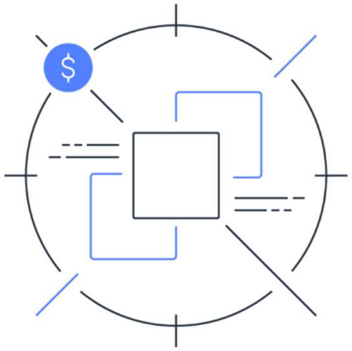
Amazon EC2 Spot Instances Workshop



Overview

Amazon EC2 Spot Instances offer spare compute capacity available in the AWS cloud at steep discounts compared to On-Demand instances. Spot Instances enable you to optimize your costs on the AWS cloud and scale your application's throughput up to 10X for the same budget.

Spot Instances can be interrupted by EC2 with two minutes of notification when EC2 needs the capacity back. You can use Spot Instances for various fault-tolerant and flexible applications, such as big data, containerized workloads, high-performance computing (HPC), stateless web servers, rendering, CI/CD and other test & development workloads.

Note to Bayer:

An earlier version of this content also available at https://ec2spotworkshops.com/launching_ec2_spot_instances.html

This version has been edited to make it more relevant to Bayer Crop Science.

Launching EC2 Spot Instances

Overview

Amazon EC2 Spot instances are spare compute capacity in the AWS Cloud available to you at steep discounts compared to On-Demand prices. EC2 Spot enables you to optimize your costs on the AWS cloud and scale your application's throughput up to 10X for the same budget.

This lab will walk you through creating an EC2 Launch Template, and then using this Launch Template to launch EC2 Spot Instances the following 4 ways: Amazon EC2 Auto Scaling, the EC2 RunInstances API, EC2 Spot Fleet, and EC2 Fleet.

Pre-Requisites

This lab requires:

- A laptop with Wi-Fi running Microsoft Windows, Mac OS X, or Linux.
- The AWS CLI installed and configured.
- An Internet browser such as Chrome, Firefox, Safari, or Edge.
- An AWS account.

If you need to install the AWS CLI, refer to: https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html

Assumptions

The lab assume you have an AWS account, and some familiarity with the CLI and console. If this doesn't describe you, please consider teaming up with a lab partner.

Short Cuts

To save some typing, certain commands from the lab are available on github. To retrieve these:

1. Install git, if needed

Linux:

Fedora/AWS Linux/…

```
sudo yum install git
```

Debian/Ubuntu

```
sudo apt-get install git
```

Mac:

```
brew install git
```

Windows:

```
Download and run installer from: http://msysgit.github.io
```

2. Retrieve the repo

```
git clone https://github.com/dotstar/spotlab.git
```

3. These scripts require python3 and jq. If you don't already have these installed, add them now:

Linux:

```
sudo yum install python3, jq -y    -or- apt-get install python3,jq
```

Mac:

```
brew install python3, jq
```

Windows:

https://www.python.org/ftp/python/3.6.4/python-3.6.4.exe

While you can install jq for Windows, the common installers require frameworks like: choco, Appveyor, or linuxbrew. If you are running Windows and don't already have jq installed, all but one of the CLI will function. Probably better to skip the CLI than go through the heavy lifting of installing jq on Windows.

Lab 1: Launching EC2 Spot instances via EC2 Auto Scaling Group

Creating a Launch Template

You can create a *launch template* that contains the configuration information to launch an instance. Launch templates enable you to store launch parameters so that you do not have to specify them every time you launch an instance. For example, a launch template can contain the AMI ID, instance type, and network settings that you typically use to launch instances. When you launch an instance using the Amazon EC2 console, an AWS SDK, or a command line tool, you can specify the launch template to use.

To create a new launch template using the command line

1. You'll need to gather the following data
   1. **AMI ID**: Specify an AMI ID from which to launch the instance. You can use an AMI that you own, or you can find a suitable AMI.
   2. **Instance type**: Choose the instance type. Ensure that the instance type is compatible with the AMI you've specified. For more information, see Instance Types.
   3. **Subnet**: Specify the subnet in which to create a new network interface. For the primary network interface (eth0), this is the subnet in which the instance is launched.
2. Once you've gathered the data, create the launch template from the command line as follows (be sure to change the following values: **SubnetId**, **ImageId**, **InstanceType**, **Tags - Value**):

