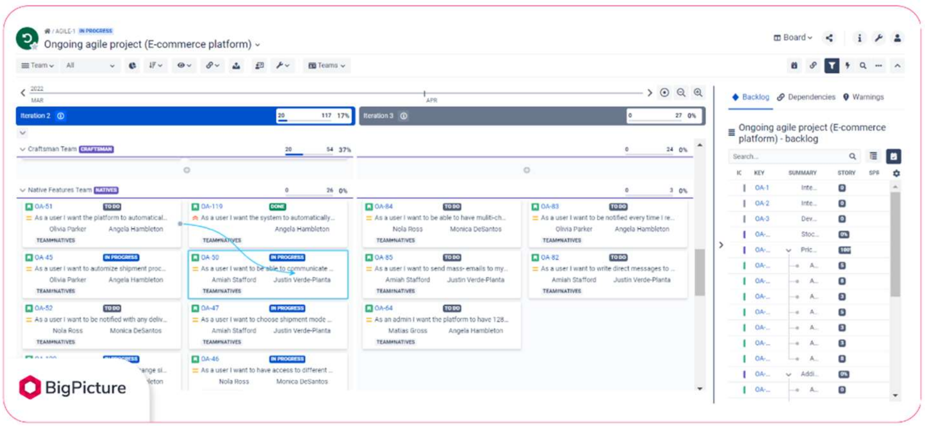
Proposal for Question 1

To properly manage Inter team dependencies, we could consider using Jira. We can consider each teams tasks as a separate mini project in which we can use Issue Links to link tasks from different teams to indicate that each task depends on another. We can also use Jira Align to better visualize the dependencies(Figure 1).


AGILE-1 IN PROGRESS
Ongoing agile project (E-commerce platform)

Iteration 2 — 20 117 17%
Iteration 3 — 0 27 0%

Craftsman Team CRAFTSMAN — 20 54 37% — 0 24 0%

Native Features Team NATIVES — 0 26 0% — 0 3 0%

OA-51 TO DO
As a user I want the platform to automatical...
Olivia Parker Angela Hambleton
TEAM#NATIVES

OA-119 DONE
As a user I want the system to automatically...
Angela Hambleton
TEAM#NATIVES

OA-84 TO DO
As a user I want to be able to have muliti-ch...
Nola Ross Monica DeSantos
TEAM#NATIVES

OA-83 TO DO
As a user I want to be notified every time I re...
Olivia Parker Angela Hambleton
TEAM#NATIVES

OA-45 IN PROGRESS
As a user I want to automize shipment proc...
Olivia Parker Angela Hambleton
TEAM#NATIVES

OA-50 IN PROGRESS
As a user I want to be able to communicate ...
Amiah Stafford Justin Verde-Planta
TEAM#NATIVES

OA-85 TO DO
As a user I want to send mass- emails to my...
Amiah Stafford Justin Verde-Planta
TEAM#NATIVES

OA-82 TO DO
As a user I want to write direct messages to ...
Amiah Stafford Justin Verde-Planta
TEAM#NATIVES

OA-52 TO DO
As a user I want to be notified with any deliv...
Nola Ross Monica DeSantos
TEAM#NATIVES

OA-47 IN PROGRESS
As a user I want to choose shipment mode ...
Amiah Stafford Justin Verde-Planta
TEAM#NATIVES

OA-64 TO DO
As an admin I want the platform to have 128...
Matias Gross Angela Hambleton
TEAM#NATIVES

OA-46 IN PROGRESS
As a user I want to have access to different ...
Nola Ross Monica DeSantos

Ongoing agile project (E-commerce platform) - backlog

FIGURE 1: BIGPICTURE VISUALIZING DEPENDENCY

To reduce breaking change, we can consider:

1) Have a strong documentation of the general practice when using shared resources.
2) Assigning a specific person from each team to review changes in code to push back against accidental breaking changes.
3) Using Jenkins to perform Automated API Testing.
4) Setting up recurring check in meetings to align on the changes and dependencies

Proposal for Question 2

One of the main considerations during application migration is to know the traffic details of the current application. Upon understanding the traffic, we will know when the least user traffic is expected and that will be the day for the migration to happen. This will reduce the probability of users that will face a potential downtime. Apart from that, introducing caching service such as Amazon CloudFront for caching static contents will also reduce the user traffic downtime.

As for the actual data migration, lets take the 3 Tier Web Application(Figure 2) below as an example.

