**Fluentd with GreyLog**

**What is a Fluentd?**

Fluentd is an open source data collector, which lets you unify the data collection and consumption for a better use and understanding of data.

Fluentd is a fully free and fully open-source log collector that instantly enables you to have a ‘Log Everything’ architecture with 600+ types of systems. If you need a secure, stable, and enterprise grade version Fluentd Enterprise is available.

**Why, Filter?**

Fluentd treats logs as JSON, a popular machine-readable format. It is written primarily in C with a thin-Ruby wrapper that gives users flexibility. Fluentd’s scalability has been proven in the field: its largest user currently collects logs from 500,000+ servers.

Fluentd decouples data sources from backend systems by providing a unified logging layer in between. This layer allows developers and data analysts to utilize many types of logs as they are generated. Just as importantly, it mitigates the risk of "bad data" slowing down and misinforming your organization.

A unified logging layer lets you and your organization make better use of data and iterate more quickly on your software.

Key Features of Fluentd:

* **Unified Logging with JSON :-** Fluentd tries to structure data as JSON as much as possible: this allows Fluentd to unify all facets of processing log data: collecting, filtering, buffering, and outputting logs across multiple sources and destinations.
* **Pluggable Architecture :-** Fluentd has a flexible plugin system that allows the community to extend its functionality.
* **Minimum Resources Required :-** Fluentd is written in a combination of C language and Ruby, and requires very little system resource. The vanilla instance runs on 30-40MB of memory and can process 13,000 events/second/core.
* **Built-in Reliability :-** Fluentd supports memory- and file-based buffering to prevent inter-node data loss. Fluentd also support robust failover and can be set up for high availability.

**Fluentd ?**

How to use fluentd in application having logs collection requirement:

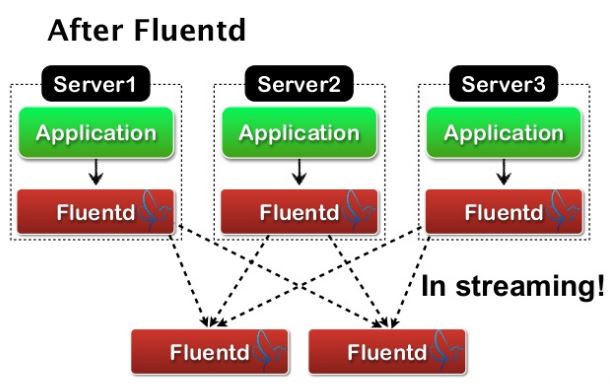
1. **Fluentd Architecture**
2. **Installing Fluentd**
3. **Required Jars & Configuration**
4. **Connecting with Spring Application**
5. **Fluentd with GrayLog2**
6. **Fluentd Architecture**

Below is the fluentd architecture. Before fluentd, application running into multiple servers used to connet to single log server, which may inturn lead to high latency situation. It caused the application to become very slow in terms logging sometimes even a day long.

Fluentd acts as a unified logging layer among all the applications running into different servers. It can get logs in JSON format to store it in-memory storage or database [nosql db].

Fluentd upon receiving logs buffer and flush to destined storage such as cloud, file, mongoDB, Redis, etc… after a fix interval of time. Fluentd supports a rich set of plugins to store logs in many numbers of storage.

A screenshot of a cell phone

Description generated with very high confidence****

1. **Installing Fluentd**

Fluentd is written in Ruby for flexibility, with performance sensitive parts written in C. To install fluentd in windows, download **td-agent**. Installing tg-agent in windows will create a folder structure in C:\ drive as follows. Installing tg-agent will install **Td-agent Command Prompt.** User this to interact with fluentd version of windows.

Local Diskk (C:)

|- opt

More on to install fluentd in windows click: <https://docs.fluentd.org/v0.14/articles/install-by-msi>

|- tg-agent

|- bin

|- conf

1. **Required Jars & Configuration**

In order to use fluentd in Spring context, we need fluentd logger dependecy. Along with it, spring required logback appenders to work with fluentd plugin as follows:

compile 'org.fluentd:fluent-logger:0.3.2'

compile 'com.sndyuk:logback-more-appenders:1.1.1'

To connect with fluentd server, fluentd server need to be started before application starts logging events. To able to connect to this server, application need to configured as well. Fluentd server can be started to listen to some port and tag, which is configured in a configuration file.

This configuration varies baed on storage. A sample configuration looks like:

**Save with file name as [td-agent.conf] // td-agent is not fixed**

<source>

@type forward

</source>

<match test.\*\*>

@type stdout

</match>

This configuration can be saved to location as **C:/opt/td-agent/conf/td-agent.conf**. Now, it can be loaded by fluentd using **Td-agent Command Prompt** application and supplying command to run the configuraion as:

fluentd -c conf\td-agent.conf

**fluent-plugin:**

Note : Before running configuration, we need install plugin for the specified configuration with below command, [ replace xyz with plugin name].

