



# Benchmarking Kubernetes Management on AWS

Comparing DIY efforts with kops and EKS vs.  
fully managed Platform9 KaaS



## Contents

Introduction.....	1
Options for managing Kubernetes on AWS .....	1
Setup — what’s involved.....	2
Setup — how long it takes.....	3
Setup — details.....	4
Day-2 — what’s involved.....	5
Day-2 — how long it takes .....	6
Day-2 — details.....	7
kops/EKS complexity comparison — setup tasks .....	8
kops/EKS complexity comparison — Day-2 tasks .....	11
Conclusion .....	12

---

***Managing production Kubernetes deployed on AWS is complex and takes significant time, resources, and expertise.***

## Introduction

By anyone’s account, managing production Kubernetes deployed on AWS is complex. And doing it successfully — so that enterprise apps are always up-and-running and cost-effective — takes significant resources and expertise. Given the broad spectrum of open-source Kubernetes tools and infrastructure configurations, there are many ways to make it work ranging from in-house, do-it-yourself efforts to various degrees of outsourcing.

This white paper presents benchmark data that will help you figure out the most cost-effective way to manage a production Kubernetes environment on AWS. With our years of experience working on every facet of the Kubernetes world and hands-on testing and evaluation of three key management alternatives, we’ve calculated the time and resources it takes for each step. This will help you better understand the time, effort, and complexity of setting up and managing Kubernetes.

With fully managed Platform9 KaaS, this complexity is our job, not yours. We do the heavy lifting in the background while you focus on getting the most out of your apps and services.

## Options for managing Kubernetes on AWS

Enterprises can manage Kubernetes either in-house (DIY), with help from a public cloud service, or by outsourcing to a fully managed KaaS.

- Kubernetes operations (kops) — in-house resources set up kops, an open-source project, to create, destroy, upgrade, and maintain Kubernetes clusters. It enables deploying a highly available (HA) Kubernetes control plane and generating Terraform configurations as well as supporting managed add-ons including the AWS Load Balancer Controller, cluster autoscalers, cert-manager, and Metrics Server.

- Amazon Elastic Kubernetes Service (EKS) — in-house resources manage EKS, AWS's native implementation for hosted Kubernetes that supports AWS EC2, ELB, VPC, and Fargate services. Its proprietary control plane runs across multiple AWS Availability Zones (AZs) and hides the control plane, including etcd, and prevents any API server customization.
- Platform9 fully managed KaaS — Platform9 deploys, manages, monitors, and upgrades clusters on AWS and includes managed open-source add-ons for vital Kubernetes components: Calico CNI, Metrics Server, CoreDNS, Dashboard, and popular monitoring tools.

## Setup — what's involved

Setting up a production Kubernetes environment with kops or EKS requires meeting various prerequisites as well as installing and configuring a number of complementary applications.

- **Installation** — Downloading and configuring Kubernetes components and tools such as kubectl.
- **Monitoring and logging** — Tracking all node, endpoint, service, pod, and ingress events. We used Prometheus and Grafana for the benchmarking data in this white paper.
- **Role-based access control (RBAC) and identity access management (IAM)** — Configuring an authorization mechanism for managing permissions and policies.
- **DNS** — Scheduling a DNS pod and service on a cluster and configuring kubelets for DNS resolutions.
- **Storage** — Setting up plug-in mechanisms (volumes) to connect ephemeral containers with persistent data stores elsewhere.
- **Cluster networking** — Setting up container-to-container, pod-to-pod, pod-to-service, and external-to-service communications. We used Calico for the benchmarking data in this white paper.
- **Load balancing and ingress controller** — Configuring Kubernetes services or ingress API to spread out workloads evenly across clusters.
- **Application package manager** — Enabling application installs, upgrades, and reporting. We used Helm for the benchmarking data in this white paper.

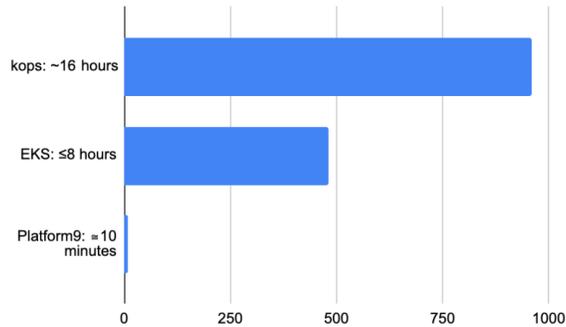
***Setting up a production Kubernetes environment with kops or EKS requires a considerable number of manual steps: checking prerequisites plus installing and configuring several complementary applications.***

