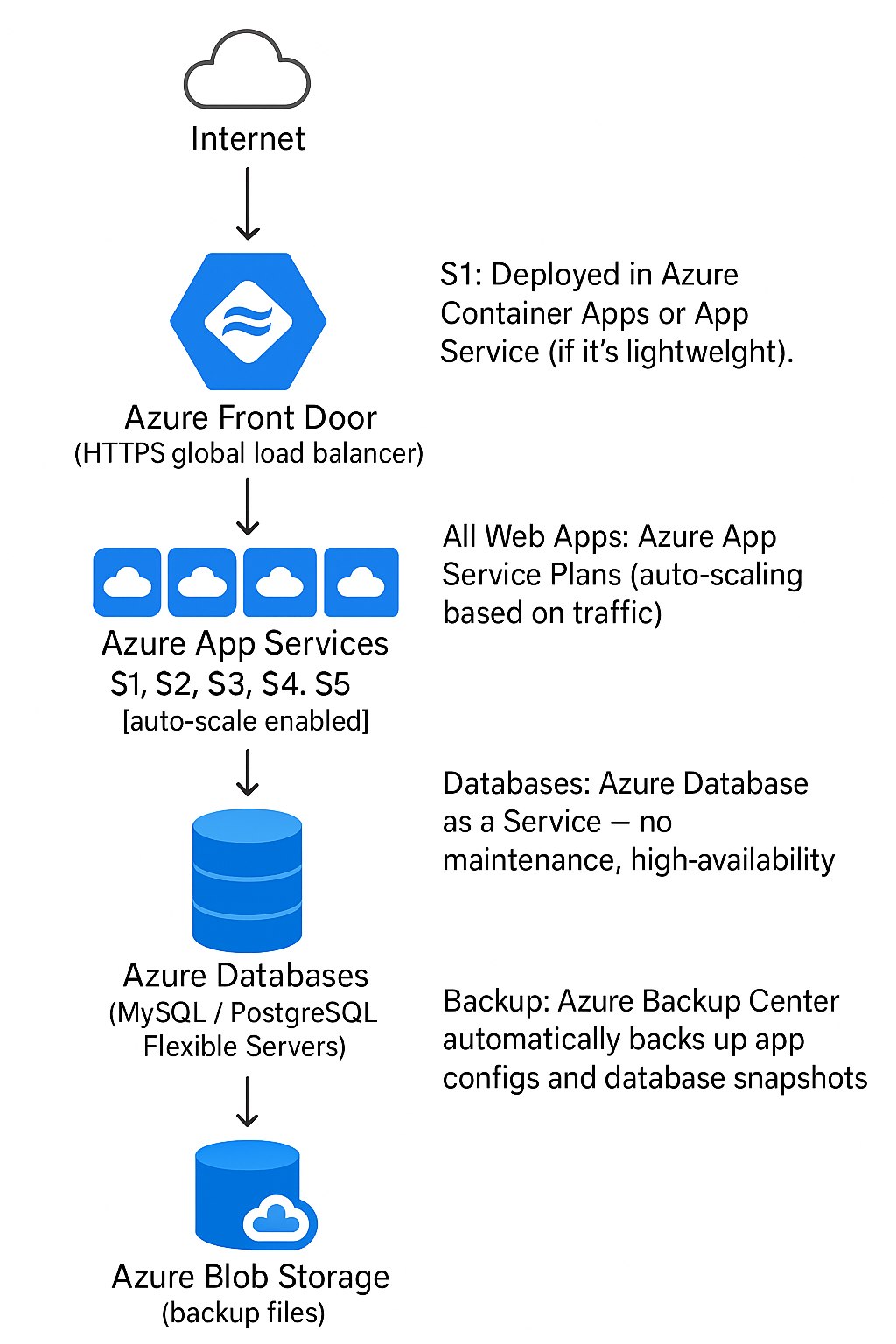
Problem 1: Using Azure PAAS Component

## Azure Services Mapping (No Infrastructure)

|  |  |  |
| --- | --- | --- |
| **System** | **Original** | **Fully Azure PaaS Equivalent** |
| S1 (Linux App + MySQL) | Standalone app | Azure Container Apps (or Azure App Service) + Azure Database for MySQL Flexible Server |
| S2 (PHP/MySQL Web app) | PHP app + MySQL | Azure App Service (PHP) + Azure Database for MySQL Flexible Server |
| S3 (PHP/PostgreSQL Web app) | PHP app + PostgreSQL | Azure App Service (PHP) + Azure Database for PostgreSQL Flexible Server |
| S4 (Python/Django/SQLite3 App) | Python/Django + SQLite | Azure App Service (Python) + Azure Database for PostgreSQL (replace SQLite) |
| S5 (Java Web App) | Java App + Data from S3/S4 | Azure App Service (Java) + Azure Database for PostgreSQL Flexible Server |
| Backup | Bacula/Restic manual | Azure Backup Center + Azure Blob Storage (automated backups for all apps & databases) |

## Azure System Architecture (Fully Serverless)



## Azure Blob Storage (backup files)

* **S1:** Deployed in Azure Container Apps or App Service (if it's lightweight).
* **All Web Apps:** Azure App Service Plans (auto-scaling based on traffic).
* **Databases:** Azure Database as a Service — no maintenance, high-availability.
* **Backup:** Azure Backup Center automatically backs up app configs and database snapshots.

## Azure Network & Security Design

|  |  |
| --- | --- |
| **Component** | **Purpose** |
| Azure VNET Integration | Private access to databases from app services |
| Azure Front Door | Global SSL, DDoS protection, auto-failover |
| Azure Private Link | Private networking between app services and databases |
| Azure Key Vault | Secrets/certificates management |
| Azure Backup Center | Full app + database backup management |
| Azure Monitor + App Insights | Centralized performance/logs monitoring |
| Azure Security Center | Threat protection, compliance reporting |

* **No public database access** — only through internal Azure links.
