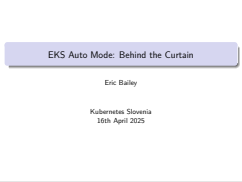
EKS Auto Mode: Behind the Curtain

Eric Bailey

Kubernetes Slovenia
16th April 2025



Briefly mention how it's presented as a turnkey or magic solution to the problem of managing Kubernetes, and that we're going to "look behind the curtain."

Table of Contents

Table of Contents
Introduction
kOps - Kubernetes Operations
EKS
EKS Auto Mode
So, which approach is best?

Introduction

- What is EKS Auto Mode?

Introduction
What is EKS Auto Mode?

1. It's basically vanilla EKS with some common add-ons, including highly opinionated or restrictive installations of the AWS VPC CNI and Karpenter.

Introduction

- What is EKS Auto Mode?
  "Amazon EKS Auto Mode fully automates Kubernetes cluster management for compute, storage, and networking on AWS with a single click."[1]

[1] https://aws.amazon.com/eks/auto-mode/

Introduction
What is EKS Auto Mode?
"Amazon EKS Auto Mode fully automates Kubernetes cluster management for compute, storage, and networking on AWS with a single click."[1]
[1] https://aws.amazon.com/eks/auto-mode/

1. It's basically vanilla EKS with some common add-ons, including highly opinionated or restrictive installations of the AWS VPC CNI and Karpenter.

Introduction

- What is EKS Auto Mode?
  "Amazon EKS Auto Mode fully automates Kubernetes cluster management for compute, storage, and networking on AWS with a single click."[1]
- What is EKS?

[1] https://aws.amazon.com/eks/auto-mode/

Introduction
- What is EKS Auto Mode?
  "Amazon EKS Auto Mode fully automates Kubernetes cluster management for compute, storage, and networking on AWS with a single click."[1]
- What is EKS?

[1] https://aws.amazon.com/eks/auto-mode/

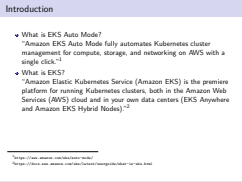
1. It's basically vanilla EKS with some common add-ons, including highly opinionated or restrictive installations of the AWS VPC CNI and Karpenter.
2. I'll refer to this as "vanilla EKS" to differentiate from EKS Auto Mode.
3. It's "Kubernetes as a Service", i.e., AWS-managed control plane and a small collection of official add-ons, as well as a broader add-on marketplace.
4. What's covered by AWS Support is sensibly minimal, and it's slightly prescriptive, e.g., feature flags, blessed CNIs, etc.

Introduction

- What is EKS Auto Mode?
  "Amazon EKS Auto Mode fully automates Kubernetes cluster management for compute, storage, and networking on AWS with a single click."[1]
- What is EKS?
  "Amazon Elastic Kubernetes Service (Amazon EKS) is the premiere platform for running Kubernetes clusters, both in the Amazon Web Services (AWS) cloud and in your own data centers (EKS Anywhere and Amazon EKS Hybrid Nodes)."[2]

[1] https://aws.amazon.com/eks/auto-mode/

[2] https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html



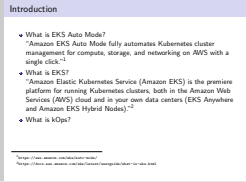
1. It's basically vanilla EKS with some common add-ons, including highly opinionated or restrictive installations of the AWS VPC CNI and Karpenter.
2. I'll refer to this as "vanilla EKS" to differentiate from EKS Auto Mode.
3. It's "Kubernetes as a Service", i.e., AWS-managed control plane and a small collection of official add-ons, as well as a broader add-on marketplace.
4. What's covered by AWS Support is sensibly minimal, and it's slightly prescriptive, e.g., feature flags, blessed CNIs, etc.

Introduction

- What is EKS Auto Mode?
  "Amazon EKS Auto Mode fully automates Kubernetes cluster management for compute, storage, and networking on AWS with a single click."[1]
- What is EKS?
  "Amazon Elastic Kubernetes Service (Amazon EKS) is the premiere platform for running Kubernetes clusters, both in the Amazon Web Services (AWS) cloud and in your own data centers (EKS Anywhere and Amazon EKS Hybrid Nodes)."[2]
- What is kOps?

[1] https://aws.amazon.com/eks/auto-mode/

[2] https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html


Introduction

- What is EKS Auto Mode?
  "Amazon EKS Auto Mode fully automates Kubernetes cluster management for compute, storage, and networking on AWS with a single click."[1]
- What is EKS?
  "Amazon Elastic Kubernetes Service (Amazon EKS) is the premiere platform for running Kubernetes clusters, both in the Amazon Web Services (AWS) cloud and in your own data centers (EKS Anywhere and Amazon EKS Hybrid Nodes)."[2]
- What is kOps?

