

# Version

- 1.0 – initial document
- 1.1 – added new task related to Asterisk
- 1.2 (current) – removed OpenVPN related task

# Process

1. The candidate provides their public key for SSH access.
2. We create virtual machines and provide the candidate with a [hosts file](#).
3. The candidate can begin working on the test tasks.
4. Once everything is completed, we have a meeting to review the solutions, test them, and discuss the details.

# Test Tasks

## 1. Primary / Replica MariaDB Setup

You have 2 servers running Debian 12 (see `maria-primary` and `maria-replica` in the hosts file).

Install MariaDB 10.11 on both servers and set up a Primary / Replica configuration.

You should also prepare the system to handle high traffic volumes.

Example script to create the database for testing purposes:

```
-- Create the database
CREATE DATABASE IF NOT EXISTS voip_db;

-- Use the newly created database
USE voip_db;

-- Create the calls table
CREATE TABLE calls (
    id INT AUTO_INCREMENT PRIMARY KEY,           -- Unique
    identifier for each call
    caller_id VARCHAR(20) NOT NULL,               -- Phone number or
    SIP ID of the caller
```

```

        callee_id VARCHAR(20) NOT NULL,                -- Phone number or
SIP ID of the callee
        call_start_time DATETIME DEFAULT CURRENT_TIMESTAMP, -- Start time
of the call
        call_end_time DATETIME,                        -- End time of the
call
        duration INT GENERATED ALWAYS AS (TIMESTAMPDIFF(SECOND,
call_start_time, call_end_time)) STORED, -- Call duration in seconds
        call_status ENUM('connected', 'failed', 'busy', 'no_answer') NOT
NULL, -- Call status
        codec_used VARCHAR(50),                        -- Codec used for
the call (e.g., G.711, G.729)
        call_direction ENUM('inbound', 'outbound') NOT NULL, -- Direction
of the call
        call_cost DECIMAL(10, 2),                    -- Cost of the
call, if applicable
        created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -- Record creation
timestamp
        updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP -- Last update timestamp
    );

```

```

-- Insert sample data (optional)

```

```

INSERT INTO calls (caller_id, callee_id, call_end_time, call_status,
codec_used, call_direction, call_cost)
VALUES
    ('1234567890', '0987654321', '2024-01-01 12:15:30', 'connected',
'G.711', 'outbound', 0.10),
    ('5551234567', '4449876543', NULL, 'no_answer', 'G.729',
'inbound', 0.00),
    ('7890123456', '5678901234', '2024-01-01 14:45:00', 'connected',
'G.711', 'outbound', 0.25),
    ('3216549870', '8765432109', '2024-01-01 15:00:00', 'failed',
NULL, 'inbound', 0.00);

```

## 2. PostgreSQL Replica Load Balancing

You have 3 servers running Debian 12 (see `postgres-primary`, `postgres-replica-1`, and `postgres-replica-2` in the hosts file).

Install PostgreSQL 9.6 on each of the servers and set up replication.

Implement load balancing on port 5433 of `postgres-primary` to distribute read-only queries across both replicas.

Example script to create the database for testing purposes:

```
-- Create the database (if it doesn't already exist)
CREATE DATABASE IF NOT EXISTS ecommerce;

-- Connect to the newly created database
\c ecommerce;

-- Create the orders table
CREATE TABLE IF NOT EXISTS orders (
    id SERIAL PRIMARY KEY,                -- Unique
    identifier for each order
    customer_id INT NOT NULL,             -- ID of the
    customer placing the order
    order_date TIMESTAMPTZ DEFAULT NOW(), -- Date and time
    when the order was placed
    total_amount DECIMAL(10, 2) NOT NULL, -- Total amount
    for the order
    status ENUM('pending', 'completed', 'shipped', 'cancelled') NOT
    NULL, -- Order status
    shipping_address VARCHAR(255) NOT NULL, -- Shipping
    address for the order
    payment_method ENUM('credit_card', 'paypal', 'bank_transfer') NOT
    NULL, -- Payment method used
    created_at TIMESTAMPTZ DEFAULT NOW(), -- Timestamp when
    the record was created
    updated_at TIMESTAMPTZ DEFAULT NOW() ON UPDATE NOW() -- Timestamp
    when the record was last updated
);

-- Optional: Insert sample data into the orders table
```

```
INSERT INTO orders (customer_id, total_amount, status,  
shipping_address, payment_method)  
VALUES  
    (1, 150.75, 'pending', '123 Main St, Springfield, IL',  
    'credit_card'),  
    (2, 250.00, 'completed', '456 Oak St, Riverton, NY', 'paypal'),  
    (3, 99.99, 'shipped', '789 Maple Ave, Dayton, OH',  
    'bank_transfer');
```

### 3. Nginx as Reverse Proxy

You have a server running Debian 12 (see [nginx](#) in the hosts file).

Set up Nginx to proxy and load balance traffic to an app located on app servers (see [app-1](#) and [app-2](#) in the hosts file).

Prepare the setup to handle a large volume of traffic and consider potential DDoS attacks.  
For simplicity, use HTTP.

### 4. Asterisk

You have three Debian 12 servers (see records starting with [asterisk-](#) in the hosts file). Set up two of them ([asterisk-1](#) and [asterisk-2](#)) as follows: after 10 to 20 seconds of ringing, 12% of incoming calls should result in a busy status, 33% should result in no answer, and the remaining calls should be answered with tt-monkeys played for 15 to 45 seconds. Use the third server ([asterisk-balancer](#)) to configure load balancing for the first two servers: [asterisk-1](#) should handle 80% of the calls, while [asterisk-2](#) should handle the remaining 20%.