

Deploying and Scaling Your First Cloud Application

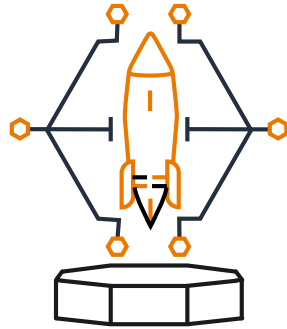
Mike Coleman – Developer Advocate, Amazon Lightsail

@mikegcoleman

What you will learn

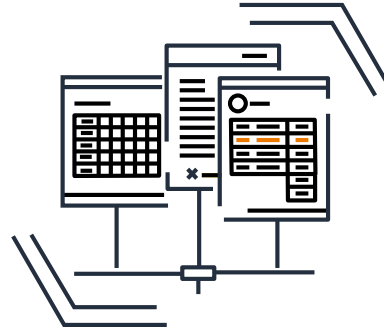
- What is Amazon Lightsail
- When to choose Lightsail
- Using Lightsail load balancers and databases
- Deploying a multi-tier application on Amazon Lightsail
- Horizontal scaling with snapshots
- Vertical scaling with EC2

Amazon Lightsail: cloud made easy



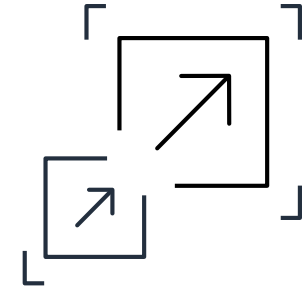
Launch

- Get started in minutes
- Low predictable pricing



Manage

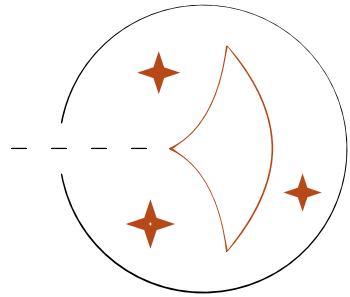
- Intuitive global console
- Robust API and CLI



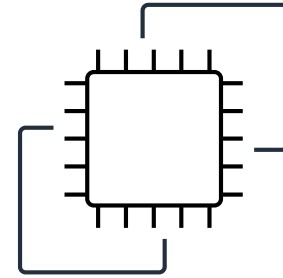
Grow

- Easily scale your ideas
- Access other AWS services

When to choose Lightsail?