[1] https://aws.amazon.com/eks/auto-mode/
[2] https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html

1. It's basically vanilla EKS with some common add-ons, including highly opinionated or restrictive installations of the AWS VPC CNI and Karpenter.
2. I'll refer to this as "vanilla EKS" to differentiate from EKS Auto Mode.
3. It's "Kubernetes as a Service", i.e., AWS-managed control plane and a small collection of official add-ons, as well as a broader add-on marketplace.
4. What's covered by AWS Support is sensibly minimal, and it's slightly prescriptive, e.g., feature flags, blessed CNIs, etc.

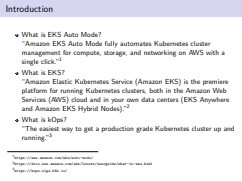
Introduction

- What is EKS Auto Mode?
  "Amazon EKS Auto Mode fully automates Kubernetes cluster management for compute, storage, and networking on AWS with a single click."[1]
- What is EKS?
  "Amazon Elastic Kubernetes Service (Amazon EKS) is the premiere platform for running Kubernetes clusters, both in the Amazon Web Services (AWS) cloud and in your own data centers (EKS Anywhere and Amazon EKS Hybrid Nodes)."[2]
- What is kOps?
  "The easiest way to get a production grade Kubernetes cluster up and running."[3]

[1] https://aws.amazon.com/eks/auto-mode/

[2] https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html

[3] https://kops.sigs.k8s.io/



1. It's basically vanilla EKS with some common add-ons, including highly opinionated or restrictive installations of the AWS VPC CNI and Karpenter.
2. I'll refer to this as "vanilla EKS" to differentiate from EKS Auto Mode.
3. It's "Kubernetes as a Service", i.e., AWS-managed control plane and a small collection of official add-ons, as well as a broader add-on marketplace.
4. What's covered by AWS Support is sensibly minimal, and it's slightly prescriptive, e.g., feature flags, blessed CNIs, etc.

kOps - Kubernetes Operations

kOps provides several managed add-ons, configurable through the cluster spec, as well as the means to specify custom static add-ons.[4]

[4] https://kops.sigs.k8s.io/addons/

kOps - Kubernetes Operations
kOps provides several managed add-ons, configurable through the cluster spec, as well as the means to specify custom static add-ons.[4]
[4] https://kops.sigs.k8s.io/addons/

kOps - Kubernetes Operations

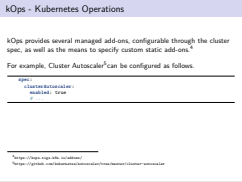
kOps provides several managed add-ons, configurable through the cluster spec, as well as the means to specify custom static add-ons.[4]

For example, Cluster Autoscaler[5]can be configured as follows.

```
spec:
  clusterAutoscaler:
    enabled: true
    # ...
```

[4] https://kops.sigs.k8s.io/addons/

[5] https://github.com/kubernetes/autoscaler/tree/master/cluster-autoscaler



kOps - Kubernetes Operations

kOps provides several managed add-ons, configurable through the cluster spec, as well as the means to specify custom static add-ons.[4]

For example, Cluster Autoscaler[5]can be configured as follows.

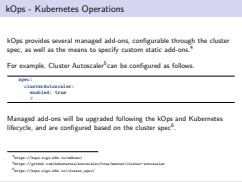
```
spec:
  clusterAutoscaler:
    enabled: true
    # ...
```

Managed add-ons will be upgraded following the kOps and Kubernetes lifecycle, and are configured based on the cluster spec[6].

[4] https://kops.sigs.k8s.io/addons/

[5] https://github.com/kubernetes/autoscaler/tree/master/cluster-autoscaler

[6] https://kops.sigs.k8s.io/cluster_spec/



kOps “Auto Mode”

Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.

[7] https://kops.sigs.k8s.io/networking/#cni

kOps “Auto Mode”
Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.

kOps “Auto Mode”

Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.

- CNI[7]
- CoreDNS
- kube-proxy

[7] https://kops.sigs.k8s.io/networking/#cni

kOps “Auto Mode”
Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.
CNI[7]
CoreDNS
kube-proxy
[7] https://kops.sigs.k8s.io/networking/#cni

1. kOps supports several CNIs, such as the AWS VPC CNI and Cilium.

kOps “Auto Mode”

Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.

- CNI[7]
- CoreDNS
- kube-proxy
- Node-local DNS cache

[7] https://kops.sigs.k8s.io/networking/#cni

kOps “Auto Mode”
Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.
CNI[7]
CoreDNS
kube-proxy
Node-local DNS cache
[7] https://kops.sigs.k8s.io/networking/#cni

1. kOps supports several CNIs, such as the AWS VPC CNI and Cilium.

kOps “Auto Mode”

Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.