To get the all plugins list : <https://github.com/fluent/fluent-plugins>

C:\opt\td-agent> fluent-gem install fluent-plugin-xyz --version=1.2.3

![A screenshot of a cell phone

Description generated with very high confidence](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4REGRXhpZgAATU0AKgAAAAgABAE7AAIAAAAUAAAISodpAAQAAAABAAAIXpydAAEAAAAoAAAQ1uocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEFua3VzaCBLdW1hciBWZXJtYW4AAAWQAwACAAAAFAAAEKyQBAACAAAAFAAAEMCSkQACAAAAAzExAACSkgACAAAAAzExAADqHAAHAAAIDAAACKAAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDE3OjA5OjAxIDE5OjM2OjE0ADIwMTc6MDk6MDEgMTk6MzY6MTQAAABBAG4AawB1AHMAaAAgAEsAdQBtAGEAcgAgAFYAZQByAG0AYQBuAAAA/+ELJmh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMTctMDktMDFUMTk6MzY6MTQuMTEwPC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPkFua3VzaCBLdW1hciBWZXJtYW48L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgA2gJYAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A9W0b4e6ZPoNo5vtTWO4ihnNuLnMUbArINiEEKAwGAPp0re/4RWP/AKCl/wD+Qv8A43V3w/8A8i1pn/XpF/6AK851343aH4f8QX2k6prVpaXVpO0bQtp1xIVAPy5ZTg5XB49aiMbxQ29Tt/8AhE4twb+07/IGBzF/8R7Uv/CKx/8AQUv/AM4v/jdeZy/tE+HQx8nX9OcY4L6bdrzg+x77fzPpg+sw6jO3hyDUVs5LyeS3SX7NbFFZ2IBIXzGVR17sPrTcUgvcpf8ACKx/9BS//OL/AON0f8IrH/0FL/8AOL/43R4Z8UN4lFwy6HqWnRW8skBkvGt8NIjlHUCOVzkMDyQAexNX9I1e31q2nntUkRYLqa1YSAAl4pGjYjBPGVOPb0o5UF2UP+EVj/6Cl/8AnF/8bo/4RWP/AKCl/wDnF/8AG63qKOVBcwf+EVj/AOgpf/nF/wDG6P8AhFY/+gpf/nF/8breqqmpWj6tLpiy5vIYEuHi2niN2ZVOcY5KNxnPFHKguZf/AAisf/QUv/zi/wDjdH/CKx/9BS//ADi/+N07UPE62upSWGnaVqGsXMAVrlbERAW4bldzSuikkc7VJbGCQARnbRg8auAQGAIDAg/iD0o5UFzC/wCEVj/6Cl/+cX/xuj/hFY/+gpf/AJxf/G63qKOVBcwf+EVj/wCgpf8A5xf/ABuj/hFY/wDoKX/5xf8Axut6ijlQXMH/AIRWP/oKX/5xf/G6P+EVj/6Cl/8AnF/8breJwCa56XxlZJ4ItvE8VneT210sBht41QTMZnVEX5nCg5cZy2PejlQaj/8AhFY/+gpf/nF/8bo/4RWP/oKX/wCcX/xurWha9FrkNzi0ubG5s5jBc2l2E8yF9oYZ2MykFWVgQxGD9RWpRyr+mFzB/wCEVj/6Cl/+cX/xug+FIjjOp3/HT/Vcf+Q63qKOVBcwf+EVj/6Cl/8AnF/8bo/4RWP/AKCl/wDnF/8AG61LXUrS9vLy1tpd81jIsVwu0jYxRXAyRg/KynjPWrVHKguYP/CKx/8AQUv/AM4v/jdH/CKx/wDQUv8A84v/AI3W9RRyoLmD/wAIrH/0FL/84v8A43R/wisf/QUv/wA4v/jdb1FHKguYP/CKx/8AQUv/AM4v/jdH/CKx/wDQUv8A84v/AI3W9RRyoLmD/wAIrH/0FL/84v8A43R/wisf/QUv/wA4v/jdb1FHKguYP/CKx/8AQUv/AM4v/jdH/CKx/wDQUv8A84v/AI3W9RRyoLmD/wAIrH/0FL/84v8A43Sf8InFuLf2nf5IweYv/iPet+ijlQXMH/hFY/8AoKX/AOcX/wAbo/4RWP8A6Cl/+cX/AMbreoo5UFzB/wCEVj/6Cl/+cX/xuj/hFY/+gpf/AJxf/G63qKOVBcwf+EVj/wCgpf8A5xf/ABuj/hFY/wDoKX/5xf8Axut6ijlQXMH/AIRWP/oKX/5xf/G6P+EVj/6Cl/8AnF/8breoo5UFzmZfA1lNIZJdQv2Y9Tuj/wDiKZ/wgOn/APP9f/8Afcf/AMRXU0UcqC5y3/CA6f8A8/1//wB9x/8AxFW/+EUizn+07/Pr+6/+N1vUUcqC5g/8IrH/ANBS/wDzi/8AjdH/AAisf/QUv/zi/wDjdb1FHKguYP8Awisf/QUv/wA4v/jdH/CKx/8AQUv/AM4v/jdb1FHKguYJ8KRHrqd+f+/X/wAbo/4RWP8A6Cl/+cX/AMbreoo5UFzB/wCEVj/6Cl/+cX/xuj/hFY/+gpf/AJxf/G63qKOVBcwf+EVj/wCgpf8A5xf/ABuj/hFY/wDoKX/5xf8Axut6ijlQXMH/AIRWP/oKX/5xf/G6P+EVj/6Cl/8AnF/8breoo5UFzB/4RWP/AKCl/wDnF/8AG6P+EVj/AOgpf/nF/wDG63qKOVBcwf8AhFY/+gpf/nF/8bpP+ETi3Bv7Tv8AIGBzF/8AEe1b9FHKguc7/wAIZab3dr67ZnOWLLC2eAO8foKP+ENtP+fy5/79wf8AxuuirN8R6v8A8I/4V1XWfI+0f2dZTXfk79nmeWhfbuwcZxjODRyoLmf/AMIbaf8AP5c/9+4P/jdH/CG2n/P5c/8AfuD/AON037d43/6F7w//AOD6f/5DrS8Oav8A8JB4V0rWfI+z/wBo2UN35O/f5fmIH27sDOM4zgUcqC5n/wDCG2n/AD+XP/fuD/43R/whtp/z+XP/AH7g/wDjddFRRyoLnO/8Ibaf8/lz/wB+4P8A43R/whtp/wA/lz/37g/+N10VZuuapPo9gt7Dp8t9DHIPtKwHMkcWDmREwTIQcZUYJGcZICscqC5n/wDCG2n/AD+XP/fuD/43UsfhaOFdsOpXsa5zhVhH/tOkfxXa3N/p9loATVprxUuGaGUeXBbMf9c7gEcgHYvVyOMAMy71HKguYo8OY6atf/8AkH/43RW1RRyoLmf4f/5FrTP+vSL/ANAFfL3xInvR4q8XWNtpmtHz76dibGzEkGoh4Y40WZiCAIHRnT5XO522mJvnr6h8P/8AItaZ/wBekX/oArQoj8KB7nw9rNvJ4i0i0iu/Detadd6f9tW3a20vzFliklE0ETgGNFw8txl0jUAFMIeg+0tB/wCRb03/AK9Iv/QBV+iqEedXXhnUf+EfSG90Q6tbL4hu7250oSxf6VbySTGM4dhG2C8b7HI+76gCtDwd4Z1Ow0SKOa5vNC8rUrq4+wW7W8ySxPOzorsyOcbSB8rKeT3xjtaKFp/Xp/kOT5t/63/zPJv+EQ8SXHivW9RgsG026vLK/tzcxG1ghkZ2X7OytEPPY7VyWlJKtnaADWafD1xqNz4l0vw34fbw+40zThLpsc1uGcCeV5E+UvEC6Aj5shsjeMMa9omhiuYJILiNJYpFKPG6hldSMEEHqCO1VNK0TStCtWttD0yz02B33tFZ26wqzYA3EKAM4A59qFordv8Ag/5hf8/8v8jzzSfh5BcXOj2+o6DdPpFtFfCW01ZbLCPIYCgEVsfK2ko7AAcMCTg4NWPBfhC70fxLpeo6voCPdtoNtaTah+5d4J494fe27eSyFF3LuyFwSABXpNFNO39ev+Yulv63ucnt1fw54i1Wa00K41iy1WZbpXs5oEkgkEaRsrrK6ZXCKQyljywIGBnD1TwtqN54tvLmXQRcXtzeWs9jrgmj/wCJbCix+ZECWEo5WU7UUq/mfMRlsekUUlpbyDv5nmMfw5Et5a3V5o0Ms8viC8mvnkdWMljIZyqNz80ZJiPldMnJXOaWw+HrT3+l2etaPDNo1i2rIltK6NEkctwjW67M8rsBIGCF2joQK9Noo6W/raxTk27/ANdf8/wR5ZpvgvWdF03TLvTdIt/7Xbw/c22pNO6P9quSIvKWY7v3vKuASSAMjIBqtovw9vPs9tZXujSrpX9vrePaXn2RMQfY2jbdFb4iwZCAUUHcDk5ya9cop31v/W9/zJ6W9fxVjy5fB97aQafb3nhn+19Ls7vUFt9OWeILbCSfdbTBXYLhUyBg70DfKvUDRPhfWH+C+j+H1ie31S3SwWURyRs0JjmiZ2DNlSVCk85Bx0PSvQKKE7fh+A763OG1XwDFN/ZMEiSa2j6z9t1We/MZadfs8kYLqAqFRmNdqrjA6dTVDTPCl9B4sSeTQvKvo9UuLqbxCJYv9JtX37IOG804Vo12MoRfLyCcLn0iilv/AF6f5A9Vb+uv+Z5ZoHwvt4YvDcWraBbSRjSZYdYSZkkEs