AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

CMP 205

Scaling Kubernetes for less with Amazon EC2 Spot Instances

Steve Cole
Senior EC2 Spot Specialist Solutions Architect
Amazon Web Services

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

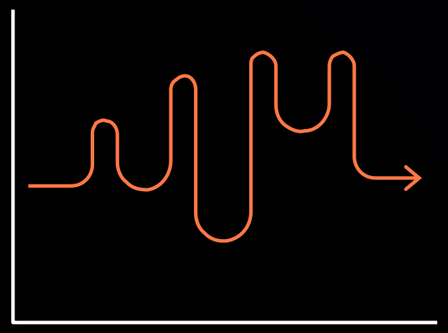
- Amazon EC2 Spot Overview
- Amazon EKS Overview
- Begin lab
- Node groups
- Continue lab

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EC2 purchase options

On-Demand Instances

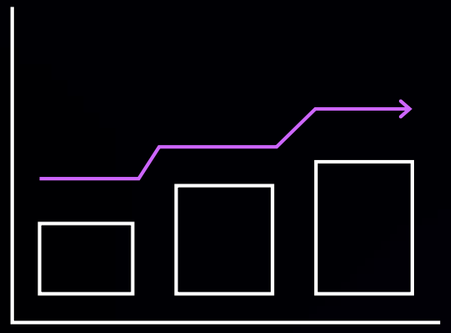
Pay for compute capacity by **the second** with no long-term commitment



Spiky workloads, to define needs

Reserved Instances

Make a 1-year or 3-year commitment, receive a **significant discount** on On-Demand prices



Committed and steady-state usage

Savings plan

Same great discounts as Amazon EC2 RIs, **more flexibility**



Flexible access to compute

Spot Instances

Spare Amazon EC2 capacity at **savings of up to 90%** on On-Demand prices



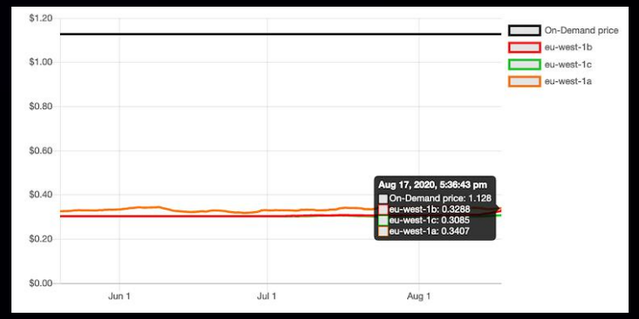
Fault-tolerant, flexible, stateless workloads

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What are EC2 Spot Instances?

LEVERAGE SPARE CAPACITY ON-DEMAND WITH STEEP DISCOUNTS, PAY BY THE SECOND







Spare Amazon EC2 capacity

Same infrastructure as On-Demand

Save up to 90% compared to On-Demand pricing

Spot price is based on long-term supply and demand trends of EC2 instances **(no bidding)**

