



CODESHIP

by CloudBees



TAYLOR JONES
SOFTWARE CHEF AT IZEA

How to Become Cloud Native



About the Author.

Taylor is Software Chef at IZEA, slowly simmering in the Florida heat. He is a contributor to the Codeship blog and mainly writes about how to transition to a microservice structure with your software team.

Codeship is a fully customizable hosted Continuous Integration and Delivery platform that helps you build, test, and deploy web applications fast and with confidence.

Learn more about Codeship [here](#).



How to Become Cloud Native

In a world of numerous programming trends, companies are starting to boast about how they're becoming "cloud native." This declaration might seem full of hot air and pretension, however, there's a huge value to embracing what cloud native truly is.

In short, cloud native is an approach to creating and deploying applications that fully taps into the advantages of delivering, scaling, and distributing your products on the cloud.

The concept isn't concerned about *where* you deploy your applications — it's more concerned with *how* you create and deploy them!

In this eBook we're going to break down the three pillars of being cloud native and what they mean on a practical level.



I believe there are three pillars that are crucial to embracing the concept of being cloud native:

1. The utilization of microservices in your architecture
2. The utilization of continuous integration
3. Utilizing containers in your applications

Adopting and mastering these three aspects will give companies the best advantages of being cloud native. To further explore this, we're going to break down each one of the pillars and what they mean on a practical level.



Microservices

[Microservices](#) have taken off as a general programming trend in the past few years. The idea is based on taking apart one giant monolithic application and decomposing it into smaller ones. It's much like [object-oriented design](#), but for architecture!

Having a microservice-oriented architecture will enable you to take advantage of the scalability that being cloud native offers. With a traditional monolithic architecture, you can only scale a few aspects of your application. These aspects most likely would be an application server and a database service. While moving these to the cloud would give you advantages, you're limited in the amount of things you can scale and fine tune.



Having microservices, however, allows us to better tap into what hosting applications in the cloud offers us. Where monolithic applications don't offer a lot of scaling options, a well-constructed microservice application suite allows us to fine tune more aspects of our application that we want to scale. This gives us a tighter grip on our performance but also saves us a lot of money in server hosting costs!

Yet, with all of our microservices deployed and fine-tuned, how can we streamline their environment from development and deployment? This is where containerization comes in!



Containers

Containerization tries to solve the problem of environment discrepancy and orchestration. It solves these problems on both a local and deployment level. Locally, your developers have a consistent environment that they can spin up and work on. They create a stable environment for developers to create and deploy code through. Furthermore, tools like [Docker](#) and [Kubernetes](#) have made it simple and efficient to orchestrate these containerized applications!

The advantage of using containerization on the cloud is that it provides consistent environment from code in development to code in production. No longer are we



debugging the difference between the production server and your local computer. With containerization, you have the assurance that the environment that code is created and tested in is the same as the environment in your production server.

Having this consistency is crucial for being cloud native. Containerization allows us to create, deploy, and scale our applications with ease because of their consistent environments.

I mentioned earlier about the journey of code from development to production. Yet how exactly does that journey play out when you're cloud native? Enter [continuous integration](#).



Continuous Integration

Without [continuous integration](#), you miss out on one of the biggest advantages of becoming cloud native: having a seamless flow from development to deployment. In



— CHESLEY BROWN, INVISION APP —

From a vision perspective, Codeship's Continuous Integration service has hit the nail on the head.

LEARN MORE



order for large organizations to really leverage the benefits of the cloud, there's a need to have a form of continuous integration in your process.

You can use tools like Codeship to help your team follow a pattern like this:

1. Your team creates a batch of code that you want to deploy.
2. You push that release up to Codeship.
3. Codeship tests and runs audits on the new code.
4. If the tests and audits are successful, Codeship deploys your new release to a production environment of your choice.

This process effectively eliminates the need to manually package up code and deploy it to a server. Once you've dictated the proper standards that you want your code to pass, any code that meets those standards will immediately be sent to your live code base.