- CNI[7]
- CoreDNS
- kube-proxy
- Node-local DNS cache
- Cluster Autoscaler (and NTH)

[7] https://kops.sigs.k8s.io/networking/#cni

kOps “Auto Mode”
Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.
CNI[7]
CoreDNS
kube-proxy
Node-local DNS cache
Cluster Autoscaler (and NTH)
[7] https://kops.sigs.k8s.io/networking/#cni

1. kOps supports several CNIs, such as the AWS VPC CNI and Cilium.

kOps “Auto Mode”

Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.

- CNI[7]
- CoreDNS
- kube-proxy
- Node-local DNS cache
- Cluster Autoscaler (and NTH) or Karpenter

[7] https://kops.sigs.k8s.io/networking/#cni

kOps “Auto Mode”
Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.
CNI[7]
CoreDNS
kube-proxy
Node-local DNS cache
Cluster Autoscaler (and NTH) or Karpenter
[7] https://kops.sigs.k8s.io/networking/#cni

1. kOps supports several CNIs, such as the AWS VPC CNI and Cilium.

kOps “Auto Mode”

Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.

- CNI[7]
- CoreDNS
- kube-proxy
- Node-local DNS cache
- Cluster Autoscaler (and NTH) or Karpenter
- AWS EBS CSI driver

[7] https://kops.sigs.k8s.io/networking/#cni

kOps “Auto Mode”
Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.
CNI[7]
CoreDNS
kube-proxy
Node-local DNS cache
Cluster Autoscaler (and NTH) or Karpenter
AWS EBS CSI driver
[7] https://kops.sigs.k8s.io/networking/#cni

1. kOps supports several CNIs, such as the AWS VPC CNI and Cilium.

kOps “Auto Mode”

Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.

- CNI[7]
- CoreDNS
- kube-proxy
- Node-local DNS cache
- Cluster Autoscaler (and NTH) or Karpenter
- AWS EBS CSI driver (and snapshot controller)

[7] https://kops.sigs.k8s.io/networking/#cni

kOps “Auto Mode”
Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.
- CNI[7]
- CoreDNS
- kube-proxy
- Node-local DNS cache
- Cluster Autoscaler (and NTH) or Karpenter
- AWS EBS CSI driver (and snapshot controller)

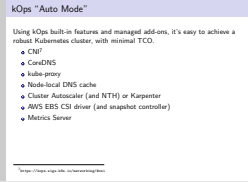
1. kOps supports several CNIs, such as the AWS VPC CNI and Cilium.

kOps “Auto Mode”

Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.

- CNI[7]
- CoreDNS
- kube-proxy
- Node-local DNS cache
- Cluster Autoscaler (and NTH) or Karpenter
- AWS EBS CSI driver (and snapshot controller)
- Metrics Server

[7] https://kops.sigs.k8s.io/networking/#cni



1. kOps supports several CNIs, such as the AWS VPC CNI and Cilium.

kOps “Auto Mode”

Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.

- CNI[7]
- CoreDNS
- kube-proxy
- Node-local DNS cache
- Cluster Autoscaler (and NTH) or Karpenter
- AWS EBS CSI driver (and snapshot controller)
- Metrics Server
- Pod Identity Webhook

[7] https://kops.sigs.k8s.io/networking/#cni

kOps “Auto Mode”
Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.
CNI[7]
CoreDNS
kube-proxy
Node-local DNS cache
Cluster Autoscaler (and NTH) or Karpenter
AWS EBS CSI driver (and snapshot controller)
Metrics Server
Pod Identity Webhook
[7] https://kops.sigs.k8s.io/networking/#cni

1. kOps supports several CNIs, such as the AWS VPC CNI and Cilium.

kOps “Auto Mode”

Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.

- CNI[7]
- CoreDNS
- kube-proxy
- Node-local DNS cache
- Cluster Autoscaler (and NTH) or Karpenter
- AWS EBS CSI driver (and snapshot controller)
- Metrics Server
- Pod Identity Webhook
- AWS Load Balancer Controller

[7] https://kops.sigs.k8s.io/networking/#cni

kOps “Auto Mode”
Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.
CNI[7]
CoreDNS
kube-proxy
Node-local DNS cache
Cluster Autoscaler (and NTH) or Karpenter
AWS EBS CSI driver (and snapshot controller)
Metrics Server
Pod Identity Webhook
AWS Load Balancer Controller
[7] https://kops.sigs.k8s.io/networking/#cni

1. kOps supports several CNIs, such as the AWS VPC CNI and Cilium.

kOps “Auto Mode”

Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.