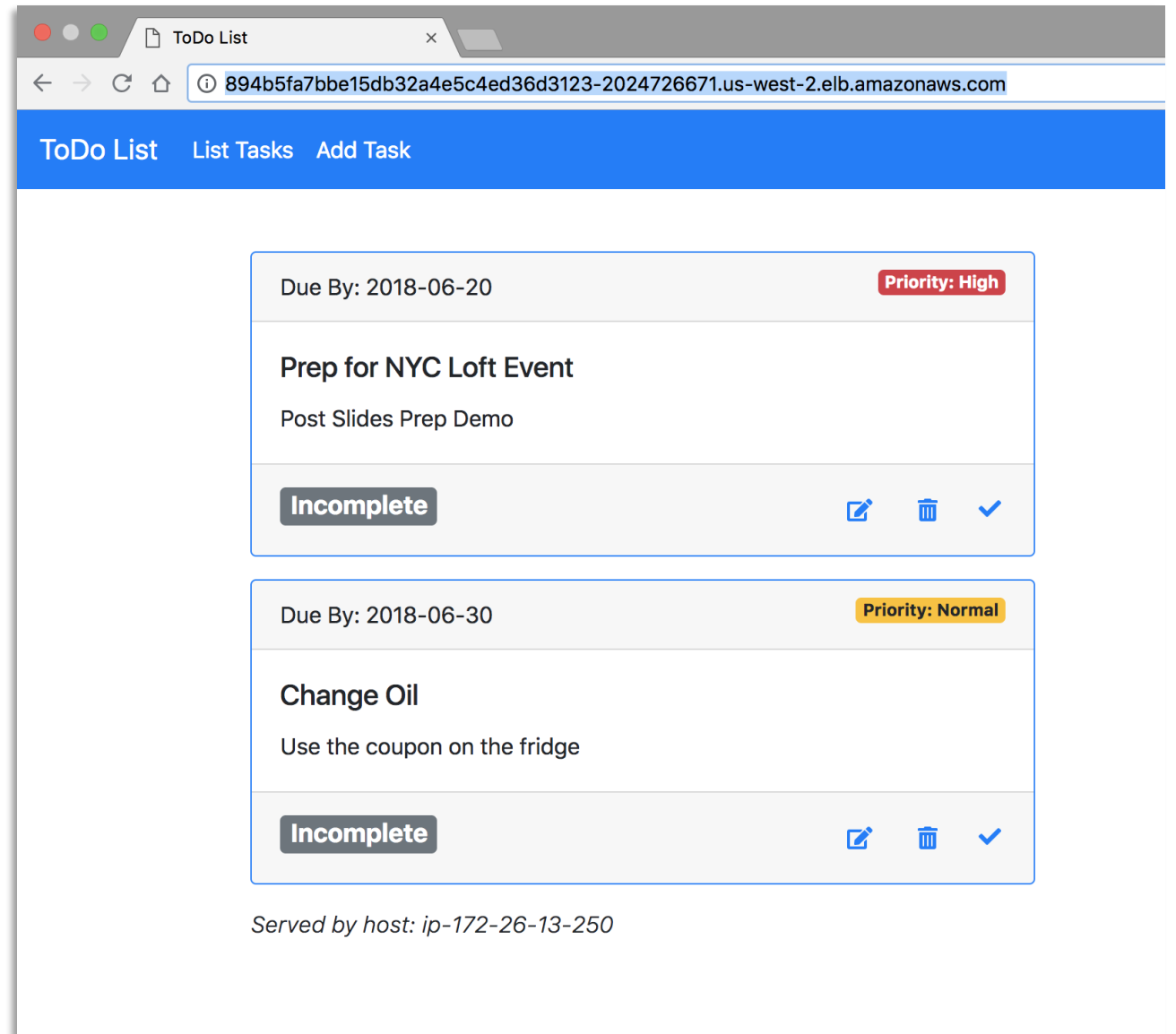
- Small-scale multi-tier apps
- Websites
- Web apps
- Testing environment
- Line-of-business software



- Large, multi-tier applications
- Workload dependent instances
- Advanced networking
- Database read replicas

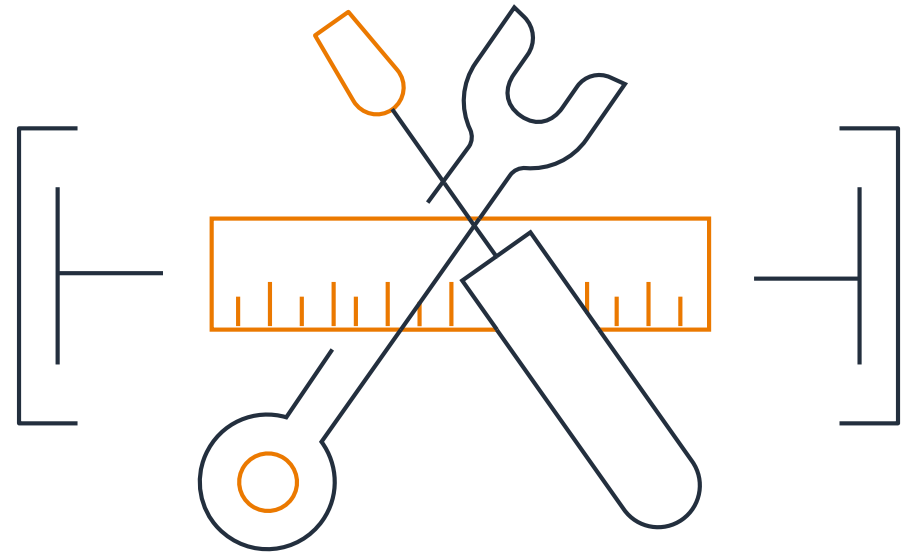
The application: simple to-do list

- PHP front-end
- MySQL back-end
- **Phase 1:**
- Deploy app into a single Lightsail instance
- **Phase 2:**
- Implement a Lightsail database
- **Phase 3:**
- Scale the web front-end
- **Phase 4:**
- Move the application to RDS / EC2

