DigitalRoute

AWS (Amazon Web Services) Marketplace

Usage Engine Private Edition Installation Guide

V1.3

Overview

This document explains how to deploy DigitalRoute's Usage Engine Private Edition on EKS, when it has been purchased from the AWS Marketplace.

Installation Environment

The installation steps require the following

- Amazon Linux 2 / Ubuntu operating system
- A domain name
    - The application will create a sub-domain for accessibility which requires DNS delegation to be configured on the parent domain
- Privilege or ability to set up DNS delegation for the subdomain name

Pre-requisite Software

Please setup and install the following applications before beginning the installation steps

- AWS CLI
    - https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html
- Terraform
    - https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli
- Helm
    - https://helm.sh/docs/intro/install/
- eksctl
    - https://docs.aws.amazon.com/eks/latest/userguide/eksctl.html
- kubectl
    - https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html
- AWS IAM Authenticator
    - https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html

Minimum IAM Policy for Installation

You do not need to setup the following if the user that performs the installation has the **AdministratorAccess** policy.

For best practice we prefer to set up a minimum IAM policy for the user to perform the installation. The policies required are the following.

You can refer to AWS documentation for guidance:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html

IAM Policies required for eksctl

- https://eksctl.io/usage/minimum-iam-policies/

IAM Policies required for helm

- AmazonEC2ContainerRegistryReadOnly

IAM Policies required for terraform and helm

- AllowTerraform

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sts:GetCallerIdentity",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeInternetGateways",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeVpcs",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeSubnets",
        "rds:AddTagsToResource",
        "rds:ListTagsForResource",
        "rds:CreateDBInstance",
        "rds:CreateDBSubnetGroup",
        "rds:DeleteDBSubnetGroup",
        "rds:DeleteDBInstance",
        "rds:ModifyDBInstance",
        "ec2:Describe*",
        "rds:Describe*",
        "rds:CreateDBParameterGroup",
```

```
            "rds:ModifyDBParameterGroup",
            "rds:DeleteDBParameterGroup",
            "rds:CreateOptionGroup",
            "rds:ModifyOptionGroup",
            "rds:DeleteOptionGroup"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "route53:GetHostedZone",
            "route53:CreateHostedZone",
            "route53:ListHostedZones",
            "route53:ChangeTagsForResource",
            "route53:ChangeResourceRecordSets",
            "route53:ListResourceRecordSets",
            "route53:GetChange",
            "route53:ListTagsForResource",
            "route53:GetDNSSEC",
            "route53:DeleteHostedZone"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "acm:*"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "kms:*"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "elasticfilesystem:Describe*",
            "elasticfilesystem:DeleteAccessPoint",
            "elasticfilesystem:CreateMountTarget",
            "elasticfilesystem:CreateFileSystem",
            "elasticfilesystem:ListTagsForResource",
            "elasticfilesystem:DeleteMountTarget",
            "elasticfilesystem:CreateAccessPoint",
```

```
            "elasticfilesystem:DeleteFileSystem",
            "elasticfilesystem:TagResource",
            "elasticfilesystem:UpdateFileSystem"
          ],
          "Resource": "*"
        },
        {
          "Effect": "Allow",
          "Action": [
            "ssm:Describe*",
            "ssm:GetParameter*",
            "ssm:ListTagsForResource",
            "ssm:PutParameter",
            "ssm:DeleteParameter*",
            "ssm:AddTagsToResource"
          ],
          "Resource": "*"
        },
        {
          "Action": "iam:CreateServiceLinkedRole",
          "Effect": "Allow",
          "Resource": "arn:aws:iam::*:role/aws-service-
role/rds.amazonaws.com/AWSServiceRoleForRDS",
          "Condition": {
            "StringLike": {
              "iam:AWSServiceName": "rds.amazonaws.com"
            }
          }
        }
    ]
}
```

Setup AWS CLI configuration

Please set up your AWS CLI for the user that will perform the installation.

You can refer to AWS documentation for guidance:

https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-quickstart.html

Installation Package

1. Download the installation setup package.
   http://www.marketplace.digitalroute.net/installation-package/5.2.1/97221680-d251-42ad-b719-1310c5c5b1a3/uepe-installation-setup-package.zip

2. Unzip the package and the contents are following folders

Folder	Description
eksctl	Creates the initial EKS cluster
terraform	Creates the supporting AWS resources
3pp	Installs the 3rd party applications using helm
usage-engine-private-edition	Installs Usage Engine Private Edition using helm

Setup EKS Cluster

1. Go to the **eksctl** folder.
2. Update the **uepe-eks.yaml** in your local directory as necessary
   a. Metadata
      i. name (cluster name, default to uepe-eks)
      ii. region (default to eu-west-1)
   b. Worker Nodes
      i. instanceType (change the instant type to match your application load)
      ii. minSize, maxSize and desiredSize (if you wish to have more nodes running)

3. Execute the following command to start the creation of the EKS cluster.

```
$ cd /path-to-installation-package/eksctl
```

```
$ eksctl create cluster -f uepe-eks.yaml --kubeconfig=./kubeconfig
$ export KUBECONFIG=`pwd`/kubeconfig
```

Install Additional AWS Resources

1. Go to the **terraform** folder and copy the **terraform.tfvars.example** to **terraform.tfvars.**
2. You need to update the following parameters in the terraform.tfvars.

terraform.tfvars	Where to get the values
aws_region	Follow as per the value configured in uepe-eks.yaml Default is **eu-west-1**
cluster_name	Follow as per the value configured in uepe-eks.yaml Default is **uepe-eks**
domain	The subdomain that will be created is <cluster_name>.<domain> Assuming you wish to access the environment through the hostname, uepe-eks.example.com, the value for this parameter is **example.com** This assumes you have a parent domain, **example.com** and you have the privilege to add a NS (NameServer) record to allow DNS delegation to the subdomain uepe-eks.example.com

```
#  _____  ______ ______   ______ _    _ ______ _____  ______
# / ___|| _____|_    _| |_    _| |  | | _____/ ___|| _____|_
# \___ \|  _|    | |     | | | |_| |  _| \___ \|  _| (_)
#  ___) | |___   | |     | | |  _  | |___ ___) | |___ _
# |____/|______| |_|     |_| |_| |_|_____|____/|_____(_)

# The below values must be set explicitly for the setup to work
correctly.

aws_region = "eu-west-1"

# Route53 hosted zone will be created for <cluster_name>.<domain>
# The cluster name must match the one created using eksctl earlier
cluster_name = "uepe-eks"
domain = "example.com"
```

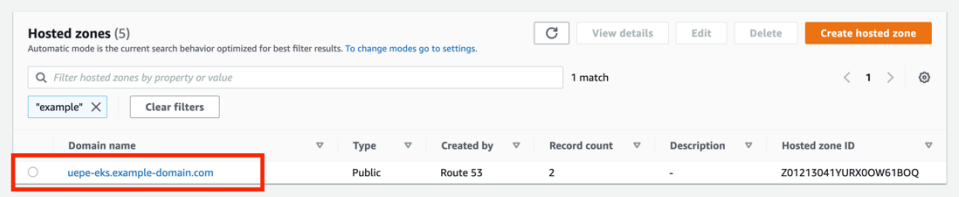
3. Execute the following commands.