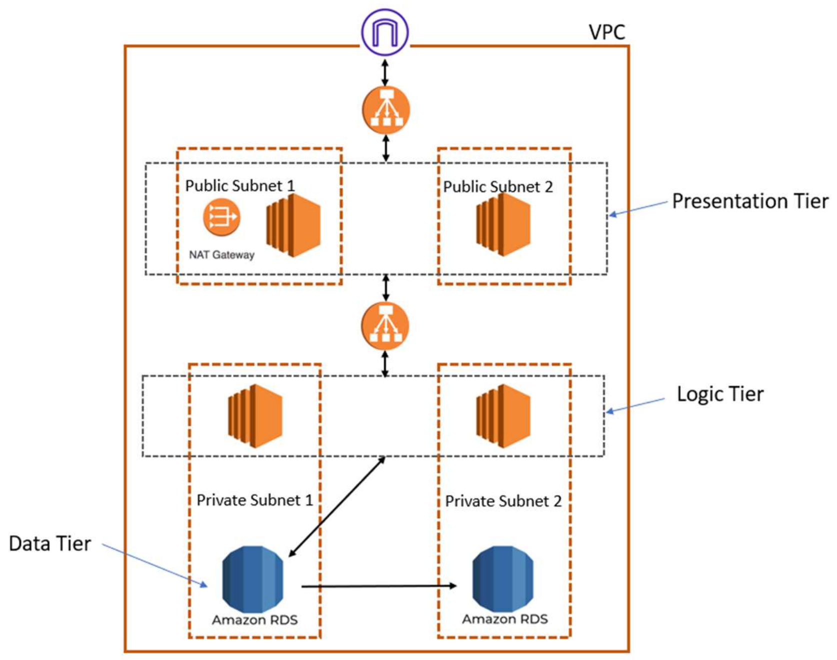


FIGURE 2: 3-TIER WEB APPLICATION

Figure 1 shows a simple 3-Tier Web Application that consists of Presentation Tier, Logic Tier, and Data Tier with the Data Tier has Multi-AZ deployment. Prior to the application, it is best practice to simulate the new application in a different environment to ensure all features, logics and presentation tier working as expected. Changes in the schema of the database for the new app is assumed.

Summary of Tech Stack considered for this solution:

1. Application Load Balancer (weighted target group)
2. AWS Elasticache for Redis
3. AWS Database Migration Service
4. AWS Schema Conversion Tool
5. AWS Multi-AZ deployment
6. AWS CloudFront (Not shown in diagram)

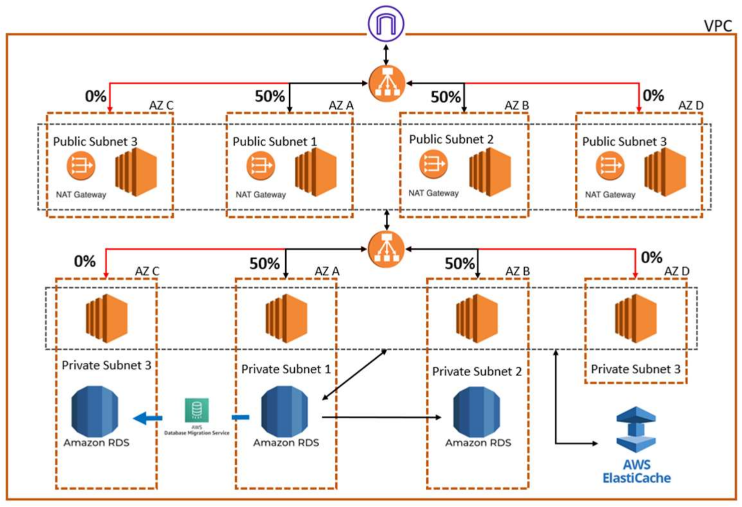


FIGURE 3: UPDATED ARCHITECTURE.

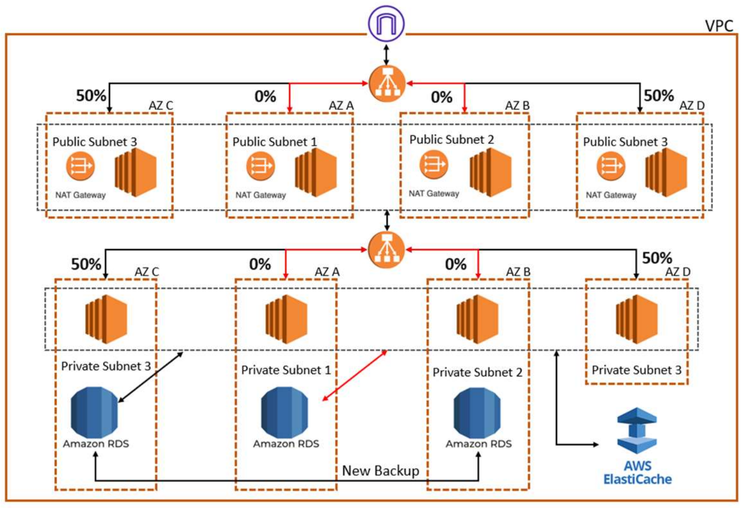


FIGURE 4: FINAL ARCHITECTURE

To do existing data migration to a new application without downtime, we can take the steps below:

1. Create Public Subnet 3 & 4 for the Presentation Tier of the new application.
2. Create Private Subnet 3 & 4 for the Logic Tier of the new application.
3. Set the load balancer with weighted target group with the initial weights for the new apps is 0%. (Figure 3)
4. We can start using the Amazon Database Migration Service which has Schema Conversion Tool (SCT). And given that we have already tested this in a different environment, there shouldn't be any problem rising.
   *During this time, the caching service (AWS CloudFront, AWS Elasticache) that we have already associated will reduce the downtime of the website while AWS DMS service is running.
5. Gradually increase the load balancer weights on the new apps to 100% to route the users to the latest application. (Figure 4)

Proposal for Question 3

For this scenario, we can go for Blue/Green Deployment method(Figure 5). The blue environment represents the current application version serving production traffic where all the users throughout Southeast Asia and green environment is staged running the latest version of the application.

