Here is the rewritten **Orgo v2 Blueprint Section 4 - Deployment Plan**, incorporating the requested corrections:

## **Section 4: Deployment Plan**

This section provides a step-by-step guide to deploying Orgo in various environments. It now includes Kubernetes scaling examples to support horizontal scaling for high-volume workflows and integrates Redis/RabbitMQ configuration steps for task queue scalability.

### **4.1 Purpose of the Deployment Plan**

**Objective:**

* Provide a structured roadmap for deploying Orgo on-premise, in hybrid setups, or cloud environments.

**Outcome:**

* A fully operational platform tailored to an organization’s infrastructure, including offline capabilities, role-based configurations, and scalable task handling.

### **4.2 Deployment Environments**

1. **On-Premise Deployment:**
   * Suitable for organizations requiring high control over data (e.g., sensitive industries like healthcare or government).
2. **Hybrid Deployment:**
   * Combines on-premise storage with optional cloud backups for redundancy.
3. **Cloud Deployment:**
   * Ideal for scalable setups with integrated cloud storage solutions.

### **4.3 Deployment Prerequisites**

1. **Hardware Requirements:**
   * **Minimum:** Dual-core 2.5 GHz processor, 8 GB RAM, 50 GB HDD.
   * **Recommended:** Quad-core 3.5 GHz processor, 16 GB RAM, 100 GB SSD.
2. **Software Requirements:**
   * **Operating Systems:** Windows Server 2016+, Ubuntu 20.04/CentOS 7+, macOS 10.14+ (for smaller setups).
   * **Dependencies:** Python 3.9+, PostgreSQL 12+, SQLite (offline operations), Redis/RabbitMQ (high-volume task queuing).
3. **Network Requirements:**
   * Secure email server with SMTP and IMAP/POP3 enabled.
   * TLS support for encrypted email transmission.

### **4.4 Deployment Steps**

1. **Setting Up the Environment:**
   * Install dependencies:

sudo apt install python3 python3-pip postgresql sqlite3 redis-server

* + Set up a virtual environment:

python3 -m venv orgo-env

source orgo-env/bin/activate

1. **Configuring the Email Server:**
   * Example Configuration (config.yaml):

smtp:

host: "smtp.organization.com"

port: 587

username: "orgo@organization.com"

password: "securepassword"

imap:

host: "imap.organization.com"

port: 993

username: "orgo@organization.com"

password: "securepassword"

1. **Database Initialization:**
   * PostgreSQL Setup:

CREATE DATABASE orgo;

CREATE USER orgouser WITH ENCRYPTED PASSWORD 'securepassword';

GRANT ALL PRIVILEGES ON DATABASE orgo TO orgouser;

* + SQLite (Offline Mode):

sqlite3 orgo\_offline.db < schema.sql

1. **Deploying Orgo Components:**
   * Clone the repository:

git clone https://github.com/your-org/orgo.git

cd orgo

pip install -r requirements.txt

python setup.py

1. **Configuring the Rule Engine:**
   * Define routing rules in YAML (/config/rules.yaml):

rules:

- condition: "subject contains 'urgent'"

action:

route\_to: "maintenance@organization.com"

escalate\_after: "2 hours"

1. **Starting the Services:**
   * Start the main application:

python main.py

* + Start task queues:

celery -A tasks worker --loglevel=info

### **4.5 Kubernetes Scaling for High-Volume Workflows**

1. **Horizontal Pod Autoscaling:**
   * Configure Kubernetes autoscaling:

apiVersion: autoscaling/v2

kind: HorizontalPodAutoscaler

metadata:

name: orgo-hpa

spec:

scaleTargetRef:

apiVersion: apps/v1

kind: Deployment

name: orgo

minReplicas: 2

maxReplicas: 10

metrics:

- type: Resource

resource:

name: cpu

targetAverageUtilization: 70

1. **Redis/RabbitMQ Integration:**
   * Example Kubernetes Deployment (deployment.yaml):

apiVersion: apps/v1

kind: Deployment

metadata:

name: redis-deployment

spec:

replicas: 3

selector:

matchLabels:

app: redis

template:

metadata:

labels:

app: redis

spec:

containers:

- name: redis

image: redis:6.2

ports:

- containerPort: 6379

### **4.6 Testing and Validation**

1. **Email Parsing Test:**
   * Send a sample email and verify parsing results:

Parsed Email:

Sender: secretary@organization.com

Subject: Water Leak in Room 102

Keywords: [urgent, leak]

Routed To: maintenance@organization.com

1. **Task Queue Load Test:**
   * Simulate high task volume with Redis:

redis-benchmark -n 100000

1. **Offline Functionality:**
   * Disconnect the system and ensure .pst file processing during downtime.

### **4.7 Deliverables**

1. **Deployment Scripts:** Ready-to-use scripts for setting up Orgo.
2. **Configuration Files:** Sample YAML files for email server, rule engine, and Kubernetes scaling.
3. **Testing Templates:** Sample workflows for simulation and validation.

### **Summary**

This deployment plan provides detailed steps for deploying Orgo in any environment, supporting horizontal scaling with Kubernetes and task queue scalability with Redis/RabbitMQ. Let me know if further refinements are needed.