48ryhIMnzCuJNp5CdBjisR/Dd/c3Mekan4dk1fVYvClnbqXuY8WVxvnVZmZnHKkffTc4529TXt1RC1t1u2ulgjFw6CNpgg3sgJIUt1IBYkD3PrTvf8Ar1/zHzb/ANdV/kcj4T8M/wBg+MvEFzJo0SNftFKmqxxxDzh5UayIcHzAxkV3ORtOc5JNdnRRRe5FgooopDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAxNQ8Jadqd/Jd3NzrCSyY3Lba3eW8YwAOI45VUdOwGTz1NWdI0G00TzvsU2oS+dt3fbdSuLvGM42+c7bevOMZ4z0FaVFABRRRQAVieNdPutX8A+INN0+Lzru80y5ggj3Bd7vEyqMkgDJI5JxW3RQBzf/AAlGr/8AQieIP+/+n/8AyVVnwVp91pHgHw/puoReTd2emW0E8e4NsdIlVhkEg4IPIOK26KACiiigArN1yXVo7BY9At45byeQRCWdh5dspBJlcZBYDHCrySQMqMsNKigDi9G8JXHgi/WTw7vv7PUbgPq0UzqJGmY4a7Q8D/fjGBgZQAja/aUUUAFFFFAHwhr3j7xjZ+JNStbPxZrkFvDdyxxRRalMqRqHICqA2AABgAVn/wDCx/G//Q5eIP8AwaT/APxVZ/if/kbtY/6/p/8A0Yay6mPwob3Ok/4WP43/AOhy8Qf+DSf/AOKo/wCFj+N/+hy8Qf8Ag0n/APiq5uiqEdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VR/wsfxv/wBDl4g/8Gk//wAVXN0UAdJ/wsfxv/0OXiD/AMGk/wD8VX3lJI4kbDN1Pevzpr798UaRc65ZCwt71rOCWfF20f33hw2UX0JO0fTPXoQDViuRPCksE3mRyKGR0fKsDyCD3FFYfh3w/J4dmvba3umk0uRlktLdzlrYndvUE9V+6Rz698klAHxvrvhqSbWtRv73UrHTYrrULhbYXbPunxIwJARGwAeNzbRnvwccwlldSQzzRW8kkNtjzpY1LJHk4GWHAyenrXbeLZdJ16/2XGrQ6Zd6bcXNtLHPBKwlTz5HVkKK2T85BDbeQOa5WzaJdH1Rf7YmtS3l+XZCNtt7838RB2jb97nPtUx2GMl0rb4dg1aGXzUadredNmPJcAFec8hhnHA+6R2plrouqX1lNeWWm3lzawZ82eGBnSPAydzAYGBzzWj4ZuYJEv8ARr6aOG31GHCSSsFWOdPmjYk8AZypPo5rZ0nV9LH/AAjd9cauLBtDDCWzEUjPMRI0mYyqlPnDBTuZencYq/6/r+upP9f1/XQ5e20HV7y0a6tNKvp7dULtNFbOyBRkElgMY4PPsav2/gnxDd6Amr2ulXc9u8mxRHbyMzDbneMLjZ2znrXV2+oWtppfhLVbvVVsrWzluLoaeglZ5cTsdqALs5Hy5ZhxXN2l1Yaj4SudNudQt9NuBqH2xPPjkKOpQqVBjRiCDjqAOetS9n/XUrr/AF5/18zKl0rb4dg1aGXzUadredNmPJcAFec8hhnHA+6R2q9b+CfEN3oCava6Vdz27ybFEdvIzMNud4wuNnbOetHhm5gkS/0a+mjht9RhwkkrBVjnT5o2JPAGcqT6Oak0+awvPBs2k3WpQafcR3wuka4jkZJV8sqQDGrEEEDqAOetN7O3l/wRLdEHhnwpqPifUoYbS1uvsrSiOa8itmkjgz3YjgcepFYrDaxHocV3nh7V9DV/DF1qGriyTRi4ltPIkd3cyFhIpVSuCGUHJBGzgHiuSg1G1hs5oJNIsrqSQttupXmEiZHGAsgXjqMqffND3BDLXRtTv7Oa7sdNu7m2twTNNDAzpHgZO5gMDjnmn6f4f1nV4Wm0rSL6+iRtrPbWzyKD1wSoPNdj4N1jw3pUOjXV9dQLc2ssjXH2gXTyxZPy+SsZEW3HJ3c/e4PAOnoSQR6P4XG+GPV2S4/s5Li4lRJDLIyAlEhcMPq6Z6HgA02B5zBomq3NtcXNtpl5NBakieWO3dlhIGTuIGFx70ttoOr3lo11aaVfT26oXaaK2dkCjIJLAYxwefY12ek+IbCw0jSEzpFtqeitMhGoi8bLFy25PIJRgQdp3D+HuKqaX4msYLjwgJrtorfTpZZLqNVcpFulY9Mc5Ujpnjil6BsjGt/BPiG70BNXtdKu57d5NiiO3kZmG3O8YXGztnPWss6ZeDSV1PyGNm0ph85SCFcDO04+6ccjOM84zg1s2M+n3vg+fSrjUoNPnjvxdRm4jkZJFMZUgGNGIIIHUAc9aPD2s2fh7Tbm5LG+u7tWtzp0in7P5ePvy/3+T8qrjBGSRgAj3f8AXQO39f1/SKmk6Na3WlXmp6pey2lnbSRw/uLcTSPI4YgBS6ADCMSSfTg06DRrG78O6nqNtf3BmsChMElqqq6PIEU7xISDzkjb+JrR0/xlqVj4IutPt9d1C3uRdwm2jiuJF2RBZN4Ug4UZKcZ549Kbof8AZq+FNYgvNdsra51BIxHDLHcMylJQx3FYiOQOME++KHuw6HPR6feytbrFaTubokW4WJj52Dg7ePm544qS20fU72znu7PTru4trfPnTRQMyRYGTuYDA455rsPDuraJGnhi61HVltH0aWQTW5t5HeQNJuVlKrtx83OSCMHAPAqXR/E+mwaLozRPpEGoaSZsf2it6SSzlgyeQSpBBwQw7dx0GBxttoOr3lo11aaVfT26oXaaK2dkCjIJLAYxwefY1ft/BPiG70BNXtdKu57d5NiiO3kZmG3O8YXGztnPWtnS/E1jBceEBNdtFb6dLLJdRqrlIt0rHpjnKkdM8cVl2M+n3vg+fSrjUoNPnjvxdRm4jkZJFMZUgGNGIIIHUAc9aHs7f1qHU5ytDT/D+s6vC02laRfX0SNtZ7a2eRQeuCVB5pttqFtBYSW8ukWVzK+dtzK8wkjyMDAWQLx1GVPvmvRPD6QxaH4VXfDFrDR3H9nJcXEqJIZZGQZRIXDDPq6Z6HgA0wPO4NE1W5tri5ttMvJoLUkTyx27ssJAydxAwuPepLbw5rd7DHNZ6NqFxFKcRvFauyv1PBA56H8jXYaT4hsLDSNITOkW2p6K0yEaiLxssXLbk8glGBB2ncP4e4qfSfGemWun2EMmoNCIre0SSJEk2qUvDI46dkIPv05NCs2l/XqJ3VzzdlZGKuCrKcEEYINa1roEk3hm+1mf7VDFAUWBhZyPFOxbDDzR8qEcdevSqmrzRXOuX09sd0MtzI8ZAxlSxI47cVuaENMTwnrFvd65ZWtzfpEscMkVwSpSUMdxWIryBxgn3xUptxuyno7IxrbQdXvLRrq00q+nt1Qu00Vs7IFGQSWAxjg8+xpLLQ9W1G1kutP0u9u7eIkSTQW7uiYGTkgYHHNdPpHiSytZPByT3rrBpskr3abXKxlpSc4A5ypHTPpUWj3mkS6DHaeI7vTpbOEzPFCqXK3kBYf8s2VPKJJVThyR9Oab6iW5m2/gnxDd6Amr2ulXc9u8mxRHbyMzDbneMLjZ2znrWda6Nqd/ZzXdjpt3c21uCZpoYGdI8DJ3MBgcc81q6fPYXvgybSbnUodPuI74XSNcRyMkqmMqQDGrEEEDqAOetb/g3WPDelQ6NdX11AtzayyNcfaBdPLFk/L5KxkRbccndz97g8Aj3f8AXQXRHLWej2t54Y1DUkvZlurDYXtzbjYyu4UESb855zjb+NFno9reeGNQ1JL2Zbqw2F7c242MruFBEm/Oec42/jWlov8AZi+FtZgutdsrW41BYxFDJFcErslDHcViI5A4wT74puhDTE8J6xb3euWVrc36RLHDJFcEqUlDHcViK8gcYJ98UD6nL1saTo1rdaVeanql7LaWdtJHD+4txNI8jhiAFLoAMIxJJ9ODWPXV6f4y1Kw8D3On2+u6hb3K3cJto47iRdkQWTeFIOFGSnHGePSjo/66h1M6DRrG78O6nqNtf3BmsChMElqqq6PIEU7xISDzkjb+JrNj0+9la3WK0nc3RItwsTHzsHB28fNzxxXQ6H/Zq+FNYgvNdsra51BIxHDLHcMylJQx3FYiOQOME++K0/DuraJGnhi61HVltH0aWQTW5t5HeQNJuVlKrtx83OSCMHAPAoE/I4+20fU72znu7PTru4trfPnTRQMyRYGTuYDA455p9toOr3lo11aaVfT26oXaaK2dkCjIJLAYxwefY12Wj+J9Ng0XRmifSINQ0kzY/tFb0klnLBk8glSCDghh27jpV0vxNYwXHhATXbRW+nSyyXUaq5SLdKx6Y5ypHTPHFDG9DGt/BPiG70BNXtdKu57d5NiiO3kZmG3O8YXGztnPWsGujsZ9PvfB8+lXGpQafPHfi6jNxHIySKYypAMaMQQQOoA561lW2oW0FhJby6RZXMr523MrzCSPIwMBZAvHUZU++aOrDoLbaDq95aNdWmlX09uqF2mitnZAoyCSwGMcHn2NWbPR7W88MahqSXsy3VhsL25txsZXcKCJN+c85xt/Gut0+8t9O0HwbqN/rAtoLBp7n7EolMk5EzcIAuzJxtJZhwaxdHfTG8M63Hda1Y2dxqQj8u3eK4Jj2S7iCViK4IHGCffFD3YHOR6feytbrFaTubokW4WJj52Dg7ePm544qxp/h/WdWhaXStIv72NG2M9tbPIFb0JUHmus8O6tokaeGLrUdWW0fRpZBNbm3kd5A0m5WUqu3Hzc5IIwcA8CtjQkgj0fwuN8Merslx/ZyXFxKiSGWRkBKJC4YfV0z0PABpiODg8OzN4d1DVbtbu2S1ZEizZSNHM5baymTG1CvoevSq1toOr3lo11aaVfT26oXaaK2dkCjIJLAYxwefY1uaPFp9t4Z1uyv9bsrS7vFjjWCSOdihjl3HcUjZeQOME/hVrSPEllayeDknvXWDTZJXu02uVjLSk5wBzlSOmfSkN6HFVpW3hzW72GOaz0bULiKU4jeK1dlfqeCBz0P5GmQahawWc0Emj2VzI5bbcyvMJI8jjAWQLx1GVPvmu50fxjptnpmnW8uoyRiG2s45ECSYBS8Mkg4GPuYPv05PFNK9hPS5wP9mX/AJ1vF9hufMujiBPJbMvO35Rj5uQRx3FS2mg6xf281xY6VfXMMBKyyQ2zusZAyQxAwMD1rtbfWbc6DqurgsX0q/mGly7cK32ncMDPQpgyAe/asrRbzSZfD8Vn4ku9PltITM8UKpcreQFhz5bKvlEkqpw5I+nNTd2Ktqc/p+g6xq0LzaXpV9exRna721s8iqeuCVBxVm00a1u/DOoail5Mt1YbDJbNbjYyu4QYk35zznG38a63wdcaZqOp+EUbUxbzaa7Rmx8l2eWRpCwdWxswdyg5YEbOAeKxNGGmJ4X1q3u9csrW5v1jWOGSK4JUpKGO4rEV5A4wT74psSOVooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAr76u9QuUvp1WXCrIwA2j1+lfAtfeN9/yELj/AK6t/M0AP/tK7/56/wDjo/woqrRQB8heIvDp/tfUNQv9TstPjur25NslwJWeYLKwJHlxsAN2R8xHT05rmo9PvZWt1itJ3N0SLcLEx87BwdvHzc8cV3zana2mvalBrus6fdaR9vuXl0ie2mlmXLuD5TeXtjc9QVkUZxk8YqtoGraEn/CNXd7qq2X9jzS+bbNBJJI4aTepUqu0j5uckEYOAeBUx2QM4+x0TVdTilk03TLy8jh/1r29u0gj/wB4gcdD1osdF1TVIZZdM028vI4f9a9vA0gj78kA46d625bjTtW8JadY/wBrw6dNYSTtJDcxylZy7Ah1MaNzgbTux0FNafT9V8H6ZYHVodOn095mlguY5Ss5cgh1MaMM4G07sdBVD6mTDYazqWmGa3tL67sbEMDIkbvFbj7zcjhfU9PWm2Wh6tqNrJdafpd7d28RIkmgt3dEwMnJAwOOa7HRPE2nW+haKY20i31DSfNA/tFb0klnLBk8glCCDghh27jpQ0e+0mfQo7XxJdaa9nE00kVuiXK3duWHSNlXyjkhSA5YfTmh6XEjnbLQ9W1G1kutP0u9u7eIkSTQW7uiYGTkgYHHNWtJ0a1utKvNT1S9ltLO2kjh/cW4mkeRwxACl0AGEYkk+nBrX0W80mXw/FZ+JLvT5bSEzPFCqXK3kBYc+Wyr5RJKqcOSPpzUen+MtSsPA9zp9vruoW9yt3CbaOO4kXZEFk3hSDhRkpxxnj0oel/kC6HPf2dcyTwpa21zMtyxFqfJIacA4+UDOTnjAJweKvWmjWt34Z1DUUvJlurDYZLZrcbGV3CDEm/Oec42/jXSeHtX0VV8M3epauto+jyyedAYJHeTdJvVlKrtx83OSCMHAPArN0X+zF8LazBda7ZWtxqCxiKGSK4JXZKGO4rERyBxgn3xRsBm2ej2t54Y1DUkvZlurDYXtzbjYyu4UESb855zjb+NLZ3/AImtbODTrC61aG2vgfJtYJJVS43HadqjhsnI4z6Vf0IaYnhPWLe71yytbm/SJY4ZIrglSkoY7isRXkDjBPvitDw/qa2/gWfULgOLnRpmj05yOC1whBUH1TaZAPeh7MFujiGVkYq4KspwQRgg1rWugSTeGb7WZ/tUMUBRYGFnI8U7FsMPNHyoRx169KyK6jQhpieE9Yt7vXLK1ub9IljhkiuCVKShjuKxFeQOME++KOgdTGttB1e8tGurTSr6e3VC7TRWzsgUZBJYDGODz7GkstD1bUbWS60/S727t4iRJNBbu6JgZOSBgcc10+keJLK1k8HJPeusGmySvdptcrGWlJzgDnKkdM+lRaPeaRLoMdp4ju9Ols4TM8UKpcreQFh/yzZU8oklVOHJH05ofUFuZtv4J8Q3egJq9rpV3PbvJsUR28jMw253jC42ds561nWujanf2c13Y6bd3NtbgmaaGBnSPAydzAYHHPNaunz2F74Mm0m51KHT7iO+F0jXEcjJKpjKkAxqxBBA6gDnrW/4N1jw3pUOjXV9dQLc2ssjXH2gXTyxZPy+SsZEW3HJ3c/e4PAI93/XQXRHLWej2t54Y1DUkvZlurDYXtzbjYyu4UESb855zjb+NFno9reeGNQ1JL2Zbqw2F7c242MruFBEm/Oec42/jWlov9mL4W1mC612ytbjUFjEUMkVwSuyUMdxWIjkDjBPvim6ENMTwnrFvd65ZWtzfpEscMkVwSpSUMdxWIryBxgn3xQPqYdro+p3tlNeWWnXdxa2+fOnigZ0jwMncwGBxzzV/wAM+FNR8T6lDDaWt19laURzXkVs0kcGe7EcDj1IrpNE8Tadb6FopjbSLfUNJ80D+0VvSSWcsGTyCUIIOCGHbuOhoGsaF53hq81DVVsE0hpPNshBI7M7Ss4dSAV2kMoOTkBOAeKb3F0OBYbWI9Diun0Ce/Tw3fS3HiTVNM0iCRIGgsi0nmvKGOPL8xFxhGySfTg5rmHOXYjoTXVaf4y1Kw8D3On2+u6hb3K3cJto47iRdkQWTeFIOFGSnHGePSl9l/11KfxaGdBo1jd+HdT1G2v7gzWBQmCS1VVdHkCKd4kJB5yRt/E1mx6feytbrFaTubokW4WJj52Dg7ePm544rodD/s1fCmsQXmu2Vtc6gkYjhljuGZSkoY7isRHIHGCffFafh3VtEjTwxdajqy2j6NLIJrc28jvIGk3KylV24+bnJBGDgHgUEvyOPttH1O9s57uz067uLa3z500UDMkWBk7mAwOOeafbaDq95aNdWmlX09uqF2mitnZAoyCSwGMcHn2Ndlo/ifTYNF0Zon0iDUNJM2P7RW9JJZywZPIJUgg4IYdu46VdL8TWMFx4QE120Vvp0ssl1GquUi3SsemOcqR0zxxQxvQxrfwT4hu9ATV7XSrue3eTYojt5GZhtzvGFxs7Zz1rBro7GfT73wfPpVxqUGnzx34uozcRyMkimMqQDGjEEEDqAOetZVtqFtBYSW8ukWVzK+dtzK8wkjyMDAWQLx1GVPvmjqw6C22g6veWjXVppV9PbqhdporZ2QKMgksBjHB59jVmz0e1vPDGoakl7Mt1YbC9ubcbGV3CgiTfnPOcbfxrrdPvLfTtB8G6jf6wLaCwae5+xKJTJORM3CALsycbSWYcGsXR30xvDOtx3WtWNncakI/Lt3iuCY9ku4glYiuCBxgn3xQ92Bzken3srW6xWk7m6JFuFiY+dg4O3j5ueOKsaf4f1nVoWl0rSL+9jRtjPbWzyBW9CVB5rrPDuraJGnhi61HVltH0aWQTW5t5HeQNJuVlKrtx83OSCMHAPArY0JII9H8LjfDHq7Jcf2clxcSokhlkZASiQuGH1dM9DwAaYjg4PDszeHdQ1W7W7tktWRIs2UjRzOW2spkxtQr6Hr0qtbaDq95aNdWmlX09uqF2mitnZAoyCSwGMcHn2NbmjxafbeGdbsr/AFuytLu8WONYJI52KGOXcdxSNl5A4wT+FWtI8SWVrJ4OSe9dYNNkle7Ta5WMtKTnAHOVI6Z9KQ3ocVWlbeHNbvYY5rPRtQuIpTiN4rV2V+p4IHPQ/kaZBqFrBZzQSaPZXMjlttzK8wkjyOMBZAvHUZU++a7nR/GOm2emadby6jJGIbazjkQJJgFLwySDgY+5g+/Tk8U0r2E9LnA/2Zf+dbxfYbnzLo4gTyWzLzt+UY+bkEcdxUtpoOsX9vNcWOlX1zDASsskNs7rGQMkMQMDA9a7W31m3Og6rq4LF9Kv5hpcu3Ct9p3DAz0KYMgHv2rK0W80mXw/FZ+JLvT5bSEzPFCqXK3kBYc+Wyr5RJKqcOSPpzU3diranP6foOsatC82l6VfXsUZ2u9tbPIqnrglQcVZtNGtbvwzqGopeTLdWGwyWzW42MruEGJN+c85xt/Gut8HXGmajqfhFG1MW82mu0ZsfJdnlkaQsHVsbMHcoOWBGzgHisTRhpieF9at7vXLK1ub9Y1jhkiuCVKShjuKxFeQOME++KbEjmZLieWGKKWaR44QREjMSqAnJwO2TzxW3pnhC71O0tZVvLK3mvi4srWeRhJdbeDtwpUZIKjcVyRgVnW2oW0FhJby6RZXMr523MrzCSPIwMBZAvHUZU++a7nQ/GGnWul6DN9th0+50pWjul+zM9zcxq5dVhk2lUDbmU8p1OSRgU9BannLKyMVcFWU4IIwQa6nQYdeutGa00nxN9mEqSsmkpeSq0yquX+VQUGQDwxBOOnTORssdQt9Vv7vUPsl4HEltZiBn+0FmO4bxwu0c89a3fDr6Rp3h6a5h1uxttcug8GbqK4/0SIjBK7ImBdhkZzwD6nideV37Fac2nc4+ti18MX1z4du9ZD28dvbRiTZJJ+8lXzBGWVRk4DEDJwDzjODira2lhLpd