jenkins script using splunk metrics around stage events can you assist wht is prerequisite

Jenkins Script with Splunk Metrics for Stage Events

To create a Jenkins script that sends stage event metrics to Splunk, there are several prerequisites you'll need to set up. Here's what you'll need:

Prerequisites

1. Splunk Setup

* **Splunk instance** with HTTP Event Collector (HEC) enabled
* **HEC token** created in Splunk with appropriate permissions
* **Splunk endpoint URL** (typically https://your-splunk-server:8088/services/collector)

2. Jenkins Setup

* **Jenkins instance** with admin privileges to install plugins
* **Jenkins plugins**:
  + Pipeline Utility Steps (for JSON handling)
  + HTTP Request Plugin (for sending data to Splunk)
  + (Optional) Splunk Jenkins Plugin if you prefer a dedicated integration

3. Network Configuration

* Network connectivity between Jenkins and Splunk servers
* Firewall rules allowing Jenkins to send data to Splunk HEC port (usually 8088)

Sample Jenkins Pipeline Script

Here's a basic example of how to send stage metrics to Splunk:

groovy

Copy

Download

pipeline {

agent any

environment {

SPLUNK\_HEC\_URL = 'https://your-splunk-server:8088/services/collector'

SPLUNK\_HEC\_TOKEN = 'your-hec-token-here'

}

stages {

stage('Build') {

steps {

script {

def startTime = System.currentTimeMillis()

*// Your build steps here*

echo "Building..."

sleep 2 *// Simulating build work*

sendStageMetricsToSplunk('Build', startTime, System.currentTimeMillis(), 'SUCCESS')

}

}

}

stage('Test') {

steps {

script {

def startTime = System.currentTimeMillis()

*// Your test steps here*

echo "Testing..."

sleep 1 *// Simulating test work*

sendStageMetricsToSplunk('Test', startTime, System.currentTimeMillis(), 'SUCCESS')

}

}

}

}

}

def sendStageMetricsToSplunk(String stageName, long startTime, long endTime, String status) {

def duration = endTime - startTime

def event = [

event: [

build\_number: env.BUILD\_NUMBER,

job\_name: env.JOB\_NAME,

stage\_name: stageName,

duration\_ms: duration,

duration\_sec: duration / 1000,

status: status,

start\_time: new Date(startTime).format("yyyy-MM-dd'T'HH:mm:ss.SSSZ"),

end\_time: new Date(endTime).format("yyyy-MM-dd'T'HH:mm:ss.SSSZ"),

node: env.NODE\_NAME,

executor: env.EXECUTOR\_NUMBER

],

source: "jenkins",

sourcetype: "jenkins:stage:metrics"

]

def jsonEvent = writeJSON returnText: true, json: event

httpRequest(

url: env.SPLUNK\_HEC\_URL,

contentType: 'APPLICATION\_JSON',

customHeaders: [[name: 'Authorization', value: "Splunk ${env.SPLUNK\_HEC\_TOKEN}"]],

httpMode: 'POST',

requestBody: jsonEvent,

quiet: true

)

}

Additional Recommendations

1. **Error Handling**: Add try-catch blocks around your stage execution to properly capture failures
2. **Common Function**: Create a shared library for the Splunk sending function to reuse across pipelines
3. **Data Enrichment**: Consider adding more context like:
   * Git commit information
   * Build cause
   * Environment variables
4. **Performance**: For high-volume Jenkins instances, consider batching events

Would you like me to elaborate on any specific aspect of this setup?

splunk token they are using with credentials plugin

Using Splunk Token with Jenkins Credentials Plugin

When integrating Jenkins with Splunk's HTTP Event Collector (HEC), it's a security best practice to store the Splunk token using Jenkins' Credentials Plugin rather than hardcoding it in your pipeline.

Prerequisites for Using Credentials

1. **Credentials Plugin installed**: Ensure you have the Jenkins "Credentials Binding Plugin" installed
2. **Splunk HEC token**: Your Splunk administrator should provide you with the HEC token

How to Store the Splunk Token in Jenkins Credentials

1. **Add the credential to Jenkins**:
   * Go to Jenkins → Manage Jenkins → Manage Credentials
   * Under the appropriate domain (usually "Global"), click "Add Credentials"
   * Select "Secret text" as the kind of credential
   * For the "Secret" field, paste your Splunk HEC token
   * Give it an ID like splunk-hec-token and a meaningful description
   * Click OK to save

Modified Pipeline Using Credentials

Here's how to modify the previous pipeline to use the stored credential:

groovy

Copy

Download

pipeline {

agent any

environment {

SPLUNK\_HEC\_URL = 'https://your-splunk-server:8088/services/collector'

}

stages {

stage('Build') {

steps {

script {

def startTime = System.currentTimeMillis()

*// Your build steps here*

echo "Building..."

sleep 2 *// Simulating build work*

withCredentials([string(credentialsId: 'splunk-hec-token', variable: 'SPLUNK\_HEC\_TOKEN')]) {

sendStageMetricsToSplunk('Build', startTime, System.currentTimeMillis(), 'SUCCESS')

}

}

}

}

stage('Test') {

steps {

script {

def startTime = System.currentTimeMillis()

*// Your test steps here*

echo "Testing..."

sleep 1 *// Simulating test work*

withCredentials([string(credentialsId: 'splunk-hec-token', variable: 'SPLUNK\_HEC\_TOKEN')]) {

sendStageMetricsToSplunk('Test', startTime, System.currentTimeMillis(), 'SUCCESS')

}

}

}

}

}

}

def sendStageMetricsToSplunk(String stageName, long startTime, long endTime, String status) {

def duration = endTime - startTime

def event = [

event: [

build\_number: env.BUILD\_NUMBER,

job\_name: env.JOB\_NAME,

stage\_name: stageName,

duration\_ms: duration,

duration\_sec: duration / 1000,

status: status,

start\_time: new Date(startTime).format("yyyy-MM-dd'T'HH:mm:ss.SSSZ"),

end\_time: new Date(endTime).format("yyyy-MM-dd'T'HH:mm:ss.SSSZ"),

node: env.NODE\_NAME,

executor: env.EXECUTOR\_NUMBER

],

source: "jenkins",

sourcetype: "jenkins:stage:metrics"

]

def jsonEvent = writeJSON returnText: true, json: event

httpRequest(

url: env.SPLUNK\_HEC\_URL,

contentType: 'APPLICATION\_JSON',

customHeaders: [[name: 'Authorization', value: "Splunk ${env.SPLUNK\_HEC\_TOKEN}"]],

httpMode: 'POST',

requestBody: jsonEvent,

quiet: true

)

}

Alternative Approach: Using Credentials Throughout the Pipeline

If you prefer to make the token available throughout the entire pipeline:

groovy

Copy

Download

pipeline {

agent any

environment {

SPLUNK\_HEC\_URL = 'https://your-splunk-server:8088/services/collector'

*// This will make SPLUNK\_HEC\_TOKEN available in all steps*

SPLUNK\_HEC\_TOKEN = credentials('splunk-hec-token')

}

stages {

*// Your stages here*

}

}

Security Considerations

1. **Credential ID**: Use a meaningful ID that clearly identifies the purpose of the credential
2. **Access Control**: Restrict who can view/use the credential in Jenkins
3. **Token Permissions**: In Splunk, configure the HEC token with minimal required permissions
4. **Rotation**: Implement a process for rotating the HEC token periodically

Would you like me to provide more details about any specific aspect of this credential management approach?