

SINCE 2024

StrongApe

E-Commerce on Microsoft Azure

Content

1	Project Introduction	6	Documentation
2	Budget Assumptions	7	Testing & Deployment
3	Risks & Obstacles	8	Training & Support
4	Proof of Concept	9	Implementation Timeline
5	Proposed Infrastructure	10	Q & A

Client's Profile



Strong Ape Apparel is a well-established brand in Singapore with a strong physical retail presence.

Client's Profile



Limited market reach to in-store customers only.



Missed opportunities to engage with online shoppers and increase sales.



Inefficient inventory management leading to potential overstocking or stockouts.



Inability to scale operations or handle sudden spikes in popular demand.



Client's Goals



Expand customer reach



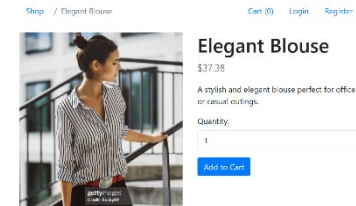
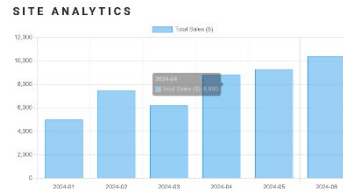
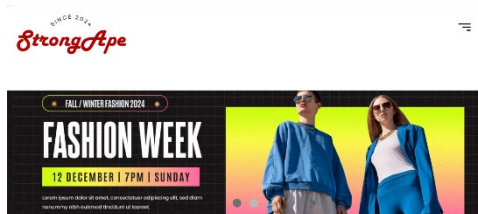
Increase online sales and engagement