/Pdal9nu4Nn2a28hn+05OG+ccJgc89a3fDR0yLw3rcN7rllZz6jbJDFFLFcMUKzI+WKRMMEKcYJ7dKYuq9TlKKDweufeigCzcaje3drb211eXE1vbAiCKSVmSIHqFBOF6dqrUUUASW9xNa3CT2sskM0Z3JJGxVlPqCORUfXrRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV+g9xpcElzK7NJlnJOCPX6V+fFfopJ/rX/3jQBnf2Rb/wB+T8x/hRV+igD4k1bwZcal4iuJItT0+K41K9uvsdpK0okmKyuu3OwoCSMDcw6iuWt9D1W7tri4tNMvJ4LYkTyxQM6RYGTuYDAwOea7TxJ4sg0y+hXTrW0n1Oxu73FzMkhe1drhypUbgjHBB5VsGl8H614d05NIvdRu4Bd200j3JuRdvNGWbjyVjIi2kHLbsnO7g8AzH4Rvc4q00PVtQhMthpd7dRhSxeG3dxgcE5A6Ciy0PVtRtZLrT9Lvbu3iJEk0Fu7omBk5IGBxzWzL4mltPDWg2+lX0kdzYXk908YB2o+5TGxBG1jgN69/WrWk6hpd1oaW/ii80+e2jaaRIlS4W8hZxz5bInlElgpw5I+lV0uHWxi2ugSTeGb7WZ/tUMUBRYGFnI8U7FsMPNHyoRx169Kr22g6veWjXVppV9PbqhdporZ2QKMgksBjHB59jWzoQ0xPCesW93rlla3N+kSxwyRXBKlJQx3FYivIHGCffFXdI8SWVrJ4OSe9dYNNkle7Ta5WMtKTnAHOVI6Z9KGT0Oa0/wAP6zq8LTaVpF9fRI21ntrZ5FB64JUHmqDo0cjJIpR1OGVhgg+hr1LQUgi0bwuA8MWrslx/ZyXFxKiSGWRkGUSFwR9XTPQ8AGvLpYmhmeKQYdGKsAc4I9xQ97DNnwz4U1HxPqUMNpa3X2VpRHNeRWzSRwZ7sRwOPUiqVvomqXtpPd2Wm3lza2+fNnht2dI8DJ3MBgYHPNdf4e1fQ1fwxdahq4sk0YuJbTyJHd3MhYSKVUrghlByQRs4B4qLS9Y0ot4cv5tXFidE3ebZ+TI0k2JGkzGVUp84YKdzLjHcU/6/4P8AwBI5Sx0XVNUhll0zTby8jh/1r28DSCPvyQDjp3qSQ61qFvp9pKb+6hIK2MLb3U84IiXp1GDt9K2bq703W/DFjarqcGlSWUtw721wkpSUu25WQxowzj5Tux0FaXh3VtEjTwxdajqy2j6NLIJrc28jvIGk3KylV24+bnJBGDgHgUhvTY5W28N65eQxy2mjahPHKcRvFauyueehA56H8jUVvouqXdrPdWum3k1vbZ8+WOBmSLAydxAwuB613mk+MtMtNPsIJNSkjENvaI6BJMApeGRxwP7hB9+nJ4pyeLNIIs7iyk0iG6025uWiOope/MHlZ1ZBD8pBVsEOAeMdOg/8/wAP8w/4BwdtoOr3lo11aaVfT26oXaaK2dkCjIJLAYxwefY1ft/BPiG70BNXtdKu57d5NiiO3kZmG3O8YXGztnPWtnS/E1jBceEBNdtFb6dLLJdRqrlIt0rHpjnKkdM8cVl2M+n3vg+fSrjUoNPnjvxdRm4jkZJFMZUgGNGIIIHUAc9aHs7f1qHUybXRtTv7Oa7sdNu7m2twTNNDAzpHgZO5gMDjnmtG38E+IbvQE1e10q7nt3k2KI7eRmYbc7xhcbO2c9a6PwbrHhvSodGur66gW5tZZGuPtAunliyfl8lYyItuOTu5+9weAcKyn0++8Hz6Vc6lDp9wl/8Aaka4jlZJVMZUgGNGIIIHUAc9aHonb+tQWpStdAkm8M32sz/aoYoCiwMLOR4p2LYYeaPlQjjr16VVtdF1S+spryy028ubWDPmzwwM6R4GTuYDAwOea29CGmJ4T1i3u9csrW5v0iWOGSK4JUpKGO4rEV5A4wT74q/pOr6Xjw3fT6uLA6GGEtn5UjSTESNJmMqpT5wwU7mXGO4xTEcvbaDq95aNdWmlX09uqF2mitnZAoyCSwGMcHn2NUK7mw8UWEd94TMl00NtYTzS3MIVysO+Vj0A5+UjpnjiuHbBc46Z4qSjX0nRrW60q81PVL2W0s7aSOH9xbiaR5HDEAKXQAYRiST6cGq0Oiaje29xd6Zp99d2VuW33KWzFUAGcsRkLxyeeK3NP8ZalYeB7nT7fXdQt7lbuE20cdxIuyILJvCkHCjJTjjPHpVzSdY0s/8ACN39xrH2FtDDedZ+VI0k5EjSZjKqUy4YKdzL07iqe/8AXYXQ5a00PVtQhMthpd7dRhSxeG3dxgcE5A6CqNdTdeKXh0DRItIu3t57O9nu2gUHbE5ZTGcEbWwM+vf1rmJJGllaSQ5Z2LMcdSakZraTo1rdaVeanql7LaWdtJHD+4txNI8jhiAFLoAMIxJJ9ODToNGsbvw7qeo21/cGawKEwSWqqro8gRTvEhIPOSNv4mtHT/GWpWHge50+313ULe5W7hNtHHcSLsiCybwpBwoyU44zx6U3Q/7NXwprEF5rtlbXOoJGI4ZY7hmUpKGO4rERyBxgn3xTe7sLoc9Hp97K1usVpO5uiRbhYmPnYODt4+bnjipLbR9TvbOe7s9Ou7i2t8+dNFAzJFgZO5gMDjnmuw8O6tokaeGLrUdWW0fRpZBNbm3kd5A0m5WUqu3Hzc5IIwcA8CpdH8T6bBoujNE+kQahpJmx/aK3pJLOWDJ5BKkEHBDDt3HQYHIxaTr2o6cksFhqN1ZW8bMjpDI8cSZO4ggYAyGz7g1bt/BPiG70BNXtdKu57d5NiiO3kZmG3O8YXGztnPWtnS/E1jBceEBNdtFb6dLLJdRqrlIt0rHpjnKkdM8cVl2M+n3vg+fSrjUoNPnjvxdRm4jkZJFMZUgGNGIIIHUAc9aHs7f1qHU5yug01vGA0WRNHOuDS9jPItqZvI2nIYkL8uOCCfY1m22oW0FhJby6RZXMr523MrzCSPIwMBZAvHUZU++a7fT7y307QfBuo3+sC2gsGnufsSiUyTkTNwgC7MnG0lmHBo6AclZ6Pa3nhjUNSS9mW6sNhe3NuNjK7hQRJvznnONv41nx6feytbrFaTubokW4WJj52Dg7ePm544ro9HfTG8M63Hda1Y2dxqQj8u3eK4Jj2S7iCViK4IHGCffFaPh3VtEjTwxdajqy2j6NLIJrc28jvIGk3KylV24+bnJBGDgHgUA9Njk9P8P6zq0LS6VpF/exo2xntrZ5ArehKg81Yg8OzN4d1DVbtbu2S1ZEizZSNHM5baymTG1CvoevSu80JII9H8LjfDHq7Jcf2clxcSokhlkZASiQuGH1dM9DwAa5bR4tPtvDOt2V/rdlaXd4scawSRzsUMcu47ikbLyBxgn8KHoMzBZeJdX0qEi21W+0+0jZoiI5JIoUycleoUZU5x/d9qx67XSPEllayeDknvXWDTZJXu02uVjLSk5wBzlSOmfSuZg1C1gs5oJNHsrmRy225leYSR5HGAsgXjqMqffNHViJ9O0TxEzQXuk6ZqZJYGCe2t5OW5Pyso68Hp6Gqf8AZl+ZreL7FcmW6P7hPKbdMc7flGPm5BHHcV32j+MdNs9M063l1GSMQ21nHIgSTAKXhkkHAx9zB9+nJ4qO31m3Og6rq4LF9Kv5hpcu3Ct9p3DAz0KYMgHv2oemvr+dvxBa/h+X6HFWmg6xf281xY6VfXMMBKyyQ2zusZAyQxAwMD1o0/QdY1aF5tL0q+vYoztd7a2eRVPXBKg4roNFvNJl8PxWfiS70+W0hMzxQqlyt5AWHPlsq+USSqnDkj6c1reDrjTNR1PwijamLebTXaM2PkuzyyNIWDq2NmDuUHLAjZwDxQK+lzkrTRrW78M6hqKXky3VhsMls1uNjK7hBiTfnPOcbfxrHrqtGGmJ4X1q3u9csrW5v1jWOGSK4JUpKGO4rEV5A4wT74rDttQtoLCS3l0iyuZXztuZXmEkeRgYCyBeOoyp980FPR2NHTPCF3qdpayreWVvNfFxZWs8jCS628HbhSoyQVG4rkjArBZWRirgqynBBGCDXo2h+MNOtdL0Gb7bDp9zpStHdL9mZ7m5jVy6rDJtKoG3Mp5TqckjAri9ljqFvqt/d6h9kvA4ktrMQM/2gsx3DeOF2jnnrQ99P67CWq1J9K8NS6xa7rXULEXTLI8VizsZpQi5OMKVHAOAzAnHA6Zxa7Dw6+kad4emuYdbsbbXLoPBm6iuP9EiIwSuyJgXYZGc8A+p4521tLCXS7+e61L7PdwbPs1t5DP9pycN844TA5560PcFsWrXwxfXPh271kPbx29tGJNkkn7yVfMEZZVGTgMQMnAPOM4OMeur8NHTIvDetw3uuWVnPqNskMUUsVwxQrMj5YpEwwQpxgnt0rlDweufeh7gtgooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK/RST/Wv/ALxr866/RST/AFr/AO8aAG0UUUAfEmueDLi98TXBi1PT1u9SubiW0snaRZJQJnUDds8sElCAC47etctb6Hq13az3Vrpd5Pb2xImmit3ZIsDJ3MBgYHPNd7rd/on/AAlun6jd6vGh0eebz7JIZTO7rdSyBVOzy8HcvJcY59MGlb+IdMvtQ0PWrrVV059LkkknsVikZ5SZmlzEQpQ7wwU7mXpzkUlt/XyYf1/mchbaDq95