ENI Address Management Architecture

Goals

- Perform ENI adapter and IP allocation.

Non-Goals

Component Overview

- Cilium Agent with IPAM API
- CNI IPAM plugin
- CLI/API interface for troubleshooting
- Operator to interact with AWS metadata server
- New CiliumNode CRD

Components

Cilium Agent

Registers the node as CRD. Provides an IPAM API to the IPAM plugin based on the information in the CRD. Will keep the CRD up to date based on IPs used.

IPAM plugin

The IPAM plugin is called from the Cilium CNI but can also be used independently from other CNI plugins. The IPAM plugin interacts with the agent's IPAM API to allocate & release IPs on demand.

Operator

The operator monitors the node CRD and pre-allocates IP as needed and adds them to the CRD for use by the IPAM plugin via the agent.

CLI/API

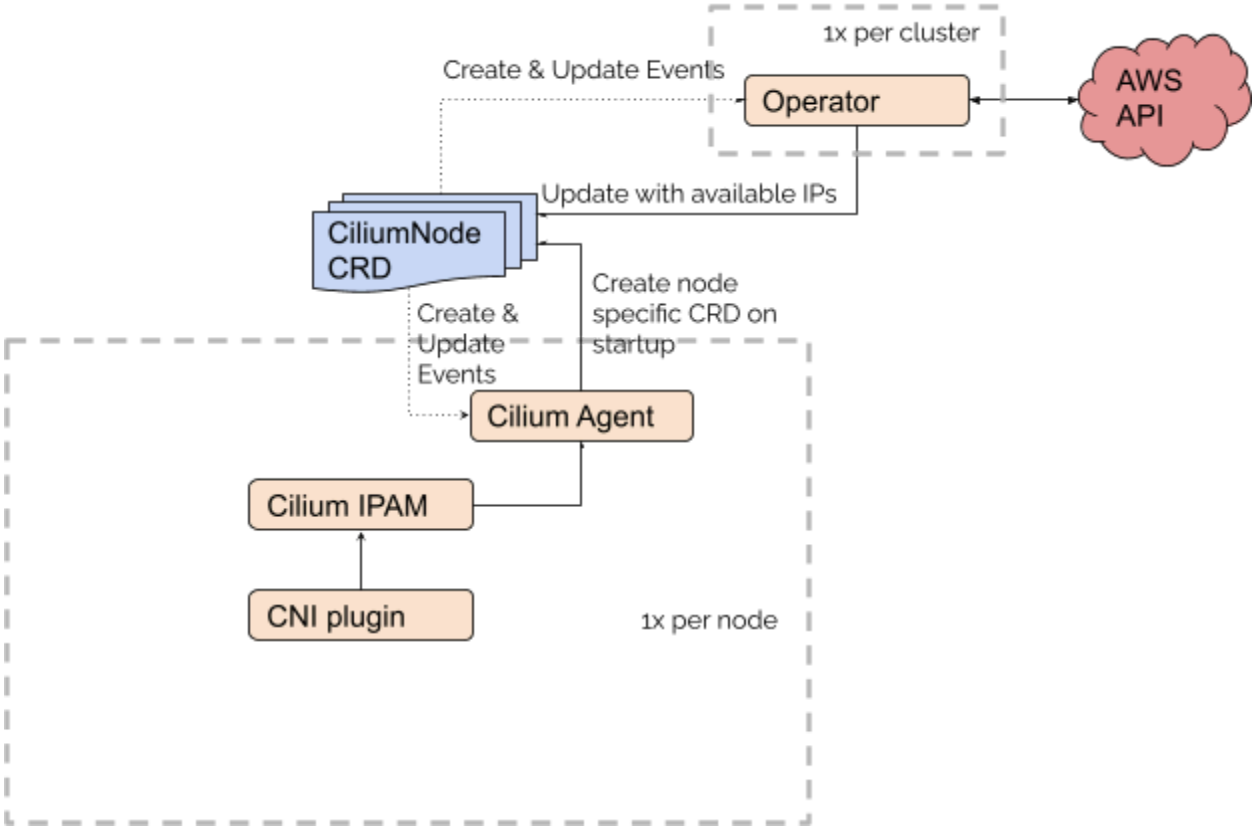
The CLI provides access to the IPAM API and allows to show available IPs as well as manually allocate and release IPs.