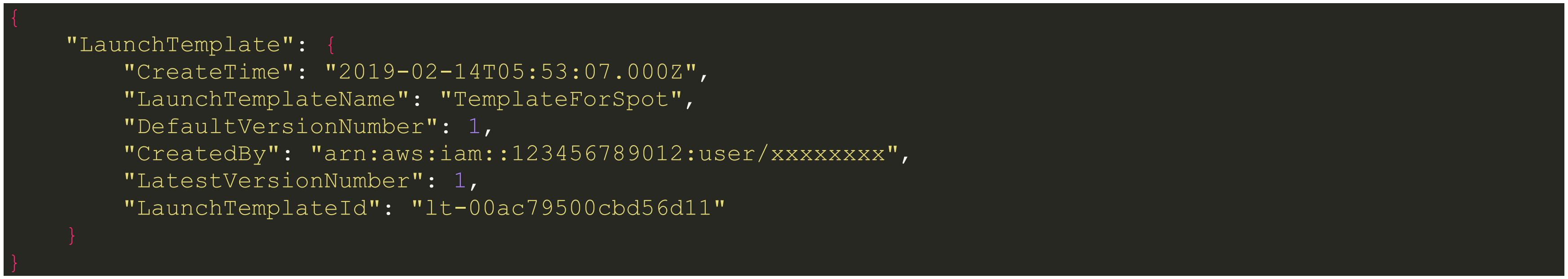
   Alternatively, go to the scripts you downloads from github (cd spotlab). Edit and run script **01-create-launch-template.**

   1. You will need to set the variable subnet to match your account.

2. This was written with an AMI which exists in us-east-2. Should you desire to run in another region, change the ami and region variables.

```
aws ec2 create-launch-template --region ap-northeast-1 --launch-template-name TemplateForSpot --version-description TemplateForSpotVersion1 --launch-template-data "{\"NetworkInterfaces\":[{\"DeviceIndex\":0,\"SubnetId\":\"subnet-93d49ac8\"}],\"ImageId\":\"ami-06cd52961ce9f0d85\",\"InstanceType\":\"m4.large\",\"TagSpecifications\":[{\"ResourceType\":\"instance\",\"Tags\":[{\"Key\":\"Name\",\"Value\":\"EC2SpotImmersionDay\"}]}]}"
```

Example return



```
{
    "LaunchTemplate": {
        "CreateTime": "2019-02-14T05:53:07.000Z",
        "LaunchTemplateName": "TemplateForSpot",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:user/xxxxxxxx",
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-00ac79500cbd56d11"
    }
}
```

Note the **LaunchTemplateId** (eg. “lt-00ac79500cbd56d11”) or **LaunchTemplateName** (eg. “TemplateForSpot”) of the newly created Launch Template for the next steps.

Launching EC2 Spot Instances via an EC2 Auto Scaling Group

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called Auto Scaling groups. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size.

With launch templates, you can also provision capacity across multiple instance types using both On-Demand Instances and Spot Instances to achieve the desired scale, performance, and cost.

To create an Auto Scaling group using a launch template

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation bar at the top right of the screen, select the same region that you used when you created the launch template.
3. In the navigation pane, choose **Launch Templates**.
4. Select your launch template.
5. Choose **Create Auto Scaling group** [Actions -> Create Auto Scaling group]
6. On the Configure Auto Scaling group details page, for Group name, type a name for your Auto Scaling group.
7. For Launch template version, select the **Default**.
8. For Fleet Composition, choose **Combine purchase options and instances** to launch instances across multiple instance types using both On-Demand and Spot purchase options.
9. While you chose to combine purchase options and instance types:
    1. For **Instance Types**, choose the optimal instance types that can be used for your application (such as m4.large and c4.large) that may be launched. Please be sure to select at **least two types**.
    2. For **Instance Distribution**, choose to replace the default instance settings [**uncheck** the box which says *Use the default settings to get started quickly*]
    3. Provide the following information.
        1. For **Maximum Spot Price**, choose Use default to cap your maximum Spot price at the On-Demand price.
        2. For **Optional On-Demand Base**, you can specify the minimum amount of the Auto Scaling group's initial capacity that must be fulfilled by On-Demand Instances. Leave this field blank to launch On-Demand Instances as a percentage of the group's desired capacity.
        3. For **On-Demand Percentage Above Base**, specify the percentages of On-Demand Instances and Spot Instances for your additional capacity beyond the optional On-Demand base amount. Specify 50 here.
10. For **Group size**, enter the initial number of instances for your Auto Scaling group. Specify 4 here.
11. For **Network**, choose a VPC for your Auto Scaling group.
12. For **Subnet**, choose one or more subnets in the specified VPC.
13. Choose **Next: Configure scaling policies**.
14. On the **Configure scaling policies** page, select **Keep this group at its initial size**, and then choose **Review**.
15. On the **Review** page, choose **Create Auto Scaling group**.
16. On the **Auto Scaling group creation status** page, choose **Close**.

You have now created an Auto Scaling group configured to launch not only EC2 Spot Instances but EC2 On-Demand Instances with multiple instance types.

Visit https://console.aws.amazon.com/ec2/ to validate that 4 instances are initiating with the name EC2SpotImmersionDay. This name came from the Launch Template which we created in step one.

