Scaling nodes

Document Level Classification

200

Introduction

Your Substrate cluster uses the following strategies to manage *Node* capacity:

1. Static minimum capacity.
2. Dynamic and automatic scaling *up* of *Nodes* to place *Pods*.
3. Dynamic and automatic *rebalancing* and scaling *down* of *Nodes* to maintain optimal density.

Static minimum capacity

Substrate clusters use Amazon EKS managed node groups to maintain a static minimum number of *Nodes*.

By default:

- The *Nodes* are `c5d.4xlarge` EC2 instances launched using Amazon EKS optimized Amazon Linux AMI.
- A `dev` account is configured with a minimum of **3** *Nodes*.
- A `live` account is configured with a minimum of **5** *Nodes*.

The minimum capacity is required and used for critical infrastructure services running in the cluster, as well as initial *Pod* placements for application workloads. If you need to either increase the minimum capacity, or include Nodes with custom capabilities (Windows OS, GPU, ARM, etc.), reach out via slack in #cloud-ops-support-ext.

Dynamic scaling *up* of *Nodes*

Your Substrate cluster is configured to automatically provision new *Nodes* as and when needed – either during deployment events, or when your application needs to scale horizontally to meet demand.

For Substrate, scaling is implemented using Karpenter as it provides additional benefits over Kubernetes' built-in Cluster auto scaler. Your Substrate cluster has Karpenter enabled by default, and Provisioner deployed that is configured to choose from multiple EC2 instance types to satisfy availability. There is no upper bound configured for scaling - Karpenter will scale out and continue to provision EC2 instances until other types of availability (for example, EC2 instance type availability, IP address availability in the VPC) or quotas are reached (for example, vCPU quota for on-demand instances). If you want to configure an explicit upper bound for your Substrate cluster, reach out via slack in #cloud-ops-support-ext.

By default, Karpenter will launch EC2 instances using the Amazon EKS optimized Amazon Linux AMI. If your workloads need custom capabilities (Windows OS, GPU, ARM, etc.), you will need to deploy a custom Provisioner and Schedule your *Pods* on the corresponding *Nodes*.

Dynamic scaling *down* of *Nodes*

Your Substrate cluster is configured to automatically rebalance *Pods* and deprovision *Nodes* to optimize density and costs. This is implemented using Karpenter, and supplemented using Compactor. Compactor provides additional heuristics for rebalancing and deprovisioning *Nodes*.