Table of Contents

[Planning to Install 3](#_Toc165042213)

[IAM permission 3](#_Toc165042214)

[EC2 Jump-box/Boot-node 3](#_Toc165042215)

[Identify certificate 3](#_Toc165042216)

[Install pre-requisite 3](#_Toc165042217)

[Update the ocp yaml file 4](#_Toc165042218)

[Terraform variable file 4](#_Toc165042219)

[Triggers terraform 5](#_Toc165042220)

[Retrieve admin console details 5](#_Toc165042221)

WARNING

Terraform script is IBM proprietary code. Do not commit to any public platform or public URL.

# Planning to Install

Use the following information to deploy Cloud pak data services on Self-managed Openshift cluster using cp-deployer. The team should have expertise in the following areas.

1. Knowledge of Openshift and Kubernate platform, respective cli utilities.
2. Knowledge of terraform
3. Knowledge of linux and openssl commands
4. Knowledge of AWS services like VPC and its components, EC2, EBS, ELB, Security groups, EFS, Route53 services, IAM etc...

# IAM permission

To deploy CloudPak components on Self-managed Openshift cluster, the IAM user require all the permission given in following [link](https://docs.openshift.com/container-platform/4.12/installing/installing_aws/installing-aws-account.html#installation-aws-permissions_installing-aws-account).

# EC2 Jump-box/Boot-node

Create the EC2 machine with below configuration. The EC2 machine is require to perform administrative task on cluster and create the cluster.

|  |  |
| --- | --- |
| AMI id | Retrieve the AMI of dev-d01-bootnote |
| Instance type | t3.large |
| EBS volume size | 50 GB |
| EBS volume type | gp3 |

# Identify certificate

Please consider below example, it can be modified to verify other domain as well. As per the Merck’s team the certificate is not require to change in deployment script, it should work in higher environment. You can verify with appropriate Merck’s team.

|  |
| --- |
| openssl s\_client quay.io:443  openssl x509 -noout -text -in <<merck certificate.cer>> |

# Install pre-requisite

Install utilities on EC2 jumpbox/bootnode

|  |
| --- |
| Copy “cp4d-deployment-master” folder from dev-d01-bootnode EC2 instance.  cd cp4d-deployment-master/selfmanaged-openshift/aws/  rm -rf nohup.out terraform.tfstate .terraform.lock.hcl .terraform tfplan terraform.tfstate.backup .terraform.tfstate.lock.info installer-files/  ## install utility  sudo yum update -y  sudo yum install -y git yum-utils jq httpd-tools yum-plugin-copr podman python39 wget unzip  podman version  ## install terraform  sudo yum-config-manager --add-repo <https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo>  sudo yum -y install terraform  ## install aws cli  curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"  unzip awscliv2.zip  ./aws/install  ## install jq  wget <https://github.com/stedolan/jq/releases/download/jq-1.6/jq-linux64>  mv jq-linux64 jq  sudo chmod +x jq  sudo mv jq /usr/local/bin  sudo cp /usr/local/bin/jq /usr/bin/  wget -qO /usr/local/bin/yq <https://github.com/mikefarah/yq/releases/latest/download/yq_linux_amd64>  chmod a+x /usr/local/bin/yq  ## install openshift-client and oc util  export OpenshiftVersion=4.12.45  wget [https://mirror.openshift.com/pub/openshift-v4/clients/ocp/${OpenshiftVersion}/openshift-client-linux-${OpenshiftVersion}.tar.gz](https://mirror.openshift.com/pub/openshift-v4/clients/ocp/$%7bOpenshiftVersion%7d/openshift-client-linux-$%7bOpenshiftVersion%7d.tar.gz)  tar -xvf openshift-client-linux-${OpenshiftVersion}.tar.gz  chmod u+x oc kubectl  sudo mv oc /usr/local/bin  sudo cp /usr/local/bin/oc /usr/bin/  sudo mv kubectl /usr/local/bin  sudo cp /usr/local/bin/kubectl /usr/bin/  ## Download and run openshift-install command only if you get error openshift-install not found  #wget https://mirror.openshift.com/pub/openshift-v4/x86\_64/clients/ocp/${OpenshiftVersion}/openshift-install-linux-${OpenshiftVersion}.tar.gz  #tar -xvzf openshift-install-linux-${OpenshiftVersion}.tar.gz  #sudo cp openshift-install /usr/local/bin/  #sudo cp openshift-install /usr/bin/ |

