Milano | 23 Maggio 2024

aws SUMMIT

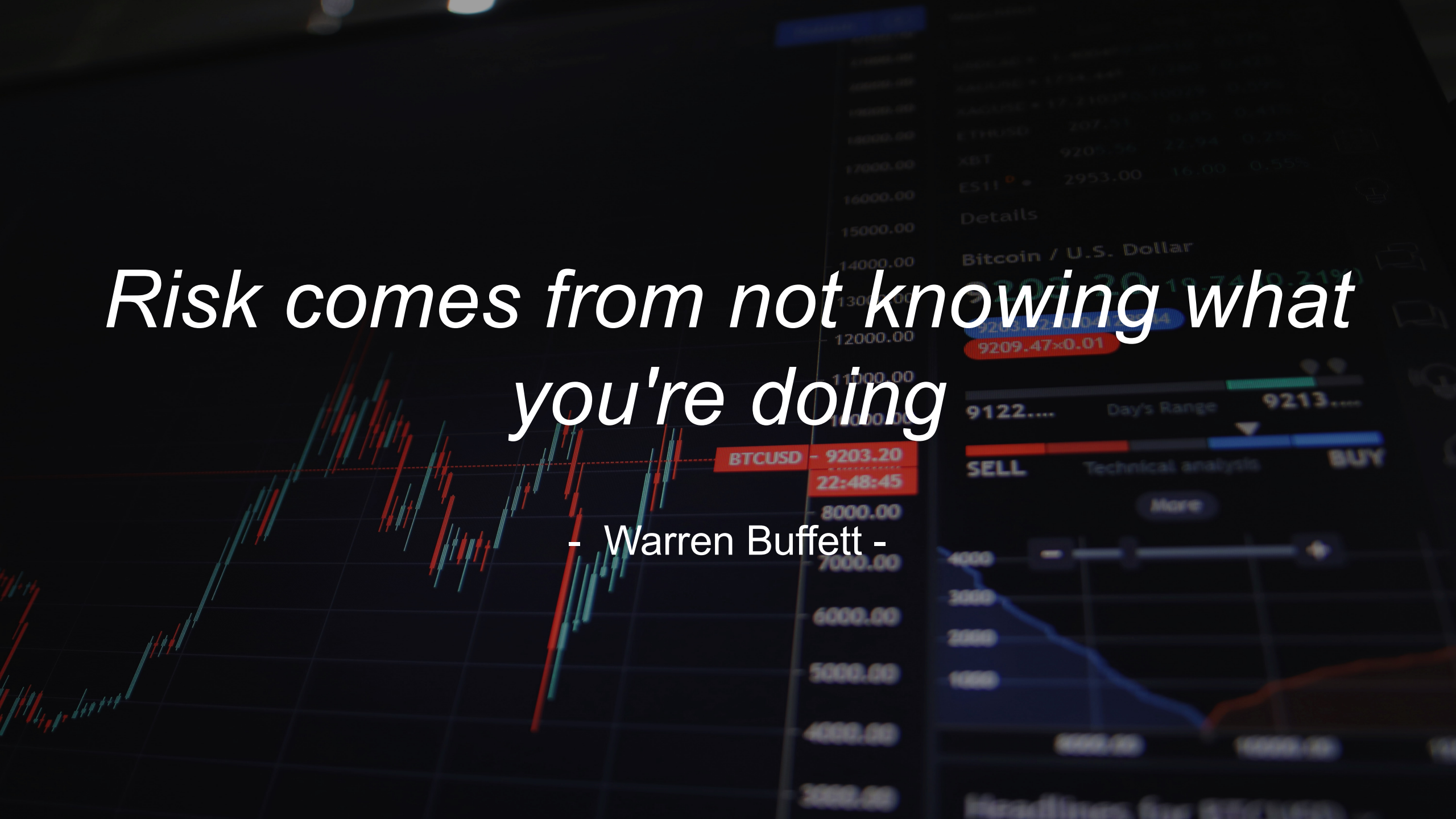
COM-403

Dominare le AWS Spot Instances: una guida pratica con AWS Fault Injection Simulation

Paolo Latella

Cloud Advisor, Recube
AWS Authorized Instructor
AWS Hero

https://www.linkedin.com/in/paololatella/
@LatellaPaolo


Risk comes from not knowing what you're doing
- Warren Buffett -
BTCUSD - 9203.20
22:48:45
Bitcoin / U.S. Dollar
9209.47×0.01
Day's Range
SELL
BUY
Technical analysis
More

AWS Spot Instances - Concepts

Spot capacity pool

A set of unused EC2 instances with the same instance type (for example, m5.large) and Availability Zone.

