

# OpenShift 4 on AWS Deep Dive

**Ayush Garg**

OpenShift Container Platform Team



What we'll  
discuss today

Prerequisite	install-config.yaml	IAM Break-Down
Prerequisite Check	Installation	S3 Break-Down
AWS Services	AWS Resources Creation and Destruction	Route 53 Break-Down
VPC Networking Architecture	VPC Break-Down	AWS specific features
Cluster Architecture	EC2 Break-Down	How OpenShift identifies its resources on AWS?

# Prerequisite

- ❖ An AWS account with IAM user with AdministratorAccess or root account [\[+\]](#)
- ❖ Programmatic access keys for user to access the AWS API
- ❖ Select a supported AWS region [\[+\]](#)
- ❖ Public hosted zone in Route 53 service
- ❖ AWS account limits [\[+\]](#)

# Prerequisite Check

CONFIDENTIAL designator

The screenshot shows the AWS IAM Management Console interface. The left sidebar contains the navigation menu with 'Users' selected. The main content area displays the 'Summary' section for a user, including the console sign-in link, password status, MFA device, and signing certificates. Below this, the 'Access keys' section shows a table with one active access key. The 'SSH keys for AWS CodeCommit' section shows no results. The 'HTTPS Git credentials for AWS CodeCommit' section is also visible. The top bar shows the date and time as 'Fri Feb 5 5:00 PM'.

**Summary**

- Console sign-in link: <https://cee-indian-shift.signin.aws.amazon.com/console>
- MFA is required when signing in. [Learn more](#)

**Console password** Enabled (last signed in Today) | [Manage](#)

**Assigned MFA device** [arn:aws:iam::467305695994:mfa/lyygar](#) (Virtual) | [Manage](#)

**Signing certificates** None

**Access keys**

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. [Learn more](#)

[Create access key](#)

Access key ID	Created	Last used	Status
AKIAWZTMXBL5MHXEDN6X	2020-11-06 20:41 UTC+0530	2021-02-05 15:39 UTC+0530 with iam in us-east-1	Active   <a href="#">Make inactive</a>   <a href="#">X</a>

**SSH keys for AWS CodeCommit**

Use SSH public keys to authenticate access to AWS CodeCommit repositories. [Learn more](#)

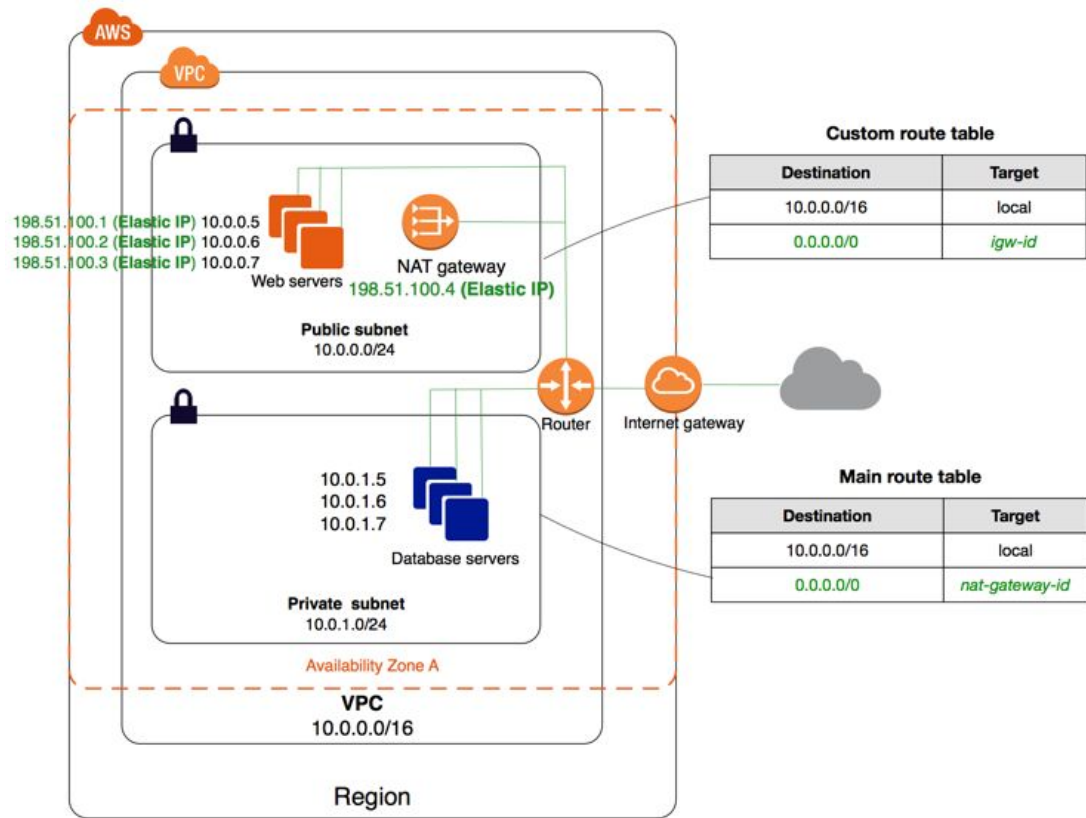
[Upload SSH public key](#)