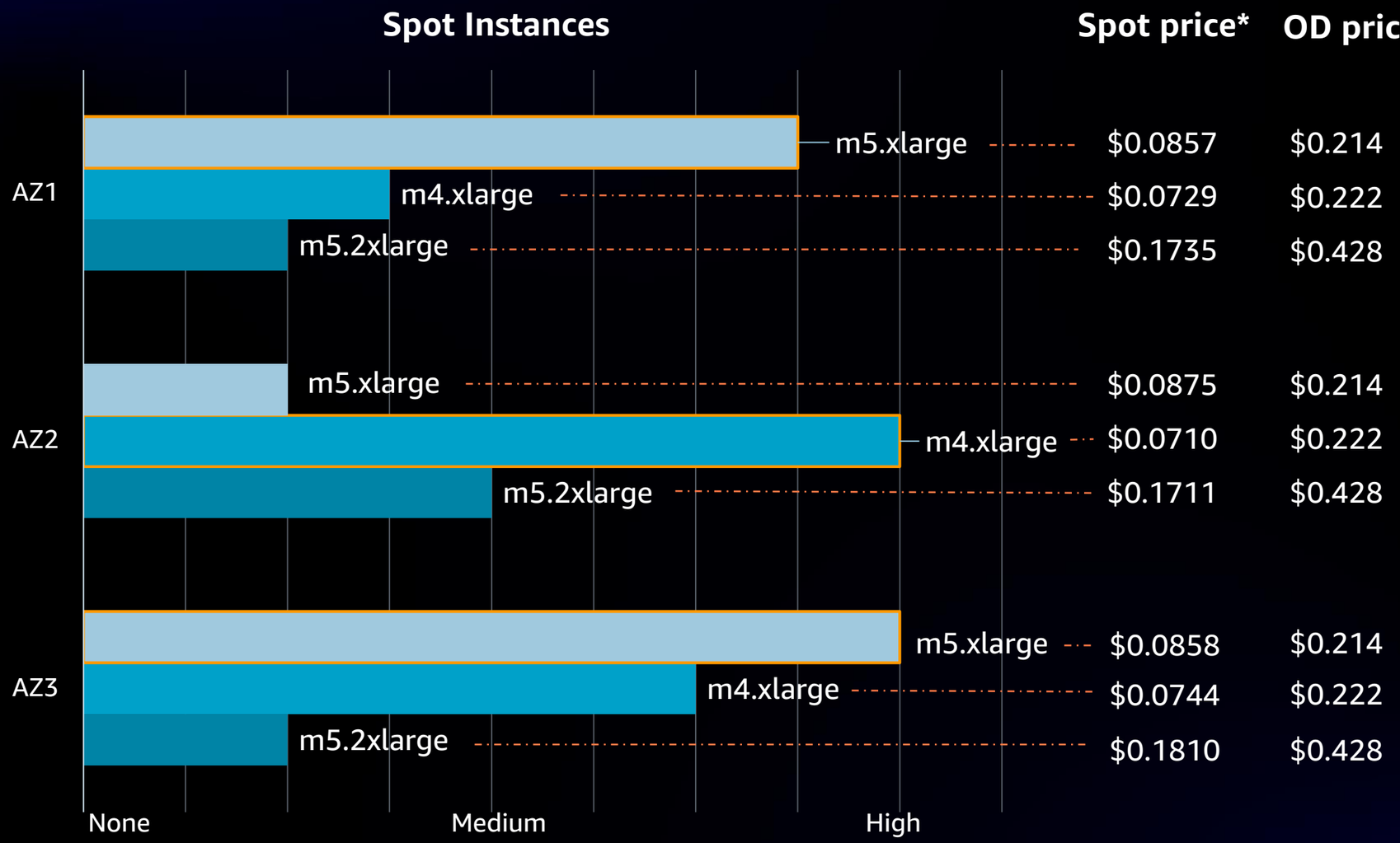
Interruptible

Spot instances can be interrupted by a 2-minute warning if Amazon EC2 needs the capacity back

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Understanding Spot capacity pools



Each instance type

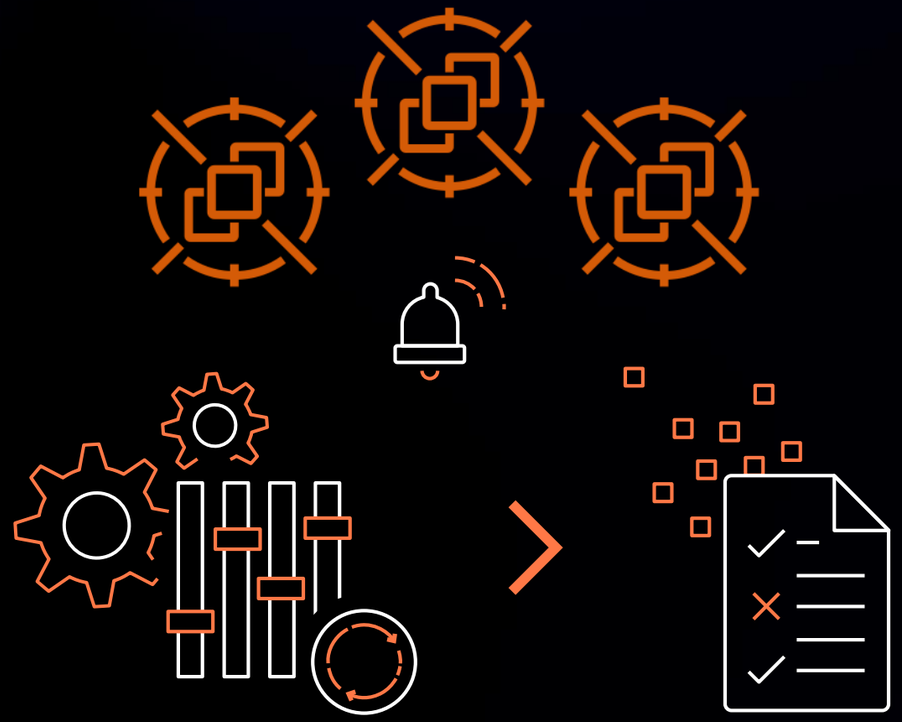
in each Availability Zone

in every region

is a separate
Spot capacity pool

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

EC2 instance rebalance recommendation for Spot Instances



A new signal

- Notifies you when a Spot Instance is at elevated risk of interruption
- Can arrive sooner than the 2-minute Spot Instance interruption notice
- Helps to proactively rebalance your workload to new or existing Spot Instances that are not at elevated risk of interruption

What do you do?