aNdWmlX09uqF2mitnZAoyCSwGMcHn2NMtdH1O9spryy067uLW3z508UDOkeBk7mAwOOea62w8UWEd94TMl00NtYTzS3MIVysO+Vj0A5+UjpnjirGjeJtNt9F0ZoW0i31DSTMF/tFb3ks5YMnkEoQQcEMO3cdH0uD3sefVr2ugSTeGb7WZ/tUMUBRYGFnI8U7FsMPNHyoRx169KyXbe7MFC5Odq9B7V0+hDTE8J6xb3euWVrc36RLHDJFcEqUlDHcViK8gcYJ98UdA6lfTW8YDRZE0c64NL2M8i2pm8jachiQvy44IJ9jWbZaHq2o2sl1p+l3t3bxEiSaC3d0TAyckDA45rp9I8SWVrJ4OSe9dYNNkle7Ta5WMtKTnAHOVI6Z9Ki0e80iXQY7TxHd6dLZwmZ4oVS5W8gLD/lmyp5RJKqcOSPpzQ+rBdjNt/BPiG70BNXtdKu57d5NiiO3kZmG3O8YXGztnPWs610bU7+zmu7HTbu5trcEzTQwM6R4GTuYDA455rV0+ewvfBk2k3OpQ6fcR3wuka4jkZJVMZUgGNWIIIHUAc9a3/BuseG9Kh0a6vrqBbm1lka4+0C6eWLJ+XyVjIi245O7n73B4BHu/wCuguiOWs9HtbzwxqGpJezLdWGwvbm3GxldwoIk35zznG38aLPR7W88MahqSXsy3VhsL25txsZXcKCJN+c85xt/GtLRf7MXwtrMF1rtla3GoLGIoZIrgldkoY7isRHIHGCffFN0IaYnhPWLe71yytbm/SJY4ZIrglSkoY7isRXkDjBPvigfUzrXQJJvDN9rM/2qGKAosDCzkeKdi2GHmj5UI469elV7bQdXvLRrq00q+nt1Qu00Vs7IFGQSWAxjg8+xrZ0IaYnhPWLe71yytbm/SJY4ZIrglSkoY7isRXkDjBPvirukeJLK1k8HJPeusGmySvdptcrGWlJzgDnKkdM+lDF0Oa0/w/rOrwtNpWkX19EjbWe2tnkUHrglQeaoOjRyMkilHU4ZWGCD6GvUtBSCLRvC4DwxauyXH9nJcXEqJIZZGQZRIXBH1dM9DwAa8uliaGZ4pBh0YqwBzgj3FD3sM2fDPhTUfE+pQw2lrdfZWlEc15FbNJHBnuxHA49SKpW+iape2k93ZabeXNrb582eG3Z0jwMncwGBgc811/h7V9DV/DF1qGriyTRi4ltPIkd3cyFhIpVSuCGUHJBGzgHiotL1jSi3hy/m1cWJ0Td5tn5MjSTYkaTMZVSnzhgp3MuMdxT/AK/4P/AEjlLHRdU1SGWXTNNvLyOH/WvbwNII+/JAOOneoo9PvZWt1itJ3N0SLcLEx87BwdvHzc8cV0V1d6brfhixtV1ODSpLKW4d7a4SUpKXbcrIY0YZx8p3Y6CtLw7q2iRp4YutR1ZbR9GlkE1ubeR3kDSblZSq7cfNzkgjBwDwKQ3pscrbeG9cvIY5bTRtQnjlOI3itXZXPPQgc9D+RqK30XVLu1nurXTbya3ts+fLHAzJFgZO4gYXA9a7zSfGWmWmn2EEmpSRiG3tEdAkmAUvDI44H9wg+/Tk8U5PFmkEWdxZSaRDdabc3LRHUUvfmDys6sgh+Ugq2CHAPGOnQf8An+H+Yf8AAODttB1e8tGurTSr6e3VC7TRWzsgUZBJYDGODz7Gr9v4J8Q3egJq9rpV3PbvJsUR28jMw253jC42ds561s6X4msYLjwgJrtorfTpZZLqNVcpFulY9Mc5Ujpnjisuxn0+98Hz6VcalBp88d+LqM3EcjJIpjKkAxoxBBA6gDnrQ9nb+tQ6mTa6Nqd/ZzXdjpt3c21uCZpoYGdI8DJ3MBgcc81o2/gnxDd6Amr2ulXc9u8mxRHbyMzDbneMLjZ2znrXR+DdY8N6VDo11fXUC3NrLI1x9oF08sWT8vkrGRFtxyd3P3uDwDhWU+n33g+fSrnUodPuEv8A7UjXEcrJKpjKkAxoxBBA6gDnrQ9E7f1qC1KVroEk3hm+1mf7VDFAUWBhZyPFOxbDDzR8qEcdevSqtrouqX1lNeWWm3lzawZ82eGBnSPAydzAYGBzzW3oQ0xPCesW93rlla3N+kSxwyRXBKlJQx3FYivIHGCffFX9J1fS8eG76fVxYHQwwls/KkaSYiRpMxlVKfOGCncy4x3GKYjl7bQdXvLRrq00q+nt1Qu00Vs7IFGQSWAxjg8+xqhXc2HiiwjvvCZkumhtrCeaW5hCuVh3ysegHPykdM8cVw7YLnHTPFSUa+k6Na3WlXmp6pey2lnbSRw/uLcTSPI4YgBS6ADCMSSfTg1Wh0TUb23uLvTNPvruyty2+5S2YqgAzliMheOTzxW5p/jLUrDwPc6fb67qFvcrdwm2jjuJF2RBZN4Ug4UZKccZ49KuaTrGln/hG7+41j7C2hhvOs/KkaSciRpMxlVKZcMFO5l6dxVPf+uwuhy1poerahCZbDS726jCli8Nu7jA4JyB0FVZLieWGKKWaR44QREjMSqAnJwO2TzxXS3Xil4dA0SLSLt7eezvZ7toFB2xOWUxnBG1sDPr39a5iSRpZWkkOWdizHHUmpGa2k6Na3WlXmp6pey2lnbSRw/uLcTSPI4YgBS6ADCMSSfTg06DRrG78O6nqNtf3BmsChMElqqq6PIEU7xISDzkjb+JrR0/xlqVh4HudPt9d1C3uVu4TbRx3Ei7Igsm8KQcKMlOOM8elN0P+zV8KaxBea7ZW1zqCRiOGWO4ZlKShjuKxEcgcYJ98U3u7C6GdZah4mt7KGw0671aK1vNwhtoJJAk+ThtqjhueDiqdto+p3tnPd2enXdxbW+fOmigZkiwMncwGBxzzXYeHdW0SNPDF1qOrLaPo0sgmtzbyO8gaTcrKVXbj5uckEYOAeBUuj+J9Ng0XRmifSINQ0kzY/tFb0klnLBk8glSCDghh27joMEcbbaDq95aNdWmlX09uqF2mitnZAoyCSwGMcHn2NX7fwT4hu9ATV7XSrue3eTYojt5GZhtzvGFxs7Zz1rZ0vxNYwXHhATXbRW+nSyyXUaq5SLdKx6Y5ypHTPHFZdjPp974Pn0q41KDT5478XUZuI5GSRTGVIBjRiCCB1AHPWh7O39ah1OcrYFl4l1fSoSLbVb7T7SNmiIjkkihTJyV6hRlTnH932qrbahbQWElvLpFlcyvnbcyvMJI8jAwFkC8dRlT75rt9PvLfTtB8G6jf6wLaCwae5+xKJTJORM3CALsycbSWYcGjoByVno9reeGNQ1JL2Zbqw2F7c242MruFBEm/Oec42/jVO0t9TiurSWwhu0nmbNq8KsGcg4+QjkkHjit/R30xvDOtx3WtWNncakI/Lt3iuCY9ku4glYiuCBxgn3xWj4d1bRI08MXWo6sto+jSyCa3NvI7yBpNyspVduPm5yQRg4B4FAPyOUsPD+tavE82l6Rf30aNtZ7a2eQBuuCVB5qeDw7M3h3UNVu1u7ZLVkSLNlI0czltrKZMbUK+h69K7zQkgj0fwuN8Merslx/ZyXFxKiSGWRkBKJC4YfV0z0PABrltHi0+28M63ZX+t2Vpd3ixxrBJHOxQxy7juKRsvIHGCfwoemgzDttB1e8tGurTSr6e3VC7TRWzsgUZBJYDGODz7GqFdrpHiSytZPByT3rrBpskr3abXKxlpSc4A5ypHTPpXMwahawWc0Emj2VzI5bbcyvMJI8jjAWQLx1GVPvmgQ+28Oa3ewxzWejahcRSnEbxWrsr9TwQOeh/I1X/sy/863i+w3PmXRxAnktmXnb8ox83II47iu+0fxjptnpmnW8uoyRiG2s45ECSYBS8Mkg4GPuYPv05PFR2+s250HVdXBYvpV/MNLl24VvtO4YGehTBkA9+1D019fzsvvBa/h+K/Q4q00HWL+3muLHSr65hgJWWSG2d1jIGSGIGBgetGn6DrGrQvNpelX17FGdrvbWzyKp64JUHFdBot5pMvh+Kz8SXeny2kJmeKFUuVvICw58tlXyiSVU4ckfTmtbwdcaZqOp+EUbUxbzaa7Rmx8l2eWRpCwdWxswdyg5YEbOAeKBX0uclaaNa3fhnUNRS8mW6sNhktmtxsZXcIMSb855zjb+NY9dVow0xPC+tW93rlla3N+saxwyRXBKlJQx3FYivIHGCffFcrQU9GXbXRtTv7Oa7sdNu7m2twTNNDAzpHgZO5gMDjnmrek6Na3WlXmp6pey2lnbSRw/uLcTSPI4YgBS6ADCMSSfTg11Pg3WPDelQ6NdX11AtzayyNcfaBdPLFk/L5KxkRbccndz97g8A5dj4z1Ky8F3dhba7qEFyLyE20cdxIuyELJvCkHCjJTjjPHpQ9L/ANdRLUpR+CddutF/tfT9MvbmzeYxxbbV97rjPmYAI29sgnnisu10fU72ymvLLTru4tbfPnTxQM6R4GTuYDA455rXsJ7C+8GzaVdalDYXKX4uka5jlZZVMZUgGNWIYEDqAOetbmieJtOt9C0UxtpFvqGk+aB/aK3pJLOWDJ5BKEEHBDDt3HQfX+ugdEcdaaHq2oQmWw0u9uowpYvDbu4wOCcgdBVGunl8Ty23hrQYNLvGhubG8nunhQHbG5ZTG2Dw3G717561zUkjSytJIcs7FmOOpNAGta+GL658O3esh7eO3toxJskk/eSr5gjLKoycBiBk4B5xnBxj11fho6ZF4b1uG91yys59RtkhiiliuGKFZkfLFImGCFOME9ulcoeD1z70PcFsXrLQ9W1G1kutP0u9u7eIkSTQW7uiYGTkgYHHNNtdH1O9spryy067uLW3z508UDOkeBk7mAwOOea6PRbzSZfD8Vn4ku9PltITM8UKpcreQFhz5bKvlEkqpw5I+nNaGieJtOt9C0UxtpFvqGk+aB/aK3pJLOWDJ5BKEEHBDDt3HQYHG2Oi6pqkMsumabeXkcP+te3gaQR9+SAcdO9FrouqX1lNeWWm3lzawZ82eGBnSPAydzAYGBzzW3Lc6drHhLTrL+1YNMmsZJ2kt545ikxdgQyGNG5wNvzY6Dmr+k6vpePDd9Pq4sDoYYS2flSNJMRI0mYyqlPnDBTuZcY7jFAHL22g6veWjXVppV9PbqhdporZ2QKMgksBjHB59jVzwz4U1HxPqUMNpa3X2VpRHNeRWzSRwZ7sRwOPUit6w8UWEd94TMl00NtYTzS3MIVysO+Vj0A5+UjpnjipdA1fQlm8M3d/q4so9HMgltPs8js7GRmEilV24IZQckEbOAeKPMJdUjkLTQdW1C3luNP0u9u7eEkSTQW7uiYGTkgYHHNUK7DR73SJtCjtPEd5p81nCZnihVLlbyAsP+WbKvlEkqpw5I+nNcfR1GwooooEFfopJ/rX/wB41+ddfopJ/rX/AN40ANooooA+Gde8MX1zeeItZD28dvbXU0myST95Kvn+WWVRk4DEDJwDzjODjk679jpkV14yhvdcsrOfUZpIYopYrhihW5D5YpEwwQpxgnt0rgDweufepjsvQff1NDT/AA/rOrwtNpWkX19EjbWe2tnkUHrglQeabBomq3NtcXNtpl5NBakieWO3dlhIGTuIGFx716J4fSGLQ/Cq74YtYaO4/s5Li4lRJDLIyDKJC4YZ9XTPQ8AGs/SfENhYaRpCZ0i21PRWmQjUReNli5bcnkEowIO07h/D3FU9P6/rb9RI4+28Oa3ewxzWejahcRSnEbxWrsr9TwQOeh/I1nMrIxVwVZTggjBBr0jSfGemWun2EMmoNCIre0SSJEk2qUvDI46dkIPv05NcFq80Vzrl9PbHdDLcyPGQMZUsSOO3FD3+/wDD/MFt934lu10CSbwzfazP9qhigKLAws5HinYthh5o+VCOOvXpVe20HV7y0a6tNKvp7dULtNFbOyBRkElgMY4PPsa2dCGmJ4T1i3u9csrW5v0iWOGSK4JUpKGO4rEV5A4wT74q7pHiSytZPByT3rrBpskr3abXKxlpSc4A5ypHTPpQxdDmLLQ9W1G1kutP0u9u7eIkSTQW7uiYGTkgYHHNaFv4J8Q3egJq9rpV3PbvJsUR28jMw253jC42ds561paPeaRLoMdp4ju9Ols4TM8UKpcreQFh/wAs2VPKJJVThyR9Oao6fPYXvgybSbnUodPuI74XSNcRyMkqmMqQDGrEEEDqAOetD2dvIfVfMyrXRtTv7Oa7sdNu7m2twTNNDAzpHgZO5gMDjnmrdno9reeGNQ1JL2Zbqw2F7c242MruFBEm/Oec42/jXU+DdY8N6VDo11fXUC3NrLI1x9oF08sWT8vkrGRFtxyd3P3uDwDkaL/Zi+FtZgutdsrW41BYxFDJFcErslDHcViI5A4wT74oYLYzbPR7W88MahqSXsy3VhsL25txsZXcKCJN+c85xt/Gqdro+p3tlNeWWnXdxa2+fOnigZ0jwMncwGBxzzW5oQ0xPCesW93rlla3N+kSxwyRXBKlJQx3FYivIHGCffFa+ieJtOt9C0UxtpFvqGk+aB/aK3pJLOWDJ5BKEEHBDDt3HQEZng+08Va00el6Re6xb6TLL5VxJa+a0EW4fMWCkL065IrlGG1iPQ4rvtA1jQvO8NXmoaqtgmkNJ5tkIJHZnaVnDqQCu0hlBycgJwDxXAucuxHQmh7j6CUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVJJcTywxRSzSPHCCIkZiVQE5OB2yeeKjooAKKKKANC28QazZWDWNnq19b2j5DW8Vy6xnPXKg4571n0UUAFFFFABVm41G9u7W3trq8uJre2BEEUkrMkQPUKCcL07VWooAKkt7ia1uEntZZIZozuSSNirKfUEcio6KADr1ooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK/RST/Wv/ALxr866/RST/AFr/AO8aAG0UUUAfAPif/kbtY/6/p/8A0Yay61PE/wDyN2sf9f0//ow1l1MfhQ3uaFt4g1mysGsbPVr63tHyGt4rl1jOeuVBxz3rPooqhBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV+ikn+tf/AHjX511+ikn+tf8A3jQA2iiigD4b13w1JNrWo397qVjpsV1qFwtsLtn3T4kYEgIjYAPG5toz34OOYSyupIZ5oreSSG2x50salkjycDLDgZPT1rtvFsuk69f7LjVodMu9NuLm2ljnglYSp58jqyFFbJ+cghtvIHNcrZtEuj6ov9sTWpby/LshG229+b+Ig7Rt+9zn2qY7DGS6Vt8OwatDL5qNO1vOmzHkuACvOeQwzjgfdI7Uy10XVL6ymvLLTby5tYM+bPDAzpHgZO5gMDA55rR8M3MEiX+jX00cNvqMOEklYKsc6fNGxJ4AzlSfRzWzpOr6WP8AhG7641cWDaGGEtmIpGeYiRpMxlVKfOGCncy9O4xV/wBf1/XUn+v6/rocvbaDq95aNdWmlX09uqF2mitnZAoyCSwGMcHn2NX7fwT4hu9ATV7XSrue3eTYojt5GZhtzvGFxs7Zz1rq7fULW00vwlqt3qq2VrZy3F0NPQSs8uJ2O1AF2cj5csw4rm7S6sNR8JXOm3OoW+m3A1D7YnnxyFHUoVKgxoxBBx1AHPWpez/rqV1/rz/r5mVLpW3w7Bq0Mvmo07W86bMeS4AK855DDOOB90jtV638E+IbvQE1e10q7nt3k2KI7eRmYbc7xhcbO2c9aPDNzBIl/o19NHDb6jDhJJWCrHOnzRsSeAM5Un0c1Jp81heeDZtJutSg0+4jvhdI1xHIySr5ZUgGNWIIIHUAc9ab2dvL/giW6IPDPhTUfE+pQw2lrdfZWlEc15FbNJHBnuxHA49SKxWG1iPQ4rvPD2r6Gr+GLrUNXFkmjFxLaeRI7u5kLCRSqlcEMoOSCNnAPFclBqNrDZzQSaRZXUkhbbdSvMJEyOMBZAvHUZU++aHuCGWujanf2c13Y6bd3NtbgmaaGBnSPAydzAYHHPNP0/w/rOrwtNpWkX19EjbWe2tnkUHrglQea7HwbrHhvSodGur66gW5tZZGuPtAunliyfl8lYyItuOTu5+9weAdPQkgj0fwuN8Merslx/ZyXFxKiSGWRkBKJC4YfV0z0PABpsDzmDRNVuba4ubbTLyaC1JE8sdu7LCQMncQMLj3pbbQdXvLRrq00q+nt1Qu00Vs7IFGQSWAxjg8+xrs9J8Q2FhpGkJnSLbU9FaZCNRF42WLltyeQSjAg7TuH8PcVU0vxNYwXHhATXbRW+nSyyXUaq5SLdKx6Y5ypHTPHFL0DZGNb+CfEN3oCava6Vdz27ybFEdvIzMNud4wuNnbOetZZ0y8GkrqfkMbNpTD5ykEK4Gdpx9045GcZ5xnBrZsZ9PvfB8+lXGpQafPHfi6jNxHIySKYypAMaMQQQOoA560eHtZs/D2m3NyWN9d3atbnTpFP2fy8ffl/v8AJ+VVxgjJIwAR7v8AroHb+v6/pHOUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFfopJ/rX/wB41+ddfopJ/rX/AN40ANooooA+AfE//I3ax/1/T/8Aow1l1qeJ/wDkbtY/6/p//RhrLqY/ChvcKKKKoRJJcTywxRSzSPHCCIkZiVQE5OB2yeeKjoooAKKKKACiiigArQtvEGs2Vg1jZ6tfW9o+Q1vFcusZz1yoOOe9Z9FABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABX6KSf61/wDeNfnXX6KSf61/940ANooooA+AfE//ACN2sf8AX9P/AOjDWXWp4n/5G7WP+v6f/wBGGsupj8KG9woooqhBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV+ikn+tf/eNfnXX6KSf61/940ANooooA+AfE/8AyN2sf9f0/wD6MNZdanif/kbtY/6/p/8A0Yay6mPwob3CiiiqEFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABX6KSf61/941+ddfopJ/rX/wB40ANooooA+AfE/wDyN2sf9f0//ow1l1qeJ/8AkbtY/wCv6f8A9GGsupj8KG9woooqhBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV+ikn+tf/AHjX511+ikn+tf8A3jQA2iiigD//2Q==) The above command will install fluent plugin for specified type. For example, to install fluent plugin for mysql: > fluent-gem install fluent-plugin-xyz