- CNI[7]
- CoreDNS
- kube-proxy
- Node-local DNS cache
- Cluster Autoscaler (and NTH) or Karpenter
- AWS EBS CSI driver (and snapshot controller)
- Metrics Server
- Pod Identity Webhook
- AWS Load Balancer Controller
- cert-manager

[7] https://kops.sigs.k8s.io/networking/#cni

kOps “Auto Mode”
Using kOps built-in features and managed add-ons, it’s easy to achieve a robust Kubernetes cluster, with minimal TCO.
CNI[7]
CoreDNS
kube-proxy
Node-local DNS cache
Cluster Autoscaler (and NTH) or Karpenter
AWS EBS CSI driver (and snapshot controller)
Metrics Server
Pod Identity Webhook
AWS Load Balancer Controller
cert-manager
[7] https://kops.sigs.k8s.io/networking/#cni

1. kOps supports several CNIs, such as the AWS VPC CNI and Cilium.
2. Mention hen and egg for some kOps components such as Metrics Server.

EKS “Manual Mode”

For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.

[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.

[9] https://aws-ia.github.io/terraform-aws-eks-blueprints/

EKS “Manual Mode”
For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.
[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.
[9] https://aws-ia.github.io/terraform-aws-eks-blueprints/

1. You can also write your own add-ons.
2. There be dragons. Managing Helm charts via Terraform is... interesting.

EKS “Manual Mode”

For EKS there exist several “managed”[8] add-ons, tens of community*and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.

- AWS VPC CNI
- CoreDNS
- kube-proxy

[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.

[9] https://aws-ia.github.io/terraform-aws-eks-blueprints/

EKS “Manual Mode”
For EKS there exist several “managed”[8] add-ons, tens of community*and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.
- AWS VPC CNI
- CoreDNS
- kube-proxy
[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.
[9] https://aws-ia.github.io/terraform-aws-eks-blueprints/

1. You can also write your own add-ons.
2. There be dragons. Managing Helm charts via Terraform is... interesting.
3. Mention how node-local DNS cache is just a special case.
4. Not possible to configure two instances of CoreDNS add-on.

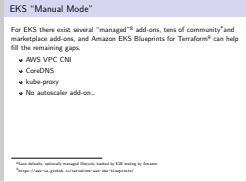
EKS “Manual Mode”

For EKS there exist several “managed”[8] add-ons, tens of community*and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.

- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on...

[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.

[9] `https://aws-ia.github.io/terraform-aws-eks-blueprints/`



1. You can also write your own add-ons.
2. There be dragons. Managing Helm charts via Terraform is... interesting.
3. Mention how node-local DNS cache is just a special case.
4. Not possible to configure two instances of CoreDNS add-on.

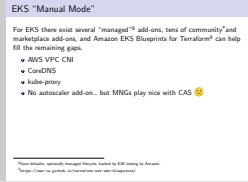
EKS “Manual Mode”

For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.

- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on... but MNGs play nice with CAS 😕

[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.

[9] `https://aws-ia.github.io/terraform-aws-eks-blueprints/`



1. You can also write your own add-ons.
2. There be dragons. Managing Helm charts via Terraform is... interesting.
3. Mention how node-local DNS cache is just a special case.
4. Not possible to configure two instances of CoreDNS add-on.

EKS “Manual Mode”

For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.

- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on... but MNGs play nice with CAS 😕
- EBS CSI driver

[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.

[9] `https://aws-ia.github.io/terraform-aws-eks-blueprints/`

EKS “Manual Mode”
For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.
- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on... but MNGs play nice with CAS 😕
- EBS CSI driver
[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.
[9] https://aws-ia.github.io/terraform-aws-eks-blueprints/

1. You can also write your own add-ons.
2. There be dragons. Managing Helm charts via Terraform is... interesting.
3. Mention how node-local DNS cache is just a special case.
4. Not possible to configure two instances of CoreDNS add-on.

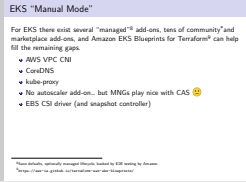
EKS “Manual Mode”

For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.

- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on... but MNGs play nice with CAS 😕
- EBS CSI driver (and snapshot controller)

[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.

[9] `https://aws-ia.github.io/terraform-aws-eks-blueprints/`



1. You can also write your own add-ons.
2. There be dragons. Managing Helm charts via Terraform is... interesting.
3. Mention how node-local DNS cache is just a special case.
4. Not possible to configure two instances of CoreDNS add-on.

EKS “Manual Mode”

For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.

- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on... but MNGs play nice with CAS 😕
- EBS CSI driver (and snapshot controller)
- Metrics Server*

[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.

[9] `https://aws-ia.github.io/terraform-aws-eks-blueprints/`

EKS “Manual Mode”
For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.
- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on... but MNGs play nice with CAS 😕
- EBS CSI driver (and snapshot controller)
- Metrics Server*
[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.
[9] https://aws-ia.github.io/terraform-aws-eks-blueprints/

1. You can also write your own add-ons.
2. There be dragons. Managing Helm charts via Terraform is... interesting.
3. Mention how node-local DNS cache is just a special case.
4. Not possible to configure two instances of CoreDNS add-on.
5. Metrics Server and cert-manager are community add-ons.