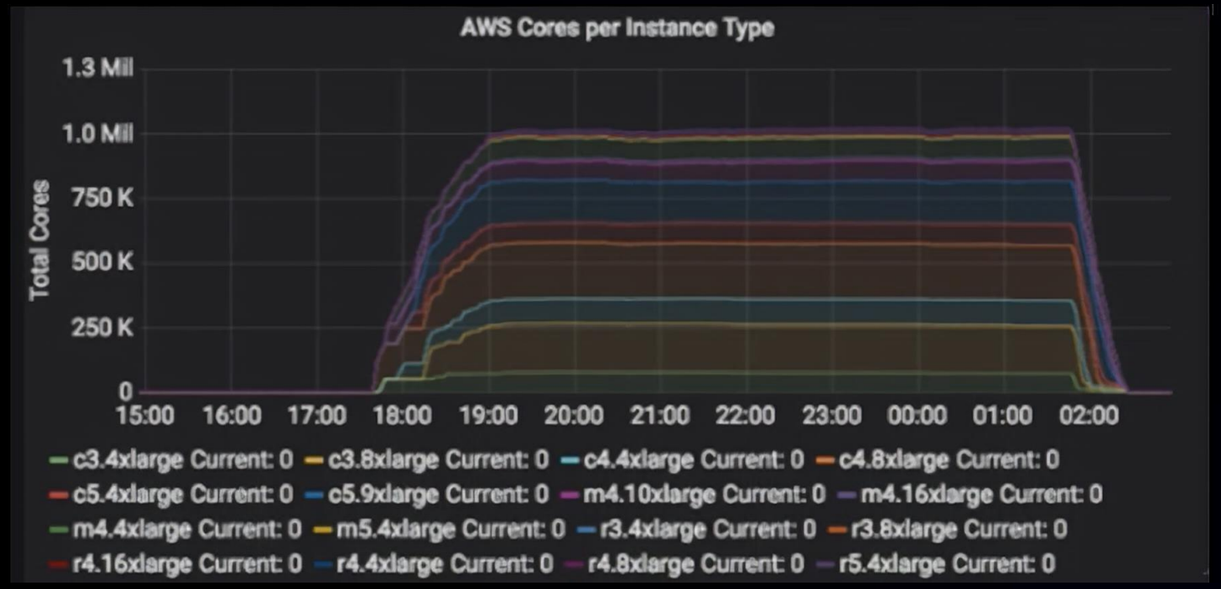
- Start checkpointing your work early to save as much state as possible
- Prevent scheduling new work on instances at elevated risk of interruption, thus increasing chance of completing work

Learn more: https://amzn.to/36pHhWM

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Spare capacity at scale



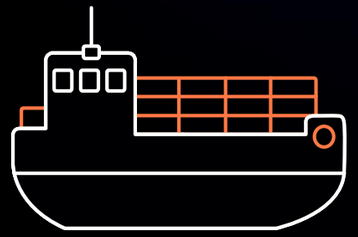
Western Digital®

More than 2.3 million simulation jobs on a single HPC cluster of 1 million vCPUs built using Amazon EC2 Spot Instances

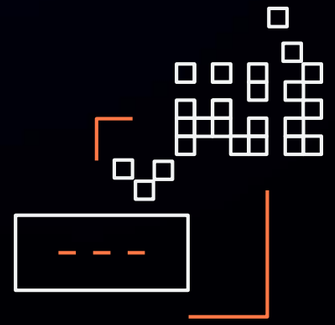
Time to results:
20 days → 8 hours

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Spot Instances are great for fault-tolerant workloads



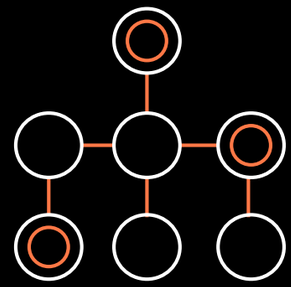
Containers



Big data



CI/CD



Batch



Web services



Machine learning



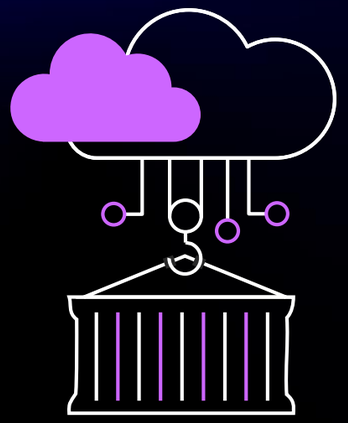
HPC

Lean on Spot for these workloads

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Containers + Spot = A perfect match



- Containers are often stateless, fault-tolerant, and a great fit for Spot Instances
- Deploy containerized workloads and easily manage clusters at any scale at a fraction of the cost with Spot Instances
- Spot Instances can be used with Amazon ECS or Amazon EKS to run any containerized workload

skyscanner

Skyscanner is a travel fare aggregator website and travel metasearch engine based in Edinburgh, Scotland

“We are currently tracking 74% savings over all regions.”

