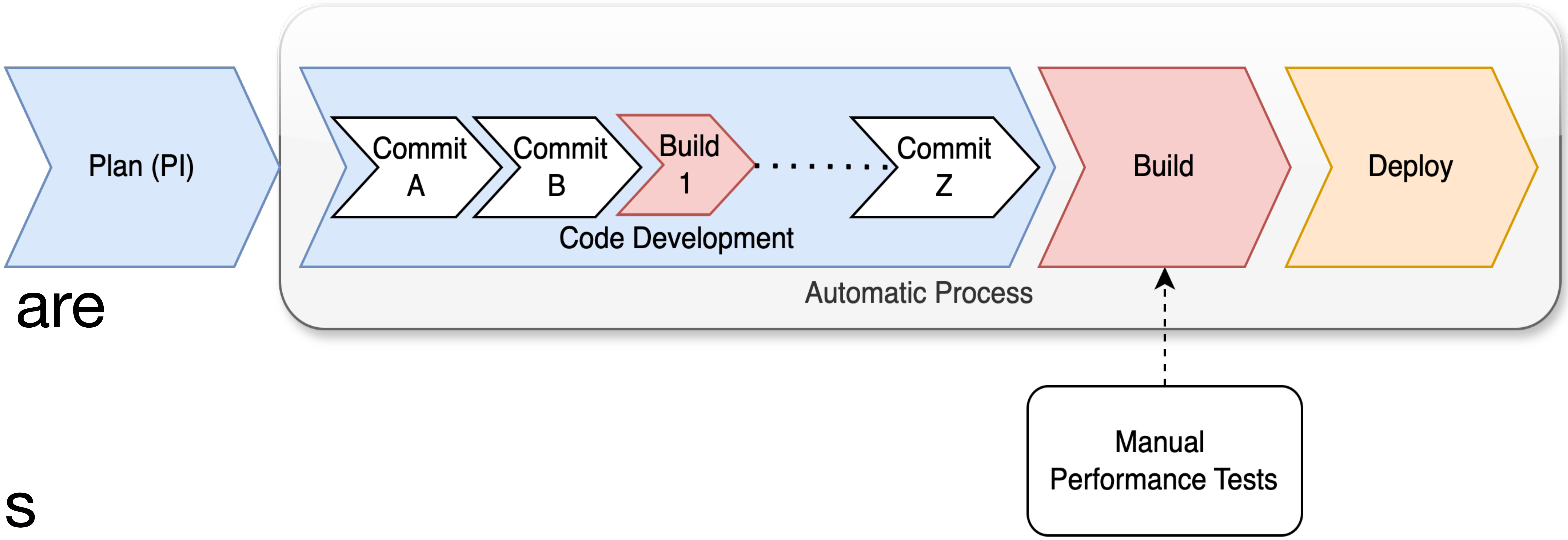
Continuous Performance Testing

Nordea Assignment

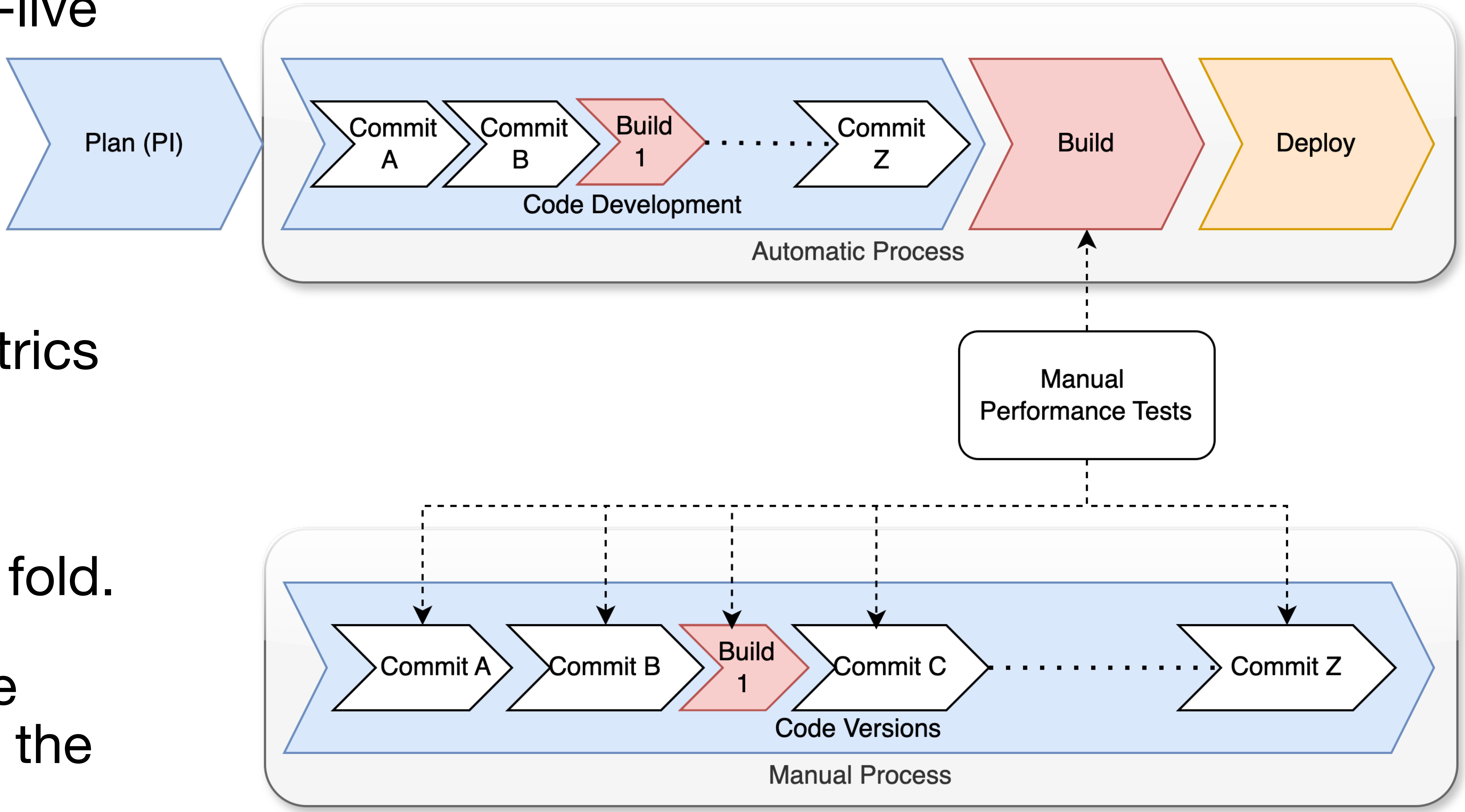
Current Process

- Application changes are made frequently to master branches.
- Multiple code change commits are made on daily basis.
- CI/CD Pipelines exists for builds and deployment of micro services. The pipelines run overnight for daily deployments.
- Performance evaluation are carried out only on go-live/release revisions.
- Manual performance tests conducted.