SSH key ID	Uploaded	Status
No results		

**HTTPS Git credentials for AWS CodeCommit**

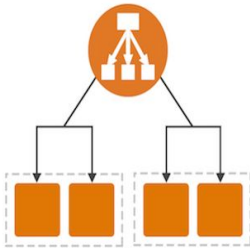
## ❖ Virtual Private Cloud (VPC)

- NAT Gateway
- Internet gateway
- Subnets
- Route Table
- Endpoints



## ❖ Elastic Compute Cloud (EC2)

- Instances
- AMIs
- Volumes
- Security Groups
- Load Balancers
- Target Groups



**ayush-ocp-bd6hb-aext** Delete

arn:aws:elasticloadbalancing:us-east-1:467305695994:targetgroup/ayush-ocp-bd6hb-aext/0a4e0b96874cf0ae

**Basic configuration**

Target type IP	Protocol : Port TCP: 6443	VPC <a href="#">vpc-039fc92b27323ebc8</a>	Load balancer <a href="#">ayush-ocp-bd6hb-ext</a>
-------------------	------------------------------	--	--

Group details **Targets** Monitoring Tags

**Registered targets (3)** Refresh Deregister Register targets

<input type="checkbox"/>	IP address	Port	Zone	Status	Status details
<input type="checkbox"/>	10.0.141.154	6443	us-east-1a	✓ healthy	
<input type="checkbox"/>	10.0.153.187	6443	us-east-1b	✓ healthy	
<input type="checkbox"/>	10.0.161.164	6443	us-east-1c	✓ healthy	