—Paul Gillespie,
Principal Architect/Tribe Lead

lyft yelp MapBox Expedia Caltech

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS



Upstream Kubernetes

- Use standard Kubernetes APIs
- Works with community tools

Highly available

- Built for production workloads
- All clusters are highly available
- Backed by a 99.9% SLA

Integrated

- Integrates AWS products and services: VPC networking, Elastic Load Balancing (ELB), IAM permissions, CloudWatch and more

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Handling interruptions



Identify	Spot Instance is being reclaimed
2-minute notification	Gracefully prepare the node for termination
Taint	Cordon to prevent new pods from being placed
Drain	Drain connections on the running pods
Replace	Replace to maintain the desired capacity

Deployment: https://github.com/aws/aws-node-termination-handler

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

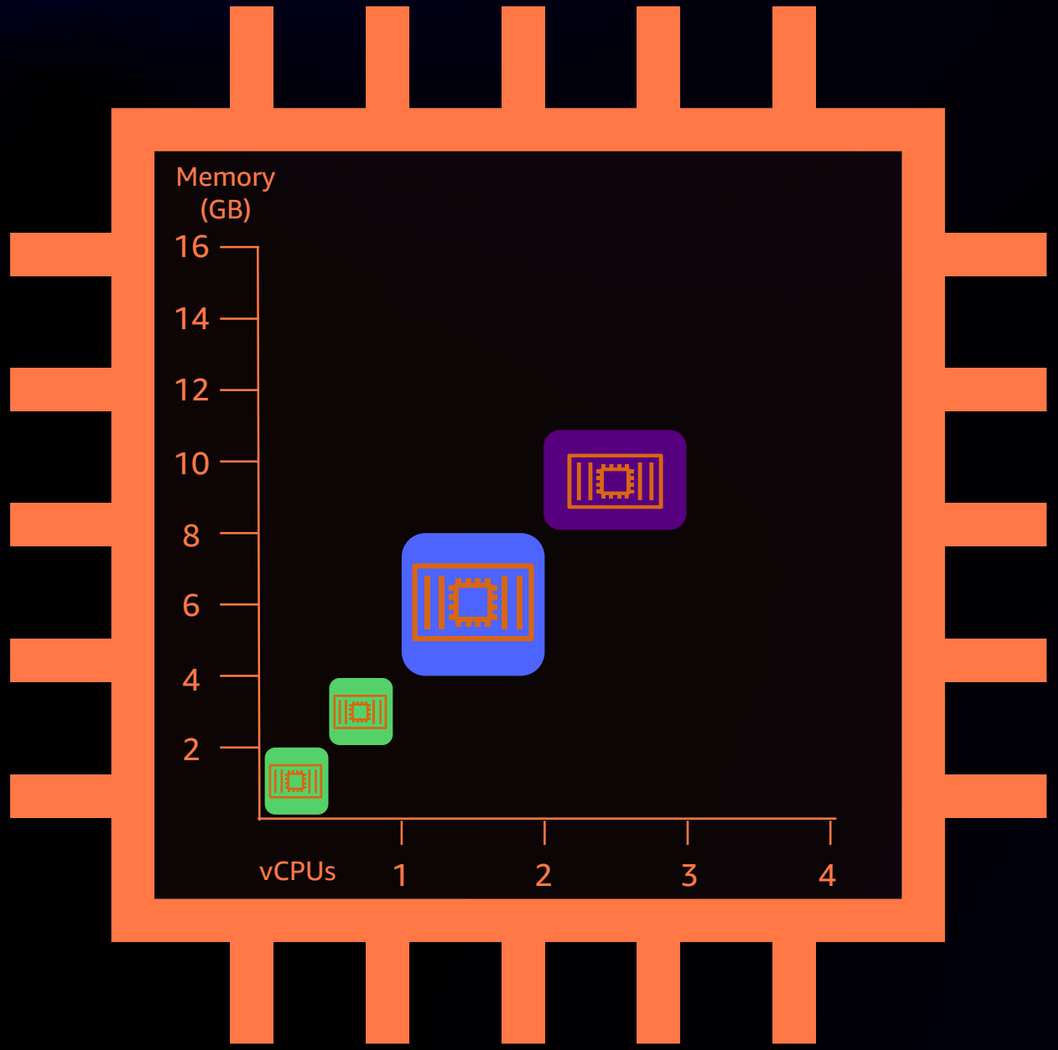


EKS & Spot workshop

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Scheduling pods and scaling resources

AN M5.XLARGE INSTANCE FROM KUBERNETES' POINT OF VIEW



Pod requests:

- web-app: 0.5 vCPU, 2 GB
- queue-processor: 1vCPU, 4 GB
- shopping-cart: 1vCPU, 3 GB

* Number of pods is also limited by the number of possible ENIs and secondary IP addresses for each instance type