Spot price

The current price of a Spot Instance per hour.

Spot Instance request

Requests a Spot Instance. When capacity is available, Amazon EC2 fulfills your request.

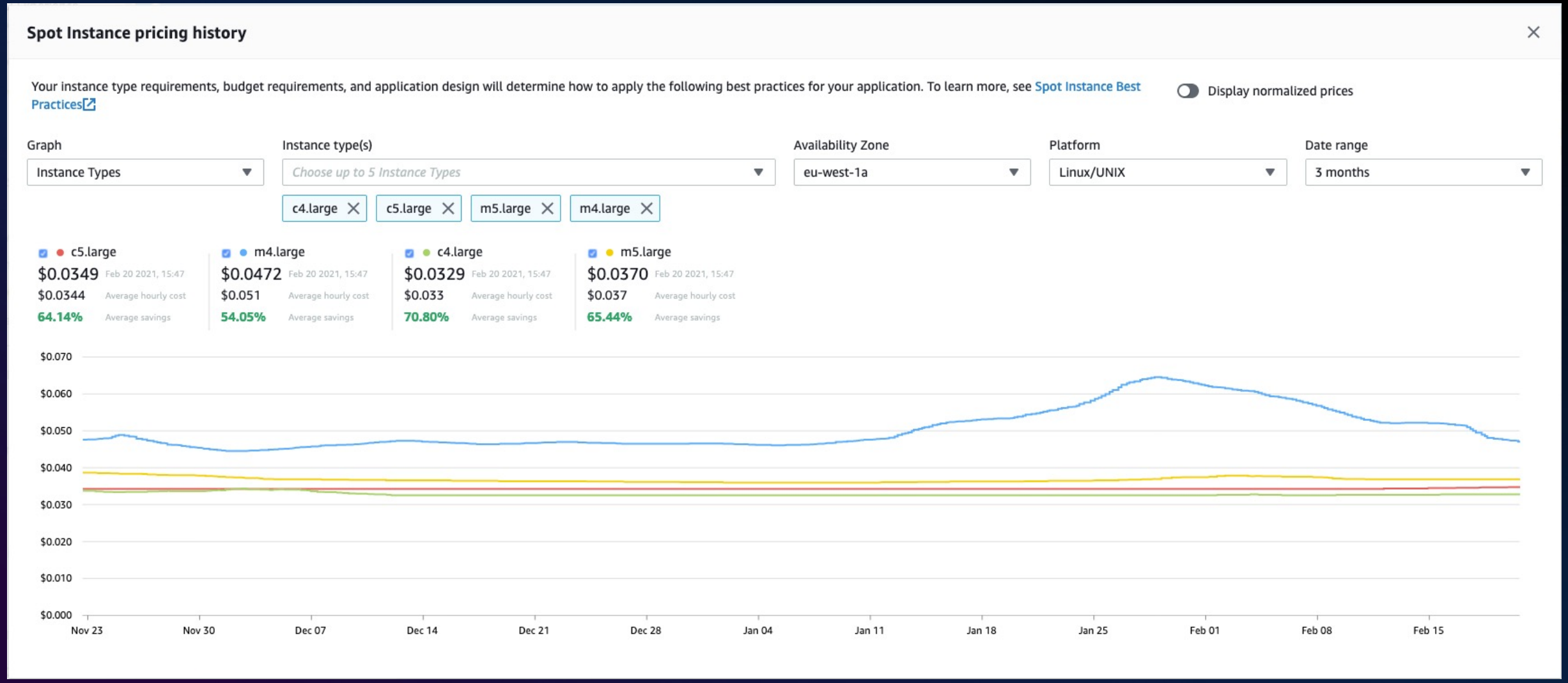
EC2 instance rebalance recommendation

Amazon EC2 signal to notify you that a Spot Instance is at an elevated risk of interruption.

