


```

+-----+
| Node |
+-----+
| - nodeId: String |
| - status: String |
| - resourceUsage: Map|
| - isLeader: bool |
+-----+
| + sendHeartbeat(): void |
| + processTask(task): void |
| + replicateData(): void |
+-----+

```

```

+-----+
| TaskScheduler |
+-----+
| - taskQueue: Queue |
| - activeTasks: List|
+-----+
| + scheduleTask(): void|
| + balanceLoad(): void|
+-----+

```

depends

^

|

```

+-----+
| DataReplicator |
+-----+
| - replicationFactor: int |
| - consistencyLevel: enum |
+-----+
| + replicate(): void |
| + syncData(): void |
| + resolveConflicts(): void|
+-----+

```

communicates with

^

|

```

+-----+
| MessageQueue |
+-----+
| - queueName: String|
| - messages: List |
+-----+
| + enqueue(msg):void|
| + dequeue(): Message|
| + acknowledge(): void|
+-----+

```

monitors

^

|

```

+-----+
| MonitoringService |
+-----+
| - metrics: Map |
| - alerts: List |
+-----+
| + collectMetrics():void |
| + sendAlert(): void |
| + analyzeLogs(): void |
+-----+

```

이 UML 클래스 다이어그램 표기법에 따라 시스템 내 주요 클래스 간의 구조와 관계를 명확하게 시각화할 수 있으며, 각 클래스는 속성과 메서드를 통해 책임과 동작을 구체화합니다. 이를 기반으로 세부 설계 및 구현이 진행 가능합니다.

필요 시 UML 모델링 도구를 통해 실제 다이어그램 시각화 및 추가 상세화를 권장합니다. [1] [2] [3]

✻

1. <https://wikidocs.net/292254>
2. <https://liddang-blog.tistory.com/97>
3. <https://brownbears.tistory.com/577>
4. <https://bluestickvic.tistory.com/13>
5. <https://swk3169.tistory.com/171>
6. <https://velog.io/@reasonz/UML-클래스-다이어그램-작성방법>
7. <https://blog.naver.com/qhanfckwsmsd/221554139159>
8. http://dslab.konkuk.ac.kr/class/2016/16SMA/Team_project/Team1/UML_T1.pdf
9. <https://kimyc1223.github.io/blog/2024/04/19/TechPost.html>
10. <http://kocw-n.xcache.kinxcdn.com/data/document/2014/dongguk/choieunman/06.pdf>