Explore advanced container orchestration with Kubernetes on your cloud platform (e.g., Amazon EKS, Azure Kubernetes Service, Google Kubernetes Engine).

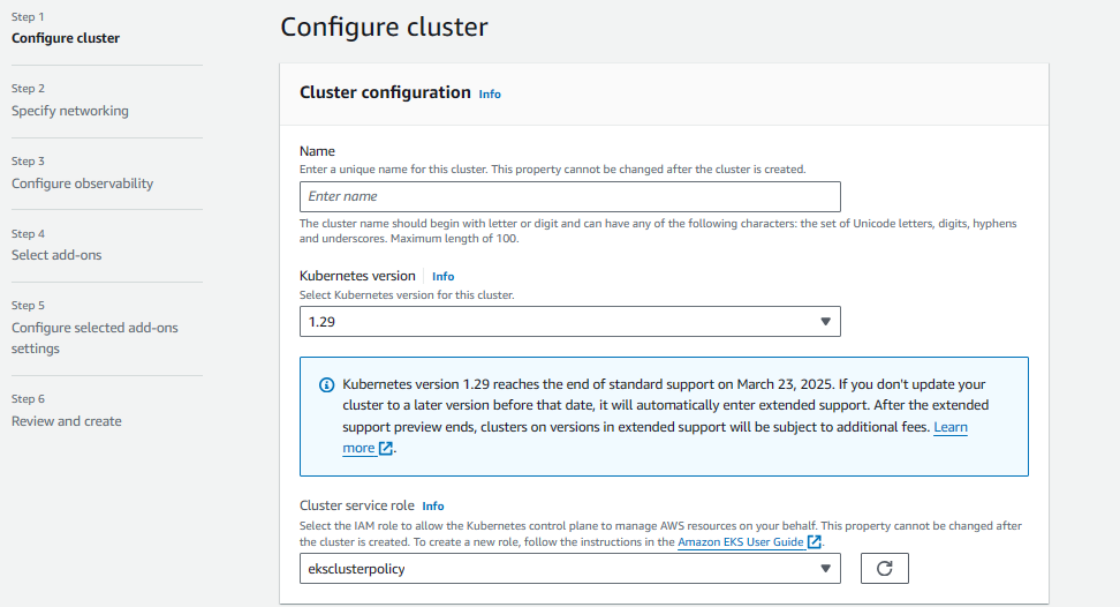
Ebuka Obiakor – 9th March, 2024

Things you need to do.

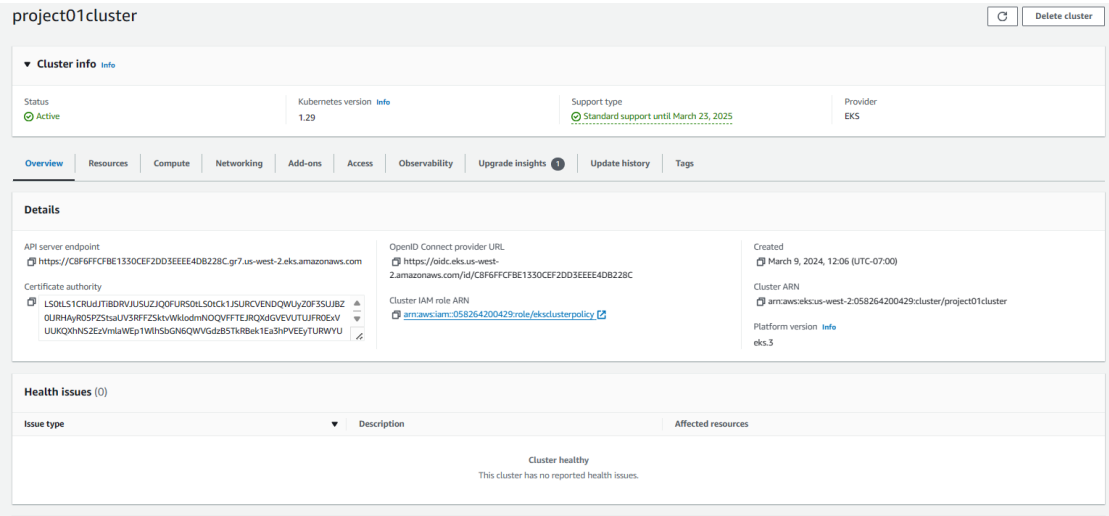
Install the Kubernetes service [Install and Set Up kubectl on Windows | Kubernetes]

Install the Amazon CLI (if not installed already)

Set up the cluster with AWS console or CLI (you can also use eksctl). I used the AWS console.