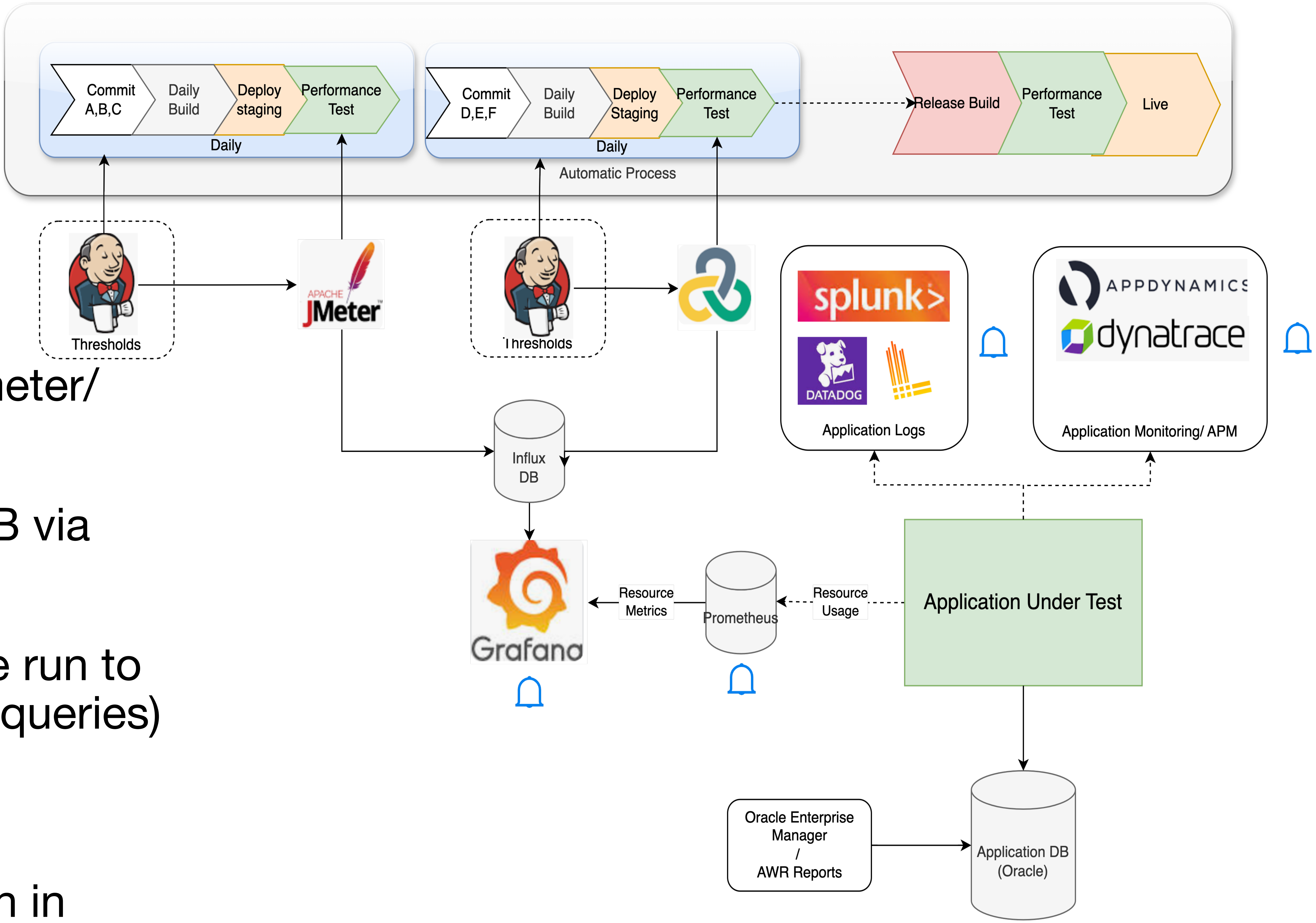
Current Process Drawbacks

- Delays in Performance testing affecting go-live of application due to late detection of performance issues.
- No regular performance trends available to monitor changes in performance of the application. Less frequent collection of metrics via test which hinders decision making.
- Time required to investigate the issue and identify a problem commit increases many fold.
- Rolling back an old commit is cumbersome due to other later commits dependency on the code.
- Inconsistency in tests with manual runs.



Automatic Testing Process

- Automatic Performance tests triggered against daily build, triggered by CI/CD tool eg Jenkins



- Jenkins supports plugins to trigger Jmeter/ LoadRunner tests.
- Test results can be shipped to InfluxDB via integration plugins.
- Within Influx daily aggregations can be run to create materialised views (continuous queries) to aggregate data.
- The results can be viewed on Grafana dashboards. With the data aggregation in Influx. Longer periods of trends can be tracked.

Type Of Tests

In pipeline

- Low load baseline tests:
    - Run low load baseline tests
- Peak Load Test
    - Run Peak load tests for x hours depending on the production profile.
- Endurance Tests
    - Run 80% of Peak load test for long durations of 8 hours

Type Of Tests

Additional Adhoc Tests/Before Go-Live

- Step Up Capacity Test:
  - Increase load in steps , retain load at each step for 30 min . Verify at which step the Performance starts degrading , and at which level application eventually crashes.
- Endurance Tests
  - Run 80% of Peak load test for long durations of 12 hours - 48 hours.
- Day in Life Tests
  - Simulate 1 day of production load in the tests.
- Failover / Resilience / Disaster Recovery Tests.

Metrics

- Monitoring:
  - Performance Tool Metrics:
    - Response Time P90, Max, Avg.
    - Throughput, Hits/sec
    - Time to First Buffer, Network Time To First Buffer, Download Time, Connection Time,.
  - Server Metrics:
    - CPU, Memory, Disk I/O , Network latency, TCP/IP connection rate, Packet retransmission rate, Thread and DB Connection pool size, HTTP Server connection pool, HTTP Requests queue length.
    - Server metrics can be collected by Prometheus or APM tools like Splunk/Appdynamics
  - JVM Metrics:
    - Utilization: Total Heap , YoungGen, OldGen, Eden / Survivour Spaces.
    - GC Pauses, GC Rate, Allocation Stalls
  - Database Metrics:
    - AWR Reports, Explain Plans, Optimizer
    - Oracle Enterprise Manager: Resource Utilisation, SQL Execution Times, Wait Events. Optimizer statistics.
  -

Alerting & Threshold

- Thresholds:
  - Jenkins Jmeter/Loadrunner plugins can be configured with minimal thresholds to trigger pipeline failures incase of breach. But these plugins are not primary source of evaluation of performance, but can be used to Pass/Fail tests.
  - Other places of setting thresholds and alerting are eg Grafana, Prometheus Splunk, Appdynamics etc.
  - These tools have also integrations with enterprise messaging applications eg Slack, where the alerts can be send to specific channels and groups.