```
$ cd /path-to-installation-package/terraform
```

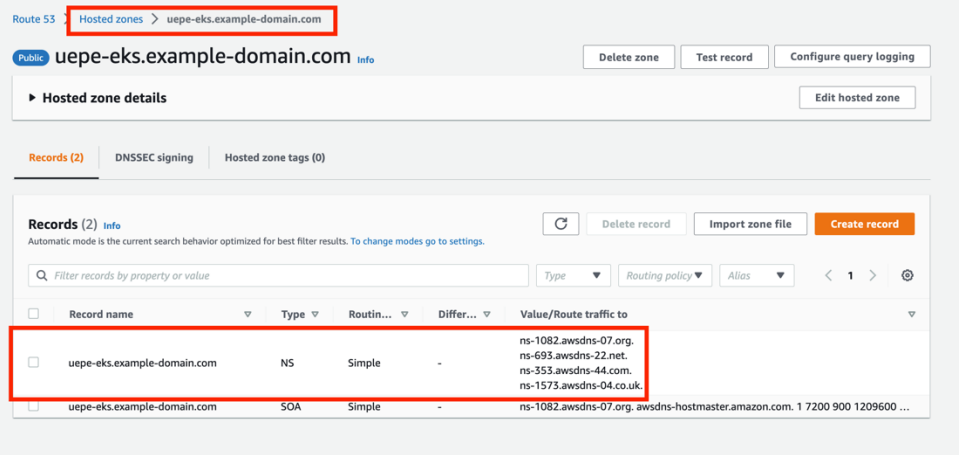
```
$ terraform init
$ terraform apply -auto-approve
```

4. The terraform will fail due to the error "timeout while waiting for ACM Certificate to be issued".
5. Retrieve the Route53 domain name servers from the Hosted Zone that is created.

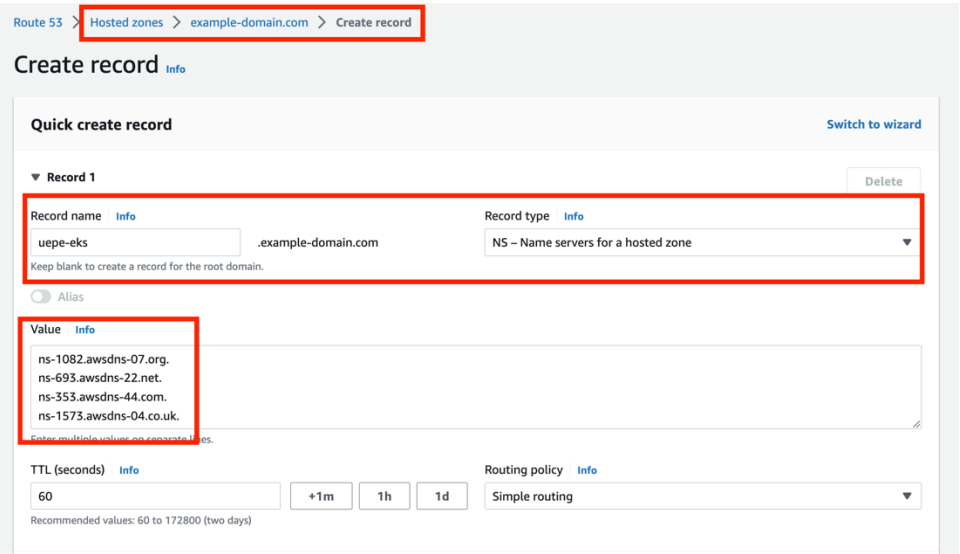
    a. Using the AWS Console, navigate to the **Route53** service.
    b. Open the Hosted Zone that was created for the domain your specified in the terraform.tfvars.



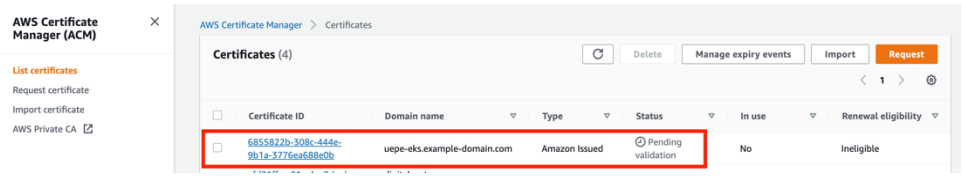
    c. Go to your cluster domain record. Copy the NS value.



    d. Note: If you are not using Route53 as the parent domain, please refer to your Domain Registrar documentation on how to add **NS record.**
    e. The steps below assume you're using Route53. On your parent domain Route53, add the NS record as per the example screenshot below.



6. Using the AWS Console, go to the **AWS Certificate Manager (ACM)** and wait for the certificate status to be verified. It will take 10-20 minutes.



7. Run the terraform command again to continue provisioning.

```
$ terraform apply -auto-approve
```

Install AWS Helpers

The application requires 3rd party applications to help with provisioning required AWS resources.

Make sure you have kubectl access to the EKS cluster.

