DevOps Project

```
provider "aws" {
  region = "us-east-1"
  acce =   AKIA4MTWGW7C33ESVYP3
  sect =  v6YxMMIvT+8GXtAZPasiyIjsRz4y4k4K5z+P4SmA
}

# Jenkins Master: Installs Jenkins and Java
resource "aws_instance" "jenkins_master" {
  ami           = "ami-04b4f1a9cf54c11d0"
  instance_type = "t2.micro"
  key_name = "main"
  tags = {
    Name = "Jenkins Master"
  }
  user_data = <<-EOF
              #!/bin/bash
              sudo apt-get update -y
              # Install Java
              sudo apt-get install -y openjdk-11-jdk
              # Add Jenkins repo and install Jenkins
              sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
              https://pkg.jenkins.io/debian/jenkins.io-2023.key
              echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
              https://pkg.jenkins.io/debian binary/ | sudo tee \
              /etc/apt/sources.list.d/jenkins.list > /dev/null
              sudo apt-get update
              sudo apt-get install jenkins
              EOF
}

# Worker2: Installs Docker and Kubernetes components (worker node)
resource "aws_instance" "worker2" {
  ami           = "ami-04b4f1a9cf54c11d0"
  instance_type = "t2.small"
  key_name = "main"
  tags = {
    Name = "Worker2"
  }
  user_data = <<-EOF
              #!/bin/bash
              sudo apt-get update -y
              # Install Docker
              sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installDocker.sh -P /tmp
              sudo chmod 755 /tmp/installDocker.sh
              sudo bash /tmp/installDocker.sh
```

```
            sudo bash /tmp/installCRIDockerd.sh
            sudo systemctl restart cri-docker.service
            sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installK8S.sh -P /tmp
            sudo chmod 755 /tmp/installK8S.sh
            sudo bash /tmp/installK8S.sh
            sudo modprobe br_netfilter
            echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables
            echo 1 > /proc/sys/net/ipv4/ip_forward
            EOF
}

# Kubernetes Master: Installs Java, Docker, and Kubernetes (master components)
resource "aws_instance" "kubernetes_master" {
  ami           = "ami-04b4f1a9cf54c11d0"
  instance_type = "t2.small"
  key_name = "main"
  tags = {
    Name = "Kubernetes Master"
  }
  user_data = <<-EOF
            #!/bin/bash
            sudo apt-get update -y
            # Install Java
            sudo apt-get install -y openjdk-11-jdk
            # Install Docker
            sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installDocker.sh -P /tmp
            sudo chmod 755 /tmp/installDocker.sh
            sudo bash /tmp/installDocker.sh
            sudo systemctl restart docker.service
            sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installCRIDockerd.sh -P /tmp
            sudo chmod 755 /tmp/installCRIDockerd.sh
            sudo bash /tmp/installCRIDockerd.sh
            sudo systemctl restart cri-docker.service
            sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installK8S.sh -P /tmp
            sudo chmod 755 /tmp/installK8S.sh
            sudo bash /tmp/installK8S.sh
            sudo kubeadm init --cri-socket unix:///var/run/cri-dockerd.sock --ignore-preflight-errors=all
            sudo mkdir -p $HOME/.kube
            sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
            sudo chown $(id -u):$(id -g) $HOME/.kube/config
            kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.24.1/manifests/calico.yaml
            EOF
}

# Worker4: Installs Docker and Kubernetes components (worker node)
resource "aws_instance" "worker4" {
```

```
92
93  # Worker4: Installs Docker and Kubernetes components (worker node)
94  resource "aws_instance" "worker4" {
95    ami           = "ami-04b4f1a9cf54c11d0"
96    instance_type = "t2.small"
97    tags = {
98      Name = "Worker4"
99    }
100   user_data = <<-EOF
101             #!/bin/bash
102             sudo apt-get update -y
103             # Install Docker
104             sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installDocker.sh -P /tmp
105             sudo chmod 755 /tmp/installDocker.sh
106             sudo bash /tmp/installDocker.sh
107             sudo systemctl restart docker.service
108             sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installCRIDockerd.sh -P /tmp
109             sudo chmod 755 /tmp/installCRIDockerd.sh
110             sudo bash /tmp/installCRIDockerd.sh
111             sudo systemctl restart cri-docker.service
112             sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installK8S.sh -P /tmp
113             sudo chmod 755 /tmp/installK8S.sh
114             sudo bash /tmp/installK8S.sh
115             sudo modprobe br_netfilter
116             echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables
117             echo 1 > /proc/sys/net/ipv4/ip_forward
118             EOF
119 }
120
```

```
user@user:~/project2$ nano 4server.tf
user@user:~/project2$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.88.0...
- Installed hashicorp/aws v5.88.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
user@user:~/project2$ terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions
  + create

Terraform will perform the following actions:

  # aws_instance.jenkins_master will be created
  + resource "aws_instance" "jenkins_master" {
      + ami                                  = "ami-04b4f1a9cf54c11d0"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
```

```
      + user_data_base64                      = (known after apply)
      + user_data_replace_on_change           = false
      + vpc_security_group_ids                = (known after apply)

      + capacity_reservation_specification (known after apply)

      + cpu_options (known after apply)

      + ebs_block_device (known after apply)

      + enclave_options (known after apply)

      + ephemeral_block_device (known after apply)

      + instance_market_options (known after apply)

      + maintenance_options (known after apply)

      + metadata_options (known after apply)

      + network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

Plan: 4 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.jenkins_master: Creating...
aws_instance.worker4: Creating...
aws_instance.kubernetes_master: Creating...
aws_instance.worker2: Creating...
aws_instance.jenkins_master: Still creating... [10s elapsed]
aws_instance.worker4: Still creating... [10s elapsed]
aws_instance.kubernetes_master: Still creating... [10s elapsed]
aws_instance.worker2: Still creating... [10s elapsed]
aws_instance.kubernetes_master: Creation complete after 18s [id=i-073200ede3b3c83
aws_instance.jenkins_master: Creation complete after 18s [id=i-04cd6acf21a6236d4]
aws_instance.worker4: Creation complete after 18s [id=i-0aebddf12e8c515af]
aws_instance.worker2: Creation complete after 18s [id=i-02538a3b7c0cf9f62]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
user@user:~/project2$
```