Optimise inventory management



Enhance scalability



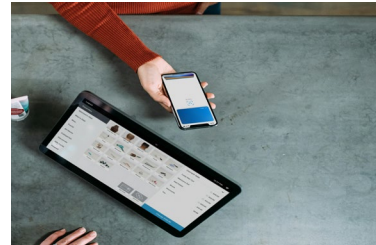
Budget Assumptions



Customer retention grew from 60% to 70% in 2024



New customers increase by 5% annually

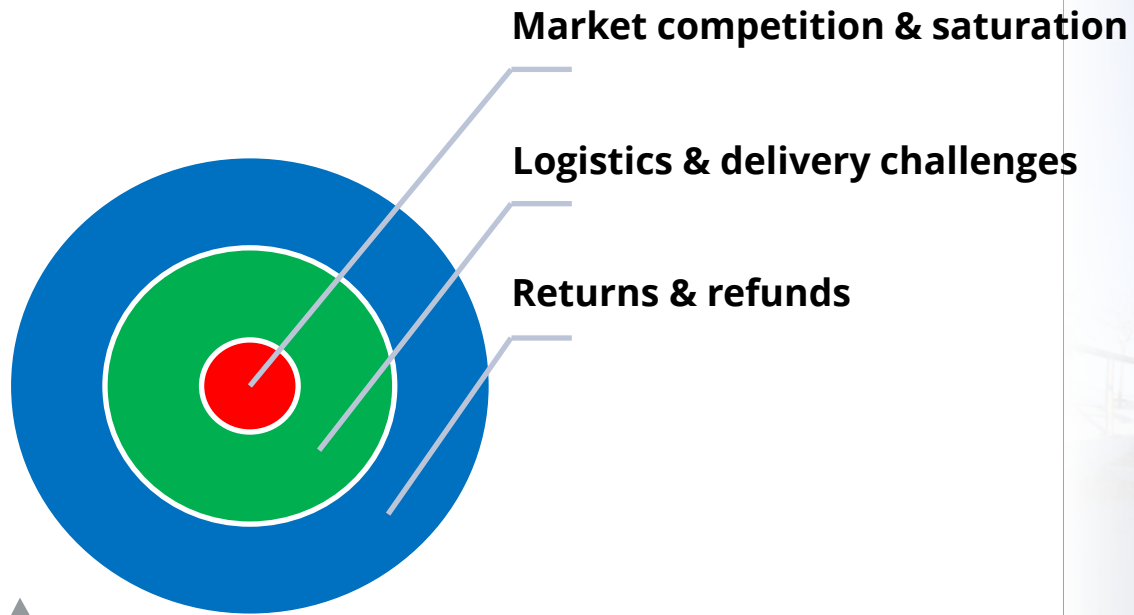


Singapore's e-commerce market grows 4.3% year-on-year



Market share will increase from 25% to 40% with an e-commerce platform

Risks



Obstacles

Data Sharing & Ownership Policies



Compliance with regulation