```
$ cd /path-to-installation-package/eksctl
$ export KUBECONFIG=`pwd`/kubeconfig
```

AWS EFS CSI Driver

The AWS EFS CSI Driver will allow EFS File Systems to be mounted as the application storage

```
$ cd /path-to-installation-package/3pp
$ ./install-aws-efs-csi-driver.sh
```

AWS Load Balancer Controller

The AWS Load Balancer Controller will allow creation of the Application Load Balancer to grant Ingress access to the application

```
$ cd /path-to-installation-package/3pp
$ ./install-aws-load-balancer-controller.sh
```

External DNS

The external-dns application allows automatic configuration and synchronization for the Route53 records to allow access to the application through hostname.

```
$ cd /path-to-installation-package/3pp
$ ./install-external-dns.sh
```

Install Usage Engine Private Edition

Required information

We need 2 parameters from the terraform output

- certificate_arn
- eks_domain_zone_name

```
$ cd /path-to-installation-package/terraform
$ terraform output

certificate_arn = "arn:aws:acm:eu-west-1:123456789012:certificate/xxxx-xxxx-xxxx-xxxx"
cluster_name = "uepe-eks"
db_endpoint = "db.example.com"
db_password = "xxxxxxxx"
db_user = "dbadmin"
efs_access_point_id = "fsap-xxxxx"
efs_id = "fs-xxxxx"
eks_domain_zone_id = "xxxxx"
eks_domain_zone_name = "uepe-eks.example.com"
name_servers = tolist([
  "ns-11.awsdns-01.com",
  "ns-22.awsdns-61.org",
  "ns-33.awsdns-25.co.uk",
  "ns-44.awsdns-31.net",
])
private_subnets = tolist([
  "subnet-xxxxx",
  "subnet-xxxxx",
  "subnet-xxxxx",
])
public_subnets = tolist([
  "subnet-xxxxx",
  "subnet-xxxxx",
  "subnet-xxxxx",
])
vpc_id = "vpc-xxxxx"
```

Installing application

1. Go to the /path-to-installation-package/usage-engine-private-edition
2. Copy **uepe-helm-values.yaml.example** to **uepe-helm-values.yaml**
3. Update uepe-helm-values.yaml with the values from terraform output by replacing the <> placeholders.

```
global:
  domain: <eks_domain_zone_name>

aws:
  acm_certificate: <certificate_arn>

postgres:
  adminUsername: dbadmin
  host: db.<eks_domain_zone_name>
```

4. Run the following command to install

```
$ cd /path-to-installation-package/eksctl
$ export KUBECONFIG=`pwd`/kubeconfig

$ cd /path-to-installation-package/usage-engine-private-edition

$ aws ecr get-login-password --region us-east-1 | helm registry login --username AWS --password-stdin 709825985650.dkr.ecr.us-east-1.amazonaws.com

$ helm upgrade --install usage-engine-private-edition oci://709825985650.dkr.ecr.us-east-1.amazonaws.com/digitalroute/uepe/usage-engine-private-edition-charts --version 5.2.1 --values=./uepe-helm-values.yaml
```

5. Wait for all the pods be in **Running** state before accessing the environment.

```
$ kubectl get pod


NAME                                                READY   STATUS    RESTARTS   AGE
aws-load-balancer-controller-6b6bc498f5-ns7vw       1/1     Running   0          10m
aws-load-balancer-controller-6b6bc498f5-pf2qs       1/1     Running   0          10m
desktop-online-b68d8f647-xmht6                      1/1     Running   0          7m9s
efs-csi-controller-6c588f4544-2kscp                 3/3     Running   0          10m
efs-csi-controller-6c588f4544-djtzk                 3/3     Running   0          10m
efs-csi-node-b6wpm                                  3/3     Running   0          10m
```