Instances (4) Info

less than

Find Instance by attribute or tag (case-sensitive) | All states

Name	Instance ID	Instance state	Instance type	Status check
Kubernetes Master	i-073200ede3b3c83ee	Running	t2.small	Initializing
Worker2	i-02538a3b7c0cf9f62	Running	t2.small	Initializing
Worker4	i-0aebddf12e8c515af	Running	t2.small	Initializing
Jenkins Master	i-04cd6acf21a6236d4	Running	t2.micro	Initializing

---

sample Public

Pin | Unwatch 1

master | 1 Branch | 0 Tags | Go to file t | Add file | Code

rabeeht2 Add files via upload — 3330e43 · now — 25 Commits

.github/workflows	Rename docker-build.yml to docker-build	5 hours ago
Dockerfile	Update Dockerfile	32 minutes ago
index.html	Update index.html	1 minute ago
pip	Create pip	27 minutes ago
sss.png	Add files via upload	now

README

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

rabeeht2

Treat username as secret ?

Password ?

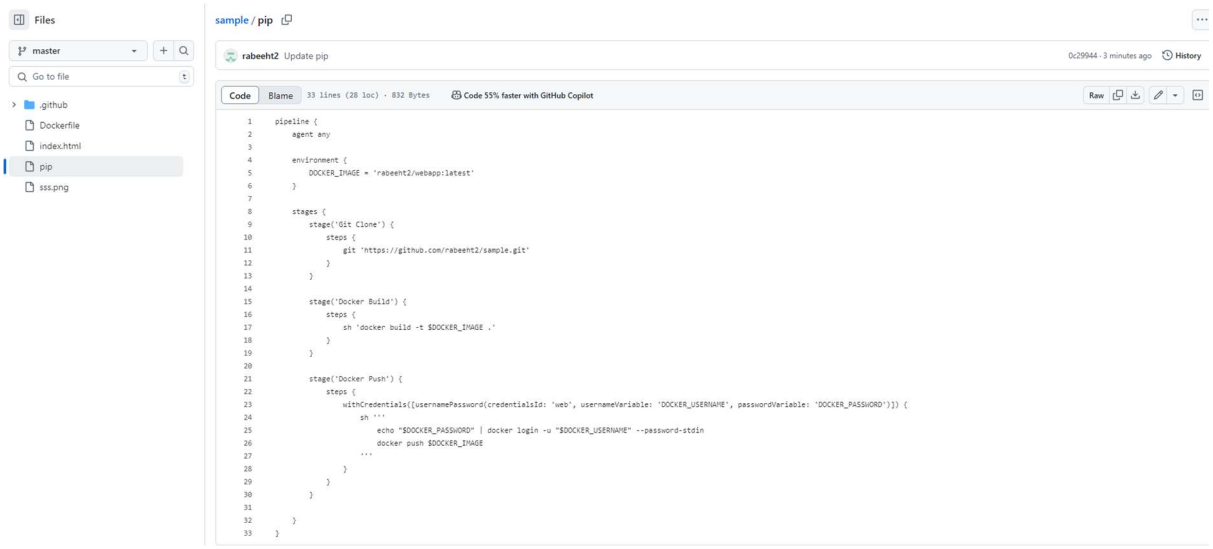
.....