Create cluster using console, select Kubernetes version, cluster access, VPC, and then proceed. (Managed EKS is used here). *Ensure to set cluster policy for EKS.*

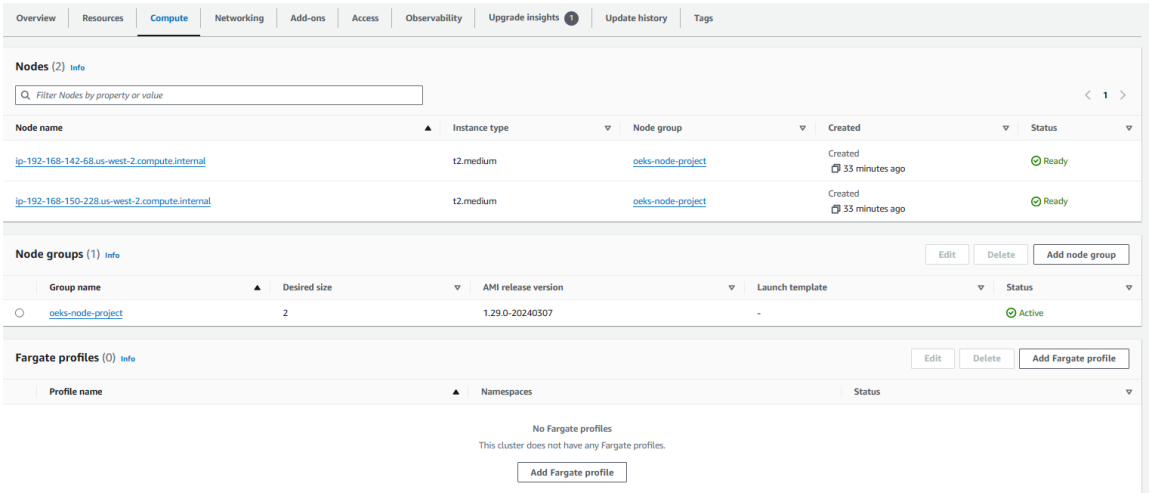


Once the cluster is set up, it can be connected to the Kubernetes service using.
“aws eks update-kubeconfig --region <region> --name <cluster_name>” . This assumes that the aws configure has been set up for the console.

```
PS C:\Users\owner> kubectl cluster-info
Kubernetes control plane is running at https://C8F6FFCFBE1330CEF2DD3EEEE4DB228C.gr7.us-west-2.eks.amazonaws.com
CoreDNS is running at https://C8F6FFCFBE1330CEF2DD3EEEE4DB228C.gr7.us-west-2.eks.amazonaws.com/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
```

```
PS C:\Users\owner> aws eks describe-cluster --name project01cluster
{
    "cluster": {
        "name": "project01cluster",
        "arn": "arn:aws:eks:us-west-2:058264200429:cluster/project01cluster",
        "createdAt": "2024-03-09T12:06:24.214000-07:00",
        "version": "1.29",
        "endpoint": "https://C8F6FFCFBE1330CEF2DD3EEEE4DB228C.gr7.us-west-2.eks.amazonaws.com",
        "roleArn": "arn:aws:iam::058264200429:role/eksclusterpolicy",
        "resourcesVpcConfig": {
            "subnetIds": [
                "subnet-0f3f74e1c9919710c",
                "subnet-06b6d065c7ae24ae5",
                "subnet-09f4b2559a23da042"
            ],
            "securityGroupIds": [
                "sg-021afa73f2e3a0f5e"
            ],
            "clusterSecurityGroupId": "sg-09d37867cbc97f75d",
            "vpcId": "vpc-0643a4ed5de6f9925",
            "endpointPublicAccess": true,
            "endpointPrivateAccess": false,
            "publicAccessCidrs": [
                "0.0.0.0/0"
            ]
        },
        "kubernetesNetworkConfig": {
            "serviceIpv4Cidr": "10.100.0.0/16",
            "ipFamily": "ipv4"
        },
        "logging": {
            "clusterLogging": [
                {
                    "types": [
```

Next, you can set up the worker nodes; in this example I used 2 worker node machines. Verified that nodes created successfully.