** Amazon EKS or Amazon VPC Container Network Interface (CNI) plugin for Kubernetes; 1 Pod = 1 IP

aws

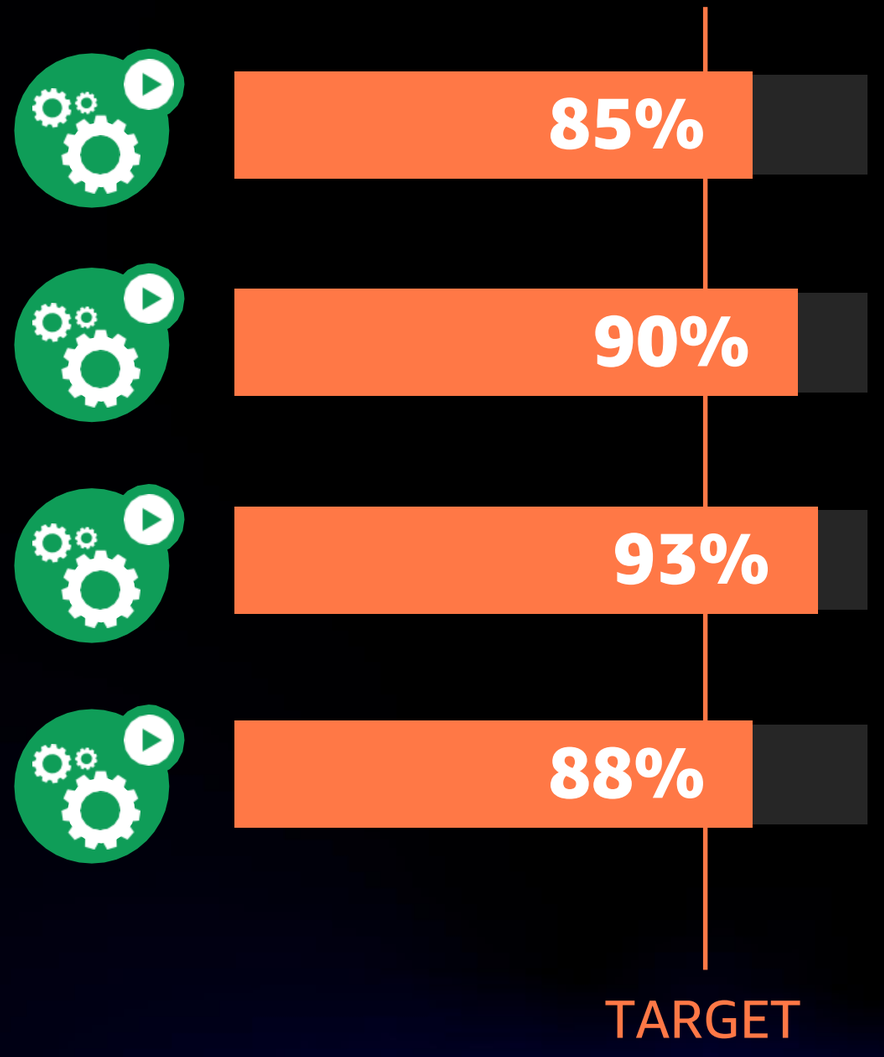
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

HPA intro

If pods are heavily loaded, then starting new pods may bring average load down

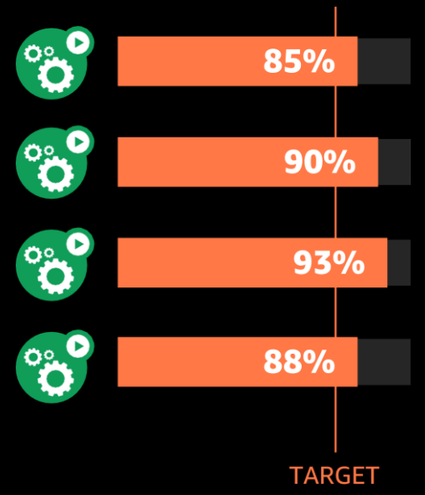
If pods are barely loaded, then stopping pods will free resources