This is incredibly relevant for being cloud native because it embraces the advantage of having small, quick deployments to the cloud. Since that infrastructure of server hosting and deployment is handled by the cloud, all we have to worry about is creating awesome features and fixes! With the right CI configuration, your cloud native assets will be able to grow at scale with ease!



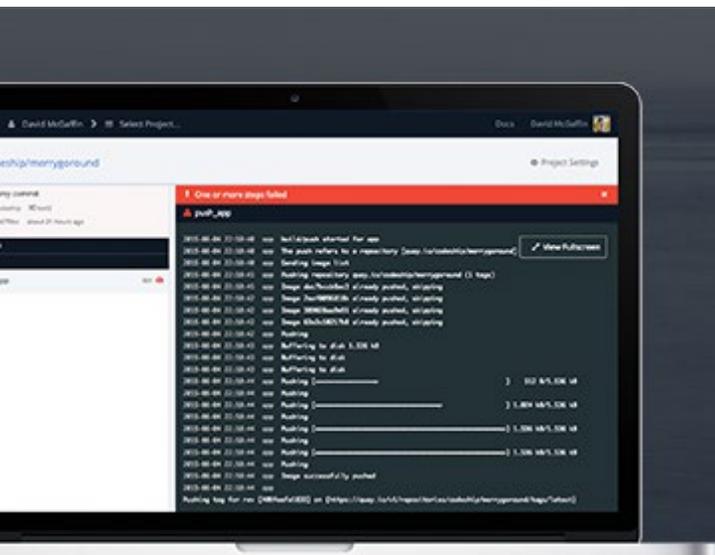
Wrapping Up

To review, we covered three crucial aspects of being cloud native:

1. The usage of microservices in your architecture
2. The usage of continuous integration
3. Utilizing containerization through tools like Docker

The combination and orchestration of these three pillars will help your teams not only leverage the best parts of being cloud native, but also create quick releases, safer code, and more streamlined development. Moving your assets to the cloud is a significant investment. Why not make the best of it that you can?

If you are interested in learning more about Cloud Native, check out [Joe Beda's Cloud Native series](#) on our blog.



START WITH THE \$0 PLAN

Sign up for Codeship's free plan.

Get 100 builds per month and
unlimited private projects for free.

CLICK HERE TO GET STARTED



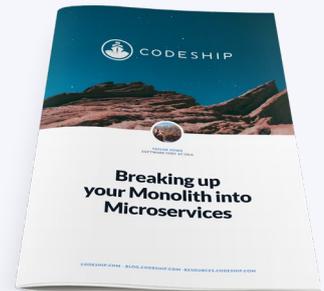
More Codeship Resources.

EBOOKS

Breaking up your Monolith into Microservices.

In this eBook you will learn about the basics of "decomposing" a monolith into microservices.

[Download this eBook](#)



EBOOKS

Why Containerization is the Future.

In this eBook we will take a look at the differences between the Traditional Stack, the Virtual Machine Stack, and the Container Stack.

[Download this eBook](#)



EBOOKS

Why Continuous Integration Is Important.

In this eBook you will learn what Continuous Integration is and why it is important.

[Download this eBook](#)





About Codeship.

Codeship is a hosted Continuous Integration service that fits all your needs.

[Codeship Basic](#) provides pre-installed dependencies and a simple setup UI that let you incorporate CI and CD in only minutes. [Codeship Pro](#) has native Docker support and gives you full control of your CI and CD setup while providing the convenience of a hosted solution.

Codeship Basic

A simple out-of-the-box Continuous Integration service that just works.

Starting at \$0/month.



Works out of the box



Preinstalled CI dependencies



Optimized hosted infrastructure



Quick & simple setup

[LEARN MORE](#)

Codeship Pro

A fully customizable hosted Continuous Integration service.

Starting at \$0/month.



Customizability & Full Autonomy



Local CLI tool



Dedicated single-tenant instances



Deploy anywhere

[LEARN MORE](#)