## Setup — how long it takes

Setting up an AWS Kubernetes deployment will typically take you:

- kops:  $\approx 16$  hours
- EKS:  $\leq 8$  hours
- Platform9:  $\approx 10$  minutes

***With fully managed Platform9 KaaS, you can get a full production setup ready in 10 minutes compared to several hours and 30 – 40 steps of manual, error-prone activities with kops or EKS.***



Task	kops	AWS EKS	Platform9
Installation	3 hours (10 steps)	1.5 hours (7 steps)	4 – 5 minutes
Monitoring and logging	4 hours (7 steps)	10 mins (3 steps)	0
Role based access control (RBAC) and identity access management (IAM)	<1 hour (4 steps)	<1 hour (5 steps)	2 minutes
DNS	2 hours (2 steps)	1.5 hours (3 steps)	0
Storage	<2 hours (3 steps)	<30 minutes (3 steps)	1 minutes
Cluster networking	1 hour (7 steps)	10-25 minutes (7 steps)	0
Load balancing and ingress controller	<2 hours (6 steps)	<1 hour (5 steps)	2 minutes
Application package manager	<10 minutes to install, <3 hours to run app	<10 minutes to install, <3 hours to run app	2 minutes
<b>Total</b>	<b>≈16 hours</b>	<b>≤8 hours</b>	<b>≈10 minutes</b>

## Setup — details

Completing setup tasks with fully managed Platform9 KaaS requires an estimated 10 minutes of choosing options using a friendly Platform9 dashboard.

Task	kops	AWS EKS
<b>Installation</b>	All steps are manual and require prerequisites and additional tool installations, builds, and configuration. Open-source tools have no warranty for stability or updates and may become deprecated.	Does not support all regions. Requires some open-source tools that have no warranty for stability or updates and may become deprecated.
<b>Monitoring and logging</b>	No warranty for stability.	Amazon CloudWatch
<b>Role based access control (RBAC) and identity access management (IAM)</b>	Can be done during cluster creation.	Integrates RBAC/IAM aws-auth ConfigMap.
<b>DNS</b>	Requires 3rd-party domain registrar.	Default installed test domain.
<b>Storage</b>	AWS requires S3.	Requires driver setup.
<b>Cluster networking</b>	Open source, no warranty for stability.	Calico is not a native AWS service yet is similar to AWS security groups.
<b>Load balancing and ingress controller</b>	Load balancing requires additional configuration.	Load balancing requires additional configuration.
<b>Application package manager</b>	Required to install manually and configure.	Helm package manager requires manual installation and configuration. Using eksctl speeds the process.

## Day-2 — what's involved

Once a Kubernetes environment is up-and-running, continuing operations require ongoing monitoring and maintenance.

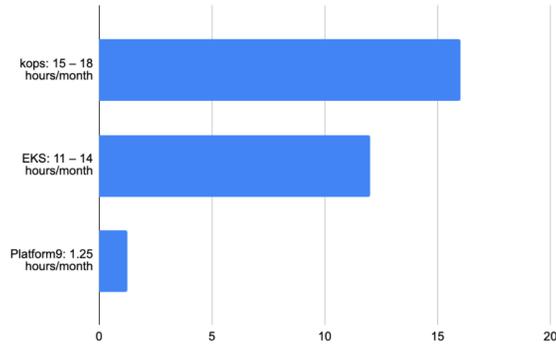
- **Kubernetes upgrades** — Upgrading the control plane, nodes, clients such as kubectl, and adjusting manifests and other resources.
- **Security** — Monitoring Kubernetes security reports and installing patches.
- **Application updates** — Aligning open source and 3rd-party app versions with Kubernetes versions.
- **Scaling** — Changing the number of replicas in a deployment.
- **Backing store** — Maintaining a consistent and highly-available key value store. We used etcd for the benchmarking data in this eBook.
- **Multicloud** — Enabling hybrid and multicloud scenarios and improving the level of high availability beyond single cluster, multi-zone deployments.
- **Multicluster** — Configuring access and authentication to multiple namespaces.
- **Other Day-2 tasks** — Establishing documentation, communication and support resources such as Slack and forums, and ensuring responsiveness.