The column "*Lifecyle*" shows that some of the instances are spot instances.

If you do not have a column named Lifecycle, adjust the display properties using the ⚙ icon.

When you have observed that the spot request created the instances, we can clean up this section of the lab:

Delete Your Auto Scaling Group

To delete your Auto Scaling group using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **Auto Scaling**, choose **Auto Scaling Groups**.
3. On the Auto Scaling groups page, select your Auto Scaling group, and choose **Actions**, **Delete**.
4. When prompted for confirmation, choose **Yes, Delete**.

You will see the instances shutting down. Several minutes later the spot request transitions from *active* to *closed*.

Lab2: Launching an EC2 Spot Instance via the RunInstances API

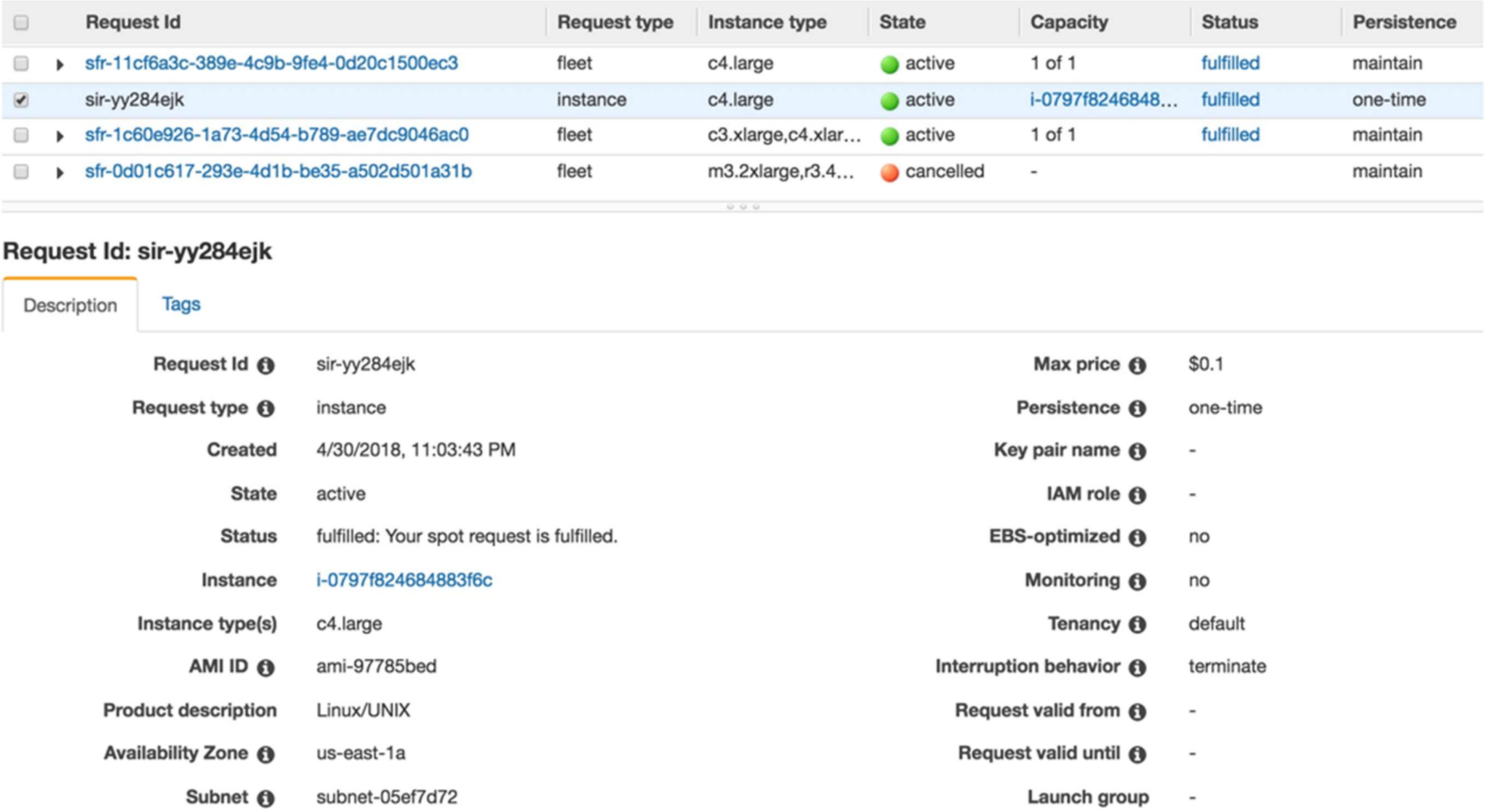
Launching an EC2 Spot Instance via the RunInstances API

To launch an EC2 Spot Instance from a launch template using the command line, use the run-instances AWS CLI command and specify the *–launch-template* parameter as well as the *–instance-market-options* parameter.