EKS “Manual Mode”

For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.

- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on... but MNGs play nice with CAS 😕
- EBS CSI driver (and snapshot controller)
- Metrics Server*
- Pod Identity Webhook

[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.

[9] https://aws-ia.github.io/terraform-aws-eks-blueprints/

EKS “Manual Mode”
For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.
- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on... but MNGs play nice with CAS 😕
- EBS CSI driver (and snapshot controller)
- Metrics Server*
- Pod Identity Webhook
[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.
[9] https://aws-ia.github.io/terraform-aws-eks-blueprints/

1. You can also write your own add-ons.
2. There be dragons. Managing Helm charts via Terraform is... interesting.
3. Mention how node-local DNS cache is just a special case.
4. Not possible to configure two instances of CoreDNS add-on.
5. Metrics Server and cert-manager are community add-ons.
6. Pod identity webhook “for free”.

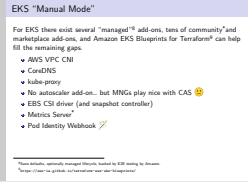
EKS “Manual Mode”

For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.

- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on... but MNGs play nice with CAS 😕
- EBS CSI driver (and snapshot controller)
- Metrics Server*
- Pod Identity Webhook 🪄

[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.

[9] https://aws-ia.github.io/terraform-aws-eks-blueprints/



1. You can also write your own add-ons.
2. There be dragons. Managing Helm charts via Terraform is... interesting.
3. Mention how node-local DNS cache is just a special case.
4. Not possible to configure two instances of CoreDNS add-on.
5. Metrics Server and cert-manager are community add-ons.
6. Pod identity webhook “for free”.

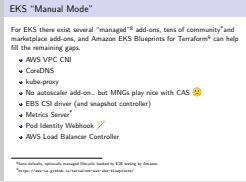
EKS “Manual Mode”

For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.

- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on… but MNGs play nice with CAS 😕
- EBS CSI driver (and snapshot controller)
- Metrics Server*
- Pod Identity Webhook 🪄
- AWS Load Balancer Controller

[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.

[9] `https://aws-ia.github.io/terraform-aws-eks-blueprints/`



1. You can also write your own add-ons.
2. There be dragons. Managing Helm charts via Terraform is… interesting.
3. Mention how node-local DNS cache is just a special case.
4. Not possible to configure two instances of CoreDNS add-on.
5. Metrics Server and cert-manager are community add-ons.
6. Pod identity webhook “for free”.
7. Need to DIY LBC.

EKS “Manual Mode”

For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.

- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on... but MNGs play nice with CAS 😕
- EBS CSI driver (and snapshot controller)
- Metrics Server*
- Pod Identity Webhook 🪄
- AWS Load Balancer Controller 😖

[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.

[9] `https://aws-ia.github.io/terraform-aws-eks-blueprints/`

EKS “Manual Mode”
For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.
- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on... but MNGs play nice with CAS 😕
- EBS CSI driver (and snapshot controller)
- Metrics Server*
- Pod Identity Webhook 🪄
- AWS Load Balancer Controller 😖
[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.
[9] https://aws-ia.github.io/terraform-aws-eks-blueprints/

1. You can also write your own add-ons.
2. There be dragons. Managing Helm charts via Terraform is... interesting.
3. Mention how node-local DNS cache is just a special case.
4. Not possible to configure two instances of CoreDNS add-on.
5. Metrics Server and cert-manager are community add-ons.
6. Pod identity webhook “for free”.
7. Need to DIY LBC.

EKS “Manual Mode”

For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.

- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on... but MNGs play nice with CAS 😕
- EBS CSI driver (and snapshot controller)
- Metrics Server*
- Pod Identity Webhook 🪄
- AWS Load Balancer Controller 😖
- cert-manager*

[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.

[9] https://aws-ia.github.io/terraform-aws-eks-blueprints/

EKS “Manual Mode”
For EKS there exist several “managed”[8] add-ons, tens of community* and marketplace add-ons, and Amazon EKS Blueprints for Terraform[9] can help fill the remaining gaps.
- AWS VPC CNI
- CoreDNS
- kube-proxy
- No autoscaler add-on... but MNGs play nice with CAS 😕
- EBS CSI driver (and snapshot controller)
- Metrics Server*
- Pod Identity Webhook 🪄
- AWS Load Balancer Controller 😖
- cert-manager*
[8] Sane defaults, optionally managed lifecycle, backed by E2E testing by Amazon.
[9] https://aws-ia.github.io/terraform-aws-eks-blueprints/

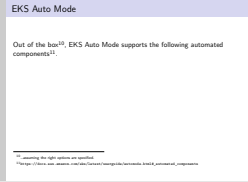
1. You can also write your own add-ons.
2. There be dragons. Managing Helm charts via Terraform is... interesting.
3. Mention how node-local DNS cache is just a special case.
4. Not possible to configure two instances of CoreDNS add-on.
5. Metrics Server and cert-manager are community add-ons.
6. Pod identity webhook “for free”.
7. Need to DIY LBC.

