MISSION CRITICAL SOFTWARE

deep3®

Experiences in serverless

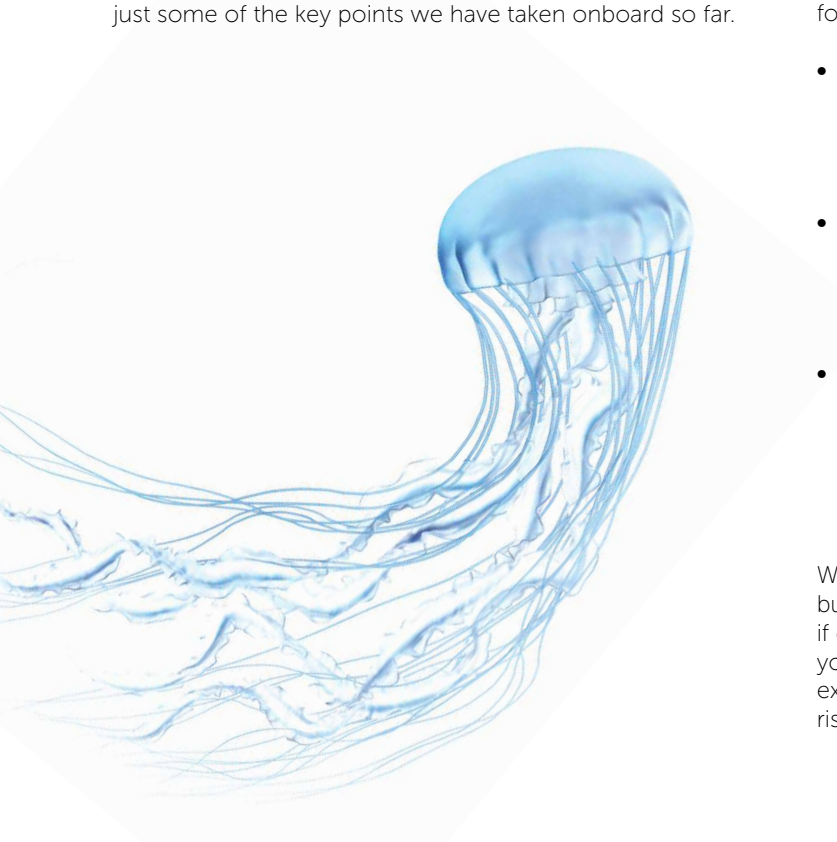
A whitepaper from Deep3®

Discovering the depths of serverless

At Deep3® we've been working with serverless technology for a number of years now, designing and building scalable, low cost and low maintenance solutions for our customers.

With our customer base in the Cyber and National Security sectors, a secure by design approach is always at the forefront of our minds when designing and developing systems and this is no different when working with serverless architectures.

As we continue to increase our adoption of serverless technology we have learnt a number of important lessons along the way. Through experience we have come to understand the benefits and potential pitfalls of what is a very different way of approaching application design and implementation. As with any organisation embarking on the journey to serverless adoption, there is plenty to learn but here are just some of the key points we have taken onboard so far.



The case for serverless

Why you should use it...

The first thing we had to understand, even before we had written a single Lambda function, was why adopt serverless in the first place? To ensure it was the right technology choice for our customers, we had to first understand the benefits. It quickly became clear that there are a number of benefits, all of which, if implemented correctly as part of a well architected system, can deliver excellent value for money over more traditional architectures. These benefits include:

- Lower Maintenance
- Greater scalability
- Lower running costs
- Minimising the application's attack surface

....And why might you not

With these benefits it would be easy to assume that serverless is the obvious way forward for any solution. However, there are reasons why it may not be right for some systems and teams. In our experience:

- **Development can be more complex** – For example, it is difficult to replicate the AWS environment locally and multiple developers deploying test systems rapidly can come with its own problems. Debugging can also present an additional challenge when using remote services.
- **Issues with underlying cloud services can have greater impact** – With more reliance on the cloud provider, the impact of an unexpected bug in one of the cloud services, or discovering the lack of a key but essential feature, is greater.
- **Troubleshooting and monitoring can be difficult** – Traditional monitoring and logging solutions don't always work as well. Different approaches have to be developed to provide alerts around system performance and incident response. This needs to be part of development and not an afterthought.

We don't believe these are insurmountable challenges but it is important to understand, and mitigate them, if choosing to adopt a serverless architecture as your solution. Having the right people with the right experience and certifications is the best way to minimise risk when designing your serverless architecture.

What else have we learnt?

All in approach to serverless

One lesson we have learnt is that for new, greenfield development in particular, going all in on a fully serverless architecture which utilises the wider services offered by the cloud provider can enable all the benefits we've outlined to be realised.

Hybrid architectures, with a mix of serverless and more traditional virtual machines can be a great step forward, but we have found that the most scalable, reliable and low cost approach is to embrace the full breadth of services around which your serverless architecture can be built. Using AWS as an example, we are talking about:

- **Lambda:** when people talk about serverless they are often referring to Lambda functions, the backbone of a serverless application is the ability to perform task-based operations on demand with no responsibility for the provisioning or underpinning architecture.
- **Simple Queuing Service (SQS):** to deal with significant load you need a queueing system to manage consumers (Lambda functions), without this you could lose data when the system reaches maximum capacity.
- **CloudFormation:** infrastructure of code quickly becomes unmanageable without a reliable and repeatable mechanism to deploy.
- **RDS (Relational Database Service):** even if your computer is on demand its common to require a database service to store and retrieve data. RDS provides many traditional options like MySQL and PostgreSQL and more recently a more efficient MySQL drop-in replacement with serverless functionality called Aurora.
- **S3 (Simple Storage Service):** with no permanent file storage possible on Lambda functions applications need a central data store. S3 is capable of huge volumes and it is fast, reliable and cost effective.

Re-think how you design and build applications to use serverless

Another lesson we have learnt is that you really shouldn't simply use serverless as a straight replacement for more traditional servers or Virtual Machines (VMs). It is important to re-think how you design and implement your application to make best use of the features of serverless and ensure you don't fall foul of using serverless functions in the same way you might use a traditional server. This is especially critical if you are rewriting an existing application.

It is not that hard to break an application if Lambda functions are used incorrectly. For example, if using something like Java with dependency injection don't expect an environment to be 'clean' when the function is invoked. The jar file never stops, it is just the handler method getting called multiple times. Your objects might become stale and potentially find themselves in an unexpected state.

MISSION CRITICAL SOFTWARE

In summary

The benefits of serverless technology are obvious and can have a huge impact in terms of cost savings, greater reliability, and (if well architected) reducing your security risks. However, serverless is not a silver bullet and, if adopted, must be done with an understanding of the potential pitfalls and also the knowledge of what does and doesn't work.

With our experience of serverless and wider cloud services Deep3® are well placed to help organisations make the right decisions when it comes to adopting this technology and ensuring the systems they design are well architected, secure and resilient.

Get in touch

We love talking with customers about their challenges and discussing how Deep3® can help them through the adoption of cloud technology, including serverless. If you would like to talk about how we can help then please get in touch:

Ryan Thornley
Head of Cloud Technology
ryan.thornley@deep3.co.uk
07709 818 381

Richard Yorke
CEO
richard.yorke@deep3.co.uk
07791 984 050

deep3®