* **App-to-DB traffic encrypted internally** via Private Endpoint.

## Azure Cost Estimation (Fully PaaS)

|  |  |  |
| --- | --- | --- |
| **Service** | **Type** | **Estimated Cost (Monthly USD)** |
| Azure App Service Plan (5 apps) | Premium P1v3 (shared across apps) | ~$150 |
| Azure Database for MySQL (for S1/S2) | Basic Single Server | ~$30 |
| Azure Database for PostgreSQL (for S3/S4/S5) | Basic Single Server | ~$30 × 2 = $60 |
| Azure Front Door (Small traffic) | Front Door Standard | ~$35 |
| Azure Backup + Blob Storage | 500GB backup storage | ~$10 |
| Azure Monitor (Logs & Metrics) | Basic | ~$20 |

* **Total Monthly Cost:** **~ $275 - 300 USD**
* **Annual Cost (Recurring):** **~ $3,600 USD/year**

## Final Fully Managed Solution Summary

|  |  |
| --- | --- |
| **Aspect** | **PaaS Solution** |
| Compute | Azure App Service (Web Apps) |
| Databases | Azure Database for MySQL/PostgreSQL Flexible Server |
| Storage | Azure Blob Storage (Backup + Static Content) |
| Backup | Azure Backup Center |
| Frontend Access | Azure Front Door (Public with SSL) |
| Monitoring | Azure Monitor + Application Insights |
| Security | Azure Security Center + Private Link + DDoS Basic |

* No server management
* No OS patching
* Built-in Auto-scaling
* 99.95% uptime SLA on all services
* Minimal IT staff needed

## Pros of this Approach

* Fully serverless — **zero VM maintenance**.
* Highly scalable and resilient.
* Simplified cost model — pay for usage.
* Built-in security, backups, failover.

## Potential Improvements (Optional):

* Use **Azure Logic Apps** to automate data flows between S1 → S2 and S3/S4 → S5.
* Use **Azure API Management** if APIs need secure publishing externally.

## Conclusion

|  |  |  |
| --- | --- | --- |
| **Comparison** | **Traditional** | **Azure Full PaaS** |
| Management | High | Very Low |
| Scalability | Manual | Automatic |
| Upfront Cost | $13K+ | $0 |
| Yearly Cost | $7K | ~$3.6K |
| Redundancy | Manual | Built-in |
| Uptime | Variable | 99.95% SLA |