EKS Auto Mode

Out of the box[10], EKS Auto Mode supports the following automated components[11].

[10] …assuming the right options are specified.

[11] https://docs.aws.amazon.com/eks/latest/userguide/automode.html#_automated_components



EKS Auto Mode

Out of the box[10], EKS Auto Mode supports the following automated components[11].

- AWS VPC CNI

[10] …assuming the right options are specified.

[11] https://docs.aws.amazon.com/eks/latest/userguide/automode.html#_automated_components

EKS Auto Mode
Out of the box[10], EKS Auto Mode supports the following automated components[11].
AWS VPC CNI

EKS Auto Mode

Out of the box[10], EKS Auto Mode supports the following automated components[11].

- AWS VPC CNI (with caveats)

[10] …assuming the right options are specified.

[11] https://docs.aws.amazon.com/eks/latest/userguide/automode.html#_automated_components

EKS Auto Mode
Out of the box[10], EKS Auto Mode supports the following automated components[11].
AWS VPC CNI (with caveats)

1. EKS Auto Mode does not support warm IP, warm prefix, or warm ENI configurations.
2. Instead of ds/aws-node it's implemented as a systemd unit that can't be reconfigured.
3. tl;dr custom networking is not supported.

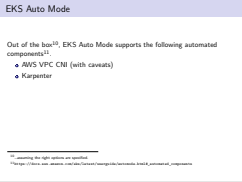
EKS Auto Mode

Out of the box[10], EKS Auto Mode supports the following automated components[11].

- AWS VPC CNI (with caveats)
- Karpenter

[10] …assuming the right options are specified.

[11] `https://docs.aws.amazon.com/eks/latest/userguide/automode.html#_automated_components`



1. EKS Auto Mode does not support warm IP, warm prefix, or warm ENI configurations.
2. Instead of ds/aws-node it's implemented as a systemd unit that can't be reconfigured.
3. tl;dr custom networking is not supported.
4. Auto Mode comes with NodePools general-purpose and platform (tainted `CriticalAddonsOnly:NoSchedule`).
5. Default NodePools dont' support SSH and such.
6. You can add your own NodePools, but probably don't wanna have different CNI configs.

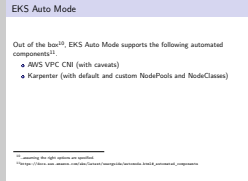
EKS Auto Mode

Out of the box[10], EKS Auto Mode supports the following automated components[11].

- AWS VPC CNI (with caveats)
- Karpenter (with default and custom NodePools and NodeClasses)

[10] …assuming the right options are specified.

[11] `https://docs.aws.amazon.com/eks/latest/userguide/automode.html#_automated_components`



1. EKS Auto Mode does not support warm IP, warm prefix, or warm ENI configurations.
2. Instead of ds/aws-node it's implemented as a systemd unit that can't be reconfigured.
3. tl;dr custom networking is not supported.
4. Auto Mode comes with NodePools general-purpose and platform (tainted `CriticalAddonsOnly:NoSchedule`).
5. Default NodePools dont' support SSH and such.
6. You can add your own NodePools, but probably don't wanna have different CNI configs.
7. You can configure this, e.g., to use a node-local DNS cache, but you'll probably have to install it off the shelf first, and then reconfigure, then install node-local DNS, then reconfigure…

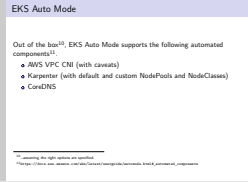
EKS Auto Mode

Out of the box[10], EKS Auto Mode supports the following automated components[11].

- AWS VPC CNI (with caveats)
- Karpenter (with default and custom NodePools and NodeClasses)
- CoreDNS

[10] …assuming the right options are specified.

[11] `https://docs.aws.amazon.com/eks/latest/userguide/automode.html#_automated_components`


EKS Auto Mode

Out of the box[10], EKS Auto Mode supports the following automated components[11].