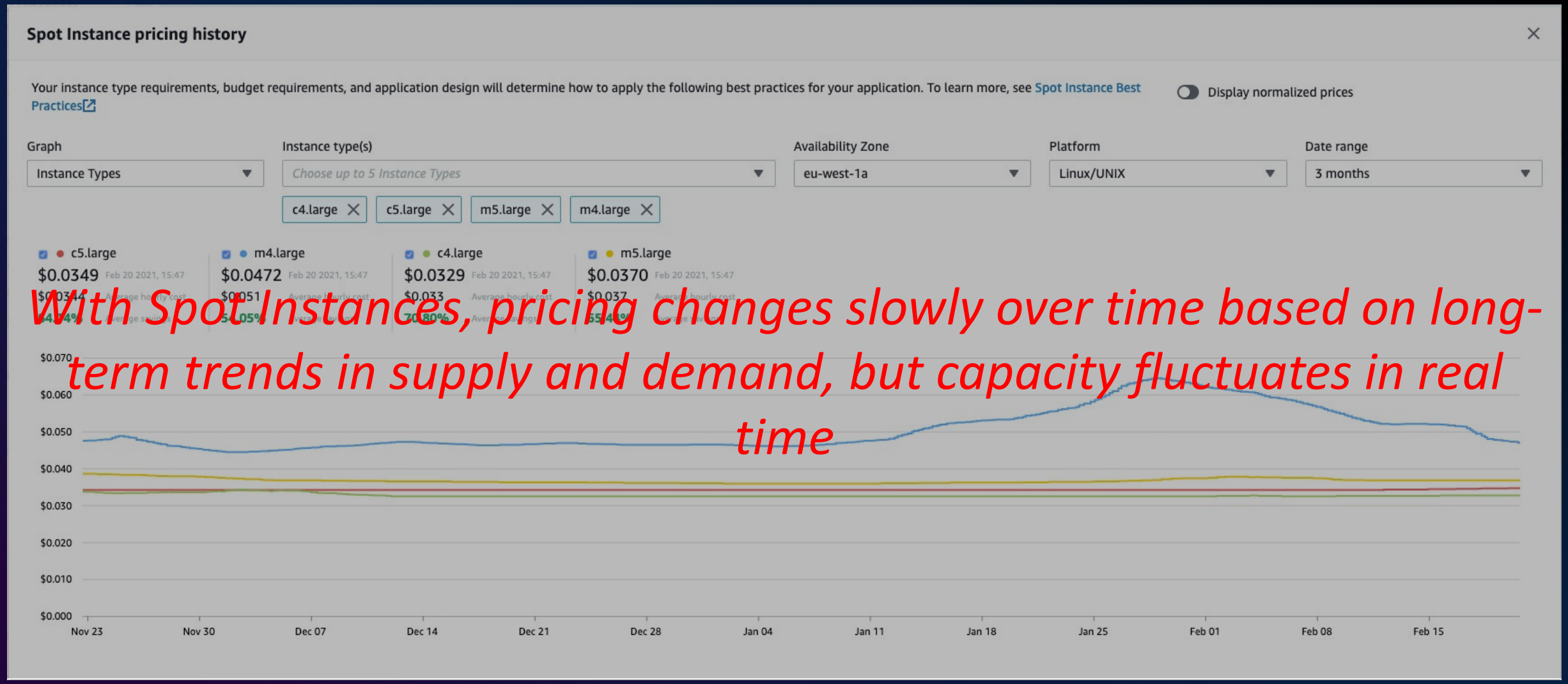
Spot Instance interruption

Amazon EC2 provides an interruption notice two-minute before it is interrupted.

AWS Spot Instances – Pricing History



AWS Spot Instances – Pricing History



With Spot Instances, pricing changes slowly over time based on long-term trends in supply and demand, but capacity fluctuates in real time

AWS Spot Instances - Saving

sfr-9661372b-d504-4f83-8efa-73e422ab0f03

Description | Tags | Instances | History | **Savings** | Scheduled Scaling | Auto Scaling

Last 3 days | Last hour

A high-level summary of your savings across all of your running and recently terminated Spot Instances.
For detailed reporting on your account-level Spot usage, visit Cost Explorer