Specify the target for the load
For example: Target = CPU utilization 70%



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cluster Autoscaler scale-up



Pending pods

HPA >>

ASG1-m5.xlarge

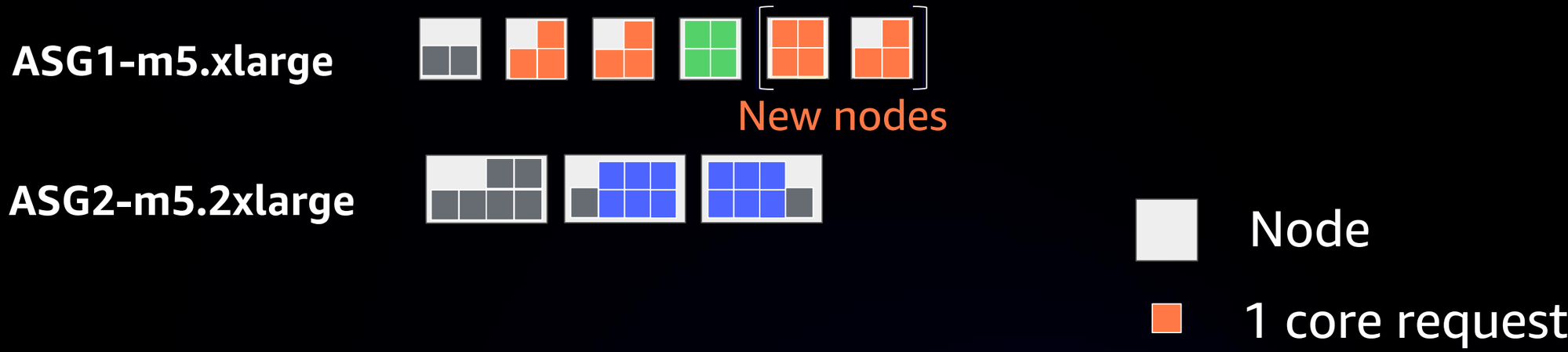
ASG2-m5.2xlarge

Node

1 core request

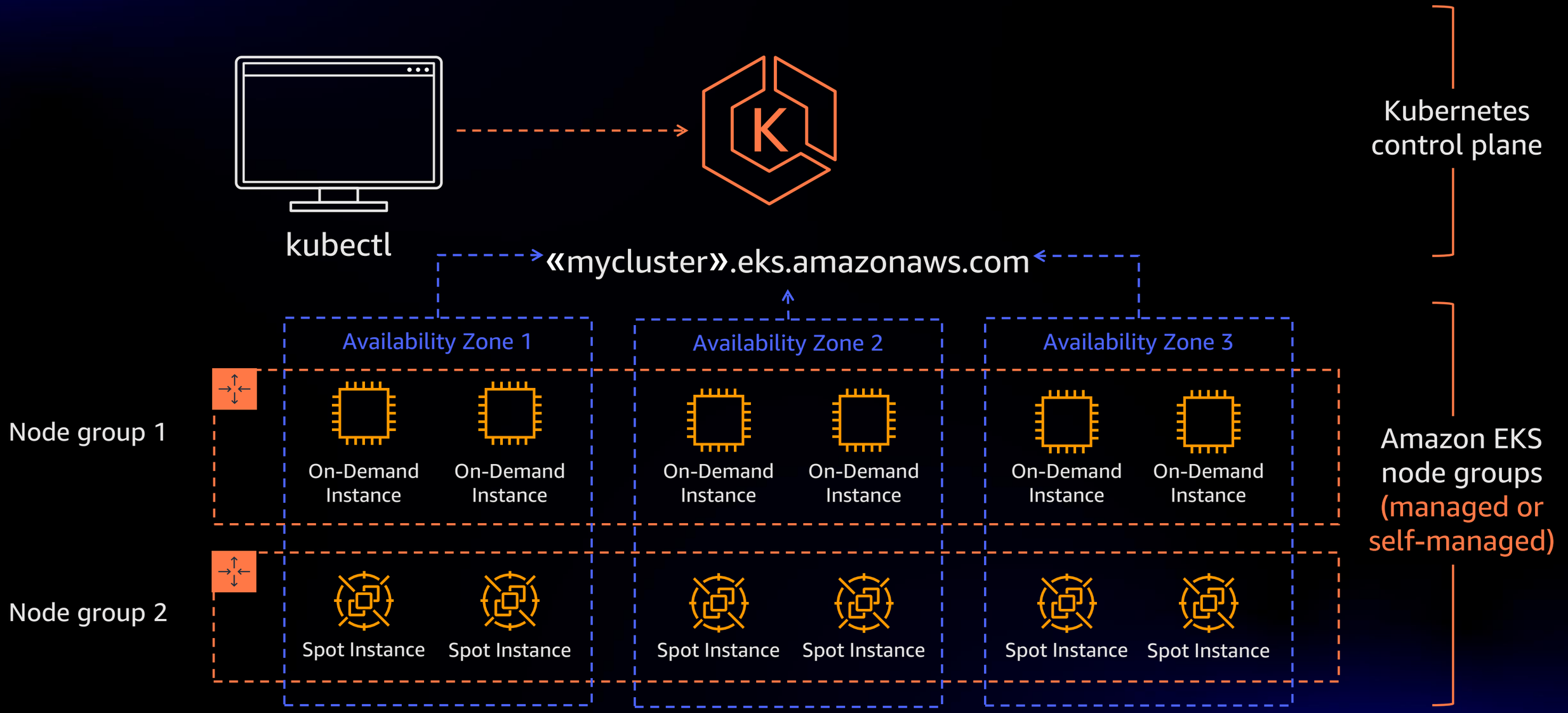
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cluster Autoscaler scale-up