Alternatively, run the script **03-run-instances.**

```
$ aws ec2 run-instances --launch-template LaunchTemplateName=TemplateForSpot,Version=1 --instance-market-
options MarketType=spot
```

That is all there is to it! You can see your Spot Instance request in the Spot console at https://console.aws.amazon.com/ec2spot.



□	Request Id	Request type	Instance type	State	Capacity	Status	Persistence
□	▸ sfr-11cf6a3c-389e-4c9b-9fe4-0d20c1500ec3	fleet	c4.large	● active	1 of 1	fulfilled	maintain
☑	sir-yy284ejk	instance	c4.large	● active	i-0797f8246848...	fulfilled	one-time
□	▸ sfr-1c60e926-1a73-4d54-b789-ae7dc9046ac0	fleet	c3.xlarge,c4.xlar...	● active	1 of 1	fulfilled	maintain
□	▸ sfr-0d01c617-293e-4d1b-be35-a502d501a31b	fleet	m3.2xlarge,r3.4...	● cancelled	-		maintain

Request Id: sir-yy284ejk

Description   Tags

Request Id	sir-yy284ejk	Max price	$0.1
Request type	instance	Persistence	one-time
Created	4/30/2018, 11:03:43 PM	Key pair name	-
State	active	IAM role	-
Status	fulfilled: Your spot request is fulfilled.	EBS-optimized	no
Instance	i-0797f824684883f6c	Monitoring	no
Instance type(s)	c4.large	Tenancy	default
AMI ID	ami-97785bed	Interruption behavior	terminate
Product description	Linux/UNIX	Request valid from	-
Availability Zone	us-east-1a	Request valid until	-
Subnet	subnet-05ef7d72	Launch group	-

After you are satisfied that the instance launched as a spot instance, proceed to clean up.

Terminating a Spot Instance

To terminate a Spot Instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance, and choose **Actions**, **Instance State**, **Terminate**.
4. Choose **Yes, Terminate** when prompted for confirmation.

Lab3: Launching EC2 Spot Instances via Spot Fleet

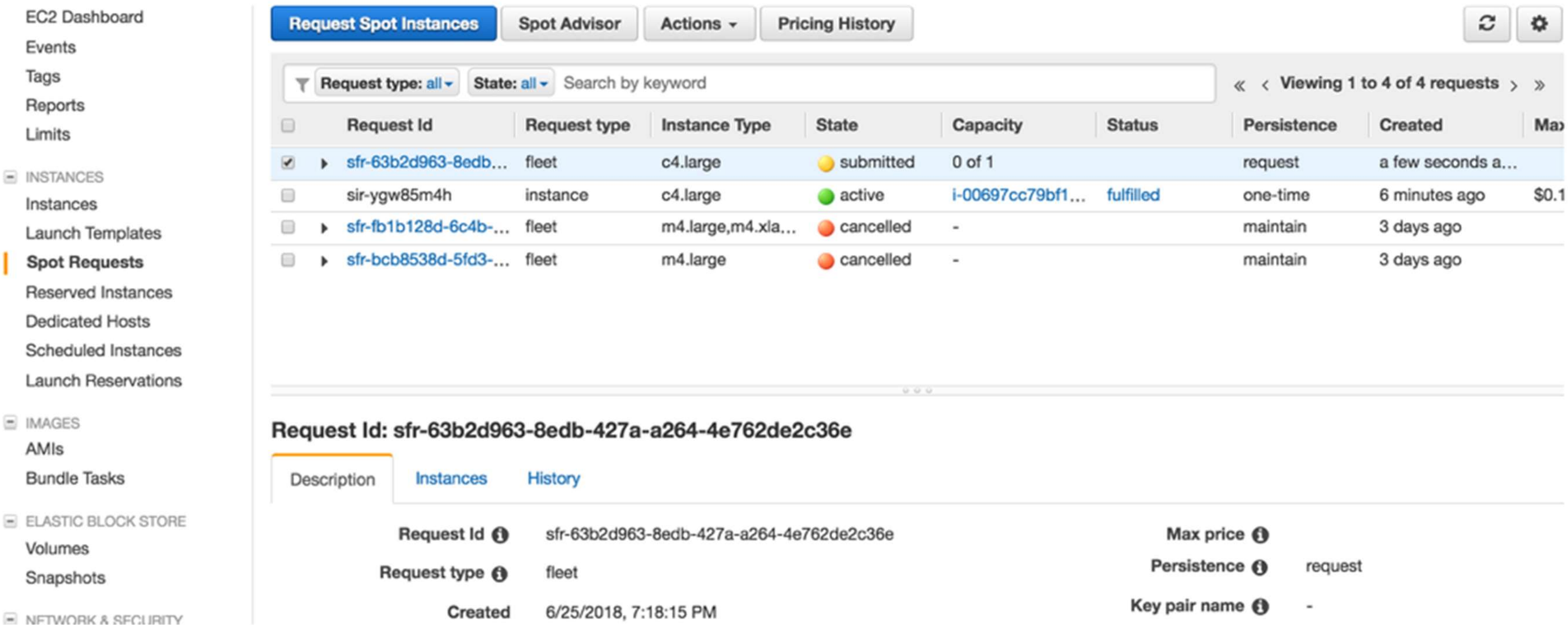
Launching EC2 Spot Instances via a Spot Fleet request

