

RabbitMQ Architecture Diagrams (PlantUML)

```
@startuml
title RabbitMQ Commercial Deployment Architecture
node "Client Devices" {
    [Producer A]
    [Producer B]
}
cloud "Kubernetes Cluster" {
    node "RabbitMQ Operator"
    node "RabbitMQ Cluster" {
        [Node 1]
        [Node 2]
        [Node 3]
    }
}
rectangle "Observability Stack" {
    [Prometheus]
    [Grafana]
}
rectangle "Identity Provider" {
    [LDAP/AD]
    [OAuth2/OIDC]
}
[Producer A] --> [Node 1] : AMQP/MQTT
[Producer B] --> [Node 2] : AMQP/WebSocket
[Node 1] --> [Prometheus]
[Node 2] --> [Prometheus]
[Node 3] --> [Prometheus]
[Prometheus] --> [Grafana]
[Node 1] --> [LDAP/AD]
[Node 2] --> [OAuth2/OIDC]
@enduml
```

RabbitMQ Load Test Architecture (PlantUML)

```
@startuml
title RabbitMQ Load Test Architecture
actor LoadGenerator as LG
node "Load Test Controller" as LTC
cloud "Kubernetes Cluster" {
    node "RabbitMQ Cluster" {
        [Node 1]
        [Node 2]
        [Node 3]
    }
}
database "Results DB" as DB
LG --> LTC : Start Load Test
LTC --> [Node 1] : Publish Traffic
LTC --> [Node 2] : Consume Traffic
[Node 1] --> DB : Metrics
[Node 2] --> DB : Queue Depth
LTC --> DB : Logs & KPIs
@enduml
```

RabbitMQ Load-Test Reporting Template

RabbitMQ Load-Test Reporting Template

1. Test Objectives

- Validate throughput, latency, and failover behavior.
- Assess cluster stability under peak load.
- Measure backlog processing capability.

2. Test Environment

- RabbitMQ Cluster Size:
- Queue Type:
- Hardware/Cloud Specs:
- Load Generator Tools:

3. Test Scenarios

- Baseline TPS Test
- Peak Throughput Test
- Backlog Replay Test
- Failover Test
- Connection Scale Test

4. KPIs Measured

- Publish TPS
- Consume TPS
- P99/P999 Latency
- Failover Time
- CPU / Memory / Disk IO
- Backlog drain rate

5. Results Summary

- Peak Publish TPS:
- Peak Consume TPS:
- Latency:
- Failover:
- Backlog Recovery:

6. Observations

- Issues:
- Tuning Suggestions:

7. Final Evaluation

- Pass/Fail Criteria: