All About DataDog

Overview of Datadog Log Management:
Log Management does / provides the following:

- It allows us to cost effectively collect, process, archive, explore and monitor all the logs without any limitations. [1]
- It has the log explorer which allows us to assess & fix issues. [1]
- It allows logs to be stored and archived. [1]
- DataDog Cloud SIEM detects security threats in our environment, allowing us to index logs. [1]
- Log Management ingests logs from hosts, containers, cloud providers, and other sources. [1]
- Once the logs are ingested, it is possible to generate metrics from them. [1]

Log Collection and Integration:
To send logs from a NodeJS, do the following:

- Create a DataDog Account. [2]
- Install the DataDog Agent. [2]
- In the **datadog.yaml** file, change **logs_enabled** to **true**. [2]
- Follow the instructions to set up a logger in the application to generate and send logs. [2]

DataDog provides several endpoints, some of which are encrypted. They are found here: [2]

Custom Log Forwarding:

- We can also use a custom process or library to forward logs to DataDog when used along with DataDog Logs. [2]
- This method of sending logs to DataDog isn't recommended as it doesn't use a DataDog Agent. Using a DataDog Agent allows us to have a native connection. [3]
- If we use it with Winston in this mode, then we need to configure the DataDog Transport for Winston. [3]

Sending Logs From NodeJS to DataDog Using Winston:

- On the DataDog platform, in our account, we need to install the **NodeJS** Integration. An integration is a pre-built connector that allows Datadog to collect, visualize, and analyze data from various services, tools, or platforms. [5]
- On our local system, we need to install the DataDog Agent for our NodeJS Application. [Ubuntu / Linux based] [2]
- We need to activate log collection in the datadog agent's **datadog.yaml** file. [6]
- Create a **nodejs.d/** folder in the **conf.d/** Agent configuration directory and a **conf.yaml** file for it. [3]
- Restart the agent. [3]
- Use Winston normally with Morgan. [3]

Using Winston and NodeJS with DataDog - Best Practices & Others:

- Create a custom logger with Winston. [4]
- It is a best practice to create a different logger for each major component of the application. This allows us to know where the logs come from. This is achieved by creating different loggers for different application services. [4]
- A good idea would be to place all the loggers in a **logger.js** file and then import them as necessary. [4]
- It is preferable to configure transports to log to files in the local storage. This ensures that we always have a copy of our logs stored locally and this also prevents us from losing visibility due to network issues. [4]
- In our log messages, we can include metadata such as User ID, Request Endpoint, IP Address, etc. This allows us to effectively pinpoint issues in our application. [4]
- If necessary, the **defaultMeta** parameter in Winston can be used to add global metadata to every log that we create. [4]
- All the information from these logs will be displayed in Log Explorer. [4]

Pricing:

- Pro -> Starting at $15 / host / month
- Enterprise -> $ 23 / host / month
- DevSecOps Pro -> $ 22 / host / month
- DevSecOps Enterprise -> $ 34 / host / month

REFERENCES:

[1] https://docs.datadoghq.com/logs/
[2] https://docs.datadoghq.com/logs/log_collection/?tab=application
[3] https://docs.datadoghq.com/logs/log_collection/nodejs/?tab=winston30
[4] https://www.datadoghq.com/blog/node-logging-best-practices/
[5] https://docs.datadoghq.com/integrations/node/
[6] https://docs.datadoghq.com/agent/logs/?tab=tailfiles#activate-log-collection