External party integration



Ownership and usage rights



Data security

Integration with existing systems



Data flow and synchronisation

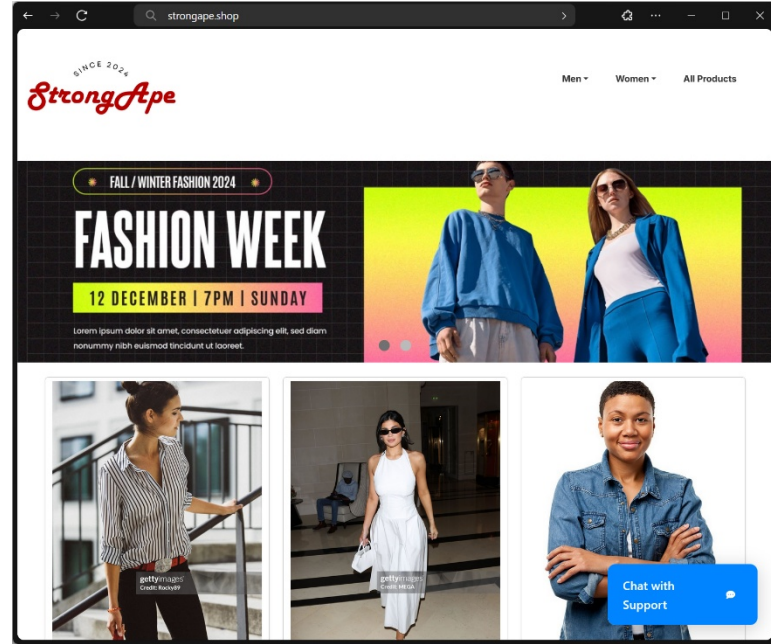


Technical difficulties in integration

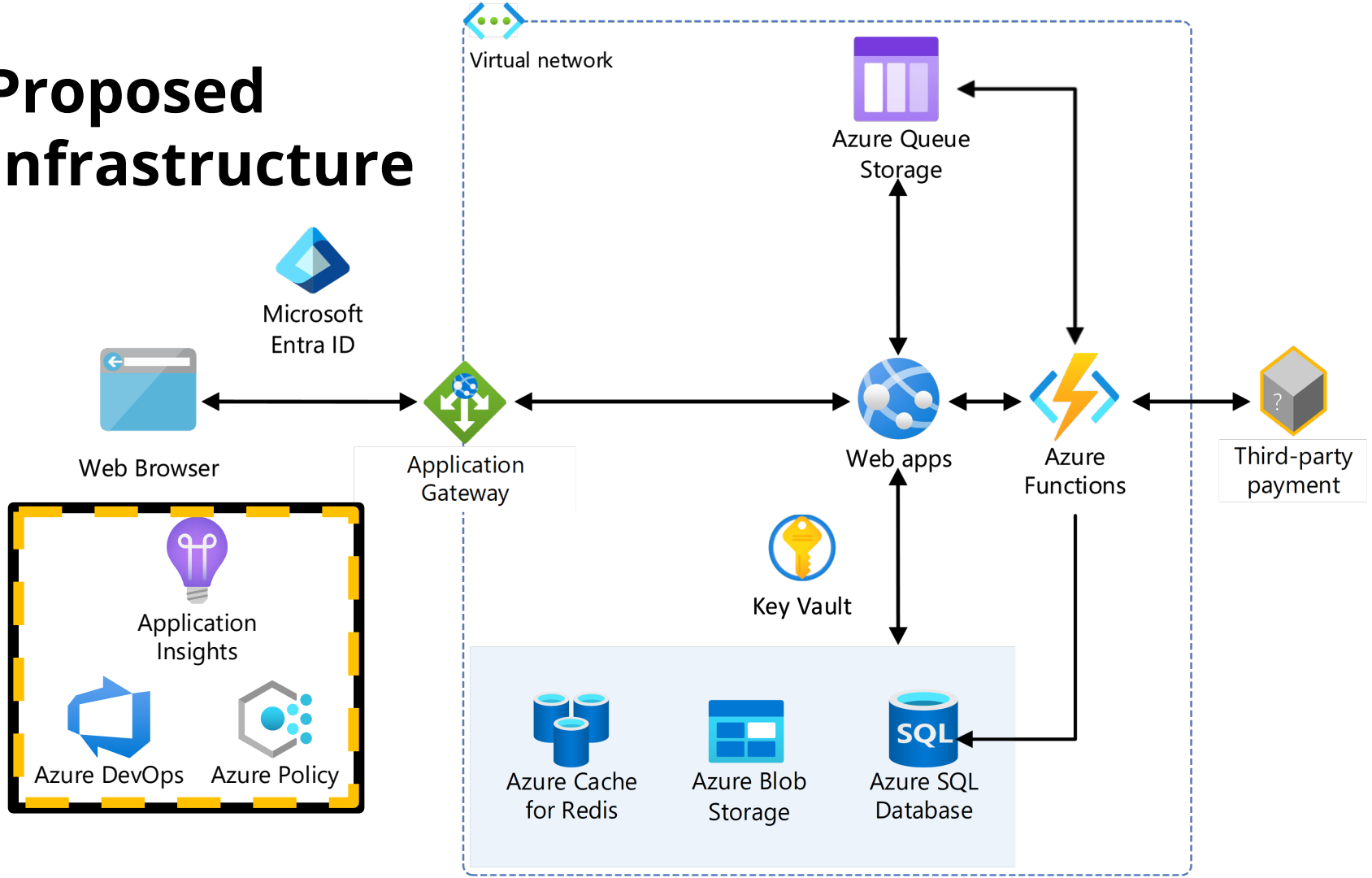


Compatibility issues

<https://strongape.shop>



Proposed Infrastructure

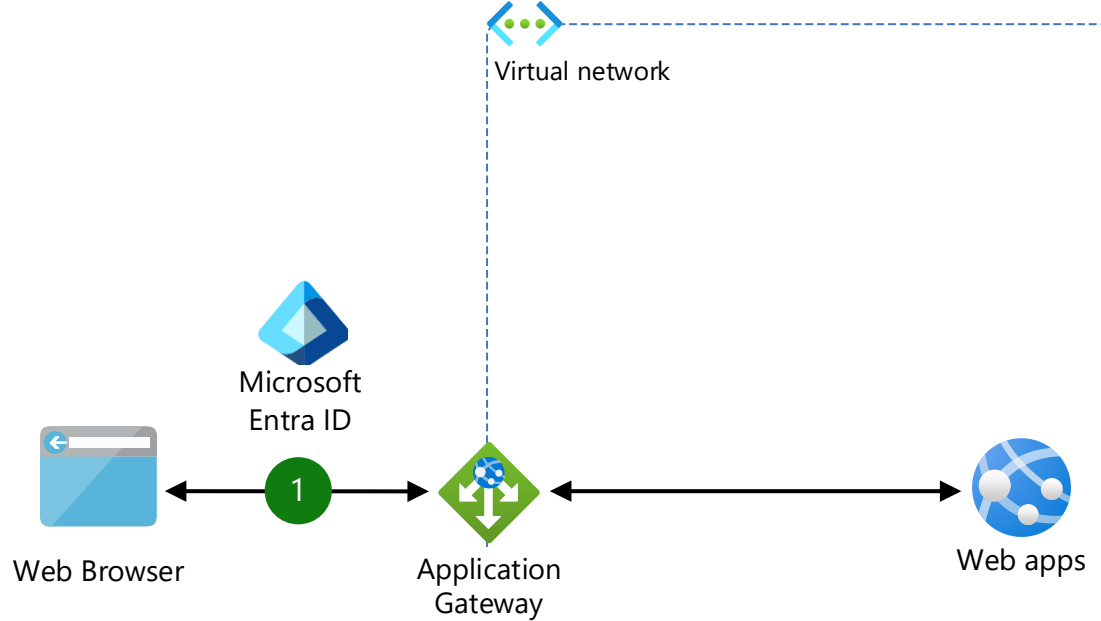
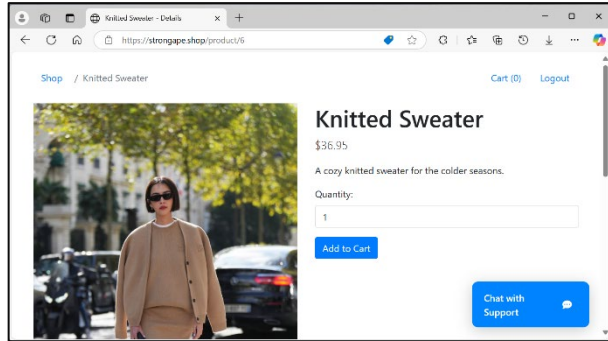


Proposed Infrastructure

Step 1

User access points:

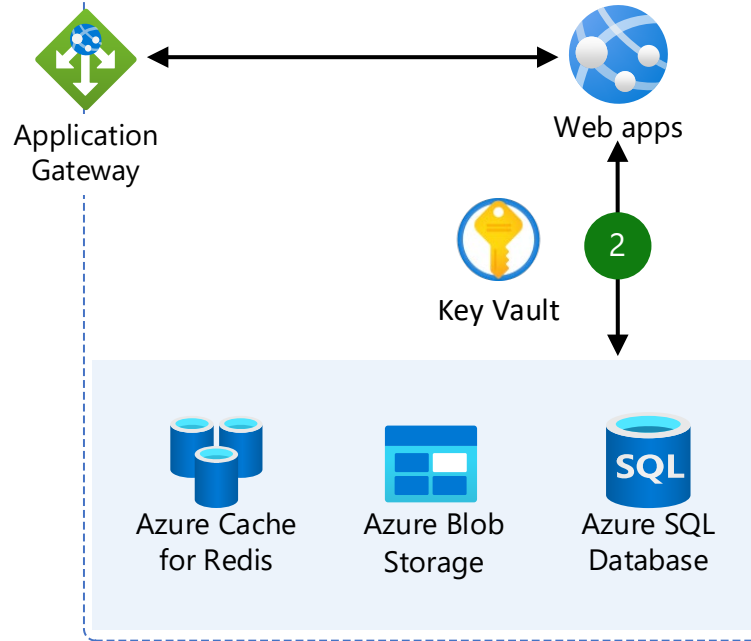
- Mobile App
- Web Browser



Proposed Infrastructure

Step 2

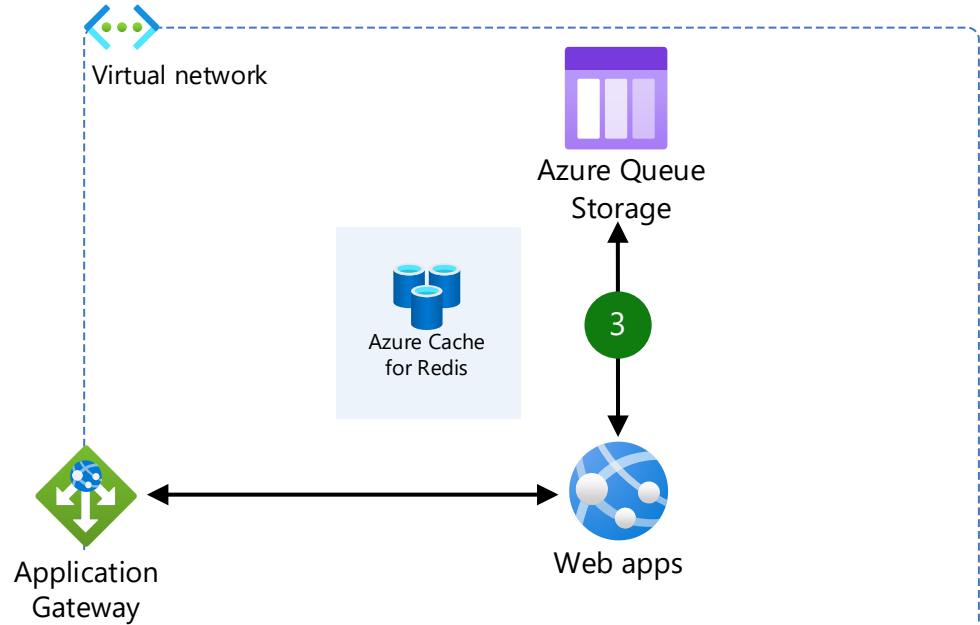
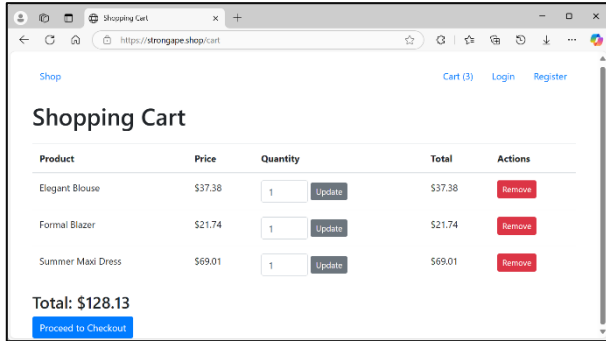
- User browse product through categories (queries SQL Database).
- Web app retrieves product catalogue (database) & images (Blob Storage).
- Page output cached in Redis for fast retrieval.