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

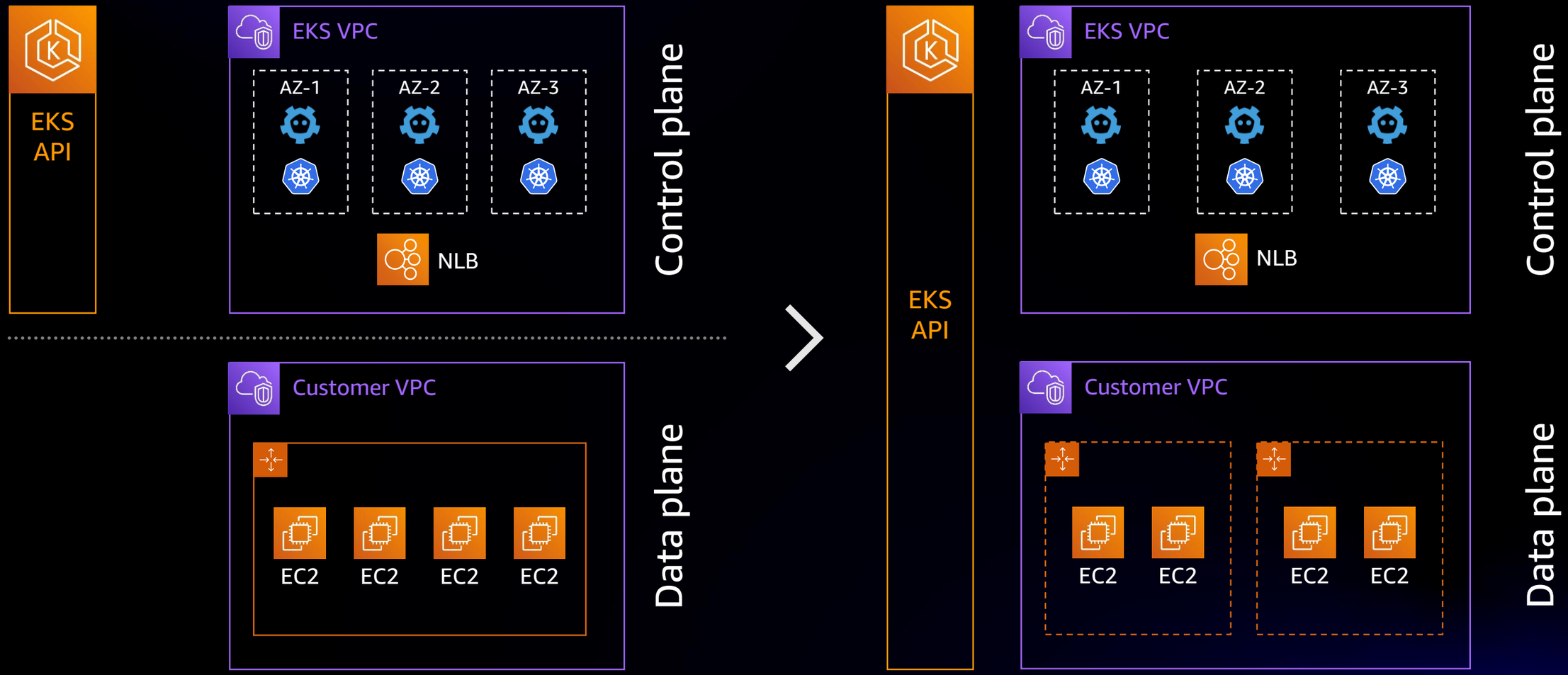
A high-level view of an Amazon EKS cluster



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS managed node groups

ON DEMAND MANAGED NODE GROUP (11/18/2019); SPOT MANAGED NODE GROUP (12/01/2020)



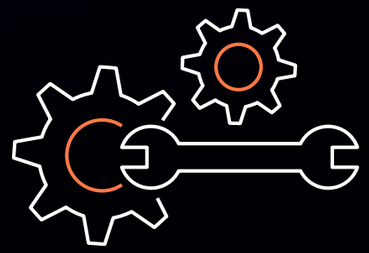
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS managed node groups with Spot Instances



Automatic node provisioning

Mixed instance types, Amazon EC2 Auto Scaling group provisioning with capacity-optimized allocation strategy, Automatic Kubernetes labels, and AWS tags (for example, Cluster Autoscaler)



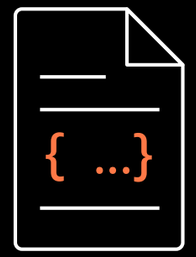
Managed node update and termination

Managed node draining during node termination and updates



Spot Instance lifecycle management

Managed Spot Instance draining integrated with Amazon EC2 Auto Scaling groups Capacity Rebalancing