You can create a Spot Fleet request and specify a launch template in the instance configuration. When Amazon EC2 fulfills the Spot Fleet request, it uses the launch parameters defined in the associated launch template.

To create a Spot Fleet request using the recommended settings

1. Open the Spot console at https://console.aws.amazon.com/ec2spot.
2. If you are new to Spot, you see a welcome page; choose **Get started**. Otherwise, choose **Request Spot Instances**.
3. For **Tell us your application or task need**, choose **Flexible workloads**.
4. Under **Configure your instances:**
    - For **Launch template**, select the Launch template you created earlier.
    - Leave the **Minimum compute** unit values as default.
    - For **Network**, select the VPC in which the subnet used in the launch template belongs.
    - Under **Availability Zone**, check all of the availability zones that have an available subnet.
5. Under **Tell us how much capacity you need**, for **Total target capacity**, specify **6 vCPUs**, and for **Optional On-Demand portion**, specify **2 vCPUs**.
6. Check the box for **Maintain target capacity**. Leave the **Interruption behavior** as **Terminate**.
7. Review the recommended **Fleet request settings** based on your application or task selection, and choose **Launch**.

The request type is fleet. When the request is fulfilled, requests of type instance are added, where the state is active and the status is fulfilled.



☐		Request Id	Request type	Instance Type	State	Capacity	Status	Persistence	Created	Ma
☑	▸	sfr-63b2d963-8edb...	fleet	c4.large	● submitted	0 of 1		request	a few seconds a...	
☐		sir-ygw85m4h	instance	c4.large	● active	i-00697cc79bf1...	fulfilled	one-time	6 minutes ago	$0.1
☐	▸	sfr-fb1b128d-6c4b-...	fleet	m4.large,m4.xla...	● cancelled	-		maintain	3 days ago	
☐	▸	sfr-bcb8538d-5fd3-...	fleet	m4.large	● cancelled	-		maintain	3 days ago	

Request Id: sfr-63b2d963-8edb-427a-a264-4e762de2c36e

Description | Instances | History

Request Id: sfr-63b2d963-8edb-427a-a264-4e762de2c36e

Request type: fleet

Created: 6/25/2018, 7:18:15 PM

Max price:

Persistence: request

Key pair name: -

Monitoring Your Spot Fleet

The Spot Fleet launches Spot Instances when your maximum price exceeds the Spot price and capacity is available. The Spot Instances run until they are interrupted or you terminate them.

To monitor your Spot Fleet using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request. The configuration details are available in the **Description** tab.

4. To list the Spot Instances for the Spot Fleet, choose the **Instances** tab.
5. To view the history for the Spot Fleet, choose the **History** tab.

Canceling your Spot Fleet Request

When you are finished using your Spot Fleet, you can cancel the Spot Fleet request. This cancels all Spot requests associated with the Spot Fleet, so that no new Spot Instances are launched for your Spot Fleet. You must specify whether the Spot Fleet should terminate its Spot Instances. If you terminate the instances, the Spot Fleet request enters the cancelled_terminating state. Otherwise, the Spot Fleet request enters the cancelled_running state and the instances continue to run until they are interrupted or you terminate them manually.

To cancel a Spot Fleet request using the console

1. Open the Spot console at https://console.aws.amazon.com/ec2spot/home/fleet.
2. Select your Spot Fleet request.
3. Choose **Actions**, and then choose **Cancel spot request**.
4. In **Cancel spot request**, choose **Confirm**.
5. Verify that the instances shut down at https://console.aws.amazon.com/ec2

Lab 4: Launching EC2 Spot Instances via an EC2 Fleet

Launching EC2 Spot Instances with On-Demand Instances via an EC2 Fleet

An *EC2 Fleet* contains the configuration information to launch a fleet—or group—of instances. In a single API call, a fleet can launch multiple instance types across multiple Availability Zones, using the On-Demand Instance, Reserved Instance, and Spot Instance purchasing models together. Using EC2 Fleet, you can define separate On-Demand and Spot capacity targets, specify the instance types that work best for your applications, and specify how Amazon EC2 should distribute your fleet capacity within each purchasing model.

