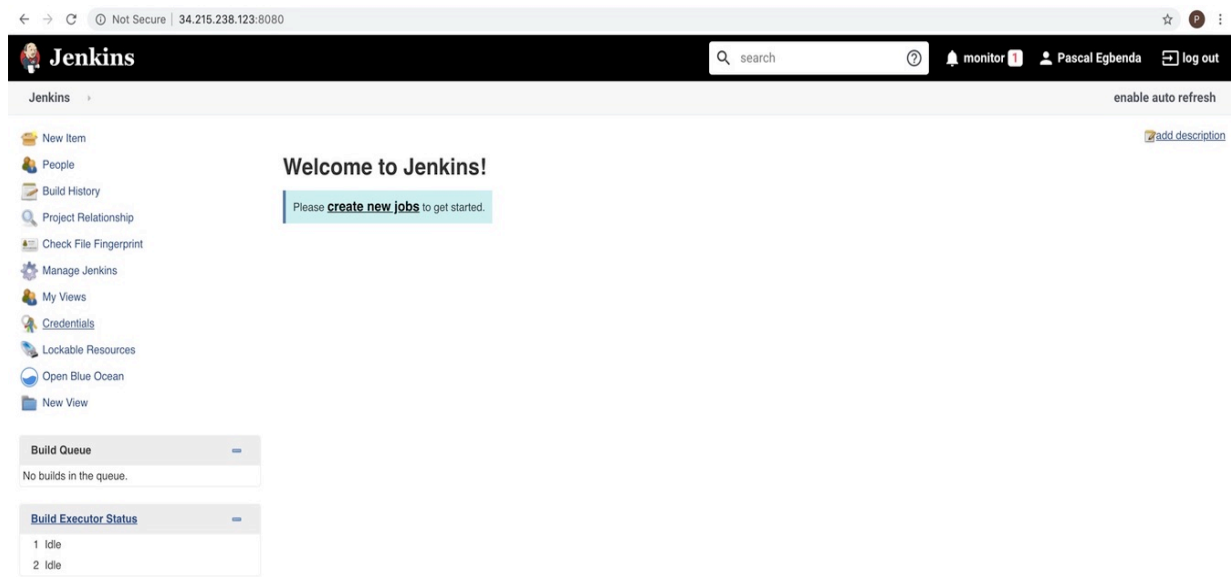


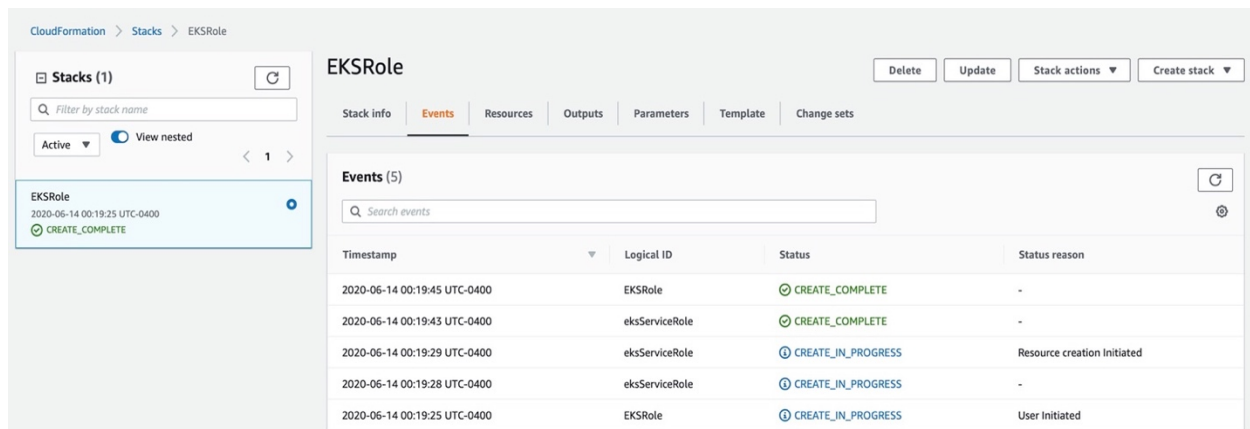


Completed Jenkins setup after all necessary plugins are installed is shown below:



## Setup EKSRole

This role will be used by Amazon EKS to create AWS resources for the Kubernetes clusters