- AWS VPC CNI (with caveats)
- Karpenter (with default and custom NodePools and NodeClasses)
- CoreDNS

[10] …assuming the right options are specified.
[11] https://docs.aws.amazon.com/eks/latest/userguide/automode.html#_automated_components

1. EKS Auto Mode does not support warm IP, warm prefix, or warm ENI configurations.
2. Instead of ds/aws-node it's implemented as a systemd unit that can't be reconfigured.
3. tl;dr custom networking is not supported.
4. Auto Mode comes with NodePools general-purpose and platform (tainted `CriticalAddonsOnly:NoSchedule`).
5. Default NodePools dont' support SSH and such.
6. You can add your own NodePools, but probably don't wanna have different CNI configs.
7. You can configure this, e.g., to use a node-local DNS cache, but you'll probably have to install it off the shelf first, and then reconfigure, then install node-local DNS, then reconfigure…

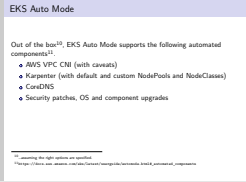
EKS Auto Mode

Out of the box[10], EKS Auto Mode supports the following automated components[11].

- AWS VPC CNI (with caveats)
- Karpenter (with default and custom NodePools and NodeClasses)
- CoreDNS
- Security patches, OS and component upgrades

[10] …assuming the right options are specified.

[11] `https://docs.aws.amazon.com/eks/latest/userguide/automode.html#_automated_components`



1. EKS Auto Mode does not support warm IP, warm prefix, or warm ENI configurations.
2. Instead of ds/aws-node it's implemented as a systemd unit that can't be reconfigured.
3. tl;dr custom networking is not supported.
4. Auto Mode comes with NodePools general-purpose and platform (tainted `CriticalAddonsOnly:NoSchedule`).
5. Default NodePools dont' support SSH and such.
6. You can add your own NodePools, but probably don't wanna have different CNI configs.
7. You can configure this, e.g., to use a node-local DNS cache, but you'll probably have to install it off the shelf first, and then reconfigure, then install node-local DNS, then reconfigure…
8. "EKS Auto Mode enforces a 21-day maximum node lifetime to ensure up-to-date software and APIs."

EKS Auto Mode

Out of the box[10], EKS Auto Mode supports the following automated components[11].

- AWS VPC CNI (with caveats)
- Karpenter (with default and custom NodePools and NodeClasses)
- CoreDNS
- Security patches, OS and component upgrades
- Pod Identity Webhook

[10] ...assuming the right options are specified.

[11] `https://docs.aws.amazon.com/eks/latest/userguide/automode.html#_automated_components`

EKS Auto Mode

Out of the box[10], EKS Auto Mode supports the following automated components[11].

- AWS VPC CNI (with caveats)
- Karpenter (with default and custom NodePools and NodeClasses)
- CoreDNS
- Security patches, OS and component upgrades
- Pod Identity Webhook

[10] ...assuming the right options are specified.
[11] https://docs.aws.amazon.com/eks/latest/userguide/automode.html#_automated_components

1. EKS Auto Mode does not support warm IP, warm prefix, or warm ENI configurations.
2. Instead of ds/aws-node it's implemented as a systemd unit that can't be reconfigured.
3. tl;dr custom networking is not supported.
4. Auto Mode comes with NodePools general-purpose and platform (tainted `CriticalAddonsOnly:NoSchedule`).
5. Default NodePools dont' support SSH and such.
6. You can add your own NodePools, but probably don't wanna have different CNI configs.
7. You can configure this, e.g., to use a node-local DNS cache, but you'll probably have to install it off the shelf first, and then reconfigure, then install node-local DNS, then reconfigure...
8. "EKS Auto Mode enforces a 21-day maximum node lifetime to ensure up-to-date software and APIs."

EKS Auto Mode

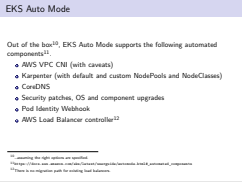
Out of the box[10], EKS Auto Mode supports the following automated components[11].

- AWS VPC CNI (with caveats)
- Karpenter (with default and custom NodePools and NodeClasses)
- CoreDNS
- Security patches, OS and component upgrades
- Pod Identity Webhook
- AWS Load Balancer controller[12]

[10] …assuming the right options are specified.

[11] `https://docs.aws.amazon.com/eks/latest/userguide/automode.html#_automated_components`

[12] There is no migration path for existing load balancers.



1. EKS Auto Mode does not support warm IP, warm prefix, or warm ENI configurations.
2. Instead of ds/aws-node it's implemented as a systemd unit that can't be reconfigured.
3. tl;dr custom networking is not supported.
4. Auto Mode comes with NodePools general-purpose and platform (tainted `CriticalAddonsOnly:NoSchedule`).
5. Default NodePools dont' support SSH and such.
6. You can add your own NodePools, but probably don't wanna have different CNI configs.
7. You can configure this, e.g., to use a node-local DNS cache, but you'll probably have to install it off the shelf first, and then reconfigure, then install node-local DNS, then reconfigure…
8. "EKS Auto Mode enforces a 21-day maximum node lifetime to ensure up-to-date software and APIs."

So, which approach is best?

So, which approach is best?

It depends.™

So, which approach is best?
It depends.™

Thank you! 

Thank you! 🙇

But actually...

