To set up the proof-of-concept system for TRDL, we will follow the given requirements and design it to run on a public cloud infrastructure using a container orchestrator. Here is a plan outlining the steps required to complete the exercise:

1. **System Architecture Design:**
   * The system will consist of a containerized HTTP server that responds to GET requests.
   * For simplicity, we will use a single container running a lightweight web server to handle the HTTP requests.
2. **Containerization and Orchestration:**
   * We will use Docker for containerization to package the TRDL service.
   * Kubernetes will be the container orchestrator of choice to manage the deployment and scaling of the containers.
   * The Docker image will be hosted in a container registry accessible by Kubernetes.
3. **Deployment Steps:**
   * Set up a Kubernetes cluster on the chosen public cloud provider (e.g., AWS, Google Cloud).
   * Deploy the Kubernetes cluster with multiple worker nodes for high availability.
   * Configure the necessary network settings, such as load balancers and public IP addresses.
   * Create a Kubernetes deployment manifest to define the deployment details of the TRDL container.
   * Apply the deployment manifest to deploy the TRDL container on the Kubernetes cluster.
   * Verify that the TRDL service is accessible by sending a GET request to the exposed service endpoint.
4. **Developer Documentation:**
   * Create documentation for developers on how to extend the implementation and run the system in the defined environment.
   * Explain the steps to package the code as a Docker image.
   * Provide instructions on how to deploy the Docker image to a container registry.
   * Document the Kubernetes deployment manifest structure and how to customize it.
   * Explain how to apply the deployment manifest to deploy the container on the Kubernetes cluster.
5. **Automated Testing:**
   * Implement unit tests for the code to ensure its correctness.
   * Set up a continuous integration/continuous deployment (CI/CD) pipeline for automated testing and deployment.
   * Integrate the unit tests into the CI/CD pipeline to run them automatically before deploying the container.
6. **Report and Documentation:**
   * Compile a detailed report outlining the system setup and deployment plan.
   * Document the reasoning behind technical decisions made during the implementation.
   * Provide insights on what would be done differently for a production system.
   * Describe the approach to monitoring the system in production.
   * Discuss the strategy for upgrading the system.
   * Outline a path-to-production plan for the system.

Now, let's estimate the time required for each task:

1. System Architecture Design: 1 hour
2. Containerization and Orchestration: 2 hours
3. Deployment Steps: 4 hours
4. Developer Documentation: 3 hours
5. Automated Testing: 3 hours
6. Report and Documentation: 7 hours

Please note that these time estimates are approximate and can vary based on individual experience and familiarity with the technologies used.

Once the tasks are completed, the deliverables will include:

* A report detailing the system setup and deployment plan.
* Developer documentation explaining how to extend and deploy the system.
* CI/CD pipeline configuration for automated testing and deployment.
* Kubernetes deployment manifest for the TRDL service.