

This article was downloaded by: 10.2.97.136

On: 26 Mar 2023

Access details: *subscription number*

Publisher: *Routledge*

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: 5 Howick Place, London SW1P 1WG, UK



The Routledge Companion to Managing Digital Outsourcing

Erik Beulen, Pieter M. Ribbers

Welcome Mission Critical Cloud

Publication details

<https://test.routledgehandbooks.com/doi/10.4324/9781351037785-17>

Arjan Eriks

Published online on: 28 Jul 2020

How to cite :- Arjan Eriks. 28 Jul 2020, *Welcome Mission Critical Cloud from: The Routledge Companion to Managing Digital Outsourcing* Routledge

Accessed on: 26 Mar 2023

<https://test.routledgehandbooks.com/doi/10.4324/9781351037785-17>

PLEASE SCROLL DOWN FOR DOCUMENT

Full terms and conditions of use: <https://test.routledgehandbooks.com/legal-notices/terms>

This Document PDF may be used for research, teaching and private study purposes. Any substantial or systematic reproductions, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The publisher shall not be liable for an loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

14

WELCOME MISSION CRITICAL CLOUD

Arjan Eriks

14.1 Introduction

January 2016, it's time for a change. After years of satisfaction with the CloudStack community, we have today reached the point that our vision of the future is essentially different from some of our partners. We as a company need to increase speed, improve release cycles and encourage change in the Cloud orchestration software. After almost a year of deliberation, Schuberg Philis has reached the decision to fork CloudStack and start in a new direction: Mission Critical Cloud all the way.

Since the start of the company in 2003, Schuberg Philis has grown by managing dedicated infrastructures and application landscapes for its customers. In 2008, when Cloud technology started to take-off we started to develop the desire to enter the Cloud and Infrastructure as a Service (IaaS) space as part of our Mission Critical services. Our enthusiasm in CloudStack and the Apache Software Foundation began in 2011, when we started working with CloudStack. Citrix had acquired cloud.com, followed by the very exciting decision to donate the software as open source to Apache. Although being open source was not unique to CloudStack, we did like the intimacy of the community and its vision on Cloud orchestration developments for Mission Critical and enterprise level applications. As an experienced integrator we dislike vendor lock in, and this bold move by Citrix gave us the extra option we needed: an alternative to OpenStack that would allow us to enter this new time in an agnostic and open-source way.

In late 2011, a team of four engineers from Schuberg Philis began building the first stage of what would become our dedicated Mission Critical Cloud. At first, as with all new technology, we started with a proof of concept, and we built both an OpenStack and CloudStack orchestration layer to test and explore the features each could provide. As with many before us, and as many after discovered, CloudStack provided a reliability and simplicity that OpenStack simply could not match, and while OpenStack offered more features, it could not deliver the quality or reliability we desired in our IaaS software. Our goal was to build a Cloud environment for both our engineers and our customers, one controlled with a standard API that would use new and exciting technology and would help steer us towards the future, while empowering us to deliver our 100% goal in everything we do. To be really certain we made a technology shift, we promised ourselves to not use any tool or brand we had

ever used before. This promise to ourselves assured that we could explore the real potential of Cloud technology. It avoided the use of historically developed design principles that may very well become obsolete in a Cloud-based world.

After the decision had been made, we rapidly began developing. From easy implementations of Nicira networking (now known as VMWare NSX) to hours of pain coaxing storage to work, we learned how to bring a Cloud-based architecture into Mission Critical environments. Some key points that we proved included:

- 1 Enterprise application workloads can run in a Cloud.
- 2 Being agnostic and using OSS means, we can drive the feature roadmap because we develop it ourselves.
- 3 Running open source at scale requires dedicated software engineers devoting their time to improving Cloudstack.
- 4 Our customers operate in sectors varying from online retail, financial services, utilities and public services. All our customers require rapid and frequent change to their applications and systems. Many of our customers operate in highly regulated markets resulting in strong (data) compliancy and security requirements. Our customers trusted us to take care of both their functional and nonfunctional requirements, ensuring that moving into the Cloud was not a blocker for them.