## Setup VPC and EKS Cluster

Create VPC and all necessary resources for EKS Cluster using CloudFormation templates.

```
Pascals-MacBook-Pro:AWS-Cloudformation pascalegbenda$ ./create.sh EKS network_eks.yml network_eks.json
{
  "StackId": "arn:aws:cloudformation:us-west-2:504716675564:stack/EKS/a0f496a0-adf8-11ea-a378-0ae10a278694"
}
Pascals-MacBook-Pro:AWS-Cloudformation pascalegbenda$
```



## Completed stack

CloudFormation > Stacks > EKS

Stacks (2)

Filter by stack name

Active View nested

1

EKS  
2020-06-14 00:36:35 UTC-0400  
CREATE\_COMPLETE

EKSRole  
2020-06-14 00:19:25 UTC-0400  
CREATE\_COMPLETE

EKS

Delete Update Stack actions Create stack

Stack info Events Resources Outputs Parameters Template Change sets

Resources (22)

Search resources

Logical ID	Physical ID	Type	Status	Status reason
ClusterControlPlaneSecurityGroup	sg-018fc5c7fd40052e4	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
ClusterControlPlaneSecurityGroupIngress	ClusterControlPlaneSecurityGroupIngress	AWS::EC2::SecurityGroupIngress	CREATE_COMPLETE	-
ControlPlaneEgressToNodeSecurityGroup	EKS-ControlPlaneEgressToNodeSecurityGroup-C1LBBQHGVSD9	AWS::EC2::SecurityGroupEgress	CREATE_COMPLETE	-
ControlPlaneEgressToNodeSecurityGroupOn443	EKS-ControlPlaneEgressToNodeSecurityGroupOn443-11C7GTL6DWQ57	AWS::EC2::SecurityGroupEgress	CREATE_COMPLETE	-
DefaultPublicRoute	EKS-Default-1WCBSMEB0AJ9Y	AWS::EC2::Route	CREATE_COMPLETE	-
EKSCluster	Blue-Green_Prod	AWS::EKS::Cluster	CREATE_COMPLETE	-

## Running Lint on Pipeline

### 1. HTML Lint

Nanodegree Cloud DevOps Capstone 1

Pipeline Changes Tests Artifacts

Branch: master 22s No changes

Commit: d8e306c a minute ago Branch indexing

Start Lint HTML Lint Dockerfile Build Docker Image Push Docker Image End

Lint HTML - <1s

Restart Lint HTML

- Check out from version control 1s
- tidy -q -e Blue-Green/Blue/index.html -- Shell Script <1s
- tidy -q -e Blue-Green/Green/index.html -- Shell Script <1s

### 2. Lint Dockerfile

Nanodegree Cloud DevOps Capstone 1

Pipeline Changes Tests Artifacts

Branch: master 22s No changes

Commit: d8e306c a minute ago Branch indexing

Start Lint HTML Lint Dockerfile Build Docker Image Push Docker Image End

Lint Dockerfile - 7s

Restart Lint Dockerfile

- Check out from version control 1s
- Checks if running on a Unix-like node <1s
- docker inspect -f, hadolint/hadolint:latest-debian -- Shell Script <1s
- hadolint /Blue-Green/Blue/Dockerfile | tee -a hadolint\_int.txt -- Shell Script <1s
- lintErrors=\$(stat --print=%s hadolint\_int.txt) if [ "\$lintErrors" -gt 0 ]; then echo "Check Error" cat hadolint\_int.txt exit 1 else echo "No Error" fi -- Shell Script <1s
- hadolint /Blue-Green/Green/Dockerfile | tee -a hadolint\_int.txt -- Shell Script <1s
- lintErrors=\$(stat --print=%s hadolint\_int.txt) if [ "\$lintErrors" -gt 0 ]; then echo "Check Error" cat hadolint\_int.txt exit 1 else echo "No Error" fi -- Shell Script <1s

### 3. Build Docker Images

✓ Nanodegree\_Cloud\_DevOps\_Capstone 1

PipelineChangesTestsArtifacts🔄✎⚙️📄Logout✕

Branch: master 22s No changes  
Commit: d8e306c 3 minutes ago Branch indexing

Start

Lint HTML

Lint Dockerfile

Build Docker Image

Push Docker Image

End

Build Docker Image - <1s Restart Build Docker Image

✓ > Shell Script

✓ > Shell Script

<1s<1s

### 4. Push Docker Images to Docker hub

✓ Nanodegree\_Cloud\_DevOps\_Capstone 1

PipelineChangesTestsArtifacts🔄✎⚙️📄Logout✕

Branch: master 22s No changes  
Commit: d8e306c 3 minutes ago Branch indexing

Start

Lint HTML

Lint Dockerfile

Build Docker Image

Push Docker Image

End

Push Docker Image - 10s Restart Push Docker Image

✓ > Shell Script

✓ > Shell Script

✓ > Shell Script

✓ > Shell Script

✓ > Shell Script

✓ > Shell Script

2s<1s2s2s<1s2s

### Images on Docker Hub

Thank you for attending DockerCon 2020! [Watch the recordings.](#)

Search for great content (e.g., mysql)

ExploreRepositoriesOrganizationsGet Help ▾pascalgbenda

pascalgbenda ▾

Search by repository name...