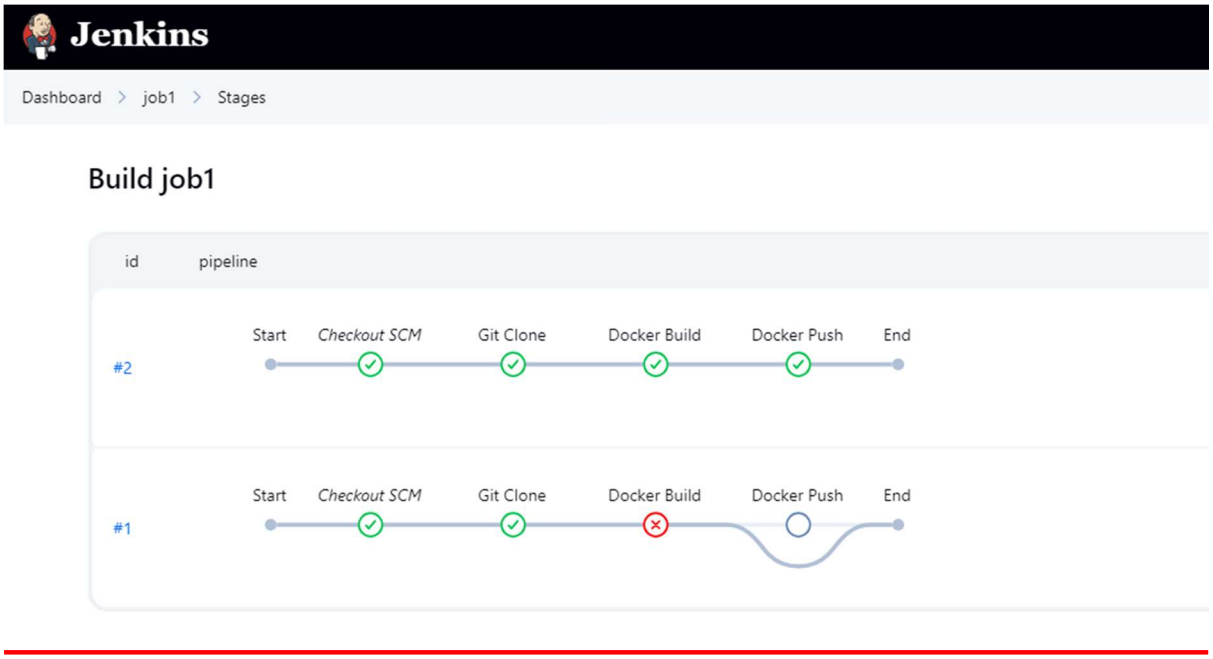
ID ?

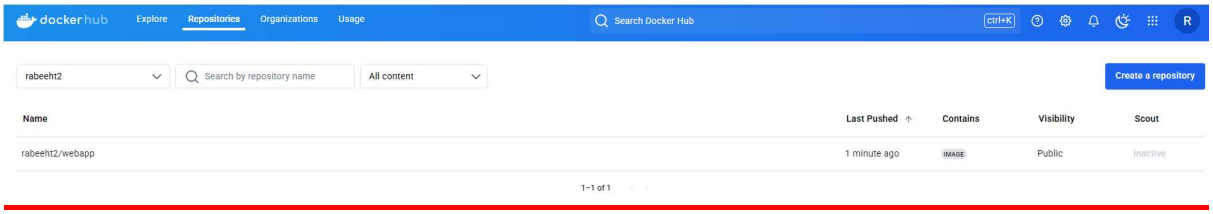
web

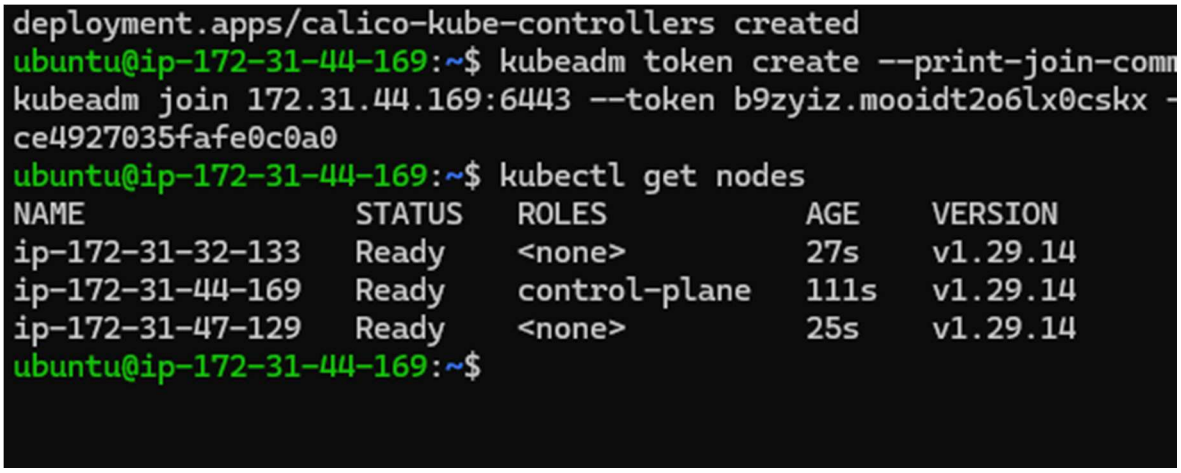
Description ?