Our first development job we took on was to integrate Nicira into CloudStack. One of our engineers, Hugo Trippaers, boarded a plane to San Francisco to visit Citrix and Nicira, and following months of coding we delivered Software Defined Networking into CloudStack. This was the second major step of our journey to IaaS, not waiting for vendors and their roadmaps, but defining a feature, building it and delivering it ourselves.

Over time, our influence grew. Citrix bought Cloud.com which was rebranded to Cloudstack and open sourced by Citrix. We started by working closely with two senior technical executives (Shannon: VP Market Development & Cloud Platforms and Sameer: Group VP and GM Cloud Platforms Group) from Citrix on the core positioning strategy of Cloudstack in the market and within Citrix. Citrix offered a commercial version of Cloudstack (Cloudplatform) including enterprise support in bugs and features. As we valued our partnership with Citrix and we saw benefits in the enterprise support levels, we decided to buy CloudPlatform.

We actively participated in the open-source community. We were invited to talk at the CloudStack events in Las Vegas in December of 2012. A group of Schuberg Philis engineers attended the conference, including Hugo Trippaers, Harm Boertien, our Chief Community Officer and Jeroen de Korte, one of our Mission Critical Engineers. The stars of the party though were Funs Kessen and Roeland Kuipers, who delivered an entertaining talk on stage. A later performance in Budapest earned them the nicknames of Cloud Pinky and Perky. We used this opportunity to meet the original founders of cloud.com and users and developers from the greater community, which was a fantastic feeling for us.

This was not only a turning point in our usage and vision of Cloud technology, but also the start of our involvement in multiple open-source communities. We started becoming more connected with the DevOps community, the Lisa conference in North America, and we delivered a speech at the CloudExpo in Santa Clara too, a bridge not just to orchestration but also a discussion of enterprise level challenges, the security implications of a Cloud and how this can be integrated into CloudStack. But for the CloudStack community, we made the conscious decision to give back to the project.

We hired developers to document our own contributions and invested in conferences and events supporting the community. This increased our impact in the community and we were able to make a significant contribution. We started, in collaboration with Citrix, running “Build Your Cloud” days, designed to make CloudPlatform and CloudStack more visible in the market. The largest event we organized with our friends from the community was the epic CloudStack Collaboration Conference Amsterdam in November 2013, where almost 500 users and developers attended to discuss and plan for the future, although we still have to give the record of best beer and pizza event to ShapeBlue conference in 2014 near London West End.

The position of CloudStack at the Apache Software Foundation (ASF) itself developed rapidly as well. While it was first an incubator project, it very excitingly evolved into a full ASF project. Chip Childers became the first ASF CloudStack VP, later followed by our own Hugo Trippaers and after that being looked after by Sebastien Goasguen. We continued to commit more code, expanding our Cloud, our development team and progressed to being the release managers for multiple releases. It was in this final stage that we experienced, for the first time, friction between the community and ourselves on priorities on new functionalities and related non-functional requirements such as security and “deployability” of applications and systems.

In Spring 2015, several of the key collaborators in the project gathered together in London to discuss a roadmap for CloudStack, and agree on a new way of working. Following that, we all agreed to work more closely together going forwards. From a personal perspective, we wanted to increase test coverage, and at the CloudStack Collaboration Conference in Tokyo in June 2015 all of the key players in the community agreed that we needed to work on changing the release process to a weekly cycle instead of the nine months cycle used up to then.

We took on the role of release manager started from release 4.6, wrote documentation about our view of the release process and encouraged debate and discussion on the mailing list, followed by stabilizing the master branch of github. This was a big change from the previous method of creating a release branch and forming different QA teams to stabilize and release it, while development continued on master. This had two major benefits: first we could release a new version at any time, and now that releases were built on top of each other, we could guarantee that a fix would always be present in later releases of the software – if it was in 4.6, it would be in 4.7. This new way of working allowed a monthly release cycle, with increased code review and more automated integration tests. Over 500 changes were tested, reviewed and included into the source code between November 2015 and January 2016. The new way of working provided shorter release cycles, major improvements in overall stability, quality and backward compatibility of the releases. We broke the monolith down into smaller pieces and enhanced the maintainability of the code.