EC2 Fleets generalize the set-it-and-forget-it model that we pioneered in Spot Fleet. EC2 Fleet provides a new building block that gives you the ability to create fleets that are composed of a combination of EC2 On-Demand, Reserved, and Spot Instances with a single API call. You tell us what you need, capacity and instance-wise, and we'll handle all the heavy lifting. We will launch, manage, monitor and scale instances as needed, without the need for scaffolding code. For more information on EC2 Fleets, review this blog post: https://aws.amazon.com/blogs/aws/ec2-fleet-manage-thousands-of-on-demand-and-spot-instances-with-one-request/ .

Alternatively, run the script **04-create-fleet.**

To create a new EC2 Fleet using the command line, run the following

```
$ aws ec2 create-fleet --launch-template-configs
LaunchTemplateSpecification="{LaunchTemplateName=TemplateForSpot,Version=1}"
--target-capacity-specification
TotalTargetCapacity=4,OnDemandTargetCapacity=1,DefaultTargetCapacityType=spot
```

Example return

```
{
"FleetId": "fleet-e678bfc6-c2b5-4d9f-8700-03b2db30b183"
}
```

Copy and paste the FleetId to notepad. We will use it during cleanup.

This EC2 Fleet has requested a total capacity of 4 instances- 1 On-Demand and 3 Spot.

Check them out by running

```
$ aws ec2 describe-fleets --fleet-ids fleet-e678bfc6-c2b5-4d9f-8700-03b2db30b183
```

Delete your EC2 Fleet

When you are finished using your EC2 Fleet, you can delete the EC2 Fleet and terminate all of the running instances.

To delete your EC2 Fleet and terminate the running instances

```
$ aws ec2 delete-fleets --fleet-ids fleet-e678bfc6-c2b5-4d9f-8700-03b2db30b183 --terminate-instances
```

If you have misplaced the fleet-id, you can see the running fleets using the following command.

```
$ aws ec2 describe-fleets | less
```

Lab 5: Finding Running Spot Instances

Finding Running Spot Instances

A Spot Instance runs until it is interrupted or you terminate it yourself.

Now that we’ve launched Spot Instances via RunInstances, Spot Fleet, EC2 Fleet, and an Auto Scaling group, let’s go check them out.

To find running Spot Instances using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**.
3. You can see both Spot Instance requests and Spot Fleet requests. If a Spot Instance request has been fulfilled, **Capacity** is the ID of the Spot Instance. For a Spot Fleet, **Capacity** indicates how much of the requested capacity has been fulfilled. To view the IDs of the instances in a Spot Fleet, choose the expand arrow, or select the fleet and then select the **Instances** tab.
4. Alternatively, in the navigation pane, choose **Instances**. In the top right corner, choose the **Show/Hide** icon, and then select **Lifecycle**. For each instance, **Lifecycle** is either *normal*, *spot*, or *scheduled*.

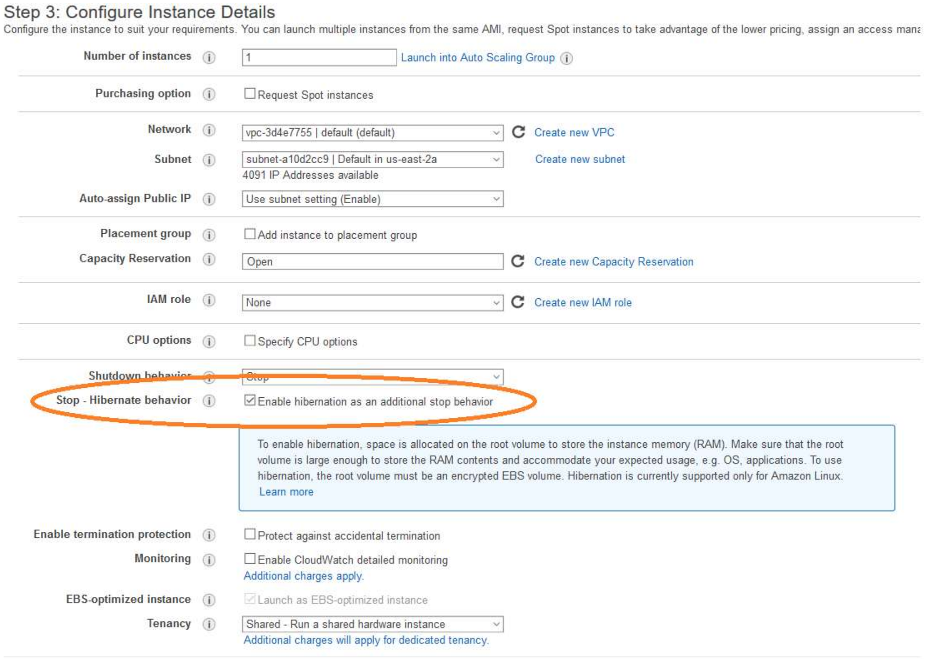
Lab 6: EC2 Hibernate

Hibernation of EC2 instances pauses execution and saves the running state to the root volume. In this lab, we look at hibernation, then hibernation with spot instances. Today, hibernation works with C, M, and R instances that have less than 150 GB of RAM. Because hibernation saves the running RAM image to disk, security best practices call for encryption of the root volume.