## ❖ Identity and Access Management (IAM)

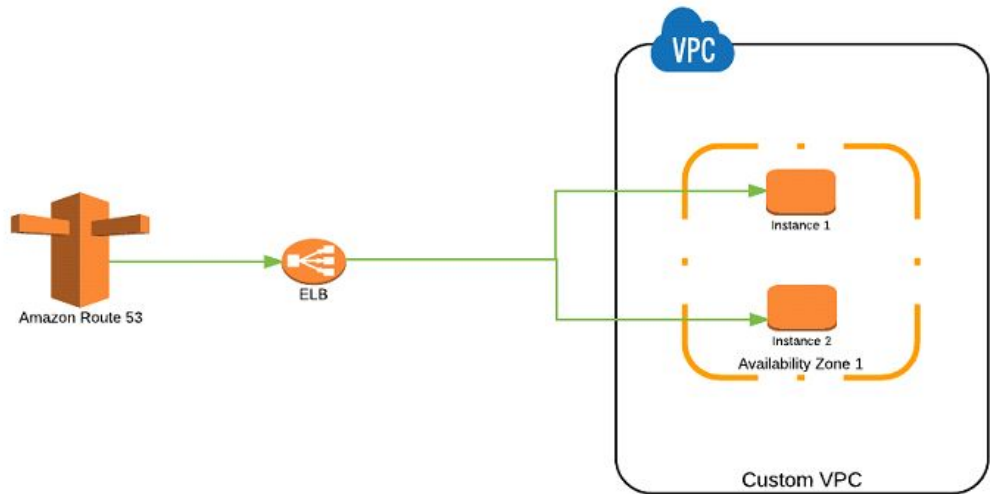
- Users
- Roles
- Security Credentials

## ❖ Simple Storage Service (S3)

- Buckets (object storage)

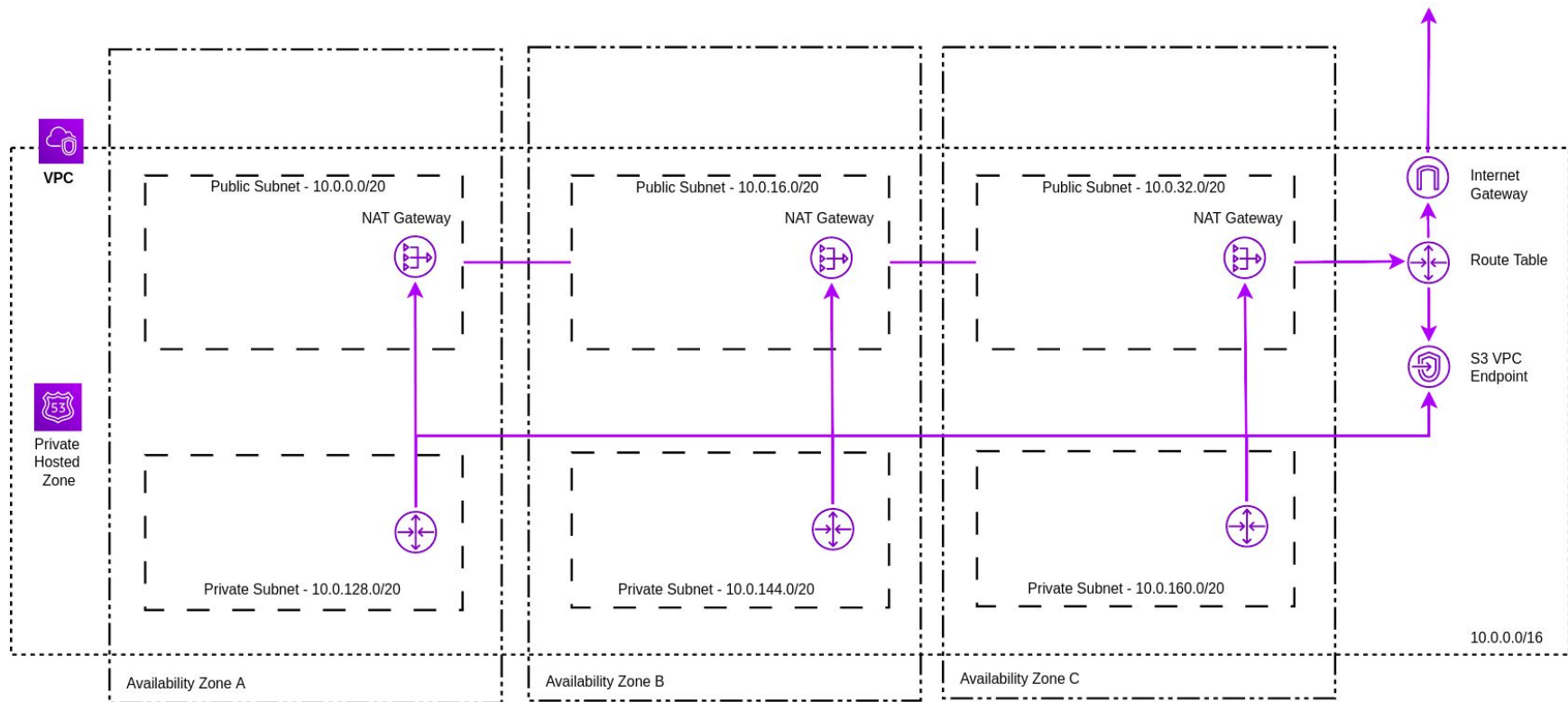
## ❖ Route 53 [ + ]

