Your Application: When using Spot Instances, it is important to make sure that your application is fault tolerant and will correctly handle interruptions. While we attempt to cleanly terminate your instances, your application should be prepared to deal with sudden shutdowns. You can test your application by running an On-Demand Instance and then terminating it. This can help you to determine whether your application is sufficiently fault tolerant and is able to handle unexpected interruptions.

Minimize Group Instance Launches: There are two options for launching instances together in a cluster. The Launch Group is a request option that ensures your instances will be launched and terminated simultaneously. The Availability Zone Group is a second request option that ensures your instances will be launched together in one Availability Zone. Although they may be necessary for some applications, avoiding these restrictions whenever possible will increase the chances of your request being fulfilled. When Launch Groups are required, try to minimize the group size because larger groups have a lower chance of being fulfilled. Additionally whenever possible, try to avoid specifying a specific Availability Zone in order to increase your chances of successfully launching.

Use Persistent Requests for Continuous Tasks: Spot Instance Requests can be one-time or persistent. A one-time request will only be satisfied once; a persistent request will remain in consideration after each instance termination. This means that after your request has been satisfied and your instance has been terminated—by you or by Amazon EC2—your request will be submitted again automatically with the same parameters as your initial request. A persistent request will continue submitting the request until you cancel it. These requests can be helpful if you have continuous work that can be stopped and resumed, such as data processing or video rendering. We recommend that you revisit these requests from time to time to examine whether or not you want to change your maximum price or the AMI. Changing parameters will require that you cancel your existing request and resubmit a new request.

Note: Terminating your instance is not the same as cancelling a persistent request. If you terminate your instance without cancelling your persistent request, Amazon EC2 will automatically launch a replacement Spot Instance given that your maximum price is above the current Spot Price.

Track when Spot Instances Start and Stop: The simplest way to know the current status of your Spot Instances is to either poll the *DescribeSpotInstanceRequests* API or view the status of your instance using the AWS Management Console. By polling the *DescribeSpotInstanceRequests* at whatever frequency you desire (e.g. every ten minutes), you can look for state changes to your requests. This will tell you when a request is successful, because it will change from “open” to “active” and it will have an associated instance ID. You can use this same approach to detect terminations by checking to see if the “instance id” field disappears.

You can also use Amazon SQS to create your own notifications. One way of doing this is to create an AMI that has a start-up script that enqueues a message on an Amazon SQS queue. You can take the same approach to detect when a Spot Instance begins the process of shutting down.

For instructions on how to build your own AMI, please see the Amazon EC2 User Guide located [here](#).

Access Large Pools of Compute Capacity: Spot Instances can be used to help you meet occasional needs for large amounts of compute capacity (note that the default limit for Spot Instances is 100 versus the default limit of 20 for On-Demand Instances.) If your needs are urgent, you can specify a high maximum price (possibly even higher than the On-Demand price), which will raise your request’s relative priority and allow you to gain access to as much immediate capacity as possible given other requests and the

Spot Instance capacity available at the time. While Spot Instances are generally not suitable for steady-state tasks such as serving web content, they can be used as a valuable source of instance capacity even for steady state applications when applications have urgent computing needs due to unanticipated or short-term demand spikes.