Spot usage and savings

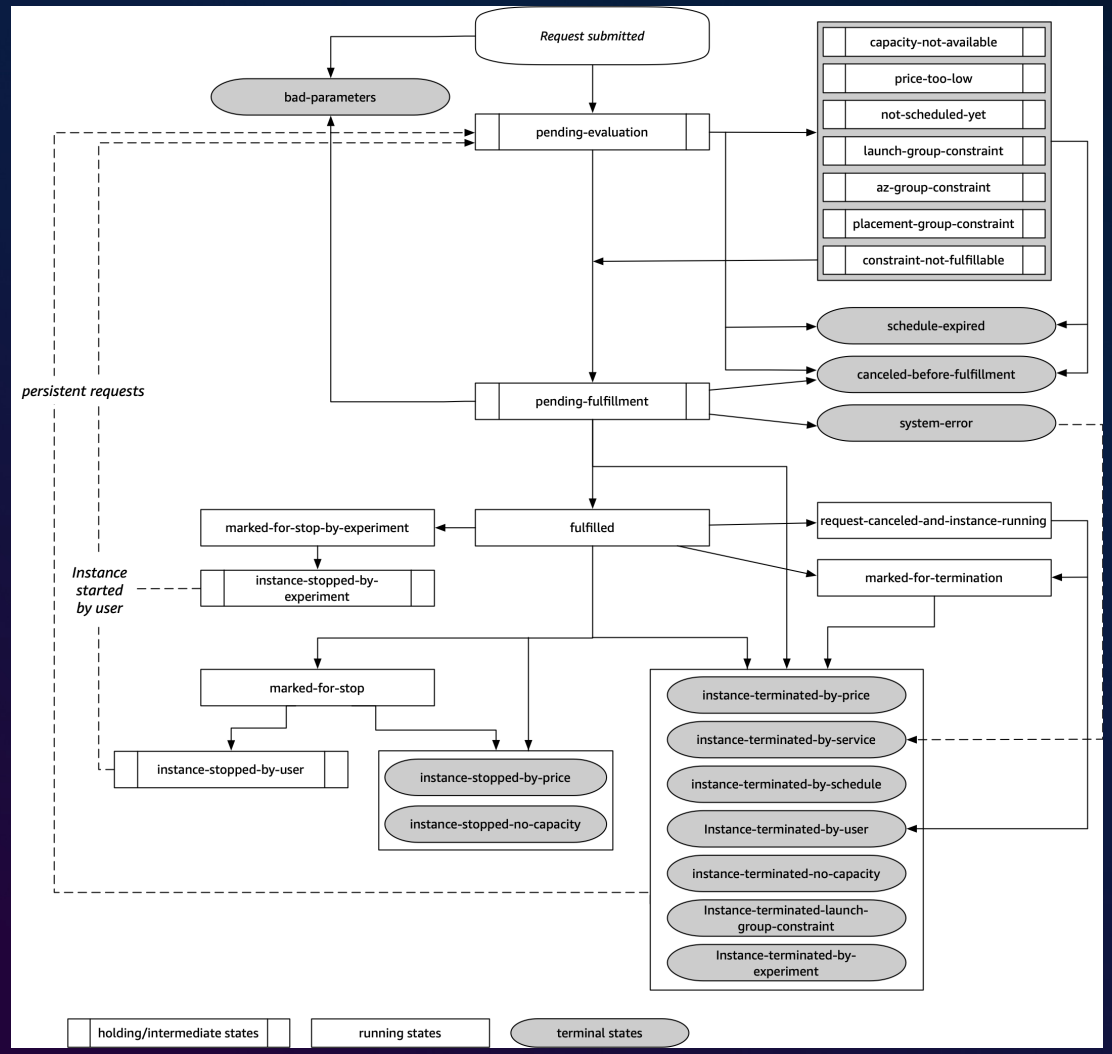
3	4	20	$0.26	$0.12	54%
Spot Instances	vCPU-hours	Mem(GiB)-hours	On-Demand total	Spot total	Savings
				$0.0302	$0.0060
				Average cost per VCPU-hour	Average cost per mem(GiB)-hour

Details

Instance type	vCPU hours	Memory (GiB) hours	Total Spot cost (USD)	Total savings
c6i.large (1)	2	4	$0.04	55%
r7a.medium (2)	2	16	$0.08	53%

* Spot savings are estimated savings and may differ from actual savings. This is because the savings shown on this page do not include the billing adjustments for your usage.