Proposed Infrastructure

Step 3

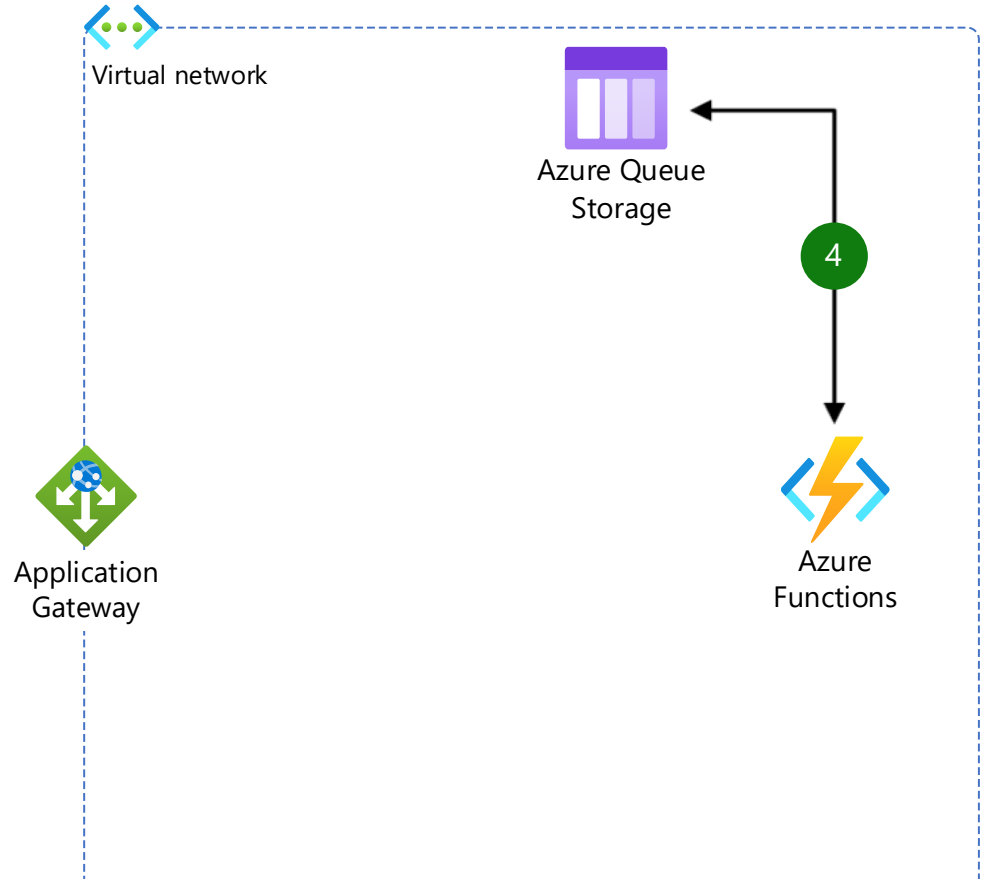
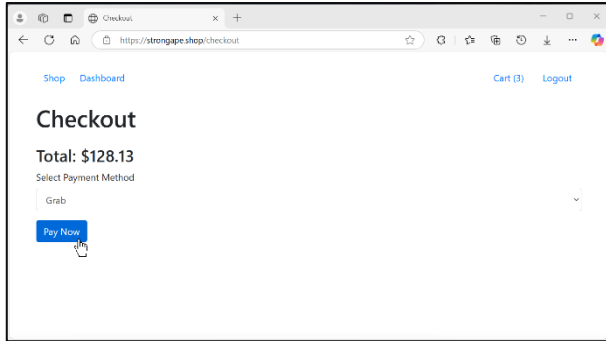
- Item added to the cart are queued in Redis, finalised in Queue Storage.



Proposed Infrastructure

Step 4

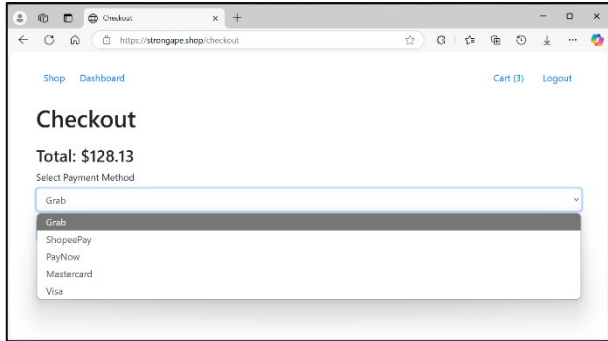
- Azure Queue Storage pushes order event to Functions.



Proposed Infrastructure

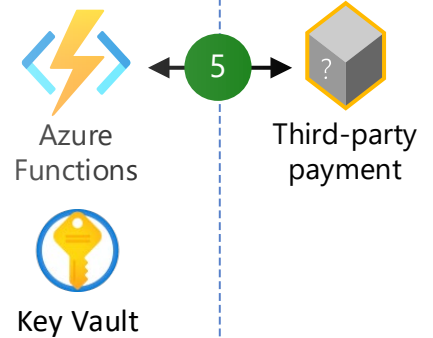
Step 5

- Azure Functions brokers payment with third-party gateway.