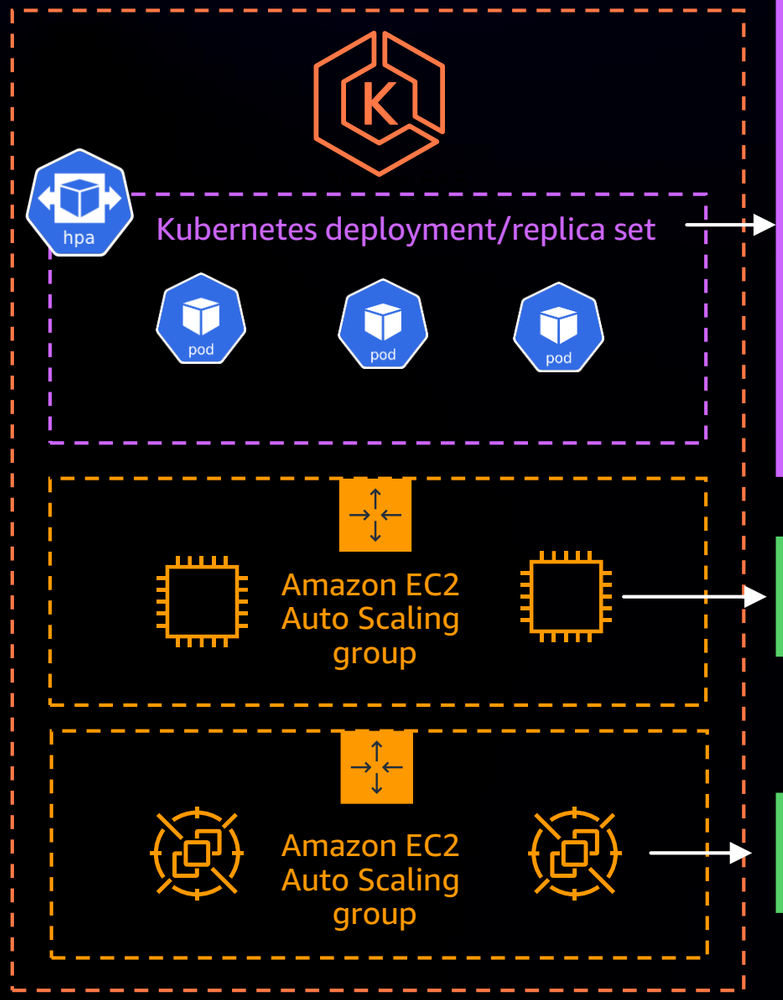
Launch template support

Custom configuration and custom AMIs for your Kubernetes nodes

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pod scheduling with nodeAffinity in managed node groups




```
apiVersion: v1
Kind: Deployment
…
spec:
  …
  affinity:
    nodeAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
        - weight: 1
          preference:
          matchExpressions:
            - key: eks.amazonaws.com/capacityType
              operator: In
              values:
                - SPOT
  …
```

```
eks.amazonaws.com/capacityType:
ON_DEMAND
```

```
eks.amazonaws.com/capacityType:
SPOT
```

Use Kubernetes labels for simple node selection constraints with *nodeSelector*

Or use richer expressions to control pod scheduling using nodeAffinty:

- Soft/preference: *preferredDuringSchedulingIgnoredDuringExecution*
- Hard/requirement: *requiredDuringSchedulingIgnoredDuringExecution*

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Influence pod scheduling: Taints and tolerations

```
- name: demo-4vcpu-16gb-spot-tainted
  minSize: 0
  maxSize: 15
  desiredCapacity: 1
  instancesDistribution:
    instanceTypes: ["m5.xlarge", "m5d.xlarge", "m4.xlarge","m5a.xlarge"]
    onDemandBaseCapacity: 0
    onDemandPercentageAboveBaseCapacity: 0
    spotAllocationStrategy: "capacity-optimized"
  labels:
    lifecycle: Ec2Spot
    intent: apps
  taints:
    lifecycle: "Ec2Spot:PreferNoSchedule"
  tags:
    k8s.io/cluster-autoscaler/node-template/label/lifecycle: Ec2Spot
    k8s.io/cluster-autoscaler/node-template/label/intent: apps
    k8s.io/cluster-autoscaler/node-template/taint/lifecycle: Ec2Spot
```

Taints are applied to a node, allow it to repel a set of pods

A node won't accept any pods that do not tolerate its taints

```
tolerations:
- key: "lifecycle"
  operator: "Equal"
  value: "Ec2Spot"
  effect: "PreferNoSchedule"
```

Tolerations are applied to pods and allow (but do not require) the pods to schedule onto nodes with matching taints

Taints and Tolerations: https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



EKS & Spot workshop

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Steve Cole

scolmzn@amazon.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.