

Jenkins

1. Install Jenkins with Helm v3

brew install helm

Configure Helm

helm repo add jenkinsci https://charts.jenkins.io

helm repo update

Create a persistent volume

kubectl apply -f jenkins-volume.yaml

reference: <https://raw.githubusercontent.com/installing-jenkins-on-kubernetes/jenkins-volume.yaml>

Create a service account

...

kubectl apply -f jenkins-sa.yaml

...

reference: <https://raw.githubusercontent.com/installing-jenkins-on-kubernetes/jenkins-sa.yaml>

Install Jenkins

jenkins-values.yaml.

reference: raw.githubusercontent.com/jenkinsci/helm-charts/main/charts/jenkins/values.yaml i

change all serviceType to LoadBalancer

```
# For minikube, set this to NodePort, elsewhere uses LoadBalancer
# Use ClusterIP if your setup includes ingress controller
# -- k8s service type
serviceType: LoadBalancer
```

...

chart=jenkinsci/jenkins

helm install jenkins -n jenkins -f jenkins-values.yaml \$chart

...

● (base) ll@LantingHou course-project-option-1-Lantinghh % kubectl get pods -n jenkins

NAME	READY	STATUS	RESTARTS	AGE
jenkins-0	2/2	Running	2 (7h36m ago)	21h

Get your 'admin' user password by running:

...

```
jsonpath="{.data.jenkins-admin-password}"
secret=$(kubectl get secret -n jenkins jenkins -o jsonpath=$jsonpath)
echo $(echo $secret | base64 --decode)
```

...

```
(base) ll@LantingHou course-project-option-1-Lantinghh % jsonpath="{.data.jenkins-admin-password}"
secret=$(kubectl get secret -n jenkins jenkins -o jsonpath=$jsonpath)
echo $(echo $secret | base64 --decode)
```

hXjJe9ehwRhjP8ONt89SNM

login

username: admin

password: hXjJe9ehwRhjP8ONt89SNM

Link to SonarQube and GitHub


click new Item

Dashboard > All >


Enter an item name

test1


» Required field

 **Freestyle project**

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

 **Pipeline**

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

 **Folder**

configure as picture

Create your own token , here i use hello

Configure

General

Advanced Project Options

Pipeline

☐ Preserve stashes from completed builds ?

☐ This project is parameterized ?

☐ Throttle builds ?

Build Triggers

☐ Build after other projects are built ?

☐ Build periodically ?

☐ Poll SCM ?

☐ Quiet period ?

☒ Trigger builds remotely (e.g., from scripts) ?

Authentication Token

hello

Use the following URL to trigger build remotely: JENKINS_URL/job/test1/build?token=TOKEN_NAME or /buildWithParameters?token=TOKEN_NAME
Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.

link to Github, In the pipeline part , choose pipeline script from SCM, and choose SCM Git . The repository is <https://github.com/Lantinghh/test>

Configure

General

Advanced Project Options

Pipeline

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

<https://github.com/Lantinghh/test>

Credentials ?

admin/*****

+ Add

Advanced

the branch is main

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Apply Configuration

reference

<https://www.jenkins.io/doc/book/installing/kubernetes/>

Github part:

Jenkinsfile

The screenshot shows the GitHub interface for the repository 'Lantinghh / test'. The left sidebar displays the file structure with 'Jenkinsfile' selected. The main content area shows the 'test / Jenkinsfile' file, which is 11 lines long. The code is a Jenkinsfile script using Groovy syntax to define a pipeline with stages for SCM checkout and SonarQube analysis.

```
1 node {
2   stage('SCM') {
3     checkout scm
4   }
5   stage('SonarQube Analysis') {
6     def scannerHome = tool 'SonarScanner';
7     withSonarQubeEnv() {
8       sh "${scannerHome}/bin/sonar-scanner"
9     }
10  }
11 }
```

creat sonar-project.properties

The screenshot shows the GitHub interface for the repository 'Lantinghh / test', specifically the 'sonar-project.properties' file. The left sidebar shows the file structure with 'sonar-project.properties' selected. The main content area shows the 'test / sonar-project.properties' file, which is 1 line long. The content of the file is a single line of text: 'sonar.projectKey=hello'.

```
1 sonar.projectKey=hello
```

settings: webHook

in payload part, the IP is Jenkins IP; the token is hello (corresponding with the above we made in Jenkins)