Spot Instances - Workflow



az-group-constraint
canceled-before-fulfilment
constraint-not-fulfillable
fulfilled
instance-stopped-by-price
instance-stopped-by-user
instance-stopped-no-capacity
instance-terminated-by-price
instance-terminated-by-schedule
instance-terminated-by-service
instance-terminated-by-user
instance-terminated-launch-group-constraint
instance-terminated-no-capacity
launch-group-constraint
limit-exceeded
marked-for-stop
marked-for-termination
not-scheduled-yet
pending-evaluation
placement-group-constraint
price-too-low
request-canceled-and-instance-running
schedule-expired

Spot Instance - Rebalance recommendations

- Notifies you when a Spot Instance is at elevated risk of interruption.
- Arrive sooner (best-effort) than the two-minute Spot Instance interruption notice
- Available as a EventBridge event and as an item in the instance metadata on the Spot Instance.
- Actions
  - Graceful shutdown
  - Prevent new work from being scheduled
  - Proactively launch new replacement instances

Rebalance recommendations - Signal

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Instance Rebalance Recommendation",
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-1234567890abcdef0"],
    "detail": {
        "instance-id": "i-1234567890abcdef0"
    }
}
```

Spot Instance – Interruption Notification

- Notifies you when EC2 reclaim the Instance
- On-Demand Instance specified in a Spot Fleet cannot be interrupted.
- Reasons for interruption
  - Capacity
  - Price
  - Constraints
- Interruption behavior
  - Terminate
  - Stop
  - Hibernate

Interruption Notification - Signal

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Spot Instance Interruption Warning",
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2a:instance/i-1234567890abcdef0"],
    "detail": {
        "instance-id": "i-1234567890abcdef0",
        "instance-action": "action"
    }
}
```

AWS Spot Instances - APIs

API	When to use?
`CreateAutoScalingGroup`	You need multiple instances with either a single configuration or a mixed configuration and you want to leverage Autoscaling capabilities
`CreateFleet`	You need multiple instances with either a single configuration or a mixed configuration and you don’t need auto scaling or want to customize workflow
`RunInstances`	You simply want to launch a specified number of spot instances using an AMI and one specific instance type
`RequestSpotFleet`	We strongly discourage using the RequestSpotFleet API because it is a legacy API with no planned investment
`RequestSpotInstances`	We strongly discourage using the RequestSpotInstances API because it is a legacy API with no planned investment.

```
AWS::EC2::SpotFleet
aws ec2 create-fleet
```

Spot Fleet – Features (1/2)

```yaml
EC2SpotFleet:
  Type: AWS::EC2::SpotFleet
  Properties:
    SpotFleetRequestConfigData:
      IamFleetRole: arn:aws:iam::983441761380:role/aws-ec2-spot-fleet-tagging-role
      TargetCapacity: !Ref SpotTargetCapacity
      TargetCapacityUnitType: units
      ReplaceUnhealthyInstances: True
      SpotMaintenanceStrategies:
        CapacityRebalance:
          ReplacementStrategy: launch-before-terminate
          TerminationDelay: 300
      Type: maintain
      AllocationStrategy: 'priceCapacityOptimized'
      SpotMaxTotalPrice: !Ref SpotBudget
      OnDemandTargetCapacity: !Ref OnDemandTargetCapacity
      OnDemandAllocationStrategy: lowestPrice
      OnDemandMaxTotalPrice: !Ref OnDemandBudget
      LaunchTemplateConfigs:
        - LaunchTemplateSpecification:
            LaunchTemplateId: !Ref EC2LaunchTemplate
            Version: !GetAtt EC2LaunchTemplate.LatestVersionNumber
      InstanceInterruptionBehavior: 'terminate'
```

- TargetCapacity
- TargetCapacityUnitType
  - instance, vCPU, RAM, I/O
- CapacityRebalance
  - launch
  - launch-before-terminate
- Type
  - maintain, request
- AllocationStrategy
  - priceCapacityOptimized
  - capacityOptimized
  - diversified
  - LowestPrice

Spot Fleet – Features (2/2)

```yaml
EC2SpotFleet:
  Type: AWS::EC2::SpotFleet
  Properties:
    SpotFleetRequestConfigData:
      IamFleetRole: arn:aws:iam::983441761380:role/aws-ec2-spot-fleet-tagging-role
      TargetCapacity: !Ref SpotTargetCapacity
      TargetCapacityUnitType: units
      ReplaceUnhealthyInstances: True
      SpotMaintenanceStrategies:
        CapacityRebalance:
          ReplacementStrategy: launch-before-terminate
          TerminationDelay: 300
      Type: maintain
      AllocationStrategy: 'priceCapacityOptimized'
      SpotMaxTotalPrice: !Ref SpotBudget
      OnDemandTargetCapacity: !Ref OnDemandTargetCapacity
      OnDemandAllocationStrategy: lowestPrice
      OnDemandMaxTotalPrice: !Ref OnDemandBudget
      LaunchTemplateConfigs:
        - LaunchTemplateSpecification:
            LaunchTemplateId: !Ref EC2LaunchTemplate
            Version: !GetAtt EC2LaunchTemplate.LatestVersionNumber
      InstanceInterruptionBehavior: 'terminate'
```