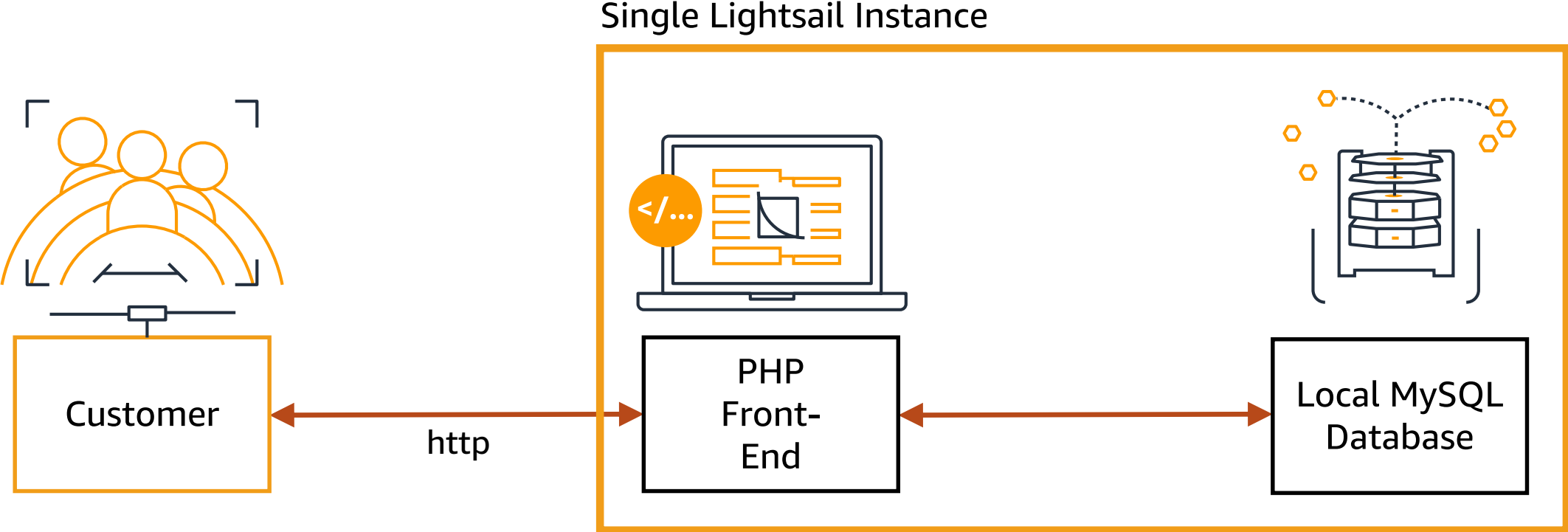


Getting our phase 1 application running

1. Create Lightsail instance
2. Clone Github repo
3. Configure database connectivity



Application architecture—version 1



Refining our application

- **Potential issues:**
 - Single point of failure
 - Responding to increasing demand
 - Database / web front end tied closely together

- **Solution:**
 - Separate database / web front end
 - Created multiple front end instances
 - Add a Lightsail load balancer