In the coming steps, we will create an EC2 instance with a long-running compute job, hibernate and restart the instance. Then, we will create a spot instance with hibernation. When a spot instance with hibernation receives an interruption, spot will persist RAM and running system state to EBS. The Spot service resumes the instance when capacity becomes available with a Spot price that is less than your specified maximum price

Create an EC2 image

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/
2. Launch Instance with Amazon Linux 2 AMI (HVM), SSD Volume Type
3. Select instance type **M4 Large,** then continue to
4. **Next: *Configure Instance Details***
5. Select a vpc and subnet for this instance
6. Select Enable hibernation as an additional stop behavior



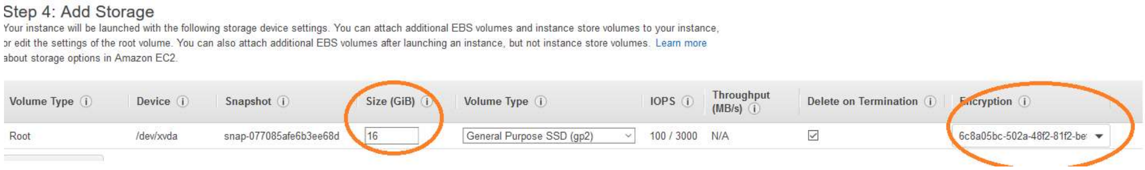
Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access mana

Number of instances	1	Launch into Auto Scaling Group
Purchasing option	□ Request Spot instances	
Network	vpc-3d4e7755 | default (default)	Create new VPC
Subnet	subnet-a10d2cc9 | Default in us-east-2a 4091 IP Addresses available	Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)	
Placement group	□ Add instance to placement group	
Capacity Reservation	Open	Create new Capacity Reservation
IAM role	None	Create new IAM role
CPU options	□ Specify CPU options	
Shutdown behavior	Stop	
Stop - Hibernate behavior	☑ Enable hibernation as an additional stop behavior	
Enable termination protection	□ Protect against accidental termination	
Monitoring	□ Enable CloudWatch detailed monitoring Additional charges apply.	
EBS-optimized instance	☑ Launch as EBS-optimized instance	
Tenancy	Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.	

To enable hibernation, space is allocated on the root volume to store the instance memory (RAM). Make sure that the root volume is large enough to store the RAM contents and accommodate your expected usage, e.g. OS, applications. To use hibernation, the root volume must be an encrypted EBS volume. Hibernation is currently supported only for Amazon Linux. Learn more

7. ***Next: Add Storage***
8. Adjust root volume size to 24 GB
9. Select Encryption

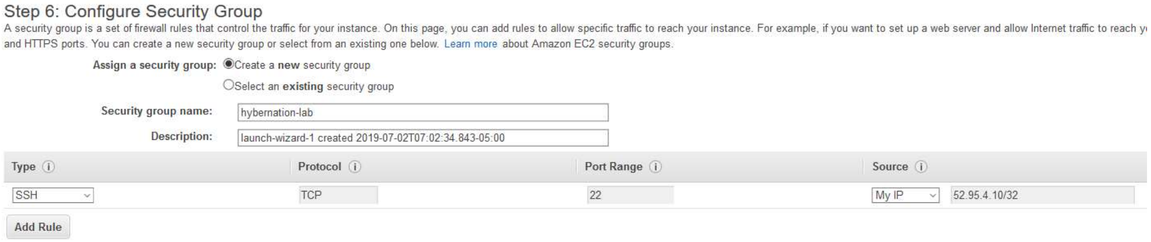


Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-077085afe6b3ee68d	16	General Purpose SSD (gp2)	100 / 3000	N/A	☑	6c8a05bc-502a-48f2-81f2-be

10. ***Next: Add Tags***
11. ***Next: Configure Security Group***
12. *Create a security group for this instance which permits access via SSH, port 22.*



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach y and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: hybernation-lab

Description: launch-wizard-1 created 2019-07-02T07:02:34.843-05:00

Type	Protocol	Port Range	Source
SSH	TCP	22	My IP 52.95.4.10/32

13. ***Review and Launch*** *the instance (create and download SSH key, if you don't already have one)*
14. ***Launch***
15. *SSH to the newly create instance*
16. *Download git, tmux and python3*

```
sudo yum install git python3 tmux -y
```