Create


Files
master
Go to file
.github
Dockerfile
index.html
pip
sss.png
sample / pip
rabeeht2 Update pip
0c29944 · 3 minutes ago
History
Code
Blame
33 lines (28 loc) · 832 Bytes
Code 55% faster with GitHub Copilot
Raw
pipeline {
agent any
environment {
DOCKER_IMAGE = 'rabeeht2/webapp:latest'
}
stages {
stage('Git Clone') {
steps {
git 'https://github.com/rabeeht2/sample.git'
}
}
stage('Docker Build') {
steps {
sh 'docker build -t $DOCKER_IMAGE .'
}
}
stage('Docker Push') {
steps {
withCredentials([usernamePassword(credentialsId: 'web', usernameVariable: 'DOCKER_USERNAME', passwordVariable: 'DOCKER_PASSWORD')]) {
sh '''
echo "$DOCKER_PASSWORD" | docker login -u "$DOCKER_USERNAME" --password-stdin
docker push $DOCKER_IMAGE
'''
}
}
}
}
}


Jenkins
Dashboard
job1
Stages
Build job1
id
pipeline
#2
Start
Checkout SCM
Git Clone
Docker Build
Docker Push
End
#1
Start
Checkout SCM
Git Clone
Docker Build
Docker Push
End


docker hub
Explore
Repositories
Organizations
Usage
Search Docker Hub
ctrl+K
R
rabeeht2
Search by repository name
All content
Create a repository
Name
Last Pushed
Contains
Visibility
Scout
rabeeht2/webapp
1 minute ago
IMAGE
Public
Inactive
1–1 of 1


deployment.apps/calico-kube-controllers created
ubuntu@ip-172-31-44-169:~$ kubeadm token create --print-join-comm
kubeadm join 172.31.44.169:6443 --token b9zyiz.mooidt2o6lx0cskx -
ce4927035fafe0c0a0
ubuntu@ip-172-31-44-169:~$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-32-133 Ready <none> 27s v1.29.14
ip-172-31-44-169 Ready control-plane 111s v1.29.14
ip-172-31-47-129 Ready <none> 25s v1.29.14
ubuntu@ip-172-31-44-169:~$

```
  GNU nano 7.2
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webapp-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: webapp
  template:
    metadata:
      labels:
        app: webapp
    spec:
      containers:
        - name: webapp
          image: rabeeht2/webapp:latest
          ports:
            - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: webapp-service
spec:
  selector:
    app: webapp
  type: NodePort
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30008
```

```
ubuntu@ip-172-31-44-169:~$ kubectlapply -f deployment.yaml
kubectlapply: command not found
ubuntu@ip-172-31-44-169:~$ kubectl apply -f deployment.yaml
deployment.apps/webapp-deployment created
service/webapp-service created
ubuntu@ip-172-31-44-169:~$ kubectl get pods
NAME                                 READY   STATUS    RESTARTS   AGE
webapp-deployment-7bb5b45668-4lxzn   1/1     Running   0          16s
webapp-deployment-7bb5b45668-qwhz4   1/1     Running   0          16s
ubuntu@ip-172-31-44-169:~$ kubectl get pods -o wide
NAME                                 READY   STATUS    RESTARTS   AGE   IP               NODE              NOMINATED NODE   READINESS GATES
webapp-deployment-7bb5b45668-4lxzn   1/1     Running   0          25s   192.168.66.193   ip-172-31-47-129   <none>           <none>
webapp-deployment-7bb5b45668-qwhz4   1/1     Running   0          25s   192.168.33.66    ip-172-31-32-133   <none>           <none>
ubuntu@ip-172-31-44-169:~$ kubectl get service
NAME             TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes       ClusterIP   10.96.0.1       <none>        443/TCP        9m17s
webapp-service   NodePort    10.99.81.207    <none>        80:30008/TCP   47s
ubuntu@ip-172-31-44-169:~$
```


Not secure
54.211.199.231:30008
Sample world.!
code
plan
build
Dev
release
deploy
Ops
Operate
test
monitor