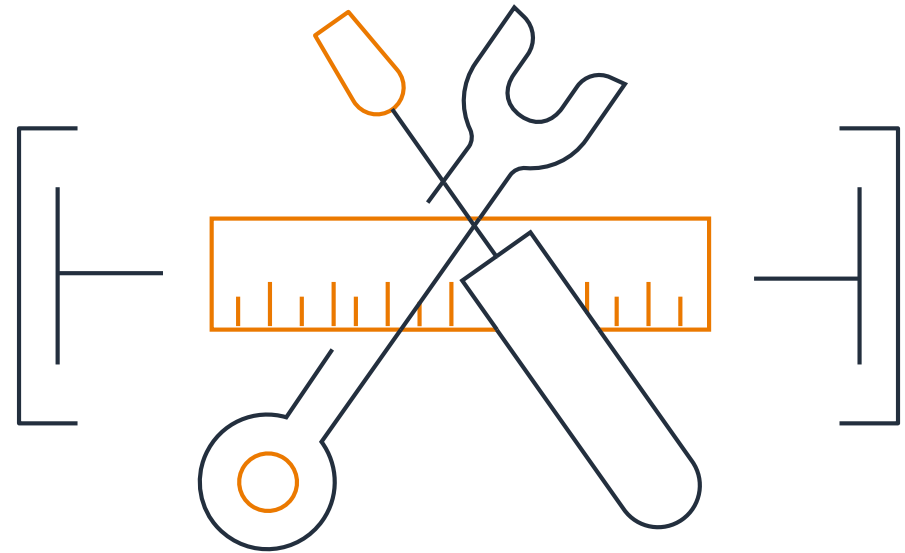
Lightsail databases



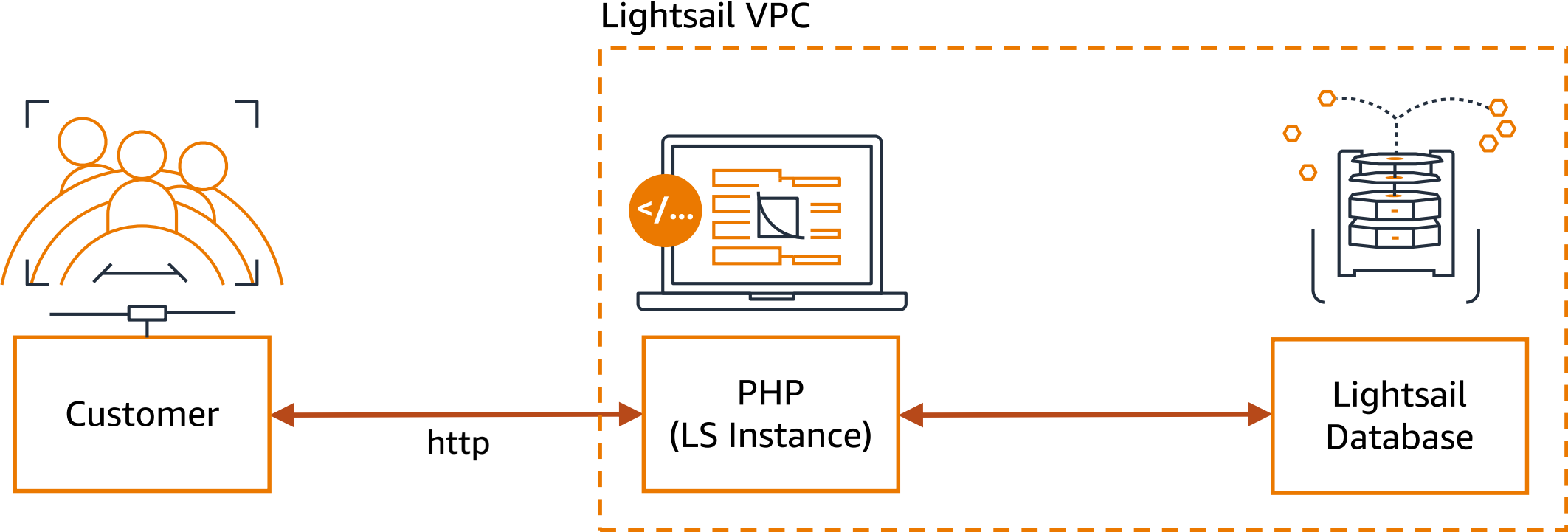
- Easy to create and manage MySQL databases
- Standard & high availability options
- Four instances sizes
- Public and private access
- Automated backups
- Fully managed