- SpotMaxTotalPrice
- OnDemandTargetCapacity
- OnDemandAllocationStrategy
  - lowestPrice
  - prioritized
- OnDemandMaxTotalPrice
- LaunchTemplateSpecification
- InstanceInterruptionBehavior
  - Hibernate
  - Stop
  - Terminate

Launch Specifications vs Launch Template

`LaunchSpecification` is an older way to define configuration

- **DirectConfiguration**: You specify instance details directly in the Spot Fleet request.
- **Limited Reusability**: Each Spot Fleet request needs to have its configuration
- **Less Flexibility**: Does not support some advanced features available in LaunchTemplate, such as versioning

`LaunchTemplate` is a newer and more flexible way

- **Reusability**: can be reused across multiple Spot Fleets or Auto Scaling groups
- **Versioning**: maintain and manage different versions of your configuration
- **Advanced Features**: Provides access to advanced EC2 features such as Elastic Inference, T2/T3 Unlimited,

Launch Specifications vs Launch Template

`LaunchSpecification` is an older way to define configuration

- **DirectConfiguration**: You specify instance details directly in the Spot Fleet request.
- **Limited Reusability**: Each Spot Fleet request needs to have its configuration
- **Less Flexibility**: Does not support some advanced features available in LaunchTemplate, such as versioning

LEGACY

`LaunchTemplate` is a newer and more flexible way

- **Reusability**: can be reused across multiple Spot Fleets or Auto Scaling groups
- **Versioning**: maintain and manage different versions of your configuration
- **Advanced Features**: Provides access to advanced EC2 features such as Elastic Inference, T2/T3 Unlimited,

Spot Fleet - Allocation strategies

- price-capacity-optimized: Optimizes for both price and capacity to minimize interruptions and cost (recommended)
- capacity-optimized: Launches Spot Instances in the most available pools to minimize disruptions.
- diversified: The Spot Instances are distributed across all Spot capacity pools.
- lowest-price: Selects the lowest priced pools, which might lead to higher interruption rates.
- InstancePoolsToUseCount: selects the lowest priced spot pools and evenly allocates your target across the Spot pools that you specify.

```
AWS::AutoScaling::AutoScalingGroup
aws autoscaling create-auto-scaling-group
```

Autoscaling Group - Features

```yaml
EC2SpotAutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    AutoScalingGroupName: AWSSummitSpotAutoscaling
    CapacityRebalance: True
    VPCZoneIdentifier:
      - subnet-0af751c807e1fe2f9
      - subnet-0f7165dbbf6c2cdc9
      - subnet-013ee744214df5698
    MinSize: 2
    MaxSize: 4
    DesiredCapacity: 2
    DesiredCapacityType: units
    MixedInstancesPolicy:
      InstancesDistribution:
        OnDemandAllocationStrategy: prioritized
        OnDemandBaseCapacity: 2
        OnDemandPercentageAboveBaseCapacity: 50
        SpotAllocationStrategy: price-capacity-optimized
        SpotMaxPrice : "1"
      LaunchTemplate:
        LaunchTemplateSpecification:
          LaunchTemplateId: !Ref EC2LaunchTemplate
          Version: !GetAtt EC2LaunchTemplate.LatestVersionNumber
        Overrides:
          - InstanceType: t3.large
          - InstanceType: t3.medium
          - InstanceType: t3.small
```

- DesiredCapacityType
- Capacity Rebalance
- InstancesDistribution:
  - OnDemandAllocationStrategy
  - OnDemandBaseCapacity
  - OnDemandPercentageAboveBaseCapacity
  - SpotAllocationStrategy
  - SpotMaxPrice
- LaunchTemplate
  - LaunchTemplateSpecification
  - Overrides