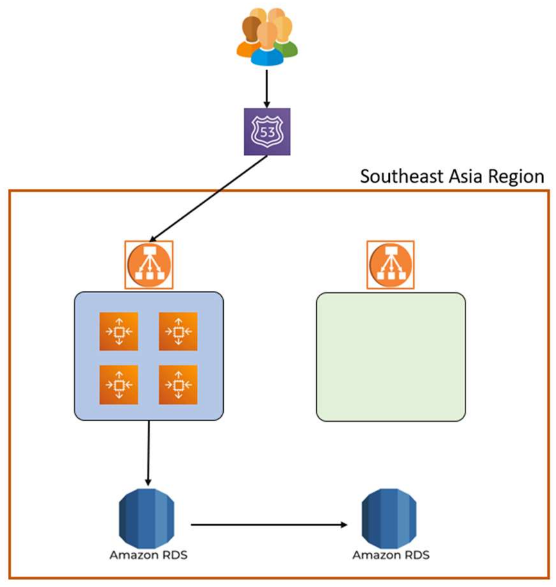


FIGURE 5: BLUE/GREEN DEPLOYMENT METHOD

The benefits of Blue/Green Deployment are:

1) We will be able to validate the environment with a small fraction of users.
2) We will have the ability to simply roll traffic back to the operational environment.
3) Works well with CI/CD workflows as the green environment is entirely new environment so there are less dependencies.

Summary of Tech Stack considered for this solution:

1) Application Load Balancer (weighted target group)
2) Amazon Route 53
3) AWS Auto Scaling
4) AWS Multi-AZ deployment

When it is time to deploy the latest application, we can use Route 53 – Geolocation Routing to route Malaysian traffic to the green environment, while the remaining is routed to the blue environment. As for the AWS Auto Scaling, we can scale up the green Auto Scaling group while taking down the blue Auto Scaling group instances out of service by putting them in Standby state. (Figure 6)

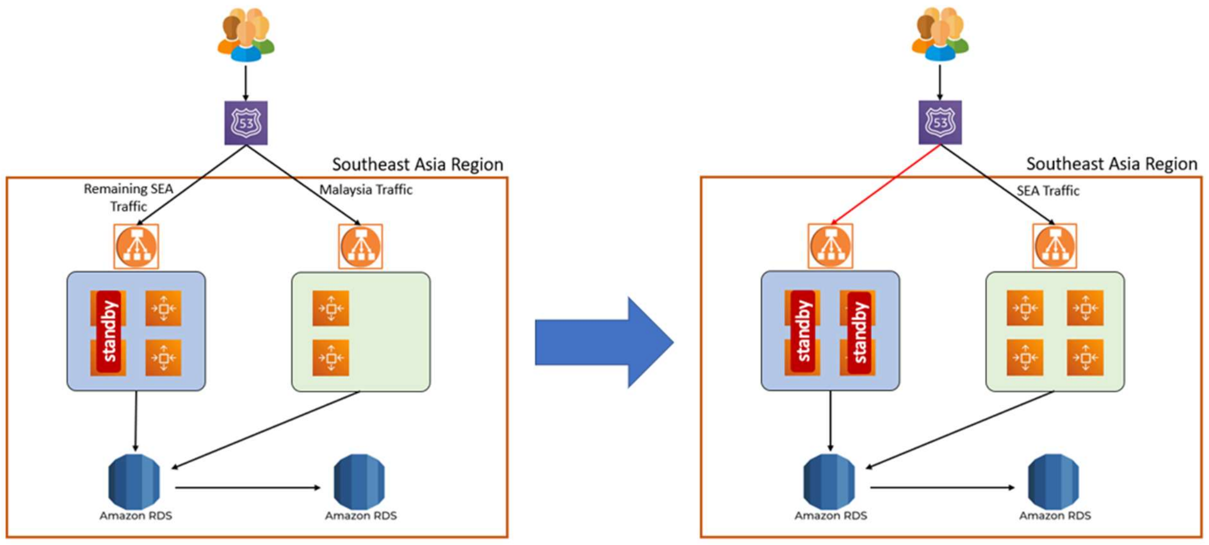


Figure 6: Route 53 and Autoscaling Transition

If there are no issues found in the green environment, we can slowly increase the autoscaling in the green environment while keeping the blue environment in standby mode.

The importance of having the resources in standby mode is we can roll back to the previous environment just by activating the resources that is on standby and reattaching the blue environment load balancer to Route53.