Getting our phase 2 application running

1. Create Lightsail database
2. Reconfigure the front end to point at the new Lightsail database



Application architecture—version 2



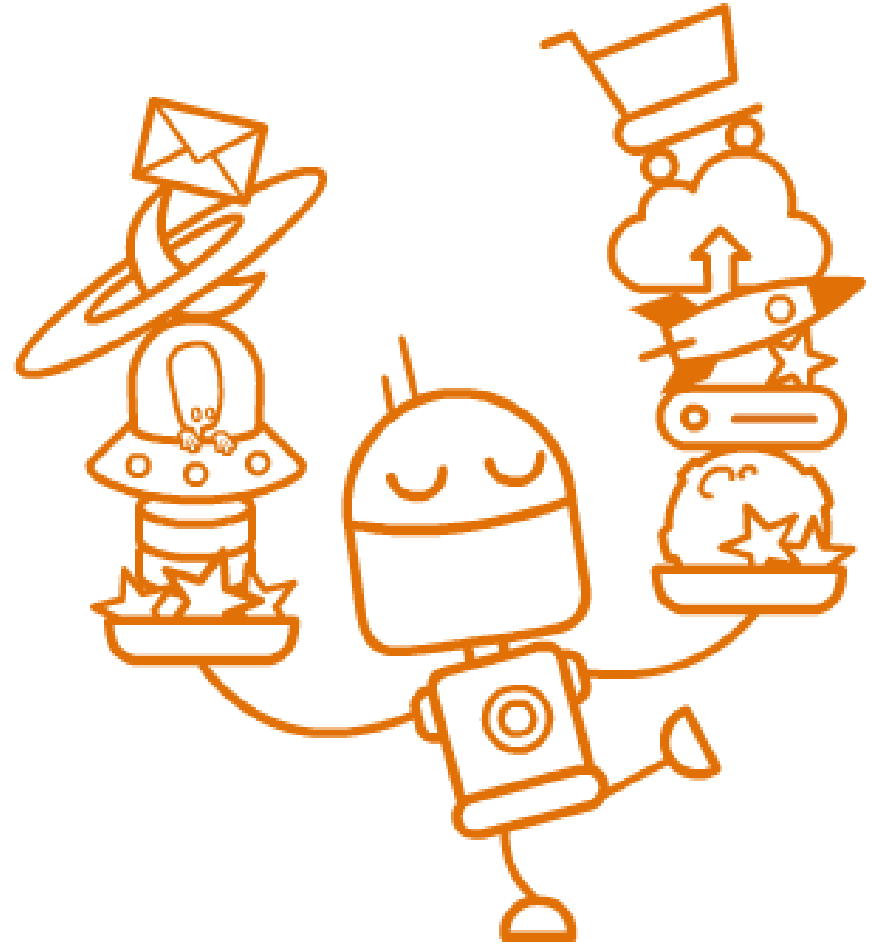
Horizontal scaling with snapshots

- Front end is stateless
- Create Lightsail snapshot
- Deploy multiple instances from snapshot
- Create Lightsail load balancer
- And front-end instances to Load balancer



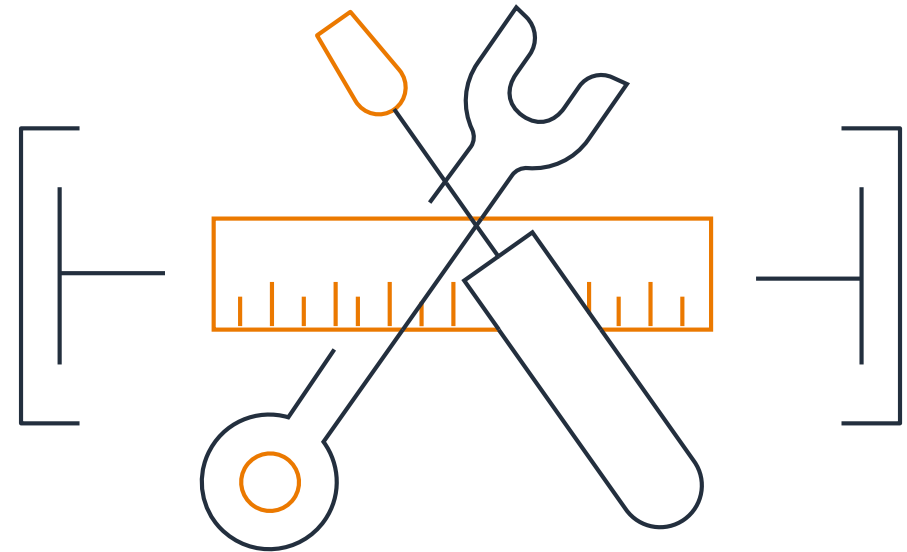
Lightsail load balancers

- Simplified Amazon load balancer implementation
- Set up in a few clicks
- Easy SSL certs
- Handles http / https traffic
- Balances across ports 80 and 443