Create Repository

pascalgbenda / capstone\_green\_deployment

Updated 24 minutes ago

☆ 0

📄 22

🔓 PUBLIC

pascalgbenda / capstone\_blue\_deployment

Updated 24 minutes ago

☆ 0

📄 23

🔓 PUBLIC

Tip: Not finding your repository? Try switching namespace via the top left dropdown.

# Checking the EKS Cluster and All Necessary Resources Created

## 1. EKS Cluster

EKS > Clusters > Blue-Green\_Prod

Blue-Green\_Prod

Refresh

Delete

Cluster configuration

Kubernetes version

Info

1.16

Status

Active

Platform version

Info

eks.1

Details

Compute

Networking

Logging

Updates

Tags

Details

API server endpoint

https://A5960947D8E1D63362DCA50FA05618BE.y4.us-west-2.eks.amazonaws.com

OpenID Connect provider URL

https://oidc.eks.us-west-2.amazonaws.com/id/A5960947D8E1D63362DCA50FA05618BE

Cluster ARN

arn:aws:eks:us-west-2:504716675564:cluster/Blue-Green\_Prod

Creation time

Jun 14th 2020 at 10:56 PM

Certificate authority

LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSOtLS0tCk1JSUN5RENDQWJDZ0F3SUJBZ0lCQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRF

Cluster IAM Role ARN

arn:aws:iam::504716675564:role/EKSRole-eksServiceRole-13NNHHDPSEOF

## 2. VPC

Create VPC

Actions

Filter by tags and attributes or search by keyword

1 to 2 of 2

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR (Network Border Group)	DHCP options set	Main Route table	Main Network
EKS	vpc-0af56fe2798b9ca8e	available	10.0.0.0/16	-	dopt-0ef148843c79f9cf6	rtb-0d504edb22986c0ba	acl-0091229d
	vpc-61365519	available	172.31.0.0/16	-	dopt-0ef148843c79f9cf6	rtb-2ec48955	acl-a7cd36dc

## 3. Cluster Nodes

Launch Instance

Connect

Actions

Filter by tags and attributes or search by keyword

1 to 4 of 4

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
Blue-Green_Prod-EKS-Nodegroup-Node	i-071a61323d57e1f4e	t3.medium	us-west-2b	running	2/2 checks ...	None	ec2-54-190-34-134.us-...	54.190.34.134	-
Blue-Green_Prod-EKS-Nodegroup-Node	i-08c863ea758505ae3	t3.medium	us-west-2b	running	2/2 checks ...	None	ec2-35-162-241-55.us-...	35.162.241.55	-
Blue-Green_Prod-EKS-Nodegroup-Node	i-0be38d9578c27f6e1	t3.medium	us-west-2a	running	2/2 checks ...	None	ec2-52-38-250-228.us-...	52.38.250.228	-
Jenkins_Ubuntu_Server	i-041c08949c684485f	t2.medium	us-west-2b	running	2/2 checks ...	None	ec2-52-32-199-97.us-w...	52.32.199.97	-

## 4. Autoscaling Group

Create Auto Scaling group

Actions

Filter: Filter Auto Scaling groups...

1 to 1 of 1 Auto Scaling Groups

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
EKS-NodeAut...	EKS-NodeLaunchConf...	3	3	2	4	us-west-2a, us-west-2b	300	0



