

# Turbonomic Blog

## Thinking Like an Architect: Understanding Failure Domains

Posted by [Eric Wright](#) on Apr 26, 2015 3:30:44 PM



Find me on: [in](#) [t](#)



Tweet



Share



Like 0



Share

There are a number of considerations when building any type of infrastructure. Whether you're building a software application or the underlying infrastructure, there is an important part of our

## Subscribe Here!

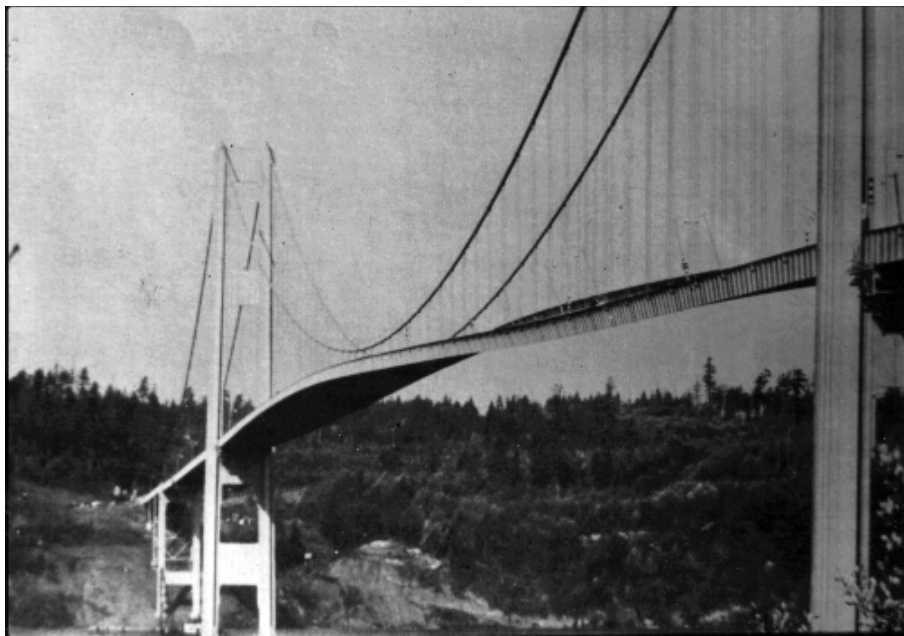
Email\*

Notification Frequency\*

- ☒ Instant
- ☐ Daily
- ☐ Weekly
- ☐ Monthly

This website stores cookies on your computer. These cookies are used to collect information about how you interact with our website and allow us to remember you. We use this information in order to improve and customize your browsing experience and for analytics and metrics about our visitors both on this website and other media. To find out more about the cookies we use, see our [Privacy Policy](#).

Accept



Here is a simple example to look at. If you're running a web application with a single Apache server and a MySQL database on two servers, you have a few failure domains to account for on the infrastructure.

- **Web server** - running a single instance of your web server is a rather obvious single point of failure
  - **Database server** - a single instance risks loss when the application is potentially unable to attach to the database
  - **Network** - while we were smart to separate the role of web and database server, this also introduces the network as a new point of failure
- These are fairly simple to see when we look at what our application environment is comprised of. So, what should we do?

### Don't Hesitate, Mitigate

Mitigation is the reduction of risk by some form of action or design. Let's break down some simple mitigation strategies to help our example application.

## Recent Posts

- [Mastering Cloud Cost Optimization: Cloud Cost Models & Discounts Overview](#)
- [Creating Opportunity: Turbonomic Announces the ParityPledge](#)
- [Mastering Cloud Cost Optimization: Frameworks for Success](#)
- [Takeaways from Turbonomic Live](#)
- [IaC Powering a Monolith to Multi-VM Architecture – Part 2](#)


Powered By G2

## Turbonomic

Review by Richard R.

★★★★★

2019-02-20



Turbonomic is my virtual co-worker

The lack of late night outage calls now that all of our systems resources are better allocated.

← • • • • • →

[Read more turbonomic reviews](#)

## Posts by Tag

## Web Server

We should be adding additional web servers to be able to handle the requests which will provide redundancy and resiliency. This means adding a load balancer into the application infrastructure to accept inbound connections and distribute the requests across the new web server farm.

## Database Server

Just like we did with our web server, we should be creating a horizontally scalable database architecture to allow for failures of certain nodes. This ensures data availability in the event of a localized outage. Luckily MySQL can be deployed in this way using MariaDB, which is a distributed relational database to allow for multi-node installations.

## Network

Since the network is a key component, it is also a key risk. We can add multiple network cards to the server, and attach the uplink ports to multiple switches so that we can withstand both a top of rack switch outage, and a single port outage, or even something as seemingly simple as a cable failure.

At the networking layer, we can have our network engineer ensure that the necessary failsafe designs are in place to prevent routing issues, switch issues, and multiple uplinks to the external network provider for better resiliency for network connectivity.

Sounds like we have a few good solutions in hand. This is where have to pause and think about the impact of our proposed solutions.

## Mitigation Introducing Risk and Complexity

We added a mitigation strategy for some of our components, but this doesn't mean that the problem is solved. Have you ever heard this joke?:

- [Cloud \(75\)](#)
- [Containers \(51\)](#)
- [Virtualization \(47\)](#)
- [Industry Perspectives \(40\)](#)
- [Features \(33\)](#)

[See all](#)

*I had a problem that I decided to use Regex statements to fix. Now I have two problems.*

Adding a few extra web servers looked easy when we put it on the idea list. One thing about web farms is that they assume you have a queuing system into the database when you are doing write functions. So, although we fixed the issue of a single point of failure, we introduced complexity that may not be accounted for in the application design.

This is a key reason that we focus on some DevOps concepts and the importance of having the infrastructure and application teams fully engaged when making architecture decisions.

### **Widening the Domain**

If we look at our mitigation strategy, we have added new servers, load balancers, and let's assume that we have also gone the extra distance to add a message queuing infrastructure to endure data integrity.

It would seem like we are done, right? Not quite.

If we widen the failure domain a little bit to something like a regional power outage, or network outage, we suddenly have a new set of problems.

We can easily get into what many call Analysis Paralysis. This is where we spend so much time looking for the ultimate solution, that we continually find reasons not to proceed. Hopefully we also love agile and lean processes so we have the ability to proceed in an iterative fashion and continually revisit to attend to deficiencies and a feature backlog that can include failure domain mitigation.

When you take a look at your application or server designs, you may also see that extending outside of a geographical region for redundancy is a potential solution. Perhaps bursting to the

cloud, or to multiple clouds!

The point of our example was to highlight that we should be acutely aware of failure domains and scenarios as we architect our solution. Nobody wants to get caught out when the outage occurs and has to say 'I didn't think of that'.

---

---

First Name\*

Last Name


Email\*

Website

Comment\*

protected by reCAPTCHA

[Privacy](#) - [Terms](#)



SUBMIT COMMENT

Next Steps

- Download Turbonomic
- Join Green Circle Community
- Request a demo
- Request pricing
- Take a class

Resources

- Product Documentation
- Resource Center
- Events
- Support
- Blog

Company

- About
- Leadership
- Newsroom
- Careers
- Awards
- Partners



500 Boylston St  
7th Fl  
Boston, MA 02116  
Tel: (844) 438-8872