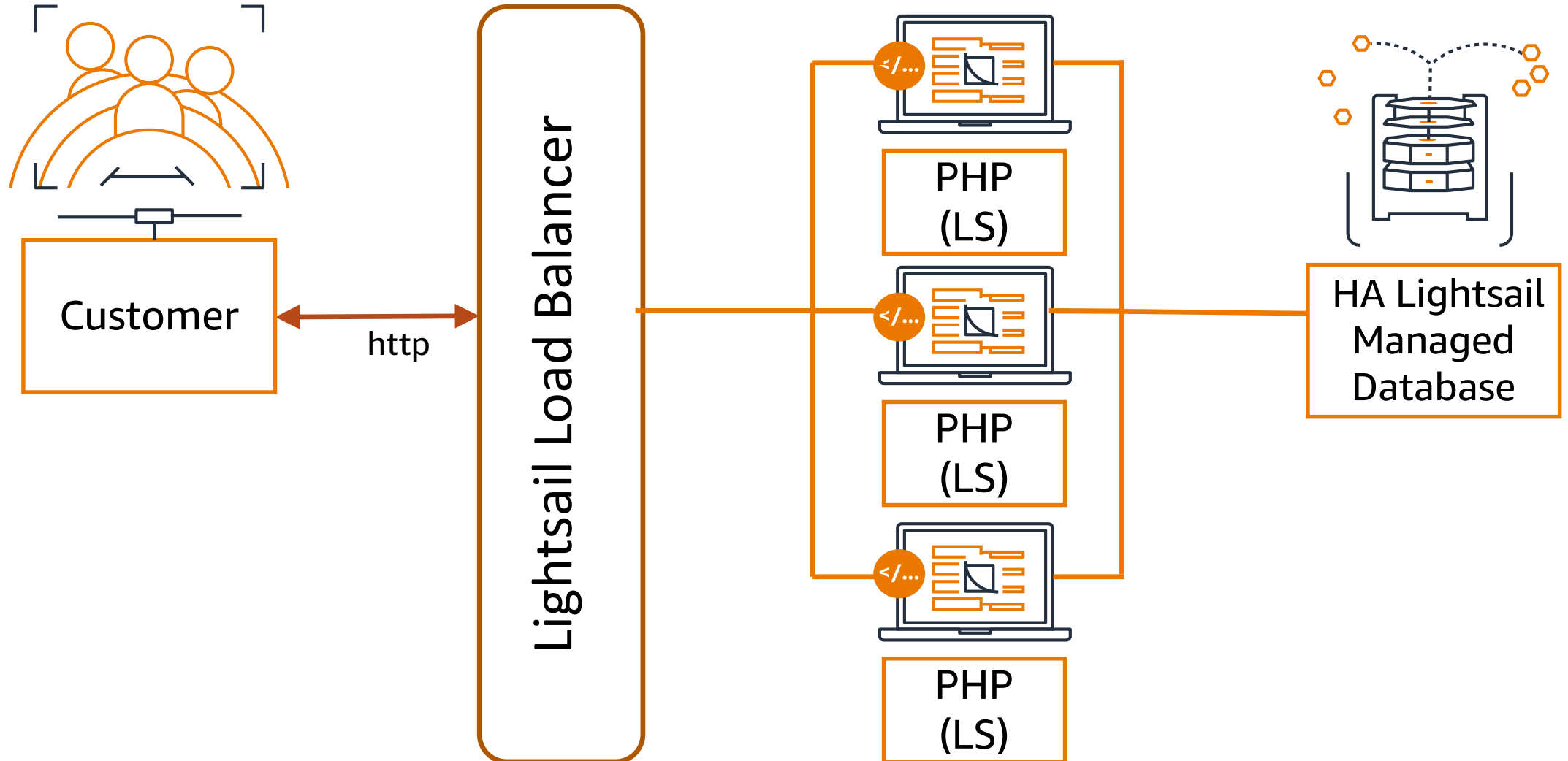


Getting our phase 3 application running

1. Create a Lightsail load balancer
2. Snapshot the web front end
3. Deploy 2 new instances from the snapshot
4. Place all three front end instances behind the load balancer
5. Optional: If you have a domain you manage, create a cname record for the load balancer



Application architecture—version 3

