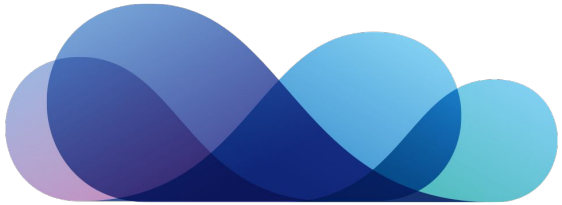


Amazon EKS

2019-01-23

Sebastian Toader
sebastian@banzaicloud.com

Amazon EKS

- Amazon EKS initial release - 2018 (June)
  - Available in only 2 regions: us-east-1, us-west-1
  - Platform: eks.1 - only a few admission controller enabled
  - Kubernetes 1.10

- The list of supported regions continuously growing
  - US West (Oregon) (us-west-2)
  - US East (N. Virginia) (us-east-1)
  - US East (Ohio) (us-east-2)
  - EU (Frankfurt) (eu-central-1)
  - EU (Stockholm) (eu-north-1)
  - EU (Ireland) (eu-west-1)
  - Asia Pacific (Tokyo) (ap-northeast-1)
  - Asia Pacific (Seoul) (ap-northeast-2)
  - Asia Pacific (Singapore) (ap-southeast-1)
  - Asia Pacific (Sydney) (ap-southeast-2)

- Supported Kubernetes versions: 1.10, 1.11
- Control plane
  - Managed
  - HA (multi AZ)
- Cluster create time: ~10-15 min (our observation)
- Worker nodes:
  - Not managed
  - HA (multi AZ)
  - GPU support
- Cost:
  - Control plane: 0.2 USD per hour
  - AWS resources
- Kubernetes upgrade
  - Rather manual: kube-dns -> coredns; patch kube-proxy; migrate to new worker nodes

Amazon EKS: Cluster create flow

- Pre-requisites
  - IAM role that Kubernetes can assume to create AWS resources
  - VPC
  - Subnets in at least two AZs
  - Security group(s) for Control Plane and worker nodes
- Provision control plane
- Once control plane provisioned create worker nodes manually

Amazon EKS: Pre-requisites

- IAM role that Kubernetes can assume to create AWS resources
  - EKS: Allows Amazon EKS to manage your clusters on your behalf

- VPC
  - Amazon EKS tags the VPC containing the subnets
    - `kubernetes.io/cluster/<cluster-name>: shared`
  - Must have DNS hostname and DNS resolution support otherwise, worker nodes cannot register with the cluster
  - Route table:
    - 0.0.0.0/0 destination is routed to the internet gateway

Amazon EKS: Worker nodes

- Subnets in at least 2 AZs
  - **Private-only**: Everything runs in a private subnet and Kubernetes cannot create internet-facing load balancers for your pods.
  - **Public-only**: Everything runs in a public subnet, including your worker nodes.
    - Internet-facing load balancers require a public subnet in your cluster.
  - Amazon EKS tags subnets
    - `kubernetes.io/cluster/<cluster-name>: shared`
  - Private subnets tag for internal load balancers
    - `kubernetes.io/role/internal-elb: 1`
- Security group(s) for Control Plane and worker nodes
  - Defines rules that governs communication between control plane and worker nodes also worker node to worker node

Amazon EKS: Pre-requisites

- Security group(s) for Control Plane and worker nodes
    - Defines rules that governs communication between control plane and worker nodes also worker node to worker node
- Control Plane security group

	Protocol	Port Range	Source	Destination
Minimum inbound traffic	TCP	443	All worker node security groups	
Recommended inbound traffic	TCP	443	All worker node security groups	
Minimum outbound traffic	TCP	10250		All worker node security groups
Recommended outbound traffic	TCP	1025-65535		All worker node security groups

Amazon EKS: Pre-requisites

- Worker node security group

	Protocol	Port Range	Source	Destination
Minimum inbound traffic (from other worker nodes)	Any protocol you expect your worker nodes to use for inter-worker communication	Any ports you expect your worker nodes to use for inter-worker communication	All worker node security groups	
Minimum inbound traffic (from control plane)	TCP	10250	Control plane security group	
Recommended inbound traffic	All TCP	All 443, 1025-65535	All worker node security groups Control plane security group	
Minimum outbound traffic	TCP	443		Control plane security group
Recommended outbound traffic	All	All		0.0.0.0/0

Amazon EKS: Pre-requisites

- CF Template: https://amazon-eks.s3-us-west-2.amazonaws.com/cloudformation/2019-01-09/amazon-eks-vpc-sample.yaml

Amazon EKS: Worker nodes

- Launch worker node groups after Control plane becomes “ACTIVE”
- CF template: https://amazon-eks.s3-us-west-2.amazonaws.com/cloudformation/2019-01-09/amazon-eks-nodegroup.yaml
- AMI - specific for region, Kubernetes version and instance type (normal vs GPU)
- User datascript

```
/etc/eks/bootstrap.sh ${ClusterName} ${BootstrapArguments}

/opt/aws/bin/cfn-signal --exit-code $? \
            --stack  ${AWS::StackName} \
            --resource NodeGroup  \
            --region ${AWS::Region}
```

- Node Instance IAM role - defines policies for worker node instances to access AWS resources

Amazon EKS: Worker nodes

- Allow worked node join the cluster

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: aws-auth
  namespace: kube-system
data:
  mapRoles: |
    - rolearn: <ARN of instance role (not instance
profile)>
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
```

Amazon EKS: Cluster authentication

- EKS uses AWS IAM Authenticator for Kubernetes: **aws-iam-authenticator**
- Kubeconfig for EKS clusters

```
apiVersion: v1
clusters:
- cluster:
    server: <endpoint-url>
    certificate-authority-data: <base64-encoded-ca-cert>
  name: kubernetes
...
kind: Config
preferences: {}
users:
- name: aws
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      command: aws-iam-authenticator
      args:
        - "token"
        - "-i"
        - "<cluster-name>"
        # - "-r"
        # - "<role-arn>"
      # env:
        # - name: AWS_PROFILE
        #   value: "<aws-profile>"
```

Amazon EKS: Cluster authentication

- EKS uses AWS IAM Authenticator for Kubernetes: **aws-iam-authenticator**

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: aws-auth
  namespace: kube-system
data:
  mapRoles: |
    - rolearn: <ARN of instance role (not instance profile)>
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
  mapUsers: |
    - userarn: <ARN of IAM user>
      username: admin
      groups:
        - system:masters
```

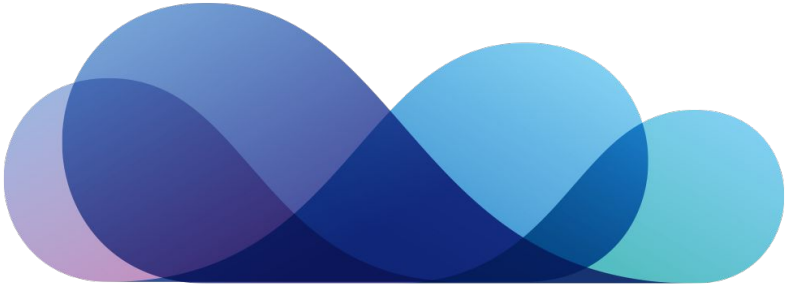
Amazon EKS: Storage class

- Amazon EKS with Kubernetes version 1.10 lacks default storage class

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: gp2
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: kubernetes.io/aws-ebs
parameters:
  type: gp2
  fsType: ext4
```

- **WaitForFirstConsumer** binding available only starting from Kubernetes version 1.12 for aws-ebs


Questions?!



BANZAICLOUD