Article
Bill McCormick · Dec 4, 2015 3m read

Atelier Beta Cloud Infrastructure

A few people wrote to me asking about the infrastructure behind the Atelier Server implementation. Its neat and a worthwhile story to share so I am writing it up here as a post on the community. I want to go in to a little detail on why it was needed and then I will outline in detail how we went about implementing this.

So why did we need to give people a “ server in the cloud ” ? Atelier is the new IDE for InterSystems products. It is Eclipse based and implemented in Java so it is cross platform. The communication between the Atelier client and the server is REST based over http. The REST api for Atelier is new and was built specifically to support not just Atelier but anyone who wished to roll up an IDE against Cache. It is also brand new and does not exist on any released product or even the current field test version, 2016.1. So to distribute the field test every user needed to install a new version of the server as well as the new client. Without a new server you can ’ t use Atelier at all. As we go forward we will ship versions that have the api and this will remove this limitation.

A second consideration was to drive home one major architectural difference between Atelier and Studio. Studio was an editor that presented what was available in a given server instance. Atelier is an independent IDE that can be used even when there is no instance of Cache available. The code you edit lives outside the database. The server is no longer the repository of the code base. Its simply the compiler / debugging environment for testing out the code you are developing.

A last consideration was the rise of containers as a critical new element in the matrix of ways people can leverage the cloud in new and different ways. For those not familiar with containers they operate differently then virtual machines in how they interact with the host OS. They are a critical aspect of the elastic cloud. You need a resource, you grab it, when you are done you discard it. If you need it again you can always just grab it again. Its the model driving DevOps and has made Docker a critical new technology. However Docker, the industry leader, today works primarily on Linux. So while this was a great solution for the server it was not suited for the client because we don ’ t want to force everyone to learn Linux to play with Atelier.

However this lead to what is really a perfect motif for what we needed with Atelier and this beta. You have the client on your own instance under your control. All your code is there. But when you need to compile something, or debug, or run a unit test well there you have it. And when you are done you can discard it. Your code is safe. Its in the client repository, and hopefully integrated to source control. In addition because we control the container we can update it any time to pick up new features. You as the beta tester don ’ t have to download a new version every couple days. When we have something worth updating for we just publish a new container.

So how did we do it? We are using Docker to build the containers. We take an install of the server code and run it into a CentOS image. We set up users, load the key etc using standard docker scripts. We give this Docker file to a company based here in Boston called Appsembler.. They also host our E-Learning educational offerings using a similar stack. They put the container up on Digital Ocean who deploy it in to AWS EC2 environments. Appsembler provides me with a JS widget that renders the Launch Server button on the field test page. Implemented in JQuery, it calls Appsembler via REST and provides them with a unique identifier for that user. Over on their side of things they then launch a container based on a set of business rules we agreed to. How long will it be up for? Can it be relaunched? The original container state or do changes need to be preserved? etc. I don ’ t have to worry about any of these things, Appsembler sorts all of that out for me.

In conclusion I just want to say I think containers are great technology. At the moment this market is owned by Docker but Microsoft is adding support in Windows for Docker and is working on their own container solution. We are preparing an official statement on container support as well. If you have any additional questions feel free to post them here and I will reply as soon as possible. Happy coding!

#Caché #Docker #Development Environment

Source URL:https://community.intersystems.com/post/atelier-beta-cloud-infrastructure