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

Autolink references

SettingsRecent Deliveries

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *
http://34.23.252.182:8080/job/test/build?token=hello

Content type
application/x-www-form-urlencoded

Secret

SSL verification
By default, we verify SSL certificates when delivering payloads.
☒ Enable SSL verification ☐ Disable (not recommended)

Which events would you like to trigger this webhook?
☐ Just the push event.
☐ Send me everything.
☒ Let me select individual events.

☒ **Active**
We will deliver event details when this hook is triggered.

Update webhookDelete webhook

Apply Configuration

Made Tools

Jenkins

Search (#+K)

Jenkins Admin

log out

DashboardManage Jenkins

+ New Item

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

Build Executor Status

Manage Jenkins

System Configuration

System

Tools

Plugins

Configuration as Code

Clouds

Appearance

SonarScanner for MSBuild installations

SonarScanner for MSBuild installations ^ Edited

Add SonarScanner for MSBuild

≡ SonarScanner for MSBuild

Name

SonarScanner

☒ Install automatically ?

≡ Install from GitHub

Version

SonarScanner for MSBuild 5.15.1.88158 - .NET Fwk 4.6

Add Installer ▾

Add SonarScanner for MSBuild

SonarQube Scanner installations

SonarQube Scanner installations ^ Edited

Add SonarQube Scanner

≡ SonarQube Scanner

Name

SonarScanner

☒ Install automatically ?

≡ Install from Maven Central

Version


SonarQube Scanner 5.0.1.3006

Add Installer ▾

Add SonarQube Scanner

Apply configuration

Build the item we made

 Status

 test

 Changes

 Build Now


 Configure


 Delete Pipeline


 SonarQube

 Rename


 Pipeline Syntax

 Build History


trend 


 Filter...

/


 #9


Apr 4, 2024, 6:10 PM




 #8

Apr 4, 2024, 6:06 PM



 #7

Apr 4, 2024, 5:42 PM



SonarQube Quality Gate

hello Passed

server-side processing: Success

Permalinks

- [Last build \(#9\), 6 hr 36 min ago](#)
- [Last stable build \(#9\), 6 hr 36 min ago](#)
- [Last successful build \(#9\), 6 hr 36 min ago](#)
- [Last failed build \(#5\), 7 hr 10 min ago](#)
- [Last unsuccessful build \(#6\), 7 hr 5 min ago](#)
- [Last completed build \(#9\), 6 hr 36 min ago](#)

SonarQube

1. Installing SonarQube using Helm 3

brew install helm

2. Apply Node Taints

kubectl get nodes

Apply taint to one of the selected nodes

```
kubectl taint nodes gke-hadoop-cluster-default-pool-2abfa0aa-9cr3
sonarqube=true:NoSchedule
```

Mark the nodes that apply the taint

```
kubectl label node gke-hadoop-cluster-default-pool-2abfa0aa-9cr3 sonarqube=true
```