- Public Hosted Zones
- Private Hosted Zones



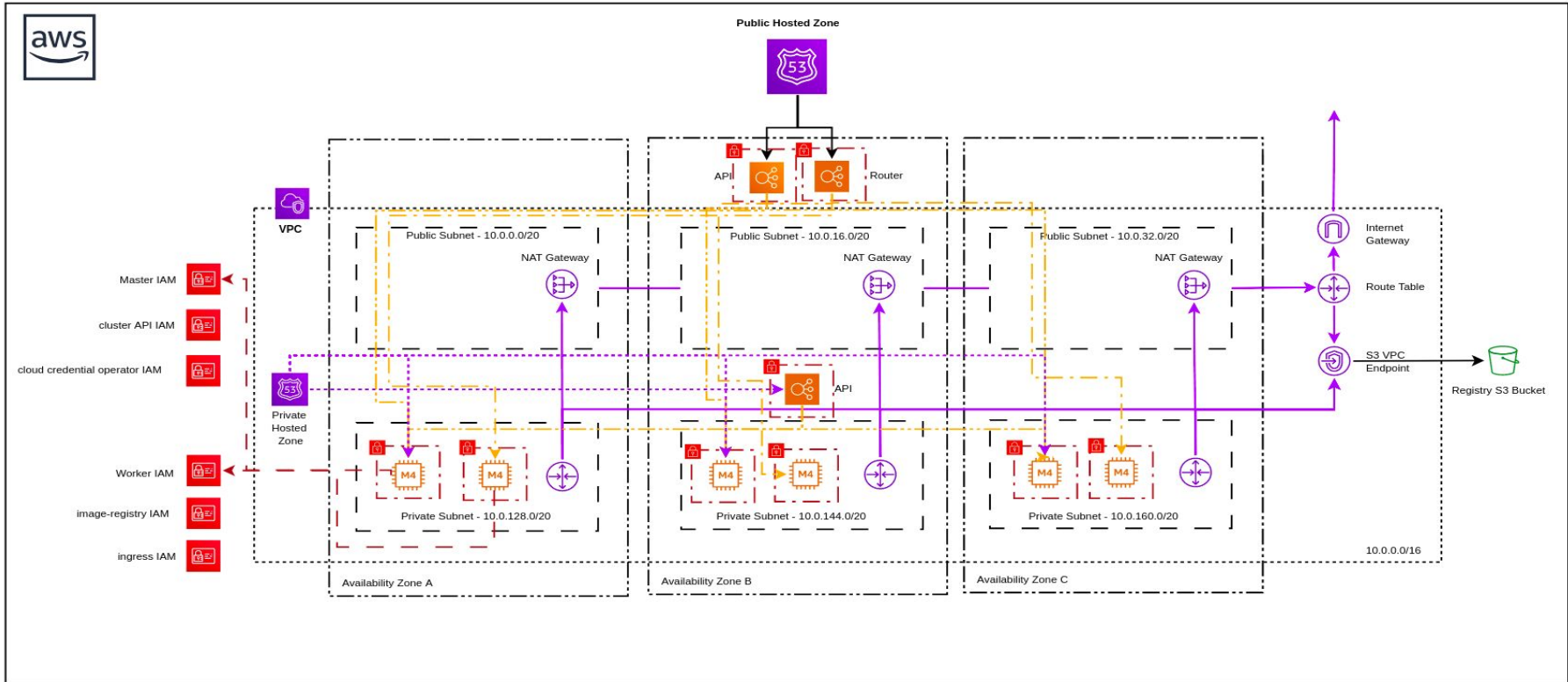
# VPC Networking Architecture

CONFIDENTIAL designator





# Cluster Architecture



# AWS specific options in install-config.yaml

- ❖ **baseDomain:** Public Hosted Zone
- ❖ **zones:** Availability Zone
- ❖ **machineNetwork:** VPC CIDR
- ❖ **platform:** AWS and Region
- ❖ **publish:** External or Internal

## Customization [ + ]

```
apiVersion: v1
baseDomain: indiashift.support
controlPlane:
  architecture: amd64
  hyperthreading: Enabled
  name: master
  platform:
    aws:
      zones:
        - us-east-1a
        - us-east-1b
      replicas: 3
...
metadata:
  creationTimestamp: null
  name: ayush
networking:
...
  machineNetwork:
    - cidr: 10.0.0.0/16
...
platform:
  aws:
    region: us-east-1
publish: External
```

# Installing on AWS

```
Activities Applications Places Terminal Fri Feb 12 11:22 AM
Terminal
DEBUG module.vpc.aws_security_group_rule.master_ingress_vxlan from worker: Still creating... [20s elapsed]
DEBUG module.vpc.aws_security_group_rule.master_ingress_internal from worker udp: Creation complete after 28s [id=sgrule-921545193]
DEBUG module.vpc.aws_security_group_rule.worker_ingress_geneve from master: Creating...
DEBUG module.vpc.aws_security_group_rule.master_ingress_kube_scheduler from worker: Still creating... [20s elapsed]
DEBUG module.vpc.aws_security_group_rule.master_ingress_internal from worker: Still creating... [10s elapsed]
DEBUG module.vpc.aws_security_group_rule.master_ingress_services_udp from worker: Still creating... [10s elapsed]
DEBUG module.vpc.aws_security_group_rule.master_ingress_geneve from worker: Still creating... [20s elapsed]
DEBUG module.vpc.aws_security_group_rule.worker_ingress_geneve from master: Creation complete after 4s [id=sgrule-3334437455]
DEBUG module.vpc.aws_security_group_rule.master_ingress_kube_controller_manager_from_worker: Creation complete after 28s [id=sgrule-2063315063]
DEBUG module.vpc.aws_security_group_rule.worker_ingress_internal: Creating...
DEBUG module.vpc.aws_security_group_rule.worker_ingress_ssh: Creating...
DEBUG module.vpc.aws_security_group_rule.master_ingress_ovnadb from worker: Still creating... [20s elapsed]
DEBUG module.vpc.aws_security_group_rule.master_ingress_kubelet_secure from worker: Still creating... [20s elapsed]
DEBUG module.vpc.aws_security_group_rule.worker_ingress_internal: Creation complete after 5s [id=sgrule-4090815038]
DEBUG module.vpc.aws_security_group_rule.master_ingress_vxlan from worker: Creation complete after 29s [id=sgrule-3874345576]
DEBUG module.vpc.aws_security_group_rule.worker_egress: Creating...
DEBUG module.vpc.aws_security_group_rule.worker_ingress_services_udp from master: Creating...
DEBUG module.vpc.aws_security_group_rule.master_ingress_services_tcp from worker: Still creating... [10s elapsed]
DEBUG module.vpc.aws_security_group_rule.master_ingress_kube_scheduler from worker: Still creating... [30s elapsed]
DEBUG module.vpc.aws_security_group_rule.master_ingress_internal from worker: Still creating... [20s elapsed]
DEBUG module.vpc.aws_security_group_rule.master_ingress_services_udp from worker: Still creating... [20s elapsed]
DEBUG module.vpc.aws_security_group_rule.master_ingress_geneve from worker: Still creating... [30s elapsed]
DEBUG module.vpc.aws_security_group_rule.worker_ingress_ssh: Creation complete after 10s [id=sgrule-3025399343]
DEBUG module.vpc.aws_security_group_rule.worker_ingress_internal from master: Creating...
DEBUG module.vpc.aws_security_group_rule.master_ingress_kube_scheduler from worker: Creation complete after 34s [id=sgrule-3128415476]
DEBUG module.vpc.aws_security_group_rule.worker_ingress_services_tcp from master: Creating...
DEBUG module.vpc.aws_security_group_rule.master_ingress_ovnadb from worker: Still creating... [30s elapsed]
DEBUG module.vpc.aws_security_group_rule.master_ingress_kubelet_secure from worker: Still creating... [30s elapsed]
DEBUG module.vpc.aws_security_group_rule.worker_egress: Creation complete after 9s [id=sgrule-3711930179]
DEBUG module.vpc.aws_security_group_rule.master_ingress_geneve from worker: Creation complete after 35s [id=sgrule-1780851536]
DEBUG module.vpc.aws_main_route_table_association.main_vpc_routes[0]: Creating...
DEBUG module.vpc.aws_security_group_rule.worker_ingress_services_tcp: Creating...
```

# AWS Resources Creation and Destruction

- ❖ **Original Logs [+]**: Proper flow of resource creation, destruction and complete install.
- ❖ **Created Resources [+]**: Only the resources created with ID and name.
- ❖ **Destroyed Resources [+]**: Destroyed bootstrap resources.

# VPC Break-Down

- ❖ **VPC:** IPv4 CIDR, DNS hostnames and resolution.
- ❖ **Subnets:** IPv4 CIDR for individual subnet per AZ.
- ❖ **Route Tables:** A route table contains a set of rules, called routes, that are used to determine where network traffic from your subnet or gateway is directed.
- ❖ **Internet Gateways:** To enable access to or from the internet for instances in a subnet in a VPC.
- ❖ **NAT Gateways:** To enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.
- ❖ **Endpoints:** A VPC endpoint enables private connections between your VPC and supported AWS services.

# EC2 Break-Down

- ❖ **Instances:** Virtual machines using RHCOS.
- ❖ **Volumes:** Root volume attached to the instances.
- ❖ **Security Groups:** A virtual firewall for your EC2 instances to control incoming and outgoing traffic.
- ❖ **Load Balancers:** Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances.
- ❖ **Target Groups:** Each target group is used to route requests to one or more registered targets. When you create each listener rule, you specify a target group and conditions.
- ❖ **AMI:** An Amazon Machine Image (AMI) provides the information required to launch an instance.

# IAM Break-Down

- ❖ **Instances IAM Roles:** Separate IAM roles attached to master and worker instances to sign the API requests with AWS credentials.
- ❖ **IAM Users for Operators:** Several IAM users get created for the operators to manage the internal, registry, machines, ingress, etc.
  - aws-ebs-csi-driver-operator
  - cloud-credential-operator-iam-ro
  - openshift-image-registry
  - openshift-ingress
  - openshift-machine-api-aws

# S3 Break-Down

- ❖ **S3 Bucket:** Storage for internal registry by default. Images gets stored inside the S3 bucket as image streams get created.

The VPC endpoint for S3 service gets created in the start as the images get pushed to the internal registry. Since the internal registry doesn't need to be accessed directly over the internet that's why endpoint is created so the images can be pushed and pulled to the internal registry over the private AWS network instead of public internet.



# Route 53 Break-Down

- ❖ **Public Hosted Zone:** This hosted zone contains the “api.<cluster\_name>.<base\_domain>” and “\*.apps.<cluster\_name>.<base\_domain>” DNS records resolving to NLB (internet-facing) and CLB respectively so the routes and API can be accessed over the public internet.
- ❖ **Private Hosted Zone:** It contains the public API and wild-card DNS records as well as the “api-int.<cluster\_name>.<base\_domain>” DNS record for internal API resolving to NLB (internal). Also, the private hosted zone gets associated with the VPC.