# Update the ocp yaml file

Update the configuration to following file “*self-managed-openshift/aws/ocp/templates.tf*”. update following resource “*data template\_file install\_config\_yaml*”

|  |  |
| --- | --- |
| compute[0].platform.aws.amiID | Retrieve the AMI of master/worker node |
| controlPlane.platform.aws.amiID | Retrieve the AMI of master/worker node |
| additionalTrustBundle | Retrieve certificate and add its contains here |
| platform.aws.hostedZone | Z08016371AU3O7MBT731 |

# Terraform variable file

Prepare the terraform variable files with below key/value pairs in following folder “*self-managed-openshift/aws/sm-private-cpd-4-8-x.tfvars*”

|  |  |
| --- | --- |
| region | us-east-1 |
| key\_name | Existing EC2 key-pair name- Instance was not created yet |
| tenancy | Default |
| access\_key\_id | srv\_deployment\_user  Access Key ID: AKIA5FTZCXIDOXGVFUVZ |
| secret\_access\_key | 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 |
| new\_or\_existing\_vpc\_subnet | exist |
| vpc\_id | vpc-014a842ba2eb31202 |
| master\_subnet1\_id | subnet-0ac011afdf8d33ffa |
| master\_subnet2\_id | subnet-0fcd3638bb76f7f4d |
| master\_subnet3\_id | subnet-0b7e5705c82d102ea |
| worker\_subnet1\_id | subnet-0ac011afdf8d33ffa |
| worker\_subnet2\_id | subnet-0fcd3638bb76f7f4d |
| worker\_subnet3\_id | subnet-0b7e5705c82d102ea |
| enable\_permission\_quota\_check | false |
| cluster\_name | sceo-wa |
| az | multi-zone |
| availability\_zone1 | us-east-1a |
| availability\_zone2 | us-east-1b |
| availability\_zone3 | us-east-1c |
| worker\_instance\_type | m6i.4xlarge |
| worker\_instance\_volume\_iops | 3000 |
| worker\_instance\_volume\_size | 500 |
| worker\_instance\_volume\_type | gp3 |
| worker\_replica\_count | 6 |
| master\_instance\_type | m6i.2xlarge |
| master\_instance\_volume\_iops | 3000 |
| master\_instance\_volume\_size | 500 |
| master\_instance\_volume\_type | gp3 |
| master\_replica\_count | 3 |
| cluster\_network\_cidr | Cluster internal network CIDR range i.e 10.128.0.0/14 -given same has dev |
| cluster\_network\_host\_prefix | 23 - given same has dev |
| service\_network\_cidr | Cluster internal service network CIDR range i.e 172.30.0.0/16 - given same has dev |
| private\_cluster | true |
| openshift\_pull\_secret\_file\_path | Path of pull secret file. /home/ec2-user/pull-secret - pull secret I received from Dan IBM |
| public\_ssh\_key | Contents of /home/ec2-user/.ssh/id\_rsa.pub - we need to create |
| enable\_fips | false |
| base\_domain | prd.merck.com |
| openshift\_username | kubeadmin |
| openshift\_password | passw0rd |
| enable\_autoscaler | false |
| storage\_option | efs-ebs |
| efs | { "enable" : "true" } |
| accept\_cpd\_license | accept |
| cpd\_external\_registry | cp.icr.io |
| cpd\_external\_username | cp |
| cpd\_api\_key | IBM entitlement key --Received from Dan |
| cpd\_namespace | cp4d-prod-operands |
| openshift\_version | 4.14.20 |
| cpd\_platform | yes |
| cpd\_version | 4.8.4/4.8.5 |
| watson\_assistant | yes |

# Triggers terraform

Run terraform commands. Please refer the following url for terraform documentation “https://registry.terraform.io/providers/hashicorp/aws/latest”.

|  |
| --- |
| terraform init  terraform plan -out=tfplan -var-file=sm-private-cpd-4-8-x.tfvars  nohup terraform apply -input=false tfplan &  tail -f nohup.out |

# Retrieve admin console details

|  |
| --- |
| # retrieve openshift admin console url  cat nohup.out | grep web-console  # retrieve openshift credential  cat nohup.out | grep kubeadmin  # retrieve CPD admin console url  oc login -u kubeadmin -p <<openshift-password>> https://api.<<clustername>>.<<basedomain>>:6443  oc get routes -A | grep cpd-cpd | aws ‘{print $3}’  # retrieve CPD credential  cd self-managed-openshift/aws/installer/cpd/  ./cpd-cli manage login-to-ocp -u kubeadmin -p <<openshift-password>> https://api.<<clustername>>.<<basedomain>>:6443  ./cpd-cli manage get-cpd-instance-details --cpd\_instance\_ns=cpd --get\_admin\_initial\_credentials=true |