Deployed a sample application unto the nodes.

```
PS C:\Users\owner> kubectl apply -f https://k8s.io/examples/application/deployment.yaml -n ebuka-namespace
deployment.apps/nginx-deployment created
PS C:\Users\owner> kubectl get deployments
No resources found in default namespace.
PS C:\Users\owner> kubectl get deployments -n ebuka-namespace
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           58s
PS C:\Users\owner> kubectl describe deployment nginx-deployment
Error from server (NotFound): deployments.apps "nginx-deployment" not found
PS C:\Users\owner> kubectl describe deployment -n ebuka-namespace nginx-deployment
Name:                   nginx-deployment
Namespace:              ebuka-namespace
CreationTimestamp:      Mon, 11 Mar 2024 05:27:24 -0600
Labels:                 <none>
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx
Replicas:               2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
```

```
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-86dcfdf4c6-d2x6x   1/1     Running   0          7m58s
nginx-deployment-86dcfdf4c6-d7qjw   1/1     Running   0          7m58s
PS C:\Users\owner> kubectl describe -n ebuka-namespace pod nginx-deployment-86dcfdf4c6-d2x6x
Name:             nginx-deployment-86dcfdf4c6-d2x6x
Namespace:        ebuka-namespace
Priority:         0
Service Account:  default
Node:             ip-192-168-150-228.us-west-2.compute.internal/192.168.150.228
Start Time:       Mon, 11 Mar 2024 05:27:24 -0600
Labels:           app=nginx
                  pod-template-hash=86dcfdf4c6
Annotations:      <none>
Status:           Running
IP:               192.168.155.85
IPs:
  IP:           192.168.155.85
Controlled By:  ReplicaSet/nginx-deployment-86dcfdf4c6
Containers:
  nginx:
    Container ID:   containerd://3d30b991534b607919c2ac893c7efb8ed9cf71be56efe745b85cdc72895e0300
    Image:          nginx:1.14.2
    Image ID:       docker.io/library/nginx@sha256:f7988fb6c02e0ce69257d9bd9cf37ae20a60f1df7563c3a2a6abe24160306b8d
    Port:           80/TCP
```

```
PS C:\Users\owner> kubectl apply -f https://k8s.io/examples/application/deployment.yaml -n ebuka-namespace
deployment.apps/nginx-deployment created
PS C:\Users\owner> kubectl get deployments
No resources found in default namespace.
PS C:\Users\owner> kubectl get deployments -n ebuka-namespace
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           58s
PS C:\Users\owner> kubectl describe deployment nginx-deployment
Error from server (NotFound): deployments.apps "nginx-deployment" not found
PS C:\Users\owner> kubectl describe deployment -n ebuka-namespace nginx-deployment
Name:                   nginx-deployment
Namespace:              ebuka-namespace
CreationTimestamp:      Mon, 11 Mar 2024 05:27:24 -0600
Labels:                 <none>
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx
Replicas:               2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
```

```
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-86dcfdf4c6-d2x6x   1/1     Running   0          7m58s
nginx-deployment-86dcfdf4c6-d7qjw   1/1     Running   0          7m58s
PS C:\Users\owner> kubectl describe -n ebuka-namespace pod nginx-deployment-86dcfdf4c6-d2x6x
Name:             nginx-deployment-86dcfdf4c6-d2x6x
Namespace:        ebuka-namespace
Priority:         0
Service Account:  default
Node:             ip-192-168-150-228.us-west-2.compute.internal/192.168.150.228
Start Time:       Mon, 11 Mar 2024 05:27:24 -0600
Labels:           app=nginx
                  pod-template-hash=86dcfdf4c6
Annotations:      <none>
Status:           Running
IP:               192.168.155.85
IPs:
  IP:           192.168.155.85
Controlled By:  ReplicaSet/nginx-deployment-86dcfdf4c6
Containers:
  nginx:
    Container ID:   containerd://3d30b991534b607919c2ac893c7efb8ed9cf71be56efe745b85cdc72895e0300
    Image:          nginx:1.14.2
    Image ID:       docker.io/library/nginx@sha256:f7988fb6c02e0ce69257d9bd9cf37ae20a60f1df7563c3a2a6abe24160306b8d
    Port:           80/TCP
```

Cleaned up (deleted) resources – example for demo only

Resources / links

https://docs.aws.amazon.com/eks/latest/userguide/getting-started-console.html
https://kubernetes.io/docs/tasks/run-application/run-stateless-application-deployment/
https://www.youtube.com/watch?v=CukYk43agA4&ab_channel=KodeKloud
https://youtu.be/O4h69KMm2tE?si=ci67GM9oGhw4SlI2
https://www.youtube.com/watch?v=DcnviAwmzM4&ab_channel=K21Academy