Here are some criteria to rule out EKS Auto Mode as a viable option.

- Need custom networking?

But actually...
Here are some criteria to rule out EKS Auto Mode as a viable option.
Need custom networking?

But actually...

Here are some criteria to rule out EKS Auto Mode as a viable option.

- Need custom networking?
  - 😇 "The IP Addresses of Pods and Nodes must be from the same CIDR Block."[13]

[13] The documentation is misleading and contradictory, YMMV.

But actually...
Here are some criteria to rule out EKS Auto Mode as a viable option.
Need custom networking?
😇 "The IP Addresses of Pods and Nodes must be from the same CIDR Block."[13]
[13] The documentation is misleading and contradictory, YMMV.

But actually...

Here are some criteria to rule out EKS Auto Mode as a viable option.

- Need custom networking?
  - 😕 "The IP Addresses of Pods and Nodes must be from the same CIDR Block."[13]
  - 💸 EKS Auto Mode does not support warm IP, warm prefix, or warm ENI configurations.

[13] The documentation is misleading and contradictory, YMMV.

But actually...
Here are some criteria to rule out EKS Auto Mode as a viable option.
Need custom networking?
😕 "The IP Addresses of Pods and Nodes must be from the same CIDR Block."[13]
💸 EKS Auto Mode does not support warm IP, warm prefix, or warm ENI configurations.
[13] The documentation is misleading and contradictory, YMMV.

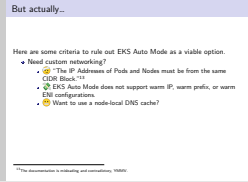
1. Mention our IPAM cost drivers deep dive.

But actually...

Here are some criteria to rule out EKS Auto Mode as a viable option.

- Need custom networking?
  - 🤕 "The IP Addresses of Pods and Nodes must be from the same CIDR Block."[13]
  - 💸 EKS Auto Mode does not support warm IP, warm prefix, or warm ENI configurations.
  - 😬 Want to use a node-local DNS cache?

[13] The documentation is misleading and contradictory, YMMV.



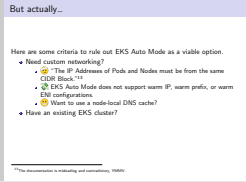
1. Mention our IPAM cost drivers deep dive.
2. Custom configuration is supported, but likely would need to be added after initial cluster provisioning.

But actually...

Here are some criteria to rule out EKS Auto Mode as a viable option.

- Need custom networking?
  - 🤕 "The IP Addresses of Pods and Nodes must be from the same CIDR Block."[13]
  - 💸 EKS Auto Mode does not support warm IP, warm prefix, or warm ENI configurations.
  - 😬 Want to use a node-local DNS cache?
- Have an existing EKS cluster?

[13] The documentation is misleading and contradictory, YMMV.



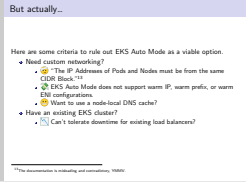
1. Mention our IPAM cost drivers deep dive.
2. Custom configuration is supported, but likely would need to be added after initial cluster provisioning.

But actually...

Here are some criteria to rule out EKS Auto Mode as a viable option.

- Need custom networking?
  - 🤕 "The IP Addresses of Pods and Nodes must be from the same CIDR Block."[13]
  - 💸 EKS Auto Mode does not support warm IP, warm prefix, or warm ENI configurations.
  - 😬 Want to use a node-local DNS cache?
- Have an existing EKS cluster?
  - 📉 Can't tolerate downtime for existing load balancers?

[13] The documentation is misleading and contradictory, YMMV.



1. Mention our IPAM cost drivers deep dive.
2. Custom configuration is supported, but likely would need to be added after initial cluster provisioning.

Feature Comparsion

Comparing the high-level features[14]

	EKS Auto Mode	EKS	kOps
Nodes	⚠️	✅	✅
Autoscaling	✅	⚠️	✅
Upgrades	✅	⚠️	⚠️
Load balancing	⚠️	✅	✅
Storage	✅	✅	✅
Networking	❌	⚠️	✅
IAM	✅	✅	✅
AMI/OS	❌	✅	✅

[14] https://docs.aws.amazon.com/eks/latest/userguide/automode.html#_features

Feature Comparsion
Comparing the high-level features[14]
EKS Auto Mode EKS kOps
Nodes
Autoscaling
Upgrades
Load balancing
Storage
Networking
IAM
AMI/OS

1. EKS Auto Mode uses their custom Bottlerocket AMI.
2. I haven't tested GPU workloads. They're supposed to work, but NVIDIA is gonna NVIDIA...
3. On EKS and kOps you can use pretty much anything, so long as you have a compatible kubelet, but Karpenter is getting increasingly prescriptive, which is not inherently bad.

Slides available on GitHub:

Slides available on GitHub: