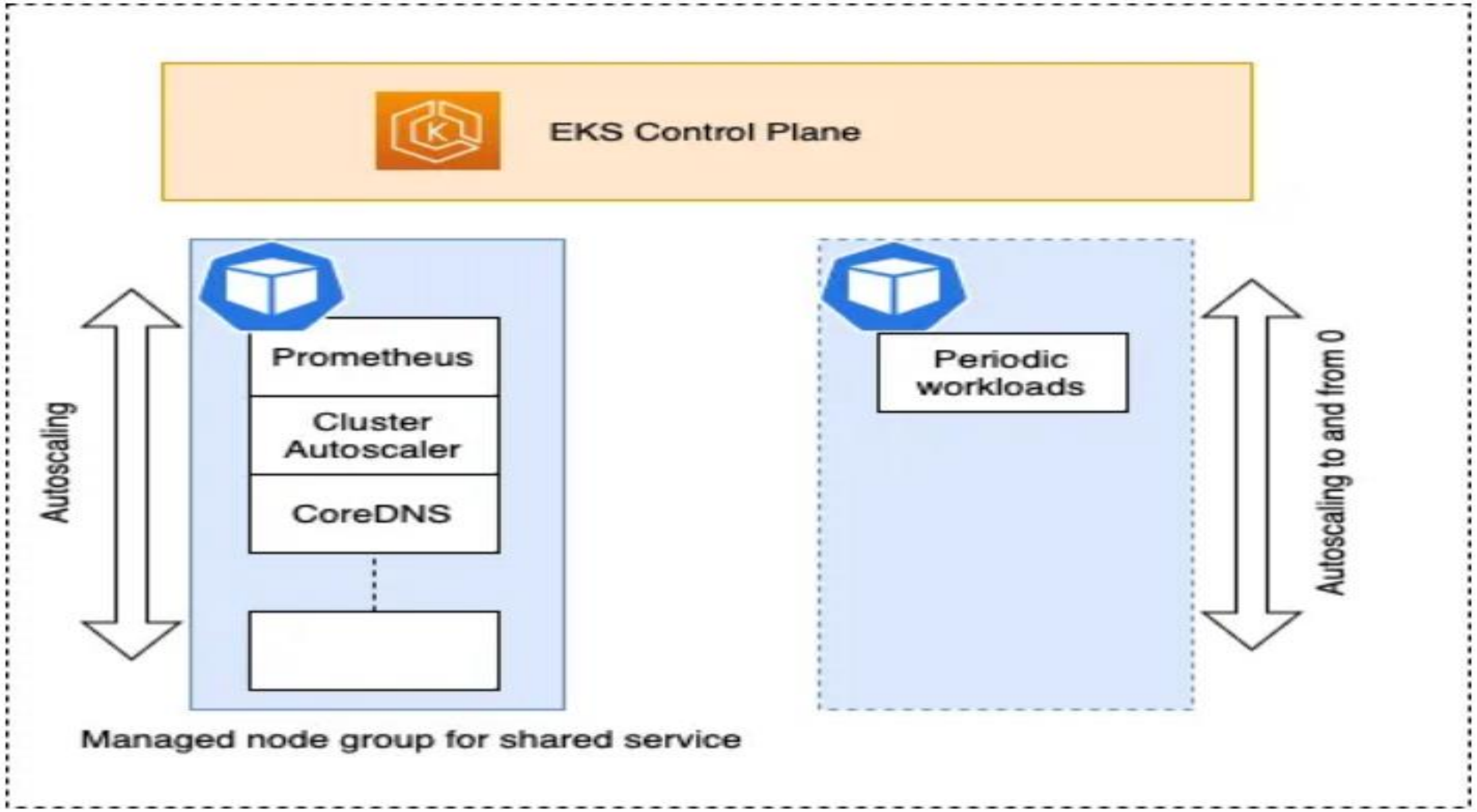
Scale down to zero

ow

Why?

- Outside of business hours, the supporting infrastructure (like nodes) sits idle. Customers use autoscaling to scale down node groups, but managed node groups required a minimum of 1 node in a node group previously. That’s one node too many, especially when you need beefier and costly nodes with GPUs.
- Scaling down to zero results in significant cost savings in such cases. In my opinion, you wouldn’t want to scale your entire cluster to zero. After all, you’d need some nodes to run Cluster Autoscaler and other shared services like Prometheus, AWS Load Balancer, CoreDNS, etc. You can use EKS on Fargate to run some of these services.

. .


EKS Control Plane
Autoscaling
Prometheus
Cluster Autoscaler
CoreDNS
Managed node group for shared service
Periodic workloads
Autoscaling to and from 0

How?

- To enable scaling to and from zero, the awesome EKS team contributed a feature to the upstream Cluster Autoscaler project.
- The new feature adds a manage node group cache that holds labels and taints associated with managed node groups.
- Cluster Autoscaler now uses the EKS DescribeNodegroup API to determine a node's label and taints when there are no nodes in the node group.
- This allows scaling to and from zero and doesn't require adding Auto Scaling group tags.

..

- Update the iam role, add workernode policy
- Create a new node group in existing cluster
- Minimum 0, desire 0
- Provide the label for the nodes
- Try to deploy workload using node selector