Acting as the release manager with so many volunteers, developers and company goals is a task that requires both tact and stamina. We had two key engineers who attempt the task: first Daan Hoogland (release 4.4) and then Remi Bergsma (4.6 till 4.8) attempted to not only organize and complete a release, but at the same time improve the speed, reliability and quality of the software. At the same time as we pushed for this change, we continued to run our Mission Critical customers on CloudStack. Our customers and colleagues depended on the bug fixes and quality assurance we were trying to deliver, and we had to develop while maintaining our service.

In the Fall of 2015, during this process, we were hit by a nasty bug. Without diving too deep, a code snippet in the HyperV module of CloudStack started reporting non-HyperV

hypervisors as down, resulting in the CloudStack management server trying to shut down and migrate the VMs running on them. Thanks to a high-availability design at a layer above the Cloud we had no downtime, but it was a reminder of unused and untested code could cause outages. In retrospect, it became clear that these risks needed to be eliminated. We rely on CloudStack for our Mission Critical workloads. We required the Cloud to behave and perform on the same level or better as dedicated environments. It turned out that our minimum requirements for stability, performance and reliability were not being met. It was a wakeup call for us: pushing for quality in an open-source fashion is key, and we needed to redouble our efforts to insist upon it to prevent a recurrence of critical bugs.

It became clear to us that things had to change. We devoted more of our time with the community to make sure that new commits were properly tested, and encouraged stability and testing in the master branch, and a faster pace of developing bug fixes and feature releases. For us, this was almost the optimal way of working and we wanted to push it even further. But it became apparent that this was not so for others. We received push back from the community, from both users and developers, and after spending considerable time, considering multiple options the way forward for us became clear. We decided to fork CloudStack.

This was not a decision we made lightly, and we are keenly aware that this move has a number of implications not only for us as a company but also for the existing CloudStack community:

- We faced the risk that the community would not follow us. We should be prepared that we needed to be fully self-supporting.
- The total size of the community would become split; this would potentially lead to a reduced number of contributions. Slowing down the pace and volume of new functional developments.
- Our existing customers were asked to trust that the fork would not negatively affect their operations or future plans.

Over time, both versions may start to deviate from each other. We have decided that our fork will remain open source to allow for as much collaboration as possible. We are aware that others in the past have chosen to close forks of CloudStack and develop internally, but we believe in the power of sharing. Being open, inclusive and transparent are values we see as a key driver in making the IT world a little better. An open fork, with the tools and procedures we feel needed to ensure higher velocity, greater reliability and better quality, is, in our opinion, the best option at this point in time.

Our future roadmap is mainly focused on our customer requirements. We are a Mission Critical company, and have a need to cater for high performance computing, support for containers and integrate with other new technologies such as Kubernetes, Mesos and Nomad. Our customers' requirements change rapidly, and we need to be agile in creating new components while also guaranteeing quality. Besides customer requirements, we will start working on improving the architecture of CloudStack. Items that spring to mind are the plug-in model, removing dead code and refactoring of important items.

So there you have it. We have decided to create a fork. It will be open for all to use and contribute to, with new governance, procedures and tools all designed to deliver quality and velocity to the project. We would like to stress our dedication to being open and inclusive, and would like to welcome anyone who has ideas about our move, or is interested in joining us on this journey, or perhaps if anyone would like additional information about our decision, please do not hesitate to contact us.

We truly hope you understand what we are doing, and that this is not an attempt to hurt the current community.

I would like to personally thank everyone in the community for the opportunity to participate and contribute to this project over the past years, without you all we would not be where we are today, and I'm incredibly grateful for that. We will blog more in the coming weeks about how we foresee our community functioning, and hope to publish more information about the project governance and bylaws very soon. For now, I welcome any comments and pull requests!

May 2018

Since 2016 the need for new functionalities in the Mission Critical Cloud has reduced gradually. The focus is on optimization. Alongside, public Cloud is developing at a very high pace. Big Cloud providers such as Amazon Web Services, Microsoft Azure and Google Cloud are heavily investing in new computing and networking technologies, (micro)services integration platforms. The risk appetite of customers to embrace public Cloud is changing and new opportunities utilizing the potential of Cloud technology are emerging. Our company is rapidly adapting to these changes to be able to support and lead our customers in this journey.