Configuration

- Number of ENIs to pre-allocate per node
  - Node resource annotation: `io.cilium.k8s.aws.eni.preallocate=8`
  - Field in CRD
- Starting interface index
  - Node resource annotation: `io.cilium.k8s.aws.eni.first-allocation-interface=1`
  - Field in CRD
- Security group
  - Node resource annotation: `io.cilum.k8s.aws.eni.security-group-tags=tag1,tag2,...`
  - Field in CRD
- Addresses per ENI
  - Node resource annotation: `io.cilum.k8s.aws.eni.addresses-per-eni=16`
  - Field in CRD
- Subnet Tags
  - Node resource annotation: `io.cilum.k8s.aws.eni.subnet-tags=tag1,tag2,...`
  - Field in CRD

Architecture

1. Each node runs a cilium-agent as DaemonSet. The agent will create a CRD CiliumNode with the node name. The CRD will contain the configuration to announce the desired state as per annotations but can also be modified.
2. The operator receives the create event and starts fulfilling the desired state by allocating ENI addresses. The ENI addresses are filled into the node resource and the resource is updated.
3. The agent receives the updated node resource via events and updates its own view of available IP addresses.
4. On allocation, the IP is removed from the node resource. The IP is used when that update is successful.
5. On release, the IP is added back to the node resource.



CRD

The CRD follows standard Kubernetes design principles. The spec describes the desired state. It is initially populated by the agent and can be modified by the user. The spec is never modified by the operator. The status describes the realized state. It is updated by the operator and the agent. The operator will fill the AvailableAddresses. The agent will remove from the AvailableAddresses and add to the UsedAddresses.

```
// +genclient
// +k8s:deepcopy-gen:interfaces=k8s.io/apimachinery/pkg/runtime.Object

// CiliumNode is a node managed by Cilium
type CiliumNode struct {
        // +k8s:openapi-gen=false
        metav1.TypeMeta `json:",inline"`
        // +k8s:openapi-gen=false
        metav1.ObjectMeta `json:"metadata"`

        Spec NodeSpec `json:"spec"`
```

```
        Status NodeStatus `json:"status"`
}

// NodeSpec is the configuration specification of the node
type NodeSpec struct {
        ENI ENISpec `json:"eni,omitempty"`
}

// ENISpec is the ENI specification of a node
type ENISpec struct {
        PreAllocate              int      `json:"preallocate,omitempty"`
        FirstAllocationInterface int      `json:"first-allocation-interface,omitempty"`
        SecurityGroups           []string `json:"security-groups,omitempty"`
        AddressesPerENI          int      `json:"addresses-per-eni,omitempty"`
        SubnetTags               map[string]string `json:"subnet-tags,omitempty"`
}

// NodeStatus is the status of a node
type NodeStatus struct {
        ENI ENIStatus `json:"eni,omitempty"`
}

// ENIStatus is the status of ENI addressing of the node
type ENIStatus struct {
        Used      map[string]string `json:"used,omitempty"`
        Available []string          `json:"available,omitempty"`
}

// +k8s:deepcopy-gen:interfaces=k8s.io/apimachinery/pkg/runtime.Object
//
// CiliumNodeList is a list of CiliumNode objects
type CiliumNodeList struct {
        metav1.TypeMeta `json:",inline"`
        metav1.ListMeta `json:"metadata"`

        // Items is a list of CiliumNode
        Items []CiliumNode `json:"items"`
}
```

Why a CRD?

The CRD ensures that IPAM is not dependent on the Cilium kvstore being available. It simplifies bootstrapping and resilience. It also provides great visibility into what is going on using standard tooling and enables resource protection with RBAC if desirable.