Monitoring for Jenkins pipelines

We have wide variety of choices for monitoring Jenkins pipelines. For example **splunk**, **ELK stack**, **AWS cloudwatch**, **datadog** and so on.

Each of these option has its own advantages. But when it comes to choosing one, it basically depends on how the Jenkins is setup, on-prem or cloud, distributed or clustered etc.

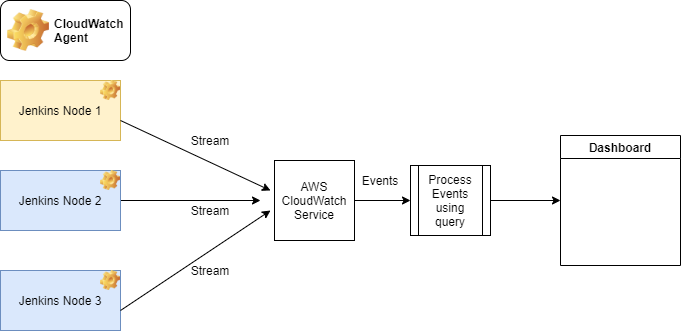
If Jenkins is setup on cloud then we have option to use cloud provider’s native monitoring services. For example, On AWS we can use AWS Cloudwatch to monitor Jenkins and all the running pipelines in it.

AWS Cloudwatch for monitoring Jenkins pipelines:-

Let’s assume we have Jenkins (master/slave) setup on AWS.

In order to monitor Jenkins we have to install and configure cloudwatch agent on the master and slave nodes. The agent should be configured such a way that it can monitor Jenkins logs. The log streams are ingested by the cloudwatch logs service in real time. We can use CloudWatch Logs Insights to interactively search and analyze log data. The logs coming from all the Jenkins nodes will get aggregated into a single flow of consistent events. These events can be sorted and grouped based on specific fields using queries, in this case it can be Jenkins build information, such as build status, timestamp, errors etc. The processed data can be now visualized using CloudWatch dashboards.

**Note – For clustered and containerized Jenkins setup**, we can use cloudWatch as well, just we have to configure agents as the container in the same cluster and configure it to monitor the logs from all the active Jenkins containers.



Advantages:-

1. Security – CloudWatch inherits all the security layers from the AWS. Though IAM roles and policies, access management can be controlled.
2. Scalability and reliability – CloudWatch has auto scalability, it can scale based on the log ingestion. It is reliable as the AWS has multiple AZs.
3. Performance – It is fast and has high throughput.

Disadvantages:-

1. The logs generated by Jenkins is very high level, we will not get fine-grain details regarding builds such as which stage build has failed or succeeded, what is the reason of failure, some build information like build number/tags, artefacts generated etc.
2. We cannot create super customized dashboards.

**NOTE: - In order to overcome the above disadvantages, we can have the below solutions:-**

1. To visualize the fine-grain details related to build, we can use appropriate plugin in Jenkins which will stream build logs to the cloudWatch. Like cloudWatch plugin.
2. We can generate custom log files in the each of the pipelines which will have the same format. Ex. TIMESTAMP | APPLICATION\_NAME | STAGE\_NAME | STATUS | LOG\_LEVEL | ERROR | ……………

This log file from all the CI/CD pipelines can be ingested by the cloudwatch service, which then can queried to generate dashboards.

AWS Cloudwatch and Amazon Elasticsearch Service for monitoring Jenkins pipelines:-

Let’s assume we have Jenkins (master/slave) setup on AWS.

In this approach we can use cloudWatch design from the previous section, the only difference is that we will be separating the data processing layer from CloudWatch to Elasticsearch.

The AWS Elasticsearch Service is fully managed clustered service. It’s highly scalable and can index huge load of data. It provides API services which can be easily integrated with open source tools like Kibana and logstash.

The events from the cloudWatch can be published to the Elasticsearch service where these events will get processed and indexed asynchronously in the data store managed by Elasticsearch.

Then kibana can be used to visualize data and generate dashboards.

Advantages:-

1. Highly scalable, reliable, and secure as it’s managed by the AWS.
2. We will be able to control minute details, and create highly customized dashboards.

Disadvantages: -

1. Not very efficient for small scale setup.
2. Costly.