After installing the plugin, we can run configuration as : > fluentd -c conf\td-agent.conf

**Configuration :**

The configuration file consists of the following directives:

* source directives determine the input sources.
* match directives determine the output destinations.
* filter directives determine the event processing pipelines.

For more information regarding configuraion : <https://docs.fluentd.org/v0.12/articles/config-file>

**source** : where all the data come from! Fluentd’s input sources are enabled by selecting and configuring the desired input plugins using source directives. Fluentd’s standard input plugins include http and forward. http turns fluentd into an HTTP endpoint to accept incoming HTTP messages whereas forward turns fluentd into a TCP endpoint to accept TCP packets.

Config Sample

<source>

@type forward

host localhost # Optional (default: localhost)

port 3306 # Optional (default: 3306)

username nagios # Optional (default: root)

password passw0rd # Optional (default nopassword)

interval 30s # Optional (default: 1m)

tag input.mysql # Optional (to match with coming tag)

</source>

**match** : Tell fluentd what to do! The “match” directive looks for events with matching tags and processes them. The most common use of the match directive is to output events to other systems (for this reason, the plugins that correspond to the match directive are called “output plugins”).

Each **match** directive must include a match pattern and a type parameter. Only events with a **tag**matching the pattern will be sent to the output destination.

Config source with match

<source>

@type forward

port 24224

tag mysql

</source>

<match mysql.\*>

@type mysql\_bulk # (specifies the output plugin to use)

host 10.98.8.100 # (host to connect with mysql database)

port 3306 # (port to connect with db)

database security\_dev # (specifies mysql database)

username devuser # (username for mysql db)

password leo$123 # (password for mysql db)

column\_names level,logger,thread,message # (column names )

key\_names level,logger,thread,message # (key name to match column)

table fluentdlog # (table name to store logs)

flush\_interval 10s # (interval time to flush logs to db)

</match>

For more: <https://github.com/tagomoris/fluent-plugin-mysql/blob/master/README_mysql.md>

Here, we can see table & columns specification in conf file. This table we will have to create accrodingly the configuration. If we are saving data in json format then create table with one column only. For example, we are using fluentd table to save date in four columns as level, logger, thread message. These details will be passed to server runnnig with conf by DataFluentAppender. We can access log event time also directly by using ${time}.

<source>

@type forward

port 24224

</source>

<match mysql.\*>

@type mysql\_bulk

host "10.98.8.100"

port 3306

database "security\_dev" **column-name**

username "devuser"

password xxxxxx

column\_names "time,level,logger,thread,message"

key\_names "${time},level,logger,thread,message"

table "fluentdlog"

flush\_interval 5s **table-name**

<buffer>

flush\_interval 5s **<flush interval time to flush to db>**

</buffer>

</match>

1. **Connecting With Spring Application**

To send logs from a spring application to fluent-plug, add required jars as discussed at point 1. Now get the following two classes as it is from the package we added as jar:

**ch.qos.logback.more.appenders**

To get classes : <https://github.com/sndyuk/logback-more-appenders>

Classes to add to configuration are: DaemonAppender<E>

DataFluentAppender

DataFluentAppender class contains following properties.

**private** FluentLogger fluentLogger; // fluentd logger

**private** **final** String tag; // tag to match with configuraion

**private** **final** String label; // logger level

**private** **final** String remoteHost; // remote host to connect with for logs

**private** **final** **int** port; // port to connect

**private** **final** Map<String, String> additionalFields; // to more fields

The above properties will used to connect with fluentd server running with specific configuration. Overriden method append will add a data Map to for logs as:

@Override

**protected** **void** append(ILoggingEvent rawData) {

**final** Map<String, Object> data = **new** HashMap<String, Object>();

data.put("message", rawData.getFormattedMessage());

data.put("logger", rawData.getLoggerName());

data.put("thread", rawData.getThreadName());

data.put("level", rawData.getLevel());

**if** (rawData.getMarker() != **null**

data.put("marker", rawData.getMarker());

fluentLogger.log(label, data, rawData.getTimeStamp() / 1000);

}

The above map keys will added to logs. Basically, the above two classed discussed need not to be change if any specific requirement is not there. The reason for having these two classes is visibility to Logger configuration which is the most important as follows:

**Configuring Logger:**

Now, define a class having DataFluentAppender as a property. Inside this class, configure logging based property to DataFluetnAppender object reference.

The properties described above need to be set to this appender to connect with fluentd server running with same configuration. The method should accept a Logger object [slf4j logger ], which will be then modified to redirect logs to fluentd server.

Now call this method from any other class by passing class name to have fluentd support for that logger.

**public** **class** LoggerInit {

**private** DataFluentAppender fle = **null**;

@SuppressWarnings({ "rawtypes", "unchecked" })

**public** Logger initLoggerContext(Logger logger){

LoggerContext loggerContext = (LoggerContext) LoggerFactory.*getILoggerFactory*();

logger.setLevel(Level.***INFO***);

fle = **new** DataFluentAppender();

fle.setName("FLUENT");

fle.setContext(loggerContext);

fle.setTag("mysql");

fle.setLabel("INFO");

fle.setRemoteHost("127.0.0.1");

fle.setPort(24224);

fle.setMaxQueueSize(20);

//Set the supplied threshold

ThresholdFilter ftf = **new** ThresholdFilter();

ftf.setLevel(Level.*toLevel*("INFO".toString(), Level.***WARN***).toString());

fle.addFilter(ftf);

//Now start the appender

fle.start();

//Now add it to the passed in logger as an appender

logger.addAppender(fle);

ConsoleAppender console = (ConsoleAppender)loggerContext.getLogger(Logger.***ROOT\_LOGGER\_NAME***).getAppender("STDOUT");

**if**(console == **null**)

console = (ConsoleAppender)loggerContext.getLogger(Logger.***ROOT\_LOGGER\_NAME***).getAppender("console");

ThresholdFilter ctf = **new** ThresholdFilter();

ctf.setLevel(Level.*toLevel*("INFO".toString(), Level.***WARN***).toString());

**if**(console != **null**)

console.addFilter(ctf);

**return** logger;

}

**public** DataFluentAppender getFluentAppender(){

**return** fle;

}

}

Getting Custom logger:

@Component

**public** **class** DemoLogger {

**private** **static** Logger *logger*;

@SuppressWarnings("rawtypes")

**public** **static** Logger getCustomLogger(Class clazz) {

LoggerInit loggerInit = **new** LoggerInit();

*logger* = (Logger)LoggerFactory.*getLogger*(clazz);

*logger* = loggerInit.initLoggerContext(*logger*);

**return** *logger*;

}

}

This class can be called to get a custom logger which will have support for fluentd in any other class.

**static** Logger *logger* = DemoLogger.*getCustomLogger*(DemoRestController.**class**);

**Logback vs SLF4J**

SLF4J

This is simple logging façade to abstract the various logging frameworks such as logback, log4j, commons-logging and default java logging implementation (java.util.logging). This primarily enables the user to inlcude desired logging framework at deployment time. It is lightweight and nearly adds a zero overhead on performance.

Note that SLF4j doesn’t replace any logging framework; it is just a façade around any standard logging framework. If slf4j doesn’t find any logging framwork in classpath, by default it prints the logs in console.

Logback

This is an improved version of log4j and natively supports the slf4j, hence migrating from other logging frameworks such as log4j and java.util.logging is quite possible.

Since the logback natively supports slf4j, the combination of using slf4j with this framework is relatively faster than the slf4j with other logging frameworks. Logging configuration can be done either in xml or groovy.

Note : SLF4J is basically an abstraction layer. It is not a logging implementation. It means that if you're writing a library and you use SLF4J, you can give that library to someone else to use and they can choose which logging implementation to use with SLF4J e.g. log4j or the Java logging API. It helps prevent projects from being dependent on lots of logging APIs just because they use libraries that are dependent on them.

So, to summarise: SLF4J does not replace log4j, they work together. It removes the dependency on log4j from your library/app.

***For References:***

<https://docs.fluentd.org/v0.12/articles/quickstart>

<https://github.com/tagomoris/fluent-plugin-mysql/blob/master/README_mysql.md>

<https://www.fluentd.org/architecture>

<https://docs.fluentd.org/v0.12/articles/config-file>

<https://www.fluentd.org/guides/recipes/graylog2>

<https://softwareengineering.stackexchange.com/questions/108683/slf4j-vs-log4j-which-one-to-prefer>

<https://dzone.com/articles/slf4j-logback>