

Exercise 2.4: Observability-Driven Design Requirements

Practice defining the observability data needed for a feature using ODD principles, listed in the previous section by considering the following feature request:

Allow teams to quickly expose a deployed service to the internet by creating an API function that will accept a service name and location and:

- Create a DNS entry for the service at *service-name.petech.com*
- Only allow TLS-encrypted traffic on port 443
- Route traffic to the service location

Think about these two questions and define the observability data we might need to be able to show.

- Is this API function working correctly?
- Is this API returning the expected value to the team or organization?
- Do we need any alerts for this feature?

Solution

Is this API function working correctly?

Some example observability data that can be used to show correctness could include:

- DNS record listing for the DNS zone.
 - This can be tested to show the correct record was added.
- Routed traffic by port
 - This should show only traffic received on port 443 was routed to a backend service.
- Blocked or redirected traffic by port
 - This should show that traffic on any port was blocked. You may want to redirect traffic on port 80 to 443
- Route table listings
 - This should show the route specified was added to the table

Is this API returning the expected value to the team or organization?

Some example observability data that can be used to show value could include:

- Time from initial check-in to availability on the internet
 - Time taken should be low enough to show a significant decrease in the time taken to make a service available
- Developer satisfaction of feature use
 - Qualitative data from surveys and interviews should indicate the feature is easy to use
- Support requests submitted for DNS entry creation
 - Ideally, this should be 0 to show that the feature is entirely self-service

Do we need any alerts for this feature?

In this case, we likely do not. There would be no robust way for the platform team to detect a service misconfiguration from an engineering team, and traffic being allowed through only port 443 should be verified through functional testing.