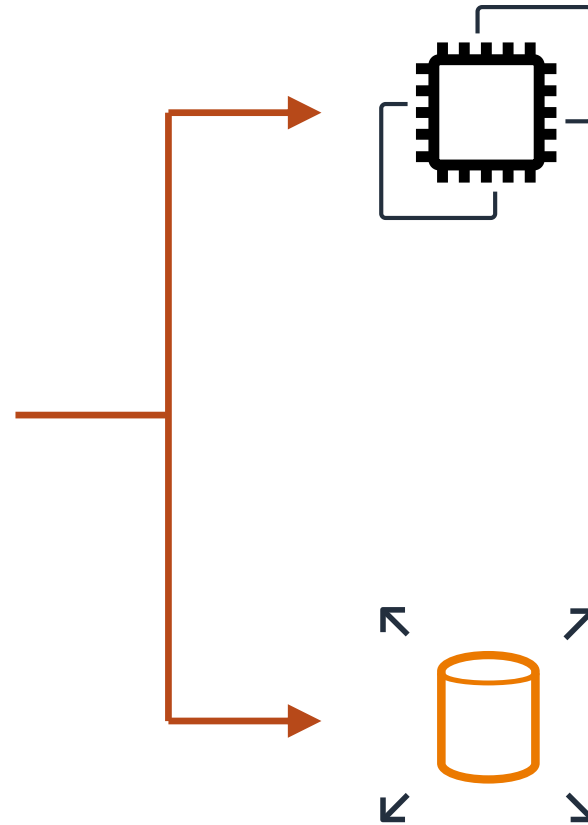


New: Upgrade to EC2



Move your data to EC2

- Snapshot instance or disk
- Choose export to EC2 from snapshot



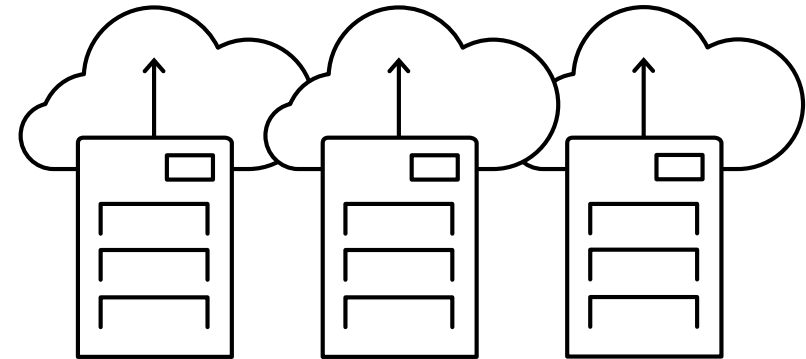
Create a new instance in EC2

- Or use EC2 export wizard (simplified set up with preconfigured VPC, etc.)
- Use EC2 creation wizard (all configuration options available)

