<https://ngj61529.live.dynatrace.com/api/v2/problems?from=now-15m&to=now&pageSize=10>

Api-Token dt0c01.SXEAMUPZCOPDGYMF4LSK2ZEU.O4YOSCOI7O2KGFYKZVGXL3UM6Z75LTF3LCSFWEKGVAJXFWTB6D3EOAPLBLNWXBIQ

**JENKINS PIPELINE**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

pipeline {

agent any

parameters {

string(name: 'WEBHOOK\_PAYLOAD', defaultValue: '{"deployment\_name": "springboot-app", "replicas": "2"}', description: 'Webhook JSON Payload')

}

stages {

stage('Parse Webhook Data') {

steps {

script {

echo "Raw WEBHOOK\_PAYLOAD: ${params.WEBHOOK\_PAYLOAD}"

def payload = readJSON text: params.WEBHOOK\_PAYLOAD

echo "Received Payload: ${payload}"

// Extract deployment name and replicas, use defaults if not provided

env.DEPLOYMENT\_NAME = payload.deployment\_name ?: 'springboot-app'

env.REPLICAS = payload.replicas ?: '2'

echo "DEPLOYMENT\_NAME: ${env.DEPLOYMENT\_NAME}"

echo "REPLICAS: ${env.REPLICAS}"

}

}

}

stage('Run Ansible Playbook') {

steps {

echo "Triggering Ansible for manual autoscaling..."

sh """

sudo -u ansible ansible-playbook /home/ansible/ansible\_scripts/autoscale.yml \

--extra-vars "deployment\_name=${env.DEPLOYMENT\_NAME} replicas=${env.REPLICAS}"

"""

}

}

}

post {

success {

echo 'Autoscaling completed successfully!'

}

failure {

echo 'Autoscaling failed. Check the logs for details.'

}

}

}

**From Postman**

[**http://35.221.23.1:8080//generic-webhook-trigger/invoke?token=webhook-token**](http://35.221.23.1:8080//generic-webhook-trigger/invoke?token=webhook-token)

**{**

**"deployment\_name": "springboot-app",**

**"replicas": "4"**

**}**

**Content-Type:** **application/json**

**Anisble SCRIPTS : deploy springboot**

**---**

**- name: Deploy Spring Boot application on Kubernetes**

**hosts: localhost**

**become: false**

**tasks:**

**- name: Ensure kubectl is installed**

**command: kubectl version --client**

**register: kubectl\_check**

**failed\_when: "'Client Version' not in kubectl\_check.stdout"**

**ignore\_errors: false**

**- name: Create deployment and service YAML**

**copy:**

**dest: /tmp/deployment.yaml**

**content: |**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: springboot-app**

**labels:**

**app: springboot-app**

**spec:**

**replicas: 1 # Updated replicas for high availability**

**selector:**

**matchLabels:**

**app: springboot-app**

**template:**

**metadata:**

**labels:**

**app: springboot-app**

**annotations:**

**metrics.dynatrace.com/scrape: "true" # Enable Dynatrace scraping**

**metrics.dynatrace.com/path: "/actuator/prometheus" # Path for Prometheus metrics**

**metrics.dynatrace.com/port: "8881" # Port for Dynatrace**

**metrics.dynatrace.com/secure: "false" # Set to false for HTTP**

**spec:**

**containers:**

**- name: springboot-app**

**image: sumanth17121988/springbootmetric:1**

**ports:**

**- name: metrics-port # Named port for Prometheus**

**containerPort: 8881 # Application container port**

**resources:**

**limits:**

**cpu: "500m"**

**memory: "256Mi"**

**requests:**

**cpu: "250m"**

**memory: "128Mi"**

**---**

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: springboot-app-service**

**labels:**

**prometheus: monitored # Label to match ServiceMonitor**

**spec:**

**selector:**

**app: springboot-app**

**ports:**

**- name: metrics-port # Named port for Prometheus**

**protocol: TCP**

**port: 8881 # Exposed service port**

**targetPort: 8881 # Container port**

**type: LoadBalancer # Expose the service externally**

**- name: Apply deployment.yaml to Kubernetes**

**command: kubectl apply -f /tmp/deployment.yaml**

**register: apply\_result**

**changed\_when: "'configured' in apply\_result.stdout or 'created' in apply\_result.stdout"**

**failed\_when: "'error' in apply\_result.stderr"**

**- name: Verify the deployment rollout status**

**command: kubectl rollout status deployment/springboot-app**

**register: rollout\_status**

**failed\_when: "'successfully rolled out' not in rollout\_status.stdout"**

**- name: Get the LoadBalancer IP**

**shell: |**

**kubectl get svc springboot-app-service -o jsonpath='{.status.loadBalancer.ingress[0].ip}'**

**register: lb\_ip**

**changed\_when: false**

**- name: Display the LoadBalancer IP**

**debug:**

**msg: "Spring Boot app is available at: http://{{ lb\_ip.stdout }}:8881"**

**Anisble script : Autoscale**

**---**

**- name: Manual Autoscaling and Update for Kubernetes Deployment**

**hosts: localhost**

**tasks:**

**- name: Set default variables if not provided**

**set\_fact:**

**deployment\_name: "{{ deployment\_name | default('springboot-app') }}"**

**replicas: "{{ replicas | default(3) }}"**

**image: "{{ image | default('sumanth17121988/springbootmetric:1') }}"**

**- name: Ensure kubectl is installed**

**command: kubectl version --client**

**register: kubectl\_check**

**failed\_when: "'Client Version' not in kubectl\_check.stdout"**

**- name: Debug variables**

**debug:**

**msg:**

**- "Deployment Name: {{ deployment\_name }}"**

**- "Image: {{ image }}"**

**- "Replicas: {{ replicas }}"**

**- name: Update the deployment (image or configuration)**

**command: >**

**kubectl set image deployment/{{ deployment\_name }}**

**{{ deployment\_name }}={{ image }}**

**register: update\_result**

**failed\_when: "'error' in update\_result.stderr.lower()"**

**- name: Scale the Deployment replicas**

**command: >**

**kubectl scale deployment/{{ deployment\_name }}**

**--replicas={{ replicas }}**

**register: scale\_result**

**failed\_when: "'error' in scale\_result.stderr.lower()"**

**- name: Verify scaling status**

**command: >**

**kubectl get deployment {{ deployment\_name }}**

**-o jsonpath='{.status.readyReplicas}'**

**register: ready\_replicas**

**until: ready\_replicas.stdout == replicas | string**

**retries: 10**

**delay: 5**

**- name: Display scaling result**

**debug:**

**msg: "Deployment {{ deployment\_name }} successfully updated and scaled to {{ replicas }} replicas.**