```
efs-csi-node-rwg6p                                3/3     Running   0          10m
external-dns-b9fb7ddc7-8qmw8                      1/1     Running   0          10m
platform-0                                        1/1     Running   0          7m9s
uepe-operator-controller-manager-547b4b756b-qvdkk 2/2     Running   0          7m9s
uepe-operator-controller-manager-547b4b756b-r426b 2/2     Running   0          7m9s
```

Accessing Usage Engine Private Edition

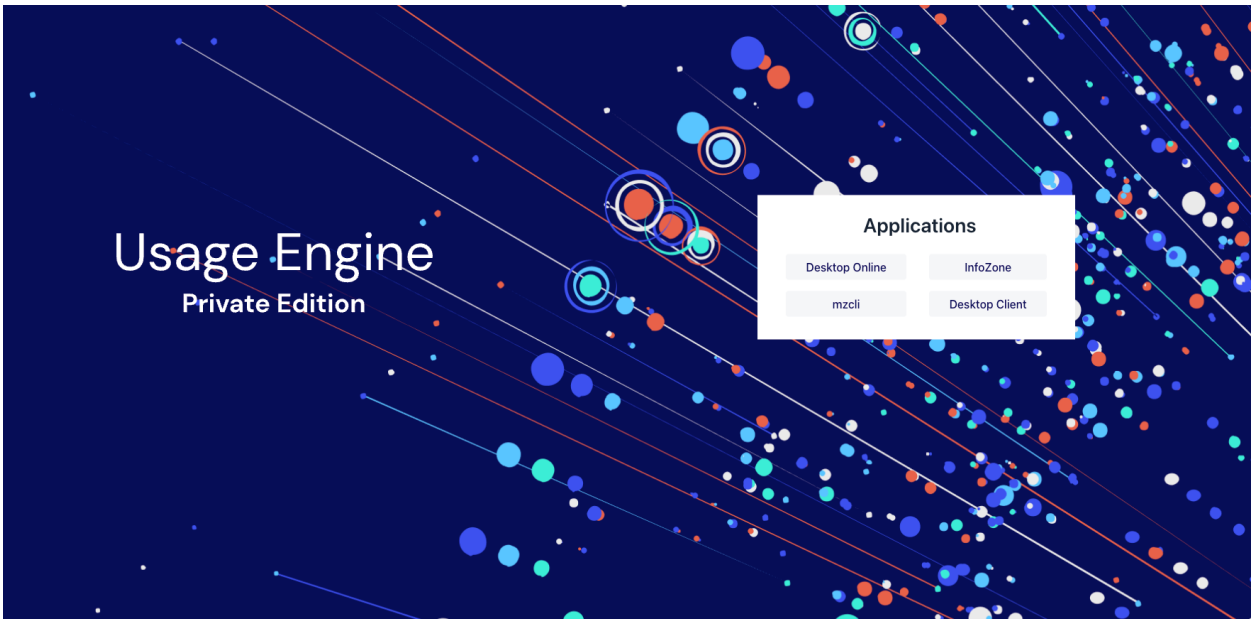
Initial password

```
$ cd /path-to-installation-package/eksctl
$ export KUBECONFIG=`pwd`/kubeconfig
$ kubectl get secret initial-password  --template={{.data.password}} | base64 -D
```

Web Portal

NOTE!, it might take some time before URL hostname is resolvable!

Using your browser, open the URL https://desktop-online.<eks_domain_zone_name>/desktop/portal



Application	Description
Infozone	Usage Engine Private Edition user documentation
Desktop Client	Java Desktop Client. Read more on Infozone.

Desktop Online	Provides UI for the application Credentials • Username: mzadmin • Password: <refer to the initial password>
mzcli	Java CLI application. Read more on Infozone.

Uninstallation

Please perform the uninstallation in sequence to allow all resources to be systematically removed.

Ensure you have access to the EKS cluster

```
$ cd /path-to-installation-package/eksctl
$ export KUBECONFIG=`pwd`/kubeconfig
```

Usage Engine Private Edition

```
$ helm uninstall usage-engine-private-edition

* Wait for 10 minutes for AWS external DNS and load balancer to cleanup.
```

AWS Helpers

```
$ helm uninstall external-dns
$ helm uninstall aws-load-balancer-controller
$ helm uninstall aws-efs-csi-driver
```

AWS Resources by Terraform

```
$ cd /path-to-installation-package/terraform
$ terraform destroy
```

AWS EKS Cluster

```
$ cd /path-to-installation-package/eksctl
$ eksctl delete cluster -f uepe-eks.yaml --disable-nodegroup-eviction
```

Version history

V1.0	2022-12-20	First version.
V1.1	2022-02-01	Updated for Usage Engine Private Edition 2.3.2
V1.2	2024-06-18	Updated for Usage Engine Private Edition 4.1.0
V1.3	2026-03-10	Updated for Usage Engine Private Edition 5.2.1