# AWS specific features in OpenShift

## ❖ Cloud Credential Operator **[+]**

The Cloud Credential Operator (CCO) manages cloud provider credentials as Kubernetes custom resource definitions (CRDs). The CCO syncs on credentialsRequest custom resources (CRs) to allow OpenShift Container Platform components to request cloud provider credentials with the specific permissions that are required for the cluster to run.

```
$ oc -n openshift-cloud-credential-operator get credentialsrequest cloud-credential-operator-iam-ro
```

```
$ oc -n openshift-cloud-credential-operator get secret cloud-credential-operator-iam-ro-creds -o yaml
```

Rotate the Cloud Credential Operator Credentials **[+]**

## ❖ Machine Management

- Machine API Operator
- Machine
- Machine Sets **[+]**
- Scaling a machine set manually **[+]**
- Machine set deletion policy **[+]**

Rotate the Machine API Operator Credentials **[+]**

Custom machineset with required volumesize and instance type **[+]**

## ❖ image-registry

- Credentials to access the S3 bucket.

```
$ oc -n openshift-image-registry get secret image-registry-private-configuration -o yaml
```

- S3 bucket name and its specifications.

```
$ oc get configs.imageregistry.operator.openshift.io/cluster -o yaml
```

- Cloud credentials for image-registry IAM.

```
$ oc -n openshift-image-registry get secret installer-cloud-credentials -o yaml
```

Rotate the openshift-image-registry operator credentials **[+]**

## ❖ Ingress

- By default, the installer creates an CLB for ingress which is outdated.
- NLB **[+]** for ingress is now supported.
- The load balancer service for ingress is present inside openshift-ingress namespace.

```
$ oc -n openshift-ingress get svc router-default
```

Rotate the openshift-ingress operator credentials **[+]**

## ❖ AWS EBS CSI Driver Operator [+]

- OpenShift Container Platform is capable of provisioning persistent volumes (PVs) using the Container Storage Interface (CSI) driver for AWS Elastic Block Store (EBS).
- To create CSI-provisioned PVs that mount to AWS EBS storage assets, OpenShift Container Platform installs the AWS EBS CSI Driver Operator and the AWS EBS CSI driver by default in the `openshift-cluster-csi-drivers` namespace.
- The AWS EBS CSI Driver Operator provides a StorageClass by default that you can use to create PVCs.

```
$ oc -n openshift-cloud-credential-operator get credentialsrequest aws-ebs-csi-driver-operator -o yaml
```

```
$ oc -n openshift-cluster-csi-drivers get secret ebs-cloud-credentials -o yaml
```

# How OpenShift identifies its resources on AWS?

- ❖ The OpenShift cluster can be deployed on AWS into an existing VPC.
- ❖ Cluster only manages those resources which are created by the installer itself.
- ❖ A tag with unique key gets added to the AWS resources which are created by the OpenShift installer. Cluster identifies whether the AWS resource belongs to the cluster or not on the basis of that unique tag.

Tags		Manage tags
<input type="text" value="Search tags"/>		< 1 > ⚙
Key	Value	
Name	shiftinstup-6tr8z-vpc	
kubernetes.io/cluster/shiftinstup-6tr8z	owned	

Thank you

Red Hat is the world's leading provider of  
enterprise open source software solutions.  
Award-winning support, training, and consulting  
services make  
Red Hat a trusted adviser to the Fortune 500.

 [linkedin.com/company/red-hat](https://linkedin.com/company/red-hat)

 [youtube.com/user/RedHatVideos](https://youtube.com/user/RedHatVideos)

 [facebook.com/redhatinc](https://facebook.com/redhatinc)

 [twitter.com/RedHat](https://twitter.com/RedHat)