## Day-2 — how long it takes

Maintaining an AWS Kubernetes deployment will typically take you:

- kops: 15 – 18 hours/month
- EKS: 11 – 14 hours/month
- Platform9: 1.25 hours/month

**Platform9 KaaS lets you offload Day-2 complexity so you can focus on getting the most out of your apps and services. With kops and EKS, you will need to spend many hours on non-value-adding, ongoing operational tasks.**



Task	kops	AWS EKS	Platform9
Kubernetes upgrades 1x/month	15-60 minutes (more 3 steps), depends on if managed node groups or normal.	15 minutes (2 steps)	~1 minute
Security 5x/month	<15 minutes (3 steps)	<20 minutes (2 steps)	~1 minute
Monitoring 10x/month	<60 minutes	<20 minutes	~5 minutes
Application updates .5x/year	<30 minutes	<15 minutes (2 steps)	~1 minute
Scaling 4x/month	20 minutes (3 steps)	30 minutes (3 steps)	~2 minutes
Backing store 1-2x/month	15 minutes (2 steps)	<30 minutes (4 steps)	~4 minutes
Multicloud tasks 1-2x/month	<1 hour	n/a	0
Multicluster tasks 1-2x/month	<1 hour	<4 hours	0
<b>Overall time/month</b>	<b>15 – 18 hours</b>	<b>11 – 14 hours</b>	<b>1 – 2 hours</b>

## Day-2 — details

Day-2 tasks with fully managed Platform9 KaaS require an estimated 1 – 2 hours per month of interaction using a friendly Platform9 dashboard.

Task	kops	AWS EKS
Kubernetes upgrades 1x/month	Cluster updates may fail due to changed VolumeType.	Requires 2 – 3 available IP addresses.
Security 5x/month	Jumping minor versions may work if alpha features are not enabled. Risk of version deprecations. Customer's full responsibility.	Customers are responsible for security in the cloud: <ul style="list-style-type: none"> <li>• The security configuration of the data plane including the configuration of security groups that allow traffic to pass from the Amazon EKS control plane into the customer VPC.</li> <li>• The configuration of the nodes and the containers themselves.</li> <li>• The node's operating system (including updates and security patches).</li> </ul> Other associated application software: <ul style="list-style-type: none"> <li>• Setting up and managing network controls such as firewall rules.</li> <li>• Managing platform-level identity and access management either with or in addition to IAM.</li> <li>• The sensitivity of your data, your company's requirements, and applicable laws and regulations.</li> </ul>
Monitoring 10x/month	Manual setup required for tools like Prometheus, can't monitor multiclouds and multiclusters.	CloudWatch can't monitor multiclouds and multiclusters.
Application updates .5x/year	Involves many open-source and 3rd-party tools.	Usually requires backup, manual downloads, and updating.
Scaling 4x/month	Horizontal pod autoscaling (HPA) requires an additional metrics server.	Scaling presented as AWS Cloud provider which is an extension and Amazon EC2 Auto Scaling groups is used by open-source Cluster Autoscaler, so it makes no warranty as it is open-source.

Task	kops	AWS EKS
Backing store 1 – 2x/month	If cluster is in legacy etcd mode, requires additional coding such as with CloudWatch to trigger an AWS Lambda for creating a new snapshot.	Can use a 3rd-party service such as Velero or Commvault.
Multicloud tasks 1 – 2x/month	Use kubectl for AWS/GKE (3 steps).	No native solution (pre-EKS-Anywhere). Requires 3rd-party solution such as Citrix ADC/Citrix ingress.
Multicluster 1 – 2x/month	Use kubectl.	Using Cilium ClusterMesh.

## kops/EKS complexity comparison — setup tasks

This table lists steps involved in setting up production Kubernetes clusters using kops and EKS. With fully managed Platform9 KaaS, these tasks are either performed in the background or require a minimal amount of setup data entered via a friendly Platform9 dashboard.

	kops	AWS EKS
<b>Task</b>	<b>Kubernetes cluster</b>	
<b>Time</b>	3 hours	1.5 hours
<b>Steps</b>	<ol style="list-style-type: none"> <li>Download and install kops and kubectl.</li> <li>Create an Amazon user and set up permissions.</li> <li>Configure access keys.</li> <li>Download, install, and set up the AWS CLI.</li> <li>Create an Amazon S3 bucket for the K8s state store.</li> <li>Download and install the jq tool.</li> <li>Generate a Route 53 hosted zone.</li> <li>Configure DNS (can use a local cluster).</li> <li>Create the Kubernetes cluster.</li> <li>Configure RBAC if needed.</li> </ol>	<ol style="list-style-type: none"> <li>Start the EKS wizard.</li> <li>Configure the cluster.</li> <li>Specify networking.</li> <li>Configure logging.</li> <li>Review and approve details that are collected in the wizard and launch installation.</li> <li>Download and install the AWS CLI.</li> <li>Create a kubeconfig for Amazon EKS.</li> </ol>
<b>Comments</b>	All steps should be done manually. There are many prerequisites and additional tools involved. <a href="#">More ...</a>	Not all regions support cluster creation, support responses take time, and additional tools and installations are required. <a href="#">More ...</a>

