

CONTAINER TECHNOLOGY - A BUSINESS VIEW

When you install an app to your phone, it starts and runs first time - no problem. Yet the deployment of software onto cloud servers remains challenging and error prone. The result is heightened risk, slower release cycles, and a loss of agility for the organisation as a whole.

CONTAINERS ARE APPS FOR SERVERS - they are pushed to a server, where they start and run first time. Which begs the question: How was this ever not the case?

The problem is twofold: first is that modern software depends on large numbers of other pieces of software, and that for our software to behave predictably each of these dependencies has to be versioned and obtainable; second is that the process of installing software can be non-trivial and hence error prone. For both these problems the current solution is pass the knowledge from developers to production through documentation - which is inevitably out of date, or may not exist. The devops movement helps, but at the cost of developing *more* software, at great expense, that *also* has dependencies.

Containers are images. By having the software be installed only once, by the original developer, with the result being wrapped into an “image” that can be referred to with a single serial number. This image runs identically, everywhere - which means that test will see the same results as the developer, and production will see the same results as test. Paperwork is eliminated, and the devops load is low or non-existent.

CONTAINER INFRASTRUCTURE IS EFFICIENT. Virtual machines brought us zero capital outlay, and near limitless power on tap. It changed our industry forever, but only took us half way. Virtual machines divide compute power into fixed size segments - we still need to decide in advance how much power an application is going to use, we still end up paying for unused capacity, and changing to the ‘next size up’ is far from simple.

Containers share resources. A single, large machine - be it physical or virtual - divides its resources between containers dynamically, and on a per-millisecond basis. We still pay for unused capacity, but there can be far less of it and the capacity can become used again near-instantaneously and with no additional administration load. There’s no free lunch and heavy workloads such as machine learning or visual effects will see no change - but traditional bursty business/cloud loads can expect to see a 10x gain in density.

CONTAINER INFRASTRUCTURE IS FLEXIBLE. Container environments start in single digit seconds rather than the double digit minutes associated with devops environments, and multiple hours when provisioned manually. We no longer need ‘a test environment’ since one can be bought into existence on demand and near instantaneously.

Containers can create new architectures. With container-per-user isolation or maybe even per-transaction, an entire class of security flaws can be eliminated and application code simplified. It’s also possible to do container-per-node scaling, leaving the containers dormant until needed at near zero cost and eliminating the need for scaling groups.

20FT IS THE BEST WAY TO START. Simplicity has been the focal point of 20ft’s development right from the start. The learning curve is short and competence ‘by the end of the day’ is a realistic goal for most teams. Looking further out, an imperative SDK takes the guesswork away from traditional declarative SDK’s, developers merely tell the infrastructure what to do and can expect that it will be done.

20ft integrates with AWS and uses your existing investments in infrastructure, skills and processes. Containers are standards based, and will run on other environments including Kubernetes and Docker.