Virtual network

Application gateway



Proposed Infrastructure

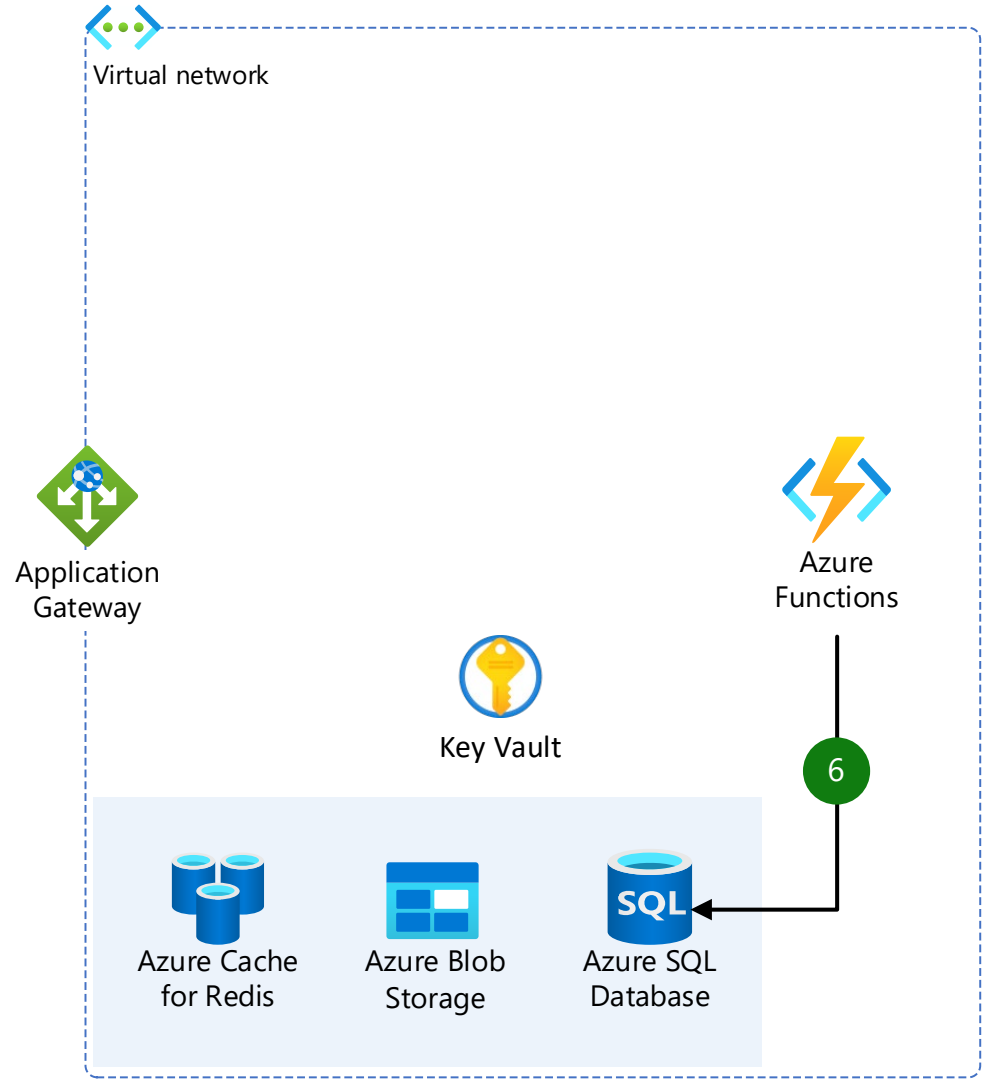
Step 6

- Azure Functions records order in SQL database.

```
mysql> select * from orders;
```

id	user_id	total_amount	created_at
1	6	72147.4	2025-01-06 04:07:57
2	4	1416.29	2025-01-06 15:27:20
3	6	1869.06	2025-01-06 17:32:48
4	7	315.389	2025-01-08 05:53:17
11	7	59.1234	2025-02-16 17:18:36
12	7	1876.77	2025-02-16 17:19:49
13	3	37.3811	2025-02-17 03:17:45
14	3	186.906	2025-02-17 03:18:27