Autoscaling Group - Allocation strategies

- price-capacity-optimized: Optimizes for both price and capacity to minimize interruptions and cost (recommended)
- capacity-optimized: Launches Spot Instances in the most available pools to minimize disruptions.
- capacity-optimized-prioritized: Prioritizes instance types while optimizing for capacity to minimize disruptions.
- lowest-price: Selects the lowest priced pools, which might lead to higher interruption rates.

```
AWS::EC2::LaunchTemplate
aws ec2 create-launch-template
```

Launch Template - Features

```yaml
EC2LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName: LaunchTemplateForSpotFleet
    LaunchTemplateData:
      ImageId: !FindInMap [AWSRegionArch2AMI, !Ref 'AWS::Region', 'HVM64']
      InstanceRequirements:
        VCpuCount:
          Min: 1
          Max: 4
        MemoryMiB:
          Min: 4096
          Max: 16384
        InstanceGenerations:
          - current
        AllowedInstanceTypes:
          - m5*
          - c*
          - r*
        SpotMaxPricePercentageOverLowestPrice: 20
        OnDemandMaxPricePercentageOverLowestPrice: 20
      TagSpecifications:
        - ResourceType: "instance"
          Tags:
            - Key: "Name"
              Value: "AWSSummitSpotFleetInstance"
            - Key: "Interrupt"
              Value: "Yes"
```

- InstanceRequirements
  - VCpuCount
  - MemoryMib
  - NetworkBandwidthGbps
  - InstanceGeneration
  - AllowedInstanceType
  - SpotMaxPricePercentageOverLowestPrice
  - OnDemandMaxPricePercentOverLowestPrice

Launch Template – Instance Weight

`SpotPrice` - from Price *per instance hour* to price *per unit hour*

Application Requirements: 2 vCPU and 8 GB of RAM
Workload Requirements: 20 Istances

Instance Type	vCPU	RAM	Pih	Puh	Weight
m5.large	2	8	0.107	0.107	1
m5.xlarge	4	16	0.214	0.107	2
m5.2xlarge	8	32	0.428	0.107	4

Spot Instances launched = Target capacity / Instance weight.

- *Example of Launched Instances:*
  - *2 x m5.2xlarge + 5 x m5.xlarge + 2 x m5.large*

Instance Weight - Example

```
LaunchTemplateConfigs:
  - LaunchTemplateSpecification:
      LaunchTemplateId: !Ref EC2LaunchTemplate
      Version: !GetAtt EC2LaunchTemplate.LatestVersionNumber
    Overrides:
      - InstanceType: m5.large
        WeightedCapacity: 1
      - InstanceType: m5.xlarge
        WeightedCapacity: 2
      - InstanceType: m5.2xlarge
        WeightedCapacity: 4
InstanceInterruptionBehavior: 'terminate'
```

Instance Weight - Example

```
LaunchTemplateConfigs:
  - LaunchTemplateSpecification:
      LaunchTemplateId: !Ref EC2LaunchTemplate
      Version: !GetAtt EC2LaunchTemplate.LatestVersionNumber
    Overrides:
      - InstanceType: m5.large
        WeightedCapacity: 1
      - InstanceType: m5.xlarge
        WeightedCapacity: 2
      - InstanceType: m5.2xlarge
        WeightedCapacity: 4
InstanceInterruptionBehavior: 'terminate'
```

Note

If you specify `InstanceRequirements`, you can't specify `InstanceType`.

