Final Exam Exercise: Advanced Cloud Computing

Exercise 1: Network Performance Testing with Linux Network Namespaces

Objective: Evaluate and compare the network performance of different inter-namespace connection methods using Linux network namespaces.

Setup:

1. Create two network namespaces, `ns1` and `ns2`, on a Linux system.
2. Connect `ns1` and `ns2` using the following methods:
   - Directly via a veth pair (e.g., `veth0` in `ns1` connected to `veth1` in `ns2`)
   - One veth pair per namespace, with both veth pairs connected to a bridge (e.g., `br0`)
   - Two veth pairs (one per namespace), each connected to a separate bridge (e.g., `br0` and `br1`), which are then connected via VXLAN
3. Configure IP addresses for each namespace and ensure connectivity between them.

Performance Testing:

1. Run `iperf` tests to measure bandwidth and latency between `ns1` and `ns2` for each connection method.
2. Record the results, including throughput and latency.
3. Compare the performance metrics across the different connection methods.

Comparison with Cloud Basic Case:

1. Record the results, including throughput and latency.
2. Compare the performance metrics between the cloud basic case and the network namespace setup.
3. Comment on any differences or similarities observed.

Deliverables:

A detailed report containing:

- Steps for setting up network namespaces and connecting them using different methods
- `iperf` test results, including throughput and latency, for each connection method
- Comparison of performance metrics across connection methods and with the cloud basic case
- Discussion on the implications of the results

Exercise 2: Kubernetes Performance Monitoring with Prometheus

Objective: Evaluate the performance of a Kubernetes cluster while running a high-performance computing (HPC) workload using the High-Performance Linpack (HPL) test.

Setup:

1. Install a Kubernetes cluster on a set of machines or use an existing one.
2. Provision the Kube-Prometheus stack, including Prometheus, Grafana, and Alertmanager.
3. Create a Pod running the HPL test, configured to utilize multiple CPUs and memory.

Monitoring and Analysis:

1. Monitor the behavior of the Kubernetes node(s) while the HPL test is running using the Kube-Prometheus stack.
2. Observe metrics such as CPU utilization, memory usage, network throughput, and disk I/O.
3. Analyze the performance data to identify any bottlenecks or issues with the node(s).
4. If the node(s) do not function correctly during the test (e.g., due to resource exhaustion), describe how to fix the problem.

Deliverables:

A detailed report containing:

- Steps for setting up the Kubernetes cluster and provisioning the Kube-Prometheus stack
- Description of the HPL test Pod configuration and execution
- Analysis of the performance data, including identification of any bottlenecks or issues
- Discussion on how to fix any problems encountered during the test
- Screenshots of Prometheus and Grafana dashboards showing relevant metrics

Note: These exercises are designed to be more challenging than the original cloud computing course exam. The student is expected to have a good understanding of Linux network namespaces, Kubernetes, and performance monitoring using Prometheus.