***With fully managed Platform9 KaaS, these setup tasks are either performed in the background or require a minimal amount of setup data entered via a friendly Platform9 dashboard.***

	kops	AWS EKS
<b>Task</b>	<b>Monitoring and logging</b>	
<b>Time</b>	4 hours	10 minutes (with updates)
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Configure kubelet with these flags: authentication-token-webhook=true, authorization-mode=Webhook</li> <li>2. Install <a href="#">jsonnet-bundler</a>.</li> <li>3. Install prometheus-operator.</li> <li>4. Compile.</li> <li>5. Apply the kube-prometheus stack.</li> <li>6. Install and compile containers.</li> <li>7. Configure.</li> </ol>	<ol style="list-style-type: none"> <li>1. Open the AWS Management console.</li> <li>2. Change logging options and save.</li> <li>3. Wait for updates.</li> </ol>
<b>Comments</b>	Typically uses Prometheus and Grafana. <a href="#">More ...</a>	Uses Amazon CloudWatch. <a href="#">More ...</a>
<b>Task</b>	<b>RBAC</b>	
<b>Time</b>	<1 hour	1 hour
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Update existing clusters switching to RBAC.</li> <li>2. Add RBAC roles to newly created clusters.</li> <li>3. Create a user in K8s.</li> <li>4. Add a user token to kubectl.</li> </ol>	<ol style="list-style-type: none"> <li>1. Install test pods.</li> <li>2. Create a user.</li> <li>3. Map the IAM user to K8s.</li> <li>4. Create the role and binding.</li> <li>5. Verify the role.</li> </ol>
<b>Comments</b>	<a href="#">More ...</a>	<a href="#">More ...</a>
<b>Task</b>	<b>Ingress</b>	
<b>Time</b>	2 hours	1.5 hours
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Add the kops nginx <a href="#">ingress add-on</a>.</li> <li>2. Create the ingress controller.</li> <li>3. Deploy the AWS Load Balancer Controller.</li> <li>4. Create an IAM OIDC provider, an IAM policy and an IAM role and ServiceAccount.</li> <li>5. Install the TargetGroupBinding CRDs.</li> <li>6. Deploy the Helm chart.</li> </ol>	<ol style="list-style-type: none"> <li>1. Create the <a href="#">ingress controller</a>.</li> <li>2. Deploy the AWS Load Balancer Controller.</li> <li>3. Create an IAM OIDC provider, an IAM policy and an IAM role and ServiceAccount.</li> <li>4. Install the TargetGroupBinding CRDs.</li> <li>5. Deploy the Helm chart.</li> </ol>
<b>Comments</b>	<a href="#">More ...</a>	<a href="#">More ...</a>
<b>Task</b>	<b>Core DNS setup and editing</b>	
<b>Time</b>	30 minutes	2 hours
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Create and register the domain name.</li> <li>2. Create and configure with Amazon Route 53 as the hosted zone.</li> </ol>	<ol style="list-style-type: none"> <li>1. Set up Amazon Route 53 permissions for your external DNS.</li> <li>2. Set up the identity provider.</li> <li>3. Install the external DNS.</li> </ol>

	kops	AWS EKS
<b>Task</b>	<b>Calico</b>	
<b>Time</b>	<1 hour	25 minutes
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Install Calico.</li> <li>2. Select an Encapsulation Mode.</li> <li>3. Enable Cross-Subnet mode in Calico.</li> <li>4. Configure Calico MTU.</li> <li>5. Configure Calico to use Typha.</li> <li>6. Configure the eBPF dataplane.</li> <li>7. Configure WireGuard.</li> </ol>	<ol style="list-style-type: none"> <li>1. Apply the Calico manifests and create DaemonSets in calico-system.</li> <li>2. Apply the frontend, backend, client, and management UI services.</li> <li>3. Forward your local port 9001 to the management-ui service.</li> <li>4. Configure network policies to isolate the services from each other.</li> <li>5. Configure network policies to allow management UI access to the service.</li> <li>6. Configure network policies to allow traffic from the frontend service to the backend service.</li> <li>7. Configure network policy to allow traffic from the client namespace to the frontend service.</li> </ol>
<b>Comments</b>	<a href="#">More ...</a>	<a href="#">Installation notes. More ...</a>
<b>Task</b>	<b>etcd</b>	
<b>Time</b>	20 minutes	n/a (built-in)
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Install pre-built binaries.</li> <li>2. Administer etcd</li> </ol>	
<b>Comments</b>	<a href="#">Installation notes. More ...</a>	
<b>Task</b>	<b>Backup/restore</b>	
<b>Time</b>	15 minutes	20 minutes
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Backup with etcd-manager.</li> <li>2. Restore with etcd-manager or etcd-manager-ctl.</li> </ol>	<ol style="list-style-type: none"> <li>1. Create an S3 bucket.</li> <li>2. Install Velero client and Velero on EKS.</li> <li>3. Backup/restore with Velero.</li> </ol>
<b>Comments</b>	<a href="#">More ...</a>	<a href="#">More ...</a>

## kops/EKS complexity comparison — Day-2 tasks

This table lists several steps involved in maintaining production Kubernetes clusters using kops and EKS. With fully managed Platform9 KaaS, these tasks are either performed in the background or require a minimal amount of configuration data entered via a friendly Platform9 dashboard.

	kops	AWS EKS
<b>Task</b>	<b>Performing an upgrade</b>	
<b>Time</b>	15 minutes	5 minutes
<b>Steps</b>	<ol style="list-style-type: none"> <li>Determine if any of the nodes will require a restart.</li> <li>Perform the rolling update.</li> <li>Validate.</li> </ol>	<ul style="list-style-type: none"> <li>Update using eksctl.</li> </ul>
<b>Comments</b>	<a href="#">More ...</a>	<a href="#">More ...</a>
<b>Task</b>	<b>Adding and removing nodes (scaling)</b>	
<b>Time</b>	20 minutes	30 minutes
<b>Steps</b>	<ul style="list-style-type: none"> <li>Configure Horizontal Pod Autoscaling (HPA) with kubectl autoscale.</li> </ul>	<ol style="list-style-type: none"> <li>Install kube-ops-view, an operational view of multiple Kubernetes clusters.</li> <li>Install the cluster autoscaler.</li> <li>Configure the autoscaler.</li> </ol>
<b>Comments</b>	<a href="#">More ...</a>	<a href="#">More ...</a>
<b>Task</b>	<b>Adding and removing nodes (multicloud environment)</b>	
<b>Time</b>	<1 hour	<4 hours
<b>Steps</b>	<ol style="list-style-type: none"> <li>Set up kops and kubectl.</li> <li>Create a cluster in AWS using kops.</li> <li>Save the configuration of the kops cluster.</li> <li>Create GKE, AKS, or private-cloud clusters.</li> <li>Save the configurations of the cluster.</li> <li>Merge both configurations.</li> </ol>	<ol style="list-style-type: none"> <li>Identify the cluster subnets that are associated with your cluster.</li> <li>Verify that your worker nodes belong only to the identified subnets.</li> <li>Update the aws-auth ConfigMap with the NodeInstanceRole of your worker nodes.</li> <li>Match the security group requirements of worker nodes.</li> <li>Set tags for worker nodes.</li> <li>Confirm that nodes can reach the API server endpoint for your EKS cluster.</li> <li>Connect the EKS worker node instance with SSH and check kubelet agent logs.</li> </ol>
<b>Comments</b>	<a href="#">More ...</a>	<a href="#">More ...</a>

## Conclusion

The benchmarking data in this eBook shows how setting up and managing production Kubernetes, either completely DIY or partially managed with AWS and EKS, is no simple task. Still, if you have extra in-house talent with requisite skills and production-Kubernetes experience, you can safely choose one of these options.

However, the era of cloud services such as fully managed Platform9 KaaS has set enterprises free. They can focus on their core business apps and let Kubernetes experts take care of cluster deployments without any form of vendor lock-in and with a considerably lower TCO.

The complexity and overhead of DIY Kubernetes is an unnecessary burden. Learn more about Platform9:

- **Start for free** — [Sign up](#) for our Freedom Plan and deploy and manage up to 20 nodes.
- **Read our Quick Start guide** — [Follow](#) the easy step-by-step process to create a cluster using Platform9 in minutes.
- **See a live demo** — [Get](#) an overview of Platform9 and how it can help your organization.
- **Build with Koolkubernetes** — [Extend](#) your Kubernetes deployment with our Github repo full of certified apps.
- **Join the Platform9 Forum** — [Learn](#) about new product releases and collaborate with other community members and the Platform9 team.
- **Connect with Slack** — [Join](#) our public Slack channel, ask questions, and chat live with Platform9 team members and product managers.

***The complexity and overhead of DIY Kubernetes is an unnecessary burden. Focus on your core business apps and let Platform9 fully managed KaaS and expertise take care of cluster deployments and on-going operations — without vendor lock-in and with a considerably lower TCO.***

***Learn more at [Platform9.com!](https://platform9.com)***