Connected to my EKS Cluster from local PC to issue commands

Setting the Blue and Green Controllers yields:

```
Pascals-MacBook-Pro:Nanodegree-Cloud-DevOps-Capstone-Project pascalegbenda$ cd Blue-Green
Pascals-MacBook-Pro:Blue-Green pascalegbenda$ ls
Blue    Green
Pascals-MacBook-Pro:Blue-Green pascalegbenda$ cd Blue
Pascals-MacBook-Pro:Blue pascalegbenda$ ls
Dockerfile          blue-controller.json          index.html
Pascals-MacBook-Pro:Blue pascalegbenda$ kubectl apply -f blue-controller.json
replicationcontroller/blue created
Pascals-MacBook-Pro:Blue pascalegbenda$
```

```
Pascals-MacBook-Pro:Blue pascalegbenda$ cd ..
Pascals-MacBook-Pro:Blue-Green pascalegbenda$ cd Green
Pascals-MacBook-Pro:Green pascalegbenda$ ls
Dockerfile          green-controller.json          index.html
Pascals-MacBook-Pro:Green pascalegbenda$ kubectl apply -f green-controller.json
replicationcontroller/green created
Pascals-MacBook-Pro:Green pascalegbenda$
```

Setting the blue-green service yields

```
Pascals-MacBook-Pro:Nanodegree-Cloud-DevOps-Capstone-Project pascalegbenda$ ls
AWS-Cloudformation  Blue-Green  blue-green-service.json
Pascals-MacBook-Pro:Nanodegree-Cloud-DevOps-Capstone-Project pascalegbenda$ kubectl apply -f blue-green-service.json
service/blue-green-lb created
Pascals-MacBook-Pro:Nanodegree-Cloud-DevOps-Capstone-Project pascalegbenda$
```

The above service created the load balancer.

Create Load BalancerActions

Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type
<input type="checkbox"/>	ad19905beff134138aae49062636683b	ad19905beff134138aae4906...		vpc-0af56fe2798b9ca8e	us-west-2b, us-west-2a	classic

Checking the Blue Deployment from load balancer on configured port yields image below:



Changing blue-green service from "blue" to "green" yields image:



Getting results from some Commands

#### 1. Kubectl get nodes

```
Pascals-MacBook-Pro:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
ip-10-0-0-35.us-west-2.compute.internal Ready    <none>   66m   v1.14.9-eks-f459c0
ip-10-0-0-69.us-west-2.compute.internal Ready    <none>   7m25s v1.16.8-eks-e16311
ip-10-0-1-101.us-west-2.compute.internal Ready    <none>   7m20s v1.16.8-eks-e16311
```





## 2. Kubectl get pods

```
Pascals-MacBook-Pro:Nanodegree-Cloud-DevOps-Capstone-Project pascalegbenda$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
blue-9wc8n    1/1     Running   0           57m
green-2d4mz   1/1     Running   0           55m
Pascals-MacBook-Pro:Nanodegree-Cloud-DevOps-Capstone-Project pascalegbenda$
```

## 3. Kubectl get services

```
Pascals-MacBook-Pro:Nanodegree-Cloud-DevOps-Capstone-Project pascalegbenda$ kubectl get services
NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
blue-green-lb LoadBalancer  172.20.113.91 ad19905beff134138aae49062636683b-1562501203.us-west-2.elb.amazonaws.com 8000:32284/TCP 53m
kubernetes    ClusterIP     172.20.0.1    <none>         443/TCP          157m
Pascals-MacBook-Pro:Nanodegree-Cloud-DevOps-Capstone-Project pascalegbenda$
```

## 4. Kubectl describe services/blue-green-lb

```
Pascals-MacBook-Pro:Nanodegree-Cloud-DevOps-Capstone-Project pascalegbenda$ kubectl describe services/blue-green-lb
Name:         blue-green-lb
Namespace:    default
Labels:       app=blue-green-lb
Annotations:  kubectl.kubernetes.io/last-applied-configuration:
               {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"app":"blue-green-lb"},"name":"blue-green-lb","namespace":"default"},
               "spec":{"ports":[{"port":8000,"targetPort":80}],"type":"LoadBalancer"}}
Selector:     app=green
Type:         LoadBalancer
IP:           172.20.113.91
LoadBalancer Ingress: ad19905beff134138aae49062636683b-1562501203.us-west-2.elb.amazonaws.com
Port:         <unset> 8000/TCP
TargetPort:   80/TCP
NodePort:     <unset> 32284/TCP
Endpoints:    10.0.1.104:80
Session Affinity: None
External Traffic Policy: Cluster
Events:
  Type     Reason              Age    From          Message
  ----     -
  Warning  UnAvailableLoadBalancer 56m    service-controller  There are no available nodes for LoadBalancer
  Normal   UpdatedLoadBalancer  14m    (x2 over 15m)  service-controller  Updated load balancer with new hosts
  Normal   EnsuringLoadBalancer  10m    (x2 over 56m)  service-controller  Ensuring load balancer
  Normal   EnsuredLoadBalancer   10m    (x2 over 56m)  service-controller  Ensured load balancer
```