Modify the SonarQube service type to load balancer:

```
service:
  type: LoadBalancer
  externalPort: 9000
  internalPort: 9000
  labels:
  annotations: {}
```

```
● (base) ll@lantinghou api-hadoop7 % kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
gke-hadoop-cluster-default-pool-2abfa0aa-9cr3    Ready    <none>    14h   v1.27.8-gke.1067004
gke-hadoop-cluster-default-pool-2abfa0aa-g624    Ready    <none>    14h   v1.27.8-gke.1067004
gke-hadoop-cluster-default-pool-2abfa0aa-n6zw    Ready    <none>    14h   v1.27.8-gke.1067004
(base) ll@lantinghou api-hadoop7 % kubectl taint nodes gke-hadoop-cluster-default-pool-2abfa0aa-9cr3 sonarqube=true:NoSchedule
node/gke-hadoop-cluster-default-pool-2abfa0aa-9cr3 tainted
(base) ll@lantinghou api-hadoop7 % kubectl label node gke-hadoop-cluster-default-pool-2abfa0aa-9cr3 sonarqube=true
node/gke-hadoop-cluster-default-pool-2abfa0aa-9cr3 labeled
(base) ll@lantinghou api-hadoop7 % helm repo add sonarqube https://SonarSource.github.io/helm-chart-sonarqube
helm repo update
kubectl create namespace sonarqube-lts
helm upgrade --install -n sonarqube-lts sonarqube sonarqube/sonarqube-lts
"sonarqube" has been added to your repositories
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "apache-hadoop-helm" chart repository
...Successfully got an update from the "sonarqube" chart repository
...Successfully got an update from the "cloudnativeapp" chart repository
```

3. Install Helm Chart

```
helm repo add sonarqube https://SonarSource.github.io/helm-chart-sonarqube
```

```
helm repo update
```

```
kubectl create namespace sonarqube
```

```
helm upgrade --install -n sonarqube sonarqube sonarqube/sonarqube -f values.yaml
```


4.kubectl get external IP

Managed pods

Name	Status	Restarts	Created on ↑
sonarqube-sonarqube-0	✔ Running	0	Apr 3, 2024, 4:03:09 PM

Exposing services ⓘ

Name ↑	Type	Endpoints
sonarqube-sonarqube	Load balancer	34.139.35.121:9000 ↗

http://34.139.35.121:9000

OVERVIEW OBSERVABILITY COST OPTIMIZATION						
☰ Filter Is system object : False ✕ Filter workloads						
<input type="checkbox"/> Name ↑	Status	Type	Pods	Namespace	Cluster	
<input type="checkbox"/> hadoop-hadoop-hdfs-dn	✔ OK	Stateful Set	2/2	default	hadoop-cluster	
<input type="checkbox"/> hadoop-hadoop-hdfs-nn	✔ OK	Stateful Set	1/1	default	hadoop-cluster	
<input type="checkbox"/> hadoop-hadoop-yarn-nm	✔ OK	Stateful Set	1/1	default	hadoop-cluster	
<input type="checkbox"/> hadoop-hadoop-yarn-rm	✔ OK	Stateful Set	1/1	default	hadoop-cluster	
<input type="checkbox"/> sonarqube-postgresql	✔ OK	Stateful Set	1/1	sonarqube	hadoop-cluster	
<input type="checkbox"/> sonarqube-sonarqube	✔ OK	Stateful Set	1/1	sonarqube	hadoop-cluster	

← → ↻ ⚠ Not Secure 34.139.35.121:9000/projects/create ☆ | 📄 | ⬇️ | ⓘ

a

useful

2

3

1

company

14848-certification

All Bookmarks

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More 🔍

ⓘ A

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)?
Create your project from your favorite DevOps platform.

First, you need to set up a DevOps platform configuration.

🔗 Import from Azure DevOps

Setup

🔗 Import from Bitbucket Cloud

Setup

🔗 Import from Bitbucket Server

Setup

🔗 Import from GitHub

Setup

🔗 Import from GitLab

Setup

Are you just testing or have an advanced use-case? Create a local project.

Create a local project

Reference

<https://dev.to/lakkimartin/install-sonarqube-on-kubernetes-aks-49o7>
<https://docs.sonarsource.com/sonarqube/8.9/setup-and-upgrade/deploy-sonarqube-on-kubernetes/>
<https://github.com/SonarSource/helm-chart-sonarqube/tree/master/charts/sonarqube>