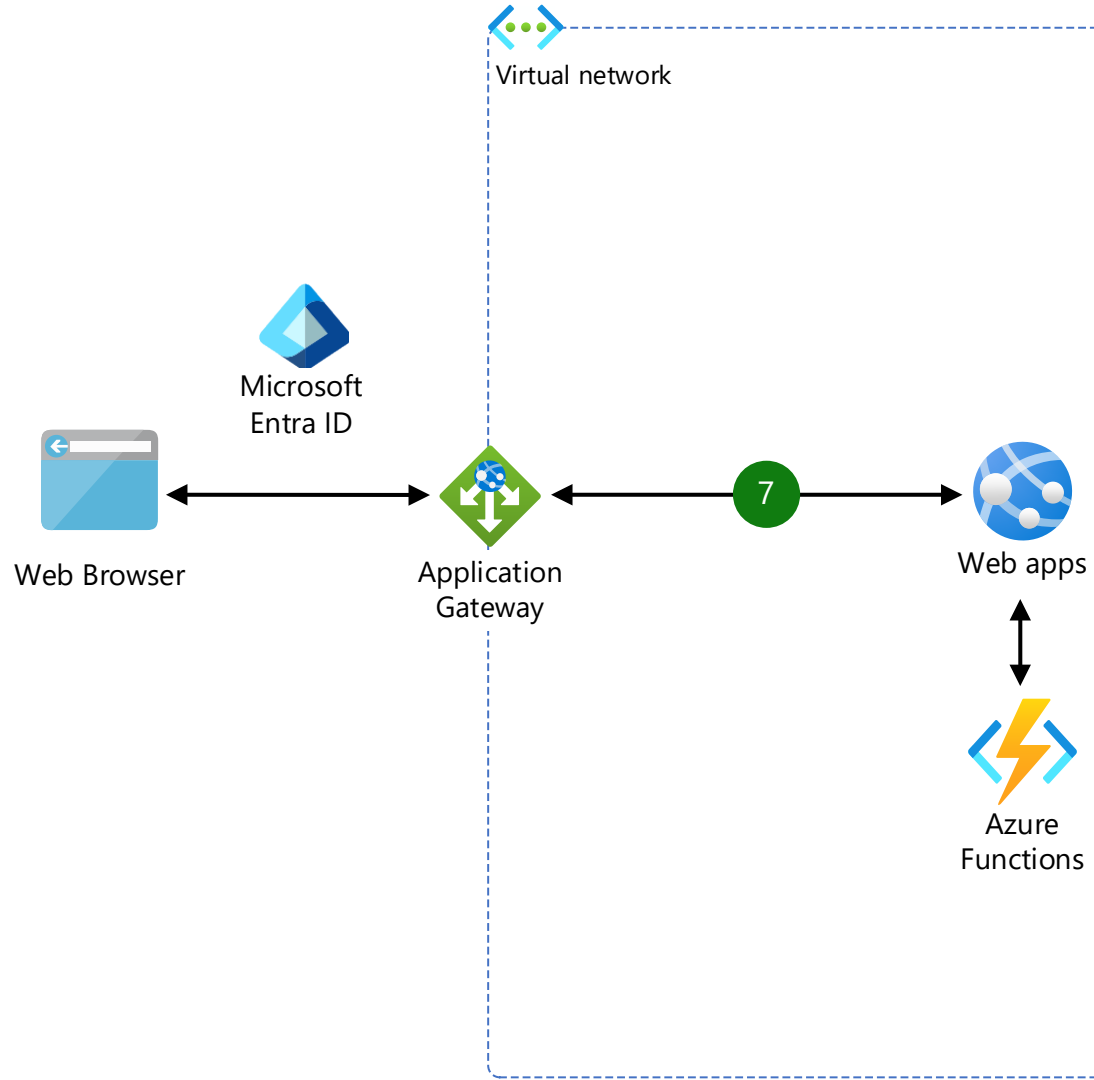
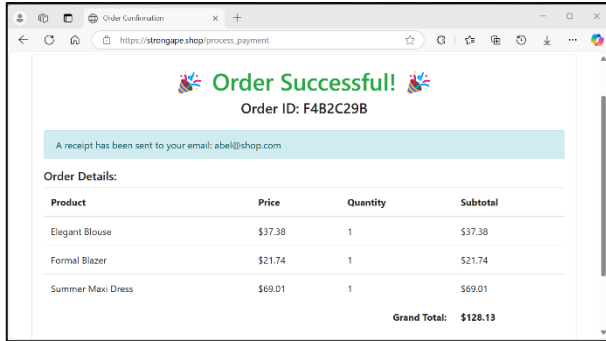
8 rows in set (0.00 sec)



Proposed Infrastructure

Step 7

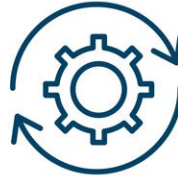
- When payment is successful, order confirmation screen is displayed.