InstanceType + Weighted Capacity
=
InstanceRequirement + Control

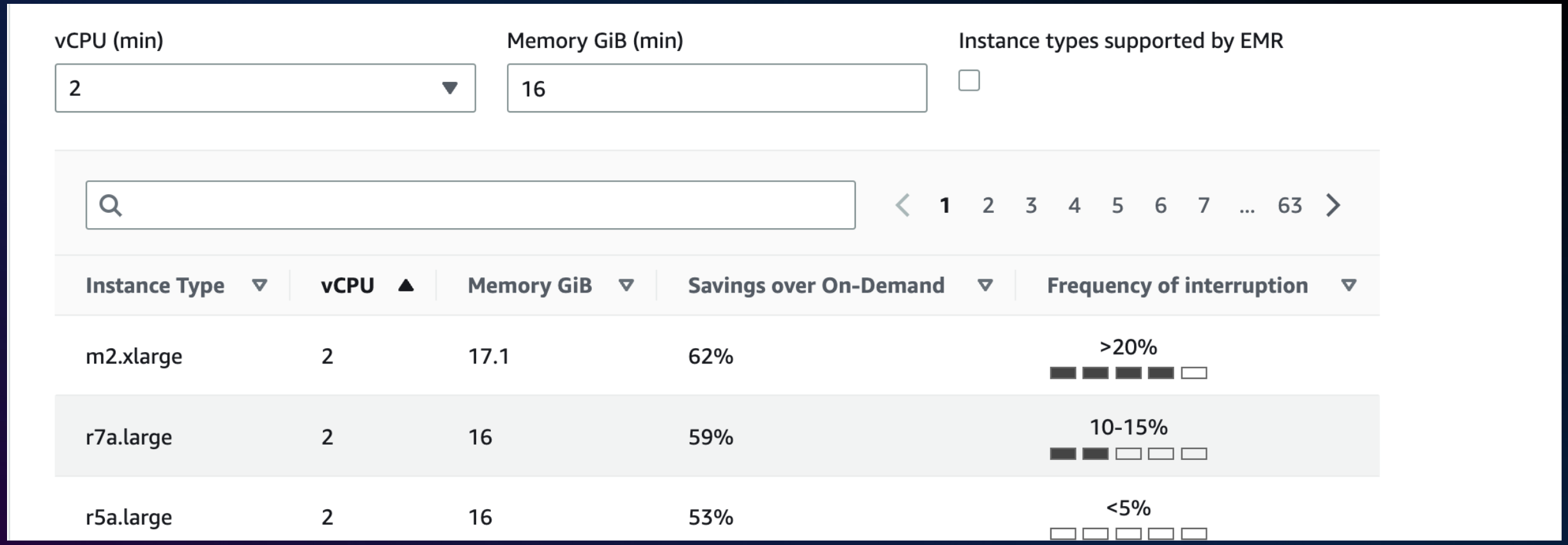
Capacity Requirement – Spot Placement Score

```
{
    "TargetCapacity": 2,
    "RegionNames": [ "eu-west-1", "us-east-1" ],
    "SingleAvailabilityZone": true,
    "InstanceRequirementsWithMetadata":
    {
        "ArchitectureTypes": [ "x86_64" ],
        "VirtualizationTypes": [ "hvm" ],
        "InstanceRequirements":
        {
            "VCpuCount": { "Min": 1, "Max": 4 },
            "MemoryMiB": { "Min": 1024, "Max": 16384},
            "InstanceGenerations": [ "current" ],
            "RequireHibernateSupport": false
        }
    },
    "DryRun": false,
    "MaxResults": 10,
    "NextToken": ""
}
```

```
aws ec2 get-spot-placement-scores --
region eu-west-1 --cli-input-json
file://attributes.json
```

```
SpotPlacementScores:
- AvailabilityZoneId: euw1-az2
  Region: eu-west-1
  Score: 9
- AvailabilityZoneId: use1-az5
  Region: us-east-1
  Score: 9
- AvailabilityZoneId: use1-az3
  Region: us-east-1
  Score: 3
```

Frequency of Interruption – Spot Advisor



AWS::FIS::ExperimentTemplate

AWS FIS – Test Spot Interruption

```yaml
FISExperimentTemplate:
  Type: 'AWS::FIS::ExperimentTemplate'
  Properties:
    RoleArn: !GetAtt FISRole.Arn
    Description: 'Test EC2 Spot instance interruption notices'
    Actions:
      stopSpotInstances:
        ActionId: 'aws:ec2:send-spot-instance-interruptions'
        Targets:
          SpotInstances: selectOneInstanceByTag
        Parameters:
          durationBeforeInterruption: 'PT5M'
    Targets:
      selectOneInstanceByTag:
        ResourceTags:
          "Name": "AWSSummitSpotInstance"
          "Interrupt": "Yes"
        ResourceType: 'aws:ec2:spot-instance'
        SelectionMode: 'COUNT(1)'
    StopConditions:
      - Source: 'none'
    LogConfiguration:
      CloudWatchLogsConfiguration:
        LogGroupArn: !GetAtt FISLogGroup.Arn
      LogSchemaVersion: 1
    Tags:
      Project: "AWSSummit24"
      Environment: "Demo"
```

`aws:ec2:send-spot-instance-interruptions`

Tools

- https://github.com/aws-samples/ec2-spot-interruption-dashboard
- https://github.com/aws/amazon-ec2-metadata-mock

Thank you!

Please complete the session survey in the mobile app