The Cloud Vs. Dedicated Servers

The Cloud vs. Dedicated Servers

I spent a lot my time these past two weeks working on the hosting option that drupan powered sites. I thought of all the possible ways to host them while I was breaking down tasks into services. I looked at AWS, Heroku, Rackspace and considered renting some dedicated servers from Hetzner. To be honest I am surprised by the outcome. I hope you find this helpful and that it gives you some insight into the process.

For those who don't know me, I enjoy servers. Setup, scaling and administration is nothing I consider to be a necessary evil, but actually enjoy it from time to time. When I started organizing network parties with my old team I spent more time with game servers, the network and working on the tournament system than gaming. I think I know pretty well (after all those years) how to install a database, add ssh keys to a system and make sure the web server is up to date. It wouldn't be a problem to do it the right way. There would come a time when I would need to seek out help from experts in database design/sharding/partitioning, which would likely be sooner than if I went the cloud hosting route. I recognize my strengths and weaknesses. Hiring is an option for me if I need it.

Before you look at the various options, you need a clear understanding of what you require. I need block storage. This means that hosting on Heroku is not possible. I want to use a CDN for the static generated sites. This has two advantages.

- Domains that are still available if the backend goes down, or a host provider has problems.
- Performance.

The load on systems at the moment is unpredictable. It is much more difficult to have 20 users create their site at once than it is to have 2 users update a site every morning and evening. It will be possible to rent dedicated frontends or backends with a custom branding. This is a great advantage.

Stability and Performance

You will find as many opinions about cloud providers as people writing. The cloud is both great and bad. Moving to Heroku and moving away 100% uptime, "crashes", if I run short of uptime. One CDN is faster than the other from different points on Earth.

There are some important things that you should always remember

- the cloud does not mean you pay nothing and get all the performance a datacenter can offer - you will have downtimes - it is easier to minimize downtime - it is faster to get processing power, storage or memory if you need it

Overall it is easier to react to problems, needs and give various options a try and see how it performs in testing or production if you choose a cloud provider. If you don't want to purchase systems "just because", or if the stars align right , you may be able to use them.

Amazon Web Services

AWS can be summarized in one sentence: "You get everything and they pay for everything they measure." It is that simple. Need a task-list? SQS. Payment processing? FPS. Block storage? EBS. And the list continues. As I mentioned, you pay for everything. You can measure IOPS, Traffic, and requests. You have to pay for it. I actually prefer this to a certain point. They make money if they charge me for the resources that I use. They are less likely to go out-of-business if they make money.

While visiting an AWS event in Leipzig I had the chance to talk to some people. Now that I considered using AWS I send a mail with my open questions. Two days later, everything was resolved. I described what I wanted to do. The - I think - sales engineer answering my questions also commented on the possibilities and shortcomings and how to compensate them. The pre-sales service was excellent.

Another thing I like is that you get an unlimited tier which can be used to test your ideas. Of course micro instances are not the most powerful option the world has ever seen, 100000 SQS requests are just enough for two request every minute. Depending on your use case this can be enough. Or, you can pay $2.70/month for a service that queries every second of the queue. Overall, you get a decent package for free.

You will need to enter your credit card information before you can view or use anything. They won't take your case down and will charge you for exceeding the free limit. It is impossible to get statistics about your next bill. For me, this is not really pleasant.

- Everything you need can be found in one provider. - Useable administration interface – Many systems (High CPU, High Memory ,...)).

- You need to learn Amazons vocabulary. There are always three letters. Each letter can stand for either "simple", or "elastic". - Calculating monthly costs is nearly impossible without usage statistics. - Not very cheap - I recommend that you contact sales before purchasing - there are certain limitations I could not find on the page/FAQ. (maximum 100 buckets with S3

e.x. - Look out for vendor lockin - IMHO bad documentation

Heroku

I felt a comfortable feeling when I used Heroku. They are doing everything right. You can use either the web interface or a command-line tool. The magic happens when you push your repository. You are not sure how something works. Clear and structured help and FAQs. You don't see all the things you would need to do if your server is private.

I cannot say how their pre sales support looks like because I did not need to contact them. This is a great sign.

The pricing is affordable and you can get almost everything for free to try stuff. Want a task queue? There are many providers that can offer you a Redis instance or RabbitMQ hosted. Although I do not recommend using the free options for anything other than testing, it is fine for me. I can see if my applications still work when deployed, what else can you want?

- Usability - Open Source based services with a free tier - Reasonable pricing

- No storage options. (If they are not necessary, you can opt for S3e.x.

Rackspace

Rackspace offers servers, storage, a CDN, and other stuff just with a lot better pricing model than AWS. There is no equivalent for FPS e.x services. But this is a very special case, if you ask my friends. You don't need to pay for everything they can measure. It is easy to understand and follow the pricing structure.

They boast of their excellent support. My first experience was with the live chat. It didn't work on OSX (Safari without add-ons) or my iPad. I also had to scroll down to find what I was looking for. But if you reach someone they give quick and clear answers.

Pricing is comparable with AWS. They do not offer a free tier. To find out if you are comfortable working with them and if they like what you do, you will need to pay. It's a good option, and it's also very affordable. It is far more affordable than S3/Cloud Front when you look at their CDN. No requests or other stuff to consider. You only pay bandwidth and storage. Simple pricing.

- Affordable pricing model - Many systems (High CPUs, high memory ,...)

- There is no free tier

You have the option to rent servers. You will need to manage them, compensate downtimes and plan for recovery and crashes. This is a viable option if your team already does this kind of work and you have the funds to buy more servers than you actually use (backup systems, spare systems if necessary to scale ...)).

If you only want to write code and have no experience with system administration dedicated servers are no option for you. You don't need to think about dedicated servers if you think you can just purchase a box and watch what happens.

Pros:

- Could be cheaper - Full control of your environment

Cons:

- This option is likely to be more expensive. - It will require more work. - Slower provisioning.

Overall Ranking

Dedicated hosting is a risky business, but it could be an option if the time and effort you put into it is acceptable.

Heroku is a great choice for those who don't require block storage or other stuff. They are doing a great job making your life as developer simple. - You may need block storage, or other features that Heroku doesn't offer. Rackspace. Better pricing model, it is that simple for me. AWS is for you if you need many additional services than what Heroku or Rackspace offer.
Evina

These cloud providers offer various levels of services, including "hosting your app" and "just tell us exactly what you need. However, be prepared to spend for everything you can.

The differences between Rackspace or AWS are not too significant. Rackspace has a clear

pricing model and better documentation.

Combining Providers

You can always combine and mix different providers. Rackspace Files can be used with your Heroku frontend and your workers and AWS storage. Depending on your preferences you could get the best out of all clouds. On the other hand it is, at least in my opinion, a disadvantage if you have to look in three different administration consoles, have three different bills and contact three different support teams if something goes wrong and you do not know what exactly happens.

Hosted services can be used to prevent vendor lockins on AWS. Most offer a free tier via their homepage.
You can also get a lot of EC2 instances and set up your own services. However, this is not something I recommend. You'll have to do the entire sysops work and will likely pay more than you would if you just went for the AWS service.

Drupan-hosted

AWS and Rackspace will be my main options. While payment is difficult in Europe, Stripe has not yet arrived, the free tier allows you to test the service without taking too much risk. Rackspace's CDN service is the only reason I will use it. It's just a lot better than Cloud Front and S3. Especially if you consider that you do not get statistics based on buckets and domains. I would not be able, therefore, to offer usage-based pricing for my customers. I would need to assume that x connections are required for every yGB transferred. I find this unfair.

>> posted on December 12, 2012 in #software-engineering