Use EBS to re-hydrate disk snapshots

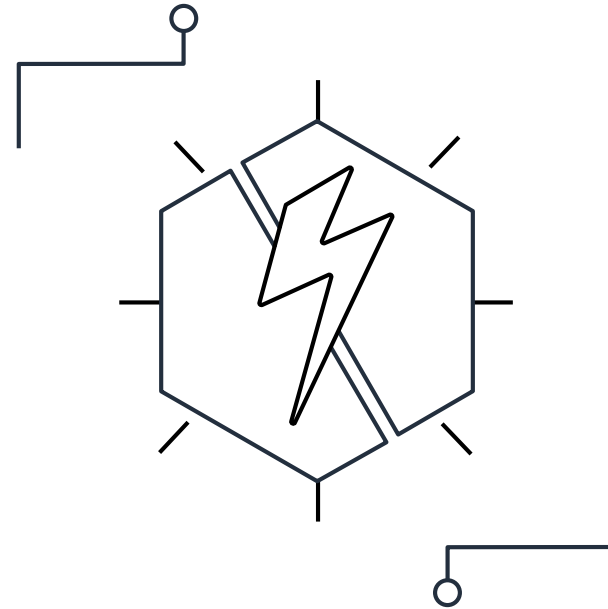
Migrating the database to RDS

1. Launch an RDS instance
2. Set Security Group (172.26.0.0/16)
3. Enable VPC peering
4. Migrate and Verify Data
5. Connect front-end to RDS

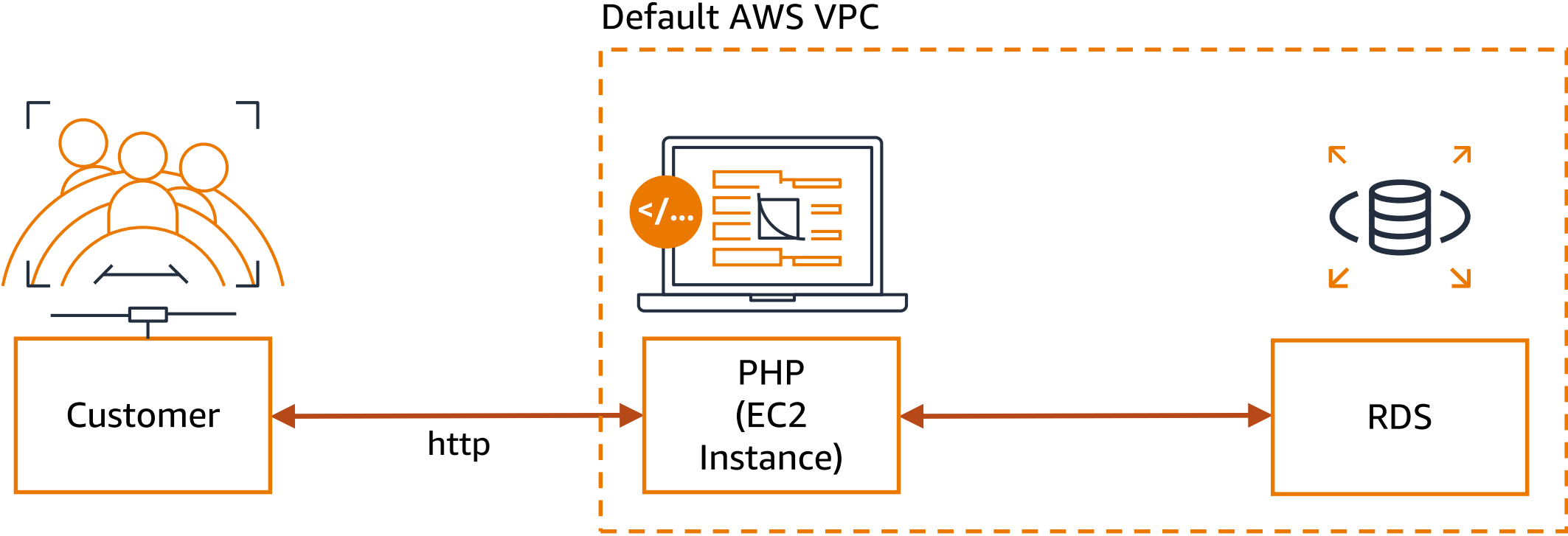


Moving your front end to EC2

1. Snapshot the Lightsail Instance
2. Create new EC2 instance with export to EC2 wizard
3. Connect EC2 front end to previously created RDS database



Application architecture—version 3 (start of prod)



Resources

<https://github.com/mikegcoleman/todo-php> ← The application

<https://reinvent.lightsailworkshop.com> ← Step-by-step walkthrough