Integration

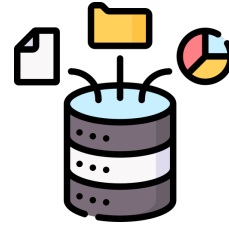


Mode of operation



BUSINESS CONTINUITY

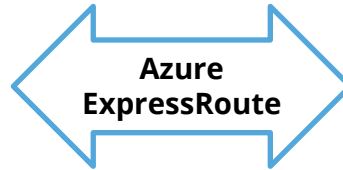
Business Continuity



Migrating old data

Integration

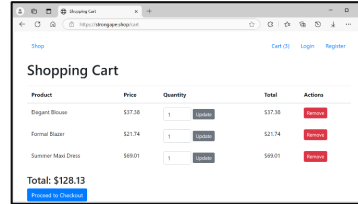
On-premise



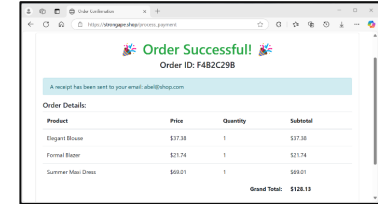
Cloud



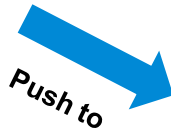
Proposed Data Sync



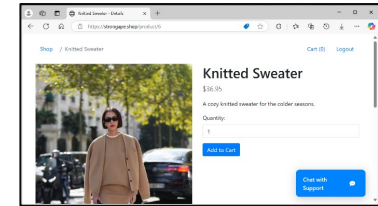
Online Order



Invoice & Shipping



Offline Order



Product & Inventory

Documentation



User Training & Guides

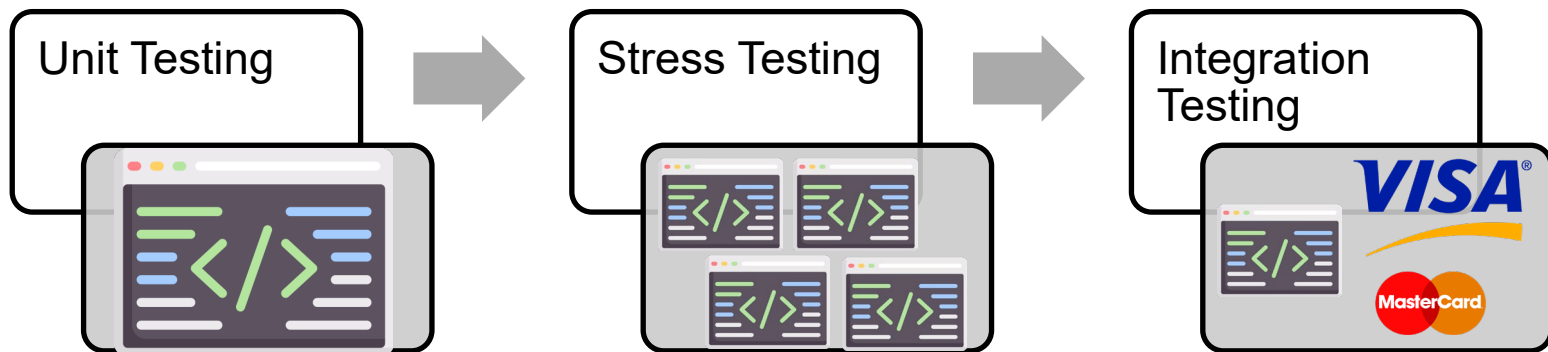


System Operations &
Troubleshooting



DevOps

Testing



Deployment



System Readiness



**Coordinating
Deployment Activities**



Milestone Completion



Training

Training & Support



User-Friendly & Low Maintenance



**Intuitive
Management**



**Turn-Key
Solution**

Support & Resources



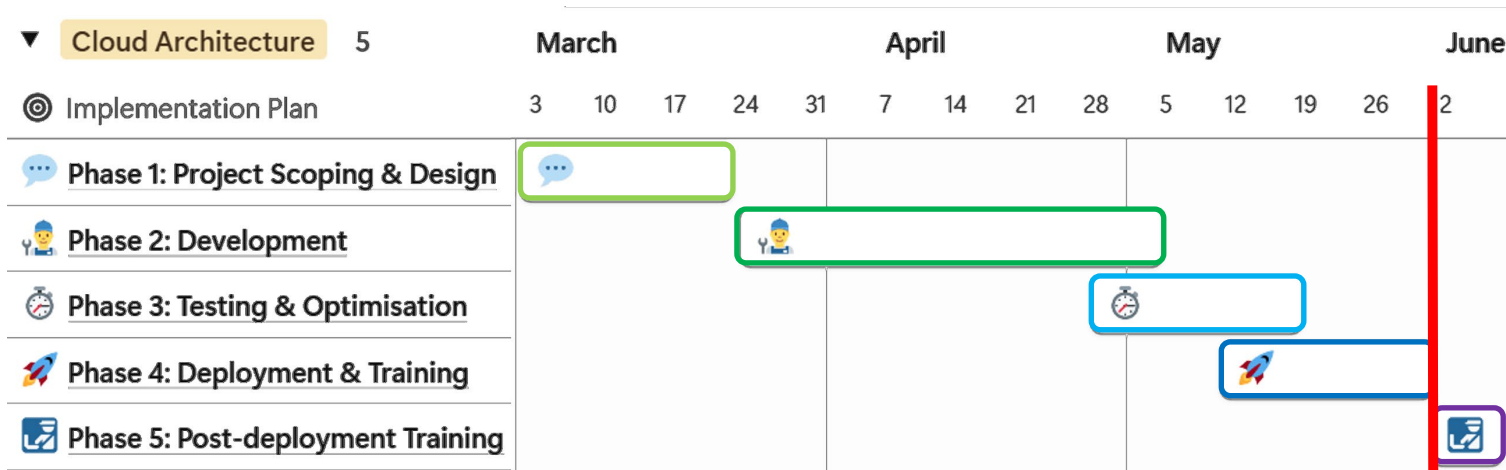
**Quick-Start Guides
& FAQs**



**24/7
Support**



Implementation Timeline



Start date: 1 Mar 25
Go-live date: 31 May 25
End date: 7 Jun 25



Thank you

Any Question?