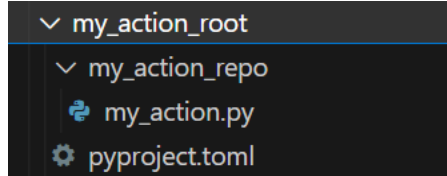


```
git clone https://github.com/robusta-dev/robusta.git
```

```
helm install robusta ./robusta/helm/robusta -f ./generated_values.yaml --set
clusterName=lizhutest --namespace robusta
```

## 2. Robusta 自定义 playbooks 配置

- a. 创建自定义 action 的 playbooks, 该 action 是当 node 状态变为 unknown 的时候发送 alert 并且展示该 node 上运行的 pod



- b. 为自定义 playbooks 创建 persistent volume

```
accessModes:
  - ReadWriteOnce
capacity:
  storage: 4Gi
local:
  path: /home/lizhu/data1
nodeAffinity:
  required:
    nodeSelectorTerms:
      - matchExpressions:
          - key: pv
            operator: In
            values:
              - worker1
persistentVolumeReclaimPolicy: Retain
volumeMode: Filesystem
```

- c. 编辑 values.yaml, 设置 playbooksPersistentVolume: true

```
helm get values robusta -o yaml > values.yaml
echo "playbooksPersistentVolume: true" >> values.yaml
helm upgrade robusta robusta/robusta --values=values.yaml

helm upgrade robusta ./robusta/helm/robusta/ --namespace=robusta --
values ./values.yaml
```

- d. Push 自定义 playbooks 到 Robusta

```
robusta playbooks push ./my_action_root
```

- e. 编辑 values.yaml 文件,使自定义的 action 得以生效

```
customPlaybooks:
  - actions:
```

```
- my_node_health_watcher: {}
triggers:
- on_node_update: {}
```

### 3. 集成 Prometheus 和 Robusta

- a. 编辑 alertManager 配置文件， account\_id 和 signing\_key 可以从 values.yaml 文件的 global config 中获得

```
receivers:
  - name: 'robusta'
    webhook_configs:
      - url: 'http://10.0.4.17:30080/api/alerts'
    http_config:
      authorization:
        # Replace <TOKEN> with a string in the format `<ACCOUNT_ID>
        <SIGNING_KEY>`
        credentials: '8b01ca9a-437b-41b6-adf9-11d6cc0b7c97 8b01ca9a-437b-
        41b6-adf9-11d6cc0b7c97'
        send_resolved: true
    route:
      receiver: 'robusta'
      routes:
        - receiver: 'robusta'
          group_by: [ '...' ]
          group_wait: 1s
          group_interval: 1s
          matchers:
            - severity =~ ".*"
          repeat_interval: 4h
          continue: true
```

- b. 编辑 Prometheus 的 alert 相关配置，添加 label: cluster\_name: lizhutest

```
groups:
  - name: example
    rules:
      # Alert for any instance that is unreachable for >2 minutes.
      - alert: service_down
        expr: up == 0
        for: 2m
        labels:
          severity: page
          cluster_name: lizhutest
        annotations:
          summary: "Instance {{ $labels.instance }} down"
```

```

description: "{{ $labels.instance }}" of job "{{ $labels.job }}" has been down
for more than 2 minutes."
- alert: high_load
  expr: node_load1 > 0.5
  for: 2m
  labels:
    severity: page
    cluster_name: lizhutest
  annotations:
    summary: "Instance {{ $labels.instance }}" under high load"
    description: "{{ $labels.instance }}" of job "{{ $labels.job }}" is under high
load."

```

- c. 编辑 values.yaml 文件，使 Robusta 可以从 Prometheus 中获取所需要的信息

```

disableCloudRouting: false
globalConfig:
  prometheus_auth: Bearer <YOUR TOKEN> # or any other auth header
  alertmanager_auth: Basic <USER:PASSWORD base64-encoded>
  alertmanager_url: ""
  grafana_url: ""
  prometheus_url: http://ip:9090
# Additional query string parameters to be appended to the Prometheus connection
URL (optional)
  prometheus_url_query_string: "demo-query=example-data&another-query=value"

```

4. 测试 alert manager 是否正常运行

```

customPlaybooks:
- actions:
  - my_node_health_watcher: {}
  triggers:
  - on_node_update: {}
- actions:
  - stack_overflow_enricher: {}
  triggers:
  - on_prometheus_alert: {}

```

```

helm upgrade robusta robusta/robusta --values=./values.yaml
helm upgrade robusta ./robusta/helm/robusta/ --namespace=robusta --values ./values.yaml

robusta demo-alert --alertmanager-url http://10.0.4.20:9093 --labels cluster_name=lizhutest

```

5. 配置 Prometheus alert playbooks  
6. Tips

- a. Robusta 的 runner 和 forwarder 不能运行在同一个 node 上，否则 k8s alert 不能发送到配置的 sink 内。但是 prometheus 的 alert 发送没有问题。
- b. 给 robusta-runner 服务配置 ingress，使 alert 能访问到 robusta 的 runner
- c. Robusta 默认会监测 pod 是否 crashloopbackoff，imagepullbackoff 和 oom kill，所以如果不需要一些默认提醒可以在 chart 的 values.yaml 内将该 trigger 注释掉。