17. *Download script to calculate pi*

git clone https://github.com/dotstar/spotlab.git

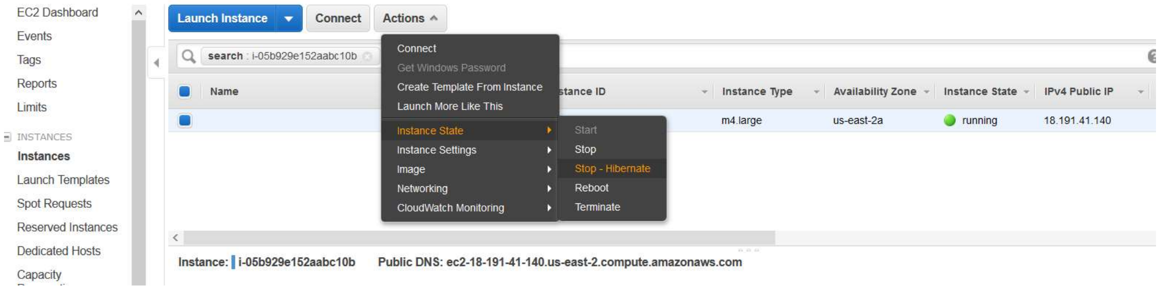
18. We will take advantage of tmux's ability to detach and reattach from a terminal session for the lab. Start tmux, then invoke the pi calculation job. Calcpi.py calculates pi to 999,999,999 digits, with output to the screen.

```
tmux

cd spotlab

python3 calcpi.py
```

19. Open the EC2 console and hibernate this instance
20. Select the instance; actions -> Instance State -> Stop-Hibernate



search : i-05b929e152aabc10b

Name	stance ID	Instance Type	Availability Zone	Instance State	IPv4 Public IP
		m4.large	us-east-2a	● running	18.191.41.140

Instance: i-05b929e152aabc10b Public DNS: ec2-18-191-41-140.us-east-2.compute.amazonaws.com

21. After the instance hibernates, restart it. Note that the IP address changes.
22. ssh to the image
23. reattach to the restarted/dehibernated tmux. You should see the calculations continue, where they left off. ^C to stop the python script.

```
tmux attach
```

24. Terminate the instance

    Actions -> Instance State -> Terminate

25. Extra Credit

Spot instances can also hibernate on interruption, restarting when capacity becomes available at your requested price.

A sample script to create a spot instance with hibernate is provided as 05-hibernation. This will need to be edited to reflect the SSH key, subnet, and security group detail for your account.

After editing 05-hibernation, run the script. This creates a spot instance which hibernates on instance interruption.

Start a spot instance, with hibernate. Use the GUI or sample script.

- How do you know it is a spot instance?
- How do you validate that hibernation is selected?
- How do you test this functionality?

Clean Up

In the event you didn't cleanup along the way, once again, here are the lab cleanup instructions.

Delete Your Auto Scaling Group

To delete your Auto Scaling group using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **Auto Scaling**, choose **Auto Scaling Groups**.
3. On the Auto Scaling groups page, select your Auto Scaling group, and choose **Actions**, **Delete**.
4. When prompted for confirmation, choose **Yes, Delete**.

Delete your EC2 Fleet

When you are finished using your EC2 Fleet, you can delete the EC2 Fleet and terminate all of the running instances.

To delete your EC2 Fleet and terminate the running instances

```
$ aws ec2 delete-fleets --fleet-ids fleet-e678bfc6-c2b5-4d9f-8700-03b2db30b183 --terminate-instances
```

Canceling your Spot Fleet Request

When you are finished using your Spot Fleet, you can cancel the Spot Fleet request. This cancels all Spot requests associated with the Spot Fleet, so that no new Spot Instances are launched for your Spot Fleet. You must specify whether the Spot Fleet should terminate its Spot Instances. If you terminate the instances, the Spot Fleet request enters the cancelled_terminating state. Otherwise, the Spot Fleet request enters the cancelled_running state and the instances continue to run until they are interrupted or you terminate them manually.

To cancel a Spot Fleet request using the console

1. Open the Spot console at https://console.aws.amazon.com/ec2spot/home/fleet.
2. Select your Spot Fleet request.
3. Choose **Actions**, and then choose **Cancel spot request**.
4. In **Cancel spot request**, choose **Confirm**.

Terminating a Spot Instance

To terminate a Spot Instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance, and choose **Actions**, **Instance State**, **Terminate**.
4. Choose **Yes, Terminate** when prompted for confirmation.

Deleting a Launch Template

If you no longer require a launch template, you can delete it. Deleting a launch template deletes all of its versions.

To delete a launch template

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Launch Templates** and select the launch template.
3. Choose **Actions**, **Delete template**.
4. Choose **Delete launch template**.